small methods
www.small-methods.com

An Effective DNA-Based File Storage System for Practical Archiving and Retrieval of Medical MRI Data

Abdur Rasool, Jingwei Hong, Zhiling Hong, Yuanzhen Li, Chao Zou, Hui Chen, Qiang Qu, Yang Wang, Qingshan Jiang, Xiaoluo Huang,* and Junbiao Dai**

DNA-based data storage is a new technology in computational and synthetic biology, that offers a solution for long-term, high-density data archiving. Given the critical importance of medical data in advancing human health, there is a growing interest in developing an effective medical data storage system based on DNA. Data integrity, accuracy, reliability, and efficient retrieval are all significant concerns. Therefore, this study proposes an Effective DNA Storage (EDS) approach for archiving medical MRI data. The EDS approach incorporates three key components (*i*) a novel fraction strategy to address the critical issue of rotating encoding, which often leads to data loss due to single base error propagation; (*ii*) a novel rule-based quaternary transcoding method that satisfies bio-constraints and ensure reliable mapping; and (*iii*) an indexing technique designed to simplify random search and access. The effectiveness of this approach is validated through computer simulations and biological experiments, confirming its practicality. The EDS approach outperforms existing methods, providing superior control over bio-constraints and reducing computational time. The results and code provided in this study open new avenues for practical DNA storage of medical MRI data, offering promising prospects for the future of medical data archiving and retrieval.

1. Introduction

Deoxyribonucleic acid (DNA) has garnered significant attention from researchers due to its substantial storage potential. DNA storage offers six times higher density than tape storage and a preservation span extending hundred times longer than traditional media.[1] Furthermore, DNA allows easy replication through molecular biology techniques to copy data.[2,3]

DNA data storage is intricately linked with synthetic biotechnology and computer technology. Computer file systems (CFS), such as the widely adopted new technology file system (NTFS), play a pivotal role in managing and organizing digital data on storage devices, as illustrated in **Figure 1A**. Similarly, a DNA file system (DFS) holds comparable significance concerning data organization and information access. Constructing

A. Rasool, J. Hong, Y. Li, C. Zou, Q. Qu, Y. Wang, Q. Jiang, X. Huang, J. Dai
Shenzhen Institute of Advanced Technology
Chinese Academy of Sciences
Shenzhen 518055, China
E-mail: qs.jiang@siat.ac.cn; huangxl@siat.ac.cn; junbiao.dai@siat.ac.cn

A. Rasool
Shenzhen College of Advanced Technology
University of Chinese Academy of Sciences
Shenzhen 518055, China

J. Hong
College of Mathematics and Information Science
Hebei University
Baoding 071002, China

Z. Hong
Quanzhou Development Group Co., Ltd
Quanzhou 362000, China

Y. Li, X. Huang
Shenzhen Key Laboratory of Synthetic Genomics
Guangdong Provincial Key Laboratory of Synthetic Genomics
Key Laboratory of Quantitative Synthetic Biology
Shenzhen Institute of Synthetic Biology
Shenzhen 518055, China

H. Chen
Shenzhen Polytechnic University
Shenzhen 518055, China

J. Dai
Shenzhen Branch
Guangdong Laboratory of Lingnan Modern Agriculture
Genome Analysis Laboratory of the Ministry of Agriculture and Rural Affairs
Agricultural Genomics Institute at Shenzhen
Chinese Academy of Agricultural Sciences
Shenzhen 518055, China

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/smtd.202301585

DOI: 10.1002/smtd.202301585

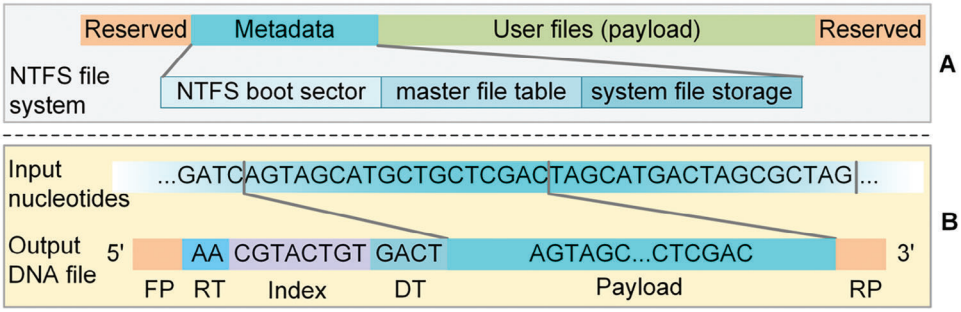

Figure 1. Architectural overview of file systems. A) the latest file system of the computer, NTFS. B) output DNA file structure with forward and reverse primer (FP, RP), regular and distinct tags (RT, DT), index, and payload for information storage.

an effective DFS is critical to ensuring data integrity, accuracy, reliability, and recovery. However, as the DNA storage medium differs fundamentally from CFS, unlike CFS, DNA storage relies on encoding models, bio-constraints, and sequencing technologies.[1,3]

One critical aspect of DNA storage is the encoding models that convert binary data into DNA files. However, existing models often overlook specific data types, such as compressed essential image files before storage.[4–6] This oversight can lead to extensive error propagation and distorted reproductions during decompression. Addressing this issue requires developing encoding models that account for diverse data types. In addition to data types, it is crucial to consider various bio-coding constraints in DNA storage. While some studies focus on GC content and homopolymer (HP) constraints,[7,8] overlooking other constraints, such as reverse complement (RC) constraints,[9] can jeopardize the reliability of the DNA storage system.[10–13] An efficient and robust DNA data storage system must take into account these vital bio-constraints to ensure long-term effectiveness.[14]

Simultaneously, several state-of-the-art encoding models[2,11,13,15–17] have been reported in recent years for robust DNA storage. Among them, rotating encoding has been utilized in cutting-edge studies.[2,17,18] However, even minor alterations (insertions, deletions, or substitutions) of nucleotides (nt) within a DNA sequence can trigger error propagation, potentially affecting all succeeding subsequences. This could result in incorrect data decoding or loss due to a single nt error. To mitigate such errors, redundant nt are introduced in DNA strands, leading to DNA data overhead and other limitations. First, this significantly reduces storage density and escalates synthesis and sequencing expenses. Second, polymerase chain reaction (PCR) based random access requires an aliquot of the complete data pool for information retrieval, potentially requiring periodic amplification to recover the data. Each amplification introduces stochastic variation in file sequence copy numbers, resulting in potential data loss of up to 2% per amplification, as reported.[19] Additionally, the file sequences require meticulous strand design (shown in Figure 1B) by appropriately indexing the DFS to prevent such unintended and unspecific amplification of PCR primers, hybridization, and incorrect barcodes.[20] A new approach that competently handles the crucial issue of incorrect data decoding or data loss is highly desirable for practical DNA storage.

The Magnetic Resonance Imaging (MRI) approach produces high-resolution images that are critical for accurate diagnosis, treatment planning, and monitoring of a wide range of diseases, including brain disorders and musculoskeletal injuries.[21,22] MRI data has also become a significant tool in biomedical research, such as aging and hereditary disease progression studies. The use of DNA storage technology to preserve medical MRI data will benefit human health management. On one hand, the capability to safely store and precisely recover MRI data across millennia via DNA storage ensures the long-term survival of these vital data. On the other hand, it makes prior health data more accessible to future generations, which should be important for longitudinal studies since it allows researchers to study sickness progression and treatment efficacy over decades. In this study, we designed and experimentally tested a novel encoding approach for an Effective DNA file Storage system, termed EDS, addressing the challenges posed by prior encoding methods, which is tested by MRI data storage. The effectiveness of our approach lies in ensuring the reliability of DNA storage by adhering to bio-coding constraints, reducing computational time, and enhancing data retrieval. Our proposed approach offers a novel fraction strategy that divides MRI images into 16 equal chunks, effectively mitigating the issues associated with rotating encoding. Additionally, the EDS approach incorporates an innovative rule-based quaternary transcoding mechanism to convert digital files into binary code and map them into DNA bases. This mechanism satisfies bio-coding constraints by controlling the repeating bases, ensuring a reliable and robust mapping process. Furthermore, our approach introduces a new indexing system for DNA files, enabling random search and access while managing the high overhead challenges in DNA file storage. By achieving these objectives, our study presents a novel practical approach that minimizes error propagation with the satisfaction of combinatorial bio-coding constraints and reduces computational time. The EDS approach was experimentally applied to encode an MRI image into DNA files, with some of them were synthesized, stored, and sequenced as examples. Notably, the proposed approach exhibits enhanced fault tolerance for image data. Moreover, we extended our methodology to other data types, evaluating its effectiveness for encoding lossless files.

2. Results

2.1. Experimental Process

Computer simulations were performed on a Windows 10×64-based Intel-R Core i7-9700k@3.60 GHz system with 32 GB RAM,

Table 1. The performance comparison of the EDS approach with MRI full image, its chunks, and other files on various factors.

File Name	Size [kb]	DNA File [n]	Max GC	Min GC	Total GC	HP	Max Len	Min Len	Avg. Len	D[a)]
mri_1	5.63	386	0.6	0.4	49%	3	108	91	106	1.69
mri_3	2.58	177	0.6	0.408	49%	4	108	91	106	1.69
mri_5	6.14	420	0.627	0.406	49%	4	108	91	105	1.69
mri_7	5.84	399	0.6	0.4	49%	3	108	91	105	1.69
mri_9	6.95	476	0.6	0.4	49%	3	108	91	105	1.7
mri_11	5.66	387	0.6	0.408	50%	4	108	91	105	1.69
mri_13	5.73	393	0.6	0.4	49%	3	108	91	106	1.69
mri_15	2.47	170	0.6	0.397	49%	3	108	91	107	1.68
mri_full_img	71.90	18 021	0.6	0.4	50%	3	102	91	108	1.65
mri_report.pdf	56.62	3863	0.604	0.4	49%	3	109	91	107	1.66
mri-img_info.xml	16.77	1145	0.61	0.391	50%	3	107	91	106	1.68
encoding.py	15.84	1082	0.6	0.381	48%	3	113	91	108	1.66
MRI.rar	72[b)]	3 435 975 476	0.61	0.403	50.1	3	106	92	103	1.821

[a)] Density (D) bits/nucleotides (bits nt^{-1}) without primer. The density was calculated by dividing the length of each binary sequence without primer sequence by the length of the DNA sequence [b)] 72 GB data.

using Python 3.8.11v. Initially, MR images of the human body, along with associated metadata, an XML file containing patient information and acquisition parameters, were collected with approval from the ethics committee and the participation of volunteers. A randomly selected MR image, along with its metadata (XML and pdf), was used for DNA storage. Additional information about this dataset can be found in Section S1.1 (Supporting Information). The proposed fraction strategy divided the MR image into 16 equal chunks. Each sub-image (chunk) underwent processing to convert it into binary representation using the Python function *TransBin*. A distinct 4-bit tag was appended to the binary code of each chunk to ensure differentiation. Subsequently, the binary data of each chunk was segmented into multiple binary strings. The proposed rule-based quaternary transcoding method was then utilized to convert the binary data into premier DNA sequences (n), serving as optimal DNA fragments or DB. Users had the flexibility to adjust parameters and include additional file types. The experiments encoded and decoded state-of-the-art (SOTA) medical datasets. The encoded DNA sequences were processed using combinatorial bio-coding constraints to meet the criteria for optimal sequences, with a GC content of 40%–60% and a HP limit of ≤ 4.[13,23] Additionally, the study considered the RC constraint to prevent unwanted secondary structure formation that might interfere with DNA synthesis accuracy, thereby minimizing potential errors. This constraint was analytically evaluated using the Monte Carlo computational algorithm[24,25] applied to the DNA sequences. The wet lab experiments and their process are detailed in Experimental Section.

2.2. EDS Coding Potential

The findings presented in this study demonstrated the efficiency and suitability of the EDS approach for storing MR scans. A comparison (**Table 1**) between the full MR image and its chunks revealed that the chunks had a higher density (1.7) compared to the full image (1.65). Furthermore, non-image files, such as XML files containing MR image information, PDF files containing patient health reports, and Python files, were compared with the proposed EDS coding. The use of a fractioned strategy for storing the MR image led to increased density while adhering to significant biological constraints, such as GC ratio of ≈49% and HP≈3, distribution. Section S1.2 (Supporting Information) presents additional potential results: encoding and decoding with diverse SOTA medical datasets, GC content satisfaction and MRI of other body parts, revealing the highest density of 1.96 bits nt^{-1} and a GC ratio of 49%. For large-scale MRI data, we have experimented EDS approach with 72 GB of MRI data using advanced computing resources (3.53 GHz AMD EPYC 7763 CPU, 1024 GB memory). It efficiently encoded the dataset in 9.032 h, demonstrating EDS's efficiency and robustness. Further analysis of varying data sizes helped estimate encoding times for 1 TB level data, revealing significant time requirements even with high-performance computing. These findings are presented in Figure S1 (Supporting Information), highlighting the need for optimization for large-scale data processing.

Additionally, Monte Carlo simulations[24,25] were employed to evaluate the adherence of DNA sequences to RC constraints (Figure S2, Supporting Information). In the provided dataset of 51 gene oligos, the compliance rate (Cr) ranged from 0.000 to 0.034, with no rates below 0.00. The absence of $C_r < 0.00$ indicates that all sequences in the dataset satisfy 83% of RC constraints. Fulfilling these biological constraints implies that the DNA sequence files are more reliable and less prone to errors. By applying this simulation, gained insights into the effectiveness and reliability of DNA data storage systems, paving the way for successful decoding and retrieval of information with the proposed EDS approach. Detailed results and steps to implement this simulation are provided in Section S1.2.4 (Supporting Information).

2.3. File Retrieval and Error Mitigation

The proposed EDS has brought transformative changes to the DFS, enhancing file retrieval and reducing error propagation. By integrating rule-based transcoding and fraction strategies, it marks a breakthrough in data storage and retrieval efficiency.

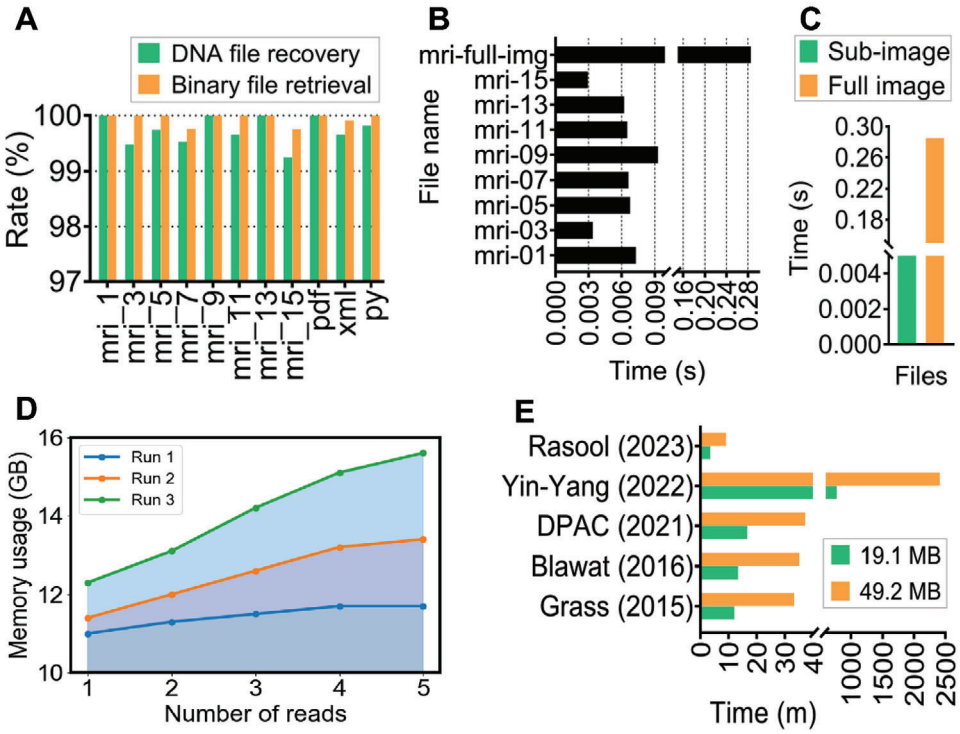

Figure 2. EDS performance based on computer simulation. A) The rate (%) of DNA file recovery and binary information recovery with chunks and other file types. B) Comparison of running-time (s) of the full image and its chunks. C) Average execution time of full image and sub-images. D) The memory utilization graph for the execution of EDS over multiple runs. E) Computational time comparison of prior works with EDS using medium-volume digital files.

In **Figure** 2A, a substantial improvement in the average DNA file recovery or fidelity rate is evident, standing impressively at 99.65%. Furthermore, the average digital file recovery or accuracy rate has reached a remarkable 99.95%, demonstrating the efficacy of this approach. In the context of computer simulations, this improvement becomes even more evident, with a 99.93% data recovery rate for image files and an extraordinary 99.97% for non-image files. The success of this achievement is attributed to the fraction strategy's exceptional capability to address rotating code errors and mitigate data loss issues during the screening process.

The significance of EDS is not merely theoretical; it is validated through rigorous mathematical modeling (see Section S1.3, Supporting Information) and computer simulations. These evaluations (Tables S3 and S4, Supporting Information) revealed the inherent advantages of the DFS within EDS, marked by its rule-based transcoding and innovative indexing tags. These features significantly enhance system throughput, facilitating easy search and access to the required DNA sequences. Importantly, the system's resilience in the face of errors is a testament to its robustness. Compared to prior work in Table S3 (Supporting Information), EDS can decode remaining data chunks and provide a substantial amount of original information. This resilience is upheld by strict adherence to the fraction strategy with the DFS.

2.4. Swift Efficiency

The scalability of the proposed approach was assessed by calculating execution times for chunks, full images, and other files. Figure 2B, a run-time comparison between chunks and full-image processing, highlighted the advantages of the fraction strategy. This strategy significantly reduces encoding-decoding time while maintaining high efficiency and reliability. For example, processing 5 KB of digital data into a DNA file sequence and its reverse operation requires a mere 0.0062 s. Larger files, such as 1 MB, can be processed within approximately 1.21 s. Moreover, the average time required for encoding and decoding chunks and full images is noteworthy. It is important to note that actual execution time may vary due to primary biological constraints, necessitating additional memory for larger files and interval adjustments between runs.

Memory utilization experiments were also conducted on two other machines: a MacBook (Quad-Core Intel i5) for run 2 and a personal computer (Intel Core i5-6500) for run 3 to compare EDS's memory consumption. Figure 2C offers insight into memory usage during sequence runs, demonstrating a rapid increase during the initial read and a steady increase in subsequent runs.

Additionally, a significant contribution lies in comparing execution or computation time with small and large-volume digital files. Figure 2D illustrates a substantial reduction in computation time when processing larger files compared to previous benchmark studies. On average, non-image files with a size of 9.91 KB can be encoded and decoded in a commendable 0.01 s. Similarly, our transcoding encodes a large file of 49.2 MB in 9 min and 12 s, whereas Yin-Yang[11] requires 2413 min within our experimental environment. Therefore, our approach provides compelling evidence of swift computational efficiency.

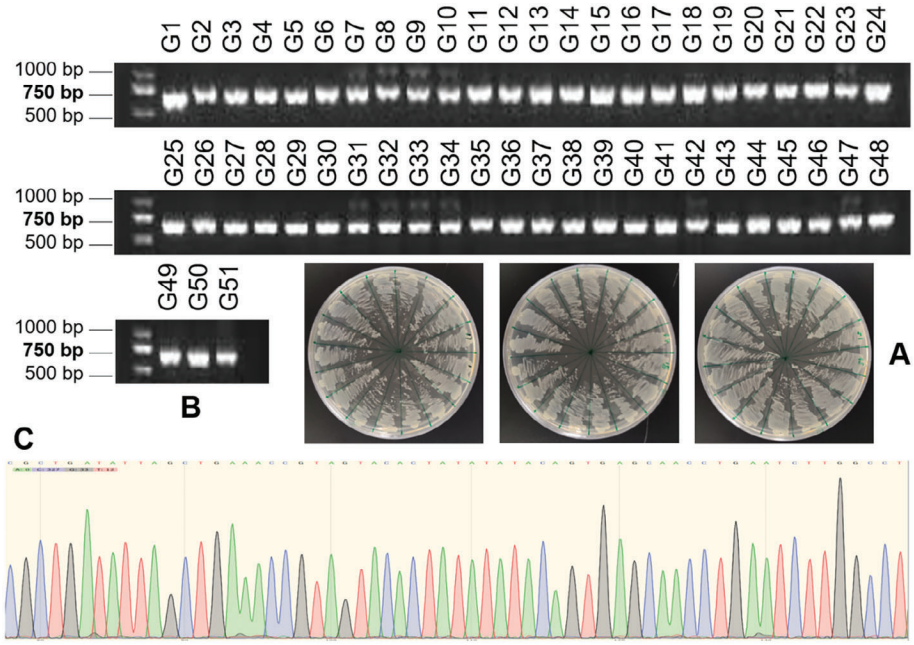

Figure 3. Validation of proposed EDS approach by biological experiments in vivo testing, using DNA sequences of all chunks of MRI (83.42 kb) and MRI report.pdf (56.62 kb). A) Storage of information in *E. coli* cells. The growth of *E. coli* cells is shown on plates. B) PCR amplification of the synthesized plasmid containing each gene. C) The peak result from Sanger-sequencing.

2.5. EDS Validation via Biological Experiments

The efficacy of EDS was initially confirmed through computer simulations. Further validation was conducted via biological experiments, demonstrating the integration of IT and BT for the end-to-end DNA data storage system. **Figure 3** displays the validation of the proposed EDS approach through a wet lab experiment involving digital files (140 KB), comprising 16 image files (83.4 KB) and one MRI report.pdf (56.6 KB). These files were encoded into DNA fragments, ready for DNA synthesis. Three DNA fragments (genes) were randomly selected from each of the 16 image files and one MRI report.pdf file, resulting in a total of 51 (17*3) sequences (see Tables S5 and S6, Supporting Information) stored in bacteria. This selection aimed to demonstrate the practicality of the proposed EDS approach for archiving MR image data into DNA. The stored information, in the form of synthesized plasmids, was stored both in vitro (in a tube) and in vivo (in *E. coli* cells). Sanger sequencing was performed to obtain sequencing results of the plasmids, which were further validated using universal primers and agarose gel electrophoresis. Furthermore, PCR amplification of mixed genes from each image file is highly successful, demonstrating the reliability of random access to gene stored data from each image component (Figure S4, Supporting Information). The results from computer simulations and biological experiments confirm the effective storage and precise retrieval of encoded sequences using the proposed EDS approach.

3. Discussion and Conclusion

The use of DNA for storing medical image data, such as MRI scans, holds immense significance in ensuring the long-term preservation and accessibility of vital medical information. This innovative approach not only facilitates improvement in patient care but also contributes to disease tracking, personalized medicine, and advanced diagnosis and treatment planning. By securely preserving vital medical MRI data in DNA, this method becomes indispensable for patients undergoing annual MRI scans. The present study, featuring a novel data storage system, signifies a substantial advancement in healthcare practices and research.

This study highlights three key advantages: (*i*) the importance of DNA file storage for medical records, (*ii*) the technical advantages presented by the EDS approach, and (*iii*) the practical effectiveness of this work. First, the storage of medical image data, including MR images alongside their metadata and related health reports, in DNA guarantees accurate diagnosis, well-informed treatment planning, and effective disease monitoring. This not only enhances patient outcomes but also advances medical knowledge significantly. Second, the EDS approach adeptly tackles encoding errors, satisfies bio-constraints, and introduces flexible indexing for DNA file management. Consequently, it ensures the reliable, accessible, and long-term preservation of health records and genetic disorder information for the benefit of future generations. Furthermore, this paper proposes an organized file system for MRI medical data storage. The proposed file system, like CFS, ensures that data is stored consistently and accurately. Considering DFS to be a new library system that arranges books (data) using DNA sequences rather than shelves, this method enables simple retrieval of information by PCR and DNA sequencing, revolutionizing how we store and retrieve data. It converts EDS through improved encoding and indexing, increasing data scalability and accessibility in medical research and diagnosis. The integration of DFS into our EDS technique thus

Table 2. The comparative analysis of the proposed EDS approach with critical parameters of prior studies.

Author	Church	Goldman	Erlich	Yazdi	Song	Welzel	Rasool
Year-Refs.	2012[15]	2013[2]	2017[13]	2020[26]	2022[4]	2023[27]	2023
Coding method	1 bit to 1 base	Rotating encoding	DNA fountain	14 bits to 8 bases	De Bruijn graph and greedy path search	Arithmetic coding, lossless method	EDS approach
Dataset	English text, JPG images, computer code	Text file, JPEG file, MP3 file	Text file, SVG file, Video file	Two JPEG images	Random 6.8MB	DNA sequences	Medical image, pdf, Python file, XML file
Sequence length	115	117	152	800–1000	190	156	106
Density*	0.83	0.33	1.57	1.74	1.3	N/A	1.8
File access	No	No	No	Yes	Yes	No	Yes
Bio-constraints	HP 3	Run length >2	GC (45%–55%), no-runlength 4	HP, GC content	GC content, Melting temperature	GC content (40%–60%), HP 3 or 4	GC content 50%, Hamming distance, RC 83%, HP ≈ 3

demonstrates the capability for structured storage and exact retrieval of large medical datasets.

The practical effectiveness of this study is demonstrated in the results section. The proposed approach achieves remarkable efficiency, with an average GC ratio of 50%, 1.8 bits nt^{-1} density with payload, and an average HP of 3, as indicated in Table 1 and Table S2, Figures S1 and S3 (Supporting Information). These parameters are critical as they reflect the reliability and robustness of the proposed DNA coding constraints. Additionally, the EDS approach supports various file types (PDF, XML, Python files), underscoring its versatility in processing different files (Table 1; Table S2, Supporting Information). Experimental validation confirms a 99.93% data recovery rate for image files (Figure 2A). This achievement can be attributed to the effective rule-based transcoding combined with the fraction strategy, ensuring the reliability of the proposed approach.

Moreover, it minimizes computational time, enabling the storage and retrieval of a 49.2 MB file in 9 min and 12 sec, outperforming previous findings (e.g., Yin-Yang[11] requires 2413 min, (Figure 2D). This improvement primarily stems from the innovative transcoding methodology, which holds potential for further optimization. The validity of DNA fragment storage is corroborated through biological experiments, as illustrated in **Figure 3** and Tables S5 and S6 (Supporting Information). Comparative analyses (Tables S3 and S4, Supporting Information) with SOTA rotating encodings demonstrate the advantages of the proposed approach in overcoming errors. Finally, the performance of the proposed EDS approach is compared against prior studies (**Table 2**), demonstrating its superiority in achieving maximum logical density while adhering to combinatorial bio-coding constraints, further underscoring its reliability. This highlights the effectiveness of our approach in the field.

The EDS system is demonstrated by preserving data from each image fraction in synthetic genes (around 600 bp each) and cloning them into plasmid vectors. The advantages of employing cloned synthetic genes for data storage involve the following points. 1) It is sequence-verified, which makes data retrieval considerably easier. Previous research employed oligonucleotides to store data that typically have mistakes in the sequence. This will significantly increase the effort required for data recovery. 2) It is easily reproduced and made in large quantities. After overnight culture and plasmid extraction, the plasmid cloned genes can be easily replicated via bacterial replication, yielding many copies. 3) The encoding density of gene-based DNA data storage is significantly higher than oligo-based DNA data storage. The present oligo synthesis is typically under 300 nt. However, gene synthesis may surpass 10 Kbp.[28,29] Given that index and random-access primers will occupy a portion of the DNA utilized for data storage, the longer the DNA sequence is, the more information will be kept in the sequence. Therefore, gene-based DNA synthesis can yield a high coding density as this study shows. The efficacy of the "EDS" system in storing medical data also serves as a foundation for future large-scale medical data storage applications. DNA may be stored in a small space and utilized to track illness across millennia. Furthermore, the DNA-stored diagnosis data may be retained alongside the genome DNA sample, providing additional information for genetic illness investigation in the following generation. Future optimization of this system may focus on testing these practical uses.

Nevertheless, gene synthesis is far more expensive than oligo synthesis. As a result, we are expecting that new technology will emerge to reduce the cost of gene synthesis. The recent development of enzymatic DNA synthesis technology could shed light on the future cost reduction of gene synthesis. Owing to a high catalytic efficiency of biological enzymes, the enzymatic DNA synthesis can potentially produce DNA fragment with longer length and higher fidelity compared to traditional chemical methods.[28,30] Moreover, the production process of enzymatic synthesis might be automated much easier and the overall reagent usage can be optimized to reduce the cost. Additionally, in order to enhancing the efficacy of the proposed EDS approach, future research could also concentrate on improving parallel gene fragment synthesis efficiency to achieve large-scaled data storage. Although the absence of error correction codes minimizes redundancy and data overhead, the EDS approach achieves an impressive high recovery rate in encode-decode processes. Incorporating error correction codes, such as RS codes, holds promise in enhancing accuracy. Challenges related to access speed in biomolecular data storage could be

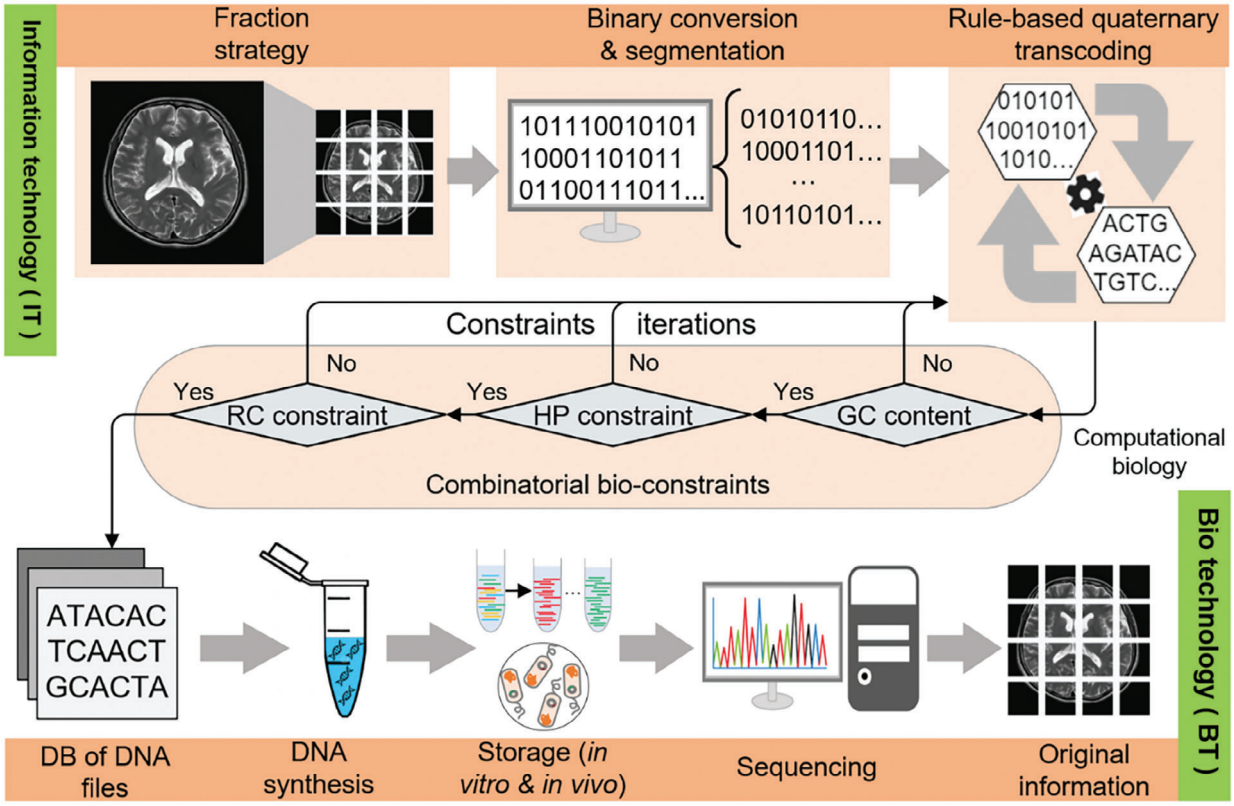

Figure 4. Proposed EDS approach for an effective DNA file storage system with IT and BT distribution.

mitigated through advancements in optimizing the whole DNA data storage pipeline, potentially offering commercial viability in the future.

4. Experimental Section

The proposed EDS approach is divided into two key phases: information technology (IT) and biotechnology (BT), as illustrated in **Figure 4**. The IT phase involves computer simulation to design the DFS using an encoding model, while the BT phase includes computational biology, DNA synthesis, and sequencing processes. These phases are further subdivided into four distinct technical segments, forming a comprehensive DNA data storage system. (*i*) Novel fraction strategy, (*ii*) Novel rule-based quaternary transcoding with an indexing system, (*iii*) Computational bio-coding constraints, and (*iv*) DNA synthesis, storage, and sequencing process.

This work's novelty lies in (*i*) and (*ii*), with detailed explanations provided in the subsequent subsections. The fulfillment of computational bio-constraints (*iii*) represents a major contribution computed by the prior research.[31] Therefore, the preliminaries of (*iii*) and (*iv*) are detailed in Sections S2 and S3 (Supporting Information), respectively. For a comparison with prior work, see Figure S5 (Supporting Information). Technical terms used in this study are defined in Table S8 (Supporting Information).

Novel Fraction Strategy: A novel strategy was introduced for storing medical images by splitting them into 16 sub-images or chunks to enhance throughput. This strategy addresses single-base error propagation and facilitates data analysis and organization within DNA, making it useful for various applications.

- Several state-of-the-art encoding models[2,17,18] have employed the rotating code technique (Figure S6, Supporting Information) to meet the biochemical constraints. However, the presence of a single nucleotide error could hinder accurate data decoding. Therefore, partitioning an image significantly mitigates such error propagation.
- This strategy enables retrieval of specific image chinks, saving time. Practitioners can identify patterns of interest, important for understanding genetic traits or diseases, from image chunks rather than accessing the entire image.
- Additionally, it reduces computational time and computing memory.

The splitting criteria were based on the specified number of rows (*split_h*) and columns (*split_w*), determined using the Python Imaging Library (PIL), alongside OS and math modules. An algorithm (**Table 3**) is proposed to determine the number of rows (*split_h*) and columns (*split_w*) parameters to split an image by identifying input file format and size.

To calculate the dimensions of the image segments and the remaining width ($w = img\_size$) and height ($h = img\_size$), nested for loops traverse the grid in row-major order. The minimum column height (C_h) (Equation 1) and row_width (R_w) (Equation 2) are computed as follows:

$$C_h = \left\lceil \frac{h}{split\_h} \right\rceil \tag{1}$$

$$R_w = \left\lceil \frac{w}{split\_w} \right\rceil \tag{2}$$

However, the last column_height C_{hl} and last row_width R_{hl} can be calculated as follows:

$$C_{hl} = h - (3 * C_h) \tag{3}$$

$$R_{wl} = h - (3 * R_w) \tag{4}$$

Table 3. Fraction algorithm to split image.

Input: Function cut with *split_h*, and *split_w* parameters, Image width ($w = img\_size$) and height ($h = img\_size$), min_height (C_h), min_width (R_w), last row width (R_{wl}) and last column height (C_{hl}).

Output: Split an image into 16 chunks.

1: Initialize count variable to 0

2: Get image_size (w and h)

3: for i in 0 to $split\_h - 1$:

4: for j in 0 to $split\_w - 1$:

5: if not last row and not last column:

6: crop image using $R_w$and C_h

7: else if last row and not last column:

8: image using $R_w$and C_{hl}

9: else if not last row and last column:

10: crop image using R_{wl} and C_h

11: else if last row and last column

12: image using R_{wl} and C_{hl}

13: end if, end for, end for

14: Calculate C_h and R_w using Equations 1 and 2, respectively.

15: Calculate C_{hl} (Equation 3) and R_{wl} (Equation 4) for edge segments.

Return: Generate smaller sub-images into 16 parts.

This mathematical foundation ensures an image is split into 16 equal chunks, even if the image dimensions are horizontal or vertical, making our algorithm different from others.

Meanwhile, why do we only split the image into 16 chunks? DNA has four bases, and if the power of the exponent is 2 for four bases (A, T, C, G), then 16 different subsets (base pair) can be generated with the least bases that enable control of the data overhead. The fraction algorithm is developed with four conditions within nested loops, iterating through a 2D grid to determine image cropping dimensions. The outer loop iterates through rows using an index, while the inner loop iterates through columns using an index. Depending on the position of the current grid cell, the fraction algorithm splits the input image using appropriate dimensions for segment cropping.

Proposed Rule-Based Transcoding: To effectively address combinatorial bio-coding constraints, mitigate error propagation associated with rotating coding,[2,17,18] and design a DFS, we introduced a novel transcoding mechanism. This novel transcoding approach is inspired by the Fountain codes,[13] which employs a mapping of {00, 01, 10, 11}↔{A, T, C, G}. The existing mapping is applied after the XORed function to evaluate the sequence's satisfaction criteria. However, sequences not meeting these constraints are directly discarded, leading to the loss of binary segments, reduced reliability of the original information, and significant effects on computational time due to the screening of each segment. The novel rule-based quaternary transcoding mechanism aids in mitigating data loss, enhancing reliability, and addressing computational challenges by adhering to bio-coding constraints and DFS through an indexing system.

The proposed mechanism reads binary segments through the encoder, categorizing tandem repeats into single-base and two-base types based on the number of base repeats in the DNA sequence. A binary sequence is divided into three fragments (i, j, and k). Encoding rules are devised based on these fragments, and subsequently, the proposed mechanism is applied. It is important to note that this transcoding is not exclusive to image files but extends comprehensive support to various file types, making it a significant contribution. The rules for proposed quaternary transcoding are outlined as follows:

1) In binary data strings, the initial two bits i were converted into their corresponding bases using Table d (derived from the Fountain codes). The encoder then employs Table d to map j bits until two bases overlap or repeat. If the last two mapped bases repeat, such as AA, TT, CC, or GG, Table D was utilized to obtain the next base(s) by mapping one or two additional bits. After Table D, the encoder progresses to Table g or G to map the next j bits as required.
2) Following initial mapping with j bits using Table d, the encoder maps each pair of j bits using Table g, the reverse of the Fountain codec scheme. This process continues until a repeated base pair is encountered. If a repeat base pair is identified, Table G is used to map the next one or two bits to generate the subsequent base. Furthermore, depending on the specific encoding requirements, the encoder shifts from Table G to d or D, and from Table D to g or G. This transition between tables continues until the last one or two bits k have been processed.
3) If the length of the binary string was predetermined, determining the mapping rule for the final one or two bits k may pose a challenge. To address this, the study proposes using Table E for processing terminal bit k. This application of Table E ensures balanced GC content remains and avoids single base duplication. When the last two converted bases from the j string repeat, Table E is applied for encoding. Conversely, if the last two converted bases j do not repeat, the encoder maps them using Table E's' default' rule and subsequently updates the final sequence. The final encoded DNA sequence comprises the bases from the i, j, and k binary fragments.

Section S6 (Supporting Information) provides a comprehensive exposition of the rule-based transcoding tables, along with two concrete examples (Figure S7, Supporting Information), flowchart (Figure S8, Supporting Information), decoding process, and Algorithm S1 (Supporting Information).The proposed EDS demonstrates efficient computational performance through its logarithmic and linear time complexities. By employing a method that halves the search space iteratively, it achieves a logarithmic time complexity, log(n), where n is the length of binary data, optimizing search processes. Coupled with a linear complexity from its indexing mechanism, the overall time complexity harmonizes to O(nlogn), highlighting an effective balance between constraints and partitioning strategies. This stands in contrast to the polynomial complexities of many other existing algorithms.[1,2,11,13] The EDS algorithm's nuanced time complexity ensures superior time efficiency and scalability, particularly in applications demanding high computational performance amidst varying input sizes and sparsity levels.

DNA File System (DFS): Additionally, an indexing method of the DFS is developed to prevent unintended amplification and facilitate flexible random searches, access, and file management, overcoming the high overhead challenges in DNA file storage (**Figure 5**).

This method simplifies the organization of stored data by utilizing the data tiering technique, which classifies data based on its usage into payload and non-payload categories. The DFS structure differs significantly from previous works[4,6,32] in various aspects, as outlined below.

1) The RT (regular tag, a base pair (bp)) efficiently helps in searching for a specific chunk (sub-image) by distinguishing it from others. The segmentation of the image into 16 chunks, as justified in Experimental Section, allows for the assigning of distinct base pairs as regular tags. Unique addresses are designed in the sequence index (4 bp) to precisely locate and retrieve the desired sequence.
2) Moreover, a DT (distinct tag, four bp) is added before a random payload (with an average length of 107 bp) to differentiate the decoded binary data for each chunk. Each DNA sequence is short enough to accommodate a single synthesizable DNA fragment.
3) Additionally, forward and reverse primers (20 nt each) are specifically designed to anneal to the target DNA sequences during PCR amplification.

Figure 5 demonstrates how a single chuck or multiple chunks can be easily searched and accessed from the DNA library using the proposed

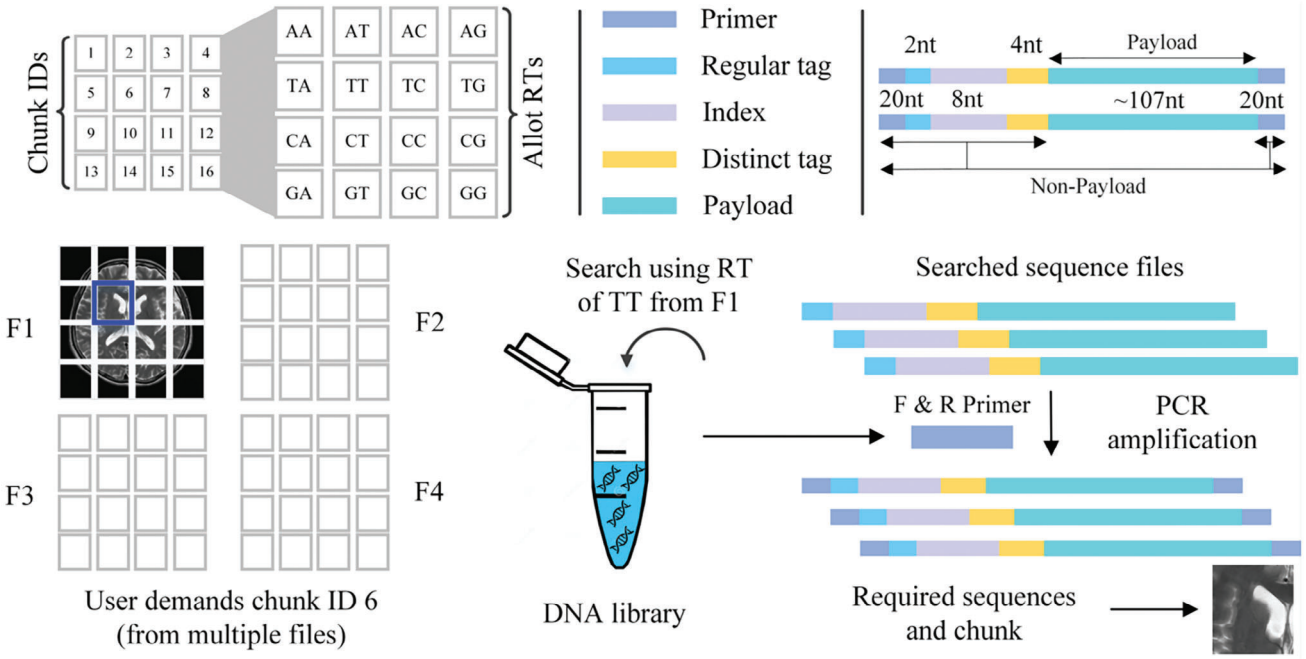

Figure 5. Information search and access through proposed indexing for DFS. For instance, one requires chunk ID 6 from file 1 (F1); DNA sequences can be searched from the DNA library using the corresponding RT of the required ID. Primers facilitate the amplification and access of the required sequence information of specific DNA fragments.

indexing system, and it should be noted that only brain MRI images are shown in the figure, and this operation is also supported for MRI images of other parts. This system ensures the overall significance of the DFS, preventing unintended amplification and enabling convenient extraction of desired information from larger gene or DNA databases (DB) with high throughput.

Computational Bio-Constraints: The satisfaction of combinatorial bio-coding constraints, crucial for reliable information retrieval in DNA data storage systems,[9] is ensured through the design of the proposed transcoding mechanism. Moreover, using GC content, HP, Hamming distance, and RC constraints, Theorem 1[31] is derived by considering the specific code length (n − 1) to produce optimal DNA codewords with Hamming distance (d) and GC content ($\mathcal{L}$).

Theorem 1. A DNA coding set of n length will be less than a codeword of $n-1$ length for a minimum Hamming distance of $(0 \le d \le n)$, while GC content $(0 < \mathcal{L} < n)$.

$$A_4^{GC}(n,\ d,\ \mathcal{L}) \le \frac{2n}{\omega} A_4^{GC}(n-1,\ d,\ \mathcal{L}-1) \tag{5}$$

$$A_4^{GC}(n, d, \mathcal{L}) \le \left\lfloor \frac{2n}{n-\mathcal{L}} A_4^{GC}(n-1, d, \mathcal{L}) \right\rfloor \tag{6}$$

Proof. In Equation (5), for a DNA sequence β_i with β_1 codeword and length n of GC content $\mathcal{L}$, there will be a position j in which $\lceil \mathcal{L}\alpha_1/2n \rceil$ DNA codes have the next DNA base C, or, in a given particular position, this code can be G. Consequently, the mean GC content will be lower than the original $\mathcal{L}$. Similarly, considering DNA codewords with the deletion of position j, it can produce $n-1$ and $\mathcal{L}-1$ DNA codewords with minimum distance. Thus, Equation (6) is analogous, but it differs from $\mathcal{L}$ for different positions where $\lceil (n-\mathcal{L})\alpha_1/2n \rceil$ represents A's or T's.

In this study, variations in Equations (5) and (6) are introduced to determine the upper bounds of DNA code on $A_4^{GC}(n,\ d,\ \mathcal{L})$ with $= d, n = \mathcal{L}$, or $\mathcal{L} = 0$ positions. Additionally, other coding bounds (i.e., lower bounds) can also be achieved by changing different variables; for example, if the code length n is constant ($n = d$), then Equation (5) can be considered with $n = \mathcal{L}$ and Equation (6) with $\mathcal{L} = 0$.

Moreover, Theorem 2 is computed with a specific term of $(d-1)$ Hamming distance with GC content to generate codes of DNA that must meet crucial bio-coding constraints. These theorems are elaborated in Section S2 (Supporting Information).

Supporting Information

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

Acknowledgements

This work was supported by the National Key Research and Development Program of China (under fund numbers 2021YFF1200100, 2021YFF1200104, and 2020YFA0909100) and Shenzhen Science and Technology Program (RCYX20221008092950122 and JCYJ20220818101407017) and Shenzhen Polytechnic Research Fund (No. 6023310010K).

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

A.R. and J.H. contributed equally to this work. A.R., Q.J., and X.H. conceived and designed the methodology, and reviewed the final draft. A.R. conducted the experiments and prepared the original manuscript. A.R., J.H., and Y.L. prepared figures for the manuscript. Q.J. and X.H. supervised the project. Q.J., X.H., Z.H., C.Z., Q.Q., Y.W., and J.D. analyzed and validated the experimental and theoretical results.

Data Availability Statement

The data that support the findings of this study are available in the supplementary material of this article.

Keywords

bio-coding constraints, computational biology, DNA data storage system, DNA synthesis, rule-based transcoding model

Received: November 15, 2023
Revised: March 29, 2024
Published online: May 29, 2024

[1] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, W. J. Stark, *Angew. Chem. Int. Ed. Engl.* **2015**, *54*, 2552.
[2] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, E. Birney, *Nature* **2013**, *494*, 77.
[3] P. L. Antkowiak, J. Lietard, M. Z. Darestani, M. M. Somoza, W. J. Stark, R. Heckel, R. N. Grass, *Nat. Commun.* **2020**, *11*, 5345.
[4] L. Song, F. Geng, Z.-Y. Gong, X. Chen, J. Tang, C. Gong, L. Zhou, R. Xia, M.-Z. Han, J.-Y. Xu, B.-Z. Li, Y.-J. Yuan, *Nat. Commun.* **2022**, *13*, 5361.
[5] M. Li, J. Wu, J. Dai, Q. Jiang, Q. Qu, X. Huang, Y. Wang, *Sci. Rep.* **2021**, *11*, 18063.
[6] M. Dimopoulou, M. Antonini, P. Barbry, R. Appuswamy, *Signal Process.* **2021**, *97*, 116331.
[7] X. Y. Li, M. X. Chen, H. M. Wu, *Brief. Bioinform.* **2023**, *24*.
[8] A. Rasool, Q. Qu, Q. Jiang, Y. Wang, Springer, ICA3PP 2021, **2022**, 284.
[9] A. Rasool, Q. Jiang, Y. Wang, X. Huang, Q. Qu, J. Dai, *Front. Genet.* **2023**, *14*, 1158337.
[10] J. Jeong, S.-J. Park, J.-W. Kim, J.-S. No, H. H. Jeon, J. W. Lee, A. No, S. Kim, H. Park, *Bioinformatics* **2021**, *37*, 3136.
[11] 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.
[12] B. Cao, X. Zhang, S. Cui, Q. Zhang, *npj Syst. Biol. Appl.* **2022**, *8*, 23.
[13] Y. Erlich, D. Zielinski, *Science* **2017**, *355*, 950.
[14] G. Qu, Z. Yan, H. Wu, *Brief Bioinform.* **2022**, *23*, 336.
[15] G. M. Church, Y. Gao, S. Kosuri, *Science* **2012**, *337*, 1628.
[16] M. Blawat, K. Gaedke, I. Hütter, X.-M. Chen, B. Turczyk, S. Inverso, B. W. Pruitt, G. M. Church, *Procedia Comput. Sci.* **2016**, *80*, 1011.
[17] 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.
[18] B. Li, L. Ou, D. Du, *ACM, SYSTOR* **2021**, 1.
[19] L. Organick, Y. J. Chen, S. D. Ang, R. Lopez, X. M. Liu, K. Strauss, L. Ceze, *Nat. Commun.* **2020**, *11*, 616.
[20] Q. Xu, M. R. Schlabach, G. J. Hannon, S. J. Elledge, *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 2289.
[21] T. Bonduelle, M. Ollivier, A. Gradel, J. Aupy, *Rev. Neurol.* **2024**, *12*, 11.
[22] J. Vosshenrich, G. Koerzdoerfer, J. Fritz, *Skeletal Radiol.* **2024**, *53*, 1.
[23] A. Rasool, J. Hong, Q. Jiang, H. Chen, Q. Qu, *Comput. Biol. Med.* **2023**, *165*, 107404.
[24] I. Maarouf, A. Lenz, L. Welter, A. Wachter-Zeh, E. Rosnes, A. G. I. Amat, *IEEE Trans Inf. Theory* **2023**, *69*, 910.
[25] R. Hanhan, E. Garzon, Z. Jahshan, A. Teman, M. Lanuzza, L. Yavits, *IEEE/ACM, ISCA*, **2022**, 495.
[26] S. M. H. T. Yazdi, R. Gabrys, O. Milenkovic, *Sci. Rep.* **2017**, *7*, 5011.
[27] M. Welzel, P. M. Schwarz, H. F. Löchel, T. Kabdullayeva, S. Clemens, A. Becker, B. Freisleben, D. Heider, *Nat. Commun.* **2023**, *14*, 628.
[28] A. Hoose, R. Vellacott, M. Storch, P. S. Freemont, M. G. Ryadnov, *Nat. Rev. Chem.* **2023**, *7*, 144.
[29] S. Kosuri, G. M. Church, *Nat. Methods* **2014**, *11*, 499.
[30] E. Yoo, D. Choe, J. Shin, S. Cho, B.-K. Cho, *Comput. Struct. Biotechnol. J.* **2021**, *19*, 2468.
[31] A. Rasool, Q. Qu, Y. Wang, Q. S. Jiang, *Mathematics* **2022**, *10*, 845.
[32] L. C. Meiser, P. L. Antkowiak, J. Koch, W. D. D. Chen, A. X. Kohll, W. J. Stark, R. Heckel, R. N. Grass, *Nat. Protoc.* **2020**, *15*, 86.

