nature computational science

Article

https://doi.org/10.1038/s43588-025-00793-x

DNA data storage for biomedical images using HELIX

Received: 6 June 2024

Accepted: 18 March 2025

Published online: 13 May 2025

Check for updates

Guanjin Qu[1], Zihui Yan[2,3], Xin Chen[1,3] & Huaming Wu[1,3]

Deoxyribonucleic acid (DNA) data storage is expected to become a key medium for large-scale data. Biomedical data images typically require substantial storage space over extended periods, making them ideal candidates for DNA data storage. However, existing DNA data storage models are primarily designed for generic files and lack a comprehensive retrieval system for biomedical images. Here, to address this, we propose HELIX, a DNA-based storage system for biomedical images. HELIX introduces an image-compression algorithm tailored to the characteristics of biomedical images, achieving high compression rates and robust error tolerance. In addition, HELIX incorporates an error-correcting encoding algorithm that eliminates the need for indexing, enhancing storage density and decoding speed. We utilize a deep learning-based image repair algorithm for the predictive restoration of partially missing image blocks. In our in vitro experiments, we successfully stored two spatiotemporal genomics images. This sequencing process achieved 97.20% image quality at a depth of 7× coverage.

DNA data storage is an emerging method that utilizes DNA molecules to store digital information. This technique offers extremely high storage density, with the potential to store up to 455 EB of information in just 1 g of DNA[1]. In addition, DNA molecules have a remarkably long storage lifespan and require no power for preservation. Research has demonstrated that DNA molecules can recover information even after 10,000 years of storage at room temperature[2]. These unique advantages make DNA data storage a promising candidate for the next-generation of storage medium, particularly for large-scale data management[3–5].

Owing to the involvement of biochemical reaction processes, DNA data storage has a relatively slower write–read bandwidth. As a result, it is better suited to serving as large-scale cold data storage intended for less frequent usage. One potential application domain could be biomedical data images, encompassing genomics pictures, medical images and similar data types. These images typically exhibit high resolution, have long-term storage requirements and are accessed infrequently. However, current DNA data storage models are mainly focused on error-correction codes[6–10] and bioinformatic algorithms[11–13], which present certain drawbacks when applied to the storage of biomedical data images. Traditional computer-based image-compression algorithms are not suitable for DNA data storage. For high-resolution biomedical data images, the use of lossless image compression requires a large amount of storage, which poses a challenge given the current high cost of DNA data storage. In addition, traditional image-compression algorithms cannot correct synchronization errors that may occur in DNA data storage, which can result in minor errors that render the entire image unrecoverable.

Although some image-compression algorithms for DNA data storage have been developed[10,14–16], they typically rely on error-correcting codes that can fail, leading to the loss of the entire image if not all errors are corrected. To address the unique characteristics of biomedical data images and the current limitations of DNA data storage, it is essential to develop a comprehensive system that integrates image compression and error correction.

In this paper, we introduce a DNA data storage system tailored for biomedical data images, called HELIX. The system comprises three key components: image compression, error correction and image restoration. For the large size of biomedical images, image-compression modules enable high compression rates while ensuring that most of

[1]Center for Applied Mathematics, Tianjin University, Tianjin, P. R. China. [2]School of Chemical Engineering and Technology, Tianjin University, Tianjin, P. R. China. [3]State Key Laboratory of Synthetic Biology, Tianjin University, Tianjin, P. R. China. e-mail: whming@tju.edu.cn

Table 1 | Comparison between HELIX and existing image-compression algorithms for DNA data storage

Reference	Multi-image storage	Image compression	Content-aware optimization	Fault tolerance	Error correction	Image repair
DNA-QLC[14]	✗	✓	✗	✗	✓	✗
HL-DNA[34]	✗	✓	✓	✓	✗	✗
Franzese et al.[35]	✓	✓	✗	✓	✗	✗
Wu et al.[36]	✗	✓	✗	✓	✗	✗
Rasool et al.[15]	✓	✓	✓	✗	✗	✗
Dimopoulou et al.[37]	✗	✓	✗	✓	✗	✗
Pan et al.[10]	✓	✓	✗	✗	Partial	✓
Bhaya et al.[38]	✗	✓	✓	✗	✗	✗
HELIX	✓	✓	✓	✓	✓	✓

the image content can be recovered even in the presence of a small number of errors. Table 1 illustrates the differences between HELIX and current image-compression algorithms for DNA data storage. By integrating the error-correction module with the image-compression module, we enable direct access to image information within the sequence during error-correction decoding, eliminating the need for indexing during the error-correction coding process. This approach substantially enhances information storage density. Meanwhile, our error-correction coding shows an exceptionally high decoding speed, capable of processing approximately 200,000 sequences per second. In addition, we introduce a deep learning-based image-restoration scheme that performs specific image restoration to improve the quality of image restoration when the decoded image contains error blocks. In a biomedical experiment involving the storage of two images totaling 60 MB, the results show that most of the image information can be recovered even at a sequencing depth of 7×, validating the reliability of the model.

Results

General principle and features of HELIX

The modeling framework of HELIX, illustrated in Fig. 1, comprises three key modules: image compression, error correction and image restoration. The image-compression module is responsible for compressing the image and segmenting it into sequences of information. The error-correction module handles DNA synthesis and sequencing errors. The image-restoration module is responsible for restoring any error blocks that may exist after the image decoding process. Users have the flexibility to decide whether to utilize the image-restoration module.

We designed an image-compression module specifically for DNA storage and biomedical data images to compress and encode images into base sequences. Compared with traditional image-compression algorithms, we consider the possibility of errors during DNA storage and use a chunking approach to ensure that no error will cause a chain reaction. In addition, we introduce a base mapping mechanism to reduce the homopolymer length of the base sequences. To improve the encoding rate of uniform background biomedical data images, we used a mechanism that does not record consecutive repetitive chunks of information. This mechanism is shown in Fig. 1d. Only the first block in each row is recorded in regions with the same color, thus greatly improving the image-compression efficiency. During decoding, if a block index is missing, it is assumed to be the same as the previous block. After the image is encoded, the sequence will be encoded for error correction so that the error can be corrected during the decoding process. HELIX employs a cascading coding scheme: additional sequences are first encoded using longitudinal outer coding, followed by inner coding for each sequence. During decoding, the inner code is decoded first to correct within-sequence errors, and then the longitudinal outer code is decoded to resolve sequence loss. HELIX performs error-correction coding without adding indexes to the sequence because the error-correction module can recognize the header information of the image-compression module during decoding, which improves the code rate.

For potentially corrupted blocks in the decoded image, we introduce a deep learning-based image-restoration algorithm. This algorithm can predict and repair erroneous blocks. Owing to the strict content requirements of some medical images, we provide users the option to enable or disable this feature. If the image-restoration algorithm is activated, a 1 B cyclic redundancy check (CRC) check bit is appended to the end of each sequence. Each sequence is then decoded and the check digit is verified. Blocks that fail the check are flagged, identifying the locations of the erroneous image blocks. Figure 1c illustrates the network structure used for the image-restoration model, which employs a loss-function mechanism combined with generative adversarial networks[17]. Unlike traditional image-restoration algorithms, our method addresses the high resolution of biomedical data images by focusing on local restoration. Instead of processing the entire image, the algorithm predicts the information of the damaged block by analyzing the context around the erroneous area.

HELIX ensures that stored images are robust and maintain high bit rates by utilizing the three modules described above. In addition, homopolymer constraints can reduce errors generated during synthesis[18–20]. We achieve this by controlling the block size of the error-correction code, ensuring that the homopolymer length is typically less than five. Next, we verify the effectiveness and advantages of HELIX through both simulations and in vitro storage experiments.

Effectiveness evaluation of HELIX in simulation experiments

We selected three biomedical datasets for our simulation experiments: spatiotemporal histology slices[21], human knee X-ray images[22] and human lung computed tomography images[23]. From these datasets, we randomly chose a subset of images for our experiments. To verify the effectiveness of our image-compression algorithm, we compared it against several common image-compression schemes, including JPEG, BMP and GIF.

Figure 2 shows the impact of different image formats and the number of bases required when a small number of errors occur. The images suffer from the same 0.1% probability of deletion, substitution and insertion errors. To evaluate the effectiveness of HELIX's image-compression algorithm, this experiment did not include the error-correction function. BMP and HELIX show the best tolerance for errors, retaining most of the image information despite errors. GIF, while having a high code rate for binary images, fails to recover the image information due to damage to key parts. JPEG suffers from color errors caused by its differential coding method. BMP error regions are more finely grained compared with HELIX because HELIX chunks the image. However, the number of bases required for BMP is substantially higher, increasing the cost of DNA digital storage. The number of bases encoded by JPEG and HELIX are similar and relatively low. Although

https://doi.org/10.1038/s43588-025-00793-x

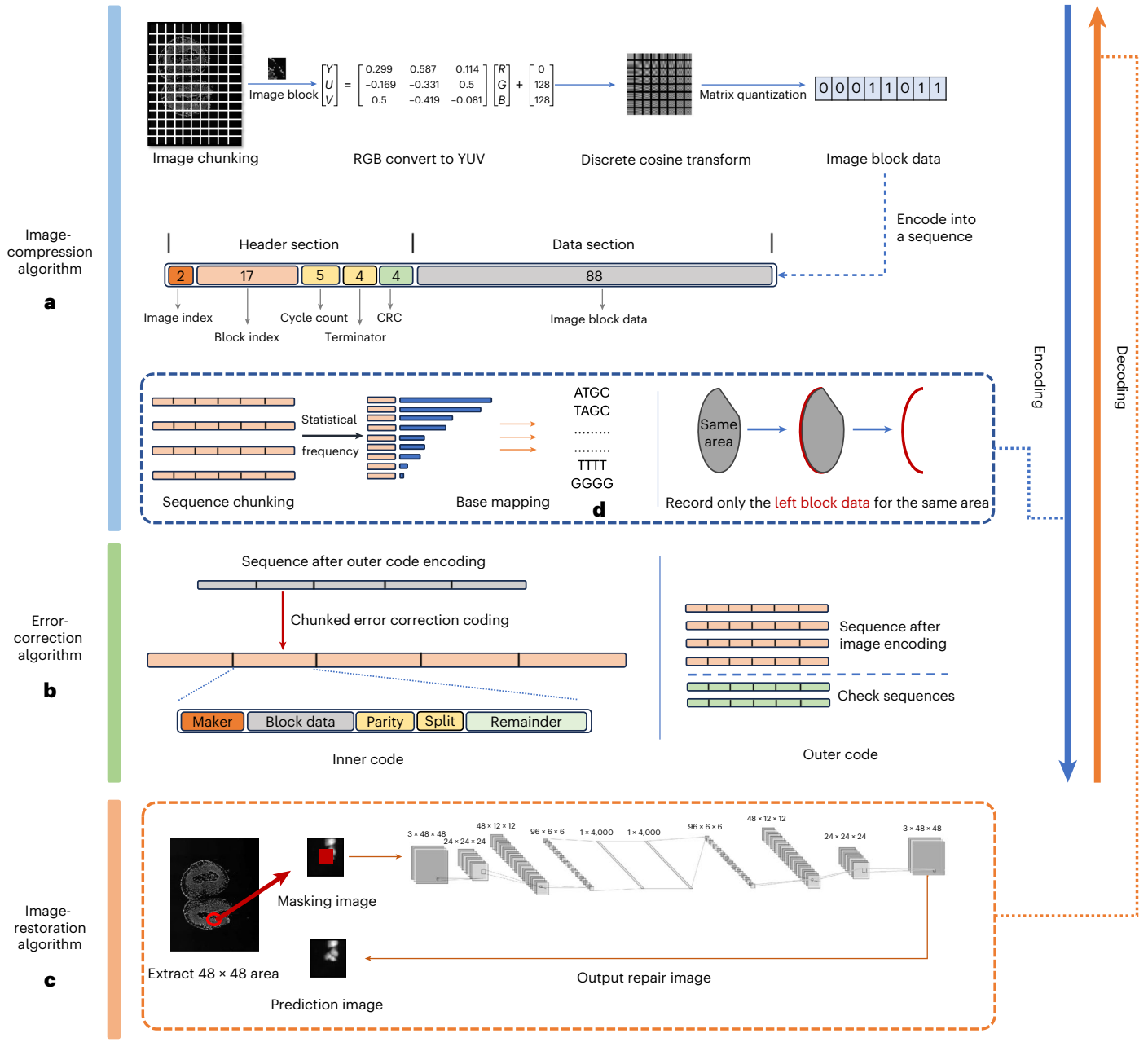

Fig. 1 | The overall modeling framework of HELIX. a, Image-compression algorithm. First, the image is divided into 16 × 16 image blocks and compressed. The compressed image information will be converted into sequences and header information will be added to each sequence. Here RGB represents the color space composed of red, green and blue, and YUV represents the color space based on luminance and chrominance. **b**, Error-correction algorithm. This algorithm is a cascade code that combines inner code and outer code. The inner code combines Levenshtein and Marker (LM) coding, and the outer code uses Reed–Solomon (RS) coding. **c**, Image-restoration algorithm. In case of image block corruption, a small surrounding area of the image is selected for prediction. The prediction results are then used to repair the corrupted block. **d**, Special mechanisms of the image-compression algorithm. On the left side, the base mapping mechanism is illustrated. Here, the image-coded sequence is divided into multiple segments. The algorithm counts the frequency of each segment and maps those with higher occurrences to base combinations with superior biochemical properties. On the right side, a unique de-duplication mechanism is depicted. Only the first block of each repeating sequence is recorded when the image-encoding process encounters consecutive repeating blocks of information.

JPEG has a higher compression ratio, it is more prone to overall errors when minor errors occur in the image. In conclusion, HELIX shows high error tolerance and effective image compression, making it well suited for DNA data storage.

We used structural similarity index measure (SSIM), peak signal-to-noise ratio (PSNR), mean squared error (MSE), feature similarity index (FSIM), multi-scale structural similarity index measure (MS-SSIM) and universal quality index (UQI) as metrics to evaluate the quality of image restoration. Figure 3a illustrates the changes in these metrics at different error rates. The horizontal axis represents the error rate, increasing by 0.3% at each step (with equal probabilities for insertion, deletion and substitution errors), and the number of sequence copies is 10. In the initial stage, the metrics remain relatively stable, indicating that the error-correcting code efficiently corrects all errors during this period. However, once the error rate exceeds the upper limit of the error-correction capability, the metrics begin to fluctuate. The slower fluctuation of the metrics is attributed to the robustness of the HELIX image-compression algorithm, which mitigates the overall

https://doi.org/10.1038/s43588-025-00793-x

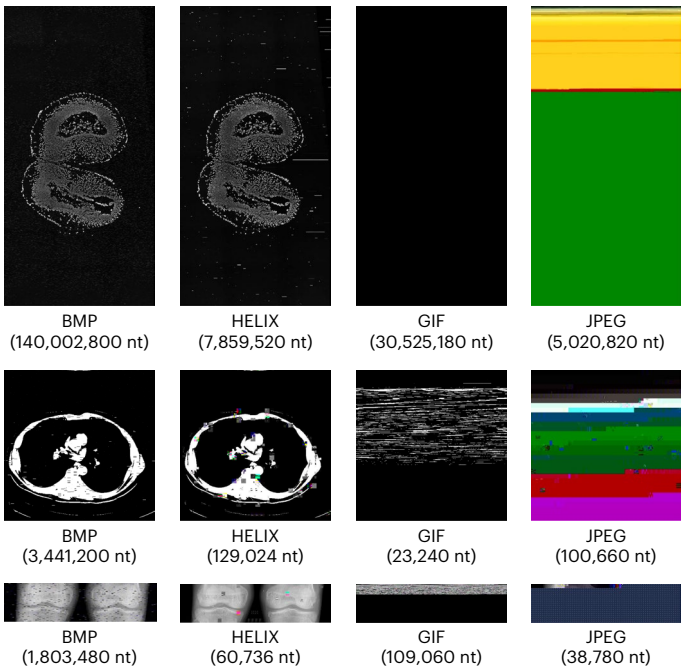

Fig. 2 | Fault-tolerance performance of different image-compression algorithms. The performance of different image-compression algorithms when 0.1% error occurs. The images used for the assessment include spatiotemporal histology slices[21] (top row), human lung computed tomography images[23] (middle row) and human knee X-ray images[22] (bottom row). It can be seen that the HELIX and BMP formats can still show a lot of the original information. In addition, the number of coding bases needed for HELIX and JPEG is much smaller than that for the other formats.

damage to the image. In addition, HELIX performs better for spatiotemporal histology (STH) images, which is mainly due to the high resolution of STH images. HELIX deterministically segments images into fixed 16 × 16 pixel blocks. Thus, for high-resolution images, block damage has a smaller effect on the overall image quality.

To evaluate the effectiveness of the HELIX image-compression algorithm, we selected several representative image-compression algorithms as benchmark algorithms[10,14,15], where HELIX without the error-correction module (HELIX_WEC) shows the effect of the image-compression algorithm. Figure 3c shows the number of bases required to store an X-ray image. It can be seen that QLC[14] and HELIX have the least bases and show a very high compression rate. EDS[15] requires the most number of bases as it is lossless compression. Figure 3d shows the image recovery under different error rates. EDS and 2DDNA[10] can not recover the image at 0.1% error rate because they do not have error resilience, and QLC can not recover the image at 0.3% error rate even though it has an error-correction algorithm. HELIX, which uses only the image-compression algorithm, not only has the highest SSIM value under lossy compression but also has an SSIM value that decreases slowly as the error grows. The above experiments prove that the image-compression algorithm of HELIX has the characteristics of high compression and high error tolerance. We also compare the error-correction module of HELIX with other existing DNA storage error-correction algorithms in Supplementary Table 3.

Figure 3b shows the changes in image metrics after restoration, with the horizontal axis representing the error rate and the vertical axis representing the image-quality metrics. It can be seen that the image-restoration algorithm improves image quality, although the improvement is modest as erroneous blocks are relatively few. Nonetheless, the algorithm effectively enhances the overall appearance of the image. In addition, Fig. 3f shows the actual results of the image-restoration algorithm at an error rate of 6% error rate. It can be seen that the image-restoration algorithm substantially enhances the readability of the image and reduces the number of consecutive erroneous image blocks. In addition, zooming in shows that the image-restoration algorithm can substantially reduce some color errors.

To evaluate the effectiveness of HELIX in optimizing biomedical images, we encoded various types of dataset. Figure 3e compares the optimization rate of HELIX for biomedical images against other common datasets. The optimization rate is defined as the ratio of information omitted by the de-duplication mechanism to the total information, with a higher ratio indicating better compression. The results show that HELIX achieves a substantially better compression effect on biomedical images compared with common datasets. This indicates that HELIX is effectively optimized for coding based on the specific characteristics of biomedical images.

Experimental validation of HELIX in vitro storage

To further validate the effectiveness of HELIX, we conducted in vitro experiments by storing 2 images, totaling 60 MB, of spatiotemporal genomics slices. After encoding, approximately 140,000 DNA sequences were generated, each with a length of 183 nucleotides (nt). Figure 4 shows the coding structure of the sequences. The net information density (the number of information bits/the encoded nucleotides) is 26.22 bits per nt, and the code rate of the error-correction code is 0.7918 bits per nt. The detailed calculation process is provided in Supplementary Section 1.

Through PCR and sequencing technology, we obtained the sequenced DNA sequences. Details of the biomedical treatment are provided in Methods. For the post-sequencing sequences, we tried two different ways of processing the data. First, we sorted the sequences by their frequency of occurrence, selecting them from highest to lowest frequency. Figure 4 illustrates the decoding results after selection. As the number of sequences increases, the image quality improves rapidly, with the best restoration achieved at around 800,000 sequences. This balance is crucial: too few sequences result in substantial data loss, while too many introduce numerous erroneous sequences. In addition, we attempted random sequence selection. Using this method, the SSIM value reached 97.20% when reading 1 million sequences. At this point, the decoded image was nearly identical to the original. This experiment confirms the reliability of HELIX for practical DNA data storage applications.

Discussion

HELIX is a DNA-based data storage solution specifically designed for biomedical images, offering a broad range of potential applications. In the domain of long-term archiving and back-up, HELIX empowers medical data centers and bioinformatics fields to store vast quantities of biomedical images in a cost-effective and stable manner. This ensures reliable, long-term data preservation, providing an invaluable resource for advancing medical research. Compared with existing DNA data storage solutions[10,14,15], HELIX incorporates a fault-tolerant mechanism that addresses potential decoding failures in long-term storage. This ensures that partial image recovery is possible even when errors occur, mitigating the risk of complete data loss during prolonged storage. Moreover, HELIX tackles the critical need for rapid data access in biomedical imaging. With an impressive decoding speed of 100,000 entries per second, it greatly enhances read bandwidth, overcoming a key limitation of existing DNA storage technologies.

To enhance the readability of images after errors, we have introduced an image-restoration algorithm that predicts missing information to some extent. However, for rigorous biomedical images, the content predicted by the restoration algorithm may not always be reliable, posing a challenge in preserving the parameters of the neural network over time. In addition, HELIX employs lossy compression for image coding, which results in some loss of stored image details.

https://doi.org/10.1038/s43588-025-00793-x

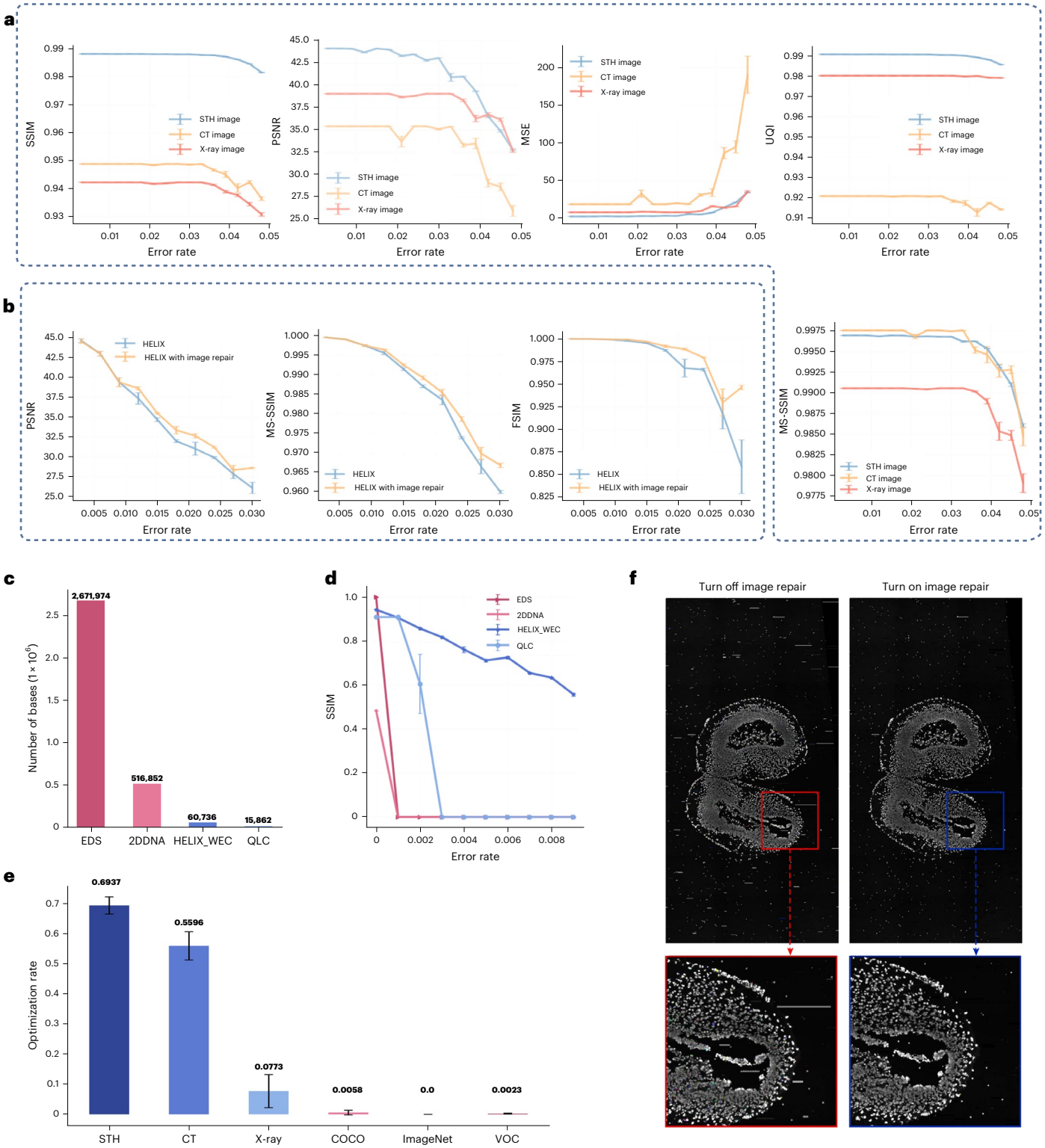

Fig. 3 | Evaluation of the effectiveness of the HELIX simulation experiment. We repeated the experiment three times with different random seeds. For **a** and **b**, we selected different random seeds and repeated the experiment three times. Data are presented as mean values ± s.e.m. **a**, Plot of changes in image recovery metrics for HELIX with increasing error rates. **b**, Impact of image-restoration algorithms on image recovery with increasing error rate. CT, computed tomography. **c**, Comparison of the amount of bases in HELIX with other image-compression algorithms when storing the same image. **d**, Image recovery of HELIX with other image-compression algorithms as the error rate varies. **e**, Optimization rate of HELIX image-compression algorithm with different datasets. We used three widely recognized generic datasets: ImageNet, Common Objects in Context (COCO) and Visual Object Classes (VOC). The experimental data are based on the compression of three randomly selected images. **f**, Effectiveness of HELIX image-restoration algorithm on partially corrupted image blocks.

https://doi.org/10.1038/s43588-025-00793-x

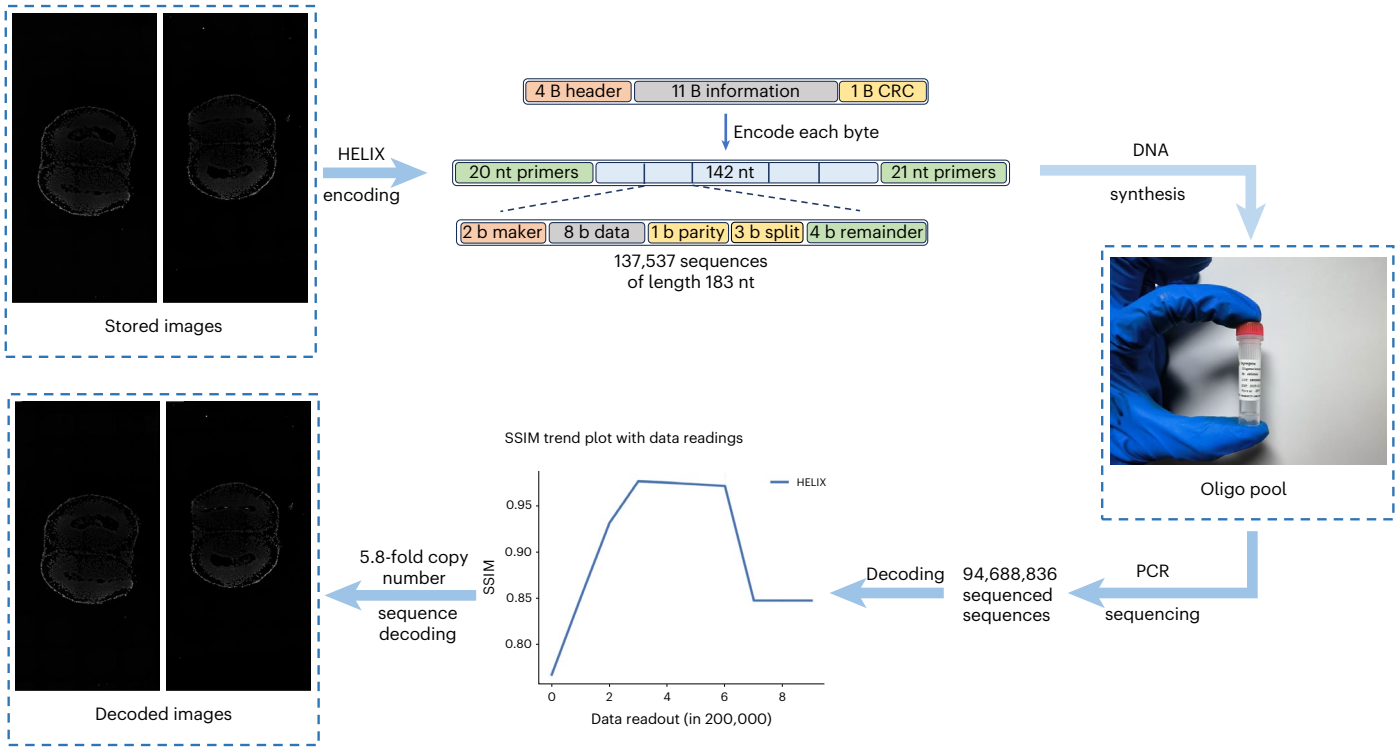

Fig. 4 | Flowchart of in vitro experiments. The storage content is 2 spatiotemporal histological images totaling 60 MB. HELIX encoding divides it into 125,000 chunks, each of which contains an 11 B information section, a 4 B index section and a 1 B checksum section. After that, each byte of each block was sequentially encoded for error correction, and after encoding, a total of 137,537 sequences were generated with a length of 183 nt. We synthesized and sequenced a total of 8,468,836 sequences that were read. The sequences were sorted by frequency of occurrence after data cleaning and decoding were attempted for different read volumes. In the end, the image we attempted to decode after reading only 800,000 sequences (at a sequencing depth of 5.8×) recovered most of the information from the original image.

While these details are typically imperceptible to the human eye, the information loss caused by lossy compression could prevent HELIX from storing highly detailed or complex images. Given the current high cost of DNA storage technology, adopting a high-bit-rate DNA storage scheme is crucial for the practical application of DNA data storage. Therefore, the combination of lossy compression and image restoration remains a promising approach for efficiently storing image data in DNA, despite these challenges.

The HELIX encoding did not exclude potentially risky DNA sequences, although constraints such as homopolymers were taken into account. Future studies will focus on avoiding specific DNA fragments to minimize biological risks associated with synthetic DNA[24,25]. HELIX is a DNA storage solution dedicated to biomedical images, which may be difficult to apply for general-purpose data. In follow-up studies, we will explore the use of HELIX to store images with similar background purity such as microscope images and satellite images, which will further enhance the versatility of HELIX. In addition, HELIX verifies that a dedicated coding scheme for a certain type of data can have better results than a general-purpose storage scheme, in terms of both code rate and robustness. Designing specialized storage solutions for large-scale storage needs, such as point cloud data and video data, is expected to be a key direction in advancing DNA data storage technology.

Methods

Image-compression module

The image-compression module compresses and encodes the original image, generating a set of quadratic sequences that facilitate the inclusion of subsequent error-correction codes. Furthermore, it employs a chunked coding method to safeguard against the impact of indel errors (insertions and deletions). When such an error occurs, its influence is confined to the specific block and does not affect the entire image. The detailed encoding process is outlined in the following section.

Image compression. For the original image, we first split it into 16 × 16 pixel blocks and converted each block into the YUV color space. This step leverages the human eye's greater sensitivity to luminance and relative insensitivity to chromaticity. Next, we apply a discrete cosine transform (to the Y, U and V components of each block, followed by quantizing the discrete cosine transform coefficients[26] to reduce data size. Unlike the JPEG compression algorithm[27], we do not perform Huffman or arithmetic coding, as these operations would decrease the image's fault tolerance.

Data segmentation and header integration. After the image compression, the information of the image block will be divided into sequences of specified length, and the header will be added to the sequences. The information of an image block can be represented by one or more sequences, and each sequence contains information of only one image block. This method can effectively reduce the transmission of errors. Specifically, each sequence consists of an index part and an information part. The information part records the data of the image block and its length is determined by the synthesis technique. The index part is usually 3 B long and is used to identify the position of the sequence and track changes in the image. It consists of five components in order, namely, the image number index, the image block index, the cycle count, the terminator and the CRC. The role of each component is explained in detail below.

- Image number index: this index serves to record the image number when storing multiple images. Given the current high cost of DNA data storage, approximately $3,000 or more for synthesizing 1 MB (ref. 28), we currently allocate 2 b for this index area by

default. As the cost of DNA synthesis decreases, there is potential to expand the image numbering index area further.
- Image block index: this index is designated for numbering the image blocks within an image. For images segmented into 16 × 16 pixel blocks, the sorting is conducted from top to bottom and left to right. The assigned location for recording this information is currently configured with 17 b, providing ample support for images containing up to 33,554,432 pixels.
- Cyclic number: the order of sequences will be completely disrupted after DNA sequencing, and some sequences may be lost. To address this, we incorporate an innovative cycle count mechanism into the index part. The cyclic number increments with each sequence as the image is encoded and resets to zero after reaching a specified threshold. Through the cycle count mechanism, on the one hand, multiple sequences within the same image block can be sorted and, on the other hand, it can facilitate the design of subsequent error-correction codes. Currently, the cycle counting area is set to 5 b, ensuring proper functionality as long as no more than 32 consecutive sequences are lost.
- CRC: CRC is a widely used error-checking code designed to verify the occurrence of errors[29]. Recognizing the critical impact of errors in the index part, CRC is incorporated into this section for validation. Sequences failing the checksum after sequencing and decoding are promptly discarded. Currently, this region is configured with a 4 b setting.

Base mapping. For the sequence after image encoding, we use four bases as a unit block and calculate the frequency of occurrence of each unit block. The unit block with high frequency is mapped to a base unit block of the same length with good biochemical properties (for example, ATGC), and the unit block with low frequency is mapped to a base unit block of the same length with poor biochemical properties (for example, AAAA). This mapping is bijective and minimizes homopolymer formation and improves GC equilibrium. We give a more detailed description in Supplementary Fig. 10, and characterize the coding sequences from statistical in vitro experiments to exemplify the effectiveness of HELIX for biochemical constraints.

Error-correction module

Outer-code coding. As the outer code deals solely with erasure errors, such as the loss of column information due to missing sequences, we adopt the well-established Reed–Solomon (RS) coding scheme[30]. RS coding generates redundancy check codes by treating the data to be transmitted as the coefficients of a polynomial, enabling error-correcting decoding. As the index part of the sequence does not have a continuous index area, each RS block is divided based on the number of cycles. By default, every 7-cycle unit constitutes one RS-coded information block, which includes redundancy, totaling 255 entries. During decoding, the sequence is first sorted based on the image index and block index, then RS blocks are divided according to the number of cycles and, finally, each RS block is decoded.

Inner-code coding. Our inner code uses a combination of Levenshtein and Marker coding (LM coding), known for its high code rate and error-correction capabilities[31]. The encoding process for each code word segment is divided into five parts: a marker bit, an information bit, a check bit, a separator bit and a syndrome bit. The marker bit is responsible for synchronization within the sequence. The information bit contains the original information. The check bit is responsible for checking whether an error has occurred or not. The split bit is responsible for splitting the check bit with the remainder bit. The remainder bits are responsible for error correction of the information bits. As the length of the information site is four bases and is usually different from the front and back positions, this mechanism ensures that the homopolymer length of the encoded sequence is usually less than five. Unlike the original LM encoding method, we omit an extra index in each sequence. Instead, we use the image index and image block index from the index region for sorting, conserving space. In addition, by chunking the information sequence, we effectively reduce the homopolymer length, ensuring that it remains under five in the strictest scenarios.

In vitro experiment

Figure 4 illustrates the entire process of DNA data storage in vitro. The selection of DNA oligo pools is from Dynegene Technologies. The three pairs of primers, namely, 0F/0R, 1F/1R and 2FU/2R (1-4), were all synthesized by Dynegene Company.

The process began with centrifuging the dry DNA oligo pools at 4 °C and 14,000 rpm. The oligos were quantified using a Nanodrop single-stranded DNA measurement, then diluted to 5 ng μl^{-1}. For the initial amplification, a PCR mix was prepared with the following conditions: 10 µl PCR mix, 0.5 µM of 0F/0R primers, 0.25 ng μl^{-1} DNA oligo pools, in a total volume of 20 µl, run for 7 cycles. After amplification, the PCR products were purified using 2.8× QuarAcces Hyper Pure Beads and quantified with Qubit double-stranded DNA. The purified PCR product was diluted to 0.1 ng μl^{-1} for a second PCR amplification, with the conditions: 10 µl PCR mix, 0.5 µM of 1F/1R primers, 0.005 ng μl^{-1} PCR product, in a total volume of 20 µl, run for 10 cycles. Following this, the PCR products were again purified using 2.8× QuarAcces Hyper Pure Beads and quantified with Qubit double-stranded DNA.

The PCR products from the second amplification were diluted to a concentration of 1 ng μl^{-1} for the third PCR step. The reaction conditions were as follows: 10 µl PCR mix, 0.5 µM of 2F/2R primers, 0.05 ng μl^{-1} of second PCR products, in a total system volume of 20 µl, with 6 cycles performed. After this final PCR step, the products were purified and quantified using Qubit double-stranded DNA. Fragment analysis was carried out using an Agilent 2100 Bioanalyzer following three cycles of PCR. The resulting fragments were then subjected to next-generation sequencing and the quality of the sequencing data was assessed.

Reporting summary

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

Data availability

Sequencing data from in vitro experiments can be found at https://doi.org/10.6084/m9.figshare.25957606.v1 (ref. 32). The images used in the HELIX test are all from publicly available datasets. The spatiotemporalomics image dataset is from ref. 21. The computed tomography image dataset is from https://doi.org/10.7937/K9/TCIA.2016.JGNIHEP5 (ref. 23). The X-ray image dataset is from https://doi.org/10.17632/t9ndx37v5h.1 (ref. 22).

Code availability

HELIX includes both Go and Python versions. The code can be found at https://doi.org/10.5281/zenodo.14699789 (ref. 33).

References

1. Church, G. M., Gao, Y. & Kosuri, S. Next-generation digital information storage in DNA. *Science* **337**, 1628 (2012).
2. Song, L. et al. Robust data storage in DNA by de Bruijn graph-based de novo strand assembly. *Nat. Commun.* **13**, 5361 (2022).
3. Ceze, L., Nivala, J. & Strauss, K. Molecular digital data storage using DNA. *Nat. Rev. Genet.* **20**, 456–466 (2019).
4. Dong, Y. et al. DNA storage: research landscape and future prospects. *Natl Sci. Rev.* **7**, 1092–1107 (2020).
5. Chen, W. et al. An artificial chromosome for data storage. *Natl Sci. Rev.* **8**, nwab028 (2021).
6. Li, X., Chen, M. & Wu, H. Multiple errors correction for position limited DNA sequences with GC balance and no homopolymer for DNA-based data storage. *Brief. Bioinform.* **24**, bbac484 (2023).

7. Yan, Z., Qu, G. & Wu, H. A novel soft-in soft-out decoding algorithm for VT codes on multiple received DNA strands. In *2023 IEEE International Symposium on Information Theory* 838–843 (IEEE, 2023).
8. Welzel, M. et al. DNA-Aeon provides flexible arithmetic coding for constraint adherence and error correction in DNA storage. *Nat. Commun.* **14**, 628 (2023).
9. Grass, R. N. et al. Robust chemical preservation of digital information on DNA in silica with error-correcting codes. *Angew. Chem. Int. Ed.* **54**, 2552–2555 (2015).
10. Pan, C. et al. Rewritable two-dimensional DNA-based data storage with machine learning reconstruction. *Nat. Commun.* **13**, 2984 (2022).
11. Qu, G., Yan, Z. & Wu, H. Clover: tree structure-based efficient DNA clustering for DNA-based data storage. *Brief. Bioinform.* **23**, bbac336 (2022).
12. Rashtchian, C. et al. Clustering billions of reads for DNA data storage. *Adv. Neural Inf. Process. Syst.* **30**, 3362–3373 (2017).
13. Zorita, E., Cusco, P. & Filion, G. J. Starcode: sequence clustering based on all-pairs search. *Bioinformatics* **31**, 1913–1919 (2015).
14. Zheng, Y. et al. DNA-QLC: an efficient and reliable image encoding scheme for DNA storage. *BMC Genomics* **25**, 266 (2024).
15. Rasool, A. et al. An effective DNA-based file storage system for practical archiving and retrieval of medical MRI data. *Small Methods* **8**, 2301585 (2024).
16. Wang, K. et al. Storing images in DNA via base128 encoding. *J. Chem. Inf. Model.* **64**, 1719–1729 (2024).
17. Pathak, D. et al. Context encoders: feature learning by inpainting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 2536–2544 (IEEE, 2016).
18. Van der Verren, S. E. et al. A dual-constriction biological nanopore resolves homonucleotide sequences with high fidelity. *Nat. Biotechnol.* **38**, 1415–1420 (2020).
19. Niedringhaus, T. P. et al. Landscape of next-generation sequencing technologies. *Anal. Chem.* **83**, 4327–4341 (2011).
20. Shendure, J. et al. DNA sequencing at 40: past, present and future. *Nature* **550**, 345–353 (2017).
21. Wei, X. et al. Single-cell Stereo-seq reveals induced progenitor cells involved in axolotl brain regeneration. *Science* **377**, eabp9444 (2022).
22. Gornale, S. & Patravali, P. Digital knee X-ray images. *Mendeley Data* https://doi.org/10.17632/t9ndx37v5h.1 (2020).
23. Albertina, B. et al. The Cancer Genome Atlas Lung Adenocarcinoma Collection (TCGA-LUAD). *The Cancer Imaging Archive* https://doi.org/10.7937/K9/TCIA.2016.JGNIHEP5 (2016).
24. Kane, A. & Parker, M. T. Screening state of play: the biosecurity practices of synthetic DNA providers. *Appl. Biosaf.* **29**, 85–95 (2024).
25. Williams, B. & Kane, R. *Preventing the Misuse of DNA Synthesis* (Institute for Progress, 2023).
26. Marcellin, M. W. et al. An overview of quantization in JPEG 2000. *Signal Process. Image Commun.* **17**, 73–84 (2002).
27. Marcellin, M. W., Gormish, M. J., Bilgin, A. & Boliek, M. P. An overview of JPEG-2000. In *Proc. DCC 2000. Data Compression Conference* 523–541 (IEEE, 2000).
28. Antkowiak, P. L. et al. Low cost DNA data storage using photolithographic synthesis and advanced information reconstruction and error correction. *Nat. Commun.* **11**, 5345 (2020).
29. Tang, D. T. & Chien, R. T. Coding for error control. *IBM Syst. J.* **8**, 48–86 (1969).
30. Wicker, S. B. & Bhargava, V. K. *Reed–Solomon Codes and Their Applications* (John Wiley & Sons, 1999).
31. Yan, Z., Liang, C. & Wu, H. A segmented-edit error-correcting code with re-synchronization function for DNA-based storage systems. *IEEE Trans. Emerg. Top. Comput.* **11**, 605–618 (2022).
32. Qu, G. et al. Sequencing data. *figshare* https://doi.org/10.6084/m9.figshare.25957606.v1 (2024).
33. Qu, G. et al. HELIX source code. *Zenodo* https://doi.org/10.5281/zenodo.14699789 (2025).
34. Li, Y. et al. HL-DNA: a hybrid lossy/lossless encoding scheme to enhance DNA storage density and robustness for images. In *2022 IEEE 40th International Conference on Computer Design (ICCD)* 434–442 (IEEE, 2022).
35. Franzese, G. et al. Generative DNA: representation learning for DNA-based approximate image storage. In *2021 International Conference on Visual Communications and Image Processing (VCIP)* 01–05 (IEEE, 2021).
36. Wu, W. et al. Deep joint source-channel coding for DNA image storage: a novel approach with enhanced error resilience and biological constraint optimization. In *IEEE Transactions on Molecular, Biological and Multi-Scale Communications* 461–471 (IEEE, 2023).
37. Dimopoulou, M. et al. A JPEG-based image coding solution for data storage on DNA. In *2021 29th European Signal Processing Conference (EUSIPCO)* 786–790 (IEEE, 2021).
38. Bhaya, C. et al. Encrypted medical image storage in DNA domain. In *ICC 2021-IEEE International Conference on Communications* 1–7 (IEEE, 2021).

Acknowledgements

This work was supported by the National Key Research and Development Program of China (number 2020YFA0712100) and the National Natural Science Foundation of China (number 62071327).

Author contributions

H.W. and X.C. conceived of and supervised the project. G.Q. derived analytical results and performed numerical calculations. G.Q. and Z.Y. analyzed the data. G.Q. wrote the original draft, and H.W. and Z.Y. reviewed and edited it. All authors read and approved the final paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43588-025-00793-x.

Correspondence and requests for materials should be addressed to Huaming Wu.

Peer review information *Nature Computational Science* thanks Zhi Ping, Leyi Wei 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.

nature portfolio

Corresponding author(s): Huaming Wu

Last updated by author(s): Jan 23, 2025

Reporting Summary

Nature Portfolio 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 Portfolio 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 uploaded the code to https://github.com/Guanjinqu/Helix.The code uses python 3.8 .
Data analysis	We uploaded the code to https://github.com/Guanjinqu/Helix.The code uses python 3.8 .We used PEAR(10.1093/bioinformatics/btt593) as sequence splicing software, version 0.9.11.

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 Portfolio 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 description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

Sequencing data from in vitro experiments can be obtained from https://doi.org/10.6084/m9.figshare.25957606.v1.
The images used in the HELIX test are all from publicly available datasets. Spatiotemporalomics image dataset from: https://www.science.org/doi/abs/10.1126/science.abp9444

CT image dataset from: https://www.cancerimagingarchive.net/collection/tcga-luad/
X-ray image dataset from: https://doi.org/10.17632/394t9ndx37v5h.1

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	We are not involved in human research. The experimental images used in the paper are from existing public image datasets.
Population characteristics	Population characteristics are not addressed in this study.
Recruitment	Recruitment is not involved in this study.
Ethics oversight	Ethics oversight is not addressed in this study.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

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	No sample size calculations were performed for this paper. For the in vitro experiments we stored 60MB, which is a larger file size than most of the currently available studies and is sufficient to demonstrate the effectiveness of HELIX. For the simulation part of the experiment we selected three different types of datasets and three images were randomly selected from each dataset. Due to the limitation of the current DNA storage size, too large a sample size will bring too high synthesis and testing costs.
Data exclusions	For the data decoding process, we excluded sequences that could not be spliced successfully.
Replication	The simulation experiments were all repeated three times to assess the effect. The decoding of files after sequencing in the in vitro experiments did not involve the effect of randomness, and the same decoding results were achieved in all three repetitions of the experiments.
Randomization	The simulation experiments were all performed three times under different random seeds, and the decoding of files after sequencing in the in vitro experiments did not involve the effect of randomness, and the same decoding results were achieved in all three repetitions.
Blinding	The experiments in this paper concentrate on evaluating the information recovery ability of the model and do not involve the influence of human subjective factors, therefore blinding is not relevant to this 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
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

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