ARTICLES
https://doi.org/10.1038/s43588-022-00231-2

nature computational science

Check for updates

OPEN

Towards practical and robust DNA-based data archiving using the yin–yang codec system

Zhi Ping[1,2,3,4,11], Shihong Chen[2,3,5,11], Guangyu Zhou[6,7,11], Xiaoluo Huang[1,11], Sha Joe Zhu[8], Haoling Zhang[1,2,3,4], Henry H. Lee[6], Zhaojun Lan[9], Jie Cui[2,3,5], Tai Chen[2,3,5], Wenwei Zhang[1,2], Huanming Yang[1,2,3], Xun Xu[1,2,4,5] ✉, George M. Church[3,6,10] ✉ and Yue Shen[1,2,3,4] ✉

DNA is a promising data storage medium due to its remarkable durability and space-efficient storage. Early bit-to-base transcoding schemes have primarily pursued information density, at the expense of introducing biocompatibility challenges or decoding failure. Here we propose a robust transcoding algorithm named the yin–yang codec, using two rules to encode two binary bits into one nucleotide, to generate DNA sequences that are highly compatible with synthesis and sequencing technologies. We encoded two representative file formats and stored them *in vitro* as 200 nt oligo pools and *in vivo* as a ~54 kbps DNA fragment in yeast cells. Sequencing results show that the yin–yang codec exhibits high robustness and reliability for a wide variety of data types, with an average recovery rate of 99.9% above 10^4 molecule copies and an achieved recovery rate of 87.53% at $\leq 10^2$ copies. Additionally, the *in vivo* storage demonstration achieved an experimentally measured physical density close to the theoretical maximum.

DNA is an ancient and efficient information carrier in living organisms. At present, it is thought to have great potential as an alternative storage medium because standard storage media can no longer meet the exponentially increasing data archiving demands. Compared with common information carriers, the DNA molecule exhibits multiple advantages, including extremely high storage density (estimated physical density of 455 EB per gram of DNA[1]), extraordinary durability (half-life >500 years (refs. [2,3])) and the capacity for cost-efficient information amplification.

Many strategies have been proposed for digital information storage using organic molecules, including DNA, oligopeptides and metabolomes[4–8]. Since current DNA sequencing technology has advantages in terms of both cost and throughput, storing digital information using DNA molecules remains the most well-accepted strategy. In this approach, the binary information from each file is transcoded directly into DNA sequences, which are synthesized and stored in the form of oligonucleotides or double-stranded DNA fragments *in vitro* or *in vivo*. Then, sequencing technology is used to retrieve the stored digital information. In addition, several different molecular strategies have been proposed to implement selective access to portions of the stored data, to improve the practicality and scalability of DNA data storage[9–11].

However, the use of basic transcoding rules (that is, converting [00, 01, 10, 11] to [A, C, G, T]) generates some specific patterns in DNA sequences that result in challenges regarding synthesis and sequencing[9,12,13]. For example, single-nucleotide repeats (homopolymers) longer than 5 nt might introduce a higher error rate during synthesis or sequencing[14,15]. Meanwhile, because of the nature of complementary base pairing (with A pairing to T and G to C), DNA molecules may form structures such as hairpins or topological pseudoknots (i.e., secondary structure), which can be predicted by calculating the free energy from its sequence. It is reported that DNA sequences with stable secondary structure can be disadvantageous for sequencing or when using PCR for random access to and backup of stored information[16–19]. Additionally, DNA sequences with GC content <40% or >60% are often difficult to synthesize. Therefore, the length of homopolymers (in nt), the secondary structure (represented by the calculated free energy in kJ mol^{-1}) and the GC content (in %) are three primary parameters for evaluating the compatibility of coding schemes.

Previous studies on transcoding algorithm development have attempted to improve the compatibility of the generated DNA sequences. Early efforts, including those of Church et al. and Grass et al., introduced additional restrictions in the transcoding schemes to eliminate homopolymers, but this came at the expense of reduced information density[1,20,21]. Later studies pioneered other base conversion rules without compromising the information density. For example, the DNA Fountain algorithm adopted Luby transform codes to improve the information fidelity by introducing low redundancy as well as screening constraints on the length of homopolymers and the GC content while maintaining an information density of 1.57 bits nt^{-1} (refs. [6,22]). However, the major drawback is the risk of unsuccessful decoding when dealing with particular binary features due to fundamental issues with Luby transform codes. This approach relies on the introduction of sufficient logical redundancy, that is, at the coding level, for error tolerance to ensure successful decoding. This is different from physical redundancy, which refers to the synthesis of excess DNA molecules, that is, increasing

[1]BGI-Shenzhen, Shenzhen, China. [2]Guangdong Provincial Key Laboratory of Genome Read and Write, BGI-Shenzhen, Shenzhen, China. [3]George Church Institute of Regenesis, BGI-Shenzhen, Shenzhen, China. [4]Shenzhen Institute of Synthetic Biology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. [5]China National GeneBank, BGI-Shenzhen, Shenzhen, China. [6]Department of Genetics, Harvard Medical School, Boston, MA, USA. [7]Department of Molecular and Cellular Biology, Harvard University, Cambridge, MA, USA. [8]Big Data Institute, Li Ka Shing Centre for Health Information and Discovery, University of Oxford, Oxford, UK. [9]School of Mathematical Science, Capital Normal University, Beijing, China. [10]Wyss Institute for Biologically Inspired Engineering, Harvard University, Boston, MA, USA. [11]These authors contributed equally: Zhi Ping, Shihong Chen, Guangyu Zhou and Xiaoluo Huang. ✉e-mail: xuxun@genomics.cn; gchurch@genetics.med.harvard.edu; shenyue@genomics.cn

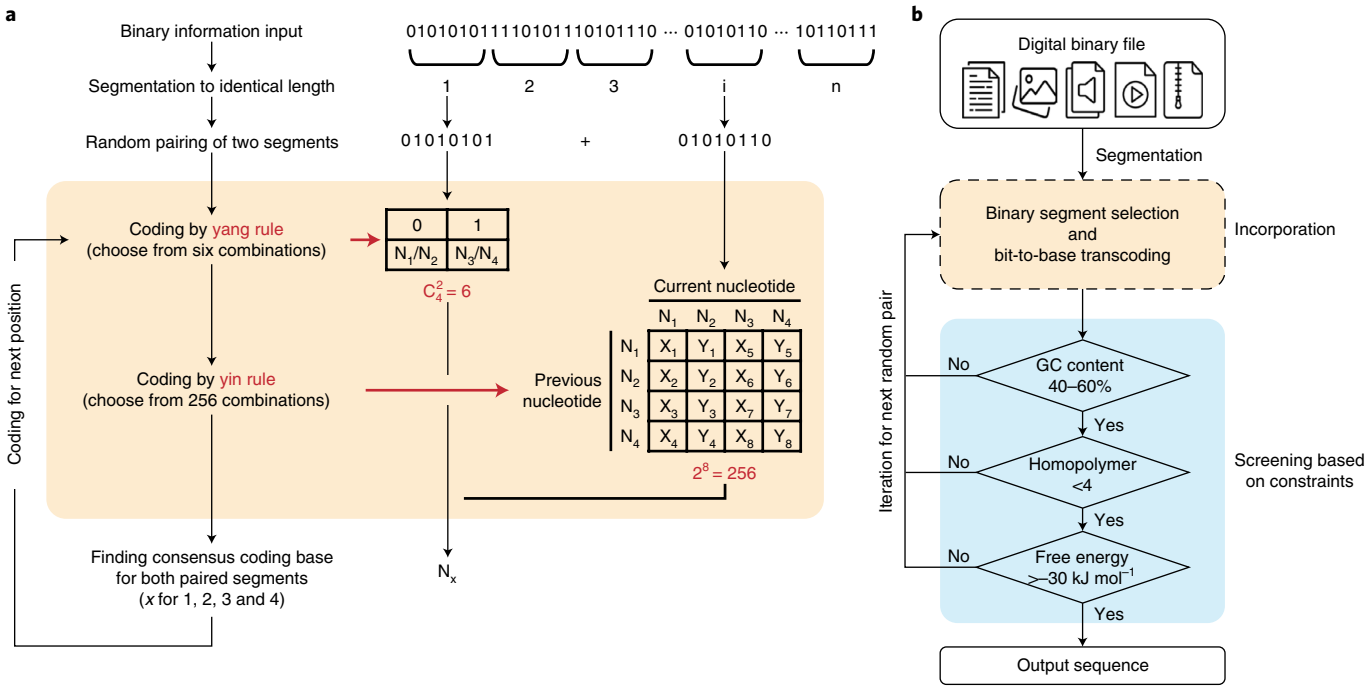

Fig. 1 | Principles of the YYC. a, The bit-to-base transcoding process of the YYC. N1, N2, N3 and N4 represent the nucleic acids A, T, C and G, respectively. X_j and Y_j represent different binary digits 0 and 1. When j is an integer chosen from 1 to 8, $X_j+Y_j=1$ and $X_j \times Y_j=0$ (that is, eight independent sets of X and Y, with X_j/Y_j being 1/0 or 0/1). C_4^2 means that the calculation for number of 2-combination in a set with 4 elements.**b**, A flowchart of the YYC encoding pipeline.

the copy number of DNA molecules for each coding sequence[23,24]. Reducing the logical redundancy could lead to a high probability of decoding failure, but excessive logical redundancy will decrease the information density and significantly increase the cost of synthesis[25]. Furthermore, specific binary patterns using these early algorithms may also create unsuitable DNA sequences, with either extreme GC content or long homopolymers (Supplementary Table 1). Therefore, developing a coding algorithm that can achieve high information density but, more importantly, perform robust and reliable transcoding for a wide variety of data types in a cost-effective manner is necessary for the development of DNA-based information storage in practical applications[25–27].

To achieve this goal, we propose herein the yin–yang codec (YYC) coding algorithm, inspired from the traditional Chinese concept of yin and yang, representing two different but complementary and interdependent rules, and we demonstrate its performance by simulation and experimental validation. The advantage of the YYC is that the incorporation of the yin and yang rules finally leads to 1,536 coding schemes that can suit diverse data types. We demonstrate that YYC can effectively eliminate the generation of long homopolymer sequences while keeping the GC content of the generated DNA sequences within acceptable levels. Two representative file formats (.jpg and .txt) were chosen for storage as oligo pools *in vitro* and a 54 kbps DNA fragment *in vivo* in yeast cells to evaluate the robustness of data recovery. The results show that YYC exhibits good performance for reliable data storage as well as physical density reaching the scale of EB per gram.

Results

The general principle and features of the YYC. In nature, DNA usually exists in a double-stranded structure. In some organisms such as phages, both strands encode genetic information to make the genome more compact. Inspired by this natural phenomenon, we used the basic theory of combinatorics and cryptography to develop a codec algorithm on the basis of Goldman's rotating encoding strategy[28,29]. Unlike other coding schemes developed using fixed mapping rules, the YYC provides dynamic combinatory coding schemes and can thus generate optimal DNA sequences to address the DNA synthesis and sequencing difficulties found when generating DNA sequences with long homopolymers, extreme GC content or complex secondary structure.

The general principle of the YYC algorithm is to incorporate two independent encoding rules, called 'yin' and 'yang', into one DNA sequence (called 'incorporation'), thereby compressing two bits into one nucleotide (Fig. 1a). Here, we use N1, N2, N3 and N4 to represent the four nucleic acids A, T, C and G, respectively. For one selected combinatory coding scheme, an output DNA sequence is generated by the incorporation of two binary segments of identical length. In the first step, the yang rule is applied to generate six different coding combinations. Then, in the yin rule, N1 and N2 are mapped to different binary digits, while N3 and N4 are also mapped to different binary digits independent of N1 and N2, leading to a total of 256 different coding combinations. Application of the yin and yang rules at one position will yield one and only one consensus nucleotide (Supplementary Fig. 1 and Supplementary Video 1). Meanwhile, according to the four different options for the previous nucleotide, the two groups (N1/N2 and N3/N4) also have independent options for the mapping to 0 and 1. Therefore, the incorporated yin and yang rules provide a total of 1,536 (6×256) combinations of transcoding schemes to encode the binary sequence. More details are described in the Supplementary information.

To demonstrate the compatibility of the YYC algorithm and quantify its featured parameters in comparison with other early DNA-based data storage coding schemes, the 1 GB data collection was transcoded by using the YYC as well as other early coding algorithms for comparison[1,20–22]. As shown in Table 1, the flexible screening process introduced after the incorporation of binary segments for both the YYC and DNA Fountain algorithms provides more possibilities for obtaining DNA sequences with desired GC content values between 40% and 60%. Like all the other coding algorithms, the YYC also introduces constraints to set the maximum homopolymer length at 4, considering computing resources

Table 1 | Comparison of DNA-based data storage schemes

			Church et al.	Goldman et al.	Grass et al.	Erlich et al.	Chen et al.	This work (YYC)
General attributes	Error correction strategy		No	Repetition	RS	Fountain	LDPC	RS
	Robustness against excessive errors		Yes	Yes	Yes	No	Yes	Yes
	Information density (bits nt^{-1})[a]		1[a]	1.58[a]	1.78[a]	1.98[a]	1.24[b]	1.95
	Physical density achieved (Ebytes g^{-1})	In vitro	0.001[a]	0.002[a]	0.025	0.21[a]	N/A[b]	2.25
		In vivo	N/A	N/A	N/A	N/A	270.7[b]	432.2[b]
Biotechnical compatibility	GC content (%) of sequences		2.5–100	22.5–82.5	12.5–100	40–60	N/A	40–60
	Maximum homopolymer length (nt)		3	1	3	4	N/A	4
	Ratio (%) of sequences with free energy >−30 kJ mol^{-1}		71.72	25.87	90.14	65.25	N/A	100

The schemes are presented chronologically based on publication date. The biotechnical compatibility is obtained according to in silico simulation of 1GB file collections (Methods). LDPC, low-density parity check. [a]Information based on data from ref. [22]. [b]Calculated value in the form of data coding in a DNA fragment integrated into the yeast genome. N/A means the data is not available in the corresponding studies.

as well as the technical limitations of DNA synthesis and sequencing. In addition, the YYC considers the secondary structure of the generated DNA sequences as part of the compatibility analysis, by rejecting all DNA sequences with free energy lower than −30 kcal mol^{-1}. In addition, the statistics of other features were analysed using the data collection and several test files in various data formats (Supplementary Figs. 2, 3 and 4 and Supplementary Tables 2, 3 and 4), suggesting that the YYC has no specific preference regarding the data structure and maintains a relatively high level of information density, ranging from 1.75 to 1.78 bits per base (Methods and Supplementary Fig. 2). For some cases, our simulation analysis suggests that a few (approximately seven) coding schemes from the collection of 1,536 might generate DNA sequences with identity between 80% and 91.85% (Supplementary Fig. 3), but at very low frequency.

Given these results, YYC offers the opportunity to generate DNA sequences that are highly amenable to both the 'writing' (synthesis) and 'reading' (sequencing) processes while maintaining a relatively high information density. This is crucially important for improving the practicality and robustness of DNA data storage. The DNA Fountain and YYC algorithms are the only two known coding schemes that combine transcoding rules and screening into a single process to ensure that the generated DNA sequences meet the biochemical constraints. The comparison hereinafter thus focuses on the YYC and DNA Fountain algorithms because of the similarity in their coding strategies.

In silico robustness analysis of YYC for stored data recovery. The robustness of data storage in DNA is primarily affected by errors introduced during 'writing' and 'reading'. There are two main types of errors: random and systematic errors. Random errors are often introduced by synthesis or sequencing errors in a few DNA molecules and can be redressed by mutual correction using an increased sequencing depth. Systematic errors refer to mutations observed in all DNA molecules, including insertions, deletions and substitutions, which are introduced during synthesis and PCR amplification (referred to as common errors), or the loss of partial DNA molecules. In contrast to substitutions (single-nucleotide variations, SNVs), insertions and deletions (indels) change the length of the DNA sequence encoding the data and thus introduce challenges regarding the decoding process. In general, it is difficult to correct systematic errors, and thus they will lead to the loss of stored binary information to varying degrees.

To test the robustness baseline of the YYC against systematic errors, we randomly introduced the three most commonly seen errors into the DNA sequences at a average rate ranging from 0.01% to 1% and analysed the corresponding data recovery rate in comparison with the most well-recognized coding scheme (DNA Fountain) without introducing an error correction mechanism. The results show that, in the presence of either indels (Fig. 2a) or SNVs (Fig. 2b), YYC exhibits better data recovery performance in comparison with DNA Fountain, with the data recovery rate remaining fairly steady at a level above 98%. This difference between the DNA Fountain and other algorithms, including YYC, occurs because uncorrectable errors can affect the retrieval of other data packets through error propagation when using the DNA Fountain algorithm. Although the robustness to systematic errors can be improved by introducing error correction codes, such as the Reed–Solomon (RS) code or low-density parity-check code[21,30,31], when the error rate exceeds the capability of such codes, the error correction will fail to function as designed. Furthermore, it is universally acknowledged that no efficient error correction strategies have been experimentally verified to be effective for insertions and deletions[32], let alone loss of the entire segment coding sequence. Therefore, in real applications, traditional error correction codes might play a limited role for improving robustness because of their inability to correct indels or the loss of the entire sequence.

As the other major factor for data recovery, the loss of partial DNA molecules can also affect the success rate of data retrieval[33]. Like early coding schemes (for example, those of Church et al., Goldman et al. and Grass et al.), the YYC is also designed like a linear block nonerasure code, with a linear relationship between data loss and the encoded sequence loss. Nevertheless, because of the convolutional binary incorporation of YYC, errors that cannot be corrected within one DNA sequence will lead to the loss of information for two binary sequences. In contrast, the DNA Fountain algorithm uses a different data retrieval strategy based on its grid-like topology of data segments, and theoretically, its data recovery cannot be guaranteed when a certain number of DNA sequences are missing[22]. In this work, in silico simulation of the data recovery rate in the context of a gradient of DNA sequence loss was performed. The results show that the YYC exhibits linear retrieval, as predicted. The data recovery percentage remains at 98% when the sequence loss rate is <2%. Even with 10% sequence loss, the YYC can recover the remaining ~90% of the data. In contrast, when the sequence loss rate exceeds 1.7%, the data recovery rate of the DNA

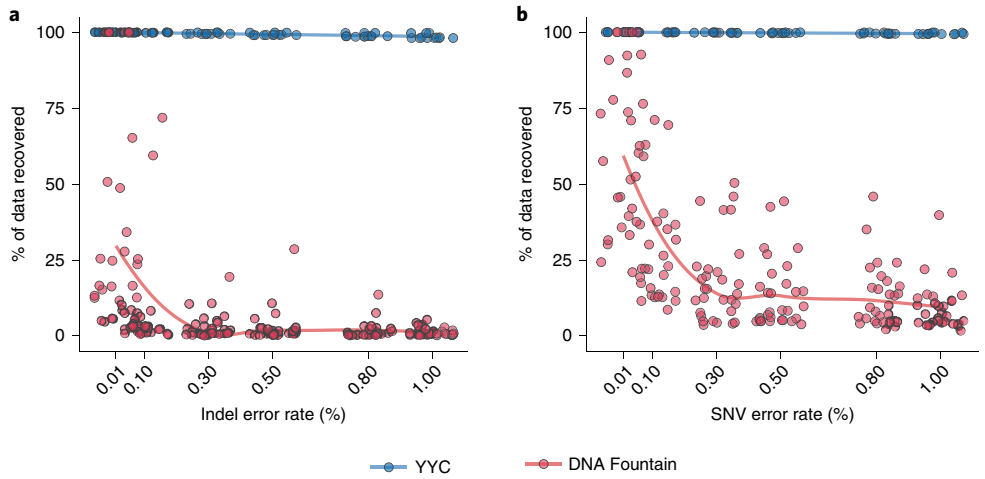

Fig. 2 | Robustness analysis for the YYC and DNA Fountain coding schemes. a,**b**, The binary data recovery rate of the YYC (blue) and DNA Fountain (red) coding strategies without any error-correction algorithm for indels (**a**) and SNVs (**b**) introduced randomly with an error rate of 0.01%, 0.1%, 0.3% 0.5%, 0.8% or 1%.

Fountain algorithm becomes highly volatile and drops significantly (Supplementary Table 5). Fountain codes function well for telecommunications and internet communications because the information transfer and verification are synchronous, thus giving the information source a chance to send more data packets for successful data recovery. However, the information writing (synthesis) and reading (sequencing) processes for DNA-based data storage are heterochronic, meaning that multiple, stepwise molecular manipulations are involved during the whole process. This makes the immediate transmission of additional data packets unrealistic for DNA-based data storage. Thus, although rateless codes including Fountain codes may improve the performance by adjusting their configuration and parameters, such coding schemes that suffer from the risk of uncertain decodability are not ideal for DNA-based data storage applications.

Experimental validation of the YYC with *in vitro* storage. To determine the compatibility of the YYC with current biochemical technologies, including DNA synthesis, PCR amplification and sequencing, we encoded three digital files (two text files, one each in English and Chinese, and an image) using the YYC and stored the encoded file in the form of 10,103 200 nt oligos *in vitro*. The sequence design of the oligos generated by the YYC transcoding is illustrated in Fig. 3.

Three oligo pools were synthesized for an experimental validation of *in vitro* storage. Pool 1 (P1) includes oligos with 25% logical redundancy. In comparison, two independent oligo pools (P2 and P3) of these three files, both transcoded by the DNA Fountain algorithm, were also synthesized using previously described settings (Fig. 3a):[22] Pool 2 (P2) includes 10,976 oligos encoding these three files individually, where it has been reported previously that logical redundancy is required for successful decoding, while pool 3 (P3) encodes the same files in a .tar archiving compressed package. The RS error-correction code was used in all three oligo pools.

The average molecule copy (AMC) number of the P1, P2 and P3 master pools is estimated to be ~10^7. A ten-fold serial dilution of P1, P2 and P3, with estimated AMC number from 10^6 to 10^0 for each oligo pool, was performed for sequencing to evaluate the minimal copy number of oligos required for successful file retrieval, as well as the robustness performance against DNA molecule loss (Fig. 3b). The sequencing results demonstrate that ~99.9% of the corresponding data from P1 can be recovered at AMC numbers above 10^3, with no preference regarding the specific data format (Fig. 3b and Supplementary Fig. 5c). As the AMC number decreases in magnitude, the decoding robustness shows an increase of instability. The average data recovery rate decreases to 71.2% at an AMC number of 10^2, ranging from 65.69% to 87.53% for each stored file. It drops further to below 10% when the AMC number is less than 10^1. In general, the YYC exhibits linear retrieval trend, which is positively correlated with the amount of data-encoding DNA molecules retained (Fig. 3c). For the DNA Fountain algorithm, the data recovery rate at an AMC number above 10^4 is comparable to that of the YYC, but it drops significantly at lower AMC numbers from 10^3 to the single-copy level (Fig. 3b). Especially for P3, the data was first .tar archived and then transcoded for storage. According to our experimental results, a maximum of 32.83% of the data package can be retrieved at lower levels of AMC number (Supplementary Data 1). However, the disruption of the compressed package leads to total loss of the original data. In addition, it has been suggested previously that most random errors introduced during synthesis or sequencing can be corrected by increasing the sequencing depth[34]. However, we found that, although lost sequences could be retrieved by such deep sequencing (Supplementary Fig. 6a), these sequences are at relatively low depth and contain more errors (Supplementary Fig. 6b). Therefore, such retrieved sequences are insufficient for valid information recovery. The current results suggest that loss of DNA molecules is the major factor affecting the data recovery rate, and that even high sequencing depth cannot improve the recovery rate if a certain amount of data-encoding DNA molecules are lost. In general, the relationship found between the information recovery rate and the sequence retention rate of each synthesized oligo pool in the *in vitro* experiment is consistent with that found in the in silico simulations, for the YYC and DNA Fountain algorithms.

To further investigate the compatibility of the coding schemes for different binary patterns from various files, we examined the performance of the YYC and DNA Fountain algorithms on test files in various formats. It is reported that information loss and decoding failure in DNA data storage can also result from original defects in the transcoding algorithms[26,35–38]. Therefore, increasing the logical redundancy could greatly improve the probability of successful decoding for all the coding schemes. However, too much logical redundancy requires the synthesis of more nucleotides and thus reduces the information density. Therefore, it is very important to keep the logical redundancy level in a controllable range for massive file archiving. Based on the transcoding simulations for these files, it is suggested that, especially for nonexecutable files, the

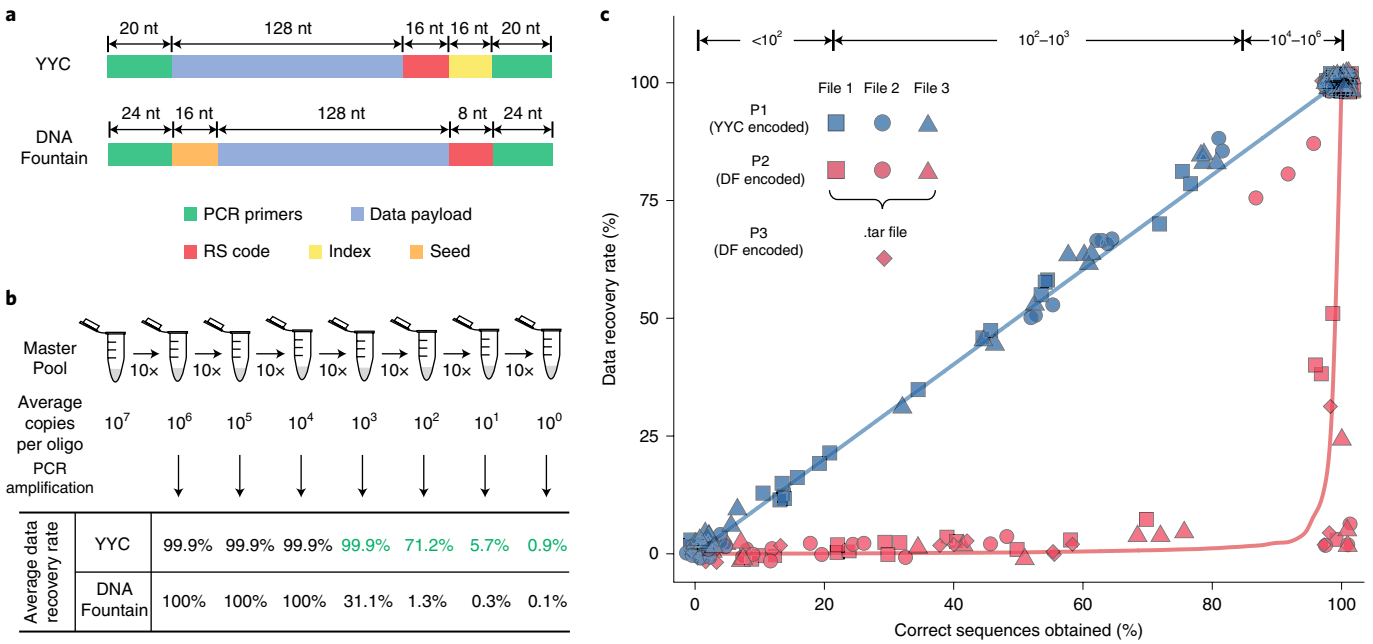

Fig. 3 | Experimental validation of *in vitro* binary data storage using the YYC and DNA Fountain coding strategies. a, The sequence design of the 200 nt oligo generated by the YYC and DNA Fountain algorithms for *in vitro* data storage. **b**, The serial dilution experiment of the synthesized oligo pool. The average copy number of each oligo sequence is calculated accordingly to the original oligo pool. The average data recovery rates are calculated based on the sequencing result of the PCR products of the diluted samples (the data recovery rates of YYC samples with low molecule copy number ($\leq 10^3$) is labeled with green color; the DNA molecule copy number for each sample after the PCR amplification exceeds 10^8). **c**, Analysis of the YYC and DNA Fountain (abbreviated as DF) algorithms by sequencing of corresponding diluted samples and calculation of the data recovery rate of each file encoded by YYC and DNA Fountain at the corresponding oligo copy number.

DNA Fountain algorithm exhibits variable requirements for the level of logical redundancy, leading to a varying information density (Supplementary Table 6). In contrast, the YYC coding scheme always requires a relatively low level of logical redundancy, resulting in more general compatibility with a broader range of file types and demonstrating a more stable information density.

Experimental validation of the YYC with *in vivo* storage. *In vivo* DNA data storage has attracted attention in recent years because of its potential to enable economical write-once encoding with stable replication for multiple data retrievals[30]. However, whether and the extent to which the robustness of a coding scheme can be maintained against spontaneous mutations or unexpected variations accumulated during long-term passaging of living cells has not been comprehensively investigated previously. Thus, we encoded a portion of a text file (Shakespeare Sonnet.txt) into a 54,240 bp DNA fragment containing 113 data blocks using the YYC and evaluated its potential data robustness for *in vivo* DNA data storage applications. The sequence design for each data block included a 456 bp data payload region and a 24 nt RS code region (Fig. 4a). The generated DNA fragment was first synthesized de novo into 60 of ~1 kbps subfragments and then assembled into 20 of ~2.8 kbps fragments (Methods). Taking advantage of the high homologous recombination efficiency of yeast, these fragments were directly transformed into yeast strain BY4741 together with the linearized low-copy centromeric vector pRS416 to enable one-step full-length DNA assembly *in vivo*. After ~1,000 generations by batch transfer of cell culture, we evaluated the robustness of the YYC scheme by subjecting 15 single colonies to whole-genome sequencing (Fig. 4b). First, in addition to indels or SNVs that could be introduced during construction or passaging of the cells, we also observed varying degrees of partial fragment loss from ~21.1 kbps to ~51.4 kbps among all 15 selected single colonies, leading to different levels of data recovery from 38.9% to 95.0% (Supplementary Table 7 and Fig. 4c,d). Since the observed indels or large deletions might lead to frameshifts of the data-encoding DNA sequence and subsequent decoding failure, a single-winner plurality voting strategy was applied to generate a consensus sequence from the reconstruction and alignment of multiple colonies (Fig. 4c). By doing so, we reconstructed a full sequence with 66 SNVs that cannot be corrected by the RS code introduced into the data block and fully recovered the stored data. In addition, to test the maximum physical density achievable in this study, we further integrated the constructed data-encoding DNA fragment into chromosome II of the yeast BY4741 genome. Therefore, for each resulting yeast cell, the data-encoding DNA is maintained at one single-copy level. By doing so, we successfully demonstrated that a physical density of ~432.2 EB g^{-1} can be achieved, suggesting a significant increase by three orders of magnitude than that demonstrated in prior work[22,39,40] (Table 1).

Discussion

The YYC transcoding algorithm offers several advantages. First, it successfully balances high robustness, compatibility and a considerable information density for DNA data storage compared with other early efforts. With the gradual popularization of DNA data storage, it is crucially important that the developed coding algorithms can perform robust and reliable transcoding for a wide variety of data types, especially for data with specific binary patterns. Before transcoding, compression algorithms such as Lempel–Ziv–Welch, Gzip or run-length encoding can be used to make the byte frequency[41] more balanced and avoid specific data patterns (Supplementary Table 1), thus improving the compatibility of the generated DNA sequences. However, because compression will change the original information structure, our results show that even partial loss of the DNA molecules will result in total failure to recover the compressed data. Current compression algorithms are not designed for DNA

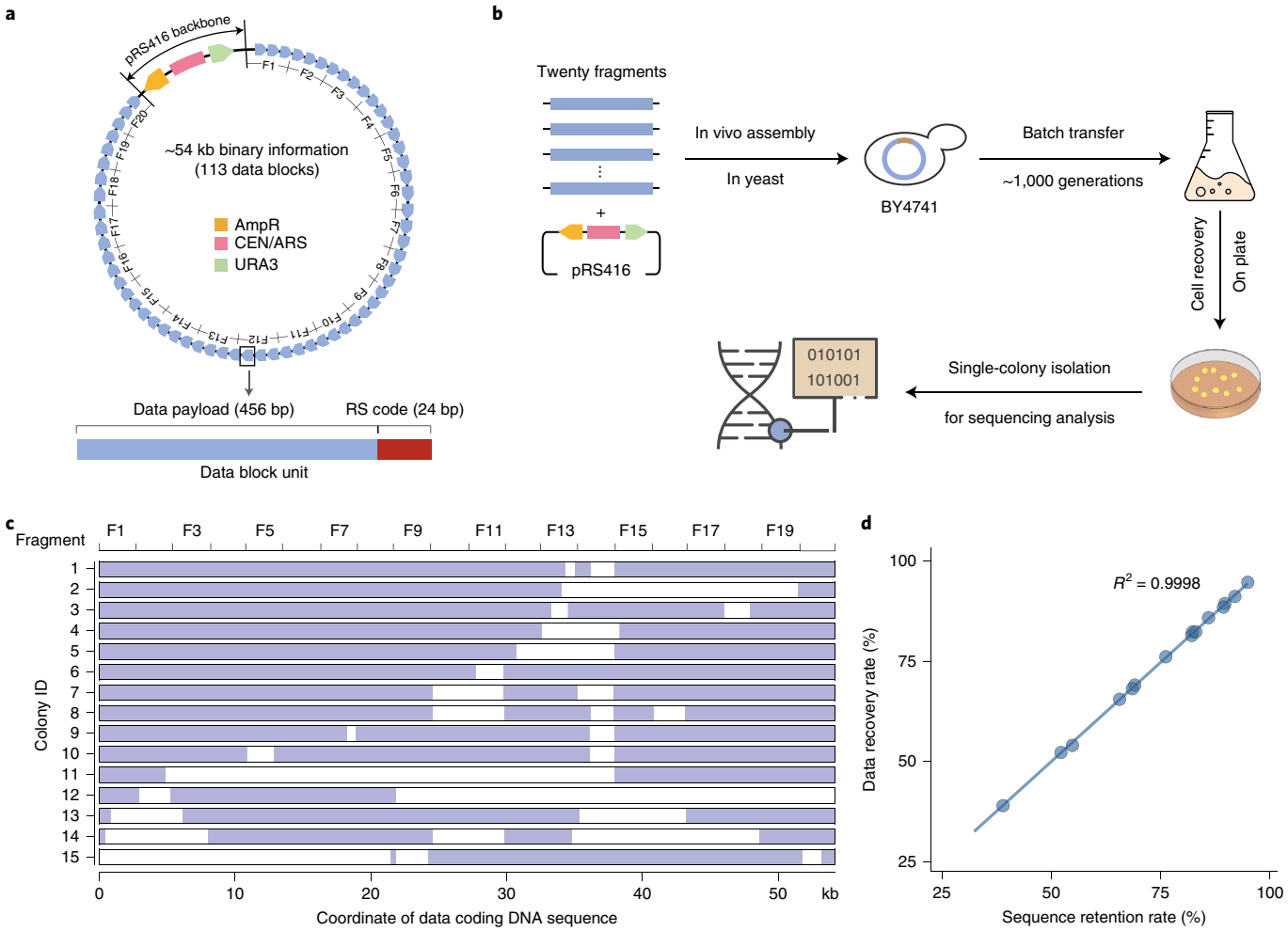

Fig. 4 | *In vivo* experimental validation. a, The design of the DNA sequence generated by the YYC for *in vivo* data storage in the form of a ~54 kbps data-encoding DNA fragment in the pRS416 vector. AmpR: ampicillin resistance gene; CEN: yeast centromere sequence; ARS: autonomously replicating sequence; URA3: a selective marker gene. **b**, The workflow of the data recovery analysis for *in vivo* data storage after batch transfer of ~1,000 generations. **c**, The fate of the ~54 kbps data-coding fragment in each selected yeast strain is indicated as preserved (light blue) or deleted (white). **d**, The correlation between the retention rate of the data-encoding DNA fragment and the data recovery in each selected yeast strain (dots). R^2 indicates the coefficient of determination. Solid line is the fitted curve obtained using the command ggplot:geom_point (), geom_smooth (method="lm", se=FALSE).

data storage, and further refinement can be performed to compress data appropriately for robust bit-to-base transcoding. Another potential advantage of the YYC is the flexibility of the rule incorporation from its 1,536 options. Considering broader application scenarios, the YYC offers the opportunity to incorporate multiple coding schemes for the transcoding of a single file, thus providing an alternative strategy for secure data archiving. Furthermore, coding schemes for DNA storage can be modularized into data transcoding, assignment of indices, error correction, redundancy, etc., thus providing more options to be combined freely be users. In our early work, we also demonstrated an integration system called 'Chamaeleo' in which the YYC could be used compatibly for bit-to-base encoding together with other modules[42]. Further optimization and functionalities can be incorporated into the system as well.

Our sequencing results show that the error rate of the synthesized oligo pools is ~1%, and in addition, ~1.2% of oligos are lost when mapping with the designed oligo sequence collections. There are two main steps in which systematic errors could be introduced: the synthesis of data-encoded oligos and PCR amplification to obtain a sufficient amount of DNA for sequencing. In general, random errors introduced during sequencing can be corrected easily by using a sufficient sequencing depth, but errors introduced during PCR amplification can be problematic. Error-correction codes can improve the information retrieval, but logically redundant sequences including both inner and outer codes can play a more important role in retrieving lost sequences and correcting errors for reliable DNA-based data storage. The length of the DNA sequence may also limit the pool capacity of a DNA storage system. These issues could be addressed in the future by using DNA synthesis technology with high stepwise efficiency, throughput and fidelity, which could yield longer DNA sequences and a high quantity of DNA and avoid amplification. For the demonstration of *in vivo* DNA storage, we find that there are random 1 nt indels and deletions of varying sizes in different data-coding regions across the selected single colonies, which could cause issues with data stability and recovery after long-term storage. Hence, it is critical that the coding strategy used should be able to retrieve as much information as possible. In addition, we also demonstrate herein that applying a voting strategy on a population of cells can further increase the possibility of fully recovering the stored information. Nevertheless, future efforts to improve the stability of exogenous artificial DNA in host cells is necessary to avoid unexpected information loss during passaging. The theoretical information density of DNA storage of 2 bits per base cannot be attained in real applications due to the setting of indices, the

error-correction strategy, intrinsic biochemical constraints and the technical limitations of the DNA synthesis and sequencing procedures[24]. The introduction of 'pseudobinary' segments in our study will also reduce the information density. Nevertheless, compared with another recent study on data storage using artificial yeast chromosomes in living yeast cells, the current results indicate better performance in terms of information density[30].

Methods

The YYC strategy. *Demonstration of the YYC transcoding principle.* In the example referred to as coding scheme no. 888 in Supplementary Fig. 1a, the yang rule states that [A, T] represents the binary digit 0 while [G, C] represents the binary digit 1. Meanwhile, the yin rule states that the local nucleotide (the current nucleotide to be encoded) is represented by the incorporation of the previous nucleotide (or 'supporting nucleotide') and the corresponding binary digit (Supplementary Fig. 1b). During transcoding, these two rules are applied respectively for two independent binary segments and transcoded into one unique DNA sequence, while decoding occurs in the reverse order. For example, given an input signal formed of 'a' and 'b' of '10110011' and '01011101', respectively, the transcoding scheme will start with the first nucleotide in each segment. According to the yang rule, '1' in 'a' provides two options [C, G]. With the predefined virtual nucleotide in position 0 as 'A', the yin rule and '0' for 'b' also provide two options [A, G] (Supplementary Fig. 1a). Therefore, the intersection of these two sets generates the unique base [G] transcoding the first binary digit in these two segments. Similarly, the rest of the two segments can be converted into a unique nucleotide sequence (Supplementary Fig. 1b and Supplementary Video 1). Note that switching the binary segments will change the transcoded result, which means that [a: Yang, b: Yin] and [b: Yang, a: Yin] will result in the generation of completely different DNA sequences. Generally, only one, fixed incorporated coding scheme is selected to transcode each dataset. Nevertheless, multiple coding schemes can be used for transcoding in encryption applications, where the corresponding information describing the coding schemes used would be stored separately.

Incorporation of the YYC transcoding pipeline. Considering the features of the incorporation algorithm, binary segments containing excessively imbalanced 0s or 1s will tend to produce DNA sequences with extreme GC content or undesired repeats. Therefore, binary segments containing a high ratio of 0 or 1 (>80%) will be collected into a separate pool and then selected to incorporate with randomly selected binary segments with normal 0-to-1 ratios.

Constraint settings of the YYC transcoding screening. In this study, a working scheme named 'YYC-screener' is established to select valid DNA sequences. By default, the generated DNA sequences (normally ~200 nt) with a GC content >60% or >40%, carrying >6-mer homopolymer regions or possessing a predicted secondary structure of <-30 kcal mol^{-1} are rejected. Then, a new run of segment pairing will be performed to repeat the screening process until the generated DNA sequence meets all the screening criteria. Considering that DNA sequencing and synthesis technologies continue to evolve rapidly, the constraint settings are designed as nonfixed features to allow user customization. In this work, the constraints are set as follows: a GC content between 40% and 60%, a maximum homopolymer length <5 and a free energy ≥ -30 kcal mol^{-1} (the free energy of the secondary structure is calculated using Vienna RNA version 2.4.6).

In silico transcoding simulation. *Computing and software.* All encoding, decoding and error analysis experiments were performed in an Ubuntu 16.04.7 environment running on an i7 central processing unit with 16 GB of random-access memory using Python 3.7.3.

Input files and parameters for simulation. The test files included 113 journal articles (including images and text), 112 .mp3 audio files from *Scientific American* and the supplementary video files from 33 journal articles.

To compare the compatibility of the different coding schemes, all the test files were transcoded by using Church's code, Goldman's code, Grass' code, DNA Fountain and the YYC in the integrated transcoding platform 'Chamaeleo' that we developed[42]. The segment length of the binary information was set as 32 bytes. For Church's code, Goldman's code and Grass' code, the original settings as previously reported were used in this study. For the DNA Fountain and YYC algorithms, the constraints were set as follows: a GC content of 40–60% and a maximum allowed homopolymer length of 4. For the free energy constraint in the YYC algorithm, the cutoff for probe design was set as -13 kcal mol^{-1} for a ~20 nt DNA sequence, and considering a length of the data-coding DNA of 160 nt, we adjusted the cutoff to -30 kcal mol^{-1} (refs. [43,44]).

Additional transcoding simulation tests were performed to evaluate the robustness and compatibility of the DNA Fountain and YYC algorithms. The DNA Fountain source code was used to perform encoding and decoding tests on nine different file formats and ten bitmap images with the default parameter settings (c-dist = 0.025, delta = 0.001, header size = 4, homopolymer = 4, GC = 40–60%) with minimum decodable redundancy. The oligo length of both strategies was set as 152 bases with indices or seeds for data retrieval and without error-correction codes. To determine the minimum redundancy required for file decoding, a test interval of minimum redundancy was set as 1%, and the maximum redundancy allowed was 300%. In some cases, the process terminated with a system error, which might be caused by stack overflow.

Experimental validation. *File encoding using the YYC and DNA Fountain algorithms.* The binary forms of three selected files (9.26×10^5 bits, 7.95×10^5 bits and 2.95×10^5 bits) were extracted and segmented into three independent 128 bit segment pools. A 16 bit RS code was included to allow the correction of up to two substitution errors introduced during the experiment. Next, four 144 bit binary segments (data payload + RS code) were used to generate a fifth redundant binary segment to increase the logical redundancy. Then, another 16 bit index was added into each binary segment to infer its address in the digital file and in the oligo mixture for decoding. Coding scheme no. 888 from the YYC algorithm was applied to convert the binary information into DNA bases. The aforementioned 'YYC-screener' was used to select viable DNA sequences. Eventually, 8,087 of 160 nt DNA sequence segments were generated. To allow random access to each file, a pair of well-designed 20 nt flanking sequences were added at both ends of each DNA sequence. Finally, an oligo pool containing 10,103 single-stranded 200 nt DNA sequences was obtained.

For DNA Fountain, the recommended default settings from its original report[22] (c-dist = 0.1, delta = 0.5, header size = 4, homopolymer = 4, GC = 40–60%), with the exception of redundancy, were used to generate the DNA oligo libraries. The minimum redundancy to ensure successful decoding was determined. Therefore, 13%, 22%, 73% and 12% logical redundancy was added for a .tar archiving compressed file, text1, text2 and image files, respectively. Finally, an oligo library encoding a .tar archiving compressed file (9,185 sequences) and an oligo library encoding the mixed three individual files (10,976 sequences) were obtained.

A part of one text file (~13 kB), was transcoded into DNA sequences by YYC for *in vivo* storage using a similar procedure, but the binary segment length was set as 87 bytes (or 456 bits). As described in the main text, the sequence was divided into 113 data blocks of 456 nt each. To increase the fidelity, a 24 nt RS code was added. The total data payload region as double-stranded DNA for *in vivo* storage is $(456+24)\times113=54{,}240$ bp.

Synthesis and assembly. The three oligo pools were outsourced for synthesis by Twist Biosciences and delivered in the form of DNA powder for sequencing.

For *in vivo* storage, the 54,240 bp DNA fragment was first segmented into 20 subfragments (2,500–2,900 bp) with overlapping regions and then further segmented into building blocks (800–1,000 bp, hereafter referred to as blocks). For each block, 20 of 80-nt oligos were synthesized with a commercial DNA synthesizer (Dr. Oligo, Biolytic Lab Performance) and then assembled into blocks by applying the polymerase cycling assembly method using Q5 High-Fidelity DNA Polymerase (M0491L, NEB) and cloned into an accepting vector for Sanger sequencing. Then, the sequencing-verified blocks were released from their corresponding accepting vector by enzymatic digestion for the assembly of subfragments by overlap extension (OE-)PCR. Gel purification (QIAquick gel extraction kit, 28706, QIAGEN) was performed to obtain the assembled subfragments. By transforming all 20 subfragments (300 ng each) and the low-copy accepting vector pRS416 into BY4741 yeast using LiOAc transformation[45] and taking advantage of yeast's native homologous recombination, the full-length ~54 kb DNA fragment was obtained. After 2 days of incubation on selective media (SC-URA, 630314, Clontech) at 30 °C, 16 single colonies were isolated for liquid culturing in YPD (Y1500, Sigma) before sequencing. One of the colonies showed very low target region coverage and was excluded from further analysis.

For the *in vivo* storage demonstration via genome integration, the full assembled fragment was inserted right after gene *YBR150C* on chromosome II with the LEU2 marker for selection via yeast transformation. The transformants were recovered on SC-Leu plates (SC-LEU, 630310, Clontech). Three positive colonies were isolated for genomic DNA extraction and sequencing.

Library preparation and sequencing. For library preparation of the synthesized oligo pool, the DNA powder was first dissolved in double-distilled water (ddH_2O) to obtain a standard solution, with an average of 10^7 molecules μL^{-1} per oligo for each synthesized oligo pool. Then, the standard solution was serially diluted by 10-fold to create the seven working solutions (WSs) of WS6 to WS0 with average concentration of 10^6 to 10^0 DNA molecules μL^{-1}, respectively, for each oligo pool. Then, each WS was amplified by PCR with three technical replicates to obtain the amplified product for P2 and each of the three different files for P1 and P3. PCR amplification was performed using 25 μL 2× Q5 High-Fidelity DNA Polymerase master mix (M0491L, NEB), 2 μL forward and reverse primer pairs each (10 μM each), 1 μL template DNA and 20 μL ddH_2O added to a final reaction volume of 50 μL. To obtain a sufficient amount of product for later sequencing, the PCR thermal cycler programme settings for P1 and P3 were as follows: 98 °C for 5 min; 23, 27, 32, 36, 40, 44 and 48 cycles of 98 °C for 10 s, 62 °C for 15 s and 72 °C for 10 s; and final extension at 72 °C for 2 min. The PCR settings for P2 were almost the same, but the annealing temperature was 60 °C for DF-F1 (Forward

primer 1) and 58 °C for DF-F2 (Forward primer 2) and DF-F3 (Forward primer 3). The concentrations of products were measured using gel electrophoresis and Qubit fluorometer, and corresponding molecules per microlitre values were also calculated (Supplementary Data 1). All amplified DNA libraries were then sequenced using DIPSEQ-T7 sequencing[46].

For *in vivo* storage, the methods for genomic DNA extraction and standard library preparation of the yeast colonies were described in previous studies[47]. The prepared samples were sequenced using the DNBSEQ-G400 (MGISEQ-2000) and DNBelab sequencing platform[48].

Data analysis. In total, >3 G PE-150 reads were generated for the *in vitro* storage experimental validation. Sequencing data with an average depth of 100× were randomly subsampled for information retrieval. The reads were first clustered and assembled to complete sequences for each type of oligo. Flanking primer regions were removed, DNA sequences were decoded to binary segments using the reverse operation of encoding and substitution errors were corrected using the RS code. The binary segments were reordered according to the address region. During this process, 'pseudobinary' segments were removed based on the address. The complete binary information was then converted to a digital file. The data recovery rate was calculated using $\frac{\text{successfully recovered binary segments}}{\text{total number of binary segments}}$ (Supplementary Data 1). For error analysis, sequencing data with average depth of 100×, 300×, 500×, 700× and 900× were randomly subsampled six times using different random seeds.

In total, >50 M PE-100 reads were generated for *in vivo* storage, in which the 10% low-quality reads (Phred score <20) by SOAPnuke were filtered[49]. Reads of the host genome were removed using samtools after mapping by BWA[50,51]. Short reads were then assembled into contigs by SOAPdenovo[52,53]. Blastn was used to find the connections between contigs[54]. A Python script was written to merge the contigs and obtain the assembled sequences for each strain. Multiple sequence alignment was conducted to align the assembled sequences by clustalW2 for the majority voting process to identify structural variations, insertions and deletions[55]. Pre-added RS codes were used for error correction of substitutions. The complete DNA sequence was decoded by reversing the operations of encoding to recover the binary information.

The physical density was calculated as

$$\frac{\text{Average information carried per nucleotide}}{\text{Average mass per nucleotide} \times \text{Average copy number} \times (1+\text{Redundancy percentage})},$$

where

$$\text{Average mass per nucleotide} = \frac{\text{Average molecular weight per nucleotide}}{\text{Avogadro constant}}$$

and

$$\text{Average information carried per nucleotide} = \frac{2 \times \text{Number of nucleotides in data payload region}}{\text{Total length used}}.$$

The average molecular weight per nucleotide is 330.95 g mol^{-1}, which is a constant.

For the *in vitro* demonstration in this work, the average copy number for effective data recovery is 100, the length of the data payload region is 128 nt, the total length is 200 nt and the redundancy is ~30% including the 'pseudobinary' sequence. Therefore, the physical density is calculated to be 1.79×10^{19} bits per gram of DNA, which equals 2.25×10^{18} bytes per gram of DNA.

For the *in vivo* demonstration in this work, the average copy number is 1 as the exogenous sequence is integrated to genome, the length of the data payload region is 51,528 bp and the total length is 54,240 bp. Therefore, the physical density is calculated to be 3.46×10^{21} bits per gram of DNA, which equals 4.322×10^{20} bytes per gram of DNA.

For Chen et al., according to their paper[30], the average copy number is 1, the average information carried per nucleotide is 1.19 bits nt^{-1}. Therefore, the physical density is calculated to be 2.707×10^{20} bytes per gram of DNA.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

Source data for all figures are provided with this paper. The sequencing raw data that support the findings of this study have been deposited in the CNSA (https://db.cngb.org/cnsa/) of the CNGBdb with accession code CNP0001650.

Code availability

The code package for the YYC is available in the GitHub repository (https://github.com/ntpz870817/DNA-storage-YYC) and Zenodo[56].

Received: 18 May 2021; Accepted: 18 March 2022;
Published online: 25 April 2022

References

1. Church, G. M., Gao, Y. & Kosuri, S. Next-generation digital information storage in DNA. *Science* **337**, 1628 (2012).
2. Allentoft, M. E. et al. The half-life of DNA in bone: measuring decay kinetics in 158 dated fossils. *Proc. Biol. Sci.* **279**, 4724–4733 (2012).
3. Bhat, W. A. Bridging data-capacity gap in big data storage. *Future Gener. Computer Syst.* **87**, 538–548 (2018).
4. Kennedy, E. et al. Encoding information in synthetic metabolomes. *PLoS ONE* **14**, e0217364 (2019).
5. Cafferty, B. J. et al. Storage of information using small organic molecules. *ACS Cent. Sci.* **5**, 911–916 (2019).
6. Koch, J. et al. A DNA-of-things storage architecture to create materials with embedded memory. *Nat. Biotechnol.* **38**, 39–43 (2020).
7. Choi, Y. et al. High information capacity DNA-based data storage with augmented encoding characters using degenerate bases. *Sci. Rep.* **9**, 6582 (2019).
8. Anavy, L., Vaknin, I., Atar, O., Amit, R. & Yakhini, Z. Data storage in DNA with fewer synthesis cycles using composite DNA letters. *Nat. Biotechnol.* **37**, 1229–1236 (2019).
9. Yazdi, S. M., Yuan, Y., Ma, J., Zhao, H. & Milenkovic, O. A rewritable, random-access DNA-based storage system. *Sci. Rep.* **5**, 14138 (2015).
10. Organick, L. et al. Random access in large-scale DNA data storage. *Nat. Biotechnol.* **36**, 242–248 (2018).
11. Tomek, K. J. et al. Driving the scalability of DNA-based information storage systems. *ACS Synth. Biol.* **8**, 1241–1248 (2019).
12. Kosuri, S. & Church, G. M. Large-scale de novo DNA synthesis: technologies and applications. *Nat. Methods* **11**, 499–507 (2014).
13. Shendure, J. et al. DNA sequencing at 40: past, present and future. *Nature* **550**, 345–353 (2017).
14. Van der Verren, S. E. et al. A dual-constriction biological nanopore resolves homonucleotide sequences with high fidelity. *Nat. Biotechnol.* **38**, 1415–1420 (2020).
15. Niedringhaus, T. P., Milanova, D., Kerby, M. B., Snyder, M. P. & Barron, A. E. Landscape of next-generation sequencing technologies. *Anal. Chem.* **83**, 4327–4341 (2011).
16. Kulski, J. K. in *Next Generation Sequencing: Advances, Applications and Challenges* (ed. Kulski, J. K.) pp. 3–60 (IntechOpen, 2016).
17. Kieleczawa, J. Fundamentals of sequencing of difficult templates–an overview. *J. Biomol. Tech.* **17**, 207–217 (2006).
18. Nelms, B. L. & Labosky, P. A. A predicted hairpin cluster correlates with barriers to PCR, sequencing and possibly BAC recombineering. *Sci. Rep.* **1**, 106 (2011).
19. Fan, H., Wang, J., Komiyama, M. & Liang, X. Effects of secondary structures of DNA templates on the quantification of qPCR. *J. Biomol. Struct. Dyn.* **37**, 2867–2874 (2019).
20. Goldman, N. et al. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature* **494**, 77–80 (2013).
21. 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. Engl.* **54**, 2552–2555 (2015).
22. Erlich, Y. & Zielinski, D. DNA Fountain enables a robust and efficient storage architecture. *Science* **355**, 950–954 (2017).
23. Organick, L. et al. Probing the physical limits of reliable DNA data retrieval. *Nat. Commun.* **11**, 616 (2020).
24. Dong, Y., Sun, F., Ping, Z., Ouyang, Q. & Qian, L. DNA storage: research landscape and future prospects. *Natl Sci. Rev.* **7**, 1092–1107 (2020).
25. Heckel, R., Shomorony, I., Ramchandran, K. & Tse, D. N. C. Fundamental limits of DNA storage systems. In *2017 IEEE International Symposium on Information Theory*. 3130–3134. (IEEE, 2017).
26. Feng, L., Foh, C. H., Jianfei, C. & Chia, L. LT codes decoding: Design and analysis. In *2009 IEEE International Symposium on Information Theory*. 2492–2496. (IEEE, 2009).
27. Matange, K., Tuck, J. M. & Keung, A. J. DNA stability: a central design consideration for DNA data storage systems. *Nat. Commun.* **12**, 1358 (2021).
28. Brualdi, R. A. *Introductory Combinatorics* (North-Holland, 1977).
29. Menezes, A. J., Katz, J., van Oorschot, P. C. & Vanstone, S. A. *Handbook of Applied Cryptography* (CRC Press, 1996).
30. Chen, W. et al. An artificial chromosome for data storage. *Natl Sci. Rev.* https://doi.org/10.1093/nsr/nwab028 (2021).
31. Fei, P. & Wang, Z. LDPC Codes for Portable DNA Storage. In *2019 IEEE International Symposium on Information Theory* 76–80. (IEEE, 2019).
32. Lenz, A., Siegel, P. H., Wachter-Zeh, A. & Yaakobi, E. Coding cver sets for DNA storage. *IEEE Trans. Inform. Theory* **66**, 2331–2351 (2020).
33. Ping, Z. et al. Carbon-based archiving: current progress and future prospects of DNA-based data storage. *Gigascience* https://doi.org/10.1093/gigascience/giz075 (2019).
34. Lee, H. H., Kalhor, R., Goela, N., Bolot, J. & Church, G. M. Terminator-free template-independent enzymatic DNA synthesis for digital information storage. *Nat Commun* https://doi.org/10.1038/s41467-019-10258-1 (2019).

35. Huang, W., Li, H. & Dill, J. Fountain codes with message passing and maximum likelihood decoding over erasure channels. In *2011 Wireless Telecommunications Symposium* 1–5. (IEEE, 2011).
36. Asteris, M. & Dimakis, A. G. Repairable Fountain codes. *IEEE J. Sel. Areas Commun.* **32**, 1037–1047 (2014).
37. Lázaro, F., Liva, G. & Bauch, G. Inactivation decoding of LT and Raptor codes: analysis and code design. *IEEE Trans. Commun.* **65**, 4114–4127 (2017).
38. Yang, L., et al. The Performance Analysis of LT Codes. (ed. Kim, Tai-hoonet. al) *Communication and Networking*, 227-235 (Springer Berlin Heidelberg, 2012).
39. Cai, Y. et al. Intrinsic biocontainment: multiplex genome safeguards combine transcriptional and recombinational control of essential yeast genes. *Proc. Natl Acad. Sci. USA.* **112**, 1803–1808 (2015).
40. Karim, A. S., Curran, K. A. & Alper, H. S. Characterization of plasmid burden and copy number in *Saccharomyces cerevisiae* for optimization of metabolic engineering applications. *FEMS Yeast Res* **13**, 107–116 (2013).
41. Wei-Jen, L., Ke, W., Stolfo, S. J. & Herzog, B. Fileprints: identifying file types by n-gram analysis. In *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*. 64–71. (IEEE, 2005).
42. Ping, Z. et al. Chamaeleo: an integrated evaluation platform for DNA storage. *Synth. Biol. J.* **1**, 1–15 (2021).
43. Noguera, D. R., Wright, E. S., Camejo, P. & Yilmaz, L. S. Mathematical tools to optimize the design of oligonucleotide probes and primers. *Appl. Microbiol. Biotechnol.* **98**, 9595–9608 (2014).
44. Yilmaz, L. S. & Noguera, D. R. Mechanistic approach to the problem of hybridization efficiency in fluorescent in situ hybridization. *Appl. Environ. Microbiol* **70**, 7126–7139 (2004).
45. Annaluru, N. et al. Total synthesis of a functional designer eukaryotic chromosome. *Science* **344**, 55–58 (2014).
46. Zhu, L. et al. Single-cell sequencing of peripheral mononuclear cells reveals distinct immune response landscapes of COVID-19 and influenza patients. *Immunity* **53**, 685–696 (2020).
47. Shen, Y. et al. Deep functional analysis of synII a 770-kilobaseÿÿsynthetic yeast chromosome. *Science* **355**, 6329 (2017).
48. Korostin, D. et al. Comparative analysis of novel MGISEQ-2000 sequencing platform vs Illumina HiSeq 2500 for whole-genome sequencing. *PLoS ONE* **15**, e0230301 (2020).
49. Chen, Y. et al. SOAPnuke: a MapReduce acceleration-supported software for integrated quality control and preprocessing of high-throughput sequencing data. *Gigascience* **7**, 1–6 (2018).
50. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* **25**, 1754–1760 (2009).
51. Danecek, P. et al. Twelve years of SAMtools and BCFtools. *Gigascience* **10**, giab008 (2021).
52. Luo, R. et al. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *Gigascience* **1**, 18 (2012).
53. Li, R. et al. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res.* **20**, 265–272 (2010).
54. Camacho, C. et al. BLAST+: architecture and applications. *BMC Bioinform.* **10**, 421 (2009).
55. Larkin, M. A. et al. Clustal W and Clustal X version 2.0. *Bioinformatics* **23**, 2947–2948 (2007).
56. Ping, Zhi, Zhang, Haoling & Zhu., Joe Towards practical and robust DNA-based data archiving using 'yin–yang codec'. *System* https://doi.org/10.5281/zenodo.6326563 (2022).

Acknowledgements

This work was supported by the National Key Research and Development Program of China (no. 2021YFF1200100, no. 2020YFA0712100), National Natural Science Foundation of China (no. 32101182) and Guangdong Provincial Key Laboratory of Genome Read and Write (no. 2017B030301011). We thank C. Hunter and C.-T. Wu from Harvard University and G. Ge from Capital Normal University for constructive discussions on the theoretical modelling. We thank the China National GeneBank (CNGB) for support with DNA fragment synthesis and assembly for the *in vivo* storage experiment.

Author contributions

Y.S., Z.P., S.C., G.Z. and X.H. designed the experiment. Z.P. and S.C. conducted simulation and data analysis. G.Z. conducted the sequencing data analysis. S.J.Z. and H.Z. wrote and improved the code of the software program. J.C. and T.C. conducted the *in vivo* DNA fragment assembly. Z.L., H.Z. and Z.P. conducted the theoretical justification. Z.P. and Y.S. drafted the manuscript. Z.P., H.Z. and Y.S. prepared the figures and tables. G.Z., S.J.Z., H.H.L., G.M.C. and Y.S. revised the manuscript. H.Y., X.X., G.M.C. and Y.S. supervised the study. All authors read and approved the final manuscript.

Competing interests

S.Z. is currently the founder of TAICHI AI Ltd, 20-22, Wenlock Road, London, England, N1 7GU. This work was completed when S.Z. was working at the University of Oxford and consulting for the BGI. G.M.C. has significant interests in Twist, Roswell, BGI, v.ht/PHNc, and v.ht/moVD. X.H., S.C., T.C., Y.S., X.X., and H.Y. have a patent filed with application number 16/858,295 and publication number 20200321079. The remaining authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43588-022-00231-2.

Correspondence and requests for materials should be addressed to Xun Xu, George M. Church or Yue Shen.

Peer review information *Nature Computational Science* thanks Manish K. Gupta and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Ananya Rastogi, in collaboration with the *Nature Computational Science* team. Peer reviewer reports are available.

nature research

Corresponding author(s): Xun Xu
George M. Church
Yue Shen

Last updated by author(s): Mar 16, 2022

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☐	☒ A description of all covariates tested
☐	☒ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	We published our custom python codes on Github and Zenodo. Source code and user manual are also available under: https://github.com/ntpz870817/DNA-storage-YYC and https://doi.org/10.5281/zenodo.6326563.
Data analysis	All encoding, decoding, and error analyzing experiments were performed in an Ubuntu 16.04.7 environment including an i7 CPU and 16 GB of RAM using Python 3.7.3, with our developed package "Chamaeleo" available under: https://github.com/ntpz870817/Chamaeleo. For NGS result analysis, we used BWA v0.7.13, samtools v0.1.19-44428cd and vSOAP 2.7.7. For multiple sequence alignment, we used clustalW2 (CLUSTAL v2.1).

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

The data that support the findings of this study have been deposited in the CNSA (https://db.cngb.org/cnsa/) of CNGBdb with accession code CNP0001650.
The figures-associated raw data is provided with the paper.
No restriction is on data availability.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	'Sample size' was determined by technical properties of DNA synthesis and sequencing. Different kinds and formats of data were chosen for DNA-based data storage, which is sufficient to prove the universality and robustness of this codec.
Data exclusions	Raw sequencing data was filtered under a pre-established criteria, 10% low-quality reads (Phred score < 20) by SOAPnuke were filtered.
Replication	The source data retrieval experiments were repeated twice for two batches of synthesized oligo pools. The attempts were always successful for the replications. The original data was successfully decoded in all technical repeats, or replications. The experiments of second batch were conducted about one year after the those of first batch.
Randomization	The simulations were run with precisely defined parameter value settings. Experimental validation also used precisely defined sequences and standard operation. Thus, randomization is not relevant to the study.
Blinding	The simulations were run with precisely defined parameter value settings. Experimental validation also used precisely defined sequences and standard operation. Thus, blinding is not relevant to the study.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

GigaScience, 8, 2019, 1–10

doi: 10.1093/gigascience/giz075
Review

REVIEW

Carbon-based archiving: current progress and future prospects of DNA-based data storage

Zhi Ping [1,†], Dongzhao Ma [1,†], Xiaoluo Huang[1,†], Shihong Chen [1], Longying Liu[1], Fei Guo[1], Sha Joe Zhu [2,*] and Yue Shen [1,*]

[1]Guangdong Provincial Key Laboratory of Genome Read and Write, Shenzhen Engineering Laboratory for Innovative Molecular Diagnostics, Guangdong Provincial Academician Workstation of BGI Synthetic Genomics, BGI-Shenzhen, Shenzhen 518083, China and [2]Big Data Institute, University of Oxford, Li Ka Shing Centre for Health Information and Discovery, Old Road Campus, Oxford OX3 7LF, UK

[*]**Correspondence address.** Sha Joe Zhu, Big Data Institute, University of Oxford, Li Ka Shing Centre for Health Information and Discovery, Old Road Campus, Oxford OX3 7LF, UK. Tel: +44-0-1865 287770; E-mail: sha.joe.zhu@gmail.com http://orcid.org/0000-0001-7566-2787; Yue Shen, Guangdong Provincial Key Laboratory of Genome Read and Write, Shenzhen Engineering Laboratory for Innovative Molecular Diagnostics, Guangdong Provincial Academician Workstation of BGI Synthetic Genomics, BGI-Shenzhen, Shenzhen 518083, China. Tel: +86-755-36307888; E-mail: shenyue@genomics.cn http://orcid.org/0000-0002-3276-7295

[†]These authors contributed equally to this work.

Abstract

The information explosion has led to a rapid increase in the amount of data requiring physical storage. However, in the near future, existing storage methods (i.e., magnetic and optical media) will be insufficient to store these exponentially growing data. Therefore, data scientists are continually looking for better, more stable, and space-efficient alternatives to store these huge datasets. Because of its unique biological properties, highly condensed DNA has great potential to become a storage material for the future. Indeed, DNA-based data storage has recently emerged as a promising approach for long-term digital information storage. This review summarizes state-of-the-art methods, including digital-to-DNA coding schemes and the media types used in DNA-based data storage, and provides an overview of recent progress achieved in this field and its exciting future.

Keywords: DNA digital storage; binary-DNA encoding scheme; *in vivo* DNA digital storage; *in vitro* DNA digital storage

Introduction

The concept of DNA-based data storage was introduced by computer scientists and engineers in the 1960s [1]. In one pioneering attempt, made in 1988 by Joe Davis in his seminal artwork "Microvenus" [2], an icon was converted into a string of binary digits, encoded into a 28-bp synthetic DNA molecule, and was later successfully sequenced to retrieve the icon [2]. Although Microvenus was originally designed for interstellar communications, it demonstrated that non-biological information could also be stored in DNA. Later, in the early 2000s, Bancroft et al. proposed a simple way to use codon triplets for encoding alphabets, suggesting great potential for DNA as a storage medium [3]. Now we ask the question: what makes DNA so inimitable for data storage?

Four unique biological features make DNA the focus of the next generation of digital information storage. First, DNA is remarkably stable compared with other storage media. With its double-helix structure and base-stacking interactions, DNA can persist 1,000 times longer than a silicon device [4], and survive for millennia, even in harsh conditions [5–8]. Second, DNA possesses a high storage density. Theoretically, each gram of single-stranded DNA can store up to 455 exabytes of data [9]. As storage strategies continue to improve, scientists have now achieved a

Received: 22 November 2018; **Revised:** 9 December 2018; **Accepted:** 3 June 2019

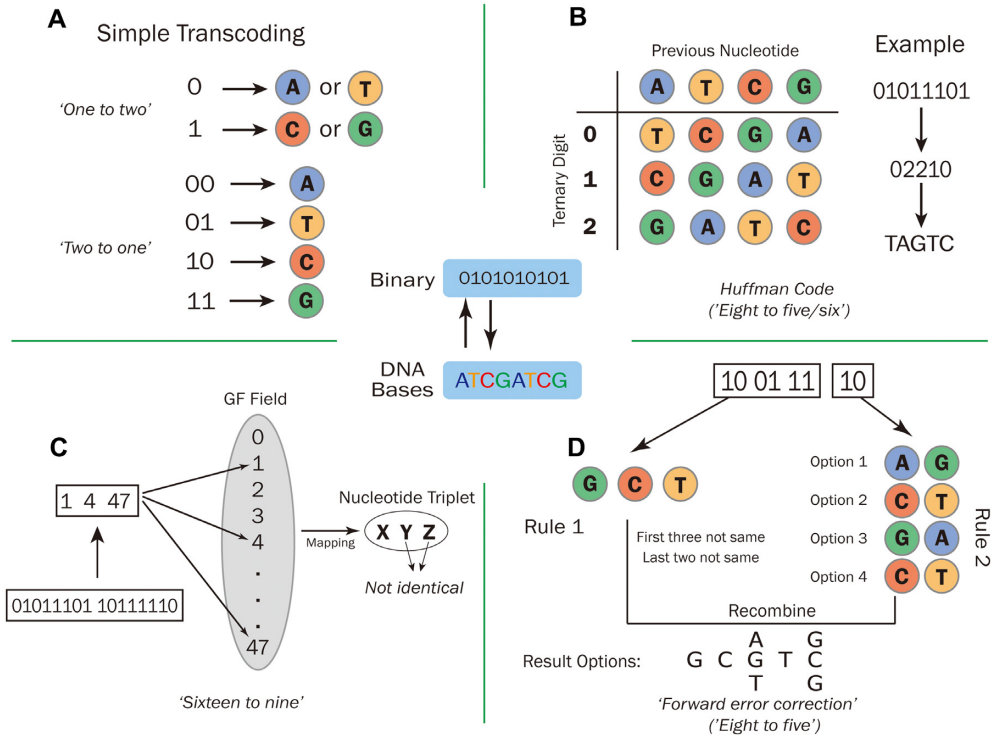

Figure 1: Binary transcoding methods used in DNA-based data storage schemes. (A) One binary bit is mapped to 2 optional bases [9]. Two binary bits are mapped to 1 fixed base [10]. (B) Eight binary bits are transcoded through Huffman coding and then transcoded to 5 or 6 bases [11]. (C) Two bytes (16 binary bits) are mapped to 9 bases [12]. (D) Eight binary bits are mapped to 5 bases [13].

density that could reach this theoretical limit. Third, DNA can be easily and rapidly replicated through the PCR, thereby providing the possibility for large-scale data backup. It should not be neglected that living cells are also perfect tools for *in vivo* information replication and backup. Last but not least, the biological properties of DNA enable current sequencing and chemical synthesis technologies to read and write the information stored in DNA, thereby making it an excellent material to store and retrieve data [9].

The recently announced Lunar Library™ project aims to create a DNA archive of a collection of 10,000 images and 20 books for long-term backup storage on the Moon. This highlights the advantage and immense potential of DNA as a medium for long-term digital data storage.

The accessibility of DNA-based data storage is mainly driven by 2 empowering techniques: DNA synthesis for "encoding," and DNA sequencing for "decoding" [14]. Typically, digital information is first transcoded into ATCG sequences using a predeveloped coding scheme. These sequences are then synthesized into oligonucleotides (oligos) or long DNA fragments to allow long-term storage. To retrieve the data, a DNA sequencing method is applied to obtain the original ATCG sequence from the synthesized DNA.

Overview of Current Coding Schemes for DNA-Based Data Storage

To summarize the findings of earlier studies, an optimal coding scheme usually outperforms in achieving 3 main features:

1) High fidelity—during data retrieval, there is a trade-off between accuracy and redundancy. While additional redundancy helps to improve accuracy, it also increases data size. Hence, to strike a balance, appropriate coding scheme and error correction strategies are applied to avoid and rectify errors induced during DNA synthesis or sequencing.
2) High coding efficiency—by having 4 elementary bases, DNA has the theoretical coding potential to store at least twice as much information in quaternary scaffolds as binary codes.
3) Flexible accessibility—from a computer science standpoint, stored data are expected to have random access. Lack of random access hampers attempts to scale up the data size because it will be impractical to sequence and decode the whole dataset each time when we only want to retrieve a small amount of data.

Correspondingly, proposed coding schemes are usually designed to fulfill all of the above characteristics. Generally, DNA-based data storage coding schemes can be differentiated by their binary transcoding methods (Fig. 1), or by the ways in which they add redundancy to increase fidelity (Fig. 2).

"Simple" code coding scheme

In 2012, Church et al. proposed a simple code to tackle errors generated by DNA sequencing and synthesis (e.g., repeated sequences, secondary structure, and abnormal GC content) [9]. By using the free base swap strategy (a "one-to-two" binary transcoding method; Fig. 1A), Church and colleagues encoded ~0.65 MB data into ~8.8 Mb DNA oligos of 159 nucleotides (nt) in length. Given the large amount of digital data that were successfully stored in DNA, this was considered to be a milestone study [15], and it also demonstrated the potential of DNA-based data storage to cope with the challenge of the information explosion. However, to allow its base swapping flexibility, this cod-

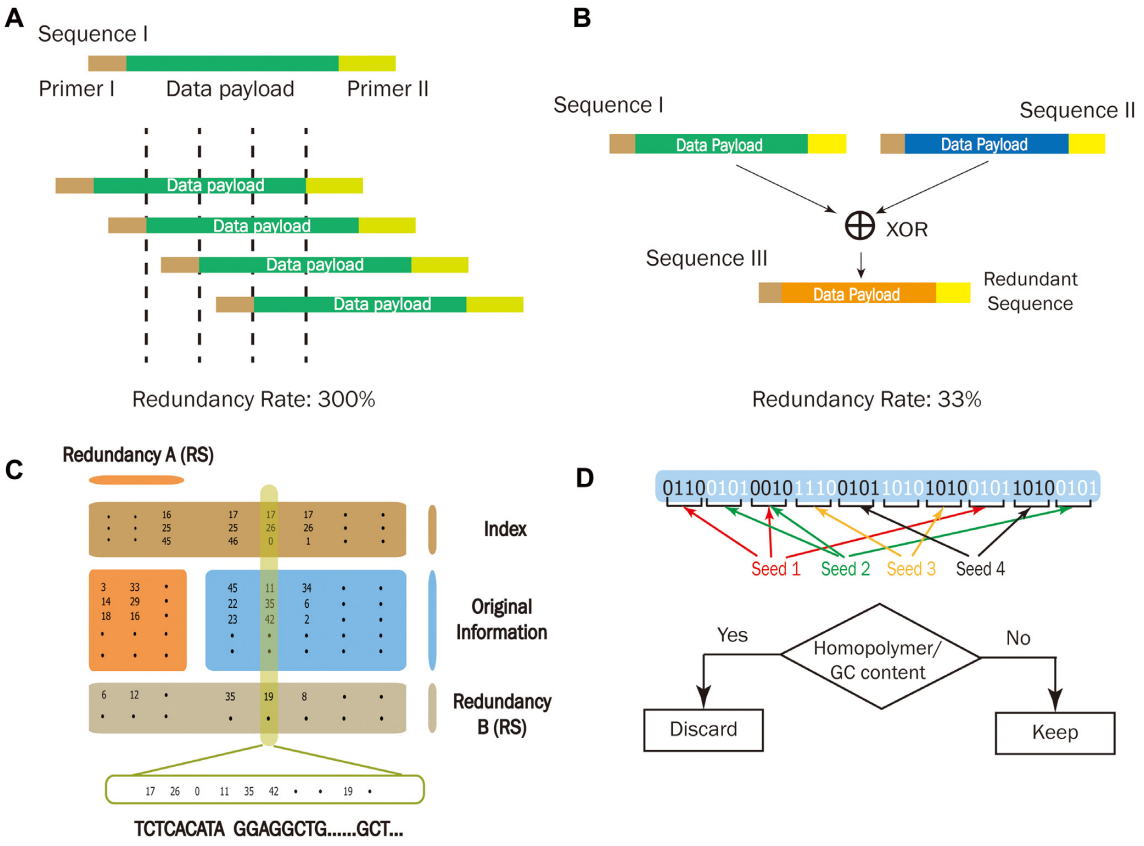

Figure 2: Redundancy types used in DNA-based data storage schemes. (A) Increasing redundancy by repetition. (B) Increasing redundancy by an exclusive-or (XOR) calculation. (C) Increasing redundancy using Reed-Solomon (RS) code for 2 rounds. (D) Increasing redundancy using fountain code.

ing scheme sacrifices information density by transcoding each binary code into 1 base. Later researchers have developed other coding strategies to overcome this issue while maintaining comparable performance.

Huffman coding scheme

Huffman code, developed by David Huffman in the 1950s, is considered to be an optimal prefixed code that is commonly used for lossless data compression. In 2013, Goldman and colleagues adopted the Huffman code in their coding scheme, which effectively improved the coding potential to 1.58 bits/nt [12]. Before transcoding into DNA nucleotides, binary data were first converted into ternary Huffman code, and then transcoded to DNA sequences by referring to a rotating encoding table (Fig. 1B). Each byte of the resulting data was substituted by 5 or 6 ternary digits (comprising the digits "0," "1," and "2" only) by Huffman's algorithm [16]. Encoding in this way, as per the rotating table, eliminates the generation of mononucleotide repeats and can compress the original data by 25–37.5%. For ASCII (American Standard Code for Information Interchange) text format files, this type of compression further outperforms by mapping the most common characters to 5-digit ternary strings [12]. However, the transcoding algorithm cannot prevent abnormal GC distribution when dealing with certain binary patterns. In addition, this coding scheme uses simple parity check coding to detect errors, and maintains a 4-fold coverage redundancy to prevent error and data loss (Fig. 2A). However, while the simple parity check coding can detect errors, it cannot correct them. Moreover, increased redundancy inevitably lowers the coding efficiency. Although not perfect, this work not only improved coding efficiency and prevented nucleotide homopolymers, but also introduced a strategy to ensure fidelity by adding redundancy.

Improved Huffman coding scheme

In 2016, Bornholt et al. improved Goldman's encoding scheme with an exclusive-or (XOR) encoding principle [13], using an XOR (⊕) operation to yield redundancy. As shown in Fig. 2B, every 2 original sequences, A and B, will generate a redundant sequence C by A ⊕ B. Therefore, with any 2 sequences (AB, AC, or BC), one can easily recover the third sequence. This coding scheme also provides the flexibility of redundancy according to the level of significance of particular data strands, namely, "tunable redundancy." It decreased the redundancy of the original data from 3-fold to half, providing an efficient way to ensure fidelity. In practice, this coding scheme successfully encodes 4 files with a total size of 151 KB and recovers 3 out of 4 files without manual intervention [13].

The need to amplify target files in a large-scale database suggests a necessity for random access in DNA-based data storage. Therefore, in 2018, Bornholt et al. put forward another error-free coding scheme that allowed users to randomly reach and recover individual files in a large-scale system. In this coding scheme, unique PCR primers are assigned to individual files after rigorous screening, thereby allowing users to randomly access their target file(s). A total of 200 MB data was successfully stored and recovered in their study, which set a new milestone by complementing the feasibility of storing large-scale data in DNA [14].

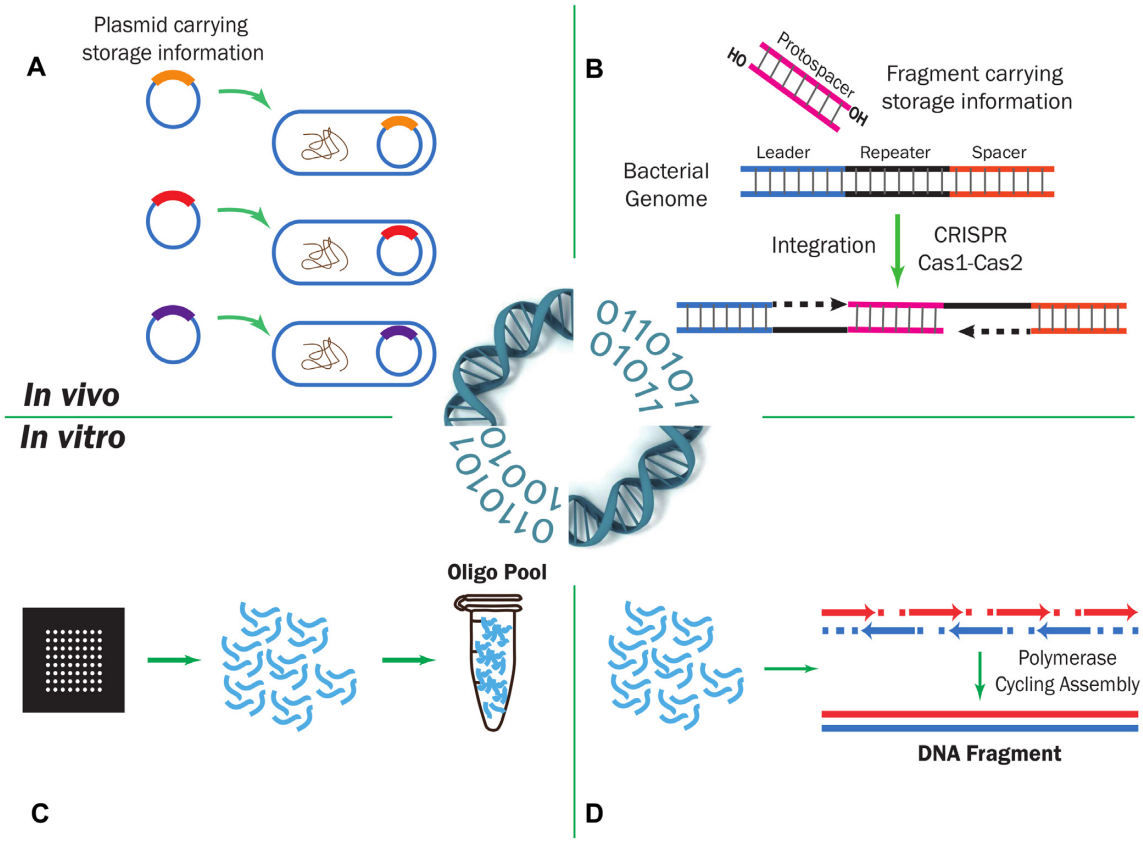

Figure 3: Two categories of DNA-based data storage application. (A) and (B) demonstrate 2 methods of *in vivo* DNA-based data storage; (C) and (D) demonstrate 2 methods of *in vitro* DNA-based data storage. (A) Array-based high-throughput DNA oligo analysis. DNA oligos carrying digital information are stored in the form of oligo pool. (B) DNA fragments synthesized by polymerase cycling assembly will carry the information to be stored. (C) Digital information inserted into a plasmid; plasmids are then transferred into bacterial cells. (D) DNA fragments carrying digital information are inserted into the bacterial genome using the CRISPR system using Cas1-Cas2 integrase.

A coding scheme based on Galois field and Reed-Solomon code

With special emphasis on error detection and correction, a coding scheme based on the Galois field (GF) and Reed-Solomon (RS) code [15] was proposed by Grass and colleagues in 2015 [17], improving potential data density to ~1.78 bits/nt. With the 2-byte (8×2 bits) fundamental information block, this coding scheme introduced a finite field (the GF) of DNA nucleotide triplets as its elements (Fig. 1C). To prevent mononucleotide repeats of >3 nt during encoding, the last 2 nucleotides of the triplet are varied, which can give 48 different triplets. A GF of 47 was used because 47 is the largest prime number smaller than 48. The information block is then mapped to the 3 elements in GF (47), i.e., 256^2 to 47^3. The RS code is applied in this scheme to detect and correct errors. As shown in Fig. 2C, 2 rounds of RS coding are applied horizontally and vertically to the matrix generated by GF transcoding, respectively.

In this pilot study, 83 KB of text data were encoded *in silico* [17]. Although the data size was not impressive, it underlined the necessity to apply error correction coding, and significantly enhanced coding efficiency. Moreover, error correction code from the information communication field was applied to DNA-based data storage for the first time.

A "forward error correction" coding scheme

Blawat and colleagues proposed a coding scheme to particularly tackle the errors generated during DNA sequencing, amplification, and synthesis (e.g., insertion, deletion, and substitution) [18]. The potential coding density was 1.6 bits/nt. Two reference coding tables are specified in advance. A 1-byte (8 bits) fundamental information block is assigned to a 5-nt DNA sequence, and the third and fourth nucleotide are swapped (Fig. 1D). Two other criteria are also applied to prevent mononucleotide repeats during this process: (i) the first 3 nucleotides should not be the same; and (ii) the last 2 nucleotides should not be the same. Consequently, an 8-bit data block (i.e., $2^8 = 256$ permutations for binary data) is transcoded into 704 different DNA blocks ($4^5 - 4^3 - 4^4$) [18]. These can be categorized into 3 clusters: clusters A and B of complete blocks (256 each), and cluster C of 192 incomplete blocks. Data can then be mapped to DNA blocks A and B as required, e.g., alternately mapped to A or B.

In this study, 22 Mb of data were successfully encoded and stored in an oligo pool. Those data were retrieved without error, thereby proving the feasibility of the "forward error correction" coding scheme. However, this was not the case for detecting and correcting single mutations. For example, "11100011" could be mapped to a DNA block "TGTAG." but if an A-to-T transversion

occurs, the DNA block will be changed to "TGTTG," which will give an error byte "11101111" after decoding.

Fountain code–based DNA-based data storage coding scheme

In 2017, Erilich and Zielinski used fountain code in their coding scheme [19]. Fountain code is a widespread method of coding information in communication systems and is well known for its robustness and high efficiency [20]. Fountain code is also known as a rateless erasure code, in which data to be stored are divided into k segments, namely, resource packets. A potentially limitless number of encoded packets can be derived from these resource packets. When it returns n ($n > k$) encoded packets, the original resource data will be perfectly recovered. In practice, n only needs to be slightly larger than k to yield greater coding efficiency and robustness for information communication [21].

Binary data nucleotide sequence transcoding is also carried out. A fundamental 2-bit to 1-nt transcoding table is adopted, in which [00, 01, 10, 11] is mapped to [A, C, G, T], respectively (Fig. 1A). First, original binary information is segmented to small blocks. These blocks are chosen according to a pre-designed pseudorandom sequence of numbers. A new data block is then created by the bitwise addition of selected blocks with random seeds attached and transcoded to nucleotide blocks according to the transcoding table. Mononucleotide repeats and abnormal GC content are prevented by a final verification step (Fig. 2D) [19].

The oligos in this coding scheme are correlated and have grid-like topology to realize extremely low but necessary redundancy. This study increased the theoretical limit of coding potential to an unprecedentedly high value of 1.98 bits/nt, and remarkably reduced the desired redundancy for error-free recovery of the source file. Moreover, the mechanism of random selection and validity verification ensures that long single-nucleotide homopolymers do not appear in the encoded sequence. However, in this coding scheme, the complexity level of encoding and decoding is not linearly correlated to the data size. Thus, decoding can be complicated and may require more resources and a longer computation time. However, although it is claimed that a 4% loss of total packets would not affect the recovery of the original file in the report, in terms of the features of DNA fountain code, loss of more packets may cause complete failure of recovery. If the ultimate aim is to permanently store the data, the amount of redundancy must be increased to ensure information integrity.

If we consider DNA-based data storage solely as an archiving process with high fidelity, then DNA fountain coding appears to be the only communication-based coding scheme. In DNA-based data storage and retrieval, the most common error is caused by a single-nucleotide mutation. To address this issue, most coding schemes create high redundancy to tackle the challenging conditions of current communication channels. However, these error correction algorithms require complex decoding procedures and large amounts of computing resources. Here, the use of a fountain coding scheme first shows that it is unnecessary to use error detection/correction algorithms, and this provides us with an alternative solution for improving the performance of DNA coding.

Overview of DNA-Based Data Storage Media

Currently, DNA-based data storage uses 2 main types of media to store encoded DNA sequences: *in vivo* and *in vitro*.

In vivo

In vivo DNA-based data storage was commonly adopted in pioneering DNA-based data storage work, such as the Microvenus project, which used bacteria as the storage medium [2]. In the 2000s, other research teams also proposed simple techniques for *in vivo* DNA-based data storage, e.g., the use of codon triplets to encode alphabets [22] or bits [23] by either transferring plasmids or introducing site-directed mutagenesis. Typically, encoded DNA sequences are first cloned into a plasmid and then transferred into bacteria (Fig. 3A). Therefore, the DNA sequences, and the information they carry, can be maintained in tiny bacteria and their billions of descendants.

Nevertheless, the capacity of bacteria for carrying plasmids is limited by the type and size of plasmid. In addition, plasmid mutation is quite common in bacteria. During bacterial replication, take *Escherichia coli* as an example, the spontaneous mutation rate is 2.2×10^{-10} mutations per nucleotide per generation, or 1.0×10^{-3} mutations per genome per generation [24], with a generation time of 20–30 minutes, which—after a few years—might ultimately alter the information stored.

Recently, Shipman et al. demonstrated a novel method to encode an image and a short movie clip into the bacterial genome using the clustered regularly interspaced short palindromic repeats–CRISPR-associated protein (CRISPR-Cas) system with Cas1-Cas2 integrase (Fig. 3B) [25]. Although, reportedly, the CRISPR-Cas system is not equally efficient to all sequences, this work greatly improved the capability of *in vivo* DNA-based data storage.

In vitro

In vitro DNA-based data storage is seen more frequently than the *in vivo* version in recent studies. The oligo library is one of the most popular forms (Fig. 3C), primarily because of the maturation of the array-based high-throughput oligo synthesis technique [26], which makes the synthesis of large numbers of DNA oligos more cost-effective.

During the synthesis process, each oligo is assigned a short tag, or index, because all oligos are mixed together for high-throughput synthesis and sequencing. The current oligo synthesis technique can generate, at most, 200-mers, with relatively high accuracy and purity [27]. Hence, the index should be as short as possible to save the information capacity in each oligo. Apparently, many more indices will be needed if more DNA oligo sequences are generated and mixed. However, similar to *in vivo* DNA-based data storage, the larger data size demands more DNA oligos for *in vitro* DNA-based data storage. This increases the size of indices in oligo and thus lowers the storage capacity and efficiency.

To overcome these problems, longer DNA fragments can be used instead of DNA oligos (Fig. 3D). In 2017, Yazdi et al. successfully encoded 3,633 bytes of information (2 images) into 17 DNA fragments, and recovered the image using homopolymer error correction [28]. Nevertheless, the current cost of DNA fragment synthesis is higher than that of oligo synthesis, which increases the overall cost of DNA fragment–based storage.

Above all, both *in vivo* and *in vitro* strategies have been used in current DNA-based data storage research. However, the nature of these 2 strategies demonstrates the use of different techniques and different application scenarios (Table 1). Although *in vivo* storage is a more complicated procedure than oligo pool synthesis in terms of backup cost, *in vivo* DNA-based data storage is more cost-effective. The cost of the *in vitro* method has

Table 1: Comparison of *in vivo* and *in vitro* DNA-based data storage

Parameter	*In vivo*	*In vitro*
Medium	PlasmidBacterial genome	Oligo libraryLong DNA fragment
Information writing	Cloning and gene editing	Oligo synthesis
Main cause for error generation	MutationSequencing	Error in synthesis/sequencing
Advantage	Long-term storageCost-effective backup	High throughputLow error rateEasy for manipulation
Disadvantage	Limited DNA sizeMutation during replication	DNA degradationCost of index region

been reduced with the development of array-based oligo synthesis and high-throughput sequencing. Considering long-term storage, DNA in an *in vivo* condition will degrade more slowly than *in vitro*. Nevertheless, errors induced by mutations during replication *in vivo* are more significant than those induced by synthesis because of the high accuracy of current DNA synthesis technology.

Other pioneering work goes beyond the aforementioned DNA-based data storage system. Song and Zeng proposed a strategy that they claim is able to detect and correct errors in each byte [29]. They transformed a short message into *E. coli* stellar competent cells and proved the reliability of their strategy; this was one of the first studies to evaluate the stability of *in vivo* storage. Lee et al. incorporated enzymatic DNA synthesis and DNA-based data storage principles, reporting an enzymatic DNA-based data storage strategy [30]. Nevertheless, the recent recombinase and CRISPR-Cas9 techniques cannot be neglected because they might also drive *in vivo* DNA-based data storage in diversiform. All of this research has laid a sound foundation for the global application of this novel storage medium.

Challenges of DNA-Based Data Storage

Although DNA sequencing and DNA synthesis techniques largely facilitated the increase in DNA-based data storage, challenges co-derived and spontaneously evolve as each paradigm shift occurs in these fields. Fig. 4 shows a timeline briefly summarizing the key breakthroughs in DNA synthesis and sequencing that have transformed the development of DNA-based data storage.

In the pre–high-throughput period, column-based oligo synthesis [31] and Sanger sequencing [32, 33] represented the dominant DNA synthesis and DNA sequencing techniques, respectively. At this stage, the high cost ($0.05–0.15 USD per nucleotide in 100-nt synthesis; $1 USD per 600–700 bp per sequencing read) and time-consuming nature of DNA sequencing (an automated Sanger sequencing machine reads 1,000 bases per day) [10, 26] remain the major challenges for DNA-based data storage, preventing its application on larger datasets. Therefore, studies during that time were only conducted as a proof-of-concept on a relatively small scale [2].

From 2000 onwards, on the completion of the Human Genome Project, both DNA synthesis and DNA sequencing techniques were transformed to the high-throughput scale. Array-based oligo synthesis gradually superseded column-based oligo synthesis and was widely commercialized [34–36], largely because of its relatively low cost ($0.00001–0.001 USD per nucleotide synthesis [10]). However, as oligo length increases—presumably because of potential false cross-hybridization during synthesis—the error rate also increases. Moreover, the length of synthesized oligonucleotides is limited to <200-mers; this is because the product yield drops as oligos are elongated thanks to limitations in the efficiency of chemical interactions. Although gene size (200–3,000 bp or above) array-based synthesis has been developed [37], these usually require additional steps for error correction, causing the final cost and time consumed to be high. Consequently, for cost-saving purposes and to reduce the complexity of DNA synthesis, the primary storage unit used in DNA-based data storage is <200 nt.

The concept of massively parallel sequencing (or next-generation sequencing [NGS]), a high-throughput sequencing method, was proposed in 2000 [38]. In the following years, sequencing by ligation and by synthesis became major players in the sequencing field. Multiple NGS platforms became commercially available (e.g., 454, Solexa, Complete Genomics), which paved the way for high-throughput DNA-based data storage. However, this emerging technique also comes with limitations. Most NGS platforms require *in vitro* template amplification with primers to generate a complex template library for sequencing. During this process, copying errors, sequence-dependent biases (e.g., in high-GC and low-GC regions and at long mononucleotide repeats), and information loss (e.g., methylation) are produced [9].

In 2012, Church and colleagues successfully demonstrated the first application of high-throughput DNA synthesis and NGS in DNA-based data storage [9]. It initiated rapid development of coding schemes incorporating NGS. Two of the most common goals at this stage were how to improve coding efficiency and how to correct sequencing errors.

While NGS remains dominant, real-time, single-molecule sequencing (or third-generation sequencing) is continually evolving [39, 40]. Despite its relatively high sequencing error rate (~10%), it is reportedly capable of long read-length sequencing, high-GC tolerant, and generates only random errors [28]. These characteristics mean it outperforms NGS counterparts and make it ideal for data retrieval in DNA-based data storage. In 2017, Yazdi et al. used Oxford Nanopore MinION technology to retrieve data stored in DNA, showing optimal robustness and high efficiency [28]. This study implies a possible shift from NGS to single-molecule sequencing because of its potential for compactness and stand-alone DNA data storage systems [13, 30]. Table 2 summarizes the frequently used sequencing platforms in DNA-based data storage. Recently, Oxford Nanopore Technologies announced plans to develop a "DNA writing" technique using their Nanopore technology. Using the same platform to both read and write, they claim it will be possible to selectively modify native bases and stimulate localized reactions, such as light pulses for encoding, which will provide real-time read and write capabilities for DNA-based data storage [41].

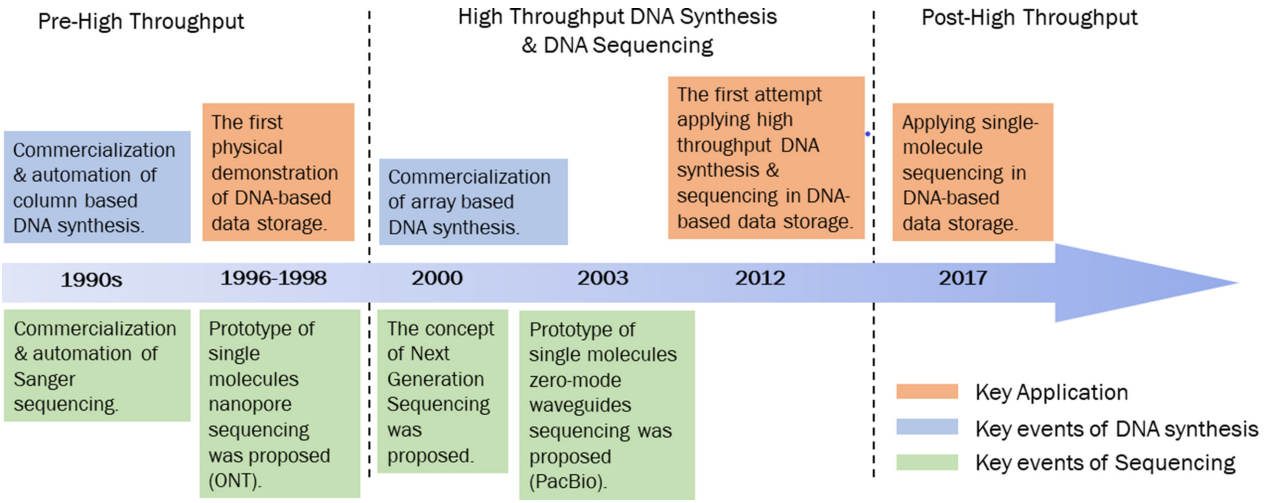

Figure 4: Key events in DNA synthesis and DNA sequencing, and their key applications in DNA-based data storage. PacBio: Pacific Biosciences.

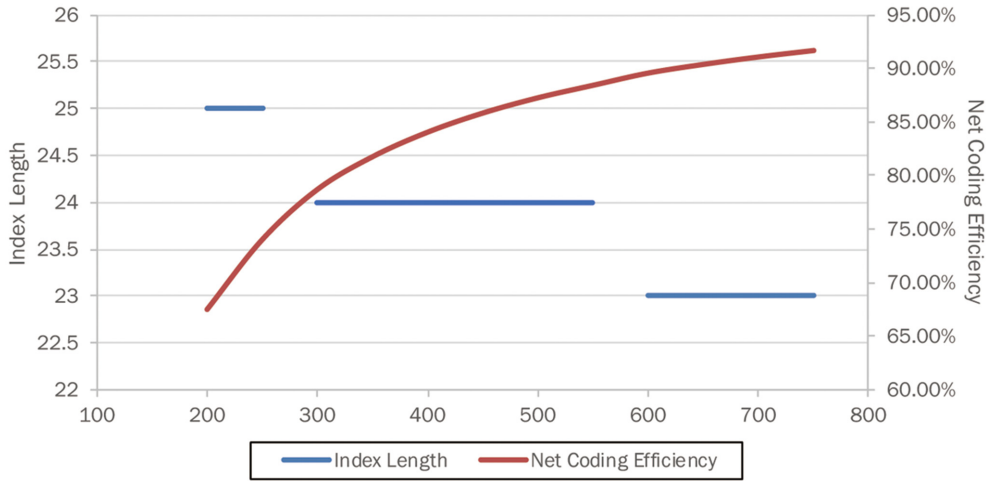

Figure 5: Interrelationship between DNA oligo length, optimal index length, and net coding efficiency in a model of 1-GB digital file transcoding.

Table 2: Summary of frequently used sequencing platforms in DNA-based data storage (data retrieved from [42])

Platform	Error rate (%)	Runtime	Instrument cost (US$)	Cost per Gb (US$)	Reference
Illumina MiSeq	0.10	4–56 hours*	99,000	110–1,000*	[12, 15, 18, 25]
Illumina HiSeq 2000	0.26†	3–10 days*	654,000	41	[8, 11]
Illumina HiSeq 2500	0.10	7 hours–6 days†,*	690,000	30–230*	[17]
Illumina NextSeq	0.20†	11–29 h*	250,000	33–43*	[13]
Oxford Nanopore MinION	8.0†	≤48 h	1,000	70†	[13, 28]

†Latest data retrieved from the industrial report (may be different from previous literature); *varied by read length and reagent kit version.

In 2018, Oxford Nanopore also launched a high-throughput sequencing platform, PromethION, stating that it has the potential to yield up to 20 Tb of data in 48 hours [43, 44]. The first metagenomics data published using the PromethION demonstrated that it is already possible to obtain 150 Gb of data from 2 flowcells in a 64-hour run [45]. Further developments and improvements are in progress. Because the performance of this technology is getting closer to that of its NGS counterparts, it may play a more prominent role in the future study of DNA-based data storage.

Perspectives on DNA-Based Data Storage

Taken together, DNA-based data storage techniques provide us with the great possibility to manipulate DNA as a carbon-based archive with excellent storage density and stability. Imperfect as it is, it may become the ultimate solution to the current data storage market for long-term archiving. We are also excited to see that multidisciplinary research companies have already joined this revolution to make DNA-based archiving commercially viable.

In terms of coding schemes, although the current theoretical limit of bit-base transcoding is 2 bits/base, newly discovered unnatural nucleic acids could expand the choice of bases for transcoding and thus increase the theoretical limit. X and Y are 2 classical unnatural nucleic acids that have demonstrated the capability to be integrated into normal cells, and in pairing, replication, and amplification [46]. Moreover, recent synthetic biology research reported 4 new synthetic nucleic acids: 6-Amino-5-nitropyridin-2-one (Z), 5-Aza-7-deazaguanine (P), Isocytosine (S), and Isoguanine (B) [47]. These new nucleic acid candidates could help to increase the coding efficiency for DNA digital storage in the not-too-far future.

Enterprises with a strong DNA synthesis background are most commonly seen, given that DNA-based data storage can significantly benefit from the breakthroughs achieved in DNA synthesis. It could be foreseen that with continuously improving enzymatic DNA synthesis techniques, DNA oligo synthesis could break the limit of 200-mers in the near future, providing us with a longer primary storage unit. This will undoubtedly improve net coding efficiency with the same lengths of PCR primers and shorter index sequences. In 1 model for the DNA-based storage of a 1-GB file under theoretical limitation, 1 DNA base represented 2 binary bits. For each DNA oligo, the length of forward and reverse primers was set at 20. In this case, we can deduce the equation representing the relationship between index length i and DNA oligo length l:

$$\log_2(l - 40 - i) + i = 32. \quad (1)$$

Hence, we could obtain the correlation between an optimal index length and DNA oligo length.

As Fig. 5 shows, as DNA oligo length increases, the index length decreases, while net coding efficiency increases. Some start-up companies are now reportedly aiming to develop industrial enzymatic DNA synthesis technology. If they can successfully synthesize oligos >200-mers, the efficiency of DNA-based data storage will markedly improve.

In addition, the scale of DNA synthesis also affects the information capacity of DNA-based data storage per unit mass. With the development of array-based DNA synthesis technology, high-throughput oligo synthesis is currently directed to the microscale level. In DNA-based data storage, the information capacity of a certain mass of DNA sequences also relates to the copy number of each DNA molecule. The correlation between information capacity C and copy number N_m of each oligo can be calculated from:

$$C = n \times (N_m \mu \delta \gamma)^{-1}, \quad (2)$$

where n represents the number of bytes carried by each oligo (normally 10–20 bytes/molecule according to different coding schemes), μ is the number of nucleotides per molecule, δ is 320 Da/nt, and γ is 1.67×10^{-24} g/Da. To date, the copy number of oligos is $\sim 10^7$ molecules in on-chip high-throughput synthesis (without dilution) [19]. According to Equation 2, this will give an information capacity level of $\sim 10^{13}$ bytes/g. If the copy number is decreased to 10^4 molecules per oligo, the information capacity will increase to $\sim 10^{16}$ bytes/g. Additionally, synthesis in microscale will also reduce the cost by several orders of magnitude and save the dilution step.

At present, several DNA synthesis companies are taking the lead in this field, based on their related expertise, and providing services related to DNA-based data storage. Twist Biosciences has reportedly already collaborated with Microsoft in a DNA-based data storage project, providing them with oligo pool services [14] using their high-throughput, array-based DNA synthesis technique. Microsoft, together with the University of Washington, launched the "Memories in DNA" project and will collaborate with the Arch Mission Foundation to construct the first Molecular Collection of the aforementioned Lunar Library. Given that these companies are starting to push this business forward, it will be interesting to see how commercial and social applications develop in the future.

Apart from companies with biology backgrounds, information technology (IT)-based industries are also playing an important role in this revolution. Because the coding schemes used in DNA-based data storage must yet be improved to yield higher coding efficiency and fidelity, efforts from the IT field could be of critical importance. For example, from random access data retrieval to scaling up data storage [13], Microsoft successfully implemented its IT philosophy in DNA-based data storage and is marching steadily towards its goal announced in 2017: a proto-commercial system in 3 years to storing some amount of data on DNA [48]. A recent paper written in collaboration with a scientist from the University of Washington described an automated end-to-end DNA-based data storage device, in which 5 bytes of data were automatically processed by the write, store, and read cycle [49]. Further efforts to speed up the coding and decoding process for daily storage applications are still essential.

We expect more entities and research organizations to join this cohort to eventually make carbon-based archiving a reality, and, furthermore, to attain immediate access storage or biological computation. Nevertheless, it remains a priority to maintain a safe and ethical framework for the development of DNA-based data storage. Because DNA is the basic building block of genetic information for living organisms, situations might arise in which synthesized sequences are introduced into living host organisms, and this could lead to biological incompatibility caused by unknown toxicity or other growth stresses. Hence, it is necessary to evaluate the safety of sequences prior to their synthesis. We long to see the day when the safety, capacity, and reliability of DNA means it will become the next-generation digital information storage medium of choice.

Abbreviations

ATCG: adenine, thymine, cytosine, guanine; bp: base pairs; CRISPR: clustered regularly interspaced short palindromic repeats; Da: dalton; Gb: gigabase pairs; GF: Galois field; IT: information technology; KB: kilobytes; MB: megabytes; Mb: megabase pairs; NGS: next-generation sequencing; nt: nucleotide; oligos: oligonucleotides; RS: Reed-Solomon; XOL: exclusive-or.

Competing interests

Z.P., D.Z.M., X.L.H., S.H.C., L.Y.L., F.G., and Y.S. are employees of BGI Shenzhen. The authors declare that they have no other competing interests.

Funding

This work was supported by the Guangdong Provincial Academician Workstation of BGI Synthetic Genomics (No. 2017B090904014), Guangdong Provincial Key Laboratory of Genome Read and Write (No. 2017B030301011), and Shenzhen

Engineering Laboratory for Innovative Molecular Diagnostics (DRC-SZ[2016]884).

Authors' contributions

Z.P., D.Z.M., and X.L.H. collected materials, reviewed the literature, and co-wrote the paper. S.H.C., L.Y.L., and F.G. supported materials collection and revised the paper. S.J.Z. and Y.S. supervised this review and co-wrote the paper. All authors read and approved the final manuscript.

References

1. Neiman MS. Some fundamental issues of microminiaturization. Radiotekhnika 1964;No.1:3–12.
2. Davis J. Microvenus. Art J 1996;**55**(1):70.
3. Bancroft C, Bowler T, Bloom B, et al. Long-term storage of information in DNA. Science 2001;**293**(5536):1763–5.
4. Bonnet J, Colotte M, Coudy D, et al. Chain and conformation stability of solid-state DNA: implications for room temperature storage. Nucleic Acids Res 2010;**38**(5):1531–46.
5. Pääbo S, Poinar H, Serre D, et al. Genetic analyses from ancient DNA. Annu Rev Genet 2004;**38**:645–79.
6. Kool ET. Hydrogen bonding, base stacking, and steric effects in DNA replication. Annu Rev Biophys Biomol Struct 2001;**30**(1):1–22.
7. Nelson DL, Cox MM, Lehninger AL. Lehninger Principles of Biochemistry. 5th ed. New York; Basingstoke: W.H. Freeman; 2008.
8. Pierce BA. Genetics: A Conceptual Approach. 4th ed. New York, NY: W.H. Freeman; 2012.
9. Church GM, Gao Y, Kosuri S. Next-generation digital information storage in DNA. Science 2012;**337**(6102):1628.
10. De Silva PY, Ganegoda GU. New trends of digital data storage in DNA. Biomed Res Int 2016;**2016**:8072463.
11. Goldman N, Bertone P, Chen S, et al. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. Nature 2013;**494**(7435):77–80.
12. Bornholt J, Lopez R, Carmean DM, et al. A DNA-based archival storage system. SIGPLAN Not 2016;**51**(4):637–49.
13. Organick L, Ang SD, Chen YJ, et al. Random access in large-scale DNA data storage. Nat Biotechnol 2018;**36**(3):242–8.
14. Shendure J, Balasubramanian S, Church GM, et al. DNA sequencing at 40: past, present and future. Nature 2017;**550**(7676):345–53.
15. Reed I, Solomon G. Polynomial codes over certain finite fields. J Soc Ind Appl Math 1960;**8**(2):300–4.
16. Huffman DA. A method for the construction of minimum-redundancy codes. Proc IRE 1952;**40**(9):1098–101.
17. Grass RN, Heckel R, Puddu M, et al. Robust chemical preservation of digital information on DNA in silica with error-correcting codes. Angew Chem Int Ed Engl 2015;**54**(8):2552–5.
18. Blawat M, Gaedke K, Hütter I, et al. Forward error correction for DNA data storage. Proc Comput Sci 2016;**80**:1011–22.
19. Erlich Y, Zielinski D. DNA Fountain enables a robust and efficient storage architecture. Science 2017;**6328**:950.
20. Byers JW, Luby M, Mitzenmacher M, et al. A digital fountain approach to reliable distribution of bulk data. In: Proceedings of the ACM SIGCOMM '98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. Vancouver, BC, Canada: ACM, 1998:56–67.
21. MacKay DJ. Fountain codes. IEEE Proc-Commun 2005;**152**(6):1062–8.
22. Wong PC, Wong KK, Foote H. Organic data memory using the DNA approach. Commun ACM 2003;**46**(1):95–8.
23. Arita M, Ohashi Y. Secret signatures inside genomic DNA. Biotechnol Prog 2004;**20**(5):1605–7.
24. Lee H, Popodi E, Tang HX, et al. Rate and molecular spectrum of spontaneous mutations in the bacterium *Escherichia coli* as determined by whole-genome sequencing. Proc Natl Acad Sci U S A 2012;**109**(41):E2774–E83.
25. Shipman SL, Nivala J, Macklis JD, et al. CRISPR-Cas encoding of a digital movie into the genomes of a population of living bacteria. Nature 2017;**547**(7663):345–9.
26. Kosuri S, Church GM. Large-scale de novo DNA synthesis: technologies and applications. Nat Methods 2014;**11**(5):499–507.
27. Ma S, Tang N, Tian J. DNA synthesis, assembly and applications in synthetic biology. Curr Opin Chem Biol 2012;**16**(3–4):260–7.
28. Yazdi S, Gabrys R, Milenkovic O. Portable and error-free DNA-based data storage. Sci Rep 2017;**7**(1):5011.
29. Song L, Zeng AP. Orthogonal information encoding in living cells with high error-tolerance, safety, and fidelity. ACS Synth Biol 2018;**7**(3):866–74.
30. Lee HH, Kalhor R, Goela N, et al. Enzymatic DNA synthesis for digital information storage. bioRxiv 2018, doi:10.1101/348987.
31. Beaucage SL, Caruthers MH. Deoxynucleoside phosphoramidites—a new class of key intermediates for deoxypolynucleotide synthesis. Tetrahedron Lett 1981;**22**(20):1859–62.
32. Sanger F, Nicklen S, Coulson AR. DNA sequencing with chain-terminating inhibitors. Proc Natl Acad Sci U S A 1977;**74**(12):5463–7.
33. Maxam AM, Gilbert W. A new method for sequencing DNA. Proc Natl Acad Sci U S A 1977;**74**(2):560–4.
34. Hughes TR, Mao M, Jones AR, et al. Expression profiling using microarrays fabricated by an ink-jet oligonucleotide synthesizer. Nat Biotechnol 2001;**19**(4):342–7.
35. Singh-Gasson S, Green RD, Yue Y, et al. Maskless fabrication of light-directed oligonucleotide microarrays using a digital micromirror array. Nat Biotechnol 1999;**17**(10):974–8.
36. Gao X, LeProust E, Zhang H, et al. A flexible light-directed DNA chip synthesis gated by deprotection using solution photogenerated acids. Nucleic Acids Res 2001;**29**(22):4744–50.
37. Tian J, Gong H, Sheng N, et al. Accurate multiplex gene synthesis from programmable DNA microchips. Nature 2004;**432**(7020):1050–4.
38. Brenner S, Johnson M, Bridgham J, et al. Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. Nat Biotechnol 2000;**18**(6):630–4.
39. Levene MJ, Korlach J, Turner SW, et al. Zero-mode waveguides for single-molecule analysis at high concentrations. Science 2003;**299**(5607):682–6.
40. Church G, Deamer DW, Branton D, et al. Characterization of individual polymer molecules based on monomer-interface interactions. Google Patents. US Patent 5,795,782(18 August 1998).
41. Karow J. Oxford Nanopore previews upcoming products, outlines Nanopore-based DNA data storage tech. GenomeWeb. 2019. https://www.genomeweb.com/sequencing/oxford-nanopore-previews-upcoming-products-outlines-nanopore-based-dna-data-storage-tech#.XOt_1qZS_EY. Accessed 29 May 2019.
42. Goodwin S, McPherson JD, McCombie WR. Coming of age: ten

years of next-generation sequencing technologies. Nat Rev Genet 2016;**17**(6):333–51.
43. De Coster W, De Roeck A, De Pooter T, et al. Structural variants identified by Oxford Nanopore PromethION sequencing of the human genome. bioRxiv 2018, doi:10.1101/434118.
44. PromethION. https://nanoporetech.com/products/promethion. Accessed 12 Jun 2019.
45. Nicholls SM, Quick JC, Tang S, et al. Ultra-deep, long-read Nanopore sequencing of mock microbial community standards. Gigascience 2019;**8**(5), doi:10.1093/gigascience/giz043.
46. Malyshev DA, Dhami K, Lavergne T, et al. A semi-synthetic organism with an expanded genetic alphabet. Nature 2014;**509**(7500):385–8.
47. Hoshika S, Leal NA, Kim MJ, et al. Hachimoji DNA and RNA: a genetic system with eight building blocks. Science 2019;**363**(6429):884–7.
48. Regalado A. Microsoft has a plan to add DNA data storage to its cloud. MIT Technol Rev 2017. https://www.technologyreview.com/s/607880/microsoft-has-a-plan-to-add-dna-data-storage-to-its-cloud/. Accessed 29 May 2019.
49. Takahashi CN, Nguyen BH, Strauss K, et al. Demonstration of end-to-end automation of DNA data storage. Sci Rep. 2019, doi:10.1038/s41598-019-41228-8.

ADVANCED SCIENCE Open Access
www.advancedscience.com

Mobile and Self-Sustained Data Storage in an Extremophile Genomic DNA

Fajia Sun, Yiming Dong, Ming Ni, Zhi Ping, Yuhui Sun, Qi Ouyang, and Long Qian**

DNA has been pursued as a novel biomaterial for digital data storage. While large-scale data storage and random access have been achieved in DNA oligonucleotide pools, repeated data accessing requires constant data replenishment, and these implementations are confined in professional facilities. Here, a mobile data storage system in the genome of the extremophile *Halomonas bluephagenesis*, which enables dual-mode storage, dynamic data maintenance, rapid readout, and robust recovery. The system relies on two key components: A versatile genetic toolbox for the integration of 10–100 kb scale synthetic DNA into *H. bluephagenesis* genome and an efficient error correction coding scheme targeting noisy nanopore sequencing reads. The storage and repeated retrieval of 5 KB data under non-laboratory conditions are demonstrated. The work highlights the potential of DNA data storage in domestic and field scenarios, and expands its application domain from archival data to frequently accessed data.

1. Introduction

The astronomically growing global data sphere has posed an imminent challenge to data storage technologies.[1,2] Modern data storage systems employ a stratified structure in which data is categorized by the access frequency, and stored in different media or computational layers accordingly.[3,4] In the pursuit of novel storage materials, DNA has shown significant potential for the storage of "cold" data that are infrequently accessed. Gigabyte-scale data storage in a DNA oligo pool and addressed retrieval of minute fractions of data have been demonstrated,[5-15] and massively parallel synthesis and sequencing technologies are substantiating the vision of DNA-based data centers for gigantic cold data archiving (Figure 1a). In contrast, "warm" and "hot" data are medium-scale data in circulation. These data are frequently distributed and accessed on-premises, at home and en route (**Figure 1**b). Although DNA oligo pools can be extremely portable and the pocket-size MinION nanopore sequencer enables rapid data readout,[16,17] data sustainability remains problematic for the frequent access demand of warm data. In oligo pools, data is statically stored in that once sampled, the DNA is not automatically replenished. Consequently, a finite number of retrievals may lead to data exhaustion for small files in a large archive.[11,14] The engagement of the polymerase chain reaction (PCR) provides a solution but at the cost of accumulating systematic data errors and biases.[18] In addition, the requirement for a professional PCR setup jeopardizes the mobility of the in vitro oligo storage system.

Live cells with active DNA replication work as mini Xerox machines for "data" stored in their genomes. Therefore, storage of digital data in intracellular amplicons presents a general model of self-sustained DNA data storage. For example, large scale data storage has been demonstrated on an artificial chromosome[19] (38 KB) and on plasmids in a bacterial population[20] (445 KB). However, genetic instability and copy number fluctuations of these exo-genomic amplicons ultimately impact data integrity.[21,22] In comparison, the genome provides a relatively stable storage environment. Currently, studies integrating DNA containing digital data into bacterial genomes relied on the CRISPR technology and thereby were limited to a few bits of information.[23–25] Moreover, previous studies were mostly done in the model organisms *Escherichia coli* and *Saccharomyces cerevisiae* in sterile and growth-controlled laboratory environments. Recently, Qian et al. used spores of *Bacillus subtilis* carrying DNA barcodes for real-world object tracking as in IoT applications,[26] suggesting that harnessing environmentally robust microbes for digital data storage may greatly expand its applicational realms.

In this work, we propose a mobile and self-sustained storage system based on long artificial DNA in the genome of *Halomonas bluephagenesis* (Figure 1c). As a non-model

F. Sun, Y. Dong, Q. Ouyang, L. Qian
Center for Quantitative Biology
Peking University
5 Yiheyuan Road Haidian District, Beijing 100871, P. R. China
E-mail: projectyasuo@pku.edu.cn; qi@pku.edu.cn; long.qian@pku.edu.cn
M. Ni, Z. Ping, Y. Sun
Academician Workstation of BGI Synthetic Genomics
BGI-Shenzhen
Huada Comprehensive Park
Yantian District, Shenzhen 518083, P. R. China
Q. Ouyang
The State Key Laboratory for Artificial Microstructures and Mesoscopic Physics
Peking University
5 Yiheyuan Road Haidian District, Beijing 100871, P. R. China
The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/advs.202206201

DOI: 10.1002/advs.202206201

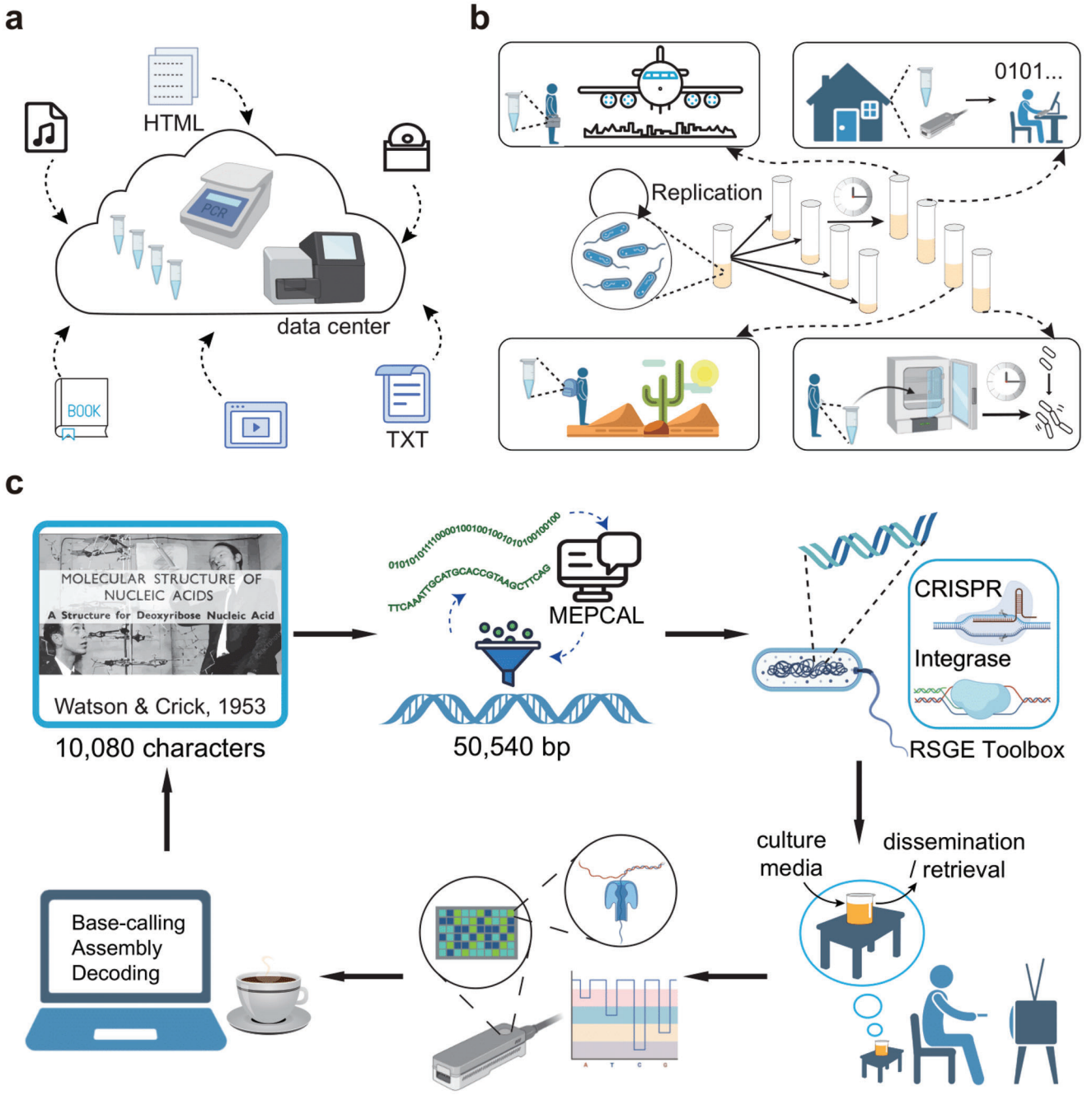

Figure 1. Digital information storage with bacterial genomic DNA and information retrieval using nanopore sequencing. a) Schematic of centralized DNA data storage, as realized in oligo pools. b) Schematic of genomic data storage system, which is convenient for data transfer and retrieval, and sustainable for data distribution and regeneration. c) Work flow of genomic data storage system. Digital information was first compressed and encoded by MEPCAL to generate information DNA. Next, the RSGE toolbox was used to integrate long DNA fragments into bacterial genomes. Information DNA automatically replicated as cells proliferated under indoor environment. For information retrieval, a portable MinION sequencer was used for real-time sequencing, and a laptop was used for data processing. After base-calling and assembly, erroneous information DNA was decoded by MEPCAL to perfectly restore the original information.

organism, this halophilic bacterium is regarded as a potential chassis for portable and open fermentation due to its unique features of anti-contamination and easiness of cultivation and cell collection.[27–32] This feature is exploited here for mobility, that is, prolonged data storage and frequent data retrieval achieved in nonlaboratory environments with minimal professional handling. Digital data is genomically stored in tens of kilobases (kb) of continuous synthetic DNA for both stability and dosage control and for a maximized storage density. The MinION sequencer is employed for real-time data readout.[33,34]

The technical challenges of developing the system are twofold. The first challenge is large scale genomic integration. Although several strategies are available for the genomic integration of large synthetic DNA in model organisms *E. coli* and yeast, such as the λ-Red system, nucleases, integrases, or their combinations,[35-38] in most other industrial bacteria, these techniques are still developing.[39-41] In particular, no such system has been reported in *H. bluephagenesis*. The second challenge is readability. Despite a few attempts, restoring information from noisy nanopore reads with ∼10% errors rich in insertions and deletions (indels) has remained a daunting task.[12,42-44] Current indel correcting codes either rely on consensus voting, which requires substantial sequencing coverage, or are expensive in computation time. These features significantly limiting the storage capacity and the retrieval speed.[10,45] Another work circumvents this issue by coding with short sequences of distinctive nanopore output signals, but this approach strongly constrains the sequence space available for coding.[16]

To solve these problems, we designed a genetic toolbox and a coding scheme. The genetic toolbox enables genomic integration of 10–100 kb-scale DNA fragments in *H. bluephagenesis*. The coding scheme efficiently targets indels without compromising storage density. The two were combined to establish a prototypic genomic storage system that aims to extend the territory of DNA storage from cold data to warm data, and make it available in non-professional facilities.

2. Results and Discussion

2.1. System Construction and Stability Evaluation

We selected the seminal article revealing the double helical DNA structure by Watson and Crick for genomic storage.[46] The article, a 5.56 KB text file, was encoded to two DNA sequences of lengths 29 and 51 kb (termed information DNA) by different strategies. The information DNA was synthesized by commercial company and delivered in the form of plasmids containing DNA fragments of 3–8 kb in length. These DNA fragments were first assembled into continuous fragments with a length of 12–18 kb, and then iteratively integrated into the genomes of *E. coli* and *H. bluephagenesis* by a recombinase-based site-specific genome engineering (RSGE) toolbox we developed (Note S1, Supporting Information). Among the integrases that have been shown to work in *E. coli*, very few had been proven functional in a different species.[47–51] Through a comprehensive screen, we obtained 16 integrases that successfully worked in *H. bluephagenesis* (Table S1, Supporting Information), with each recognizing a specific pair of attB and attP sites with extraordinary sequence specificity.

To integrate information DNA into bacterial genomes, we employed a landing pad strategy. A "receiver cassette" containing 16 attB sites was inserted in *E. coli* and *H. bluephagenesis* genomes by CRISPR-mediated homologous recombination via a dual plasmid system (Figure S1, Supporting Information). Next, we assembled the synthesized DNA fragments with plasmid vectors via Golden Gate assembly,[52] resulting in plasmids carrying an integrase gene and its corresponding "sender" attP site, as well as information DNA of lengths 12–18 kb. The plasmids were then transformed into host cells. Upon transformation, the expressed integrase promoted integration of information DNA at its corresponding attB site within the receiver cassette (**Figure 2**a). Two and three rounds of transformation were conducted for information DNA coded by two strategies to generate *H. bluephagenesis* and *E. coli* carrying up to 51 and 29 kb of continuous synthetic DNA, respectively. No evident decrease in doubling time was observed for the information-bearing strains, albeit the growth of these cells was slightly affected depending on the length of integrated information DNA (Movie S, Figures S2a and S5, Supporting Information).

In engineered host genomes, long tracts of synthetic DNA are often prone to fragmental loss and translocation, as well as spontaneous mutations. Therefore, during a passaging experiment lasting 2000 generations, Sanger sequencing was performed at seven time points for overlapped fragments covering the entire information DNA. Among the above-mentioned genetic errors, only one base substitution was observed in the cells at 100th generation (Table S8, Supporting Information). Antibiotics were found to be dispensable within a limited culture time (∼100 generations). Fragmental DNA loss was observed during an extended culture period without antibiotics (Figure 2b). *H. bluephagenesis* thrives in high salt environments where most bacteria undergo growth arrest. The advantage of this resistance to biological contamination in data storage was illustrated when we cultured the information-bearing *H. bluephagenesis* strain under 6% w/v NaCl condition in either sterile or open environments for 18 h. In both experiments, the intact information DNA was successfully retrieved by PCR at the endpoint. In contrast, information DNA stored in *E. coli* was not retrievable after 18 h of open culture under the regular 1% saline condition (Figure 2c).

2.2. Frequent and Rapid Data Retrieval from a Benchtop Storage System

In contrast to data storage in oligo pools, cell growth offers automatic data regeneration from loss incurred by long-term storage and frequent retrievals. We first tested how the genomic storage system withstood host dormancy, which represented a long-term storage scenario. The information-bearing *H. bluephagenesis* was allowed to grow until saturation in a high-salt LB medium. The saturated culture medium was then mixed with equivoluminal 50% glycerin and placed in a −20 °C refrigerator. After frozen for 14 months, the mixture was thawed at room temperature and 1:100 diluted into fresh medium. The bacteria regained its maximal density after overnight incubation in a shaker. The saturated culture medium (OD ≈ 1.0) contained ∼10^9 CFU mL^{-1} bacteria as extrapolated from serial dilution and colony generation experiments (Figure 2d). We tested the sequencing speed of exceedingly small samples loaded on the MinION sequencer. Unamplified genomes extracted from one milliliter of saturated culture medium resulted in a sequencing speed of ∼340 Mbp h^{-1}, which would allow for information recovery in a 10-min sequencing time (see Section 4). When the sample volume was reduced by 1000-fold (1 µL), the sequencing speed decreased by only ∼tenfold (Figure 2d). Notably, the full process of information retrieval, including genome extraction, library construction, sequencing, assembly, and decoding, was completed within one to a few hours (Note S2, Supporting Information).

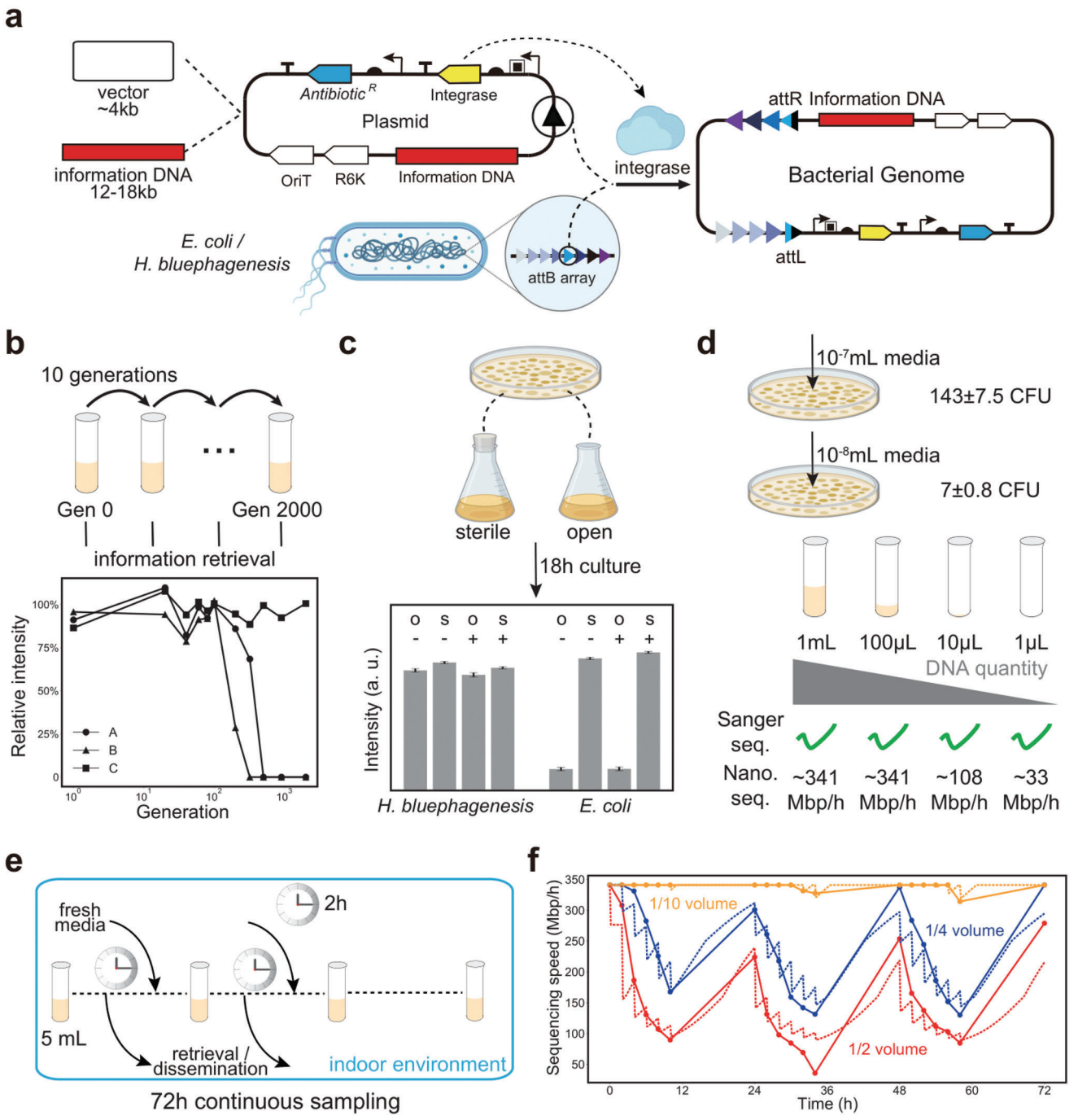

Figure 2. Construction and characterization of the genomic data storage system. a) Schematic of integration of information DNA into bacterial genomes using the RSGE toolbox. b) Bacterial passaging experiments. Chart below: the intensities of PCR bands of strains cultured without antibiotics relative to that of strains cultured with antibiotics. A, B, and C refer to three non-overlapping ~1 kb regions covering the junctions between the information DNA and the genome. The intensities of the strain cultured in antibiotic-containing media kept steady and close to those of the genomic control (Figure S3, Supporting Information). c) Open culture of information-bearing strains. Chart below: the intensities of PCR bands of *E. coli* and *H. bluephagenesis* after 18 h culture in different conditions (Figure S4, Supporting Information). O, open; S, sterile. Data were shown as mean ± SEM of $n = 20$ regions (~2 kb each) collectively covering the entire information DNA. d) Colony counting and readability. Different volumes of saturated culture media were used for colony counting and information retrieval with Sanger sequencing and nanopore sequencing. Numbers for nanopore sequencing indicate sequencing speeds. e) Continuous sampling scheme of a desktop DNA information storage system. f) MinION sequencing speeds of extracted genomic DNA at each sampling point. Colored numbers indicated the sampling volume. At each point,100 μL of culture media was used for the speed test. Dots and solid lines were experimental results; dotted lines were model predictions (Note S4, Supporting Information).

Next, we designed a continuous sampling scheme to challenge the system with recurrent information retrievals (Figure 2e). The revived *H. bluephagenesis* culture medium was placed on a bench, uncovered and unshaken (i.e., a household setting). During a 72-h period, sampling was done every 2 h for a total of six times every day. At each sampling timepoint, certain volumes of the master culture medium (2.5 mL/1.25 mL/0.5 mL from a 5 mL pool) were taken for sequencing, and the same volumes of the fresh medium were replenished. The bacteria, along with the information DNA in their genomes, slowly replicated and regained quantity during sampling intervals. The recovery behavior was fitted to a bacteria growth model that serves to predict the availability of information given more frequent retrievals or larger sampling volumes (Note S3, Supporting Information). In our experiments, all sampling schemes supported recurrent data readout at nearly the maximal sequencing speed. To probe a scenario where cell availability would affect sequencing speeds, we reduced the culture medium sampling volume to 100 µL, and then extracted DNA for sequencing. The largest variations in sequencing speed (~15-fold reduction) were between the first and the sixth samples each day for the 2.5 mL group. This was consistent with the cell density estimates by the model (Figure 2f).

2.3. Coding Strategy

Compared to data storage in oligo pools, the cellular storage system poses unique coding challenges. First, sequences with potential biological activities (e.g., recognition sites of enzymes and recombination elements) may induce host interactions. Second, nanopore sequencing generates indels and context-specific error profiles at high rates. An ideal coding scheme should be able to handle significant coding constraints while be efficient at correcting indel errors. To this end, we designed an error correction code named Mixed Error Processing Coding for Arbitrary Length (MEPCAL).

MEPCAL employed a layered structure combining nested Reed-Solomon (RS) code,[53] RaptorQ code,[54] and an anchoring approach. In this study, its processing unit was base-256 numbers (information symbols), which was directly convertible to 4-bp DNA symbols. The MEPCAL encoder proceeded in four steps (**Figure 3**). First, information symbols were generated from the original information and appended by RS repair symbols at 8.6% redundancy rate before they were partitioned into encoding groups. Second, each encoding group was further organized into 16 encoding sets (packets), from which RaptorQ code, a variant of the fountain code, was applied to generate a surplus of repair packets. Regardless of their origin, the same number of packets would suffice for the restoration of the encoding group (Note S4, Supporting Information), providing sufficient coding flexibility at minimal redundancy cost. In Step 3, all packets were transcoded to DNA sequences, and a designated "leading base" was inserted between DNA symbols. These leading bases served as anchors for indel detection in the decoding process. Packets were then subjected to five filters to reject sequences with high error rates, low signal-to-noise ratios in the nanopore ionic current streams, extremely error-prone sub-sequences, inappropriate GC ratios and potential biological activities (potential open reading frames (ORFs), RSGE recombination sites, repetitive sequences, Golden gate exicision sites, etc.). These filters were customizable to the specific sequencing platform, the host organism and the genetic cloning method, to collectively enhance stability in DNA synthesis and transformation, and reduce error rates ab initio in sequencing. Next, from all packets, 16 qualified sequences were selected, and RS code was applied again at 50% redundancy rate to provide the second level of error protection. Finally, all symbols were organized linearly according to the order of encoding sets and then encoding groups and indexed by 5-bp and 10-bp interval indices, respectively. Through the MEPCAL encoder, the compressed 5.56 KB text data[55] was coded into a DNA sequence of 50 540 bp (0.886 bit/base), of which 27.87% was for RS redundancy, 18.04% was for leading bases, and 9.77% was for indexing.

The decoding process of MEPCAL included 1) segmentation: locating the interval indices between encoding groups and encoding sets by a maximal likelihood algorithm, 2) indel calling: identifying DNA symbols in each encoding set to infer length variations due to indels, 3) RS decoding of substitutions and erasures, and 4) RaptorQ decoding. During indel calling, each encoding set was aligned to a regularly spaced pattern of leading bases through a parsimony algorithm. Expansions and contractions of leading base spacings were identified as indels and treated as erasures of the affected symbols (Figure 3). As a result, data corruption due to a miscalled indel was locally confined, and was extremely unlikely to propagate outside the encoding set.

In parallel, we encoded the compressed file with a binary Bose–Chaudhuri–Hocquenghem (BCH) code[56,57] at 1.552 bit/base (28 672 bp DNA). This coding scheme neither performs ab initio error reduction nor corrects indels (Section 4). As BCH code can be regarded as a binary version of RS code, it targets substitution errors at the same rate as RS code does.

2.4. Nanopore Sequencing and Information Retrieval

To obtain sufficient data for the analyses of nanopore error patterns and MEPCAL's error correction capacity, we performed ligation-sequencing of the revived *H. bluephagenesis* strain for a prolonged period on MinION. 388 849 reads were obtained with an average read length 6.5 kb and average coverage of 603×. Per-base coverage was uniform across the reference sequence (information DNA and the genomic backbone) (**Figure 4**a), and all reads mapped to continuous tracts on the reference (Figure S11c, Supporting Information). Both results confirmed the genetic stability of integrated information DNA.

A close examination of raw reads from the BCH-encoded strain revealed the error patterns of nanopore sequencing, including biased base substitution rates (Figure 4b) and indel length distributions (Figure 4c). The total error frequency was 14% at the nucleotide level, with indel rates around 7.8%. For comparison, next generation sequencing (Hi-seq, BGI-Shenzhen) of the same strain yielded a total error frequency of 0.58% dominated by substitutions (Table S9, Supporting Information). Non-random and context-dependent error profiles in nanopore reads were observed (Figure 4d), which were characteristics of the flowcell and the base-calling algorithm employed (Note S5, Supporting Information). Specifically, right-skewed distributions of per-site error frequencies indicated error hotspots as verified by the 5-mer context analysis (Figure 4e,f). These error

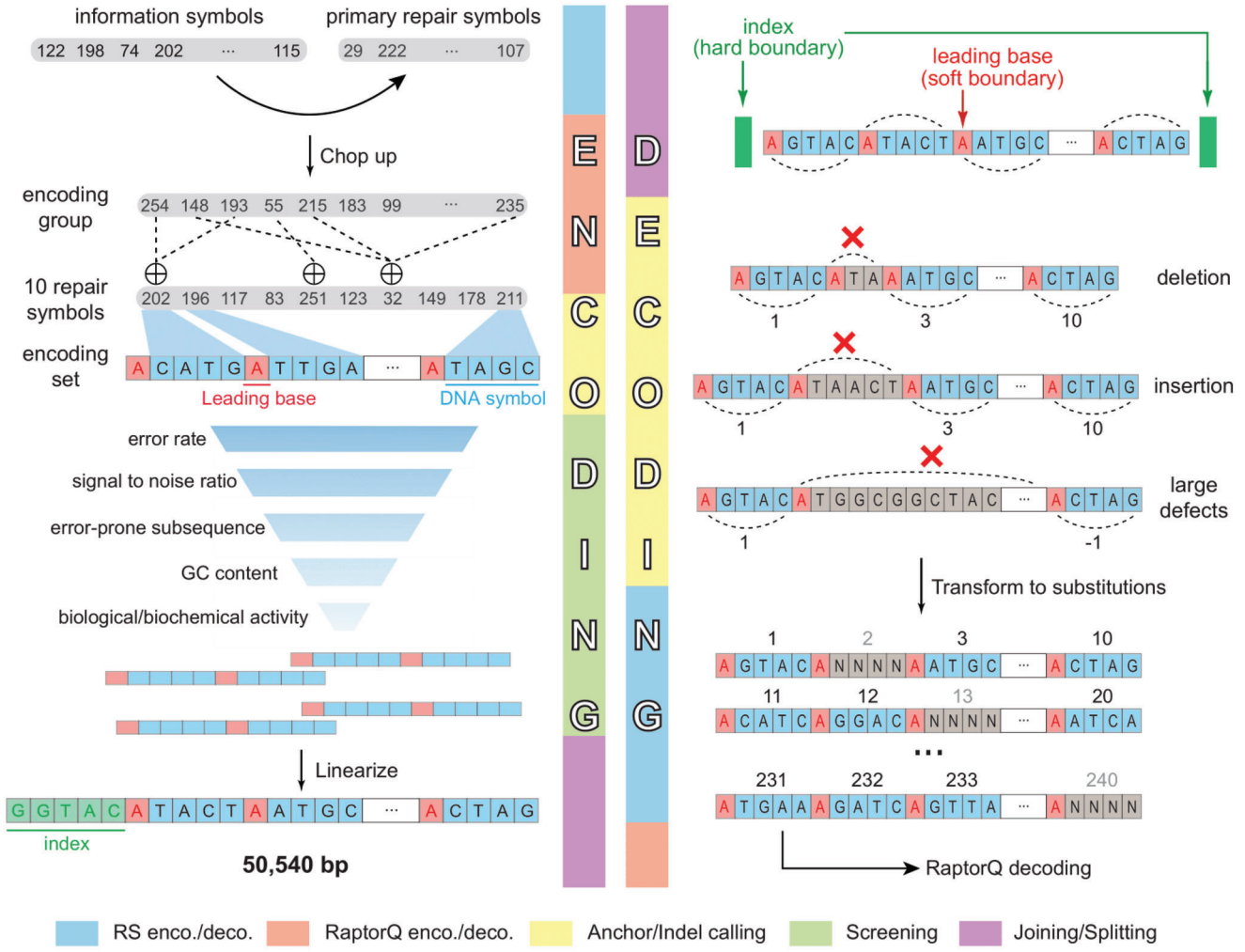

Figure 3. Algorithm pipeline of MEPCAL. Left: Encoding. First, the compressed information along with the primary repair symbols generated by RS code were divided into encoding groups. Next, secondary repair symbols were generated using RaptorQ code for each encoding group. These symbols were then transformed into DNA sequences (encoding set), in which a leading base was added before every DNA symbol. These DNA sequences were screened using five sequential filters. Screened sequences were concatenated and appended with tertiary RS repair symbols (not shown), and interval indices were added between sequences. Finally, sequences of encoding groups were concatenated, with interval indices added between sequences, generating the final information DNA. Right: decoding. Information DNA was first divided in accordance to the preset interval indices. Indel calling was then performed for each encoding set to restore potential DNA symbols, with each symbol endowed a serial number based on its position in the sequence. Vacant serial numbers were considered as erasures, and conflicting serial numbers were judged by maximal likelihood. DNA symbols and their corresponding serial numbers (unique mapping) were then sent to the RS decoder and finally the RaptorQ decoder.

patterns served as references for Filters 1 and 3 of the MEPCAL pipeline. Consequently, MEPCAL-encoded information DNA exhibited significantly fewer sequencing errors than the backbone genomic sequence ($p < 0.01$, Kolmogorov–Smirnov test), with a 20–30% reduction in indel frequencies (**Table 1**).

We used Flye[58] for the de novo assembly of the complete information DNA sequence. Consensus sequences with different assembly coverages were constructed from randomly sampled reads of the encoding region. The minimum assembly coverage capable of generating non-gapped consensus was 9.13× with a post-assembly error rate of ~0.3%. The error rate steeply decreased and rested at ~0.035% from 30× coverage and on (**Figure 5**a). Each of the constructed consensuses was successfully decoded by the MEPCAL decoder to restore the original information (red dots in Figure 5b). In comparison, the consensus sequence of BCH-encoded strain exhibited ~0.5% total errors at 375 × coverage. Although the BCH redundancy was sufficient to correct substitutions, decoding failed due to the existence of ~0.35% indel errors.

Given a sequencing speed of 340 Mbp h^{-1} and a host genome size of 4.2 Mbp, reads sufficient for lossless information retrieval (~10 × coverage) can be obtained in <10 min. To probe MEPCAL's error correcting capability beyond the assembly limit, we added pseudorandom errors (with equivalent fractions of substitutions, insertions, and deletions) to the information DNA sequence to create pseudo-sequences with 0% to 3% errors. Remarkably, information was perfectly restored from any sequence with an error rate <1.8%, while up to 2.8% errors were tolerable

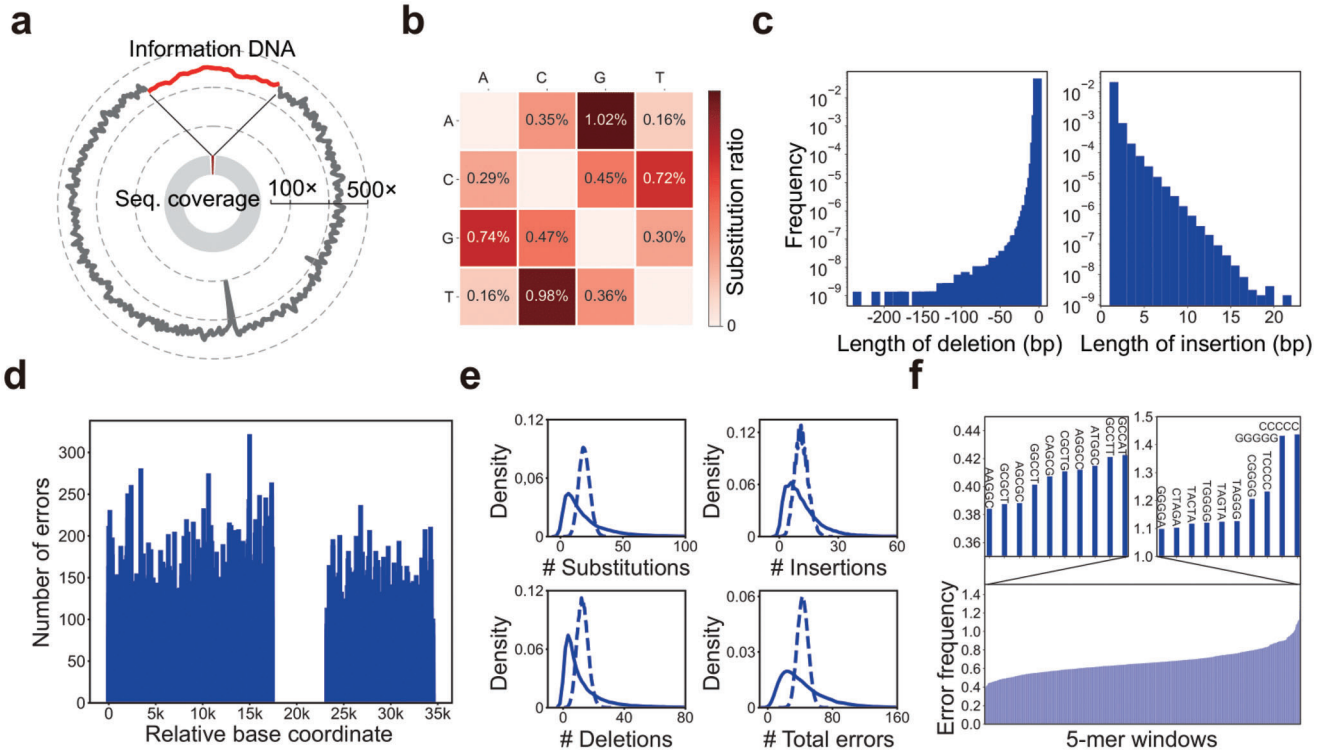

Figure 4. Analysis of error patterns in nanopore sequencing. a) Per-site sequencing coverage of the bacterial genome with information DNA integrated. Data in the whole genome were smoothed with 1000-bp windows. Data in the region of information DNA were magnified and labelled in red. b) Matrix of base substitution frequencies. The error rate (i, j) indicated the frequency of the i-th base being replaced by the j-th base. c) Length distribution of insertions and deletions. d) Per nucleotide error frequency along the information DNA. e) Error distribution in d for substitutions, insertions, deletions, and total errors, respectively. Dotted line in each subgraph was the Poisson distribution with the same mean. f) 5-mer context-dependent error frequencies. The above two subgraphs showed the ten 5-mers with the lowest and highest error frequencies, respectively.

Table 1. Error statistic of nanopore sequencing results of bacteria with MEPCAL-encoded information DNA integrated.

Statistical scope	Number of reads	Number of bases	Number of substitutions	Number of insertions	Number of deletions	Number of errors
Whole genome	353 748	2 419 277 016 bp	92 730 369 bp (3.83%)	81 648 812 bp (3.37%)	97 455 858 bp (4.03%)	271 835 039 bp (11.24%)
Encoding region	6908	33 947 519 bp	1 196 299 bp (3.52%)	867 027 bp (2.55%)	1 090 817 bp (3.21%)	3 154 143 bp (9.29%)
Subsample of non-encoding region	3831	23 653 240 bp	928 623 bp (3.93%)	836 552 bp (3.54%)	962 690 bp (4.07%)	2 727 865 bp (11.53%)

(Figure 5b), suggesting the potential of combining MEPCAL with less accurate but faster assembly algorithms.[59]

$$d = \frac{2}{1+I_g} \times d_N \times d_E \times \frac{S}{S+I_s} \times \frac{R}{R+1} \quad (1)$$

2.5. A Trade-Off between Coding Density and Robustness

With the current algorithmic parameters, simulated data indicated a roughly linear decay of the effective coding density in the error rate range of 0.25–3% (Figure 5c). We developed a model to quantify the trade-off between density and robustness, and to determine the optimal combination of parameters in MEPCAL under different error rates (Note S6, Supporting Information). Briefly, the coding density of MEPCAL is bit/base, where I_g and I_s are the lengths of interval indices between encoding groups and encoding sets, respectively; d_N and d_E are fractions of the primary and the tertiary repair symbols, respectively; S is the number of DNA symbols in each encoding set, and R is the length ratio of DNA symbols to leading bases.

For decoding, RS code possesses definite error correction capability, namely, information is recoverable if $Pr(\bar{r} \geq d_E) \geq d_N$, where $\bar{r}$ is the proportional difference between correctly and incorrectly recovered DNA symbols (Note S6, Supporting Information). $\bar{r}$ critically depends on S, R, and the post-assembly

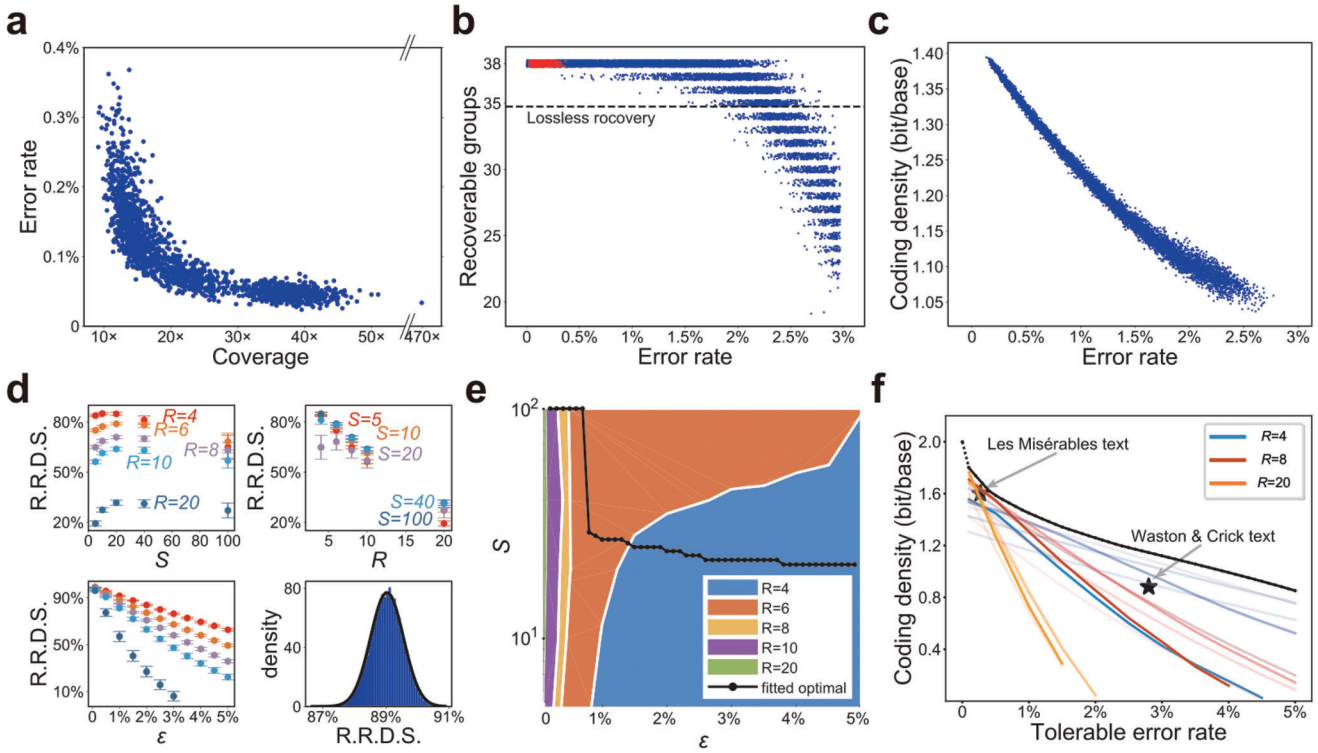

Figure 5. Information retrieval using MEPCAL. a) Scatter plot of the error rate of the consensus sequence versus the assembly coverage. b) The number of recoverable encoding groups under different error rates. Red dots represented the assembled sequences (data in a); blue dots represented simulated sequences with random errors added. Dots were slightly vertically shifted to show density. Dotted line indicated the minimum number of encoding groups required for perfect information retrieval. c) Effective coding density under different error rates, with bases not used for decoding excluded from the calculation. d) Dependency of the ratio of recoverable DNA symbols (R. R. D. S.) on S, R, and ϵ in MEPCAL. Identical colors indicated the same values of S and R. Data points were plotted as mean ± standard error of 2000 simulations (20000 DNA symbols for each simulation). The subgraph at the bottom right showed the distribution of R. R. D. S. when $R = 4$, $S = 10$, and $\epsilon = 0.02$. The black line was the best-fit Gaussian distribution. e) Parameter selection of S and R that maximizes the coding density under different error rates. f) Maximal coding density under different error rates (black line). Colored lines represented the density-robustness trade-off for combinations of different S and R. For each R, the value of S was 5, 10, 20, 40, and 100. Deeper colors indicated larger S.

error rate ϵ, namely, how well the leading base alignment algorithm locates indels and how well the mis-alignments are constrained within the encoding set. Simulating this part of decoding, we found that $\bar{r}$ followed approximately a normal distribution $\mathcal{N}(f(S,R,\epsilon),\ g(S,R,\epsilon))$, where the mean f and the variance g^2 were fitted from simulated data (Figure 5d). This generated an envelope curve demarcating the upper bounds of coding density and the corresponding parameter selection regimes (Figure 5e,f).

To examine the validity of the model and the performance of MEPCAL on larger data sizes, we encoded the classic novel "Les Misérables" (4.2 MB text) by Victor Hugo with the optimized parameter combination (S, R, d_N, d_E) deduced from the model to cope with an error rate of 0.25–0.35%, yielding a DNA sequence of 21.7 Mbp with a coding density of 1.6 bit/base (Note S7, Supporting Information). Decoding of erroneous pseudo-sequences resolved up to 0.28% of mixed errors (Figure S15, Supporting Information), suggesting a coverage of 10–20× in nanopore sequencing was sufficient for error-free information retrieval. The deviation from the envelope curve in Figure 5f came from a small chance of misidentification of interval sequences and leading bases.

3. Discussion and Conclusion

While implementing data storage in DNA pools through massively parallel DNA synthesis and next generation sequencing is essential to meet the challenge of large scale data storage, data writing and reading are carried out on physically large platforms with professional protocols to achieve high throughputs. This condition makes the frequent access of relatively small amounts of data from a civilian environment unhandy and uneconomical. Therefore, it is indispensable to develop lightweight and mobile storage systems with handy data operations (replication, distribution, retrieval, etc.) to supplement the mainstream DNA pool strategy. In this work, we reported a genomic DNA storage system supporting frequent information retrieval and distribution while being portable and self-sustainable. The system enabled 1) durable data storage in two host growth modes (dormant and thriving), 2) automatic data regeneration on a benchtop with resistance to contaminations, 3) rapid data retrieval within one to a few hours, and 4) error-free decoding of nanopore reads at <10 × sequencing coverages. These features point to applications of a read-only storage model of medium storage lifespan, medium data volume, and frequent data accessing.

Two storage modes were demonstrated: an active mode for data regeneration and a dormant mode for long-term storage. In the active mode, our 100-day passaging experiment suggested the genomic environment of bacteria provided information stability and error-proof information replication, considering that bacterial genomic mutation rates are 10^{-9}–10^{-10}/(bp·generation),[60] while commercial PCR enzymes produce errors at 10^{-5}–10^{-6}/(bp·cycle).[61] In our experiment, the dormant mode, that is, when the bacteria were refrigerated, was tested for 14 months, and the data-carrying strain successfully rebooted to assume the active mode in one day. Both modes and their transition were applicable in household settings. In particular, maintaining the active mode required only a normal container and culture medium supplements. The high-salt growth condition essentially ensures the thrive of the data-carrying host in spite of environmental microbial contaminants. The robust sequencing performance in Figure 2d,f likely spares the data retrieval protocols from sub-milliliter liquid handlings. Storage time was also enhanced by antibiotic selection, as is commonly used in current cellular storage studies.[19,20,22] In antibiotic-free media, our genomic storage system preserved the entire information for 100 generations (4–5 days). In comparison, previous studies using the yeast artificial chromosome for data storage showed fragmental loss of DNA (in antibiotic media) and loss of entire YACs (in antibiotic-free media).[20,22] This suggested that with or without antibiotics, bacterial genome provides a more stable environment for data storage than exo-genomic elements.

It is meaningful to compare the storage capacity of the genomic storage system with an archival system of DNA oligo pools. For considerations of dilute solution, the indexing demand and a lower limit of 10 copies/100 µL for retrievability,[14] 1 mL oligo solution can hold a maximum of ~3 PB data. For 1 mL bacterial culture of the genomic storage system with the population storage model,[20] assuming a 50 kb information DNA length, a minimum of 10 copies for readability and a 10^9 CFU mL^{-1} bacterial concentration, the maximal storage capacity is ~1 TB (Note S8, Supporting Information). The magnitude difference matches that between a commercial hard disk drive and a USB flash drive. It is possible to increase the storage capacity of the genomic storage system, as the RSGE toolbox supports sequential DNA integrations up to hundreds of kb. However, the amount of artificial DNA tolerable by a host microbe and the biological impacts of these genomic integrations await elucidation.[62,63] It is also of interest to investigate biases in the long-term dynamics of a non-clonal population. Population drift typically happens on the scale of 10^9 generations for actively proliferating cells. Comparing this with the baseline bias derived from DNA synthesis and PCR in oligo pool storage[18] will elucidate its impact on data integrity in a mixed-population storage scenario.

Speed and robustness are vital for data retrieval. Compared to storage with 100–200 nt oligos, genomic storage with kb-scale encoding lengths enables real-time nanopore sequencing and enhances computational efficiency for de novo assembly (Note S9, Supporting Information). Congruously, several studies attempted to extend the length of oligos in a DNA pool.[12,17] Yet, in vitro maintenance and amplification of long DNA molecules face difficulties.[64] To deal with enormous errors in nanopore sequencing, we developed a flexible and robust coding scheme named MEPCAL. Notably, RaptorQ was employed not for error correction but for effective error reduction ab initio given an arbitrary set of user-defined constraints. For indel correction, Press et al. applied convolutional code and a greedy search algorithm to correct up to 3.59% errors in next generation sequencing readouts.[45] However, this approach exhibited high computational complexity, and long codewords only exacerbate the problem. Chen et al. used superposition and a modified forward-backward algorithm to cope with indels in nanopore reads at an error rate of 0.43%.[20] However, the efficiency of the algorithm might have been contingent on short sporadic indel errors. Our results suggested that hierarchical grouping and base anchoring was sufficient to correct significant indel errors at low sequencing coverages without harming the coding density and decoding speed. Consistently, MEPCAL achieved a better coding density-robustness trade-off among existing coding strategies, and exhibited scant performance loss when scaling up to larger data sizes (**Table 2**). While MEPCAL was initially developed to handle massive errors in long DNA fragments and nanopore sequencing, it can be easily adapted for DNA pools as well (Note S11, Supporting Information).

The leading bases used in this work were a string of adenines (A). They can be replaced by pseudorandom watermarks or any patterned sequence to balance the overall GC content or to encode extra information into these redundancy bits. In the latter case, a bespoke decoding algorithm may achieve optimized efficiency. From the biological aspect, the effective operation of the RSGE toolbox in *H. bluephagenesis* suggested its versatility given the taxonomic distance between *Halomonas* and *Escherichia*. It is conceivable that diverse microbial physiology can be leveraged for specific storage needs, such as dormant states,[26] rapid replication[65] or non-sterile storage (this work). Further studies are required to assess the utility of the RSGE toolbox in different species, as well as the practicality of various bacteria for information storage.

4. Experimental Section

Bacterial Strains and Culturing Conditions: Strains used in this study included *H. bluephagenesis* TD01 (Genus: *Extremophile Halomonas spp.*, obtained from ref. [66]) and *E. coli* S17-1 (Bluepha) and TOP10 (TRANS). With appropriate antibiotics added, sugar-free LB medium was used for *E. coli* strains (1% peptone, 0.5% yeast extract, 1% sodium chloride, pH = 7.0), while salt-rich LB medium was used for *H. bluephagenesis* TD01 (1% peptone, 0.5% yeast extract, 6% sodium chloride, pH = 7.0). All the above percentages are mass/volume ratio.

Synthesis of Information DNA: Information DNA coded by BCH code was 28 672 bp in length, which was divided into 10 fragments. Information DNA coded by MEPCAL was 50 540 bp in length, which was divided into 6 fragments. All these fragments were synthesized by GENEray (http://www.generay.com.cn/). To obtain these long DNA fragments, short ologos of 80–100 nt were first synthesized and then assembled by PCR, resulting in longer DNA fragments with hundreds of nt in length. Next, these DNA fragments were further assembled by Gibson assembly and then loaded on a carrier plasmid, which was transformed into *E. coli* for proliferation. The correctness of the DNA sequence was verified by Sanger sequencing of the plasmid carrying synthesized DNA. The synthesized products were delivered in the form of plasmids and puncture bacteria (*E. coli*).

Genomic Integration of attB Array: The sequence of 16 attB sites were artificially synthesized and then connected back and forth, forming an attB array with a length of 897 bp. Two plasmids were used to integrate attB array into bacterial genome, one of which carried *cas9* gene (pCas for *E.*

Table 2. Comparison to prior work.

	Amount of information stored/KB	Coding density/(bit/base)	Storage carrier	Sequencing method	Minimum coverage for information retrieval	Resistance to indel
Church et al.Ref. [5]	674	0.633	DNA pool	Illumina	~ 3000×	No
Goldman et al.Ref. [6]	635	0.290	DNA pool	Illumina	51×	No
Grass et al.Ref. [7]	83	0.862	Silica sphere	Illumina	372×	No
Blawat et al.Ref. [8]	22 528	0.892	DNA pool	Illumina	160×	No
Bornholt et al.Ref. [9]	151	0.226	DNA pool	Illumina	40×	No
Erlich et al.Ref. [11]	2097	1.569	DNA pool	Illumina	10.5×	No
Organick et al.Ref. [12]	205 005	0.822	DNA pool	Illumina	5×	No
	32/1.3			Nanopore	36×/80×	No
Press et al.Ref. [44]	128	0.595	DNA pool	Illumina	~3×	Yes
Shipman et al.Ref. [24]	3.8	0.725	CRISPR array in *E. coli* genome	Illumina	>150–1580×	No
Chen et al.Ref. [19]	37.8	1.245	Artificial chromosome in *S. cerevisiae*	Nanopore	16.8×	Yes
This work	5.5	1.552	Long DNA fragment in bacterial genome	Sanger/HiSeq	MfA[a)]	No
		0.886		Nanopore	<9.13×	Yes
	4239(simulated)	1.600	/			Yes

[a)] MfA: minimal coverage for assembly; see Note S10, Supporting Information, for details of "Coding density."

coli TOP10, and pSEVA321 for *H. bluephagenesis* TD01), while another carried attB array and the gene of guide RNA (pTarget for *E. coli* TOP10, and pSEVA241 for *H. bluephagenesis* TD01). For *E. coli* TOP10, the first plasmid was transformed into competent cells, and the second plasmid was then introduced into screened strain via electro-transformation. For *H. bluephagenesis* TD01, these two plasmids were first assembled via Gibson assembly and transformed into *E. coli* S17-1 cells separately. Next, the first plasmid was introduced into *H. bluephagenesis* TD01 cells through conjugation, and the second plasmid was then introduced into screened strain via another conjugation.

Genomic Integration of Information DNA: Information DNA were integrated into the genome of *E. coli* TOP10 following 2 steps. First, information DNA were assembled with the plasmid vector to form an intermediate plasmid containing the resistance gene, the integrase gene, the attP site, the replication origin site along with information DNA. Next, the plasmid was introduced into *E. coli* TOP10 via chemical transformation. Information DNA were then integrated into the genome of *E. coli* TOP10. This integration strategy was iteratively performed for twice to insert BCH-coded information DNA (28.7 kb) into the genome of *E. coli* TOP10. The integration into the genome of *H. bluephagenesis* TD01 following 3 steps. First, information DNA were assembled with the plasmid vector to form an intermediate plasmid containing the resistance gene, the integrase gene, the attP site, the replication origin site along with information DNA. The plasmid was then transformed into *E. coli* S17-1 competent cells, which still existed as plasmid and replicated as cell proliferated. Finally, *E. coli* S17-1 containing the plasmid were conjugated with *H. bluephagenesis* TD01, and the plasmid was transferred from *E. coli* S17-1 to *H. bluephagenesis* TD01 and integrated into the genome of the latter. This integration strategy was iteratively performed twice/3 times to insert BCH-coded/MEPCAL-coded information DNA (28.7 kb/50.5 kb) into the genome of *H. bluephagenesis* TD01.

Determination of Growth Curve: The growth curve of bacteria with information DNA integrated was measured in shaker. Strains to be tested were cultured in liquid media and grown for 20 h under 200 rpm, 37 °C to reach the maximum cell density, which served as seed culture. For growth curve of bacteria cultured in shaker, 5 µL of seed culture was added to 5 mL of sterile fresh liquid medium and then cultured in shaker (200 rpm, 37 °C). For sterile setting, the absorbance was determined at 0, 2, 4, 5, 6, 7, 8, 9, 10,11, 12, 16, and 20 h after the dilution (Figure S2a, Supporting Information). For desktop culture system, the absorbance was determined at 0, 2, 4, 6, 8, 10, 14, 16, 18, 20, 24, 26, 28, 30, 32, and 34 h after the dilution (Figure S2b, Supporting Information). For each strain under each condition, 3 biological repeats were measured.

Microscopic Imaging of Bacterial Growth: Bacteria with information DNA integrated were cultured in in liquid medium for 12 h under 200 rpm, 37 °C, and then 1:100 diluted and further cultured in in liquid medium for 4 h under 200 rpm, 37 °C. This bacterial media was then centrifuged under 5000 g for 1 min. Next, enriched bacterial cells were added into a microfluidic chip contains 4 channels, which was then centrifuged under 3500 g for 10 min. The centrifuged chip was then connected to 4 syringes, each of which was put into a pump. A thermostat was applied to monitor the temperature (37 °C). An inverted microscope (60 × oil immersion objective) was used for imaging, which lasted 12 h with a 15 min interval for each time of shooting.

Passaging of Bacteria: Bacteria with information DNA integrated were cultured in liquid medium overnight under 200 rpm, 37 °C. This bacterial media was regarded as the 0th generation. The passaging was carried out every 12 h, in which 5 µL of bacterial media was added to 5 mL of sterile fresh liquid medium and grown under 200 rpm, 37 °C. Each time of passaging was considered to be 10 generations ($2^{10} \approx 1001$).

Open Culture: Bacteria with information DNA integrated were cultured in both sterile and non-sterile conditions to confirm their resistance to biological contamination. Deep hole plate (24 holes) was used to culture the strain, with half of the holes sealed with a sterile film (Figure S4a, Supporting Information). Strains were cultured in liquid medium and grown for 20 h under 200 rpm, 37 °C. This bacterial media was then 1:1000 diluted, and then added to sealed (sterile) and open wells. TOP10 and TD01 with information DNA integrated were cultured in media with and without antibiotics, respectively. The plate was then placed in shaker (room temperature, 500 rpm) for 16 h.

Co-culture of Information-Bearing Strains: Two strains with different length of information DNA integrated (strain A: 11 312 bp; strain B: 28 712 bp) were separately cultured in liquid media until saturated. Next, 200 µL of saturated culture media of strain A, 200 µL of saturated culture media of strain B and 6 mL of fresh media were mixed and placed in shaker (200 rpm, 37 °C) for 8 h. This bacterial media was regarded as the 5th generation ($6400/200 = 32 = 2^5$). The subsequent passaging was carried out every 12 h, in which 5 µL of saturated media was added to 5 mL of

sterile fresh liquid medium and grown under 200 rpm, 37 °C. Each time of passaging was considered to be 10 generations ($2^{10} \approx 1001$). 3 biological replications were conducted along with control (strain A only and strain B only). The relative proportions of the two strains were quantified by fluorescent quantitative PCR at 5th, 10th, 20th, 40th, and 100th generations.

Continuous Sampling Experiments: Bacteria with information DNA integrated were cultured in liquid medium overnight under 200 rpm, 37 °C, and then diluted to reach OD600 = 1.0. Next, 5 mL of this medium was added into a 15 mL tube, and kept uncovered and unshaken under room temperature. Sampling was carried out at 12:00, 14:00, 16:00, 18:00, 20:00, and 22:00 every day, until 12:00 on the fourth day. At each sampling point, the absorbance of the medium was measure, and then 2.5 mL/1.25 mL/0.5 mL of medium was removed, with equal volume of fresh medium added. No extra fresh media was added to compensate the evaporation.

Bacterial Cryopreservation and Revival: Bacteria with information DNA integrated were cultured in liquid medium overnight under 200 rpm, 37 °C, to reach the maximum cell density. 500 µL of saturated media was mixed with 500 µL 50% glycerin in a 1.5 mL Eppendorf tube. The mixture was then placed into a household refrigerator (−20 °C). The cryopreserved bacteria were kept in the refrigerator for more than 1 year. For revival, the frozen mixture was melted at room temperature, and 50 µL of melted mixture was added to 5 mL fresh media. Next, the media was cultured overnight under 200 rpm, 37 °C for recovery.

Source Coding: Huffman code was used in the source coding step to compress the text that was encoded. The source entropy of the article encoded was $H = 4.391511$ bit/position, and the average code length of Huffman code used was $L = 4.417229$ bit/position, which was close to the upper limit of compression specified by Shannon's first theorem.

Bose–Chaudhuri–Hocquenghem Encoding: The original information (binary numbers) was first divided into 224 groups. In each group, BCH (255,207) was used to generate 255-bit BCH codeword from 207-bit information symbols based on generator polynomial. Finally, 1 parity bit was added to the end of each group of BCH codeword, so that the Hamming weight (number of 1s) of codeword in each group was an even number. In this way, each group of codeword corresponded to exactly 128 bases. 224 groups of the above BCH code were used to store the information, resulting in $128 \times 224 = 28\,672$ bp of DNA sequence.

Mixed Error Processing Coding for Arbitrary Length Encoding: The original information (5564 bytes) was first padded into 5600 base-256 numbers (symbols). RS code based on GF(2^{12}) was used to generate 480 primary repair symbols. These symbols were re-transformed into base-256 numbers (6080 symbols) and then grouped into 38 encoding groups. Next, each encoding group (160 symbols) was used as information symbols of RaptorQ encoder to produce 399 times of secondary repair symbols (63840 symbols), and every 10 symbols were then grouped to form an encoding set. In each encoding set, each of the 10 encoding symbol was transformed into 4-bp DNA and preceded by an additional "A," which was called "leading base," as a separation identifier, resulting in a 50-bp DNA. These sequences were then screened in 5 consecutive steps: Error rate screening, signal-to-noise ratio screening, error-prone sequence screening, GC-ratio screening, and bio-related screening. In each encoding group, 16 sequences passing through screening were retained, which were re-transformed into base-256 numbers and served as information symbols for RS code based on GF(2^8) to generate 80 tertiary repair symbols. Every 10 symbols were then transformed into a 50-bp DNA sequence in the same manner as described above and preceded by a 5 bp interval index. As a result, each group contained 24 55-bp sequences with a total length of 1320 bp, which was comprised of 240 DNA symbols. Finally, a 10-bp index was added before each group, generating information DNA of $(1320 + 10) \times 38 = 50\,540$ bp in length.

Sequence Screening: Sequence screening was comprised of 5 steps. The first 3 screening were based on the analysis of error patterns in nanopore sequencing. Filter 1: Error rate screening. The single molecule sequencing machine used can accommodate 5 bases (i.e., every 5 adjacent bases generate an electrical signal when passing through the nanopore; the windows are overlapped, thus ideally each base corresponds to an electrical signal), thus $4^5 = 1024$ kinds of 5-mer windows are available. Approximately 35 000 000 bp of nanopore sequencing results were applied to align with the original sequence (reference sequence) to obtain the error frequency of each 5-mer. Each 50-bp sequence contained 46 5-mer windows, whose average error rate can be calculated based on the error frequency analysis. In order to determine the screening threshold, the average error rate of 10 000 random sequences with a length of 50 bp was calculated, and the threshold that caused 60% of the sequences to be discarded was defined. Filter 2: Signal-to-noise ratio screening. Certain different sequences produce similar electrical signals, causing base-calling errors. In order to reduce these errors, the electrical signals generated by adjacent *k*-mer windows were required to make evident difference. Here, a *k*-mer model[67] was used to calculate the "signal-to-noise ratio" of each 50-bp sequence. In this model, the electrical signal generated by each window follows a Gaussian distribution with a determined mean and variance. Signal-to-noise ratio was defined as the average negative logarithm of the overlapping area of the Gaussian distributions of electrical signals generated by two adjacent windows, that is,

$$R = \frac{1}{n-5} \sum (-\ln A) \tag{2}$$

where R was the signal-to-noise ratio and A was the above-mentioned overlapping area

$$A = \int_{-\infty}^{\infty} \min \left[f\left(x; \mu_1, \sigma_1\right), f\left(x; \mu_2, \sigma_2\right) \right] \tag{3}$$

μ_1,σ_1, and μ_2,σ_2 referred to the mean and variance of the Gaussian distribution of two adjacent electrical signals, respectively. In order to determine the screening threshold, the average signal-to-noise ratio of 10 000 random sequences with a length of 50 bp were calculated, and the threshold caused 60% of the sequences to be discarded was defined. Filter 3: Error-prone sequence screening. Each 5-mer window exhibited a specific error frequency. Among the 1024 windows, 64 with the highest error frequency were excluded. In other words, the 50-bp sequence containing any of these 64 windows was discarded. Filter 4: GC ratio screening. 10 "A" were added to each 50-bp sequence, causing the GC ratio of the sequence to deviate by 50%. Therefore, the GC ratio was screened to obtain sequences with a GC ratio of 40–60%, while other sequences were discarded. Filter 5: Bio-related screening, all DNA sequences with potential biological activities were excluded in this step. Specifically, the biologically related constraints were set as follows: 1) Sequences with possible ORFs or their reverse complement sequences are discarded, for these sequences might initiate gene expression in cells. For this constraint we employed the ORF finder (NCBI).[68] 2) Sequences with recombination sites of RSGE recombinases and their reverse complement sequences were discarded, for these sequences can easily trigger site-specific recombination, thereby damage the integrity of information DNA. Specifically, all attB sites and attP sites in the RSGE toolbox are supposed to be absent in the information DNA. In addition, the recombination sites of four excisionases were excluded to enable the extended functionality of RSGE toolbox (e.g., excision of vector sequences after integration of information DNA, as illustrated in Figure S7, Supporting Information). All the sequences of these sites are listed in Table S1, Supporting Information. 3) Sequences with low complexity regions were discarded, for these regions usually exhibit evident repetitiveness, which were prone to recombination and fragmental DNA loss. In addition, repetitive sequences were hard to synthesize and sequence. Here, RepeatMasker[69] was used as a reference to identify low complexity sequences. 4) Sequences with recognition sites of Golden Gate-related excision enzyme were discarded. This operation was to ensure successful sequence assembly by the Golden Gate method.[52] Specifically, GGTCTC for BsaI, GAAGAC for BbsI, and CGTCTC for BsmBI, as well as the reverse complements of the above sequences were excluded in the sequence of information DNA. Besides, all interval sequences inserted in the information DNA conformed to these rules.

Mixed Error Processing Coding for Arbitrary Length Decoding: Information DNA was first divided into 38 encoding groups based on "maximum likelihood grouping" determined by interval indices. Using the same

strategy, each encoding group was further divided to obtain 24 encoding sets, which served as the basic unit of recognition of DNA symbols. The first step of recognition was finding the position of all leading bases ("A") in the sequence. Only DNA symbols with normal length (4 bp) were recognized and then given a serial number of 1–10. When DNA symbols with the same number appeared in an encoding set, the corresponding leading bases of these DNA symbols were compared, and the correct one was determined by the principle of "maximum likelihood." The serial numbers of recognized DNA symbols were then transformed into absolute serial numbers based on the order of their encoding set in encoding group (from 1 to 240). The ordered DNA symbols were transformed into base-256 numbers and then used for RS decoding. RaptorQ decoding was finally done, which was dispensable when the first 35 groups of information were successfully decoded by RS code.

Nanopore Sequencing: The genome of bacteria was extracted and then processed for library construction before sequencing. The library was prepared using Ligation Sequencing Kit (Oxford Nanopore Technologies, catalog no. SQK-LSK109) or Rapid Barcoding Kit (Oxford Nanopore Technologies, catalog no. SQK-RBK004), which was then sequenced on MinION single-molecule sequencing device (Oxford Nanopore Technologies, ONT) by loading certain amount of DNA sample on a flowcell (R9.4.1/R10.3). The sequencing device was operated using the bundled software, MinKNOW to monitor running status and perform base-calling of the raw data.

Quantitative Model and Optimal Parameter Selection: MEPCAL contains 5 major optional parameters: The number of extra encoding group(s) G, the number of DNA symbols in an encoding group E, the number of DNA symbols in an encoding set S, the length ratio of DNA symbols to leading bases R, the number of tertiary repair symbol in an encoding group M.

$$\mathrm{MEPCAL} = \mathrm{MEPCAL}(G,\ E,\ S,\ R,\ M) \tag{4}$$

Other numerical attributes of MEPCAL can be inferred by the above 5 parameters. The coding density of MEPCAL was

$$d = \frac{2}{1+I_g} \times d_{\mathrm{N}} \times d_{\mathrm{E}} \times d_{\mathrm{S}} \times d_{\mathrm{R}}\ \text{bit/base} \tag{5}$$

where

$$d_{\mathrm{N}} = \frac{N}{N+G} \tag{6}$$

$$d_{\mathrm{E}} = \frac{E}{E+M} \tag{7}$$

$$d_{\mathrm{R}} = \frac{R}{R+1} \tag{8}$$

$$d_{\mathrm{S}} = \frac{S}{S+I_s} \tag{9}$$

The sufficient condition for decoding was

$$\sum_{i=1}^{N+G} D(i) \geq N \tag{10}$$

where $D(i)$ referred to the decodable condition of RS code for each encoding group

$$\mathrm{D}(i) = \begin{cases} 1, & \textit{if decodable} \\ 0, & \textit{if undecodable} \end{cases} \tag{11}$$

The judgment of D depended on the acquisition of valid DNA symbols. The sufficient condition for $D(i) = 1$ was

$$2\theta + (1 - r - \theta) \leq \frac{M}{E+M} \tag{12}$$

or equivalently

$$r - \theta \geq \frac{E}{E+M} \tag{13}$$

where r was the ratio of valid DNA symbols in each encoding group after decoding. Note that a DNA symbol was valid only when both its sequence and order (serial number) are correct. θ was the ratio of invalid DNA symbols in each encoding group after decoding. Invalid DNA symbols referred to DNA symbols with incorrect sequence or order. 1-r-θ was the ratio of unrecoverable DNA symbols in each encoding group.

$$r + \theta \in [0,\ 1] \tag{14}$$

In silico simulations under different combinations of parameters in MEPCAL as well as different error rates were performed to determine r and θ. Assuming that the separation of encoding groups and encoding sets was accurate, r and θ were exclusively related to the underlying ability of DNA symbol recognition in MEPCAL. Both r and θ were dependent variables of R, S, and error rate ϵ. For each combination of the above 3 independent variables, r and θ approximately followed Gaussian distribution. Let

$$\bar{r} = r - \theta \tag{15}$$

Here $\bar{r}$ was related to R, S, and ϵ, which also followed Gaussian distribution when fixing the above 3 variables (Figure 5d)

$$\bar{r} \sim N\left(\mu, \sigma^2\right)|_{R=R_0, S=S_0, \varepsilon=\varepsilon_0} \tag{16}$$

where μ and σ were the mean and variance of the Gaussian distribution. Different combination of these 3 variables were selected and then performed 1000 simulations (2400 DNA symbols for each simulation) for each combination. The mean and variance of these samples were the best estimates (unbiased and effective) of μ and σ.

Multivariate polynomial fitting was performed to obtain a quantitative relationship between μ, σ, and R, S, ϵ, where the highest power of polynomials was 3.

$$\mu = \sum_{i=0,\ j=0,k=0}^{i+j+k\leq 3} aR^i S^j \varepsilon^k \tag{17}$$

and

$$\sigma = \sum_{i=0,\ j=0,k=0}^{i+j+k\leq 3} bR^i S^j \varepsilon^k \tag{18}$$

Let

$$p = p\left(\bar{r} \geq \frac{E}{E+M}\right) \tag{19}$$

where p referred to probability. The sufficient condition for complete information retrieval was

$$(N+G) \times p \geq N \tag{20}$$

Therefore, the sufficient condition for complete information retrieval in MEPCAL was

$$p\left(\bar{r} \geq \frac{E}{E+M}\right) \geq \frac{N}{N+G} \tag{21}$$

Note that

$$d_N = \frac{N}{N+G},\ d_E = \frac{E}{E+M} \tag{22}$$

Therefore, the sufficient condition for decoding was

$$p\left(\bar{r} \geq d_E\right) \geq d_N \tag{23}$$

In which $\bar{r}$ follows a Gaussian distribution with known mean and variance, and d_E and d_N were constants less than 1. As a result, the trade-off between coding density and error correction capability in MEPCAL can be summarized as

$$\begin{cases} d = \frac{2}{1+l_g} \times d_N \times d_E \times \frac{S}{S+l_s} \times \frac{R}{R+1}\ bit/base \\ p\left(N\left(f\left(d_R, d_S, \varepsilon\right), g(d_R, d_S, \varepsilon)^2\right) \geq d_E\right) \geq d_N \end{cases} \tag{24}$$

Since

$$\bar{r} \sim N\left(\mu, \sigma^2\right) \tag{25}$$

We can get

$$\frac{\bar{r}-\mu}{\sigma} \sim N(0, 1) \tag{26}$$

Therefore,

$$\begin{aligned} p\left(\bar{r} \geq d_E\right) &= p\left(\frac{\bar{r}-\mu}{\sigma} \geq \frac{d_E-\mu}{\sigma}\right) = 1 - p\left(\frac{\bar{r}-\mu}{\sigma} \leq \frac{d_E-\mu}{\sigma}\right) \\ &= 1 - \Phi_0\left(\frac{d_E-\mu}{\sigma}\right) \end{aligned} \tag{27}$$

where Φ_0 was the distribution function of the standard Gaussian distribution $N(0,1)$. When the error rate changed, the values of d_E and d_N changed accordingly, resulting in different maximum coding densities. Here, the authors fixed

$$\frac{d_E-\mu}{\sigma} = -2.33 \tag{28}$$

So that

$$\max(d_N) = -\Phi_0(2.33) = 0.99 \tag{29}$$

For a specific error rate, σ was fixed. When d_R and d_S increased, μ decreased, which in turn reduced d_E. Since the coding density was proportional to the product of these 3 parameters, the optimal trade-off between them determined a highest encoding density.

Statistical Analysis: No statistical methods were used to predetermine sample size. The experiments were not randomized. Data are not preprocessed unless explicitly declared. The summary of statistics used in this study was shown in Table S12, Supporting Information. Data analysis and visualization were mostly performed in Anaconda3 (conda 4.10.3, https://www.anaconda.com/) using python3 language. Some standard python extension packages were used for these works, which included numpy, pandas, scipy, math, matplotlib, seaborn, re, random, pickle, Levenshtein, and pylab.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

F.S. and Y.D. contributed equally to this work. This work was funded by the National Key R&D Program of China (2021YFF1200100 to D.Y., 2021YFF1200500 and 2020YFA0906900 to L.Q.) and the National Natural Science Foundation of China (31901063 and 12090054 to L.Q.).

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

F.S., Y.D., L.Q., and Q.O. developed the initial concept. F.S. and Y.D. designed and performed the biological experiments. M.N. and Z.P. carried out the next-generation sequencing. Y.S. processed the sequencing results of NGS. F.S. designed and developed the coding methods. F.S. analyzed the sequencing results under the supervision of L.Q. F.S., L.Q., and Y.D. wrote the manuscript. F.S., L.Q., and Q.O. edited the manuscript.

Data Availability Statement

The data that support the findings of this study are openly available in Online materials at https://bdainformatics.org/dataRepository, reference number 03.

Keywords

biomaterials, DNA data storage, error correction codes, genome engineering, nanopore sequencing

Received: October 24, 2022
Revised: January 11, 2023
Published online: February 3, 2023

[1] A. Extance, *Nature* **2016**, *537*, 22.
[2] S. Vitak, Technology alliance boosts efforts to store data in DNA, https://www.nature.com/articles/d41586-021-00534-w (accessed: January 2023).
[3] J. J. Levandoski, P. Larson, R. Stoica, presented at *IEEE 29th ICDE*, Brisbane, QLD, Australia, April **2013**.
[4] Y. Hsu, R. Irie, S. Murata, M. Matsuoka, presented at *IEEE 11th CLOUD*, San Francisco, CA, USA, July **2018**.
[5] G. M. Church, Y. Gao, S. Kosuri, *Science* **2012**, *337*, 1628.
[6] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, E. Birney, *Nature* **2013**, *494*, 77.
[7] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, W. J. Stark, *Angew. Chem., Int. Ed. Engl.* **2015**, *54*, 2552.
[8] M. Blawat, K. Gaedke, I. Huetter, X.-M. Chen, B. Turczyk, S. Inverso, B. W. Pruitt, G. M. Church, *Procedia Comput. Sci.* **2016**, *80*, 1011.
[9] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, K. Strauss, *IEEE Micro* **2017**, *37*, 98.
[10] S. Yazdi, R. Gabrys, O. Milenkovic, *Sci. Rep.* **2017**, *7*, 5011.
[11] Y. Erlich, D. Zielinski, *Science* **2017**, *355*, 950.
[12] L. Organick, S. D. Ang, Y. J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Kamath, P. Gopalan, B. Nguyen, C. N. Takahashi, S. Newman, H. Y. Parker, C. Rashtchian, K. Stewart, G. Gupta, R. Carlson, J. Mulligan, D. Carmean, G. Seelig, L. Ceze, K. Strauss, *Nat. Biotechnol.* **2018**, *36*, 242.

[13] L. Anavy, I. Vaknin, O. Atar, R. Amit, Z. Yakhini, *Nat. Biotechnol.* **2019**, *37*, 1229.
[14] L. Organick, Y. J. Chen, S. D. Ang, R. Lopez, X. Liu, K. Strauss, L. Ceze, *Nat. Commun.* **2020**, *11*, 616.
[15] J. Koch, S. Gantenbein, K. Masania, W. J. Stark, Y. Erlich, R. N. Grass, *Nat. Biotechnol.* **2020**, *38*, 39.
[16] K. Doroschak, K. Zhang, M. Queen, A. Mandyam, K. Strauss, L. Ceze, J. Nivala, *Nat. Commun.* **2020**, *11*, 5454.
[17] R. Lopez, Y. J. Chen, S. D. Ang, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Seelig, K. Strauss, L. Ceze, *Nat. Commun.* **2019**, *10*, 2933.
[18] Y. J. Chen, C. N. Takahashi, L. Organick, C. Bee, S. D. Ang, P. Weiss, B. Peck, G. Seelig, L. Ceze, K. Strauss, *Nat. Commun.* **2020**, *11*, 3264.
[19] W. Chen, M. Han, J. Zhou, Q. Ge, P. Wang, X. Zhang, S. Zhu, L. Song, Y. Yuan, *Natl. Sci. Rev.* **2021**, *8*, nwab028.
[20] M. Hao, H. Qiao, Y. Gao, Z. Wang, X. Qiao, X. Chen, H. Qi, *Commun. Biol.* **2020**, *3*, 416.
[21] T. Wein, Y. Wang, N. F. Hülter, K. Hammerschmidt, T. Dagan, *Curr. Biol.* **2020**, *30*, 3841.
[22] Z. Ping, S. Chen, G. Zhou, X. Huang, S. J. Zhu, H. Zhang, H. H. Lee, Z. Lan, J. Cui, T. Chen, W. Zhang, H. Yang, X. Xu, G. M. Church, Y. Shen, *Nat. Comput. Sci.* **2022**, *2*, 234.
[23] S. L. Shipman, J. Nivala, J. D. Macklis, G. M. Church, *Science* **2016**, *353*, aaf1175.
[24] S. L. Shipman, J. Nivala, J. D. Macklis, G. M. Church, *Nature* **2017**, *547*, 345.
[25] S. S. Yim, R. M. McBee, A. M. Song, Y. Huang, R. U. Sheth, H. H. Wang, *Nat. Chem. Biol.* **2021**, *17*, 246.
[26] J. Qian, Z. X. Lu, C. P. Mancuso, H. Y. Jhuang, R. D. C. Barajas-Ornelas, S. A. Boswell, F. H. Ramírez-Guadiana, V. Jones, A. Sonti, K. Sedlack, L. Artzi, G. Jung, M. Arammash, M. E. Pettit, M. Melfi, L. Lyon, S. V. Owen, M. Baym, A. S. Khalil, P. A. Silver, D. Z. Rudner, M. Springer, *Science* **2020**, *368*, 1135.
[27] X. Z. Fu, D. Tan, G. Aibaidula, Q. Wu, J. C. Chen, G. Q. Chen, *Metab. Eng.* **2014**, *23*, 78.
[28] X. Chen, L. Yu, G. Qiao, G. Q. Chen, *J. Ind. Microbiol. Biotechnol.* **2018**, *45*, 545.
[29] I. P. Parwata, D. Wahyuningrum, S. Suhandono, R. Hertadi, *IOP Conf. Ser.: Earth Environ. Sci.* **2018**, *209*, 012017.
[30] M. Liu, H. Liu, M. Shi, M. Jiang, L. Li, Y. Zheng, *Microb. Cell Fact.* **2021**, *20*, 76.
[31] X. Jiang, J. Yin, X. Chen, G. Chen, *Methods Enzymol.* **2018**, *608*, 309.
[32] J. Ye, G. Chen, *Essays Biochem.* **2021**, *65*, 393.
[33] S. Lipworth, H. Pickford, N. Sanderson, K. K. Chau, J. Kavanagh, L. Barker, A. Vaughan, J. Swann, M. Andersson, K. Jeffery, M. Morgan, T. E. A. Peto, D. W. Crook, N. Stoesser, A. S. Walker, *Microb. Genomics* **2020**, *6*, mgen000453.
[34] E. Steinig, S. Duchêne, I. Aglua, A. Greenhill, R. Ford, M. Yoannes, J. Jaworski, J. Drekore, B. Urakoko, H. Poka, C. Wurr, E. Ebos, D. Nangen, L. Manning, M. Laman, C. Firth, S. Smith, W. Pomat, S. Y. C. Tong, L. Coin, E. McBryde, P. Horwood, *Mol. Biol. Evol.* **2022**, *39*, msac040.
[35] N. Snoeck, M. D. Mol, D. V. Herpe, A. Goormans, I. Maryns, P. Coussement, G. Peters, J. Beauprez, S. D. Maeseneire, W. Soetaert, *Biotechnol. Bioeng.* **2019**, *116*, 364.
[36] C. Huang, L. Guo, J. Wang, N. Wang, Y. X. Huo, *Appl. Microbiol. Biotechnol.* **2020**, *104*, 7943.
[37] B. Su, D. Song, H. Zhu, *Microb. Cell Fact.* **2020**, *19*, 108.
[38] M. Juhas, J. W. Ajioka, *Microb. Cell Fact.* **2016**, *15*, 172.
[39] T. K. Guha, A. Wai, G. Hausner, *Comput. Struct. Biotechnol. J.* **2017**, *15*, 146.
[40] T. Cerisy, W. Rostain, A. Chhun, M. Boutard, A. C. Tolonen, *mSphere* **2019**, *4*, 6.
[41] K. Nishida, A. Kondo, *Metab. Eng.* **2021**, *63*, 141.
[42] S. Goodwin, J. D. McPherson, W. R. McCombie, *Nat. Rev. Genet.* **2016**, *17*, 333.
[43] A. Doricchi, C. M. Platnich, A. Gimpel, F. Horn, M. Earle, G. Lanzavecchia, A. L. Cortajarena, L. M. Liz-Marzán, N. Liu, R. Heckel, R. N. Grass, R. Krahne, U. F. Keyser, D. Garoli, *ACS Nano* **2022**, *16*, 17552.
[44] Y. Dong, F. Sun, Z. Ping, Q. Ouyang, L. Qian, *Natl. Sci. Rev.* **2020**, *6*, 6.
[45] W. H. Press, J. A. Hawkins, S. K. Jones Jr., J. M. Schaub, I. J. Finkelstein, *Proc. Natl. Acad. Sci. U. S. A.* **2020**, *117*, 18489.
[46] J. D. Watson, F. H. Crick, *Nature* **1953**, *171*, 737.
[47] F. Dafhnis-Calas, Z. Xu, S. Haines, S. K. Malla, M. C. Smith, W. R. Brown, *Nucleic Acids Res.* **2005**, *33*, e189.
[48] L. Yang, A. A. Nielsen, J. Fernandez-Rodriguez, C. J. McClune, M. T. Laub, T. K. Lu, C. A. Voigt, *Nat. Methods* **2014**, *11*, 1261.
[49] J. Fernandez-Rodriguez, L. Yang, T. E. Gorochowski, D. B. Gordon, C. A. Voigt, *ACS Synth. Biol.* **2015**, *4*, 1361.
[50] C. A. Merrick, J. Zhao, S. J. Rosser, *ACS Synth. Biol.* **2018**, *7*, 299.
[51] M. G. Durrant, A. Fanton, J. Tycko, M. Hinks, S. S. Chandrasekaran, N. T. Perry, J. Schaepe, P. P. Du, P. Lotfy, M. C. Bassik, L. Bintu, A. S. Bhatt, P. D. Hsu, (Preprint) bioRxiv, https://doi.org/10.1101/2021.11.05.467528, v2, submitted: November **2021**.
[52] C. Engler, R. Gruetzner, R. Kandzia, S. Marillonnet, *PLoS One* **2009**, *4*, e5553.
[53] I. S. Reed, G. Solomon, *J. Soc. Ind. Appl. Math.* **1960**, *8*, 300.
[54] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, L. Minder, RaptorQ Forward Error Correction Scheme for Object Delivery, https://tools.ietf.org/html/rfc6330/ (accessed: January 2023).
[55] C. E. Shannon, *Bell Syst. Tech. J.* **1948**, *27*, 379.
[56] R. C. Bose, D. K. Ray-Chaudhuri, *Inf. Control* **1960**, *3*, 68.
[57] A. Hocquenghem, *Chiffers* **1959**, *2*, 147.
[58] M. Kolmogorov, J. Yuan, Y. Lin, P. A. Pevzner, *Nat. Biotechnol.* **2019**, *37*, 540.
[59] J. Ruan, H. Li, *Nat. Methods* **2020**, *17*, 155.
[60] M. Lynch, M. S. Ackerman, J. F. Gout, H. Long, W. Sung, W. K. Thomas, P. L. Foster, *Nat. Rev. Genet.* **2016**, *17*, 704.
[61] P. McInerney, P. Adams, M. Z. Hadi, *Mol. Biol. Int.* **2014**, *2014*, 287430.
[62] M. Itaya, K. Tsuge, M. Koizumi, K. Fujita, *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 15971.
[63] J. Zhou, C. Zhang, R. Wei, M. Han, S. Wang, K. Yang, L. Zhang, W. Chen, M. Wen, C. Li, W. Tao, Y. J. Yuan, *Sci. China: Life Sci.* **2022**, *65*, 851.
[64] M. E. Allentoft, M. Collins, D. Harker, J. Haile, C. L. Oskam, M. L. Hale, P. F. Campos, J. A. Samaniego, M. T. Gilbert, E. Willerslev, G. Zhang, R. P. Scofield, R. N. Holdaway, M. Bunce, *Proc. Biol. Sci.* **2012**, *279*, 4724.
[65] M. T. Weinstock, E. D. Hesek, C. M. Wilson, D. G. Gibson, *Nat. Methods* **2016**, *13*, 849.
[66] Q. Qin, C. Ling, Y. Zhao, T. Yang, J. Yin, Y. Guo, G. Q. Chen, *Metab. Eng.* **2018**, *47*, 219.
[67] F. Brennen, kmer_models, https://github.com/nanoporetech/kmer_models/ (accessed: January 2023).
[68] NCBI, Open Reading Frame Finder, https://www.ncbi.nlm.nih.gov/orffinder/ (accessed: January 2023).
[69] A. F. A. Smit, R. Hubley, P. Green, RepeatMasker, http://repeatmasker.org (accessed: January 2023).

DNA Image Storage Using a Scheme Based on Fuzzy Matching on Natural Genome

Jitao Zhang[1,2], Shihong Chen[3,4,5], Haoling Zhang[2,3,4,6], Yue Shen[2,3,4,5,6(✉)], and Zhi Ping[2,3,4,6(✉)]

[1] College of Life Sciences, University of Chinese Academy of Sciences, Beijing 101408, China
[2] BGI-Shenzhen, Shenzhen 518083, China
[3] Guangdong Provincial Key Laboratory of Genome Read and Write, BGI-Shenzhen, Shenzhen 518120, China
[4] George Church Institute of Regenesis, BGI-Shenzhen, Shenzhen 518120, China
[5] China National GeneBank, BGI-Shenzhen, Shenzhen 518120, China
[6] Shenzhen Institute of Synthetic Biology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
{shenyue,pingzhi}@genomics.cn

Abstract. Among lots of emerging storage technologies, DNA storage is with great potential for its high data storage density and low maintenance cost. However, DNA synthesis and sequencing, the two enabling technologies for DNA storage, are of high cost and inefficient in information writing and reading, which postpones the commercialization of DNA storage. Considering the expensive DNA synthesis cost, a DNA storage system based on natural genomes is devised to compress images by using fuzzy matching and image processing technology, which can reduce the cost of storing images in the DNA medium. According to our devised DNA storage scheme, the number of nucleotide sequences to be synthesized can be reduced by about 90% and the visual quality of retrieved images can be compared with conventional algorithms. Furthermore, because of no dependence among index sequences generated by fuzzy matching, the robustness of our scheme is better than that of those DNA storage schemes directly using conventional algorithms to compress images. Finally, we have investigated the factors that may influence images' fuzzy matching, including genome size, GC content and relative entropy, which can be used to design a criterion to select better genomes for a given image.

Keywords: DNA-based data storage · DNA synthesis · Fuzzy matching · Super-resolution · Image denoising

1 Introduction

With the rapid development of information technology, the ability of creating data becomes more and more powerful. According to previous report, the total amount of global data is predicted to grow from 45 Zettabytes (ZB) to 175 ZB in 2025 [1].

Y. Wei et al. (Eds.): ISBRA 2021, LNBI 13064, pp. 572–583, 2021.
https://doi.org/10.1007/978-3-030-91415-8_48

Tremendous growth of global data demands a much higher information storage capacity. DNA-based data storage is an emerging technology for its nonvolatility, remarkable durability, incomparable storage density and capability for cost-efficient information duplication [2–8]. The general steps of DNA storage include encoding, writing (DNA synthesis), reading (DNA sequencing) and decoding. The binary data stream of digital files is transcoded into DNA sequences and these sequences will be synthesized into oligonucleotides (oligos) or double-stranded DNA fragments for storage in the writing procedure. These sequences will be sequenced by utilizing sequencers and then decoded into original files when data retrieval.

Although DNA-based data storage has significant advantages over traditional storage technologies, high cost of synthesis is a key factor to hinder its commercialization [6, 9]. Another issue of DNA storage is that DNA synthesis currently is an error-prone process [5, 8–10], which may result in difficulties of accurate data retrieval. These errors can be addressed by introducing error-correction code (ECC) into DNA storage, which can reconstruct the missing information or correct errors but at the cost of additional DNA synthesis [4, 11]. Another strategy to avoid errors and maintain low synthesis cost was proposed by Tabatabaei et al. [12] and it was called DNA punch cards. It was designed to modify the topology of native DNA sequences to store data via enzymatic nicking. Although this strategy is benefit to the cost reduction of DNA synthesis, it greatly sacrifices information density, which is supposed to be one of the major advantages of DNA storage [13].

In this paper, we propose a novel DNA storage scheme based on native genomes to archive images. The principle of the scheme is to search the most similar sequence in a native genome for each encoded nucleotide sequence of images (referred hereinafter as fuzzy matching) and the indices of searched sequences in the genome will be recorded and then encoded into nucleotide sequences for synthesis. The difference between matched genome sequences and encoded nucleotide sequences of an image may cause noises or distortions during information retrieval but can be reduced by image processing technologies. To improve the visual quality of retrieved noisy images, an image denoising method and a super-resolution (SR) method are introduced into our DNA storage scheme. In total, since the fuzzy matching and image downsampling applied in our elaborate scheme, the data of an image is compressed into index sequences and the amount of nucleotide sequences synthesized to store the image can be reduced by 90.62% at most. Common image compression algorithms can achieve even better compression rates, but binary digits of compressed images are associated with each other, which can result in catastrophic error propagation when undesired errors occur. In comparison, oligos representing indices of matched sequences in our scheme are not interdependent, resulting in better robustness of the scheme when image retrieval. Furthermore, we analyzed three factors that may affect the quality of image retrieval.

2 Data and Software Availability

2.1 Image Dataset

The images used in this study belong to three datasets, including DIV2K [14], Set5 [15], Set14 [16]. DIV2K is an image dataset with 2k resolution used in some computer vision

challenges from 2017 to 2018, including 800 training images, 100 testing images, and 100 images for validation. The other two datasets contain 5 and 14 images, respectively. The images in these datasets are 24-bit images, most of which are color images. They are commonly used in computer vision research.

2.2 Genome Files and Source Code

Genome files are reference sequences downloaded from the genome database in NCBI. The genome for the experiment of comparing different transformation rules is the genome of *Saccharomyces cerevisiae S288C*, of which the genome size is 12.16 Mb and the GC content is 38.2%. 20 genomes and 100 genomes are used in the experiments to investigate the influence of genome size and GC content, respectively and the GC content of them is from 20% to 70%. The source codes include the implementation of fuzzy matching, image processing algorithms and simulation experiments. The list of genome files and source code are available in the github repository https://github.com/zhangjitaoBGI/DNA-storage-based-on-reference-genome.

2.3 Testing Environment

Google colabpro, GPU: Tesla V100, python 3.7.10, pytorch 1.7.1, cuda 10.1.

3 Methods

3.1 Overview of Genome-Based DNA Storage System

In order to achieve information compression for reducing the number of nucleotides to be synthesized, we devised a genome-based DNA storage system to store images, via searching a nucleotide sequence similar to an encoded sequence of an image in a natural known genome. As Fig. 1 shows, in this system, an image is first downsampled using bicubic kernel function [17] to reduce the data size. The downsampled image is divided into several blocks according to a square matrix and pixels in each block are represented by a nucleotide sequence of a natural genome. Then, the indices of genome sequences will be recorded and further encoded into nucleotide sequences for synthesis by any reported encoding method. More specifically, pixels in each block are rearranged into a mean-difference sequence, which includes the mean of these pixels and the differences between the mean and each pixel. To select a genome sequence to represent each mean-difference sequence, all nucleotide sequences of a genome are converted into digital sequences according to some transforming rules. Combined with their indices, these digital sequences are used to construct a digital sequences dictionary and the indices of selected digital sequences will be recorded for data retrieval. Apart from all indices of an image, other necessary information for decoding (e.g. taxonomy ID (txid) of the genome, the transforming rules, etc.) is called file header, which is also encoded into nucleotide sequences according to the same encoding method.

To retrieve a stored image, DNA strands related to the image are sequenced and decoded to generate indices of genome sequences and a file header. Based on the genome

information and transforming rules in the file header, the digital sequences dictionary can be reconstructed and these indices can be retrieved in it to find all mean-difference sequences of the stored image. By adding each difference to a mean within each sequence, pixels can be calculated and the stored image can be recovered.

To improve the visual quality of reconstructed images, image denoising is used to reduce the noise caused by fuzzy matching. Then, a SR method based on deep learning is applied to scale the denoised image back to its original size. As long as the length of each index in the digital sequence dictionary is shorter than that of each mean-difference sequence, the data of downsampled images will be compressed, and thus it reduces the number of nucleotide sequences to be synthesized.

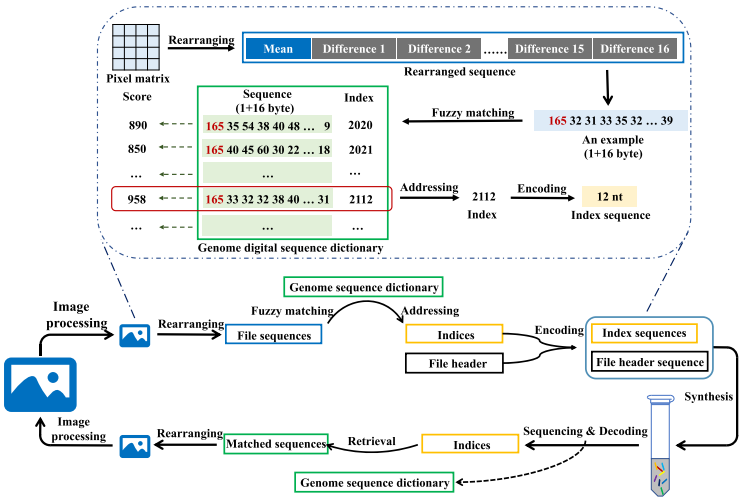

Fig. 1. The workflow of the DNA-based storage scheme. Lower: the overall procedures to store and read an image. Upper: an example of transforming a block of pixels into an index in a natural genome.

3.2 Transforming Nucleotide Sequences into Digital Sequences

As shown in Fig. 1, a key step in our DNA-based data storage system is fuzzy matching, whose target is to find the most similar oligo in a given genome for each mean-difference sequence of an image. To make full use of human visual system's characteristics, one of which is more sensitive to luminance, the RGB color channels of color images are converted into YCbCr channels. Considering color images' three channels, pixels in each channel of an image are divided by one matrix and mean-difference sequences of three channels are fuzzy matched in the same genome respectively. Here, we take one channel as an example to show the details of fuzzy matching. In the process of fuzzy matching, a mean-difference sequence corresponding to one block of an image is comprised of one mean and m differences which are integers in the range of [0,255] and [−31,32] (differences out of this interval are assigned to the nearest value, −31 or 32),

respectively. For convenience, all differences plus 31 and the interval becomes [0,63]. To map each integer to a nucleotide sequence, a quadruplet and a triplet are selected to represent a mean and a difference, respectively.

In Fig. 2a, the process to transform a genome into a digital sequence dictionary can be divided into two steps. The first step is to transform a genome into an oligo dictionary and each oligo is generated by segmenting the genome sequence with a sliding window whose step size is 1 and the window size is $4 + 3$ m nt (m is equal to the count of pixels in one block mentioned in Fig. 1, e.g., 4 or 16). Then, the second step is to convert the oligo dictionary into a digital sequence dictionary and three different rules are designed to implement the process (Fig. 2b). Two kinds of strategies, encoding and mapping, are used in these three rules. Encoding means a nucleotide sequence is directly encoded into an integer based on a simple encoding scheme.

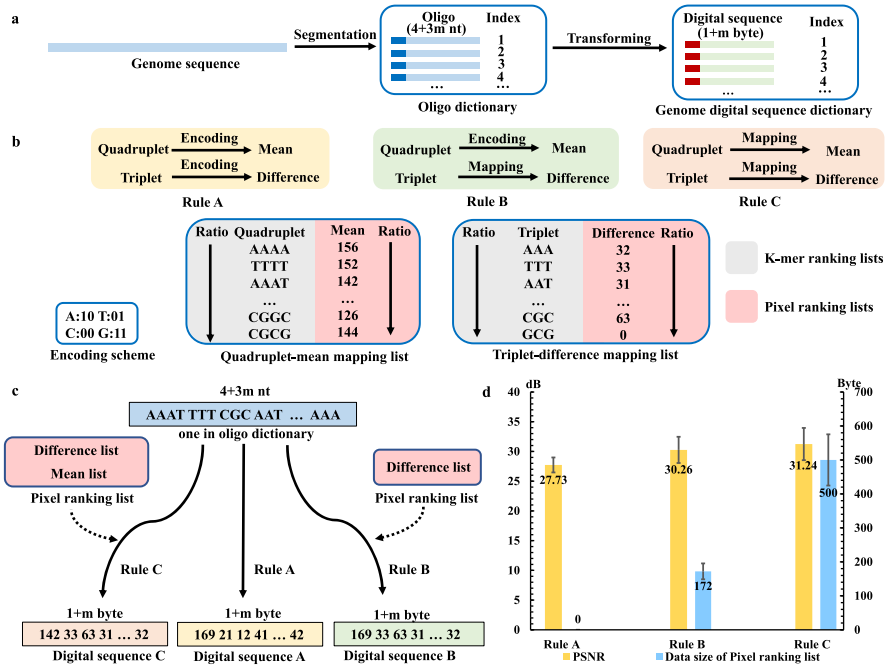

Fig. 2. Transforming nucleotide sequences into digital sequences. a. An oligo dictionary generated by segmenting a genome is transformed into a digital sequence dictionary to accomplish fuzzy matching. **b.** The principle of three rules applied to transform oligo dictionaries. The encoding scheme is used to directly encode integers and the two mapping lists are used to map an integer to a k-mer. **c.** An example shows an oligo in the oligo dictionary is transformed into different digital sequences according to three rules. **d.** The average peak signal-to-noise ratio (PSNR) of reconstructed images and average bytes of pixel ranking lists when three rules are used to fuzzy match.

Another strategy, mapping, denotes a quadruplet or triplet is mapped to an integer according to a related mapping list. By collecting a k-mer [18] ranking list of the genome

and a pixel ranking list of the image and establishing the mapping relations between them, a mapping list can be generated. K-mer ranking lists include a quadruplet list and a triplet list, which are generated by counting the relative frequency of quadruplets and triplets of oligos in an oligo dictionary and ranking their ratios in descending order. Pixel ranking lists also include two kinds of lists comprised of means and differences respectively, which are generated by counting the relative frequency of means and differences of all mean-difference sequences in one channel. Following means and differences are mapped to quadruplets and triplets respectively, a quadruplet-mean mapping list and a triplet-difference mapping list can be generated by these four lists. According to these mapping lists, an oligo dictionary can be transformed into a digital sequence dictionary. Considering two sections of each oligo, three rules are designed to transform each oligo into a digital sequence by utilizing different strategies in two sections. Figure 2c shows an example to transform a nucleotide sequence based on different rules.

3.3 Image Denoising

When stored images are obtained by sequencing and decoding, these images need to be denoised in order to reduce the information loss generated by the encoding and fuzzy matching procedures. The image denoising scheme used in this study is based on the method proposed by Jeremy in his introduction of the deep learning framework fastai [19]. The input images with noise are divided into a training set as well as a test set in the ratio of 9:1, using a batch size and an image size of 32, 128, respectively. U-net, a widely used image denoising model [20], is constructed using the pre-training model Resnet34 [21], which is a network can be trained end-to-end by using very few images and fast. To enable the trained model to perceive the difference between the generated image and the target image, perceptual loss [22] is introduced into the image denoising model.

3.4 Image Super-Resolution

Image super-resolution is to recover high-resolution (HR) images with better visual quality and refined details from low-resolution images [17, 23]. In our scheme, the scaling factor is $\times 2$ and the denoised images is a half of original images because of the down-sampling process. To scale the denoised images to its original images, a SR model based on deep learning, Enhanced SRGAN (ESRGAN) [24], is introduced to scale the denoised images. SRGAN [25] is a generative adversarial network (GAN) for SR task, which can generate photo-realistic natural images based on downsampled images. ESRGAN model improved the structure of SRGAN and it won the first place in the PIRM2018-SR Challenge. Considering the open nature, stability and remarkable performance of ESRGAN, it is selected to solve our SR task.

3.5 The Compression Rate

The definition of compression ratio (CR) in image compression is the ratio between uncompressed image size and compressed image size [26]. In this scheme, the compression rate is the reciprocal value of CR and its definition is the ratio between the data

size of indices of an image and the data size of pixels of the image (the unit of data size is byte). The data of an image to be stored is reduced when the image is processed by downsampling and fuzzy matching. In total, the compression rate of the image can be calculated by following formula:

$$\begin{aligned} c &= \frac{1}{s^2} \times \frac{1}{3} \times \left(\frac{\frac{\lceil \log_2 v_1 \rceil}{8}}{x^2} + \frac{\frac{\lceil \log_2 v_2 \rceil}{8}}{y^2} + \frac{\frac{\lceil \log_2 v_3 \rceil}{8}}{z^2}\right) \\ &= \frac{1}{24s^2}\left(\frac{\lceil \log_2 v_1 \rceil}{x^2} + \frac{\lceil \log_2 v_2 \rceil}{y^2} + \frac{\lceil \log_2 v_3 \rceil}{z^2}\right) \end{aligned} \tag{1}$$

where, s is the scaling factor of an image, x, y, z are the side of matrixes used to divide Y, Cb, Cr color channels of the image, respectively. v_1, v_2, v_3 are the number of sequences in the digital sequence dictionary corresponding to the Y, Cb, Cr channels.

4 Result

4.1 The Comparison Among Three Transforming Rules

To evaluate these rules, an experiment was performed to measure noise introduced in the fuzzy matching process and the data size of pixel ranking lists that these rules were dependent on. PSNR can measure the difference between an image to be stored and its related noisy image generated by fuzzy matching. Higher PSNR implies smaller difference and better result of fuzzy matching. Both k-mer ranking lists and pixel ranking lists are necessary to construct mapping lists in rule B and rule C, but k-mer ranking lists can be recovered by utilizing the genome information and parameters saved in the file header instead of storing the whole lists. The data set used to perform this experiment was the train set of DIV2K and the downsampled image size was 128×128. The average PSNR of the data set is shown in Fig. 2d. It's evident that the PSNR of rule C is better than that of other rules, but another factor, the size of pixel ranking lists it required, must be put into consideration. In rule B, the pixel ranking list is a difference list and it needs to be stored. On the basis of rule B, the sequence recording pixel ranking lists becomes longer because a mean list also needs to be stored when rule C is applied to transform oligos. It is a tradeoff between PSNR and the size of pixel ranking list to be stored and higher PSNR requires more nucleotide sequences. Rule B can significantly enhance PSNR of the reconstructed images at the cost of storing relatively small size of pixel ranking list and rule C is a better choice for an image with large size. Considering the image size of our data set is 128×128, images generated by following rule B are chosen to perform subsequent image processing experiments.

4.2 Image Processing

When images stored in form of oligos are retrieved, images with noises introduced in the fuzzy matching need to be denoised. To reduce noises of reconstructed images, a model based on U-net architecture and perceptual loss [19] was introduced into the DNA storage system. In Fig. 3a, compared with original images, there are some distortions

in the edge of items in stored images and a lot of them are eliminated in the denoised images, resulting in better visual quality and better PSNR and Structural similarity [27] (SSIM).

Apart from image denoising, ESRGAN was utilized to scale denoised images to the size of original images before downsampling. As shown in Fig. 3b, the scaled images are realistic and its visual quality is close to that of compressed images processed by JPEG algorithm. The PSNR and SSIM of scaled images, two common metrics to measure image difference, are much lower than those of JPEG images, but these two metrics can not reflect visual quality of reconstructed images precisely and the subjective human visual perception is more important in practical work [24]. With the development of the SR method, image quality has great potential to be improved to process lower-resolution images when larger scaling factors are selected, which can result in a lower compression rate.

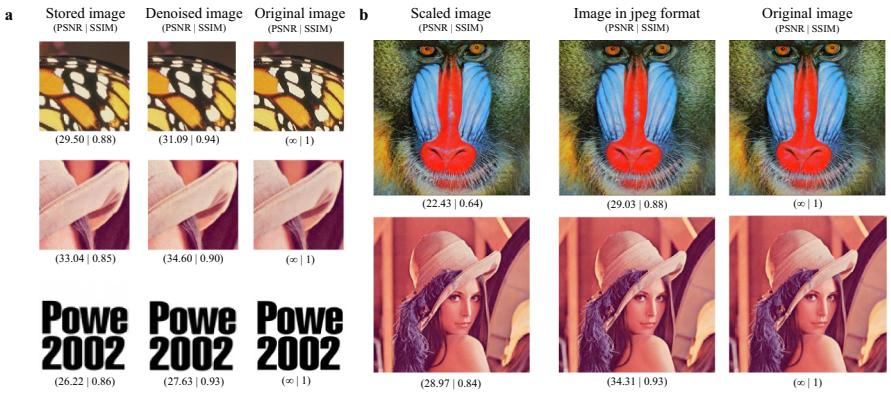

Fig. 3. **a.** The Comparison among stored images, denoised images and original images before processing. **b.** The Comparison among scaled images using the ESRGAN model, JPEG images with about 10% of compression rate and unpressed original images.

4.3 The Compression Rate of the Genome-Based DNA Storage System

To save the cost of DNA synthesis, data of images is compressed in our devised DNA storage system. The information to be stored is divided into a file header and index sequences. The data in the file header includes a fixed-length part and a run-length part. The fixed-length part records txid of the genome used in fuzzy matching and the parameters used to construct digital sequence dictionaries, while the run-length part keeps pixel ranking lists. The compression rate of this system depends on the transforming rule and the scaling factor. In this experiment, the size of original image was 256 × 256, the downsampling factor was ×2, the matrixes used to divide each image's three channels were 2 × 2, 4 × 4, 4 × 4 respectively and the data size of each index in digital sequence dictionaries was 3 bytes (the count of sequences in each dictionary constructed by the genome *Saccharomyces cerevisiae S288C* was about 12

million and 2^{24} could cover all sequences' indices). The fixed-length parts of three rules were 13 bytes,16 bytes, 16 bytes respectively and the run-length parts are shown in Fig. 2d. In total, the length of file header of three rules were 13 bytes, 188 bytes, 516 bytes respectively and the size of index sequences was 18,432 bytes. The average data size of 800 images used to compare three rules was 196,662 bytes and the compression rates of these three rules were 9.38%, 9.47%, 9.63%, respectively. The format of stored images was BMP and the data size of each image could be reduced by more than 90% under above conditions, which implies the synthesis cost to store images can be highly reduced. If the downsampling factor or the matrix size used in each image channel increases for a given genome, the compression ratio and synthesis cost will further decrease.

4.4 Factors to Influence the Result of Image's Fuzzy Matching in Genomes

Although the procedure of fuzzy matching has been developed, factors that may have an effect on its result need to be investigated to establish a criterion for selecting a better genome in the fuzzy matching procedure. It's obviously that genome size is an important factor, because a nucleotide sequence with fewer differences is more likely to be searched in a larger genome with more kinds of nucleotide sequences. Meanwhile, the GC content is also a common feature of a genome, which determines the ratios of four nucleotides in a genome. In addition, the mapping lists used in the fuzzy matching are based on the distributions of k-mers and pixels. When an image is matched in different genomes, the difference between pixels' frequency distribution of the image and k-mers' frequency distribution in each genome is different, which may also influence the matching result. Relative entropy, a measure of the difference between a probability distribution and a reference probability distribution, is introduced to explore the influence of this factor. Smaller relative entropy implies smaller difference of two probability distributions. In these investigations, images were fuzzy matched in sub-dictionaries with specific number of digital sequences, which were generated by randomly selecting digital sequences in the whole digital sequence dictionaries constructed by genomes. This strategy could precisely control the size of dictionary used in the fuzzy matching.

In Fig. 4a, 20 genomes and 200 images in DIV2K dataset were selected to perform fuzzy matching experiment. Five kinds of genome size were explored in the experiment. The average PSNR of 200 images matched in each sub-dictionary was calculated to reflect the result of fuzzy matching. With increased genome size, the mean PSNR of twenty genomes also increases. At the same time, larger genome size also increases the length of index in the digital sequence dictionary, resulting in higher compression rate and synthesis cost. Trading off between PSNR and compression rate should be put into consideration when choosing the size of genomes.

The effect of GC content is shown in Fig. 4b 100 genomes were divided into 10 groups based on their GC content and 800 images in DIV2K training set were fuzzy matched in each of 100 genomes. The size of each genome's sub-dictionaries was unified into 4^{11}. The PSNR of each group is the mean value of ten genomes' PSNR. In Fig. 4b, the trend of 10 groups' histograms is just like a saddle. When GC content varies from 50% to 20% or 70%, the average PSNR of reconstructed images in each group rises up, which shows choosing genomes with GC content close to 50% is not a good choice and genomes with relatively extreme GC content have higher priority.

Apart from those two factors, the relationship between PSNR and relative entropy is demonstrated in Fig. 4c. When nucleotide sequences were transformed into digital sequences, 64 triplets of genomes were mapped to differences in [0,63] according to their relative frequencies. The relative entropy between each of 800 images and a genome was calculated based on the data in Fig. 4b and the PSNR and relative entropy when 800 images were fuzzy matched in each of 800 genomes are shown in Fig. 4c. It's clear that there is a relative strong negative correlation between PSNR and relative entropy (R^2 is 0.7738), but choosing a genome for a given image only based on relative entropy is not a reliable strategy. Although these three factors provide a reference when choosing genomes for a given image, more features of genomes and images need to be collected to develop a better scheme to accomplish such a task.

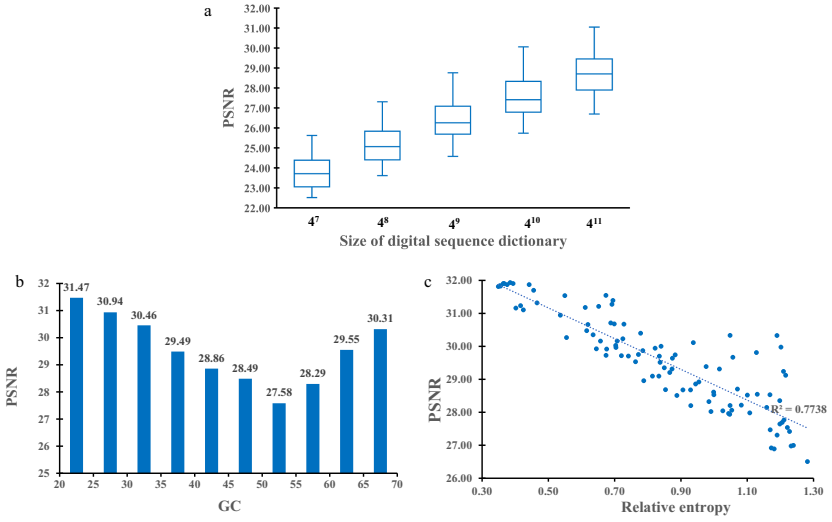

Fig. 4. Three factors to influence fuzzy matching result. a. The average PSNR of reconstructed images at different genome size. The genome size represents the number of nucleotide sequences in sub-dictionaries constructed by each of 20 genomes used in this experiment. The box plot of each genome size is the statistical result of 20 genomes' PSNR (each genome's PSNR was 200 reconstructed images' average PSNR when downsampled images were matched in the sub-dictionary related to the genome). **b.** The average PSNR of reconstructed images when images were fuzzy matched in genomes with different GC content. **c.** The average PSNR of reconstructed images and average relative entropy between the genome and 800 images when images in the dataset were fuzzy matched in each of 100 genomes. The R^2 shows the correlation between PSNR and relative entropy.

5 Conclusion

Based on readily available natural genomes, we designed a DNA storage system to store images in a lossy compression manner, which can mitigate the prohibitively high cost of

DNA synthesis before great progress has been made in the DNA synthesis technology. Following our scheme, the number of nucleotide sequences to store an image in DNA medium can be reduced by about 90% and the visual quality of recovered images is comparable to that of images compressed by JPEG algorithm. What's more, we also explored three different factors that may have an effect on fuzzy matching of images, but more features of images and genomes need to be explored to devise a strategy to select better genomes for a given image. However, we note that visual quality of recovered images can be further improved and more customized image processing algorithms should be developed.

Acknowledgment. This work was supported by the National Key Research and Development Program of China (No. 2020YFA0712100) and the Guangdong Provincial Key Laboratory of Genome Read and Write (No. 2017B030301011). We are thankful for support on computing resource provided by China National GeneBank (CNGB).

References

1. Reinsel, D., Gantz, J., Rydning, J.: Data age 2025: the digitization of the world from edge to core. Seagate Data Age, 1–28 (2018)
2. Church, G.M., Gao, Y., Kosuri, S.: Next-generation digital information storage in DNA. Science **337**(6102), 1628 (2012)
3. Goldman, N., et al.: Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. Nature **494**(7435), 77–80 (2013)
4. 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. **54**(8), 2552–2555 (2015)
5. Tabatabaei Yazdi, S.M.H., Yuan, Y., Ma, J., Zhao, H., Milenkovic, O.: A rewritable, random-access DNA-based storage system. Sci. Rep. **5**(1), 1–10 (2015)
6. Zhirnov, V., Zadegan, R.M., Sandhu, G.S., Church, G.M., Hughes, W.L.: Nucleic acid memory. Nat. Mater. **15**(4), 366 (2016)
7. Erlich, Y., Zielinski, D.: DNA Fountain enables a robust and efficient storage architecture. Science **355**(6328), 950–954 (2017)
8. Ceze, L., Nivala, J., Strauss, K.: Molecular digital data storage using DNA. Nat. Rev. Genet. **20**(9), 456–466 (2019)
9. Dong, Y., Sun, F., Ping, Z., Ouyang, Q., Qian, L.: DNA storage: research landscape and future prospects. National Sci. Rev. **7**(6), 1092–1107 (2020)
10. Bornholt, J., Lopez, R., Carmean, D.M., Ceze, L., Seelig, G., Strauss, K.: A DNA-based archival storage system. In: Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 637–649 (2016)
11. Li, B., Ou, L., Du, D.: Image-based Approximate DNA Storage System. arXiv preprint arXiv:2103.02847 (2021)
12. Tabatabaei, S.K., et al.: DNA punch cards for storing data on native DNA sequences via enzymatic nicking. Nature Commun. **11**(1), 1–10 (2020)
13. Han, M., Chen, W., Song, L., Li, B., Yuan, Y.: DNA information storage:bridging biological and digital world. Synthetic Biol. J. 1–14 (2021)

14. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 126–135 (2017)
15. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: Proceedings of the 23rd British Machine Vision Conference (BMVC), pp. 135.1–135.10. BMVA Press (2012)
16. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Boissonnat, J.-D., et al. (eds.) Curves and Surfaces 2010. LNCS, vol. 6920, pp. 711–730. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27413-8_47
17. Wang, Z., Chen, J., Hoi, S.C.: Deep learning for image super-resolution: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **1**, 3365–3387 (2020)
18. Compeau, P.E., Pevzner, P.A., Tesler, G.: Why are de Bruijn graphs useful for genome assembly? Nat. Biotechnol. **29**(11), 987 (2011)
19. Jeremy, H.: https://nbviewer.jupyter.org/github/fastai/course-v3/blob/master/nbs/dl1/lesson7-superres.ipynb. Accessed 19 Jul 2021
20. Komatsu, R., Gonsalves, T.: Comparing u-net based models for denoising colorimages. **AI1**(4), 465–486 (2020)
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241 (2015)
22. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
23. Anwar, S., Khan, S., Barnes, N.: A deep journey into super-resolution: a survey. ACM Comput. Surv. (CSUR) **53**(3), 1–34 (2020)
24. Wang, X., et al.: ESRGAN: enhanced super-resolution generative adversarial networks. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11133, pp. 63–79. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11021-5_5
25. Ledig, C., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4681–4690 (2017)
26. Yu, H., Winkler, S.: Image complexity and spatial information. In: 2013 FifthInternational Workshop on Quality of Multimedia Experience (QoMEX), pp. 12–17. IEEE (2013)
27. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, pp. 1398–1402 (2003)