
Supplementary information

Parallel molecular data storage by printing epigenetic bits on DNA

In the format provided by the
authors and unedited

Supplementary Materials for

Parallel Molecular Data Storage by Printing Epigenetic Bits on DNA

Cheng Zhang^{1†*}, Ranfeng Wu^{1†}, Fajia Sun^{2†}, Yisheng Lin^{1†}, Yuan Liang^{1,3}, Jiongjiong Teng³, Na Liu⁴, Qi Ouyang^{2*}, Long Qian^{2*}, Hao Yan^{5,6*}

Correspondence to: zhangcheng369@pku.edu.cn; qi@pku.edu.cn; long.qian@pku.edu.cn; hao.yan@asu.edu

This PDF file includes:

- Supplementary Materials
- Supplementary Notes
- Supplementary Figures S1 to S62
- Supplementary Tables S1 to S3
- Supplementary References

Other Supplementary Materials for this manuscript include the following:

- MDAR Reproducibility Checklist

Table Of Contents

Parallel Molecular Data Storage by Printing Epigenetic Bits on DNA	1
Supplementary Materials	4
Supplementary Notes	5
NOTE 1. Parallelism of epi-bit DNA storage	5
Note Fig. N1. The parallelisms in DNA storage	5
NOTE 2. Cost and speed analysis (calculations and contributing factors).....	7
Note Fig. N2. The cost and speed influenced by the parallelisms in molecular and operation layers	9
Note Fig. N3. The cost of data recovery algorithm	11
NOTE 3. Errors in the storage pipeline and error correction	13
Note Fig. N4. The repeatability of the panda’ s thresholds	15
Note Fig. N5. Determination of the threshold for methylation calling	16
Note Fig. N6. The number of clustered reads for each of the 25 templates with different barcodes	18
Note Fig. N7. The consensus barcodes and the incorrectly clustered barcodes	19
Note Fig. N8. The analysis of the misalignment errors in iDNAdrive data retrieval.....	19
Note Fig. N9. The methylation calling results in nanopore sequencing under different sequencing coverages	21
Note Fig. N10. Two-site encoding for epi-bit DNA storage.	22
Note Fig. N11. Data retrieval simulation of the tiger rubbing	24
NOTE 4. Storage capacity and scaling potentials	25
Supplementary Figures	26
Fig. S1. Comparisons of the writing cost between de novo DNA synthesis and epi-bit storage	26
Fig. S2. The illustration of the main operation steps of the parallel DNA storage with epigenetic bits	28
Fig. S3. The Comparison between traditional movable-type printing and the parallel epi-bit DNA storage.....	29
Fig. S4. The illustrations and results of the methylation detection assays	30
Fig. S5. The illustrations and gel results of the basic methylation writing tests on a single epi-bit site	31
Fig. S6. The flowchart of the design processes of the DNA bricks and carrier templates	32
Fig. S7. The sequences of DNA carriers	40
Fig. S8. The illustrations of the parallel epi-bit writing by DNA self-assembly typesetting and methyltransferase assisted enzymatic methyl transfer.....	41
Fig. S9. Illustrations of the methylation information retrieval from nanopore sequencing reads	42
Fig. S10. Overview of nanopore sequencing results of the stored text “DNA”	43
Fig. S11. Alignment of basecalled reads of the DNA carriers	44
Fig. S12. The raw nanopore electric signals at the modification sites	46
Fig. S13. Comparisons of the methylation efficiency using varying concentrations of M.SssI.....	47
Fig. S14. The epi-bit writing test on T960 DNA carrier to store “DNA” letters	48
Fig. S15. The results analysis of the stability test experiments	49
Fig. S16. Scheme of image encoding processes.....	50
Fig. S17. Eight images of base chemical structures and the corresponding bitmaps	51
Fig. S18. The main experimental steps of the parallel methylation storage.....	52
Fig. S19. The workflow of parallel epi-bit storage.....	53
Fig. S20. The nanopore sequencing test of 50 kinds of barcoded DNA carries in one-pot reaction.....	54
Fig. S21. Sequencing throughputs required for retrieving files from the methylation-based storage system	55
Fig. S22. Nanopore sequencing cluster results of DNA carriers 7-15.....	57
Fig. S23. Nanopore sequencing cluster results of DNA carriers 16-24.....	58
Fig. S24. Overview of sequencing results in one-pot nanopore sequencing reaction containing 25 kinds of barcoded DNA carriers.....	59
Fig. S25. Alignment of base-called reads of 25 kinds of barcoded DNA carriers in one-pot nanopore	60
Fig. S26. Methylation calling results of three different post-sequencing processing methods	61
Fig. S27. The cluster density of reads with 25 kinds of methylation information.....	62
Fig. S28. The Hamming differences between methylation patterns in the 25 kinds of DNA carriers	63
Fig. S29. The parallel storage of the first molecule structural image.....	64
Fig. S30. The parallel storage of the second molecule structural image	66

Fig. S31. The parallel storage of the third molecule structural image.....	67
Fig. S32. The parallel storage of the fourth molecule structural image	68
Fig. S33. The parallel storage of the fifth molecule structural image	69
Fig. S34. The parallel storage of the sixth molecule structural image	70
Fig. S35. The parallel storage of the seventh molecule structural image	71
Fig. S36. The parallel storage of the eighth molecule structural image	72
Fig. S37. Methylation calling results in nanopore sequencing of the 25th 960nt DNA template	73
Fig. S38. The repeatability of the thresholds in Fig.4g	73
Fig. S39. The workflow of retrieving the methylation information from nanopore sequencing results in the experiments of the large-scale epi-bit storage.....	74
Fig. S40. Data analysis of the four samples of storage of the tiger rubbing image.....	75
Fig. S41. Overview of Epi-bit calling results of Sample 4, which underwent compression and error correction coding.....	76
Fig. S42. Simulated decoding of the stored color image (panda).....	77
Fig. S43. The numbers of errors in Panda for each encoding group	78
Fig. S44. Comparisons of the data retrieving when using sequencing batches of different sizes	79
Fig. S45. Retrieved epi-bit streams of the panda image storage experiment, well 1-90	80
Fig. S46. Retrieved epi-bit streams of the panda image storage experiment, well 91-180	81
Fig. S47. Retrieved epi-bit streams of the panda image storage experiment, well 181-270	82
Fig. S48. Retrieved epi-bit streams of the panda image storage experiment, well 271-360	83
Fig. S49. Retrieved epi-bit streams of the panda image storage experiment, well 361-450	84
Fig. S50. Retrieved epi-bit streams of the panda image storage experiment, well 451-540	85
Fig. S51. Retrieved epi-bit streams of the panda image storage experiment, well 541-630	86
Fig. S52. Retrieved epi-bit streams of the panda image storage experiment, well 631-720	87
Fig. S53. Retrieved epi-bit streams of the panda image storage experiment, well 721-756	88
Fig. S54. Comparison of iDNAdrive.....	89
Fig. S55. The demographic analysis of 60 student volunteers involved in iDNAdrive	90
Fig. S56. The experimental pipeline of iDNAdrive	91
Fig. S57. Preparation of the epi-pit data writing kit and the illustrations of DNA templates used in iDNAdrive.....	92
Fig. S58. Modes of 0-to-1 substitution errors in iDNAdrive.....	93
Fig. S59. Encoding, decoding and spelling error correction in iDNAdrive	94
Fig. S60. The recovered texts in iDNAdrive. GPT-4 was used for spelling error correction.....	95
Fig. S61. Epi-bit information copy.....	96
Fig. S62. Random access of epi-bit information.	97
Supplementary Tables.....	98
Table S1. List of alternative unconventional DNA storage schemes	98
Table S2. Error analysis for epi-bit storage.....	99
Table S3. The DNA sequences used in the experiments.....	100
Supplementary references.....	124
Acknowledgements to the student volunteers.....	125

Supplementary Materials

All oligonucleotides (listed in Table S3) were obtained from Sangon Company (China). The methyl and fluorophore modified DNA molecules were purified with high-performance liquid chromatography. The DNA sequence designs were analyzed with the use of NUPACK (<http://www.nupack.rog>) to ensure that minimal cross-talk between undesirable domains. For convenience, all DNA strands were stored in 10-100 μ M stocks in ultrapure H₂O at -20 °C (NanoDrop OneC, Thermo Scientific). Recombinant DNMT1 protein (#0230, #31800) were obtained from NEB and Active Motif. M.SssI methyltransferase (#M0226S), T4 PNK (#M0201S), T4 DNA Ligase (#M0202S), SAM (#B9003S) were obtained from NEB (China). NucleoSpin Gel and PCR Clean-up (740609.250) were obtained from Macherey-Nagel. Restriction endonuclease Gla I (E493) was obtained from SibEnzyme. 2 $\times$ TransStart®FastPfu PCR SuperMix (#AS221-01) were obtained from TransGen Biotech (China).

Supplementary Notes

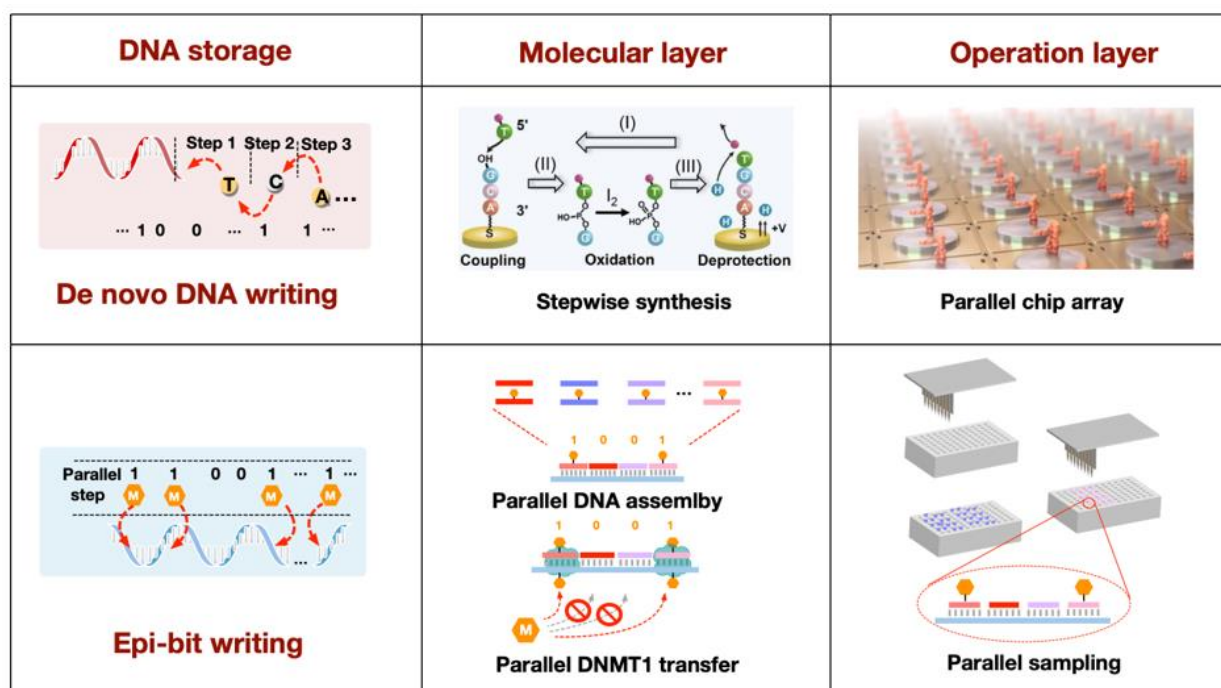
NOTE 1. Parallelism of epi-bit DNA storage

(1) Analysis of the writing parallelism

In epi-bit DNA storage, the parallelism of data writing manifests itself on two levels: the molecular level and the technical operational level:

$$\text{write parallelism} = \text{bit parallelism} \times \text{operational parallelism}$$

On the molecular level, the bit parallelism is defined as the number of bits written/synthesized in one reaction spot and in one reaction cycle. The bit parallelism is determined by the molecular mechanism of data writing. On the operational level, the write reactions can be implemented on highly parallelized platforms with each reaction independently controlled, such that a large number of reactions can proceed simultaneously (Note Fig. N1).



Note Fig. N1. The parallelisms in DNA storage. The parallelisms in the molecular layer and the operation layer. The molecular layer parallelisms of de novo and epi-bit DNA storage.

The parallelism of epi-bit writing

The epi-bit DNA storage achieves dual level parallelisms on both the molecular level and the operational level. Specifically, this study proposes an epi-bit framework to shift information bits

from bases to base modifications, so that through one-time hybridization (DNA self-assembly), multiple movable type blocks can be parallelly typeset onto the template chain. Then, the DNMT1 enzyme is used for parallel epi-bit printing (Note Fig. N1). The two processes together enact the molecular-level parallelism of epi-bit writing. In our particular demonstration, on the molecular level, 350 bits are written simultaneously in a single reaction. Meanwhile, the high-throughput automated sampling operations jointly achieves highly parallel data writing.

The parallelism of *de novo* synthesis

Overall, the parallel writing of *de novo* synthesis based DNA storage mainly utilizes the parallelism on the operational level parallelism. On the molecular level, writing of each data bit is implemented by adding bases one after another through iterative reactions, which is essentially serial. In current mainstream DNA storage, *de novo* DNA synthesis can be made parallel via control mechanisms to select which bases to add to which strands on the operational level. This enables the high-throughput synthesis of different oligo sequences in different spots on an integrated solid-state microarray chip and is often called array-based synthesis¹. Technically, massive parallelism on the operational level is achieved through independent (or addressable) control of each reaction by electrochemical synthesis (e.g., CustomArray/GenScript), light-directed synthesis (e.g., LC Sciences), thermal controlled synthesis (e.g., Evonetix) and micro-inkjet printing technologies (e.g. TwistBioscience)²⁻⁵. For example, it was reported that *de novo* DNA storage can be achieved using chip technology for highly parallel synthesis of 1.3 MB DNA data in 2020². Meanwhile, inkjet printing is also one of the highest throughput methods developed to synthesis DNA in recent years. In the year 2023, Verardo et al. achieved far more than 9600 sites of synthesis by inkjet printing³. At present, both of these parallel operation techniques have been widely applied in commercial *de novo* DNA synthesis.

(2) Impacts of bit parallelism on the speed and cost of data writing

The epi-bit DNA storage achieves dual level parallelisms, of which the high bit-parallelism on the molecular level bears profound implications on the speed and cost of data writing.

Impact of molecular level bit-parallelism on write speed

The fact that all information on a template molecule can be written by methylation transfer simultaneously in one reaction makes sample addition (a fast step) decoupled from the writing

reaction (a rate-limiting step). As a consequence, data writing proceeds at the speed of typesetting/sample addition, not having to wait for the printing/methyl transfer reactions. In contrast, in serial *de novo* DNA synthesis, the writing of the next bit has to wait until the current cycle of base addition ends (one cycle is approximately 1-20 mins). Therefore, the write speed is much enhanced by the molecular parallelism, especially given the state-of-art liquid dispensing platforms.

Impact of molecular level bit-parallelism on write cost

First, due to the catalytic activity of the DNMT1 enzyme, the enzyme usage does not need to scale proportionally with the number of bits (parallel catalysis). Hence, the per bit enzyme cost could be effectively reduced with increased bit parallelism (Note Fig. N1). Second, because universal DNA templates and bricks are used to encode arbitrary data, the templates and bricks can be synthesized in bulk or amplified cheaply. This brings about another factor for cost reduction (Note Fig. N1).

(3) Impacts of reaction throughput on the speed and cost of data writing

The operational level parallelism, i.e. the reaction throughput, refers to the number of reactions running at each independently addressable reaction sites on the microarray chip or the number of micro-inkjet printheads independently controlled on the liquid sampling platforms. An increase in throughput is often associated with increases in synthesis scale and spot density, and a decrease in reaction volumes. These help to save on reagent use and price, and thereby reduces the cost of data writing per bit. In the meantime, an increase in throughput also increase the writing speed proportionally because more data writing reactions can be implemented simultaneously.

NOTE 2. Cost and speed analysis (calculations and contributing factors)

Analyses of costs includes aspects in computational process (e.g. on silicon analysis) and wet experimental process, including writing cost (e.g. DNA synthesis, methylation transfer), reading cost (e.g. nanopore sequencing), storage cost (e.g. energy charge) and encoding/decoding cost.

(1) Cost calculations

In the epi-bit DNA storage, the cost can be broken down to the cost of predesigned DNA (for typesetting, C_{DNA}), and the cost of methylation transfer (for epi-bit printing, C_{m}).

For the cost of DNA, brick DNA synthesis and modification are negligible ($\$5\text{E}-5/\text{bit}$), whereas the synthesis of long DNA templates takes a significant fraction of the cost ($\$5.5\text{E}-3/\text{bit}$). However, long DNA templates can be PCR amplified instead of commercially synthesized. This brings the total DNA cost down to $\$6.8\text{E}-4/\text{bit}$.

For the cost of enzymes, the main cost is incurred by the DNMT1 methyltransferase at $\$2.5\text{E}-3/\text{bit}$. The T4 ligase is only used when sequence barcodes are required to be linked to the templates.

Combining the two sources, the overall cost range is estimated to be $\sim \$0.00325/\text{bit}$ - $0.00916/\text{bit}$, the lowest of which is ~ 5 -fold of de novo synthesis-based data writing ($\sim 0.0007/\text{bit}$)⁶. It should be noted that the current calculation is based on lab-scale 10 uL reaction systems and commercial enzymatic or DNA synthesis prices. The DNA and enzyme were used at saturating concentrations without further optimization. The final product yield is ~ 0.06 pmol for each information strand, which is also larger than the oligo synthesis yield.

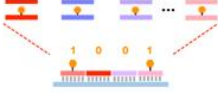
Writing cost:

The writing cost for epi-bit storage is $\sim \$0.00325/\text{bit}$ - $0.00916/\text{bit}$ for 10 uL reaction systems and at commercial reagent and enzyme prices in China.

It should be noted that “reusable” means that the DNA sequences but not the actual molecules are reused, which is the similar concept as reported in Roquet et al⁷. Likewise, the enzymes and other reagents are consumed with the experiments. Actually, due to the catalytic mechanism, a larger bit parallelism will increase the efficiency of enzymatic usage and reduce enzyme consumption per bit in each reaction. In general, enzyme consumption scales with the number of bits but not with the length of the underlying DNA sequence (the number of bases). Therefore, the costs associated with enzyme consumption is independent of the coding density (Fig. S1D, flat line). Given the calculations above, the write cost of the prototypic epi-bit system is already within close margin to that of the well-developed de novo synthesis technology for DNA storage.

Storage cost:

Given preliminary test results on methylation stability (Note Fig. N2), it is likely that methylated DNA can be preserved in more or less similar conditions to unmodified DNA. It is estimated in its freeze-dried form that DNA preservation takes less than $\$0.36 \times 10^{-15}/\text{bit}/\text{year}$ of energy consumption, without the need for periodic turnovers of storage media like tape or hard disks require.

Epi-bit storage	Molecular level influences	Operational level influences
	 Parallel DNA assembly	 DNMT1 transfer reaction
 Cost	- The numbers of epi-bits written in a single reaction - Numbers: From 350 to 3500 → Increasing 10 folds - Cost: Decreasing ~2 folds (enzyme \$/bit)	- The reaction volume - Volume: From 10 uL to ~10 pL → Decreasing 1000000 folds - Cost: Decreasing 1000000 folds (enzyme \$/bit) (Roquet et. al., bioRxiv, 2021)
	- Amount of DNA prefabrication - Amount: From 1 to 100 OD → Increasing 100 folds - Cost: Decreasing ~5.85 folds (\$/ug) - Cost of enzymes: 10 folds decrease expected (enzyme \$/bit)	- Amount of brick modification - Amount: From 1 to 100 OD → Increasing 100 folds - Cost: Decreasing ~8 folds (\$/ug)
 Speed	- The number of modifications per brick - Numbers: From 2 to 4 → Increasing 2 folds - Speed: Increasing 2 folds	- The liquid dispensing rate - Rate: From ~0.08 to 10 kHz → Increasing 125 folds - Speed: Increasing 125 folds

Note Fig. N2. The cost and speed influenced by the parallelisms in molecular and operation layers.

Reading cost:

The epi-bit data retrieval was done by nanopore sequencing followed by read analysis and decoding. Of the total cost, the most part was spent on nanopore sequencing, while the computational pipeline costs negligibly small. With regard to sequencing, to ensure reliable data recovery, we used a rather high sequencing depth in the demonstrations. It is expected that the sequencing cost to drop significantly with further improvements such as reducing the sequencing coverage, applying more efficient methylation calling algorithms and error correction codes, and employing better barcoding strategies.

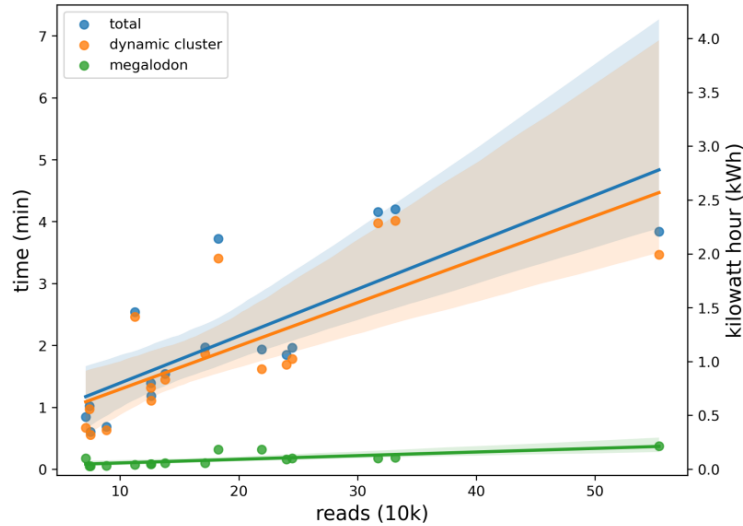
Encoding and decoding cost

The encoding process in this work is mainly comprised of error correction and sparsification, while the decoding is essentially the reverse process of these two procedures. In fact, sparsification is not a must-have step, and would not add significant computing complexities. Regarding error

correction, we used linear block code (i.e., BCH code) for error correction, which is computationally cheap due to the short codeword length (255 bit per group) and the simplicity of the code itself. As a result, encoding and decoding cost no more than dynamic clustering, and is negligible compared to wet-lab costs.

(2) Cost analysis on data recovery

Since the running time of the algorithms was mostly related to the number of reads (sequencing depth), we therefore investigated the relationships between the algorithm time and the number of reads. As evident from Note Fig. N3, runtime and number of reads are linearly related. Megalodon performs rapid methylation calling in linear time ($\mathcal{O}(n_{reads})$), whereas dynamic clustering constitutes the major portion of the total runtime for the entire algorithm. Because we employed a k-means strategy for clustering the reads, dynamic clustering takes linear time in the number of reads. Despite this, dynamic clustering significantly minimizes errors, and its time overhead is deemed acceptable in comparison to the processes of data writing and sequencing. Furthermore, the cost of this process is rather low when compared to data writing and sequencing, requiring ~ 19.5 kWh of power in total for the recovery of 252504 bits (sequencing of the panda image). To provide a reference, this is equivalent to $\$ 8.17\text{E-}6 \sim 2.11\text{E-}5$ /bit based on the electricity price in China or in the US. In the data retrieval of the Panda image, the cost of the computation pipeline is $\sim \$4.4\text{E-}5$ / bit. These costs included algorithm development trials and have not been optimized.



Note Fig. N3. The cost of data recovery algorithm. The horizontal axis represents the number of reads, the left vertical axis represents the algorithm runtime, and the right vertical axis represents power consumption.

(3) Cost in relation to archive size and sequencing requirements

The cost also relates to archive size and sequencing requirements, because the total write cost is the product of write cost per bit and the archive size (number of bits). For the former factor, if the archive size is small and the data is to be synthesized independently of other data, this will incur a higher cost due to a fixed batch price for commercial synthesis services. For the latter factor, the synthesis should yield sufficient numbers of molecules for DNA sequencing. For data retrieval based on PCR, it is suggested that a minimum copy number of 10 molecules is sufficient for PCR amplification⁸. Without PCR, the minimal sequencing requirement is 50 fmol per the vendor's suggestion (Oxford nanopore technologies, on MinION platform). For the current epi-bit system, the experiment of 10uL system produces 0.06 pmol of each DNA templates bearing information, and the data is readily retrievable on the MinION platform.

(4) Cost and time limits for *de novo* synthesis

Conversion between speed, length and cost metrics

For *de novo* synthesis, the factors of speed, length and cost metrics are relatively independent.

Speed: the speed is determined by the reaction time per cycle of nucleotide addition and the throughput. *de novo* DNA synthesis protocols add one nucleotide per cycle and the speed can be significantly increase by massively parallel platforms. **Length:** the major determinant of the synthesis length is the elongation cycle efficiency, i.e., the efficiency with which each nucleotide is integrated into the oligo strand. Failure of nucleotide integration results in truncated oligos or in a small chance deletions or insertions in the final synthetic DNA. **Cost:** the major determinants of cost for today's *de novo* synthesis include the reaction volume and the size of each individual reaction spot. A reduction in volume lead to fewer reagent use and hence a reduced cost¹. Although these aspects have their major determinants, some technologies tackle more than one of them simultaneously. For example, in electrochemical oligo synthesis, effectively increasing the spot density and reducing the spot size will increase the speed and reduces the cost simultaneously.

The limits of *de novo* synthesis

The cost and time limits for *de novo* synthesis based on phosphoramidite chemistry and TdT-mediated enzymatic strategy here.

The major limitation of the phosphoramidite synthesis comes from parallelization. Finding a way to reliably increase the spot density and decrease the spot size while still maintaining independent control and insulation is important. Likewise, further technical advances in micro-inkjet printing platforms may help break the bottleneck in synthesis cost and time.

Recently, many DNA storage studies have suggested the enzymatic synthesis may be more efficient than phosphoramidite synthesis when considering cost and time per base (cycle). For example, Lee et al.⁹ gave a detailed analysis about the cost and time of the two synthesis methods. In their results, the time per cycle by using enzymatic synthesis is 5-6 times lower than that of phosphoramidite synthesis. However, considering the enzymatic DNA synthesis is still in its infancy, there is a lot room to decrease the cost and improve the speed.

(5) The possible improvements on cost and speed

The cost improvements of epi-bit DNA storage

Given the technological maturity in nanopore sequencing, like in mainstream DNA data storage, the storage cost of epi-bit storage mainly comes from the data writing process. In this work, we estimated the data writing costs of epi-bit DNA storage mainly from two aspects: the cost of predesigned DNA (for typesetting, C_{DNA}) and the cost of methylation transfer (for epi-bit printing, C_{m}).

As discussed above, the estimated cost is closely related to the parallelisms both on the molecular level and the operational level (Note Fig. N2). In the epi-bit storage, the cost is influenced by the molecular level parallelism in movable type self-assembly and methyl transfer (Note Fig. N2). First, by the effective usage of enzymes on a larger bit parallelism. For example, the bit parallelism can increase 10 folds by designing more DNA bricks and templates to write 3500 bits instead of 350 bits in a single reaction (the self-assembly scale of which is equivalent to the reported work to form large DNA origami¹⁰). On the other hand, due to the catalytic activity of the DNMT1 enzyme, the enzyme usage need not be 10-fold higher. Second, because large quantities of DNA movable types and enzymes can be used over numerous data writing iterations, these consumables can be synthesized or purchased in bulk to greatly reduce the cost. For example, the cost of large-scale synthesis of 100 OD 20 nt DNA is ~\$0.00648/ug, which is almost 5.85 times

cheaper than that of 1 OD DNA (~\$0.03791/ug). Additionally, as the DNA bricks are typically of primer-length (~20 nt), their synthesis is much cheaper than DNA oligos (e.g. the prices of synthesizing 1 OD 20 nt and 100 nt DNA are ~\$0.00648/ug and ~\$0.73697 /ug, respectively).

The writing speed improvements of epi-bit DNA storage

In the epi-bit system, the steps of typesetting and methylation transfer can be decoupled. Therefore, the write speed is essentially the speed at which the DNA bricks are sampled (i.e., typesetting). In this sense, the writing speed can be significantly increased by combining advanced automatic sampling technology.

In this work, we employed a self-made primitive liquid dispensing device with four nozzles working collectively at 0.08kHz sampling frequency for the large-scale storage experiment, yielding a write speed of 40 bits/s (Fig. 5). However, on state-of-the-art commercial microjet printing platforms, inkjet printheads are capable of dispensing inks at 10kHz rates from nozzle rows each containing thousands of nozzles working independently in parallel⁷. Therefore, achieving faster speed for epi-bit data writing depends on the improvements of the parallelism and control in the sampling system, including increases in the number of nozzles, sampling rates and control precision (Note Fig. N2). On the other hand, further increasing the bit parallelism, by designing more orthogonal templates and complementary bricks, can improve the write speed at fixed sampling rates (Note Fig. N2).

NOTE 3. Errors in the storage pipeline and error correction

(1) Possible error sources

In the four sets of experimental demonstrations, we obtained error rates in the range of 0.625% (by sequence-based barcoding, Fig. 4) ~ 3.67% (by epi-bit barcoding, Fig. 5). The error rates we quoted in the main text for each experiment were composite errors resulted from both writing (due to methylation efficiencies) and reading (from sequencing & read alignment). Table S2 listed the possible error sources and their patterns in the storage workflow and possible experimental or algorithmic solutions to these errors. Since we have not done a comprehensive characterization of errors occurring during prolonged molecule storage, these errors were omitted in Table S2. Notice that the epi-bit storage framework rarely produces insertion-deletion errors as in mainstream framework based on *de novo* synthesis. In this study, we employed a few methods to reduce the

composite error rates (* in Table S2), such that the remaining errors could be rectified by error correction codes.

As a common sequencing technology, nanopore sequencing exhibits distinctive error profiles, where the errors are sequence-specific or context-dependent. Here, we discuss the effects of these errors on the epi-bit storage from the following aspects.

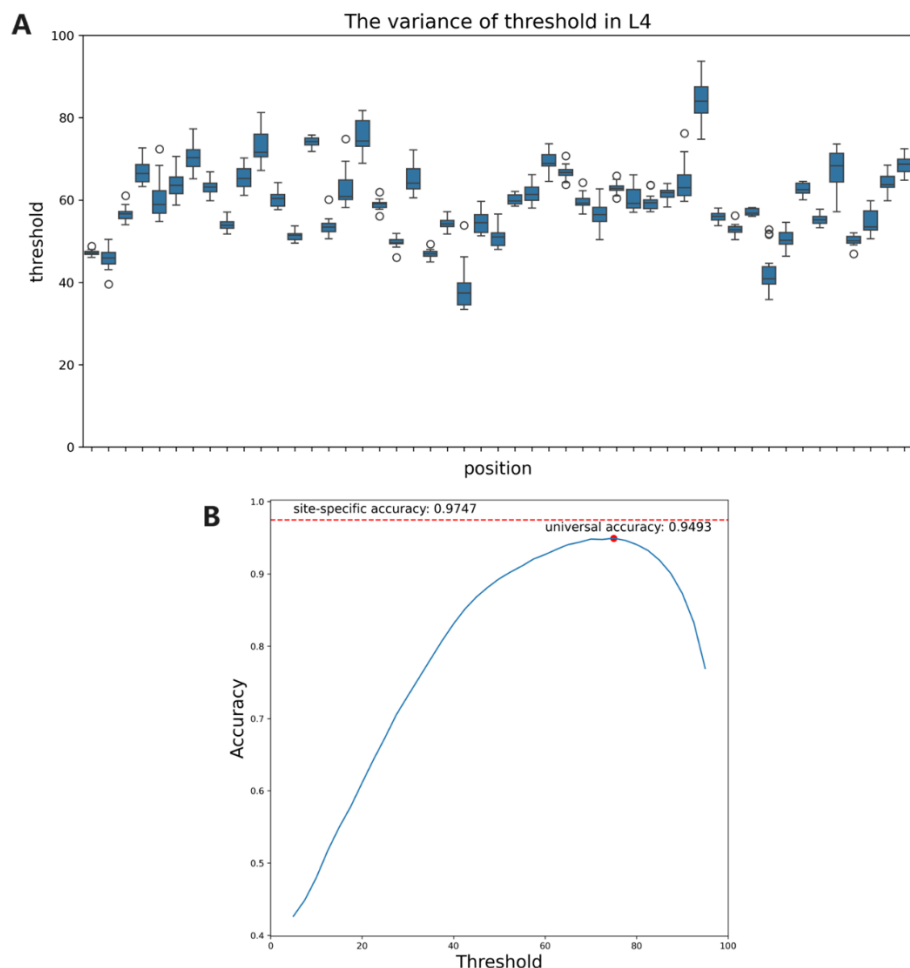
Read errors due to 5mC miscalling

First, since the template DNA sequences are predesigned and remain unchanged throughout batches of data writing, the recognition of epi-bits can be improved by comparing the output signals of methylated sequences with that of the unmethylated reference sequence¹¹. When using Megalodon to perform methylation calling, significant signal variations are observed between the methylated and unmethylated samples at each epi-bit site, thus achieving accurate results (Fig. 2e). Our experimentally characterized single-read error rate of 5mC calling was on par with other published data¹². Correctly aligning the reads (Fig. 5b Sample (2)) and performing site selection (Fig. 5b Sample (3)) successfully decreased the error rate to ~3.7% which was manageable by error correction coding (Fig. 5d&e). It is foreseeable that the error rate will further decrease in light of the ongoing development in 5mC calling algorithms and other powerful sequencing technologies such as PacBio SMRT HiFi sequencing.

Context-specific 5mC errors

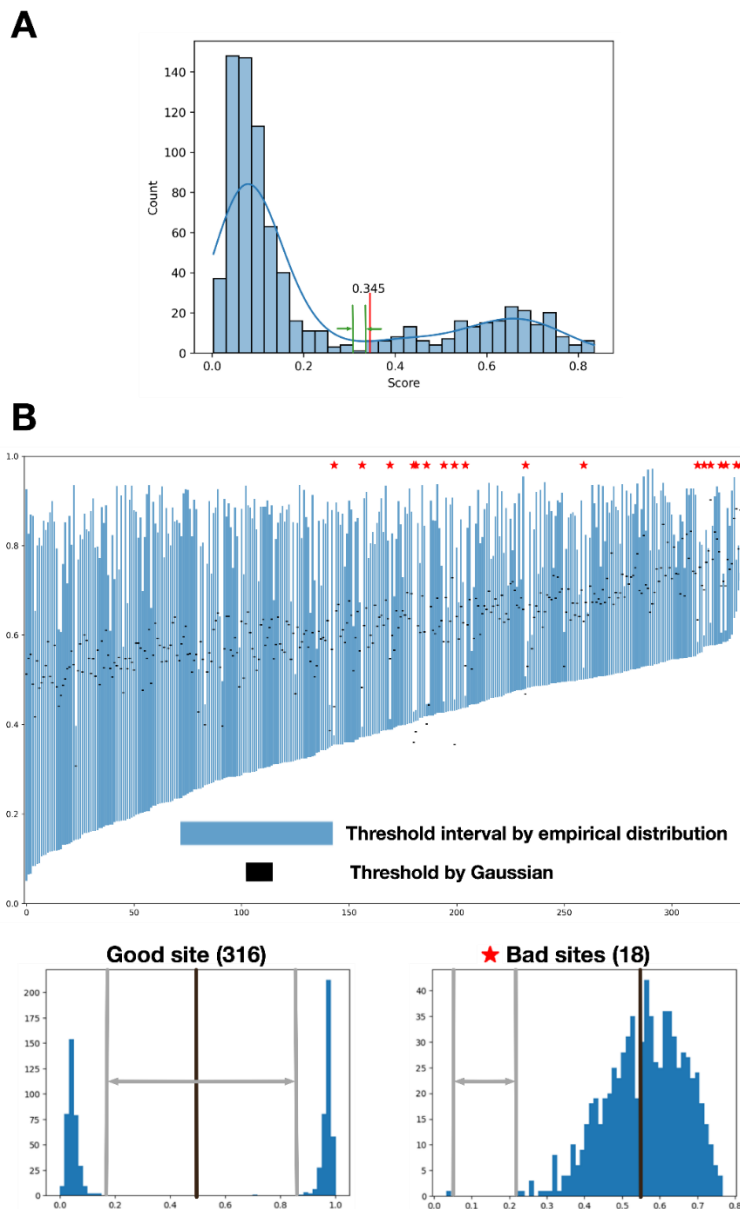
Second, it is reported that context-dependent errors are usually found in nanopore sequencing¹³⁻¹⁴. Specifically, the flanking sequences around the CpG site (a few nucleotides upstream and downstream of the CG site) did have an impact on the error rate of methylation calling¹². In our study, due to the diversity of sequence contexts around the CpG sites on the DNA bricks, the methylation probability distribution of different epi-bit sites were not the same. In the large-scale storage experiment of the Panda image, each template was used with >700 different barcodes to store different data segments. Therefore, we had sufficient data to be able to evaluate and set an independent threshold for each epi-bit site (i.e., site-specific thresholds), which greatly improved the retrieval accuracy of the Panda image. We have calculated the site-specific threshold along epi-bit positions on the L4 template across the 16 sequencing runs of the panda data (Note Fig. N4A). The vast majority of the site-specific thresholds exhibited relatively small fluctuations over different sequencing runs, suggesting repeatability to some extent. However, one can calibrate these site-specific thresholds to specific sequencing data to reduce errors. Nevertheless, as Note

Fig. N4B indicates, using a universal threshold across all sites yielded ~3% extra errors compared to using site-specific thresholds. This was consistent with our observation that at most epi-bit sites, the distributions for 1's and 0's are quite disjoint that the error rate is robust to specific threshold selections (Note Fig. N5). Lastly, thresholds are also affected by the data writing efficiency. Therefore, further improvements in the data writing protocol, such as tuning DNMT1 concentrations and reaction times, will help to stabilize the epi-bit distributions and reduce errors.



Note Fig. N4. The repeatability of the panda's thresholds. (A) The variance of site-specific thresholds in L4 template. We have calculated the site-specific threshold along epi-bit positions on the L4 template across the 16 sequencing runs of the panda data. The vast majority of the site-specific thresholds exhibited relatively small fluctuations over different sequencing runs, suggesting repeatability to some extent. The horizontal axis is the CpG cite positions in L4, while the vertical axis depicts the calculated site-specific threshold. Box plots show the distribution of the data in second bar plots ($n = 16$ for each position). (B) The relationship between error rate and

universal threshold. using a universal threshold across all sites yielded ~3% extra errors compared to using site-specific thresholds. This is consistent with our observation that at most epi-bit sites, the distributions for 1's and 0's are quite disjoint that the error rate is robust to specific threshold selections (Note Fig. N5). The horizontal axis is the universal threshold, while the vertical axis is the error rate of Panda.

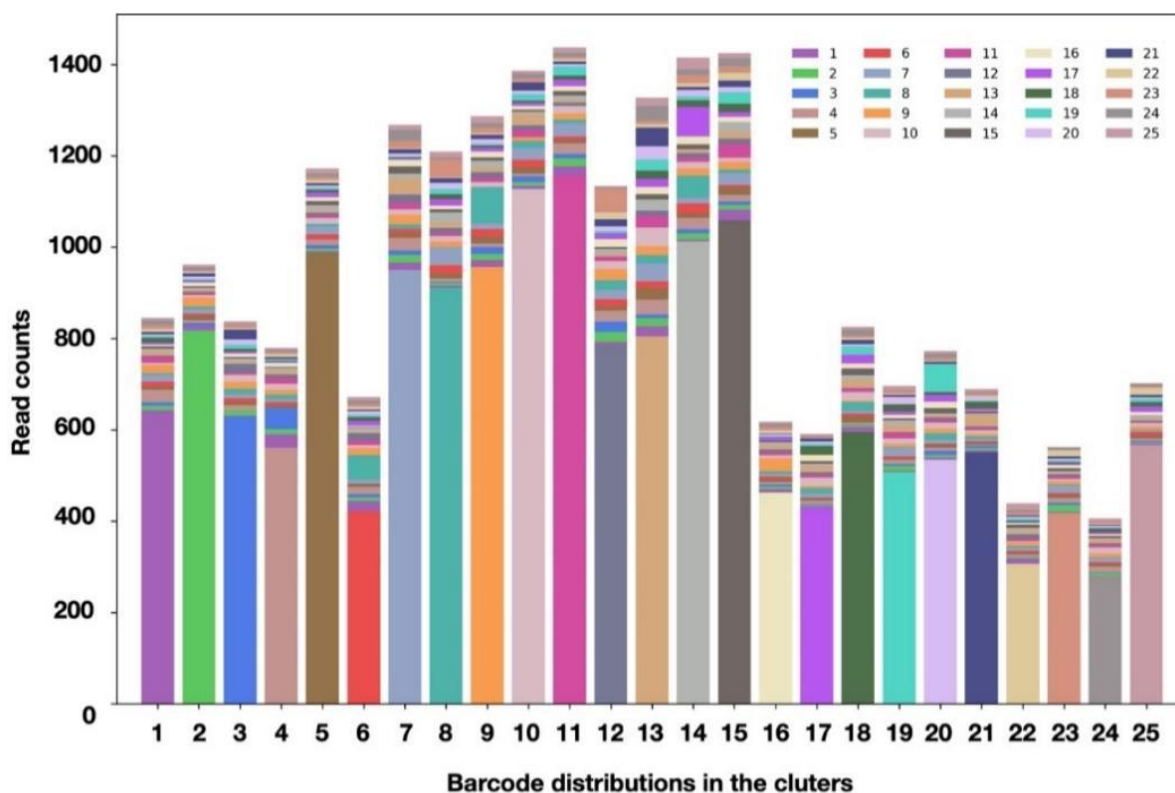


Note Fig. N5. Determination of the threshold for methylation calling. (A) Threshold selection for the enlarged epi-bit storage (Fig. 4g in main text). The red line was the threshold determined by dual-Gaussian distributions. The interval enclosed by green lines was the threshold interval determined by the empirical distribution. (B) Per-site threshold selection for large-scale epi-bit

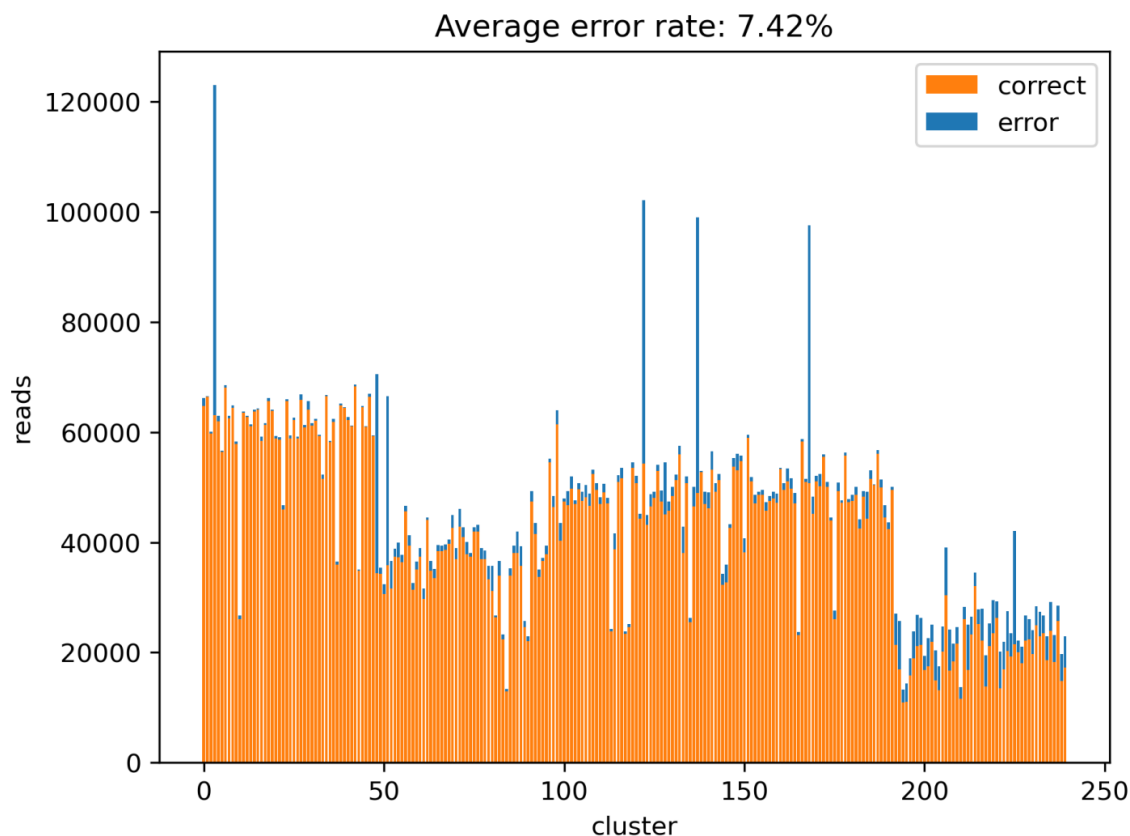
storage (Fig. 5 in main text). In 94.61% of the sites, the thresholds selected by mixed Gaussian model was in the threshold interval selected by the empirical distribution. In 18 sites (indicated by red stars), the threshold by mixed Gaussian fell outside the interval determined by empirical distributions. These 18 discrepant sites had promiscuous distributions.

Errors due to read mis-alignment

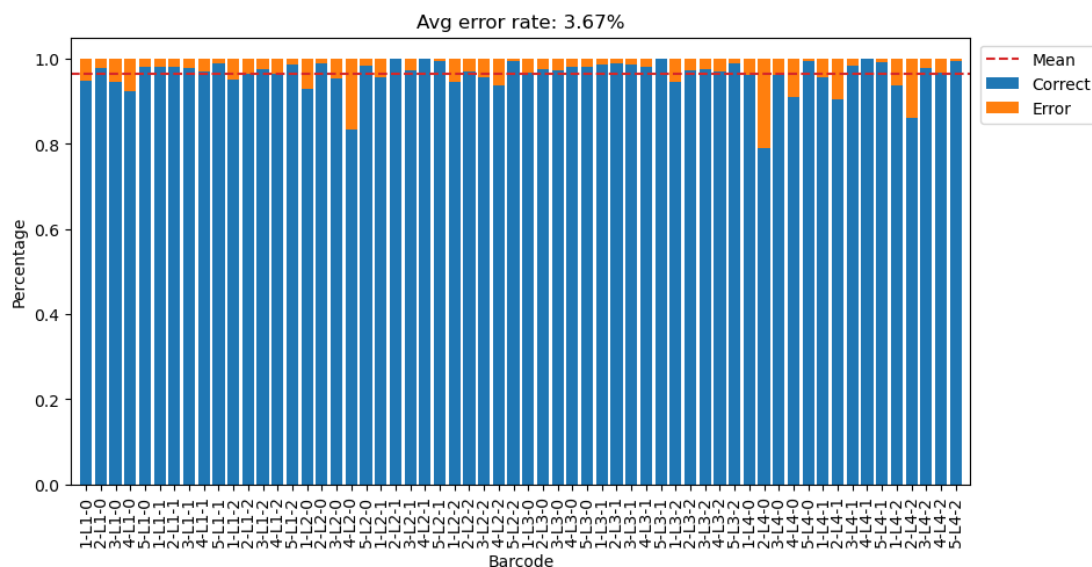
In the experiment of storing images of DNA bases and derivatives, about 25.5% of barcodes were found not matching the information they carry (Note Fig. N6). As we decomposed the error sources into methylation errors and mis-ligation of barcodes (Fig. 4i), mis-ligation of barcode sequences *per se* was at a rate of 0.9%. This was because integrating barcodes with the template DNA involved multiple processes of T4 ligation and gel purifications. Alternatively, we utilized CpG sites as barcode sites and implemented dynamic clustering to correct for potential barcode errors, in the large-scale experiments (Fig. 5b). For example, in the storage of the Panda image, the barcodes were represented in epi-bits, which avoided enzymatic ligation and purification procedures and hence the 0.9% mis-ligation errors. With the dynamic clustering algorithm, alignment of barcodes from erroneous single-read methylation profiles was obtained at a reduced error rate of 7.42% (Note Fig. N7), attesting the effectiveness of applying dynamic clustering to epi-bit barcodes. Lastly, in the iDNAdrive experiment (Fig. 6), we employed a new strategy of introducing sequence barcodes, which suppressed the overall mis-alignment rate to 3.67% (Note Fig. N8). This represents a huge improvement from the 25.5% misalignment rate in Note Fig. N6. These improved barcoding strategies may be further combined to facilitate read alignment and information retrieval. Collectively, they lay the foundation for the practicability of epi-bit DNA storage.



Note Fig. N6. The number of clustered reads for each of the 25 templates with different barcodes. Here reads with different barcodes are labelled with corresponding colors. The largest proportions of the barcodes were consistent with predesigned barcodes for each cluster. The prediction results demonstrate the feasibility to retrieve the multiple modification information from the DNA carriers by using the one-pot nanopore sequencing in the high-throughput way.



Note Fig. N7. The consensus barcodes and the incorrectly clustered barcodes. The number of clustered reads for each of the 240 carriers in methylation barcode schema.



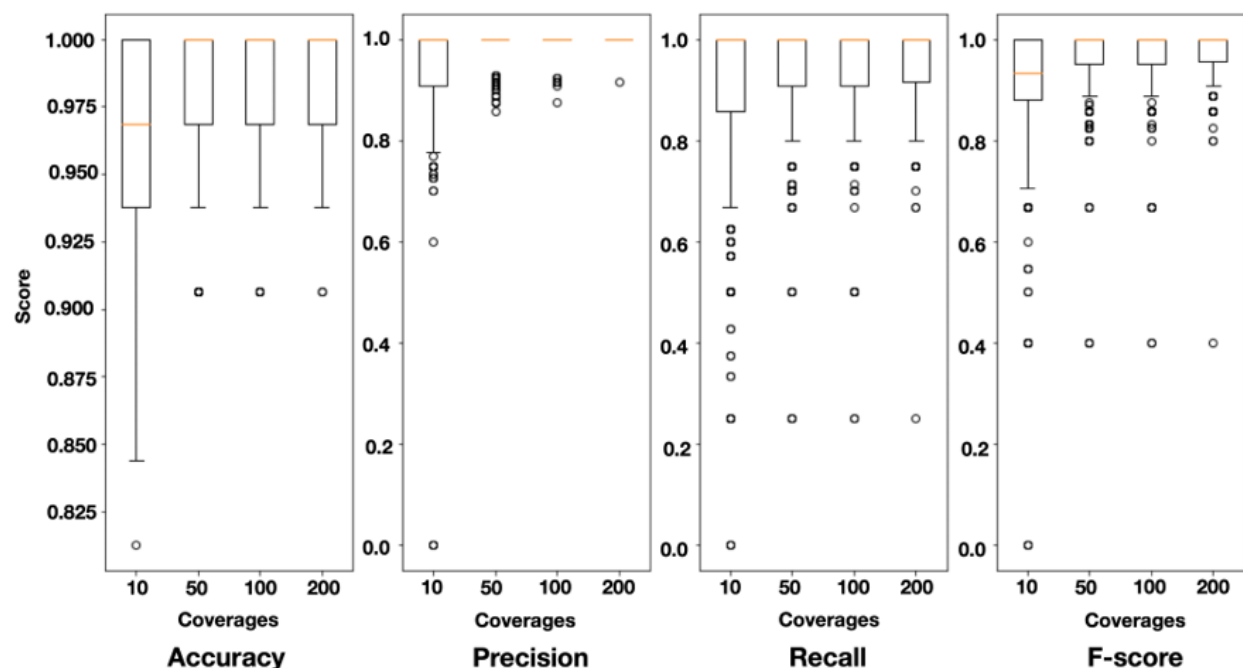
Note Fig. N8. The analysis of the misalignment errors in iDNAdrive data retrieval. The barcoded templates are aligned on the x-axis, with the y-axis showing the percentage of misalignment errors. The analysis was done in a similar way to Fig. 4h. Barcodes were first determined by base calling of template sequences (reads). Reads were further clustered based on

their epi-bit patterns. The inconsistencies between barcode groups and epi-bit clusters were designated as misalignment errors. Therefore, the misalignment error reflects the combination of mis-barcoding and errors in epi-bits.

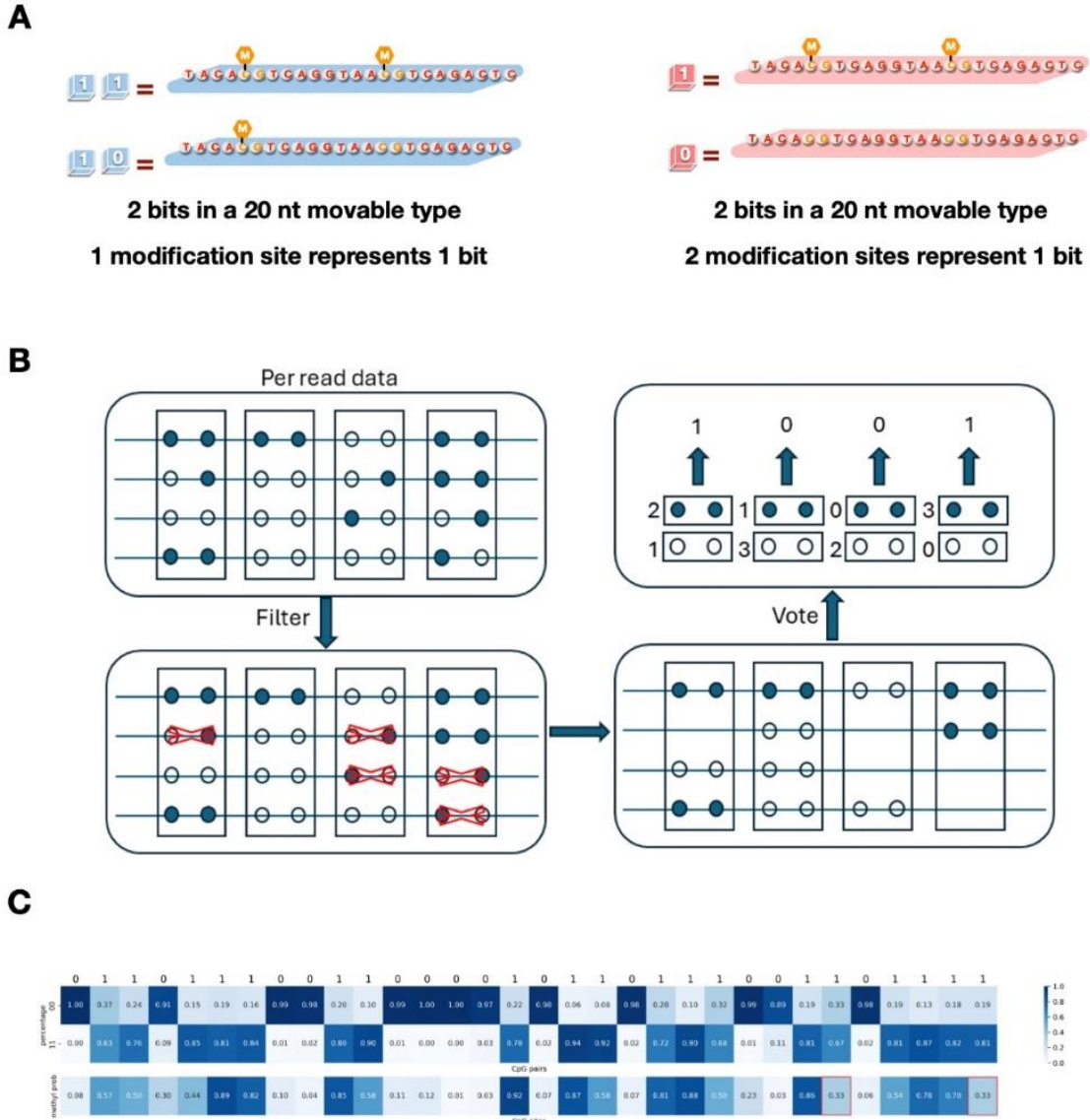
Errors' impact on sequencing throughput and cost

After removing the context-dependent methylation calling errors, in general, the nanopore reading of epi-bit information is reliable, as well demonstrated by the retrieval of more than 275,000 bits information. However, compared to base calling, the elevated error rates of methylation calling, although not context dependent, do have nonnegligible impacts on sequencing throughput and resource cost. In particular, the error rate limits the sequencing throughput (the number of strands resolvable in one-pot nanopore sequencing) when the strands are barcoded in epi-bits. In our analysis of the sequencing reads, while the consensus voting of methylation was quite reliable as the read number exceeds dozens (Note Fig. N9), the proper alignment of reads having the same barcode, especially when the barcodes were present in the form of epi-bits, presented challenges. We devised the dynamic clustering algorithm to improve read alignment and thereby the retrieval accuracy to be able to resolve 240 kinds of methylation barcoded information strands in one-pot nanopore reaction (Fig. 5b). Better barcoding strategies such as the one used for iDNAdrive (Fig. 6) will certainly reduce mis-alignment errors as well.

The elevated error rates, especially in the barcode regions, will require more sequencing reads and longer computation time to be resolved for accurate informational retrieval. These will incur additional resource costs for information readout. We expect that better barcoding strategies, more accurate algorithms and experimental design optimizations will ultimately reduce the resource cost at the sequencing end for epi-bit data storage.



Note Fig. N9. The methylation calling results in nanopore sequencing under different sequencing coverages. The accuracy, precision, recall and F-score of methylation detection under sequencing coverages of 10×, 50×, 100× and 200×. Here, the orange lines are median values, and the range of the box is from 25% to 75%. The accuracy of the detection of nucleotide sequence is closely related to the sequencing depth in nanopore sequencing. Specifically, as the sequencing depth increased, the number of errors in the assembled consensus sequence is significantly reduced. However, we found that the accuracy of the methylation detection was unaffected by the sequencing depth beyond a certain threshold. Particularly, the accuracy of methylation detection under 200× sequencing depth does not exhibit any significant increase compared with that of 50× sequencing depth. Nevertheless, increasing sequencing depth reduced the variance in accuracy of the methylation.



Note Fig. N10. Two-site encoding for epi-bit DNA storage. (A) Diagram of the one site encoding scheme (left) vs. the two-site encoding scheme (right). **(B)** The sequencing result analysis pipeline of two-site coding. **(C)** Comparison of errors between the two-site (upper) and the one-site (lower) encoding schemes. The upper panel rows show the percentages of 00 and 11 reads at each CpG pairs and the lower panel shows the methylation probability at each CpG site, with errors indicated by the red boxes.

(2) Error correction and its impacts

Choice of the error correction code (ECC)

Despite errors occurred in the sequencing process, the collective errors in epi-bit storage can be corrected by error corrective codes. In this work, we used the BCH code for error correction in the large-scale storage experiment, which successfully eliminated all the errors to fully recovered the

stored data for the panda image (Fig. 5). Alternatively, we also managed to eliminate errors using a two-site encoding scheme (Note Fig. N10). In this scheme, the two epi-bit sites on a single brick were correlated in that we only used (0,0) and (1,1) to represent valid information (0 and 1, respectively), whereas (0,1) and (1,0) indicated errors. By deleting invalid two-site locations in each read, and vote across the remaining valid positions, we were able to restore the original information with 100% accuracy. Notably, unlike sequence-based DNA storage, epi-bit storage mainly causes substitution errors (e.g. 0 to 1 or 1 to 0). Therefore, the ECC should be highly efficient in dealing with substitutions.

For the sequence-based storage, two ECCs are the most commonly used: RS code and fountain code^{6,15}. However, we consider these two codes may not be very suitable for epi-bit storage. First, RS code is a non-binary linear block code, which requires stream conversion before it can be used for epi-bit storage. For example, if we use RS code based on Galois field $GF(2^8)$, then every 8 epi-bits need to be integrated to generate an integer from 0 to 255. In this case, any single error on these 8 epi-bits causes a distortion on the integer, which harms all 8 epi-bits. Second, Fountain code is an erasure code designed for an erase channel. Erasures correspond to insertion/deletion and oligo dropout errors, which are not the error type occurring in epi-bit storage. Therefore, we did not choose the above two error correction methods. Besides RS code and fountain code, we also considered other codes such as polar code¹⁶, which is relatively new and broadly used in communications (e.g., 5G communication). However, we found that the efficiency of polar code is inferior when compared to BCH code at a high code rate (the ratio of payload length to codeword length).

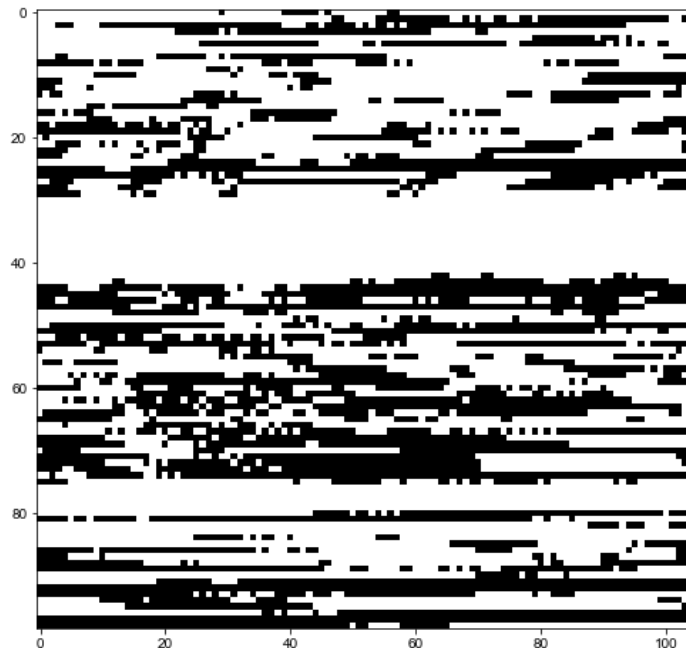
Impacts of error correction

Impacts on coding density: The coding density (bit/base) is supposed to be decreased when introducing error correction code. For example, when we stored the image of a tiger rubbing (Fig. 5b), two different data processing methods were used. The first method was direct storage without error correction code yielding a data size of 10,400 bits (Fig. 5b, sample1-3). In the second method, the bit stream was first compressed using run-length coding, and then the BCH error correction code was added, yielding a data size was of 11,750 bits, ~13% larger than that in the first method (Fig. 5b, sample-4). In addition, it is found that without error correction code, data underwent compression was non-retrievable due to error propagation. To demonstrate this phenomenon, an *in silico* simulation was performed to store the tiger rubbing image, where data was compressed

but not encoded with error correction code. As Note Fig. N11 indicates, after decoding, almost no useful information can be retrieved.

Impacts on sequencing depth: In the epi-bit storage, as long as the error correction code is sufficient for perfect data retrieval, the more error correction code is introduced, the less sequencing depth is required. This is because the sequencing depth provides for physical redundancy that serves the purpose of error correction by consensus voting. In other words, more sequencing depth naturally leads to less errors. However, physical redundancy is generally less efficient than logic redundancy (i.e. error corrective coding). Therefore, when logic redundancy is introduced by error correction code, physical redundancy is less required for data recovery^{17,18}.

Impacts on storage cost: The error correction code naturally leads to an increase in computational cost, although it is neglectable compared to the costs of synthesis and sequencing. Meanwhile, all the encoding and decoding processes can be finished within minutes, not requiring special or professional computing equipment. In addition, error correction code introduces logic redundancy to the data, and thereby increases the data writing cost in proportion to the logic redundancy. However, the sequencing cost will be reduced because less sequencing coverage is required for data recovery.



Note Fig. N11. Data retrieval simulation of the tiger rubbing. The original data was compressed without any error correction code added. Here, we used an error rate of epi-bits derived from our real nanopore sequencing results.

NOTE 4. Storage capacity and scaling potentials

In recent years, many DNA storage strategies have been published employing different forms of molecular information bits including DNA mutations, modifications, hairpin structures, DNA nicks and combinatorial short DNA bricks. While we listed the data volumes stored by different strategies so far in Table S1, here we also analyze the theoretical storage capacity for three representative approaches including *de novo* synthesis, epi-bit storage and combinatorial storage⁷.

For *de novo* synthesis and the epi-bit system, the storage capacity scales linearly with the total length of DNA molecules $C = \rho(l \times N)$, where ρ is the coding density, l is the length of each DNA strand, and N is the number of different DNA strands. Since the DNA molecules are individually addressable by barcoding, there can be an unlimited number of molecules ($N \rightarrow \infty$) in one storage system. Therefore, the storage capacity is in principle unlimited.

For the combinatorial storage strategy⁷, the storage capacity is determined by the Layer arrangement of the Components, i.e. the capacity in one storage system is given by $C = \prod_{i=1}^N M_i$, where N is the number of Layers and M_i is the number of components in the i -th Layer. Because data is represented in un-ligated identifiers which are not individually addressed, one storage pool cannot be mixed with another. Therefore, the storage capacity is limited to C .

For a more specific comparison, Roquet *et al* proposed that 110 Components partitioned to 15 Layers of 5 Components each and one Layer of 35 Components would generate a storage capacity of 1.07 Terabits⁷. And the storage system is composed of $0-10^{12}$ different molecules depending on the stored information. For *de novo* synthesis and the epi-bit framework, the same storage capacity can be achievable by a pool of 10^9 independently addressable 200 nt DNA oligos or by a pool of 10^{10} addressable 1300 nt DNA templates, respectively.

In general, the storage capacity grows linearly with the number of available barcodes. Although the entire barcode space grows exponentially with the barcode length, the concern of barcode cross-talk and off-target effects require that the barcodes are different from each other by a certain Levinshtein distance, which will eventually cap the total number of usable barcodes even at long barcode lengths. There have been a handful of works contributed to barcode design for DNA storage. These works have invoked primer design principles and tree structures for the encoding of large-scale data while maintaining a relatively high logic density. With these specific barcode designs, the projected storage capacity has reached $\sim 2^{180}$ bytes¹⁹. The epi-bit storage framework is compatible with these barcoding designs. On top of these strategies, the epi-bit framework has the

additional option of using epi-bit sites for barcoding. The area of primer design for DNA storage will continue to develop with the collective effort from the research community.

Additionally, in practical terms, random access puts a limit on how much DNA data can be reliably read out from one dissolved state reaction system. Specifically, there should be at least 10 copies for each different molecule²⁰ and the overall reaction system needs to be in a dilute state²¹. These constraints on storage capacity apply to all three DNA storage systems regardless of the underlying mechanism.

Supplementary Figures

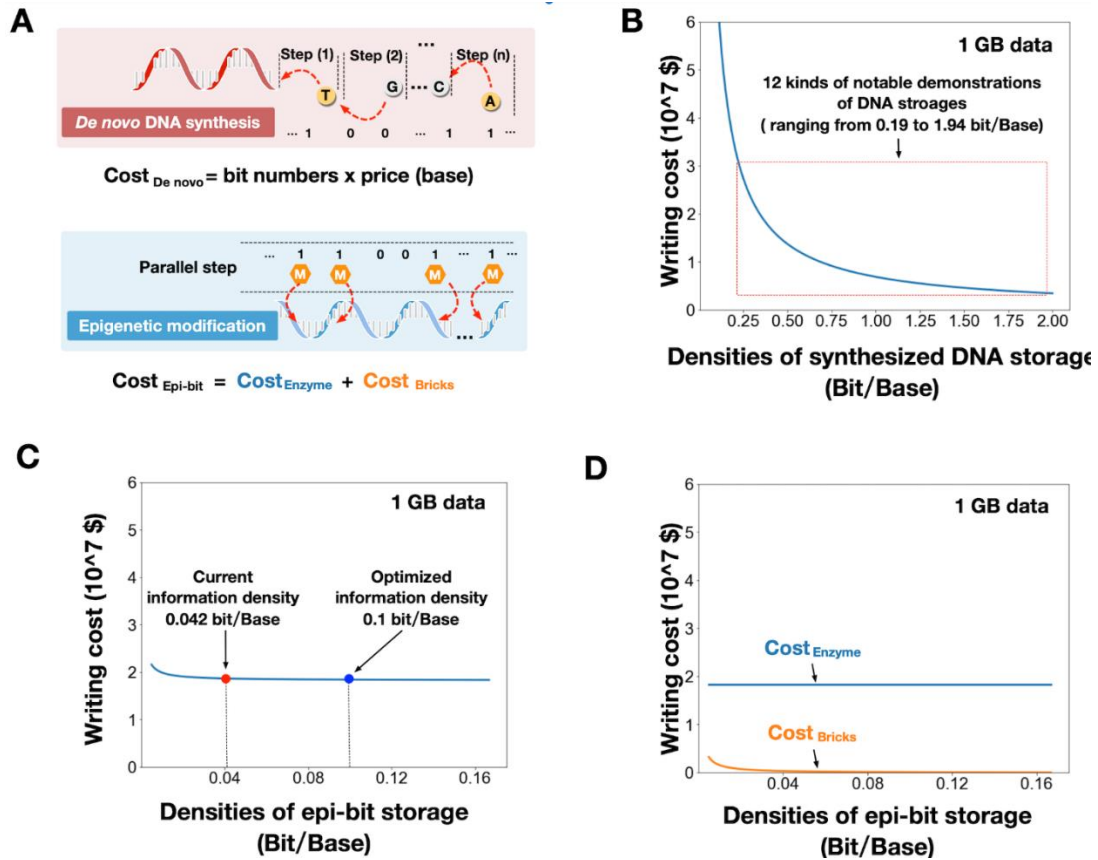


Fig. S1. Comparisons of the writing cost between *de novo* DNA synthesis and epi-bit storage. (A) schematics of data writing by *de novo* synthesis and epi-bit storage. The density-cost relationships for data storage by *de novo* synthesis (B)* and epi-bit writing (C)**. (D) The costs of DNA bricks and enzymes in epi-bit storage**.

Remark 1: Cost of *de novo* synthesis is calculated based on the DNA synthesis price in ref (3) in the main text, which is $\sim \$7 \times 10^{-4}/\text{nt}$.

Remark 2: Cost of epi-bit writing is calculated based on our experimental protocols, empirical estimates and enzyme sales prices offered in mainland China. Let ρ be the information density (bit/base), and the breakdown is as follows:

(i) Cost of templates and bricks (fixed overhead): let l be the length of bricks, x be the number of epi-bits per brick, n be the number of bricks per template and k be the number of different templates. The cost includes: synthesis of long templates nlk nt, synthesis of bricks nlk nt, and modification of $nkx \cdot 2^{x-1}$ sites. We use $l = x/\rho$ nt, $x = 2$, $n = 35$, and $k = 5$. The modification cost is \$11.632/site, and synthesis of long DNA and short oligos cost \$0.1454/nt and \$0.1018/nt, respectively.

(ii) Cost of writing (variable cost): the major costs are from T4 ligase, DNMT1, and S-adenosylmethionine. The per bit cost = $P / (T \times C)$, where P is the enzyme price, C is the bit throughput, and T is the number of reactions the purchased enzyme can be used. We used 10 μ L reaction systems with a bit throughput $C = 350$ bits. $P_{T4} = \$290.8$, $T_{T4} = 800$, $P_{DNMT1} = \$2617.2$, $T_{DNMT1} = 3125$, $P_{SAM} = \$63.39$, $T_{SAM} = 20000$. Therefore, the cost of enzyme = \$0.00344/bit.

Remark 3: The storage density in our experiments (0.10 bit/base in Fig.5) was relatively low compared to mainstream strategies based on oligo pools. It was ~20- to ~10-fold lower than the average density of 1.08 bit/base reported for traditional DNA storage (calculated from 12 notable demonstrations of *in vitro* DNA storage, Table S1. Actually, it should be noted that the density of 0.1 bit/base in the current epi-bit storage is already four to five orders of magnitude higher than the densest storage media available today. In fact, the coding density is important because it strongly affects the cost of data writing by the *de novo* synthesis strategy. However, as our system employs a fundamentally different data writing strategy, this factor needs to be re-evaluated from a different perspective. In the *de novo* synthesis framework, because the cost grows with the number of nucleotides synthesized, dense encoding of data requires less nucleotides to be synthesized which directly saves on the writing cost. Fig.S4B indicates the per bit writing cost of *de novo* synthesis is inversely related to the storage density: a density increase from 0.19 to 1.94 bit/base causes a 10-fold cost reduction. Nevertheless, by estimating the per bit writing cost of the epi-bit storage, the information density is not a critical factor for the parallel epi-bit writing system and the writing cost is insensitive to information density (Fig.S4C). It is demonstrated by estimating the cost of storing 1GB data by *de novo* synthesis versus by our epi-bit system (Fig. S4 B and C). Currently with lab-scale experimentation and commercial enzyme prices, epi-bit writing

costs ~\$ 0.003 /bit regardless of the information density (Fig. S4C). This cost is already within close margin to the cost of DNA storages by *de novo* synthesis, which is \$0.001 / bit at the maximal 2 bit/base density.

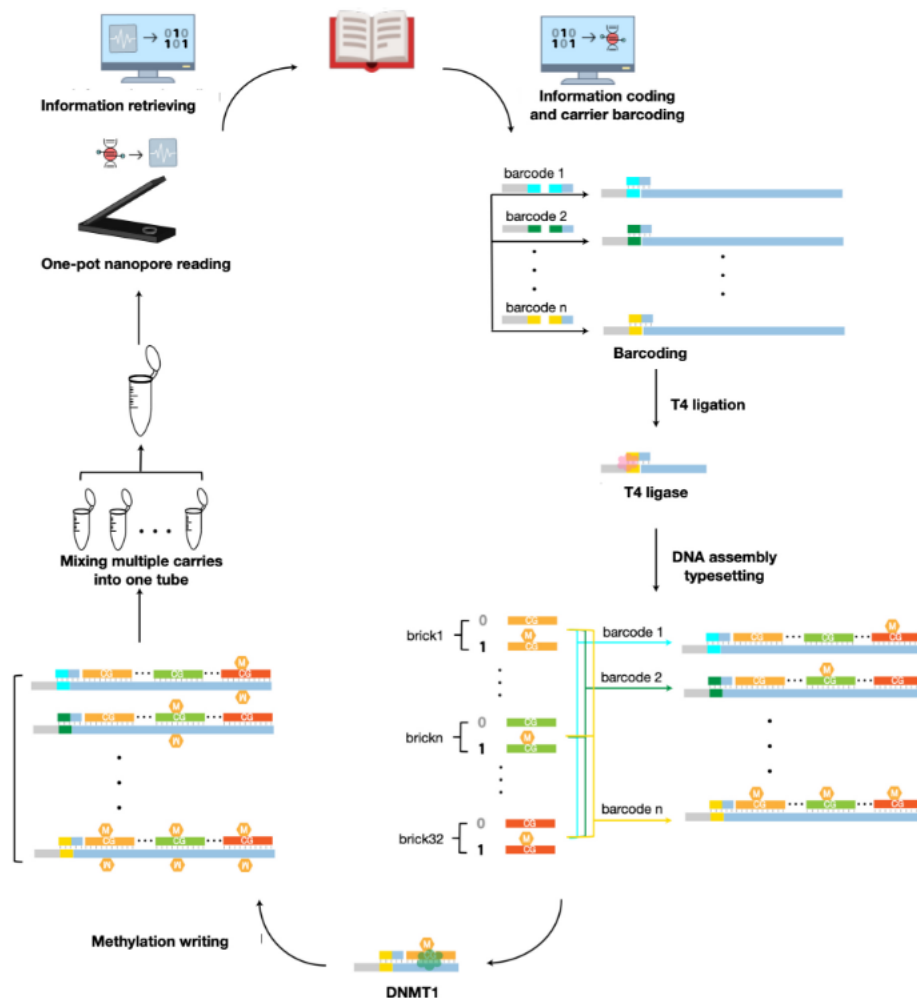


Fig. S2. The illustration of the main operation steps of the parallel DNA storage with epigenetic bits. The whole storage pipeline consisted of the steps of information encoding, programmable typesetting, methylation writing and nanopore reading. First, the digital information is transcribed into binary bit strings. Second, the information is written onto the DNA carriers by programmable methyl modifications via a methyltransferase. Next, the stored modification information is detected by one-pot nanopore sequencing. Finally, the information is restored *in silico*. The whole approach is write once read many, and thus suited for random access to maintain the integrity of the DNA storage archive.

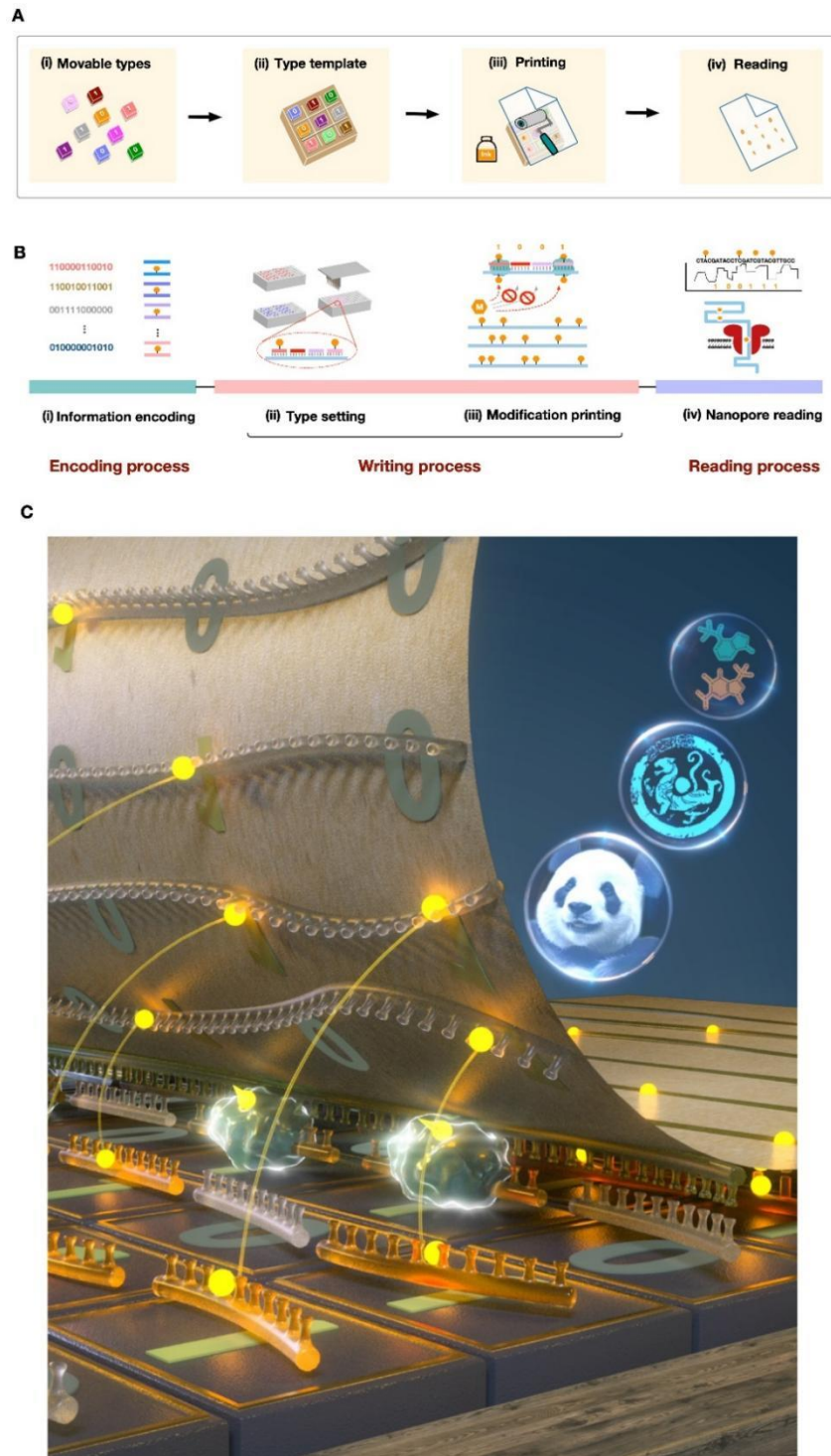


Fig. S3. The Comparison between traditional movable-type printing and the parallel epi-bit DNA storage. The comparison between traditional movable-type printing (**A**) and the parallel epi-bit DNA storage (**B**). (**C**) A cartoon picture of the parallel epi-bit DNA storage.

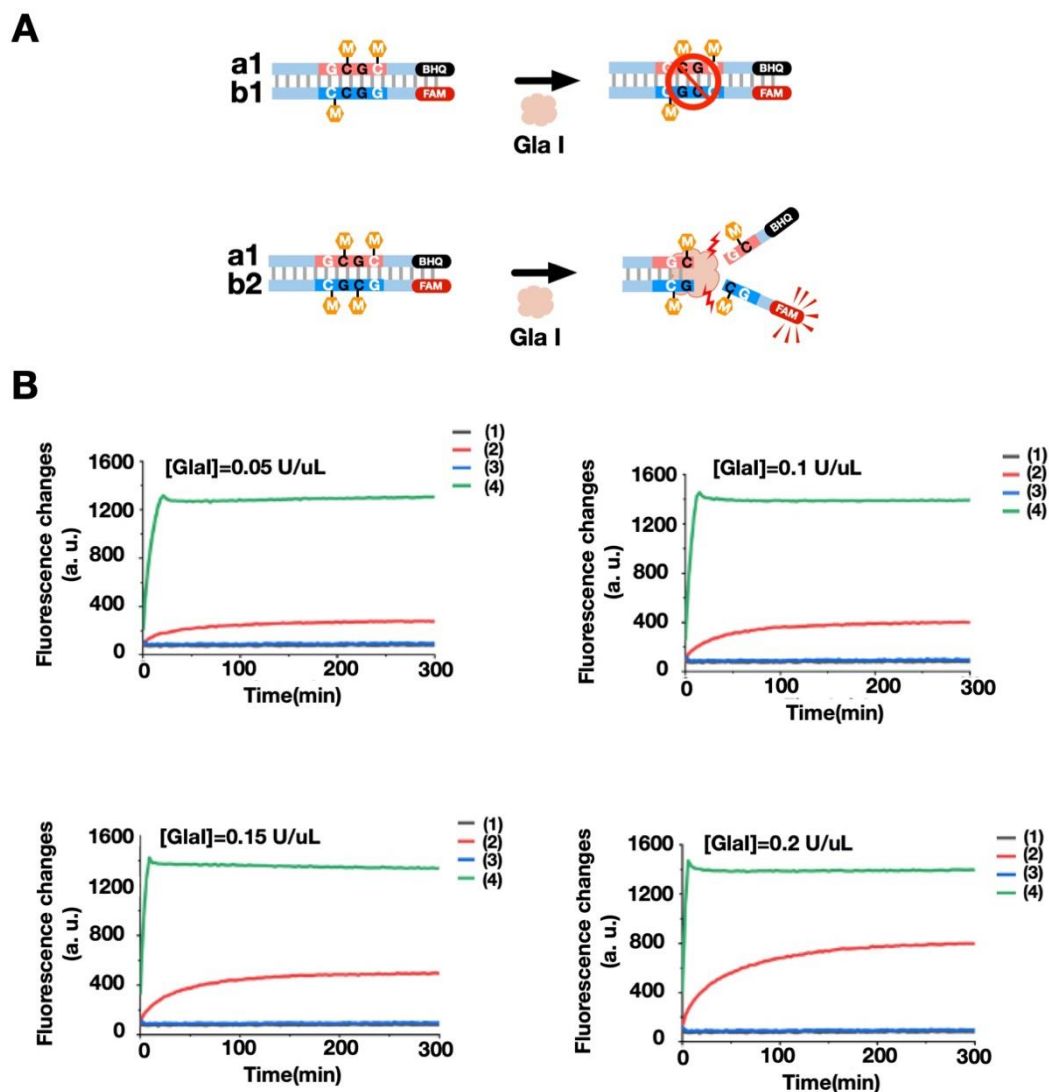


Fig. S4. The illustrations and results of the methylation detection assays. To verify the feasibility of the methylation-based data storage, a digestion assisted method was established with the restriction endonuclease *GlaI*, which only cleaves C5-methylated DNA within a four-fold methylated GCGC site (**A**)^{22, 23}. (**B**) The fluorescent assays of varying concentrations of *GlaI* at 0.05, 0.1, 0.15 and 0.2 U/uL. Line 1: a1b1; Line 2: a1b1+*GlaI*; Line 3: a1b2; Line 4: a1b2+*GlaI* (b2 has the same sequences with b1 modified with 2 methyl groups).

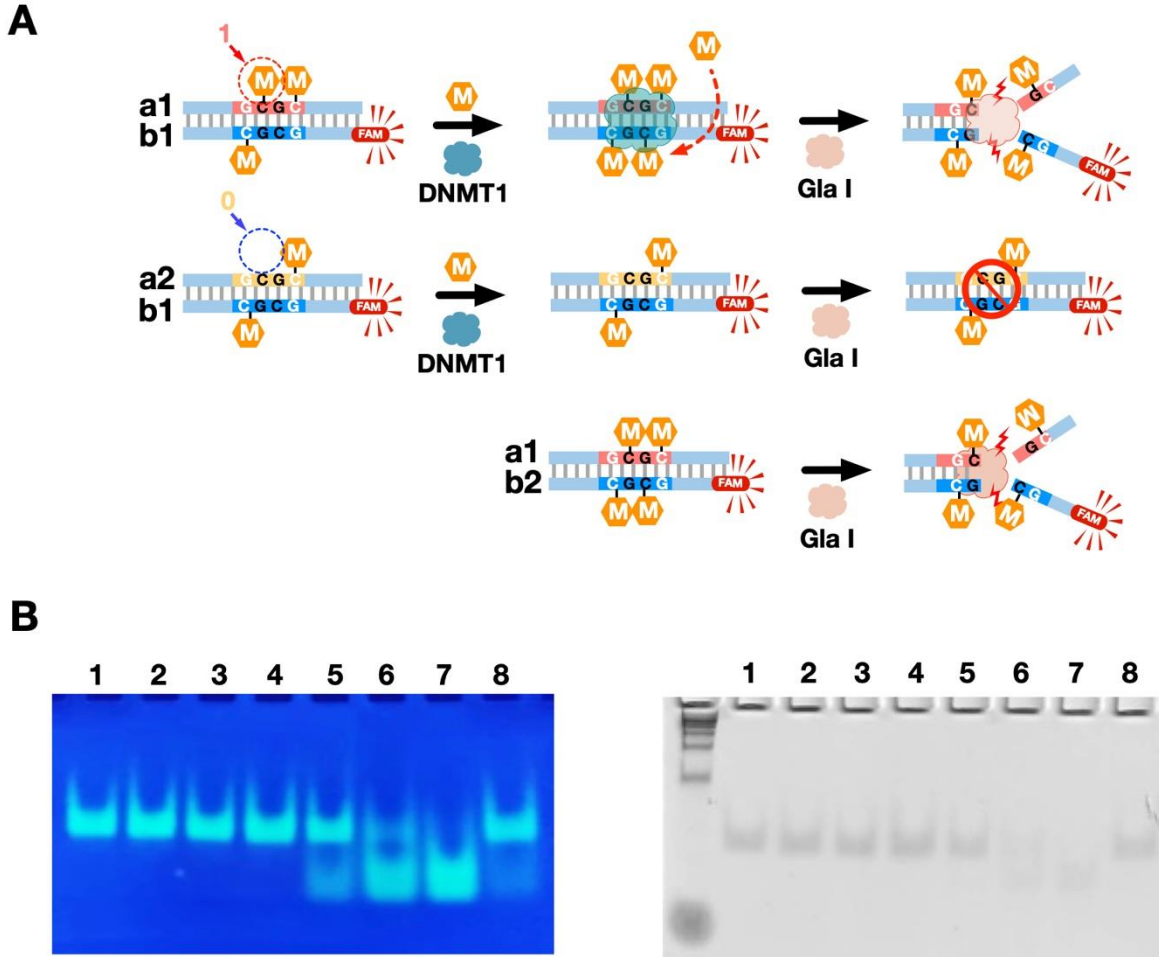


Fig. S5. The illustrations and gel results of the basic methylation writing tests on a single epi-bit site. (A) The digestion test design. The methyl writing of epi-bit value 1 was implemented by using a methylated DNA duplex in a predesigned position (indicated by a red arrow). Catalyzed by DNMT1, a methyl group will be transferred from SAM onto the cytosine in close proximity to the nicking site on DNA B1. While inputting the initial epi-bit value 0 at the designated position (indicated by a blue arrow), GlaI cannot digest the DNA complex on the recognition site. (B) In the gel results detected by UV light and Stains-All dye. Lane 1: a1b1; Lane 2: a1b2; Lane 3: a2b1; Lane 4: a1b1+DNMT1; Lane 5: a1b1+GlaI; Lane 6: a1b1+DNMT1+GlaI; Lane 7: a1b2+GlaI; Lane 8: a2b1+DNMT1+GlaI (a1 and a2 used in the gel experiments are unmodified and b2 has the same sequences with b1 modified with two methyl groups at CpG sites).

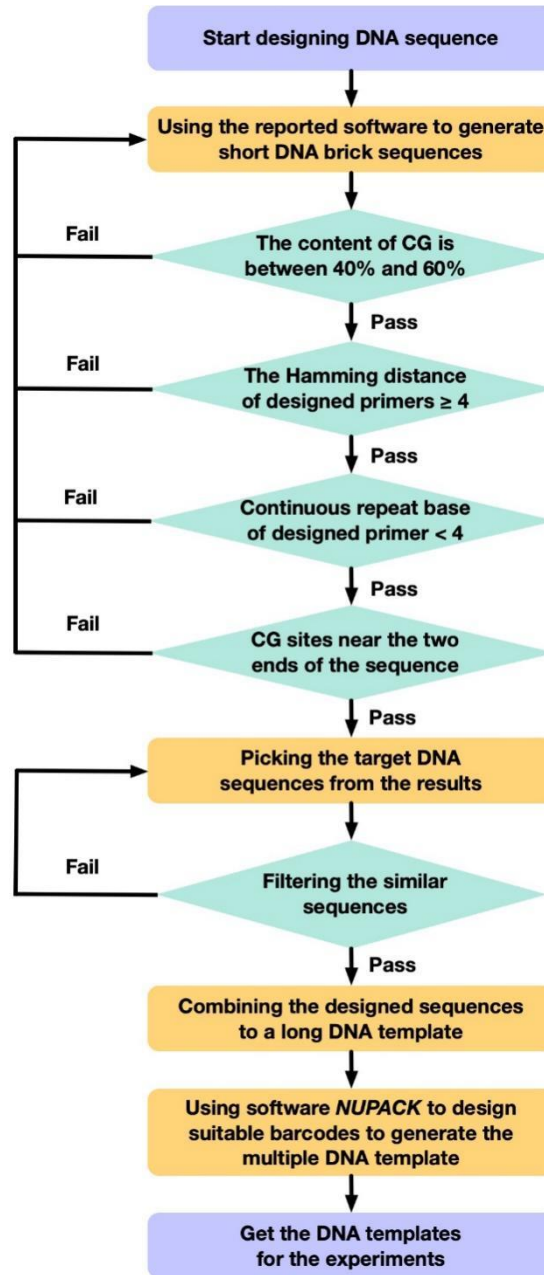
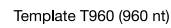
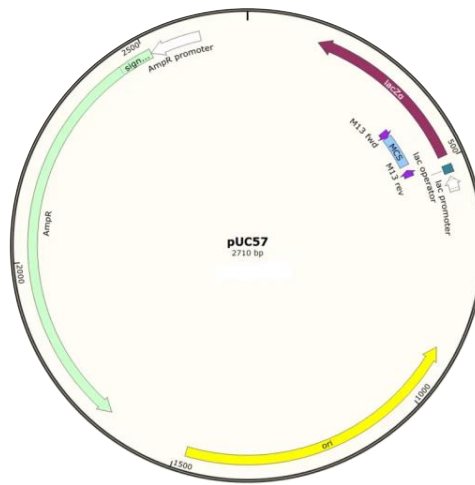


Fig. S6. The flowchart of the design processes of the DNA bricks and carrier templates.

Firstly, a series of target DNA bricks were generated²⁴. To minimize the secondary structures, several standards should be met: 1) dinucleotides of CpG are designed in the desired position serving as potential epi-bit sites within each DNA brick; 2) the CG content should be between 40% and 60%; 3) the numbers of the nucleotide repeats should be less than 4; 4) the Hamming distance between the designed primers should be ≥ 4 . Then, we performed mismatch analysis among DNA bricks and templates.



33



Multiple clone sites

Template T1200 (1200 nt)

3'

GTGTGTCCTCATGCTAATAGCGACATCCCACTACAAGCGAAGTAGTGCTGAGACGACTACCTCTAGGAAGAGAGCTCACCGCCTAAAGTGATTATTCGCCATTGGTCAATTTCACGCTGATCAACGGTCT
CACACAGGAATACGATTATCGCTGTAGGTGATGTCGCTTCATCACCAGACGTGCTGATAGGAGATCCCTCTCTCGAGTGGCGAGTTCAACATAATAGCGGTAAACGATAAACGTGCACTAGTGGCAAG

CAGTTATAGTGCAGCTCATCCACAAGATTGACGCGACATGCATACAATATTGGCTCGATAGGGCCACCAAGCGAATACGTTGTGGTACTAGTACAATTTGACGCCACTGCTCTACCACTTGAACGATGA
GTCAATATCAGCTGCAGTAGGTGTCTAACTCGCGCTGTACGATGTGTTAATAACGAGCATACCCGGTGGTCCGCTATGACCAACCACTGATCATGTAACTGCGGTGCAGAGATGGTGAACCTGCTACT

GGGTGACGCAACATACGCGCTGATTTGGTGAATCTTACGCTGTACAGAGCTGCTCATACGCAATATAGAGGTGCACCAACGAGATATGTTGTGTAGTACGCTGCAATCTGCTCGGTACGAT
CCGCTGCGGTGTACTTACGCGCTACAACCACTGAGAGATTGCGACAGTCTGTCAGACAGGATGCGCTTAACTCTCCAGGTTGTCGGGCAATCAACACTGATGCTTAAAGTAAAGCTAGCTATGAC

CTGCGACAGGTGAACTGATCTCTCGCTTAATAAGAGGTGCTAAGCGCTTAATAAGAGTAGTGCTGCTCGTGAAGTGAACATGCCACGCGCAACCAATGATTGCTGTAAAGTGCCTACACTCA
GAGCGTGCTCCATCTGACTAGAGAGCGGATATTACTCCAGATTTCGGAATTACTTCCATCAGACAGGACTCTTTACTGTGACCGGTGGTGGTAACTAACGACATTGACGAGATGTGAGT

GTAGCTTCAACGACCAACCTAGTTCGAAGGCGCTTGACCAACAGTAGTGTGTACGCTCACCAATAAGTGGCGTGATGACCACCAAGATGTTGCTGCTACTGCGAAGGTGTTCCGAAATGTT
CATCGAAGTGTGCTGGTTGTGATCAAAAGTCTCGCAACTGGTGTCTAACACATCGAAGTGGTATTACACGCGCATAGTGGTGGTTCTACACAGACGATAGCGCTTCACACAGGCTTAACA

CTCGCTCATTAATTACGAGAGCTGAAACGCACTGACTCGTAGACTGGTACCGCTGATGCGAATGTGCCACTGCGATAAATCGCTAAGTCCACTAGCTGCGGACGAATTTTCTCCATCGCA
GAGCGGAGTATTAACTGCTCGGACTGCGGTACTGAGACTCTGACCATTGGCGAGTAACGGTTACACGGTGTACGCTATTATGAGCATTCAAGTGATGCGACGCTGCTTAATAAAGGAGTAGCT

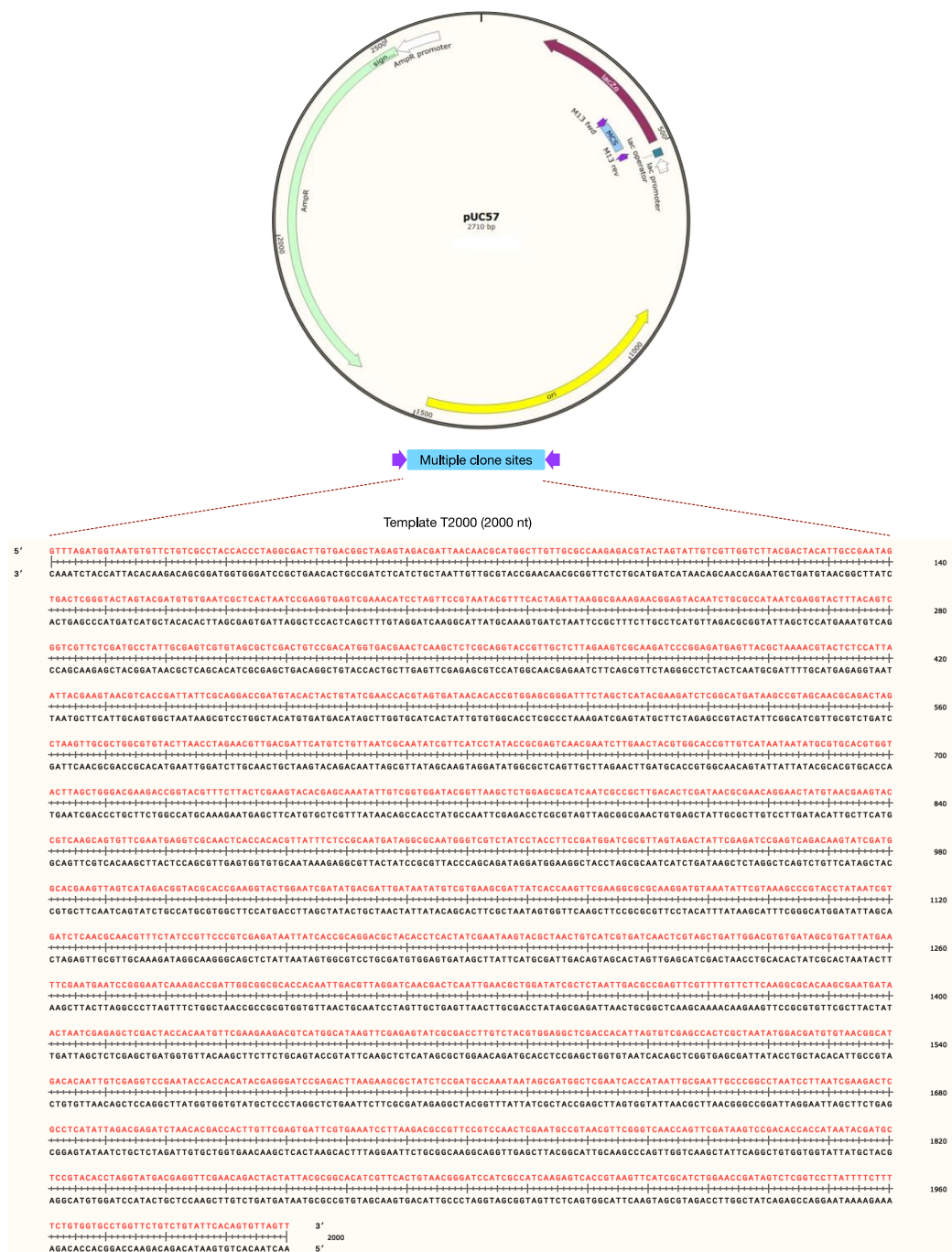
CTCGGAATAAGACAATCATCGCTGACGGTGAAGTGACTGAATTGCTGACGCCACTGATGTTGACCCAGCGCTGTAATAGTGGTGGCAATACTCGCATAACCGCGACGCGACTAGACGACTGAAGCG
GAGCCTTATCTGTGTAGTACGCGACTGCCAATCCACTGACTTAAGACTGCGTAGCATCAAGTGGTGGTAGCATATCACCACGGTTATGAGCGGATTGGCGCTGCTGATCTGCTGACTTCGC

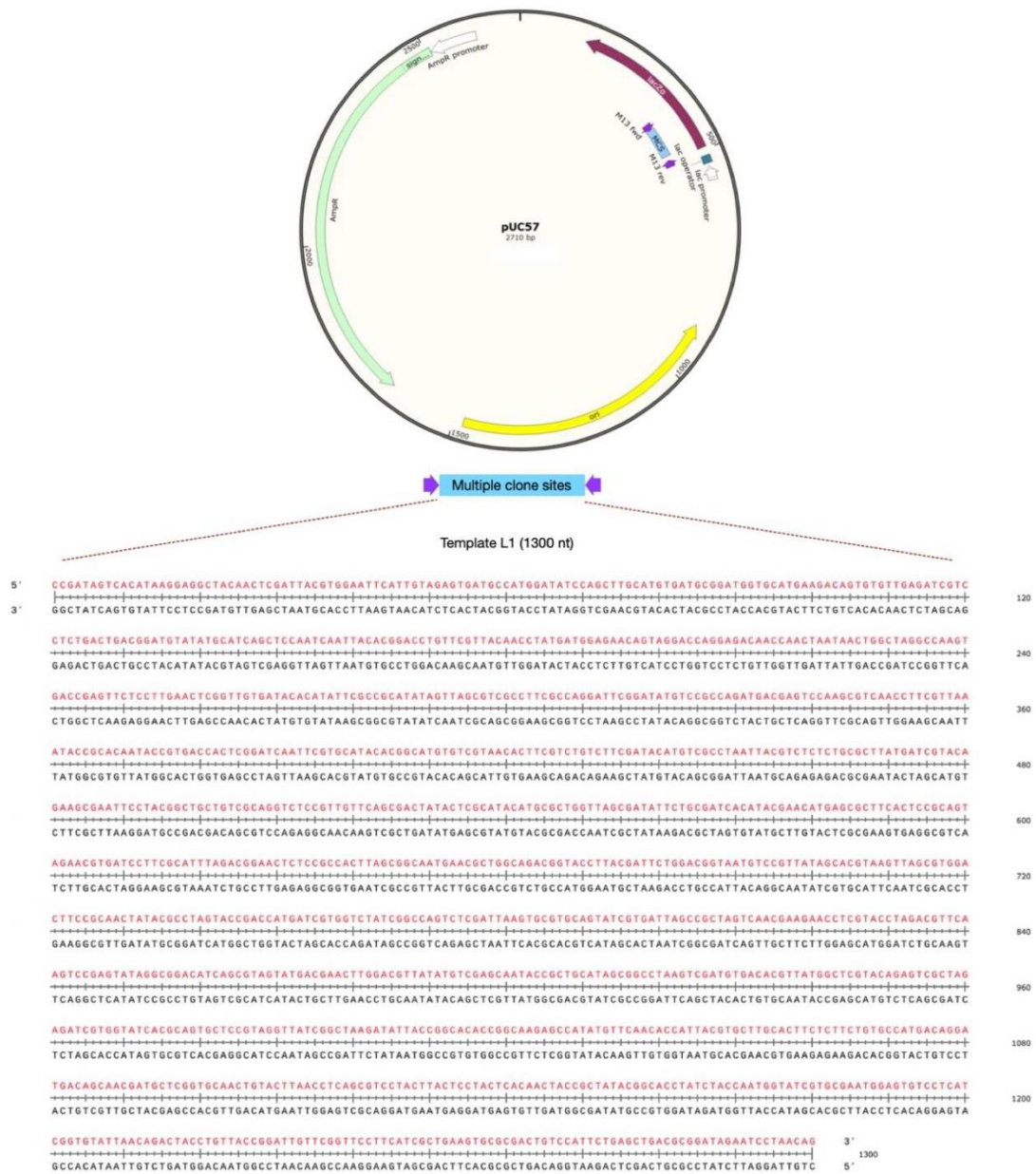
AAAGTACGTGCTGCTGTTGTGCGGTACCATGATTACAGATAGCGCTGGATCTTGGCGCCGTAACGAGCAACTACACGATACGCTGCTGACTCAAGCTGACGCGCTGCTGGATGTTCACTAGAG
TTTGTATGACGACAGAGACCGATGCTAGTAACTGAGTGTCTCCGCACTAGAACCGGGGCAATGCTGTTGATGTGTGATGCGGACGATGAGTGGAGTCCGAGCTGCAAGTGAATCTG

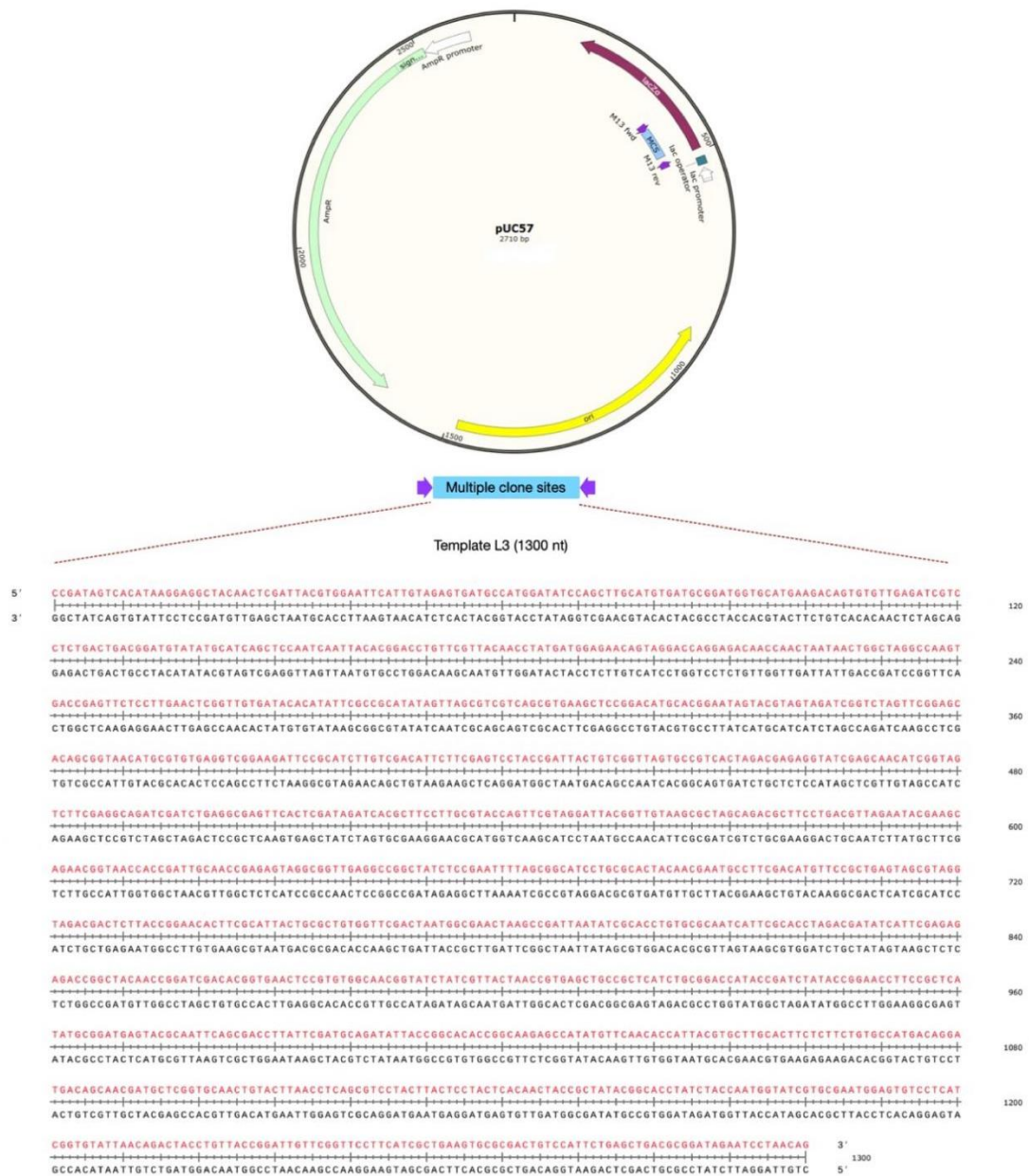
CGCCGCGCTAGTAAATCGGACGCTGCGACAGCTAATGCTAGTACGCGTACGCGCTGATTCGGAACACTCCAGAGATTTAGCATGTGTTGCGAGAGAGCTGCAATGTTGATGCTTCGCGCGAGGCTGCTA
GCCGCGCATCAATAAGCGCTGAGCGTGTGATTAAGCTAGGCCGATGCCGAGCTAAGGCTGTTGAGGTGCTCTAATCGCTACAAAGCTGCTGGAGTTTACCATCGAAGCGCTACGAGCAT

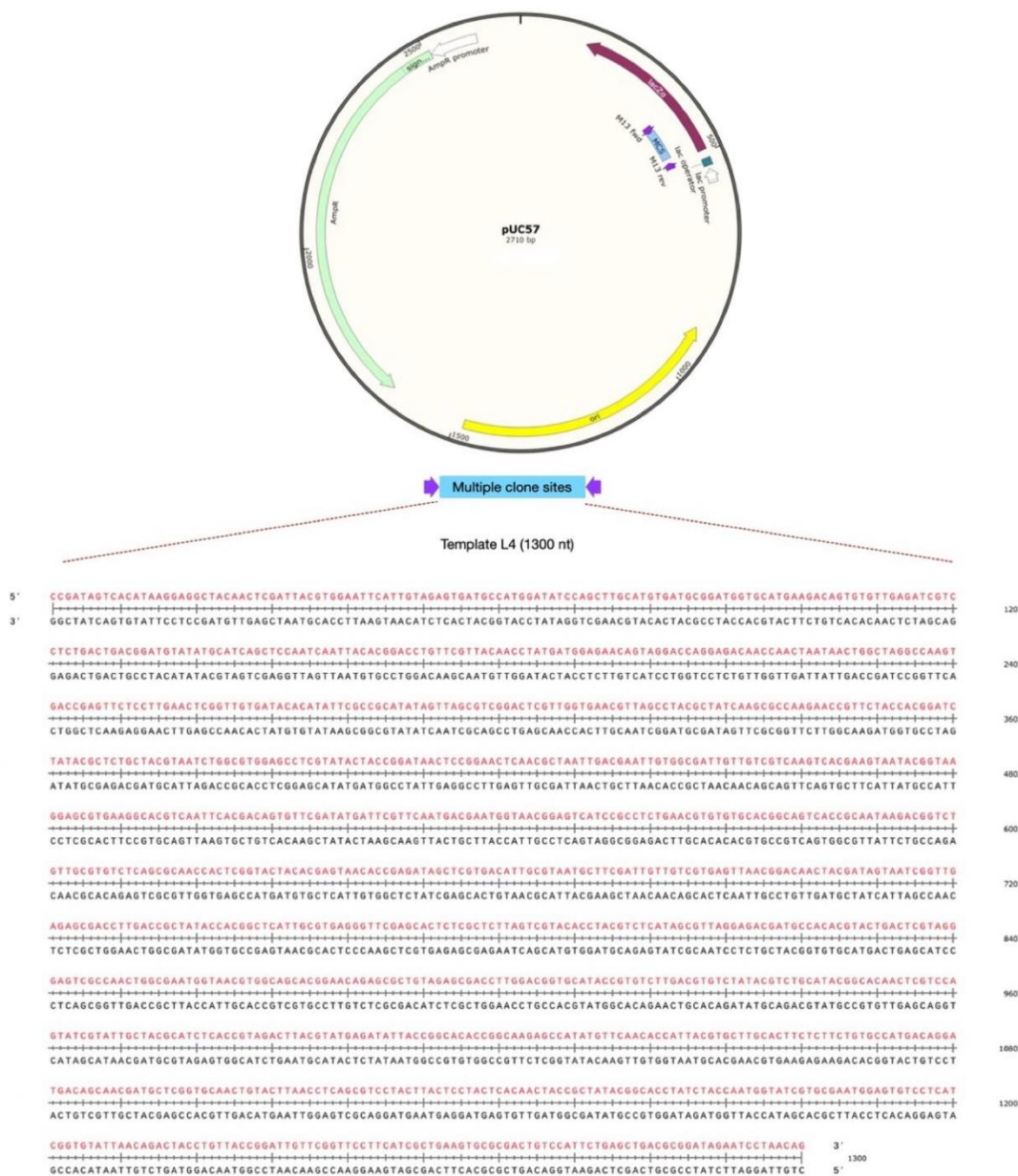
TAAGAGACTACTTATTTCGATGCGCGAG
ATTCTCTGATTGAATAAGCTACGCCGCTC

3'
1200
5'









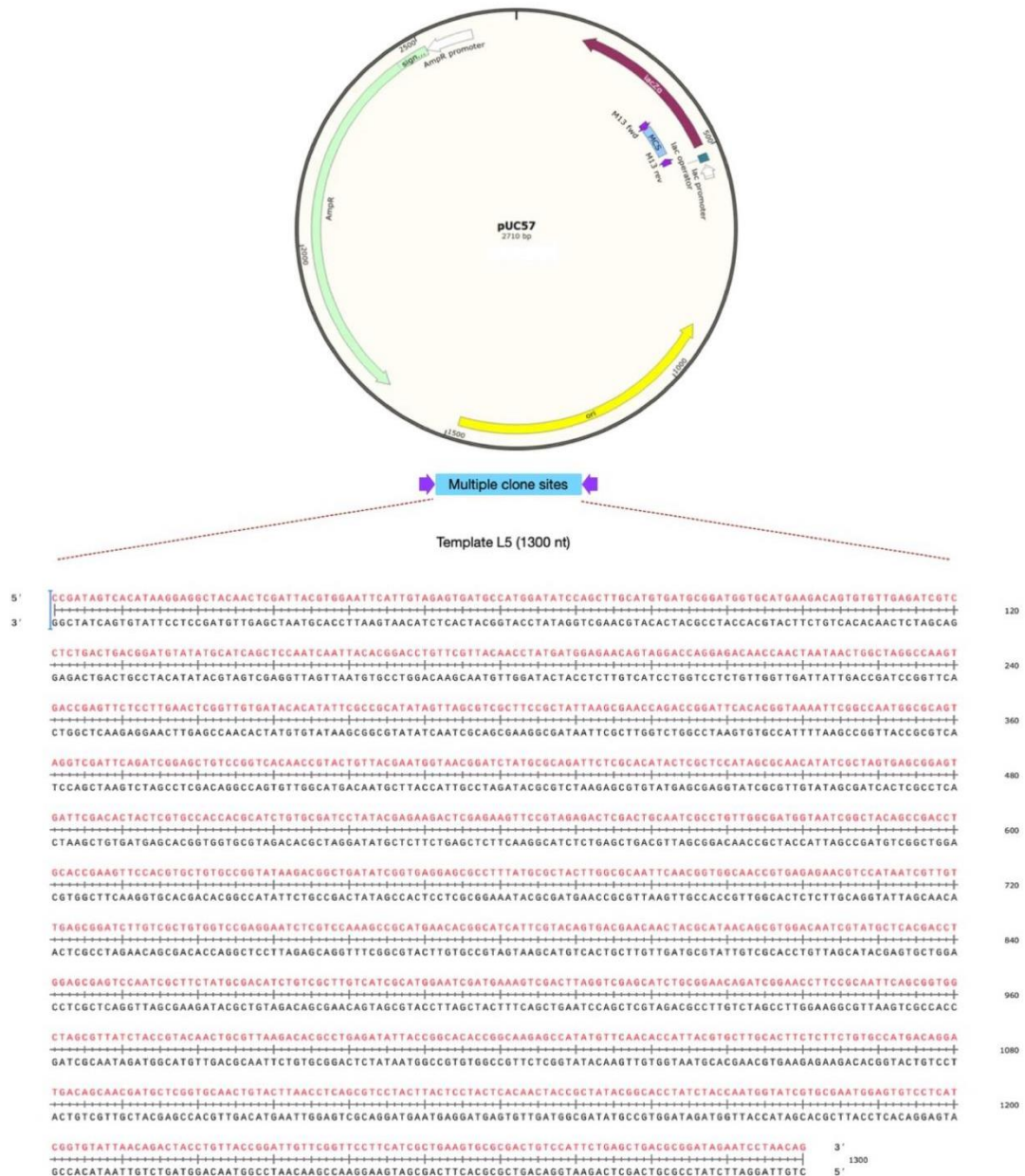


Fig. S7. The sequences of DNA carriers T960, T1200, T2000 with the lengths as 960 nt, 1200 nt and 2000 nt, respectively, and DNA carriers L1, L2, L3, L4 and L5 with the lengths as 1300 nt. The DNA sequences were artificially synthesized and individually inserted into the vector pUC57 for preservation.

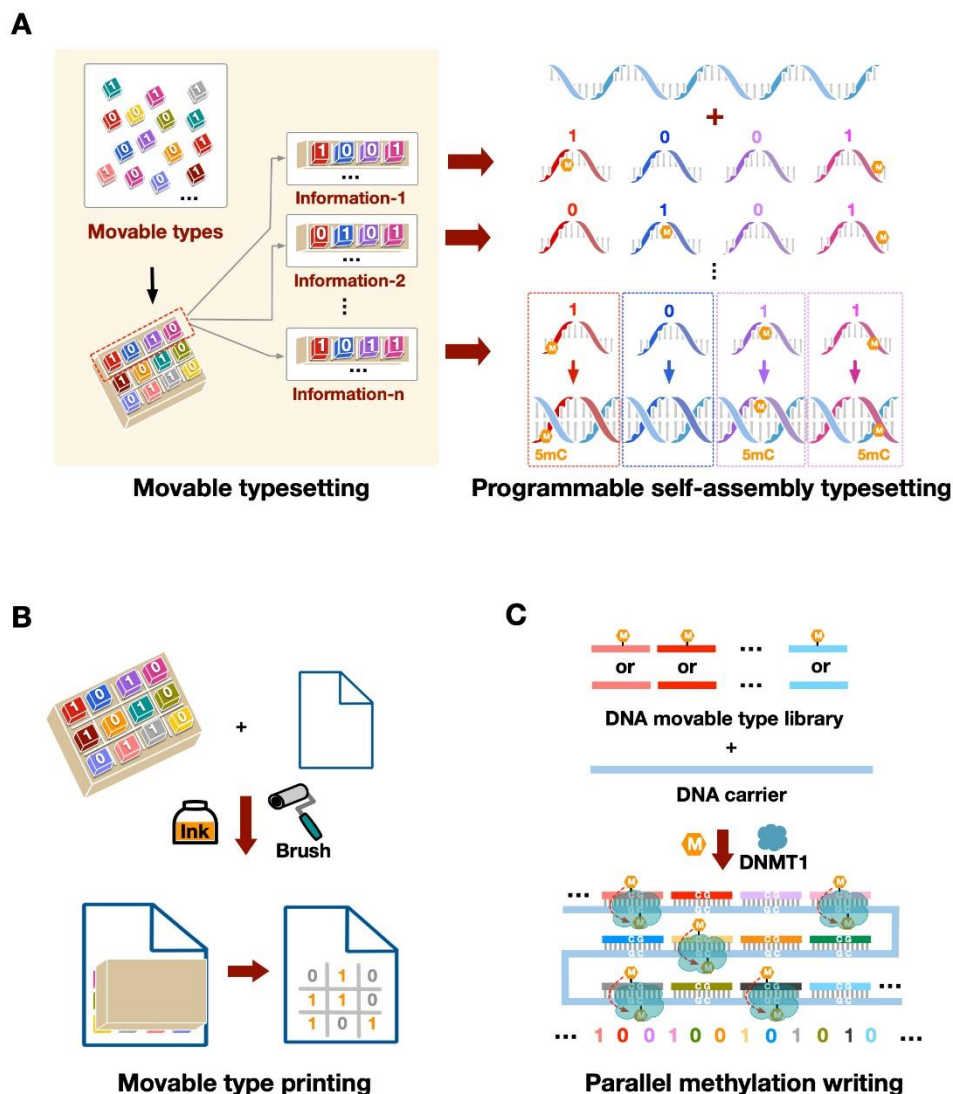


Fig. S8. The illustrations of the parallel epi-bit writing by DNA self-assembly typesetting and methyltransferase assisted enzymatic methyl transfer. Similar to the movable-type printing, the modifications in the predesigned methylation patterns are transferred onto long ssDNA carrier with the help of the methyltransferase. By mimicking the inheritance of DNA methylation patterns in nature, DNMT1, the maintenance methyltransferase, is used as a tool enzyme to catalyze the cytosine methylation of the hemi-methylated brick-carrier complex. **(A)** The process of a typical movable-type writing. **(B)** The principle of the traditional movable-type printing in information storage. **(C)** The illustrations of parallel epi-bit writing by the methyltransferase.

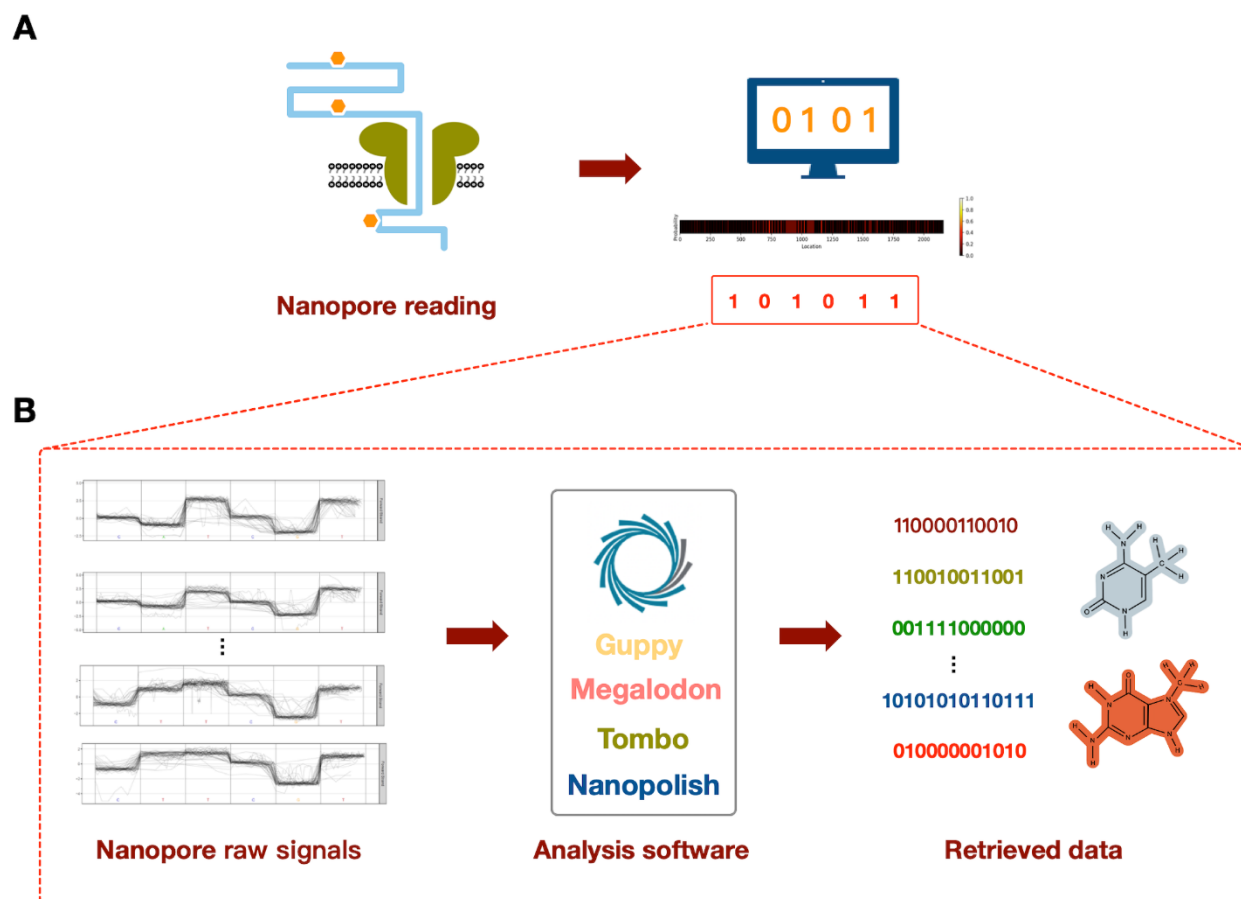


Fig. S9. Illustrations of the methylation information retrieval from nanopore sequencing reads. (A) The signal analysis in the main processes of epi-bit storage. (B) The processes of the methylation information retrieval from the raw signals of nanopore sequencing, where the software Guppy, Megalodon, Tombo²⁵ and Nanopolish²⁶ were used in the experiments.

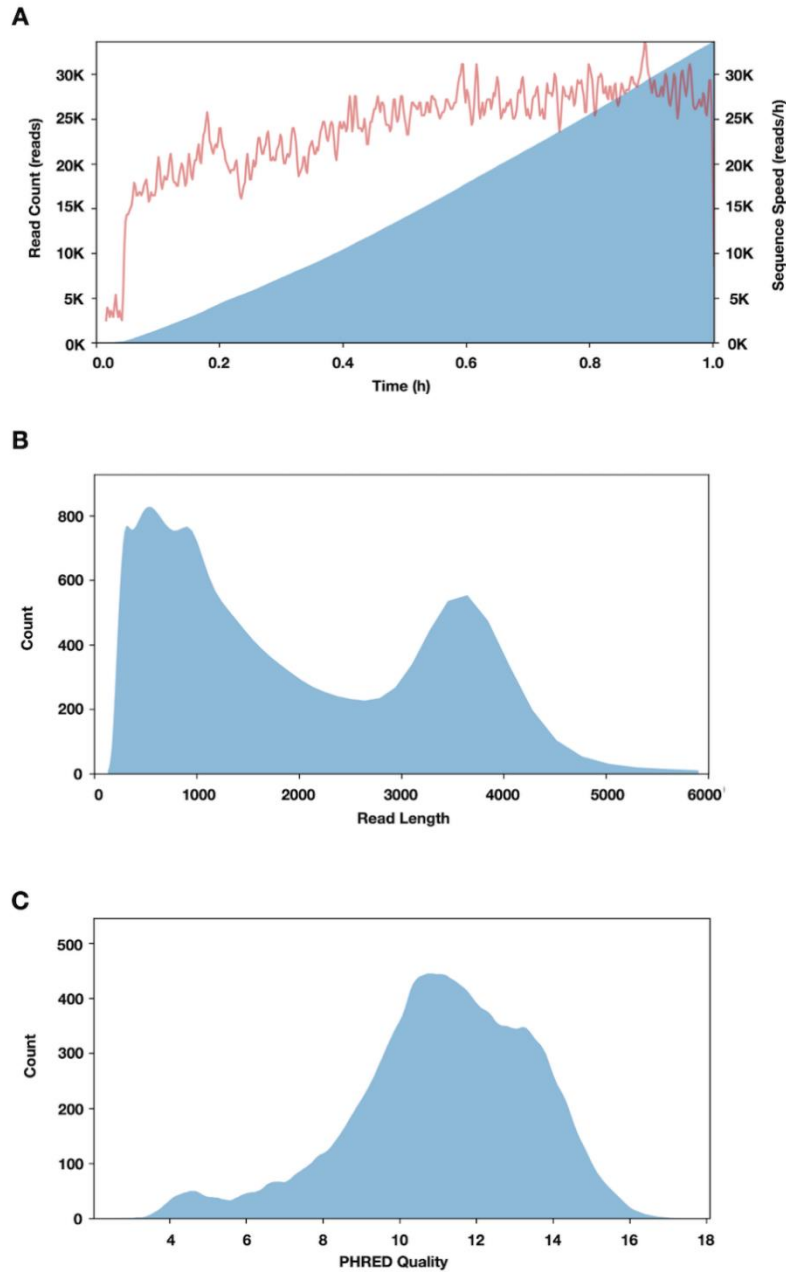


Fig. S10. Overview of nanopore sequencing results of the stored text “DNA”. (A) Real-time sequencing speed and cumulative outputs during sequencing. The sequencing time was 1 hour. (B) The distributions of base-called read lengths. Two peaks were observed. The first peak corresponds to the sequencing results of the carriers T960 and T1200. The second peak is due to carrier ligation by the T4 ligase in the ligation sequencing kit. (C) The distributions of base-called reads PHRED quality. The median sequencing PHRED quality is 11.63. The sample was a mixture of T960 carries and T1200 carriers, in which a three-letter text “DNA” was stored.

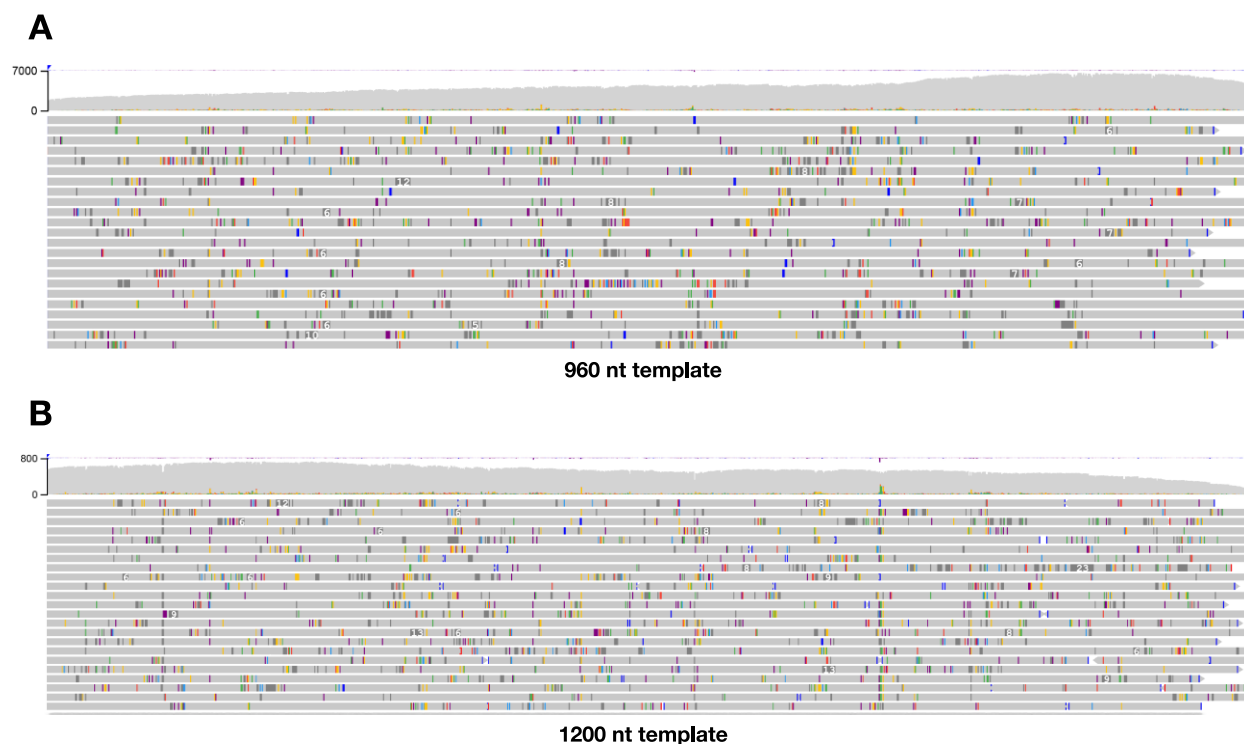
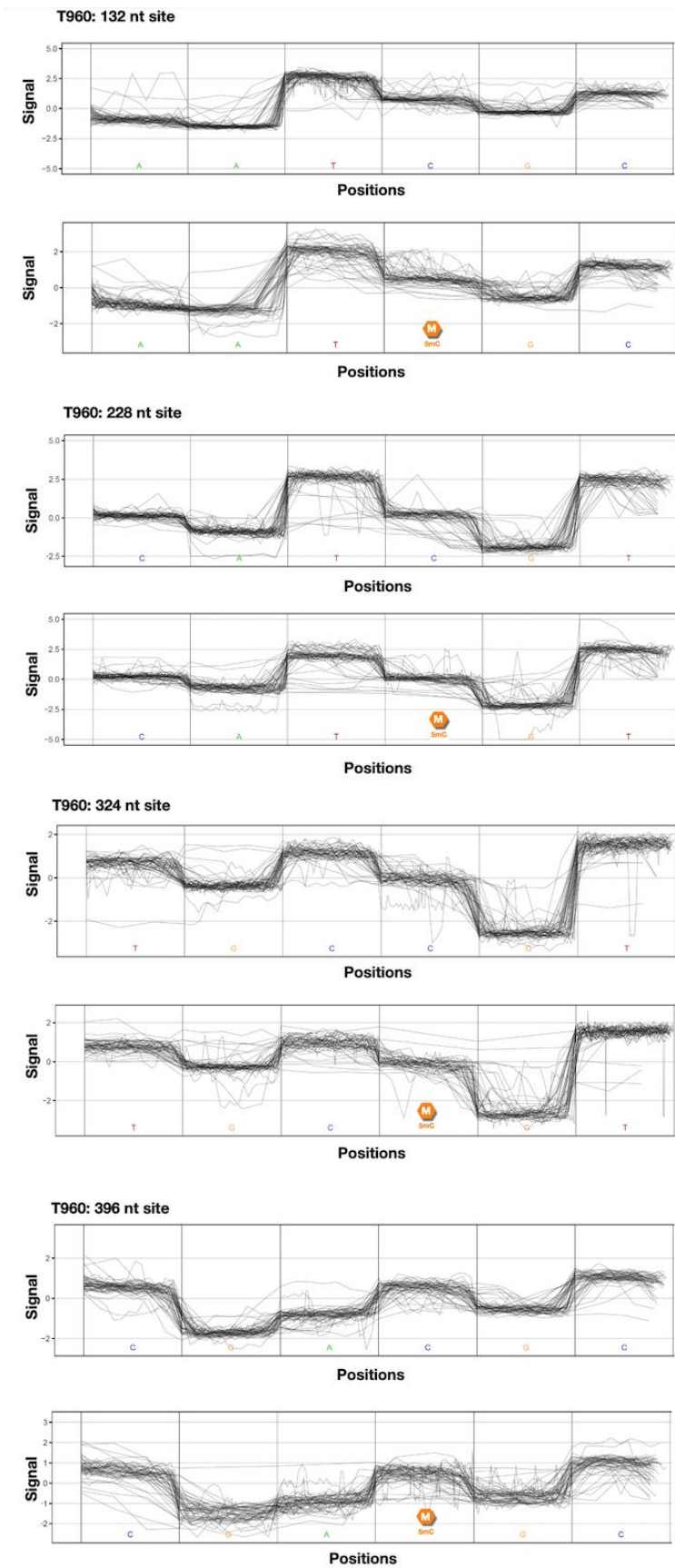


Fig. S11. Alignment of basecalled reads of the DNA carriers. (A) the T960 carrier. (B) the T1200 carrier. The first panel in each sub-figure shows the per-site sequencing coverage over the full-length template. Color codes in the alignment panels are: (1) mismatches of A: green, G: yellow, C: blue, T: orange; (2) insertions: purple; (3) deletions: dark grey; (4) gaps: black spaces. For the T960 reference sequence, the coverage at the 5' end was lower than that at the 3' end, and conversely, the coverage at the 3' end of the T1200 reference sequence was lower.



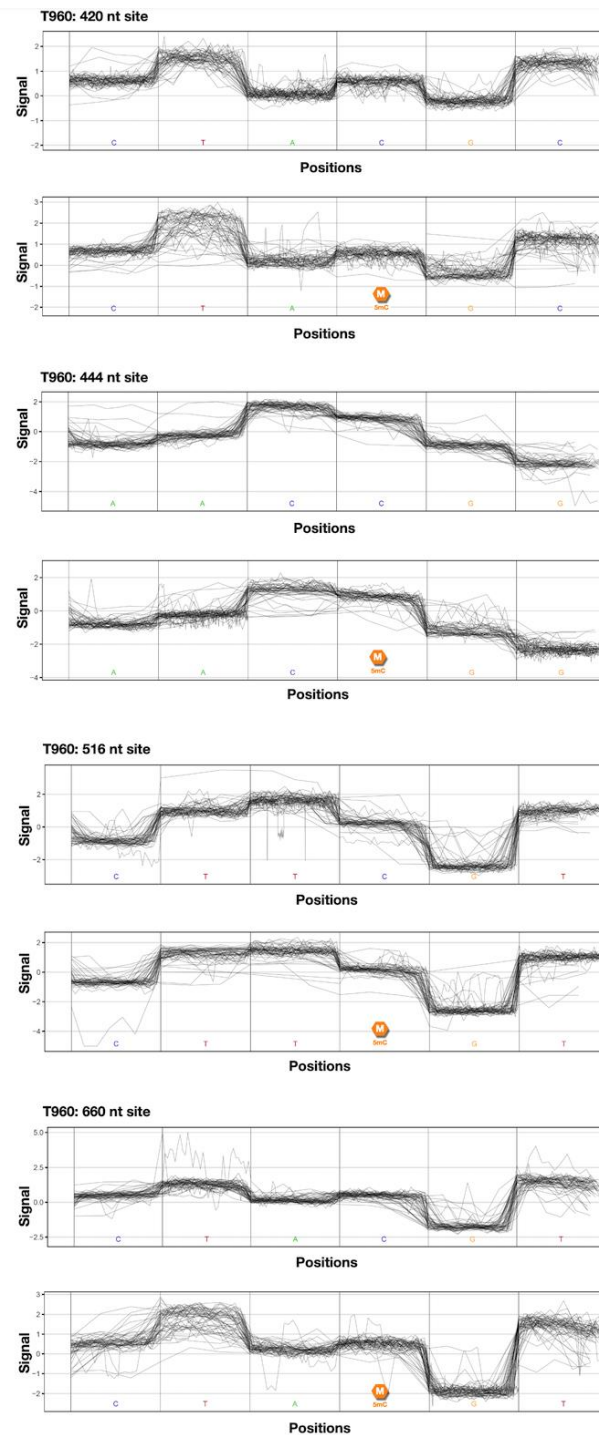


Fig. S12. The raw nanopore electric signals at the modification sites. Here eight CG sites are selected for illustrating. Methylation sites are indicated by an orange block. Raw nanopore electrical signals show current shifts (squiggles) at or around the methylated sites. These signal shifts can be captured and analyzed by Megalodon using several built-in neural network models to achieve discriminations of the methylation modifications.

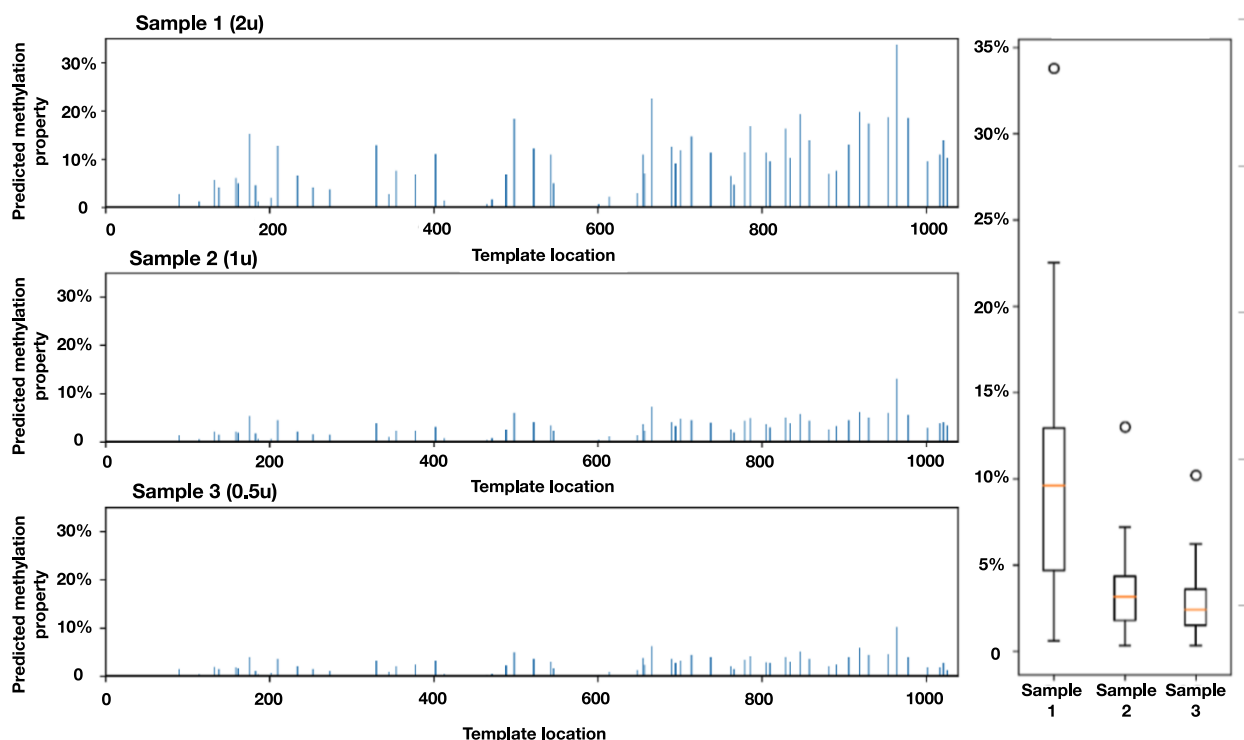


Fig. S13. Comparisons of the methylation efficiency using varying concentrations of M.SssI.

The effects of M.SssI concentration on methylation were analyzed. Three individual experiments were implemented using M.SssI concentrations at 2u, 1u, and 0.5u in 20 μ L solution. The histograms show the methylation calling results corresponding to three different concentrations. The horizontal axis indicates the base position and the vertical axis indicates the predicted methylation probabilities. The boxplots illustrate statistical results of methylation probabilities at different M.SssI concentrations. As shown in the figure, with the concentrations of M.SssI reducing from 2 to 0.5 u/ μ L, a decrease in the predicted scores is observed. Box plots show the distribution of the data in bar plots (n = 70 for sample).

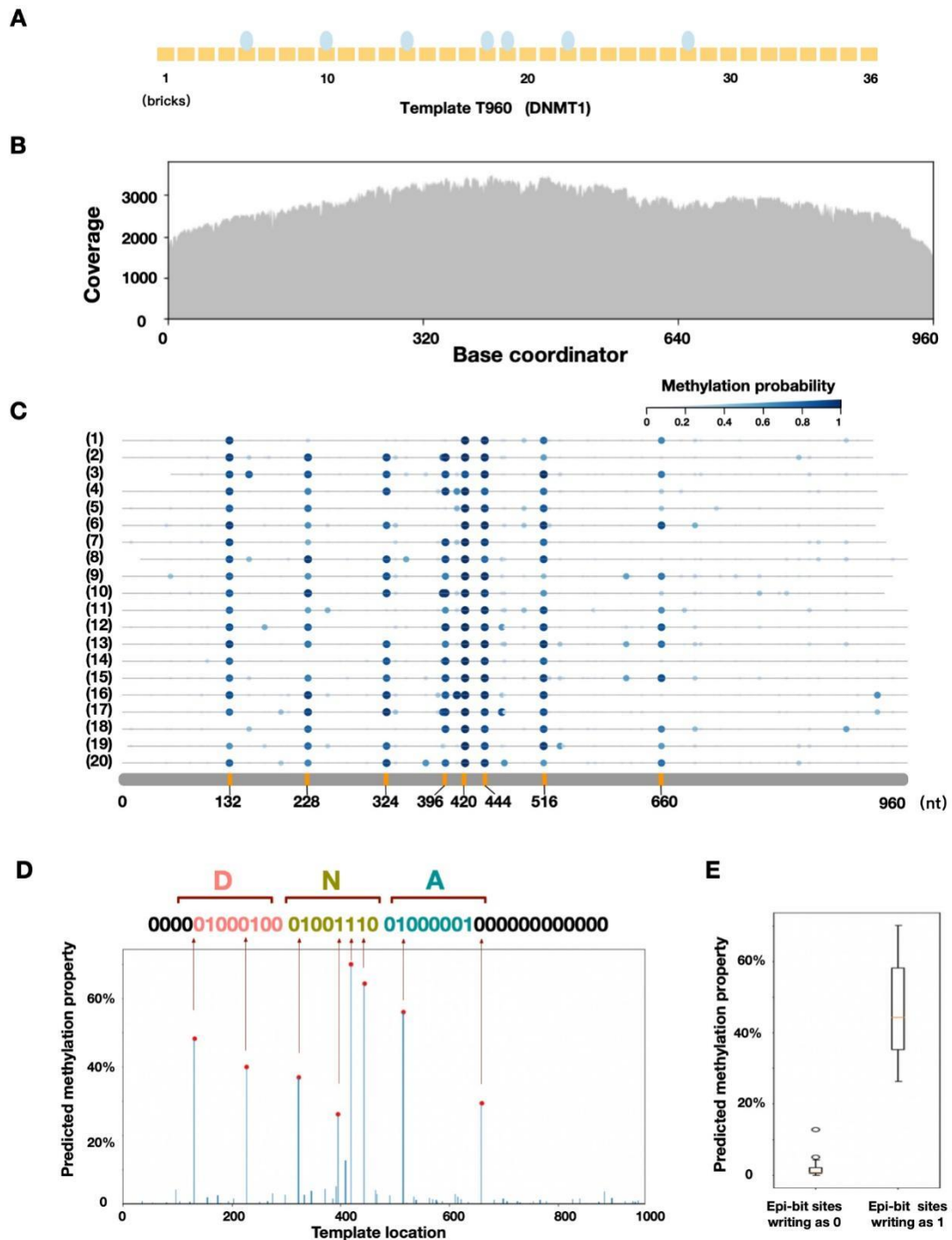


Fig. S14. The epi-bit writing test on T960 DNA carrier to store “DNA” letters. To demonstrate the analysis pipeline of Megalodon, we stored the text string "DNA" to the T960 template (36 potential epi-bit sites) using DNMT1. **(A)** The distribution of the designed modification pattern. The string “DNA” was transcoded to a methylation pattern by a binary transformation of ASCII codes. The information coded 960-nt sequence contained eight methylation sites. **(B)** The

coverages in nanopore sequencing. 6,802 valid reads are obtained from nanopore sequencing with an average sequencing coverage of 2,816 $\times$. **(C)** The methylation calling results from individual sequencing reads. **(D)** The predicted methylation probabilities by megalodon. Red dots indicate the predesigned “1” methylation sites. Here the “per site” mode of Megalodon was used, *i.e.*, reads were firstly aligned and then called collectively. **(E)** Boxplot of the predicted methylation probabilities at epi-bit 0s and 1s. Box plots show the distribution of the data in bar plots ($n = 62$ for epi-bit 0, $n=8$ for epi-bit 1).

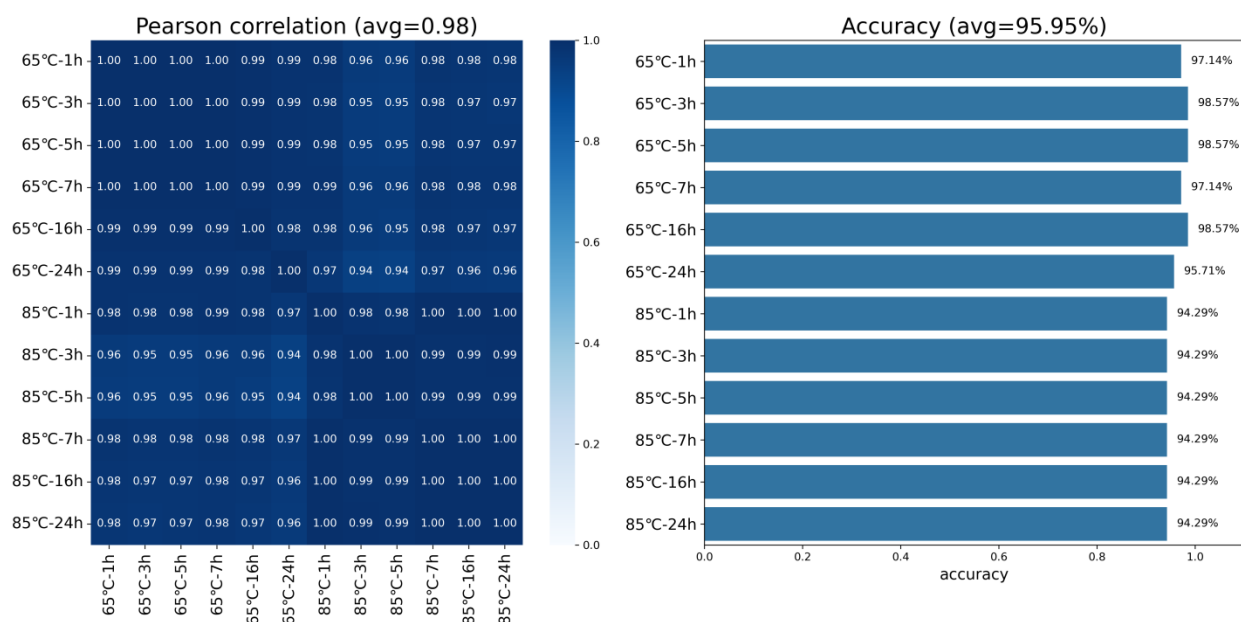


Fig. S15. The results analysis of the stability test experiments. **(A)** The correlations of the sequencing results treated with different heating time. **(B)** The average accuracies of the samples treated with different heating time. The epi-bit writing test on DNA template L4.

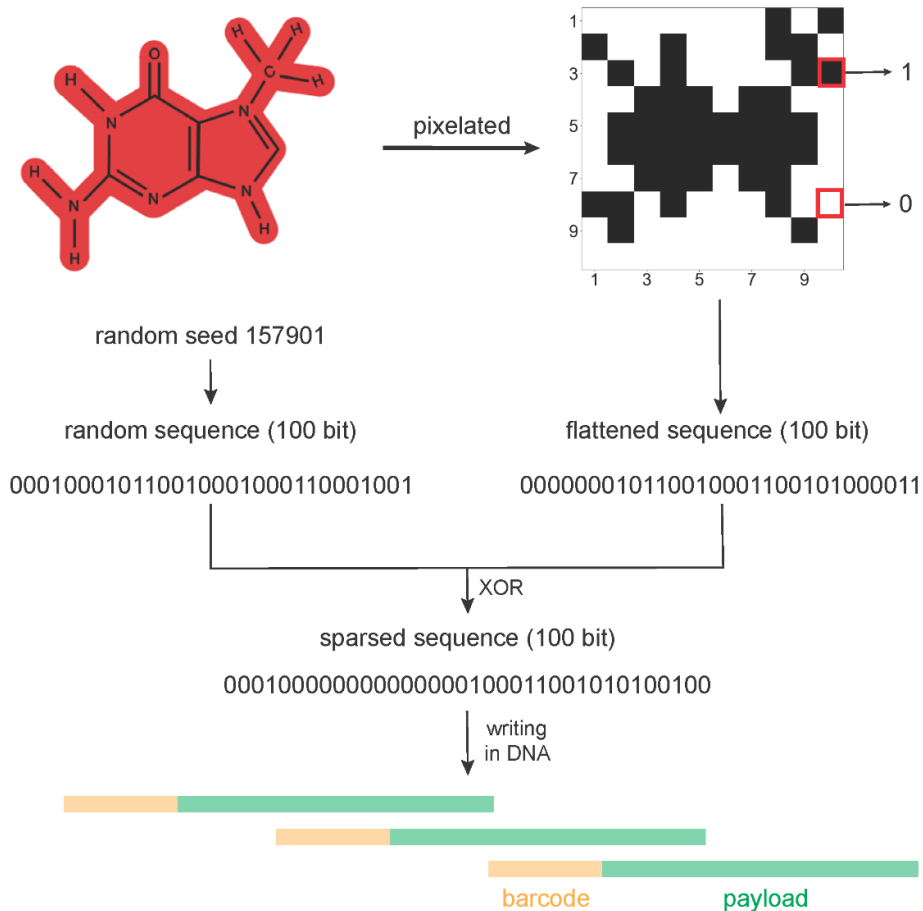


Fig. S16. Scheme of image encoding processes. The images of base chemical structures were firstly transformed into bitmaps and then flattened, resulting in streams of binary numbers. Next, bit-wise XOR of information with a stream of random binary numbers of the same length was performed for each stream of binary numbers, to generate a sparse sequence containing the content of epi-bit 1s less than 33% and no consecutive epi-bit 1s with a length longer than 3.

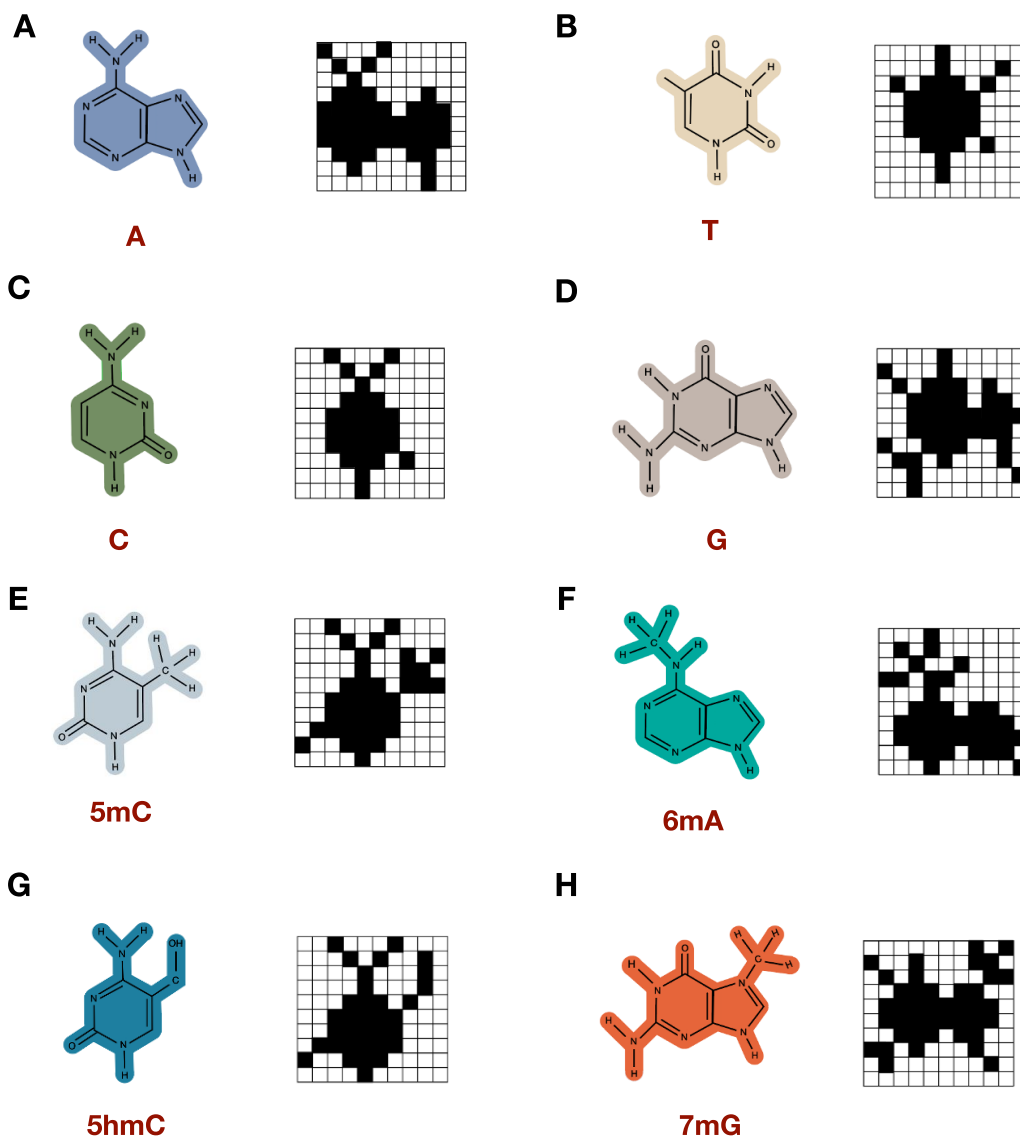


Fig. S17. Eight images of base chemical structures and the corresponding bitmaps. The size of each binary graph is 10×10 bits.

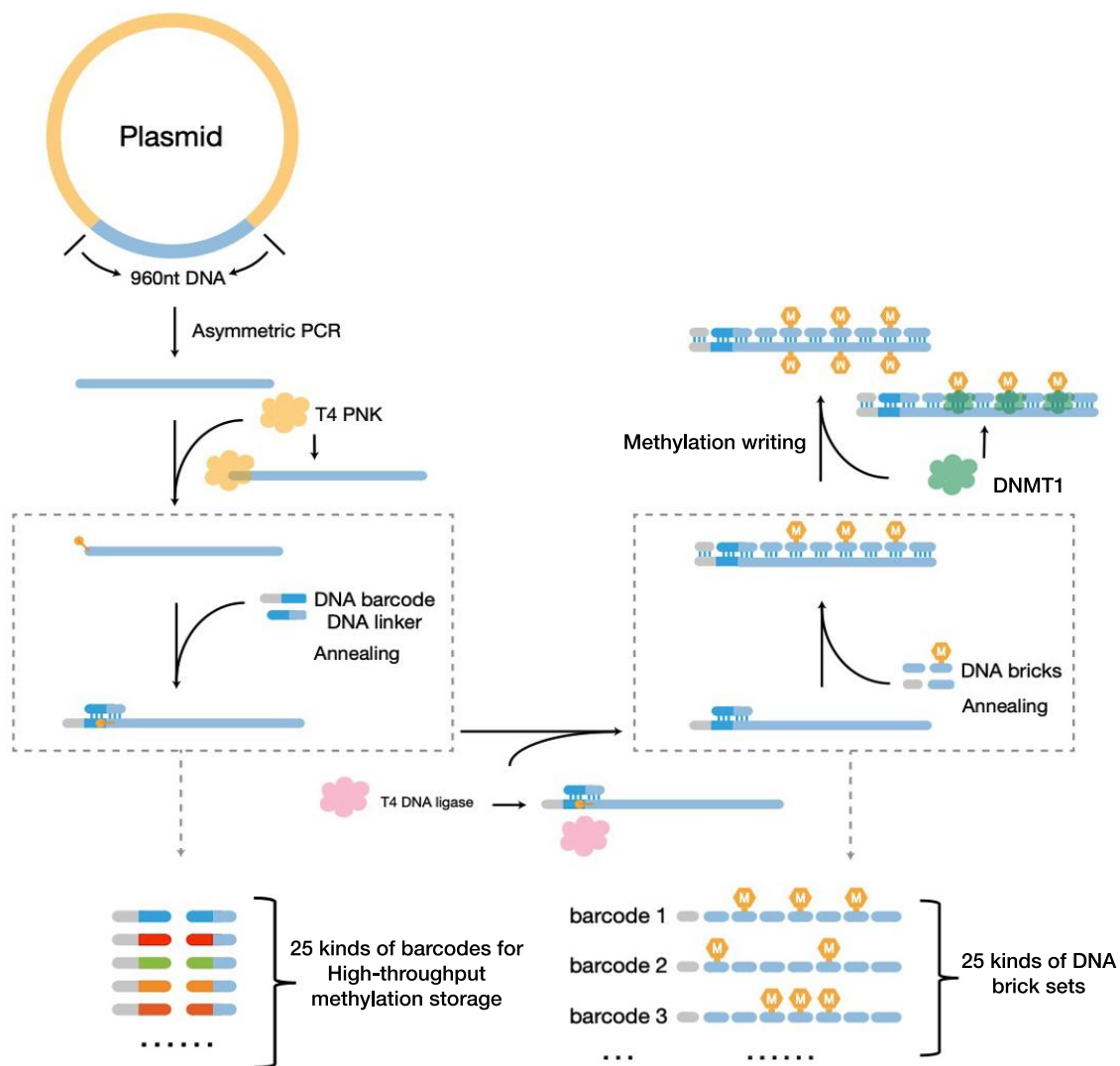


Fig. S18. The main experimental steps of the parallel methylation storage. The experimental procedure consisted of three main steps: T960 carrier preparation, barcode ligation and methyl transfer.

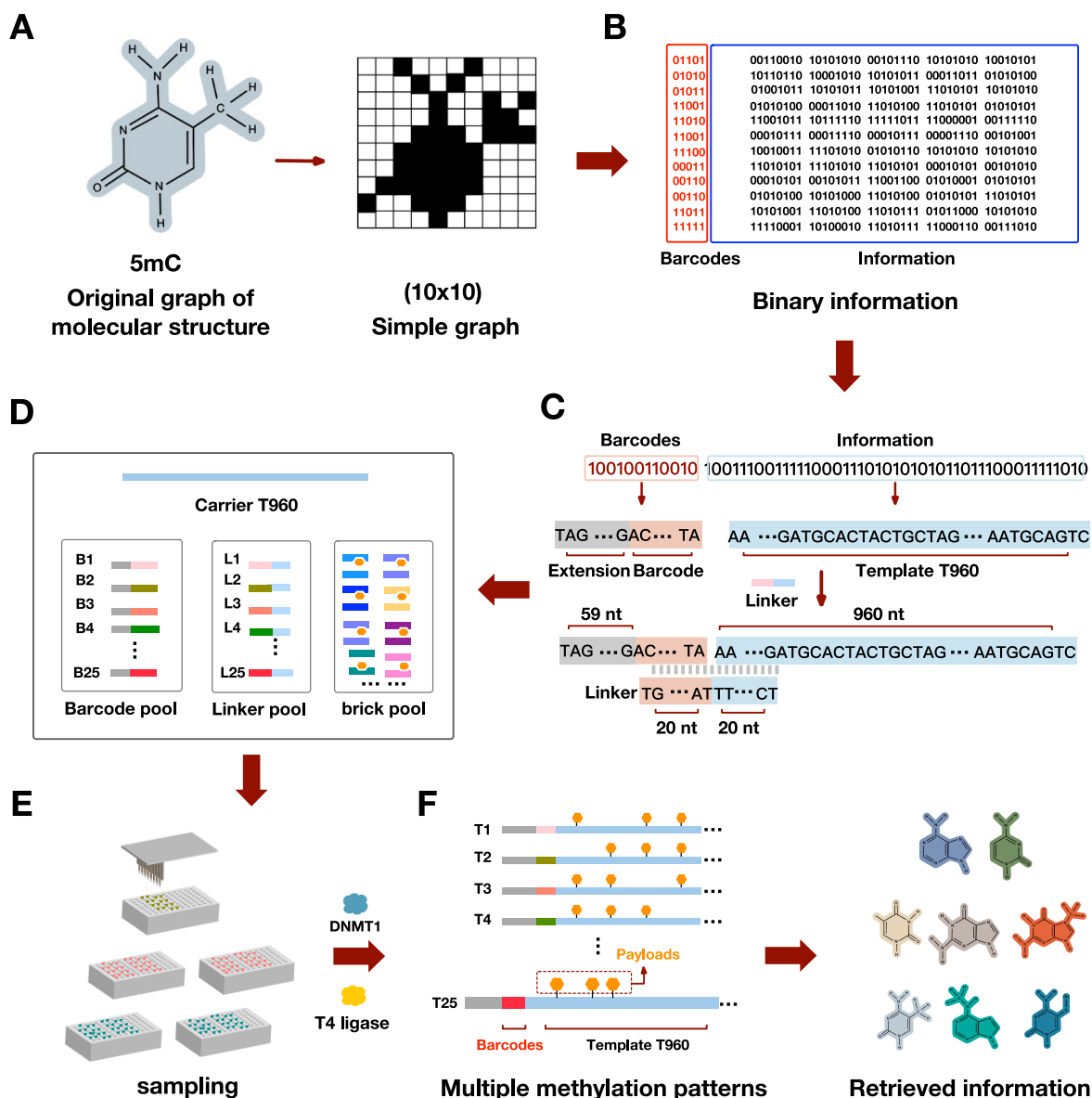


Fig. S19. The workflow of parallel epi-bit storage. (A) Illustration of the original image and the corresponding bitmaps (one of the eight stored images is shown here as an example). (B) The transformed binary stream of the information in the bitmaps. Here white pixels correspond to 0s (no methylation) and black pixels correspond to 1s (methylation). (C) The design details of labelling 960 nt carriers with different barcodes, where specific DNA linkers are used to connect the barcodes and carriers. (D) The DNA molecules used in the storage process, including 960 nt DNA carriers, barcode pool, linker pool and brick pool. (E) Sampling of appropriate reactants for information writing. (F) Information writing on 960 nt DNA carriers.

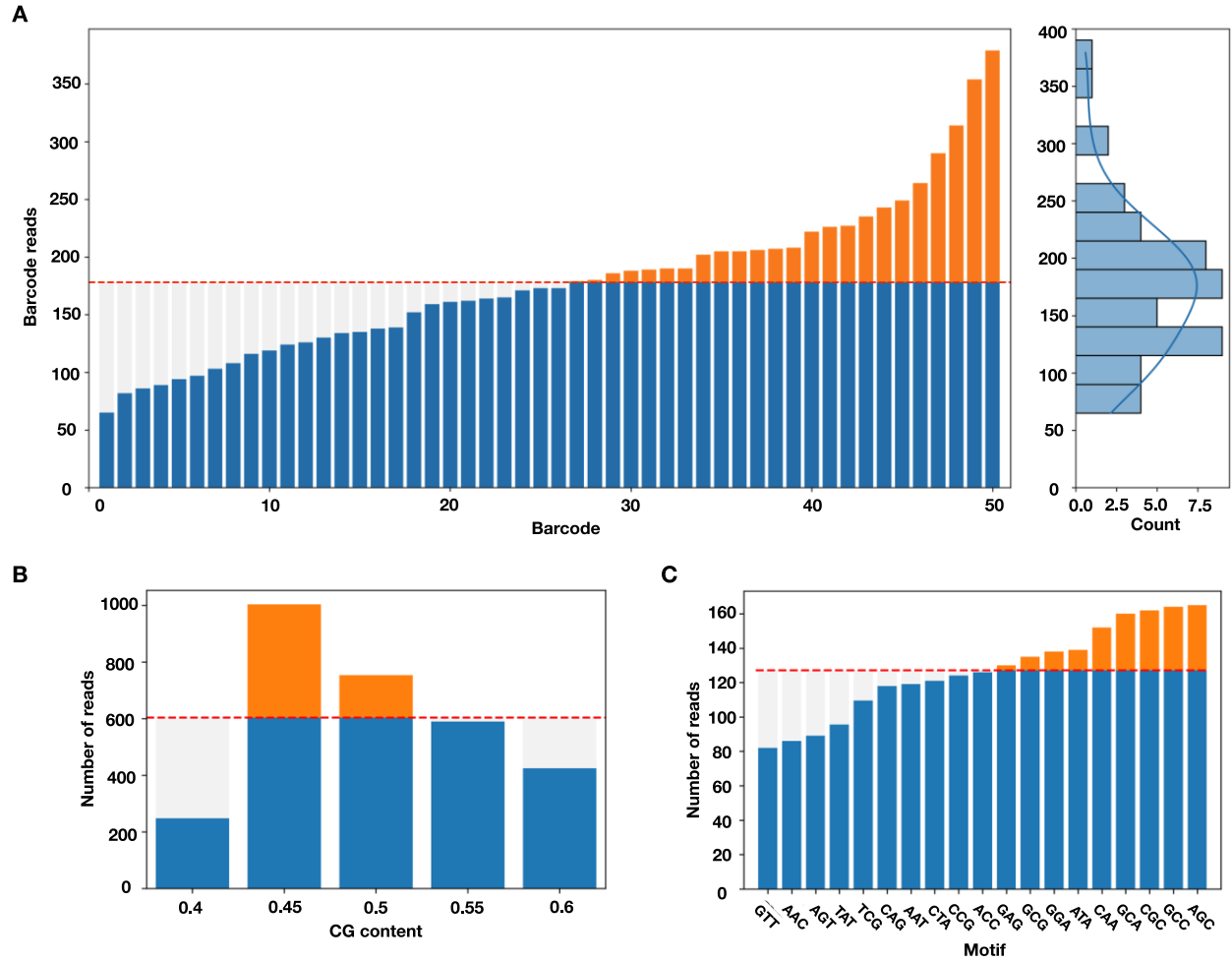


Fig. S20. The nanopore sequencing test of 50 kinds of barcoded DNA carriers in one-pot reaction. To test the possibility to read a large number of barcoded DNA carriers in one-pot nanopore sequencing, 50 kinds of barcoded DNA carriers are used in the test experiments. **(A)** The read coverages of 50 kinds of DNA carriers with different barcodes. The results indicate all 50 kinds of barcoded DNA carriers can be obtained and discriminated by the barcodes. **(B)** The GC contents of sequencing reads. **(C)** Statistical sequencing read results of the DNA sequence motif at the 3' ends of the barcodes.

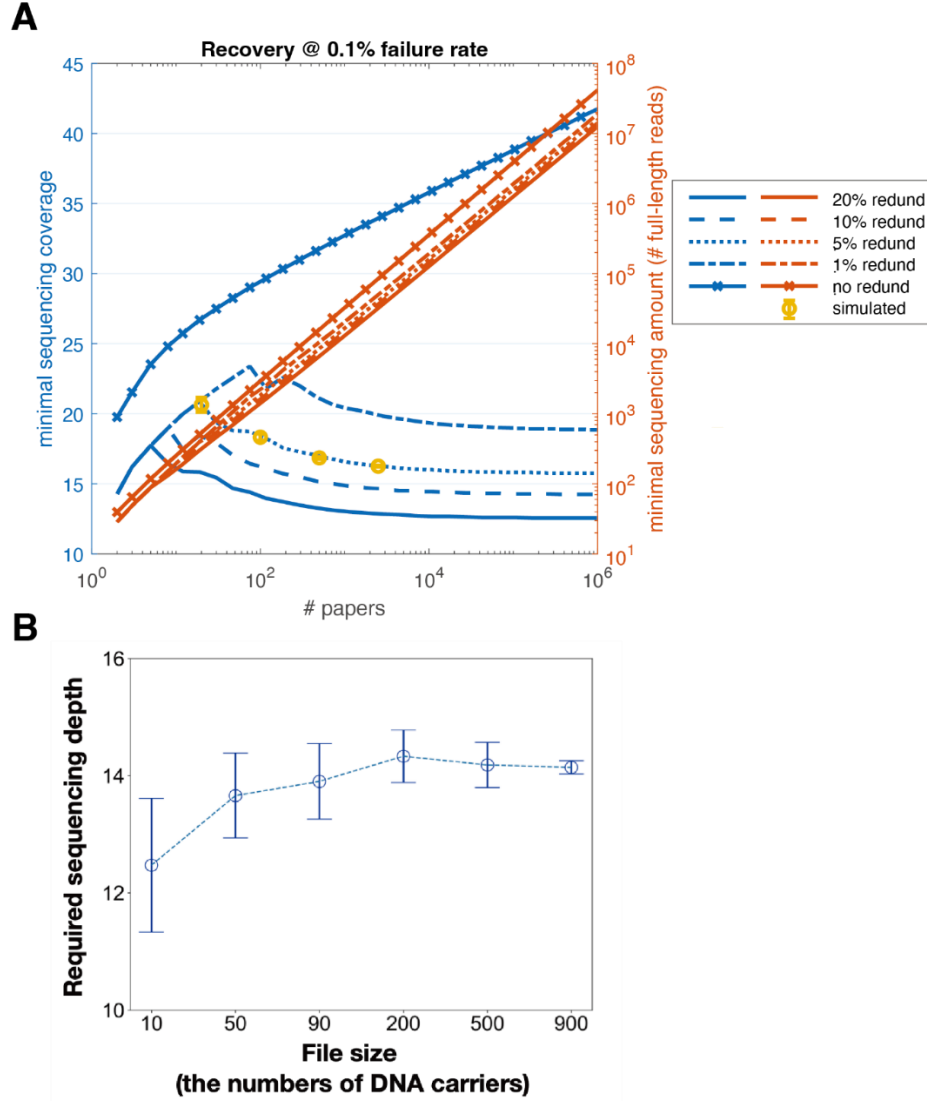


Fig. S21. Sequencing throughputs required for retrieving files from the methylation-based storage system. Large files stored in methylation storage system are supposed to be distributed on different barcoded DNA templates. Each template has to be sequenced for a certain coverage n_0 for the methylation modifications to be called with high confidence. **(A)** Theoretical and simulated sequencing depths for retrieving all information at various redundancy rates $1 - q$ and failure rate of 0.1%. Here we illustrate the numeric solution and simulated results for the minimal sequencing coverage k and sequencing amount Nk when the decoding failure rate $1 - \Pr\{\text{decoding}\} = 0.1\%$ with $n_0 = 10$ and N spans six orders of magnitudes. The theoretical calculation is confirmed by sequence sampling simulations at $N = 20, 100, 500, 2500$ and $q = 0.95$. Yellow circles represent the mean of 0.999 quantiles of minimal sequencing depths of 2000 random sequencing samples. Error bars indicates standard deviations of $n = 50$ simulations. **(B)** Simulated minimal sequencing

depths for retrieving $N_{1,2} = 10/50$ (small file), $N_{3,4} = 90/200$ (medium file), $N_{5,6} = 500/900$ (large file) from a $N = 1000$ DNA template pool. Redundancy $1 - q = 0.1$. Data are shown as mean $\pm$ standard deviation of $n = 10$ simulations. These results suggest that logic redundancy is necessary from preventing the system from exploding. With logic redundancy, large files required almost constant sequencing coverages for recovery. Our definition of q requires an error correction scheme effective for dropout errors, which indicates that Fountain codes or nested RS codes could be involved in methylation storage system.

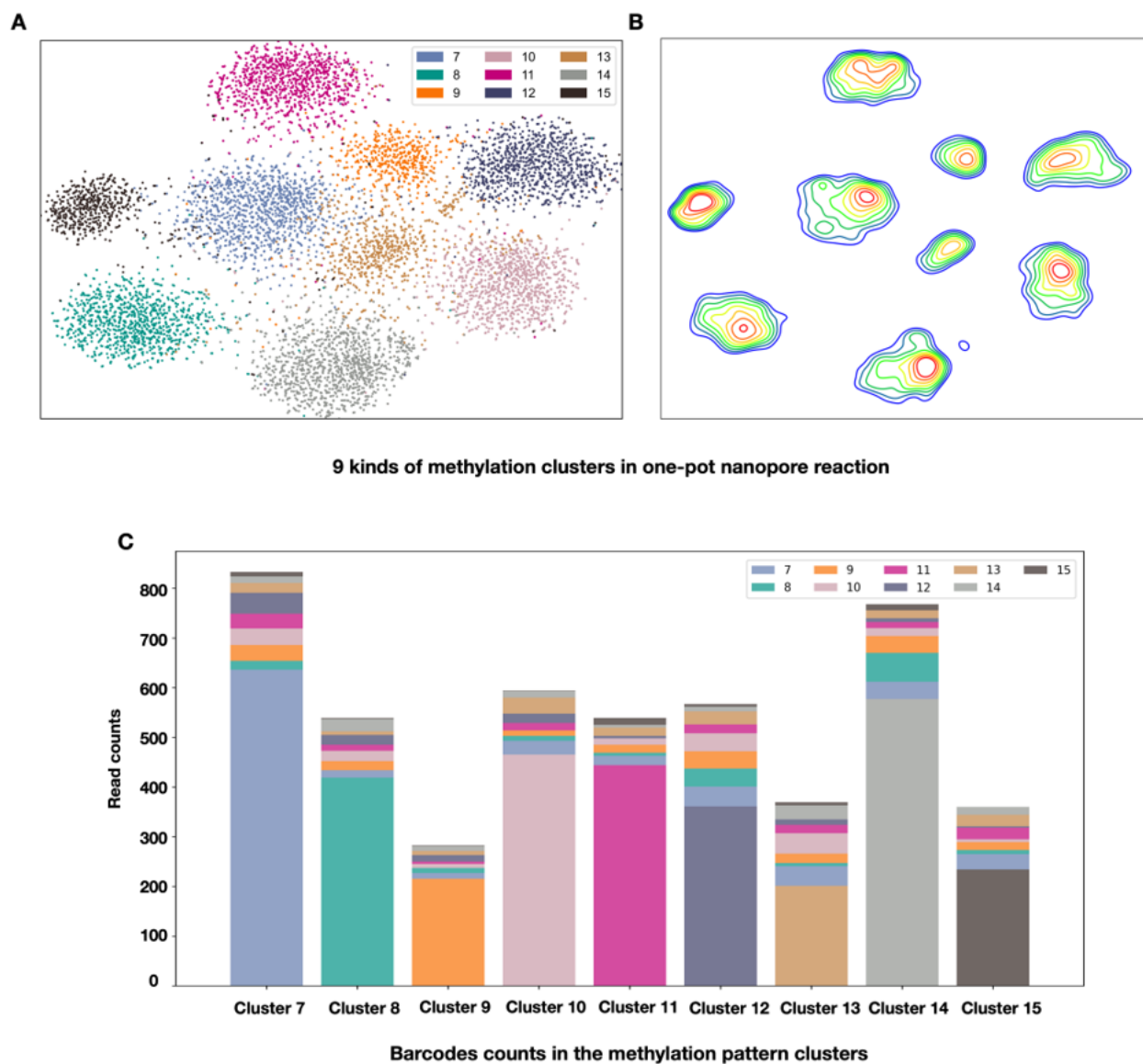


Fig.S22. Nanopore sequencing cluster results of DNA carriers 7-15. (A) The clustering results of raw reads. The reads with different barcodes are labelled by different colors. Each dot corresponds to a single read. (B) The density map of the nine clusters. (C) The number of clustered reads for each of the 9 carriers with different barcodes. The results indicate the correctly clustered reads take the largest proportions.

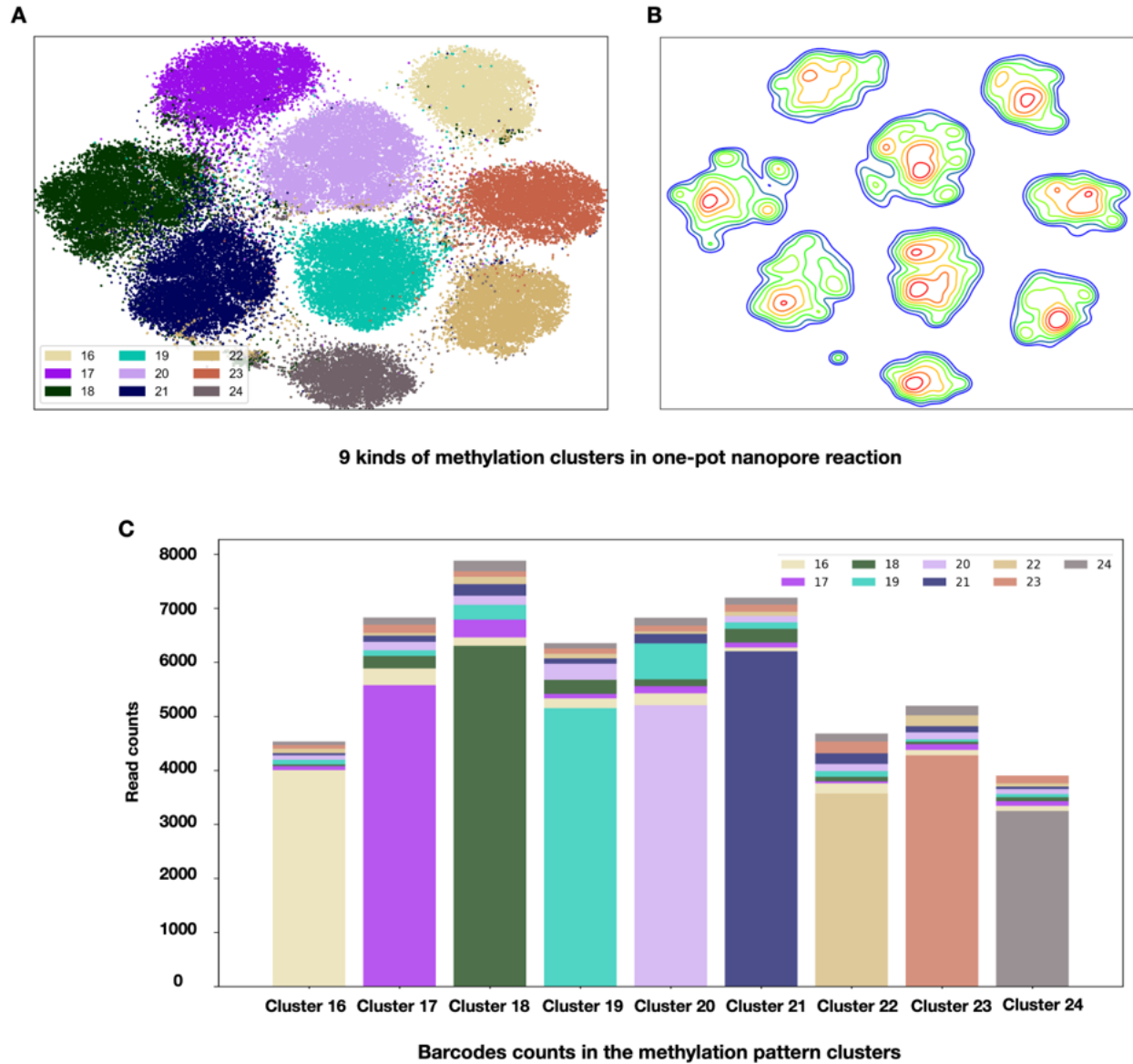


Fig. S23. Nanopore sequencing cluster results of DNA carriers 16-24. (A) The cluster results of raw reads. The reads with different barcodes are labelled by different colors. Each dot corresponds to a single read. (B) The density map of the nine clusters. (C) The number of clustered reads for each of the 9 carriers with different barcodes. The results indicate the correctly clustered reads take the largest proportions.

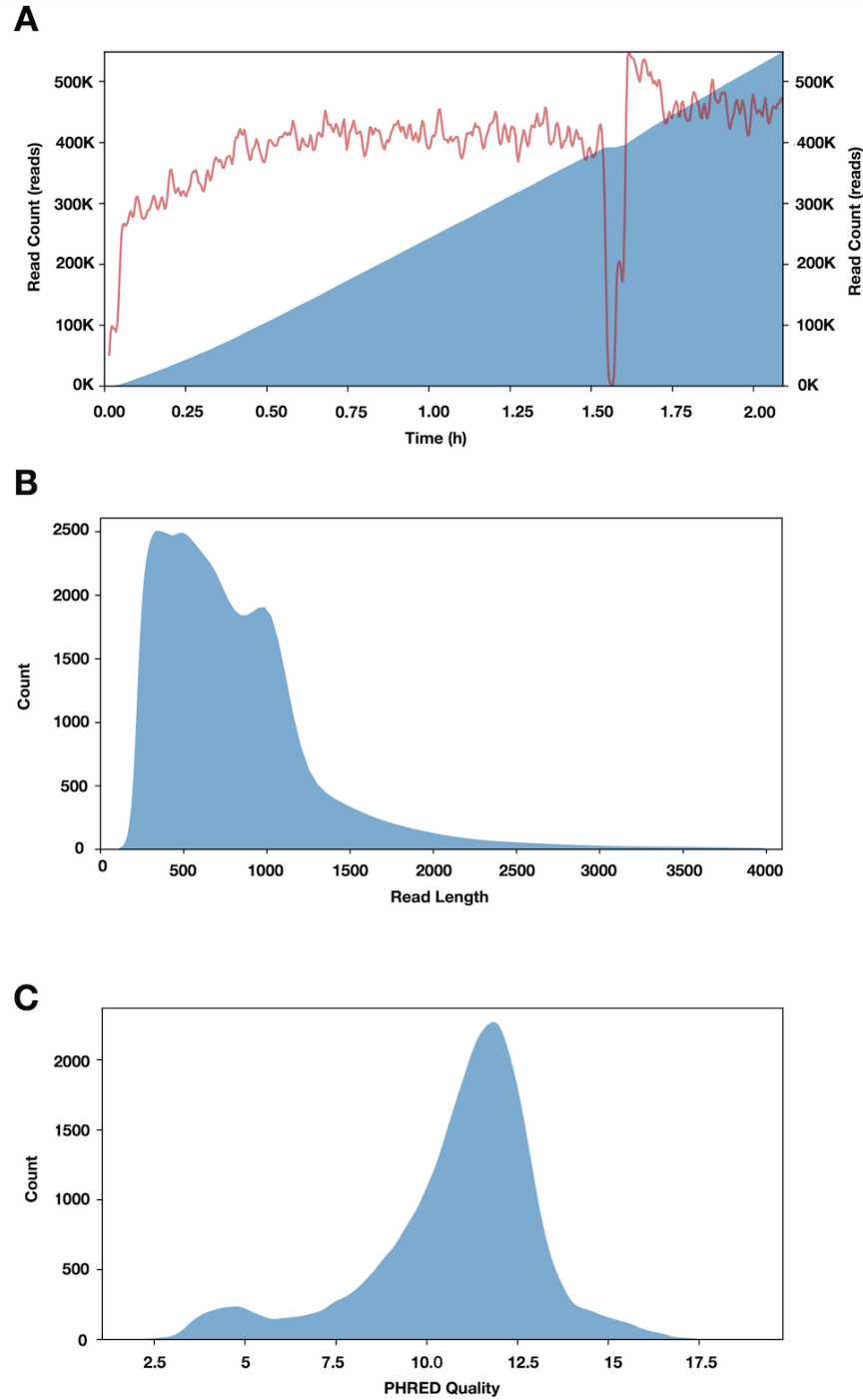


Fig. S24. Overview of sequencing results in one-pot nanopore sequencing reaction containing 25 kinds of barcoded DNA carriers. (A) Real-time sequencing speed and cumulative outputs during nanopore sequencing. **(B)** The distributions of base-called reads lengths. **(C)** The distributions of PHRED quality of base-called reads.



Fig. S25. Alignment of base-called reads of 25 kinds of barcoded DNA carriers in one-pot nanopore. The upper panel shows the per-site sequencing coverage over the full-length template. Color codes in the alignment panels are: (1) mismatches, A-green, G-yellow, C-blue, T-orange; (2) insertions-purple, (3) deletions-dark grey and (4) gaps-black spaces.

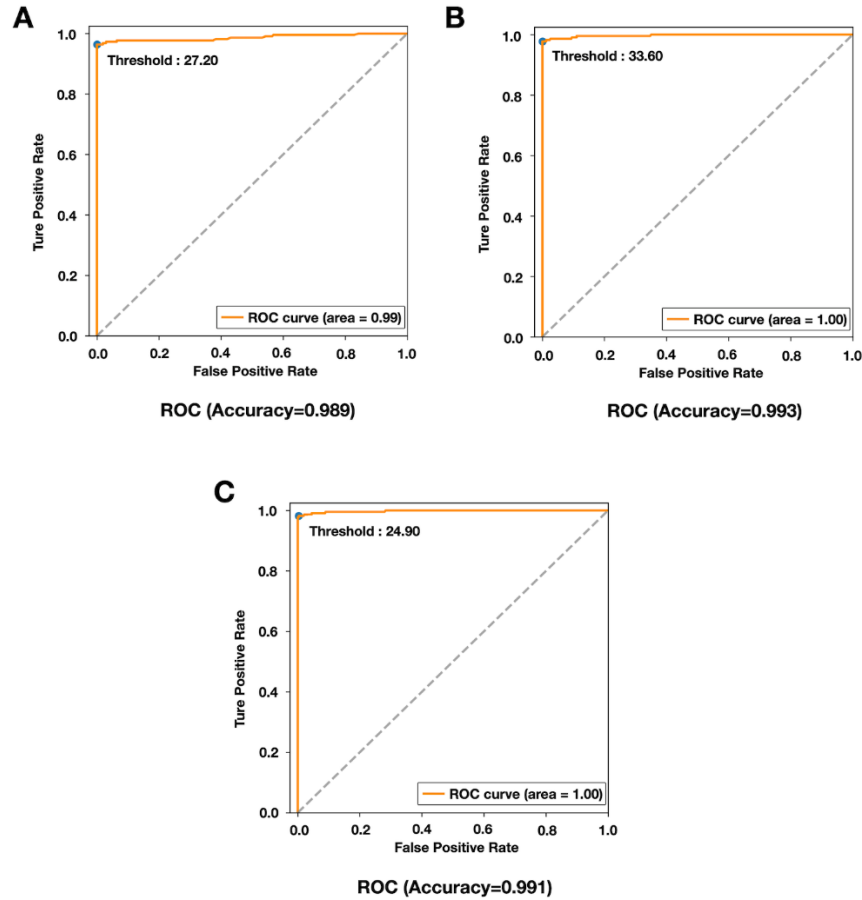


Fig. S26. Methylation calling results of three different post-sequencing processing methods.

Fig. S34 presented different workflows of retrieving methylation information from nanopore reads. Specifically, the grouping of reads can be conducted in three manners: (1) cluster-only, (2) barcode-only and (3) cluster and then screening by barcodes. Here we compared the ROC (receiver operating characteristic) curves and the accuracy based on the optimal ROC threshold (Youden Index) of these three methods. **(A)** cluster-only, **(B)** barcode-only and **(C)** cluster and then screening by barcodes. The line graphs show the corresponding ROC curves for each method. It is evident that the AUC (Area Under Curve) of the second method (barcode-only) was the lowest among the three methods, as well as the accuracy. Nevertheless, in this case, the accuracy can still reach 0.98. The first method (cluster-only) was much better than the second method, with an AUC of 1.0 and an accuracy of 0.99. The ROC curve of the third method (combination of clustering and barcode) was closer to the upper left corner, which reflected a better prediction result. Therefore, we adopted the last method combining clustering and barcode as the analysis method.

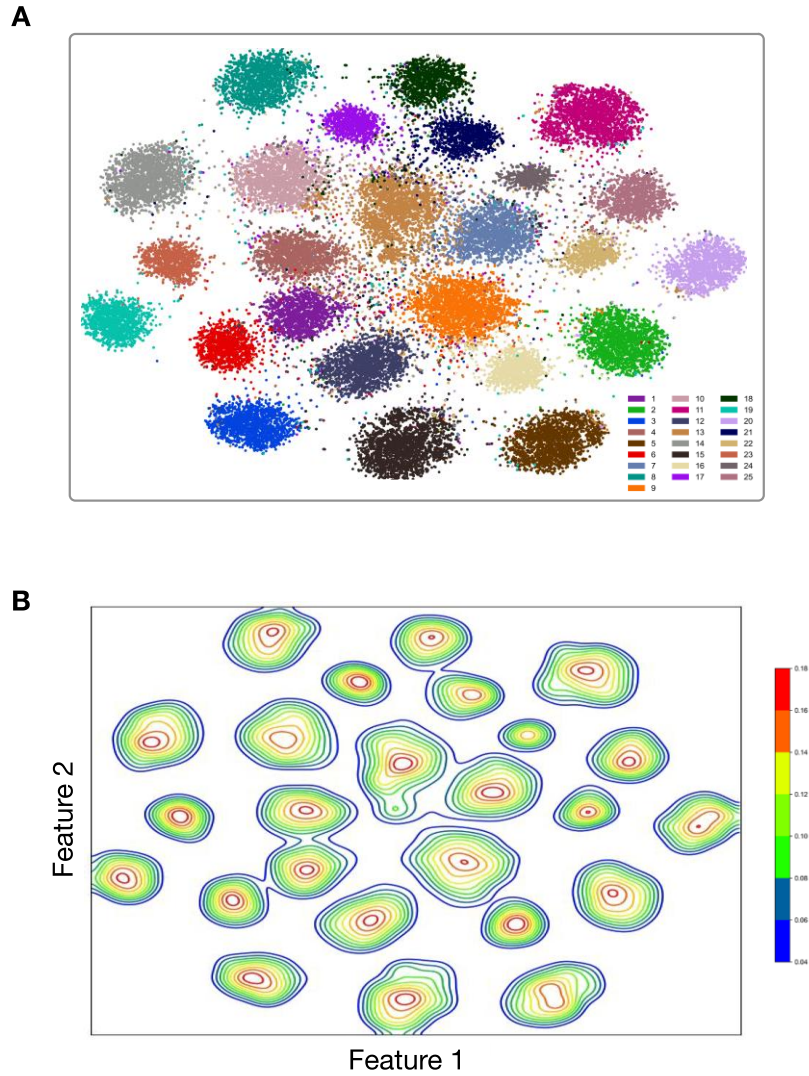


Fig. S27. The cluster density of reads with 25 kinds of methylation information. In the parallel nanopore sequencing, the loading base sequences of all 25 kinds of DNA carriers were identical, but with different barcodes labelled. Here, the sequencing results were clustered by the specific methylation patterns predicted by Megalodon. The 25 encoded methylation patterns are successfully matched to the 25 specific clusters (**A**), and the densities of 25 clusters are shown in (**B**).

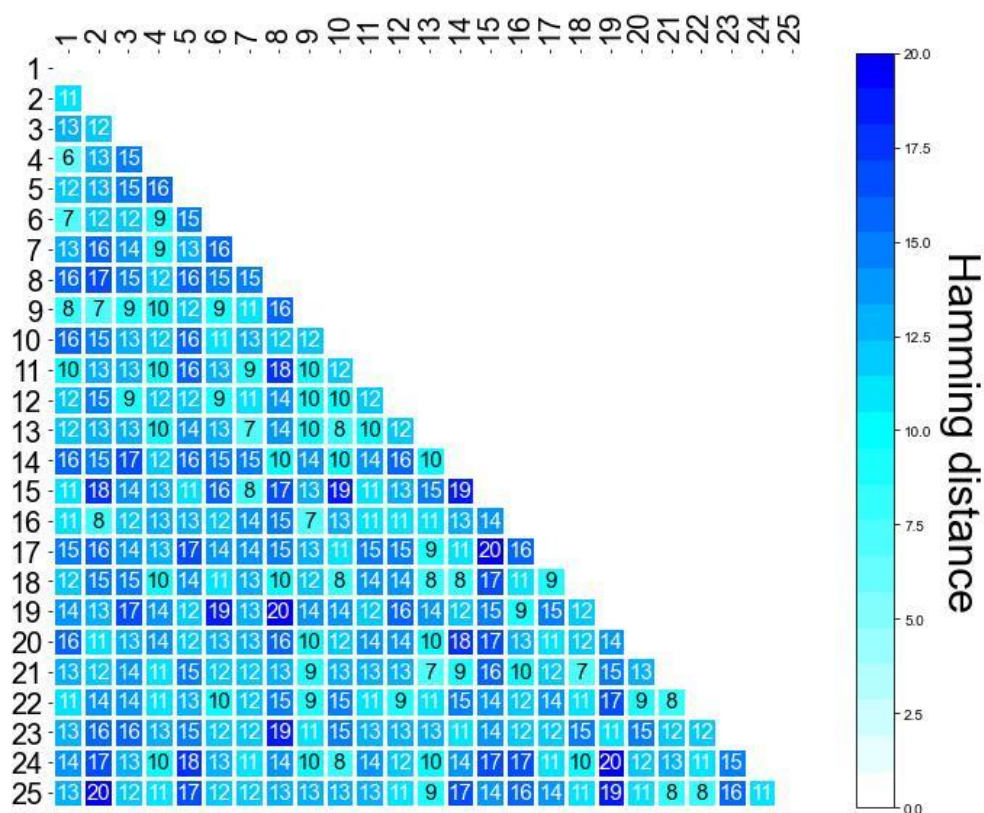


Fig. S28. The Hamming differences between methylation patterns in the 25 kinds of DNA carriers. Note that the base sequences of the loading regions in the DNA carriers are identical, so are the positions of the CG sites.

carriers T1, T2, T3 and T25. (C) Methylation detection results obtained from Megalodon of each DNA carrier. The bit stream on top shows the methylation information, where 1s indicate methylated sites and 0s indicate unmethylated sites. The bar plot shows the methylation probabilities of the specific epi-bit sites in the experiments. Sites with methylation write-in representing value 1 are marked with red dots. The boxplot on the right shows the mean values and the distribution of Epi-bit 0s (unmethylated sites) and Epi-bit 1s (methylated sites). In fact, some outliers were found, indicating not all methylation information was completely retrieved. The falsely predicted epi-bits were mainly found at the second used epi-bit site (Extended Data Fig. 5) on the DNA carrier, implying the problems may lie in designs. Nevertheless, the vast majority of the positives were distinguishable. Box plots show the distribution of the data in first bar plots (n = 59 for epi-bit 0, n=11 for epi-bit 1). Box plots show the distribution of the data in second bar plots (n = 59 for epi-bit 0, n=11 for epi-bit 1). Box plots show the distribution of the data in third bar plots (n = 59 for epi-bit 0, n=11 for epi-bit 1).

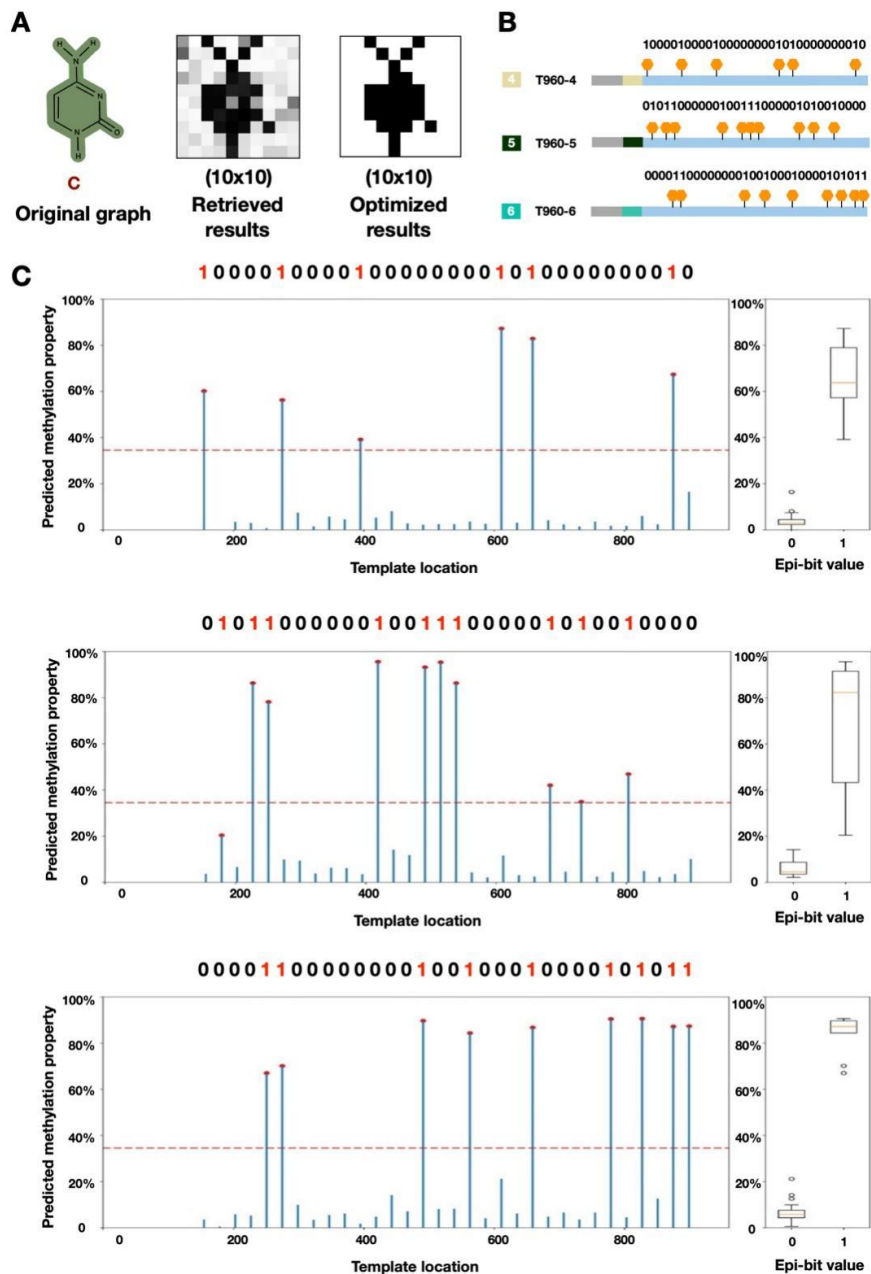


Fig. S30. The parallel storage of the second molecule structural image. (A) The original image, the retrieved image based on methylation probabilities and the retrieved image by threshold optimization. (B) The methylation information of the first encoded image, stored in DNA T960 carriers T4, T5, T6 and T25. (C) Methylation detection results obtained from Megalodon of each DNA carrier. Box plots show the distribution of the data in first bar plots (n = 64 for epi-bit 0, n=6 for epi-bit 1). Box plots show the distribution of the data in second bar plots (n = 60 for epi-bit 0, n=10 for epi-bit 1). Box plots show the distribution of the data in third bar plots (n = 61 for epi-bit 0, n=9 for epi-bit 1).

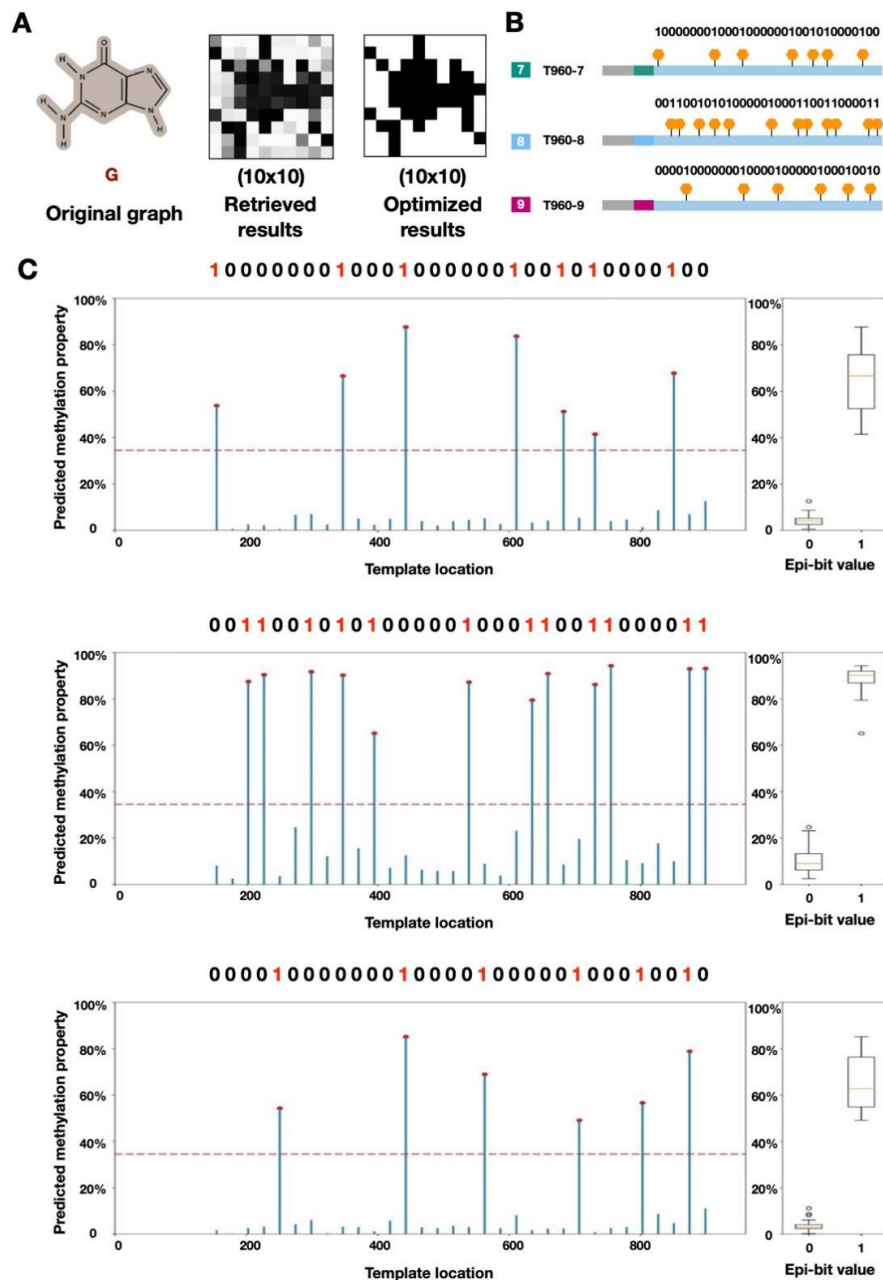


Fig. S31. The parallel storage of the third molecule structural image. (A) The original image, the retrieved image based on methylation probabilities and the retrieved image by threshold optimization. (B) The methylation information of the first encoded image, stored in DNA T960 carriers T7, T8, T9 and T25. (C) Methylation detection results obtained from Megalodon of each DNA carrier. Box plots show the distribution of the data in first bar plots (n = 63 for epi-bit 0, n=7 for epi-bit 1). Box plots show the distribution of the data in second bar plots (n = 58 for epi-bit 0, n=12 for epi-bit 1). Box plots show the distribution of the data in third bar plots (n = 64 for epi-bit 0, n=6 for epi-bit 1).

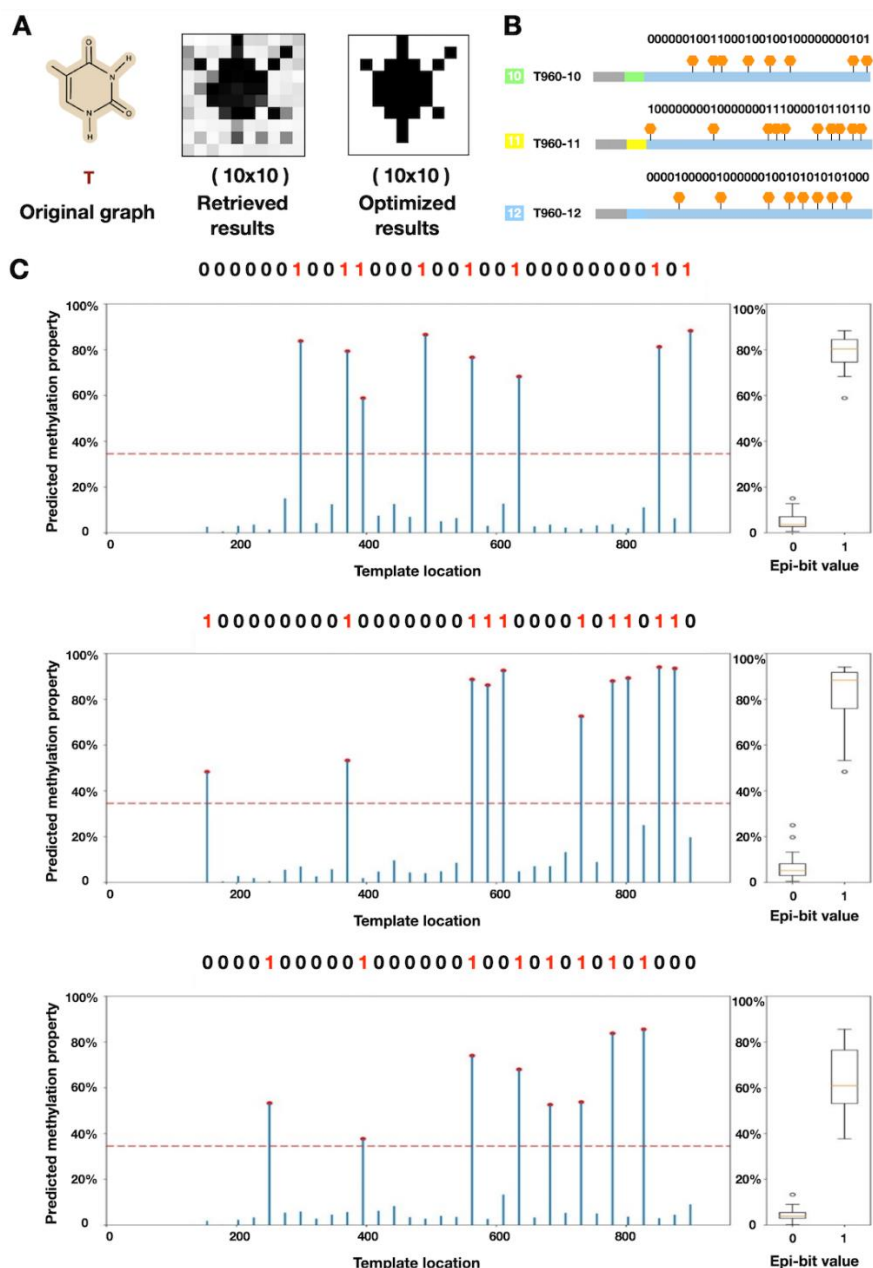


Fig. S32. The parallel storage of the fourth molecule structural image. (A) The original image, the retrieved image based on methylation probabilities and the retrieved image by threshold optimization. (B) The methylation information of the first encoded image, stored in DNA T960 carriers T10, T11, T12 and T25. (C) Methylation detection results obtained from Megalodon of each DNA carrier. Box plots show the distribution of the data in first bar plots ($n = 63$ for epi-bit 0, $n=7$ for epi-bit 1). Box plots show the distribution of the data in second bar plots ($n = 58$ for epi-bit 0, $n=12$ for epi-bit 1). Box plots show the distribution of the data in third bar plots ($n = 64$ for epi-bit 0, $n=6$ for epi-bit 1).

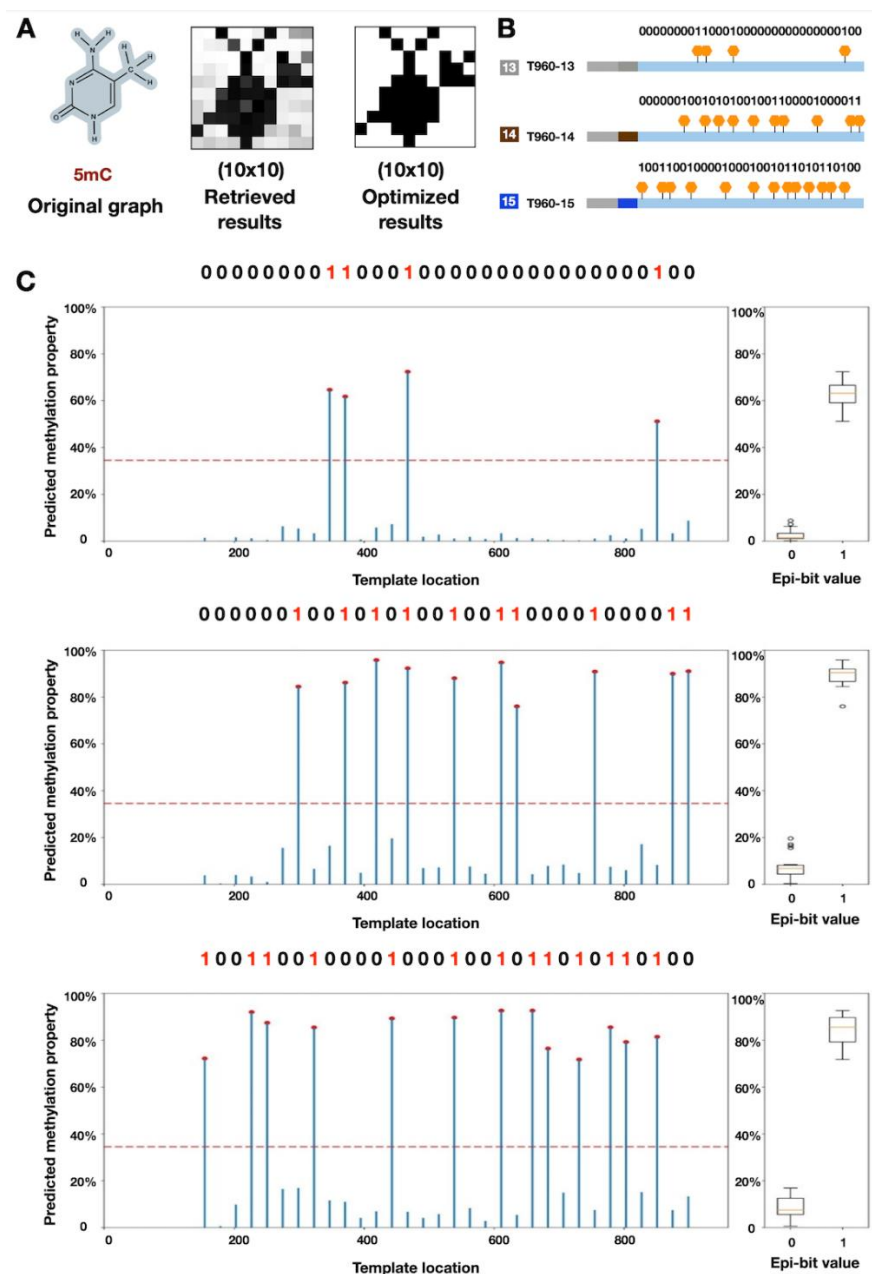


Fig. S33. The parallel storage of the fifth molecule structural image. (A) The original image, the retrieved image based on methylation probabilities and the retrieved image by threshold optimization. (B) The methylation information of the first encoded image, stored in DNA T960 carriers T13, T14, T15 and T25. (C) Methylation detection results obtained from Megalodon of each DNA carrier. Box plots show the distribution of the data in first bar plots (n = 65 for epi-bit 0, n=5 for epi-bit 1). Box plots show the distribution of the data in second bar plots (n = 60 for epi-bit 0, n=10 for epi-bit 1). Box plots show the distribution of the data in third bar plots (n = 57 for epi-bit 0, n=13 for epi-bit 1).

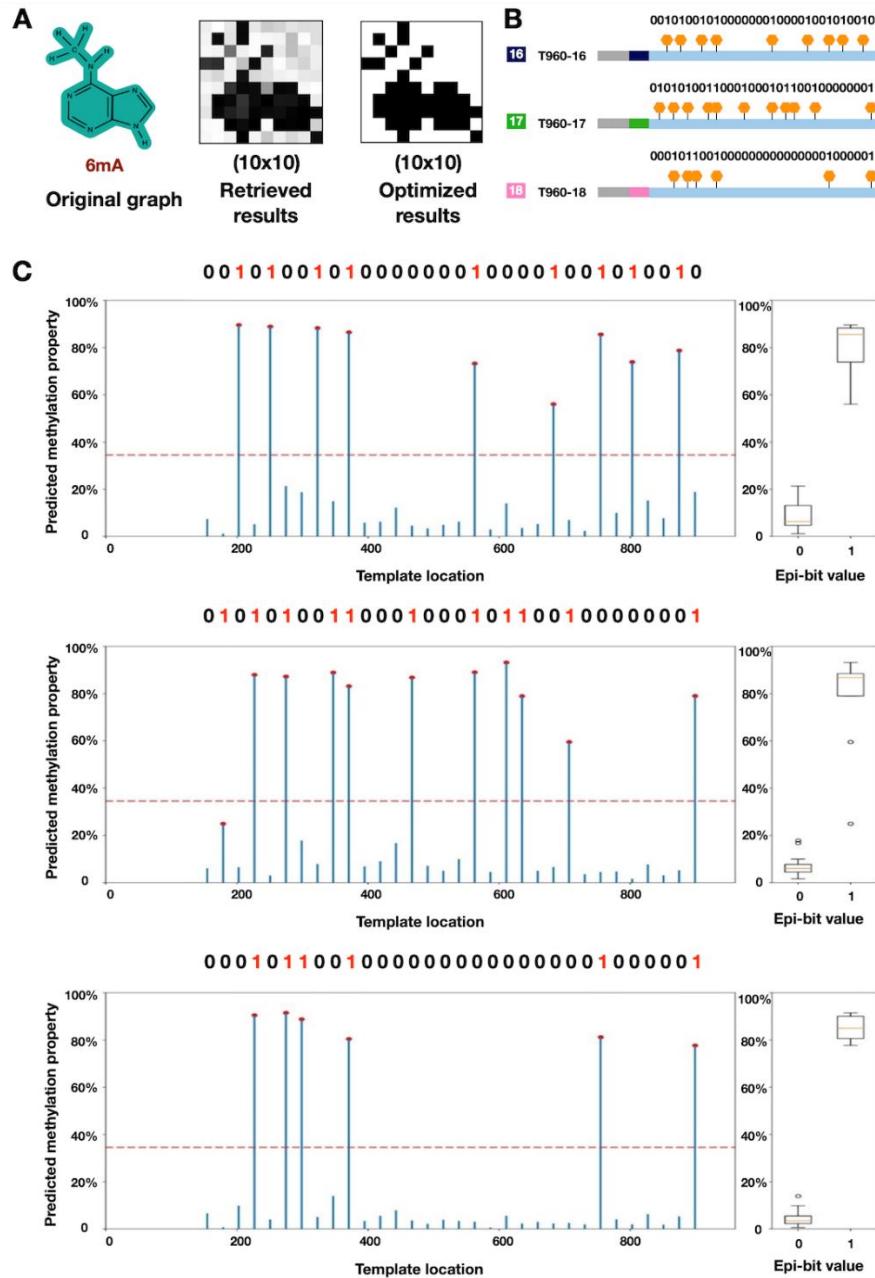


Fig. S34. The parallel storage of the sixth molecule structural image. (A) The original image, the retrieved image based on methylation probabilities and the retrieved image by threshold optimization. (B) The methylation information of the first encoded image, stored in DNA T960 carriers T16, T17, T18 and T25. (C) Methylation detection results obtained from Megalodon of each DNA carrier. Box plots show the distribution of the data in first bar plots ($n = 61$ for epi-bit 0, $n=9$ for epi-bit 1). Box plots show the distribution of the data in second bar plots ($n = 59$ for epi-bit 0, $n=11$ for epi-bit 1). Box plots show the distribution of the data in third bar plots ($n = 64$ for epi-bit 0, $n=6$ for epi-bit 1).

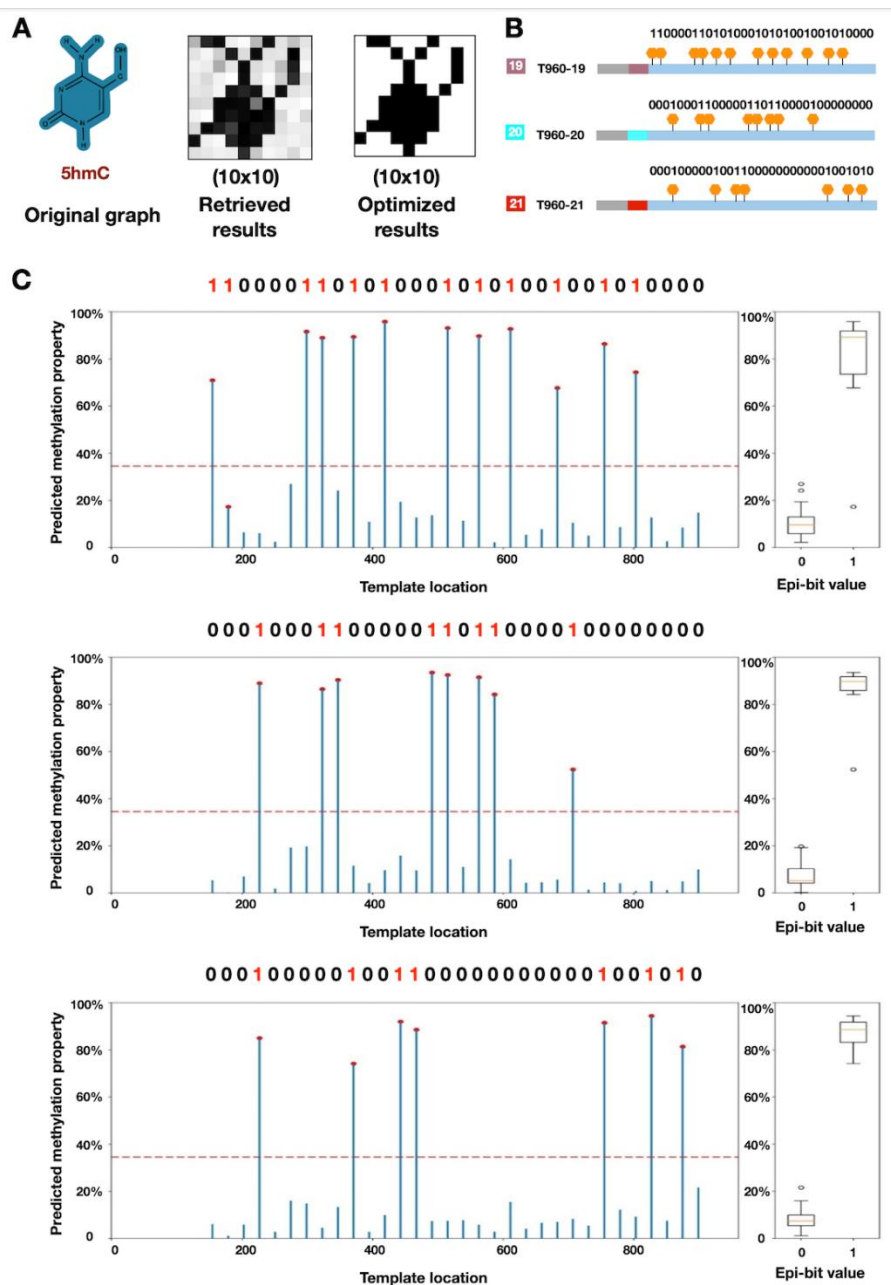


Fig. S35. The parallel storage of the seventh molecule structural image. (A) The original image, the retrieved image based on methylation probabilities and the retrieved image by threshold optimization. (B) The methylation information of the first encoded image, stored in DNA T960 carriers T19, T20, T21 and T25. (C) Methylation detection results obtained from Megalodon of each DNA carrier. Box plots show the distribution of the data in first bar plots (n = 58 for epi-bit 0, n=12 for epi-bit 1). Box plots show the distribution of the data in second bar plots (n = 61 for epi-bit 0, n=9 for epi-bit 1). Box plots show the distribution of the data in third bar plots (n = 62 for epi-bit 0, n=8 for epi-bit 1).

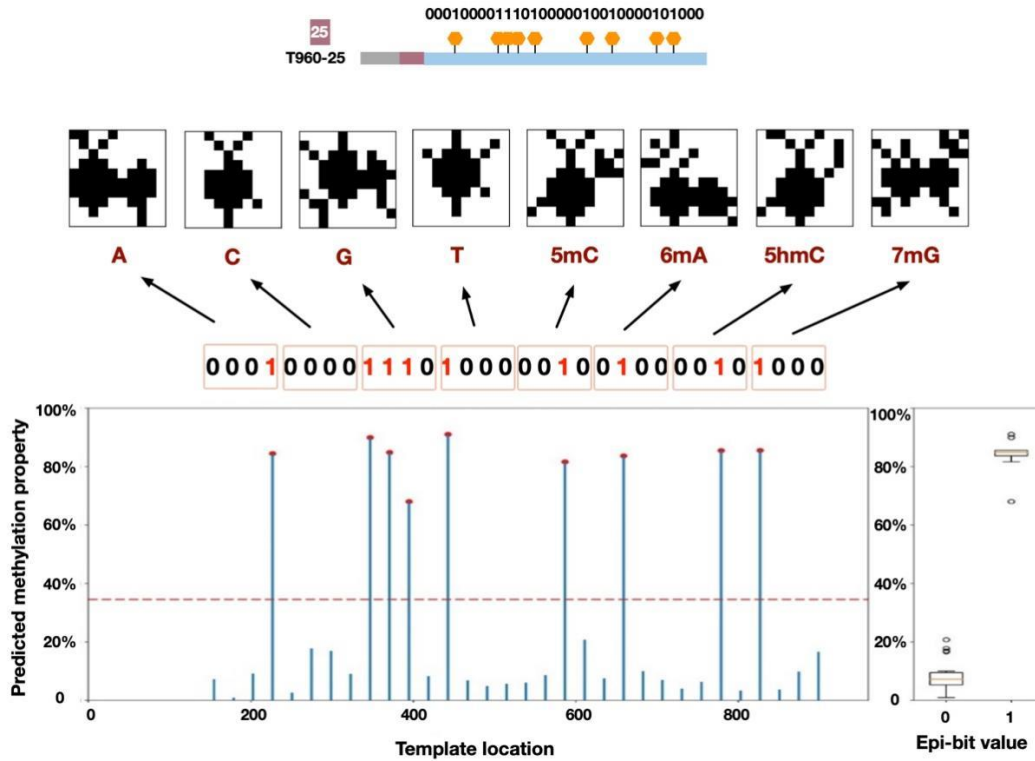


Fig. S37. Methylation calling results in nanopore sequencing of the 25th 960nt DNA template. The 25th carrier contained the rest of the information bits of eight images. Box plots show the distribution of the data in first bar plots ($n = 61$ for epi-bit 0, $n=9$ for epi-bit 1).

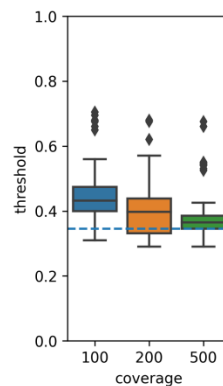


Fig. S38. The repeatability of the thresholds in Fig.4g. We sampled this result for different coverages and calculated the corresponding thresholds based on the sampling result. The thresholds are highly reproducible at high coverages. The horizontal axis represents the sampling coverage, while the vertical axis depicts the calculated threshold value. For each coverage level, there are 100 sample points. Box plots show the distribution of the data in second bar plots ($n = 100$ for each coverage).

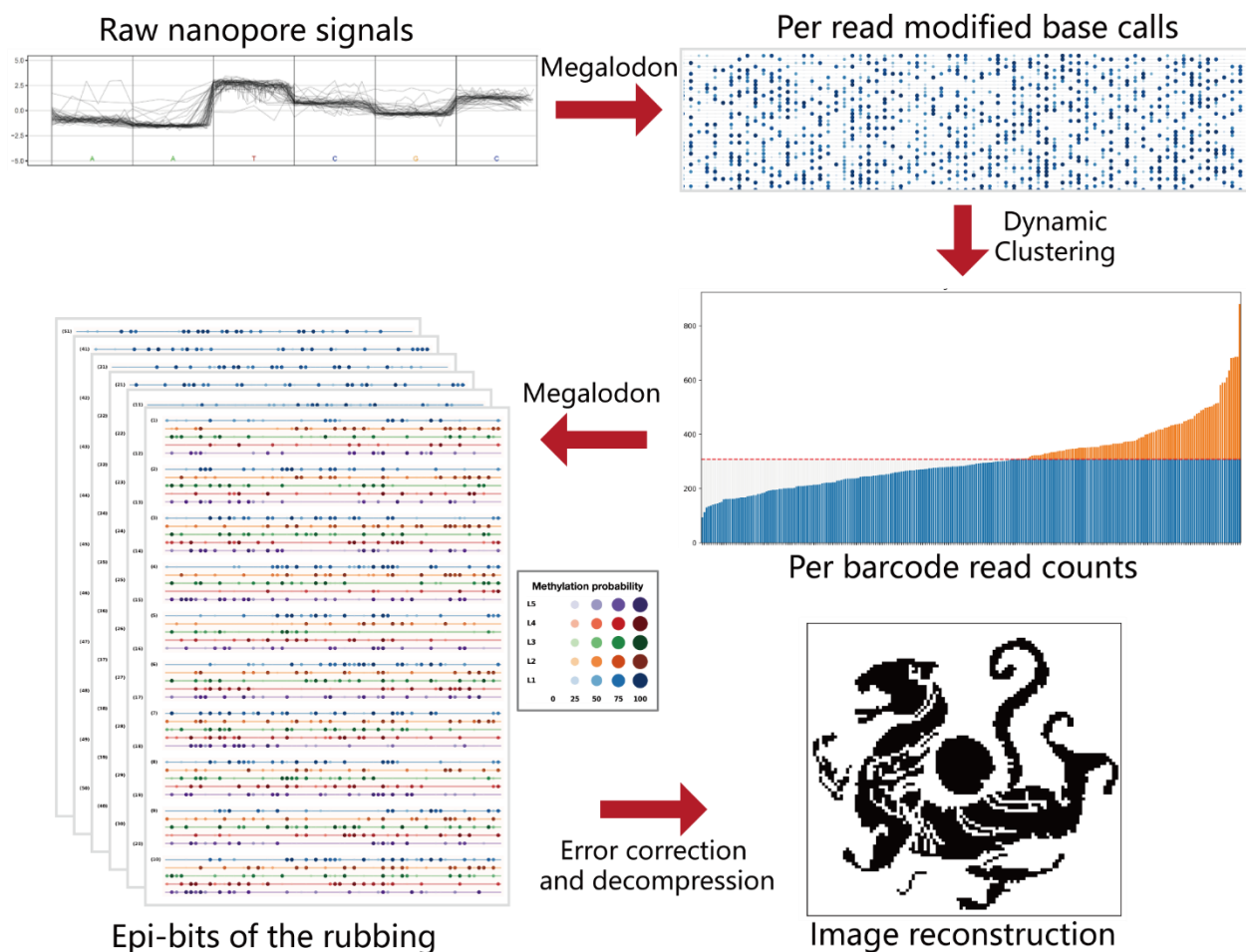


Fig. S39. The workflow of retrieving the methylation information from nanopore sequencing results in the experiments of the large-scale epi-bit storage. The initial outputs of nanopore sequencing are fast5 files with raw electrical signals. At first, we used Megalodon to obtain methylation probability of each modification site on each read. Then we performed Dynamic Clustering (Extended Data Fig. 7) for all reads. After Dynamic Clustering, we performed methylation calling on each cluster individually. Next, we retrieved the epi-bit sequences based on the methylation barcodes. Finally, we reconstructed the image of the tiger rubbing by error correction and decompression based on the retrieved epi-bit sequences.

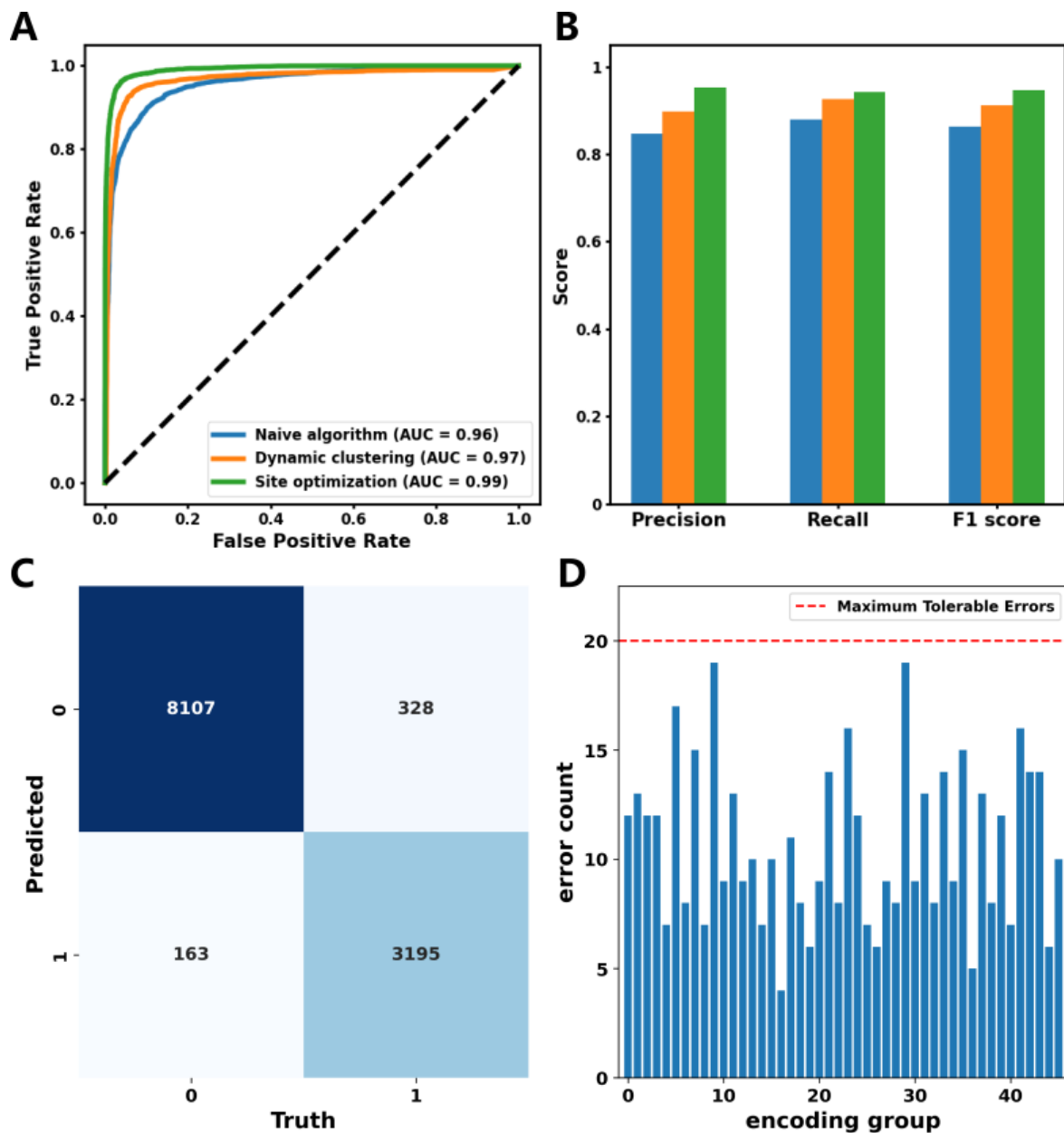


Fig. S40. Data analysis of the four samples of storage of the tiger rubbing image. (A) Comparisons of the ROC curves of Samples 1-3. (B) Comparisons of precision, recall and F1-scores of Samples 1-3. (C) The confusion matrix of Sample 4 with the accuracy of 0.9584. (D) The numbers of errors in Sample 4 for each encoding group. The red dashed line indicated the maximum number of errors tolerable by the error correction code.



Chinese rubbing
 Compressed, error corrected
 Automatic sample (4)
 16833 bits (2.05KB)
 Well 1 to 51

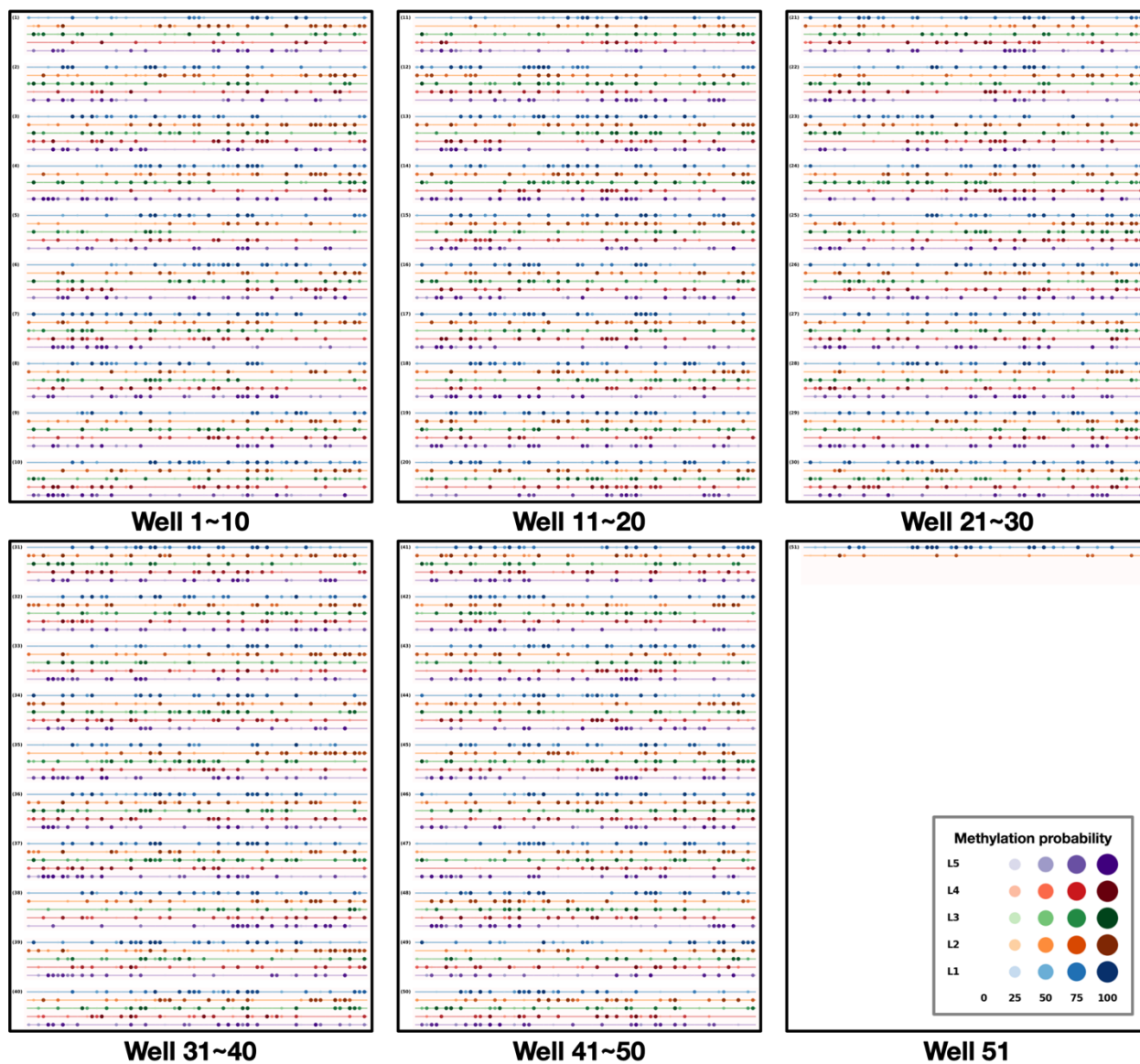


Fig. S41. Overview of Epi-bit calling results of Sample 4, which underwent compression and error correction coding.

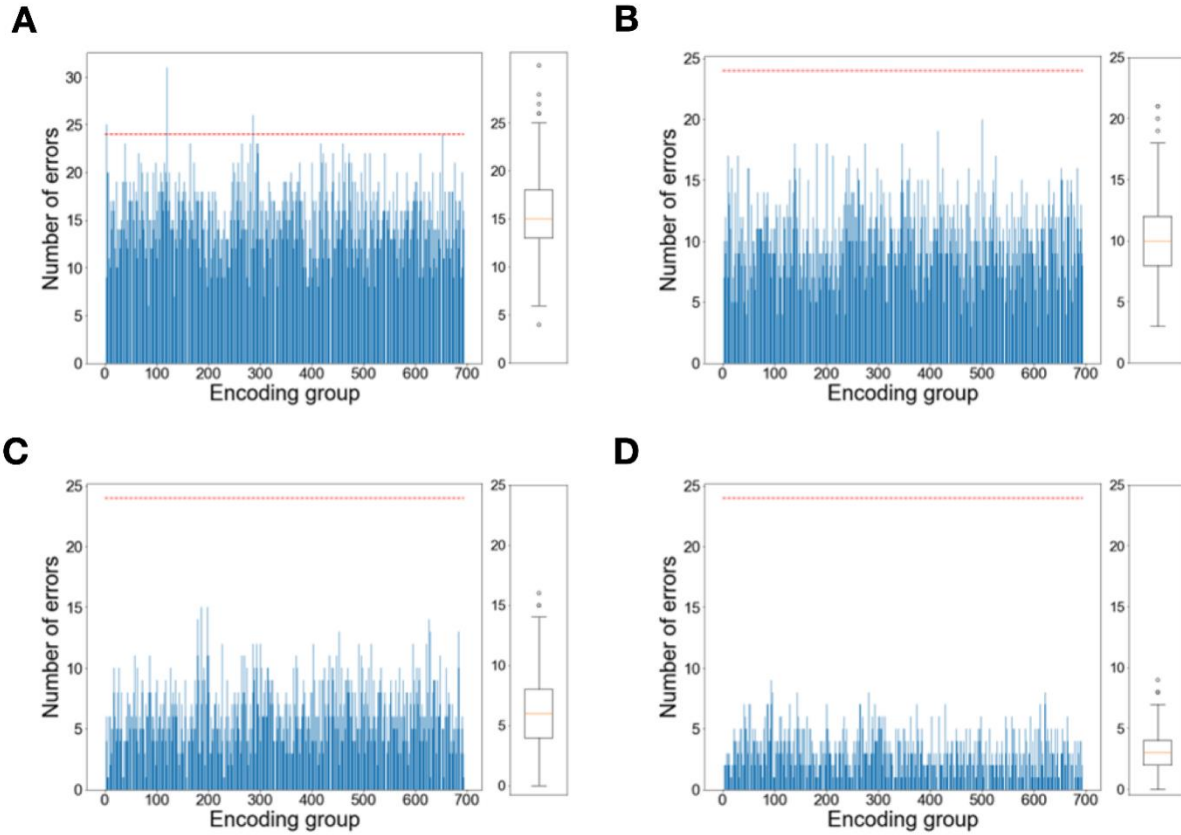


Fig. S42. Simulated decoding of the stored color image (panda). The four subplots illustrate the decoding results of each encoding group under an average accuracy of (A) 93.6%, (B) 95.6%, (C) 97.6% and (D) 99.6%, respectively. The four bar plots show the number of errors in each of the 693 encoding groups. The red dashed line indicates the maximum number of errors that can be corrected by the BCH code. Box plots show the distribution of the data in bar plots ($n = 693$ for each box plot). The central line within each box represents the median, the edges of the box represent the 25th and 75th percentiles (interquartile range), and the whiskers extend to the most extreme data points within 1.5 times the interquartile range. Outliers are indicated by dots.

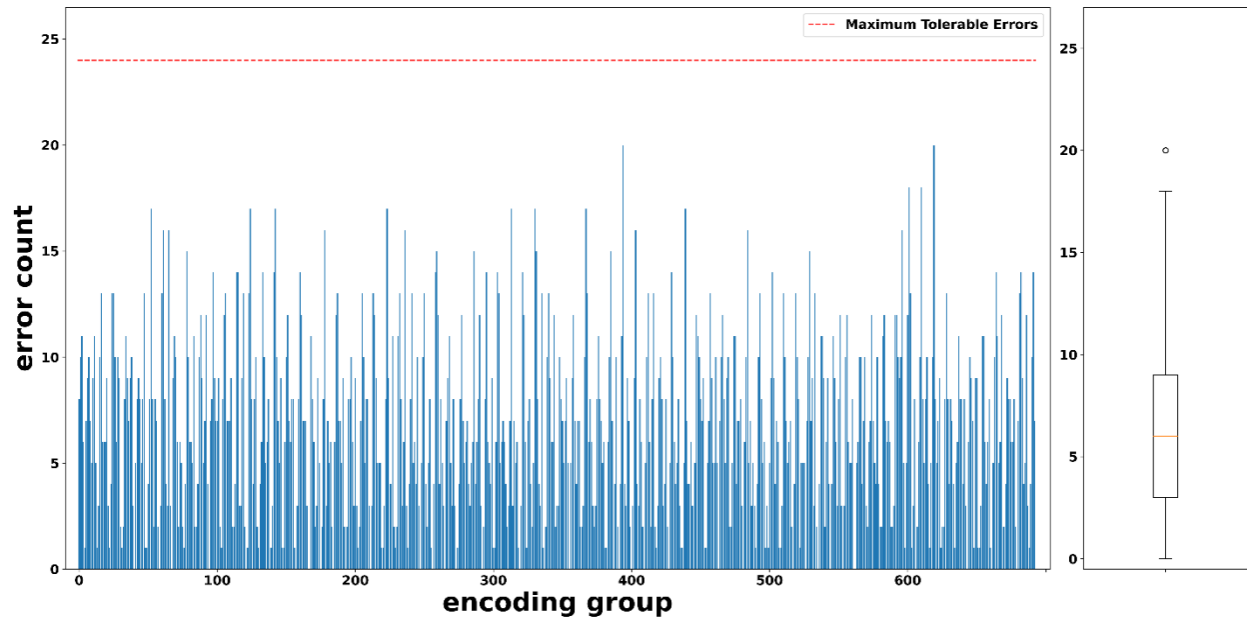


Fig. S43. The numbers of errors in Panda for each encoding group. The Bar plot shows the number of errors in each of the 693 encoding groups. The red dashed line indicated the maximum number of errors tolerable by the error correction code. Box plots show the distribution of the data in bar plots ($n = 693$ for each box plot). The central line within each box represents the median, the edges of the box represent the 25th and 75th percentiles (interquartile range), and the whiskers extend to the most extreme data points within 1.5 times the interquartile range. Outliers are indicated by dots.

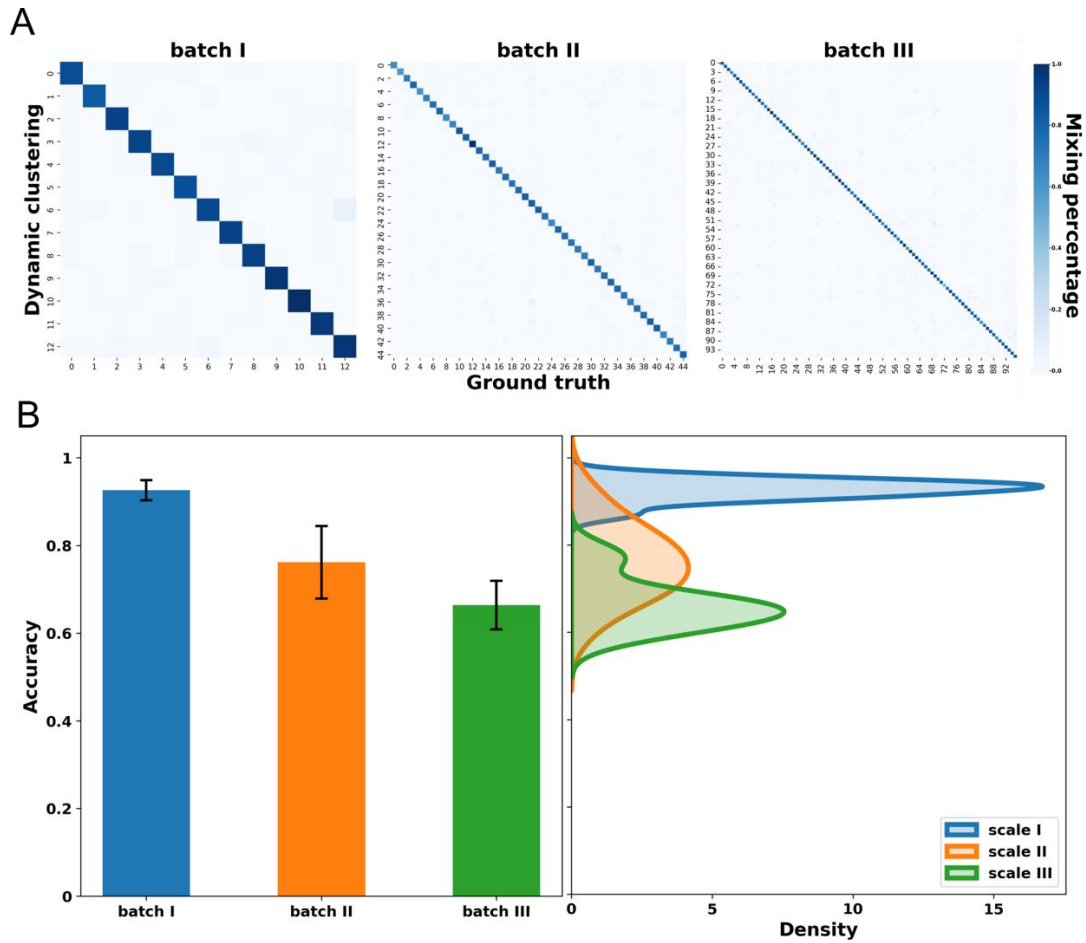


Fig. S44. Comparisons of the data retrieving when using sequencing batches of different sizes. A sequencing batch is one that has a certain number of wells mixed together and then sequenced in one pot. **(A)** Confusion matrices of the mixed sequencing batches as I (13 wells), II (48 wells) and III (96 wells). We took the clustering results of information bits and barcode bits as the ground truth. **(B)** Comparison of the accuracies in I, II and III sequencing batches. Box plots show the distribution of the data in second bar plots ($n = 13$ for batch I, $n=48$ for batch II, $n=96$ for batch III).



Panda

Compressed, error corrected

252504 bits (30.82KB)

Well 1 to 90 (total 756 wells)

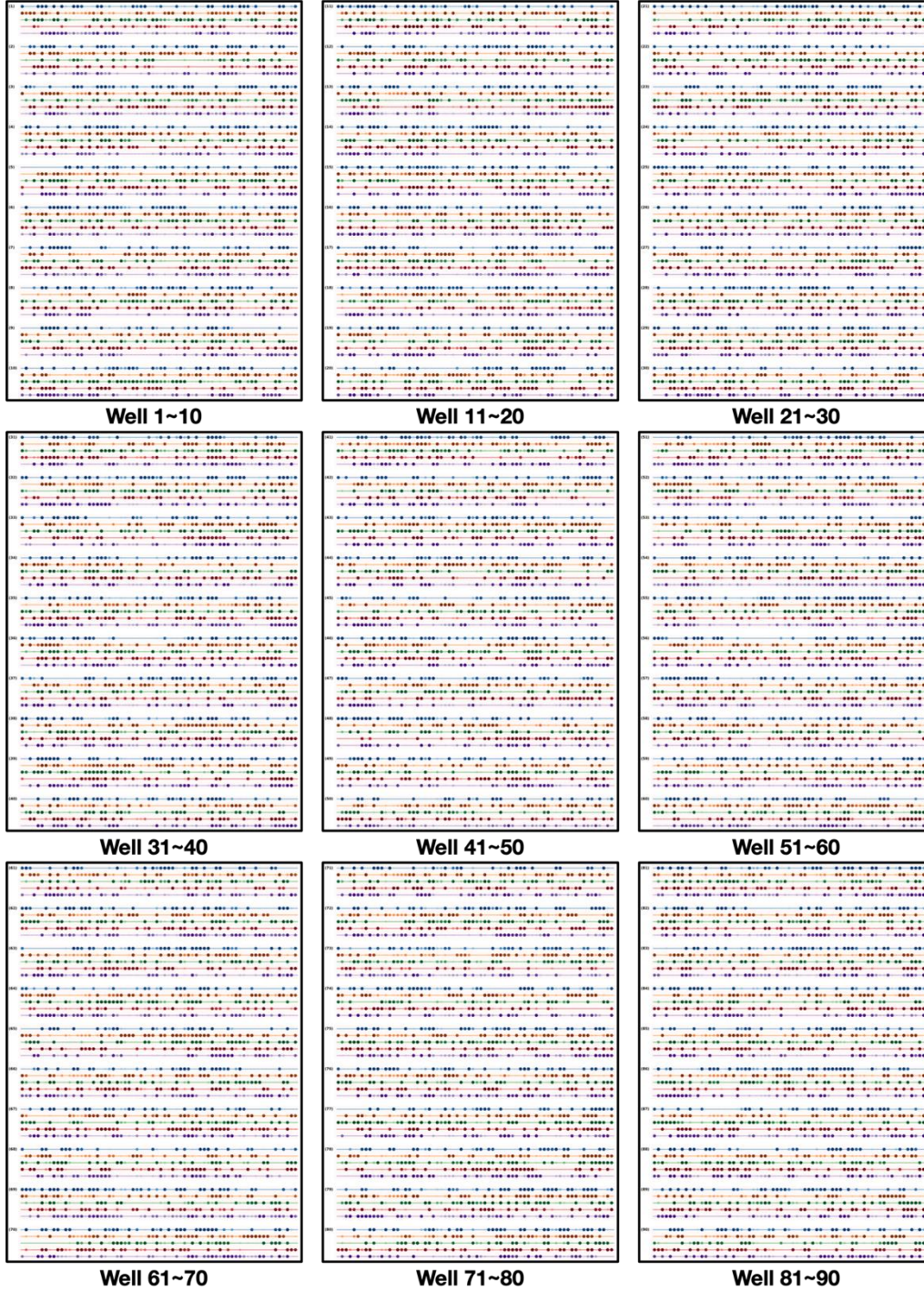


Fig. S45. Retrieved epi-bit streams of the panda image storage experiment, well 1-90.



Panda

Compressed, error corrected

252504 bits (30.82KB)

Well 91 to 180 (total 756 wells)

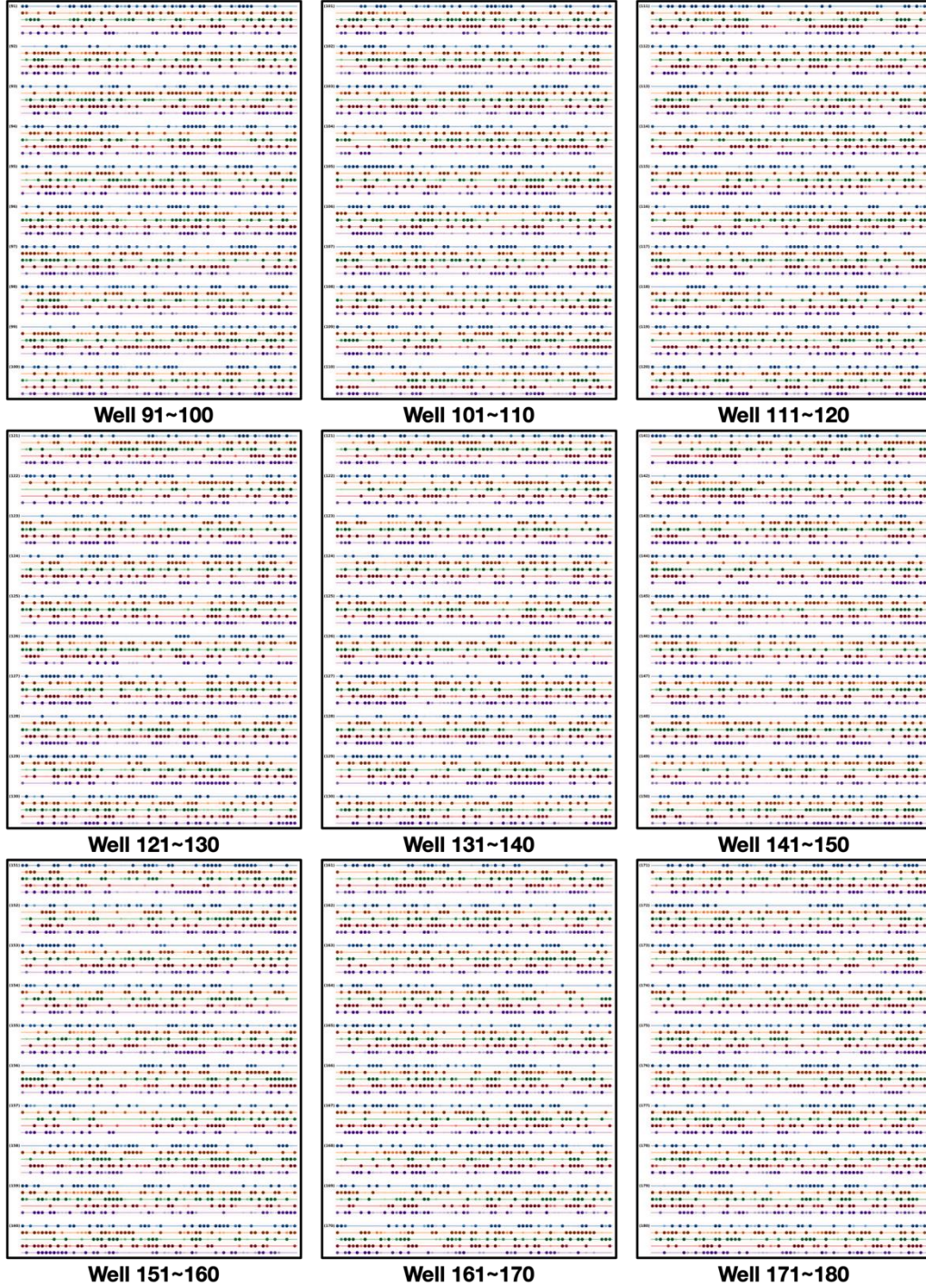


Fig. S46. Retrieved epi-bit streams of the panda image storage experiment, well 91-180.



Panda

Compressed, error corrected

252504 bits (30.82KB)

Well 181 to 270 (total 756 wells)



Fig. S47. Retrieved epi-bit streams of the panda image storage experiment, well 181-270.

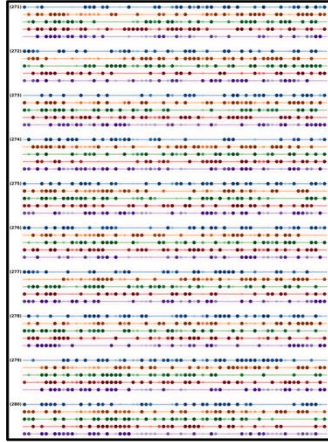


Panda

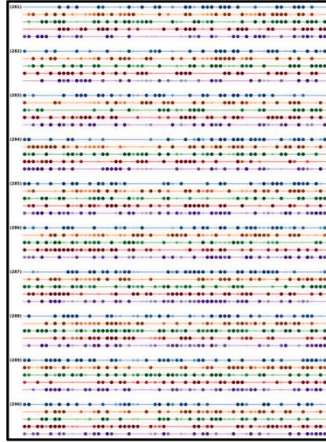
Compressed, error corrected

252504 bits (30.82KB)

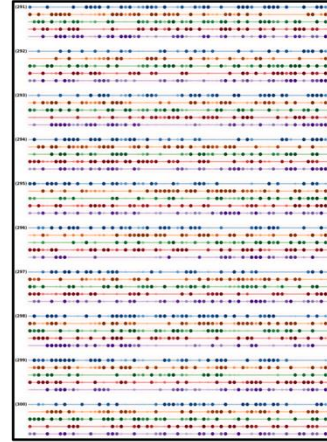
Well 271 to 360 (total 756 wells)



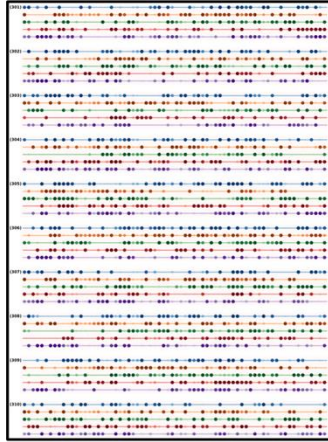
Well 271~280



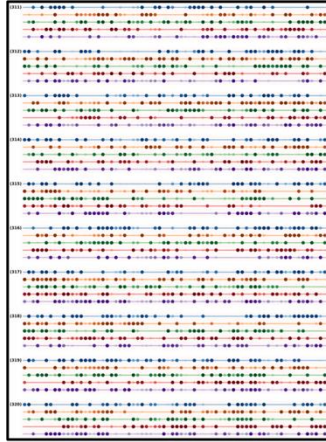
Well 281~290



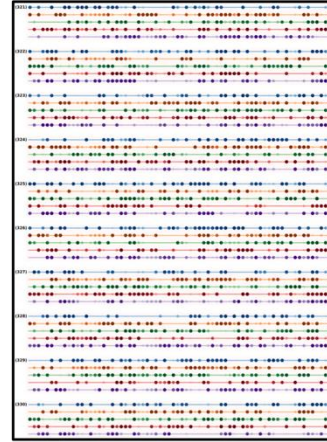
Well 291~300



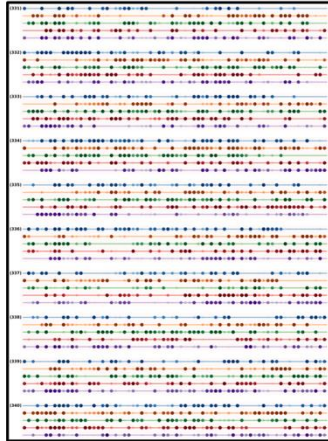
Well 301~310



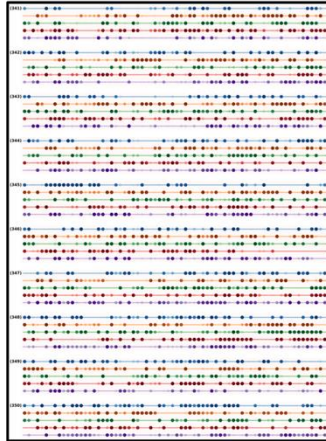
Well 311~320



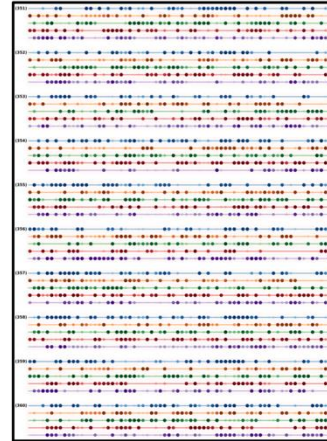
Well 321~330



Well 331~340



Well 341~350



Well 351~360

Fig. S48. Retrieved epi-bit streams of the panda image storage experiment, well 271-360.

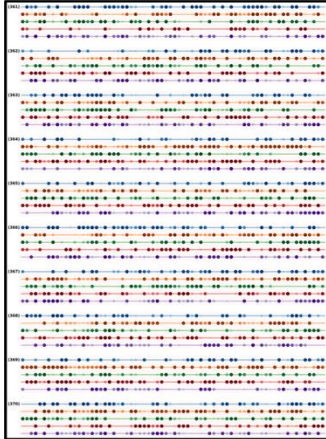


Panda

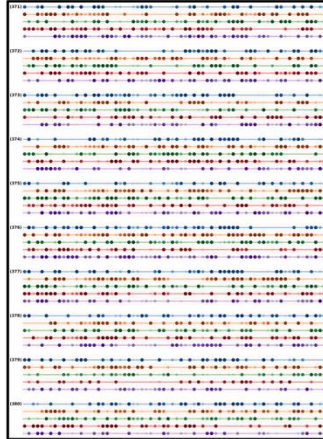
Compressed, error corrected

252504 bits (30.82KB)

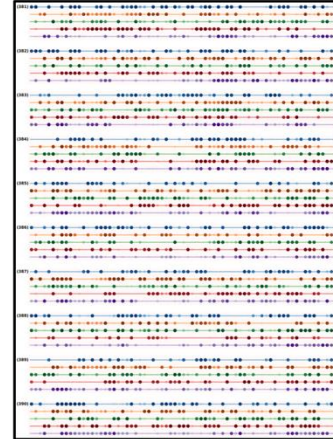
Well 361 to 450 (total 756 wells)



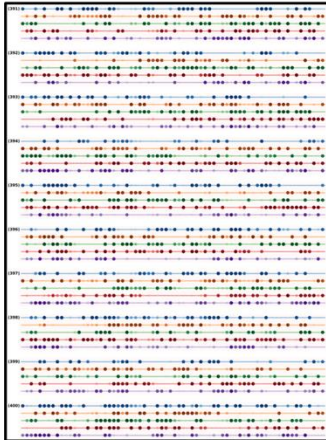
Well 361~370



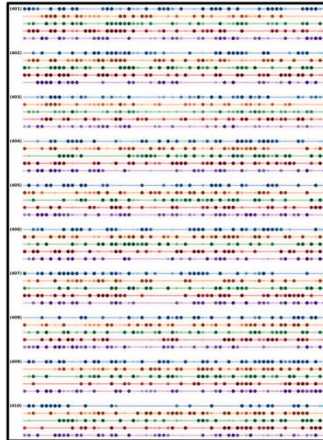
Well 371~380



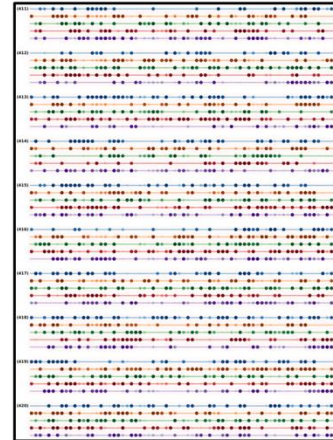
Well 381~390



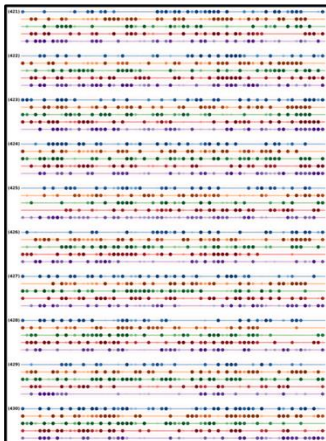
Well 391~400



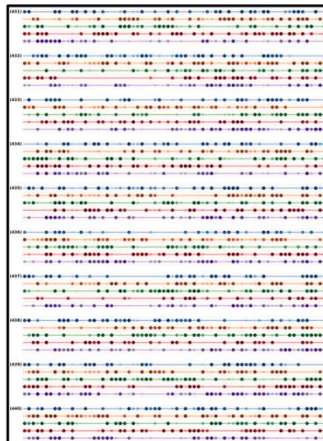
Well 401~410



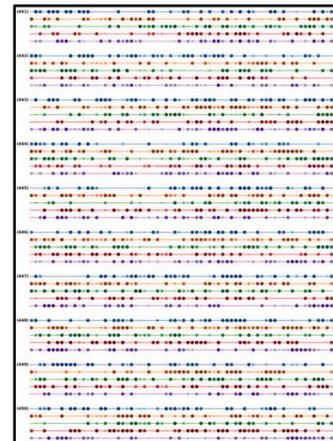
Well 411~420



Well 421~430



Well 431~440



Well 441~450

Fig. S49. Retrieved epi-bit streams of the panda image storage experiment, well 361-450.

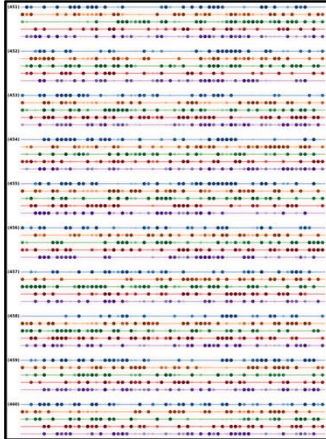


Panda

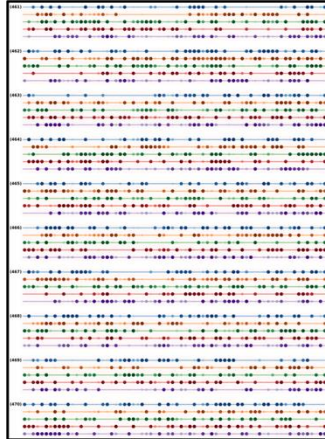
Compressed, error corrected

252504 bits (30.82KB)

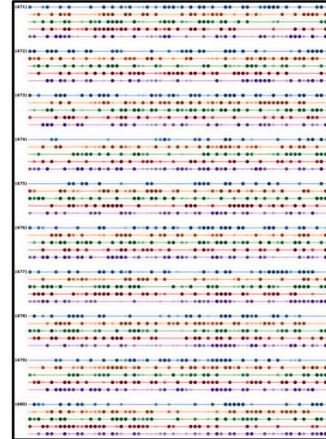
Well 451 to 540 (total 756 wells)



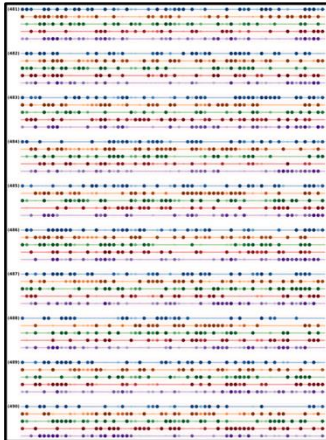
Well 451~460



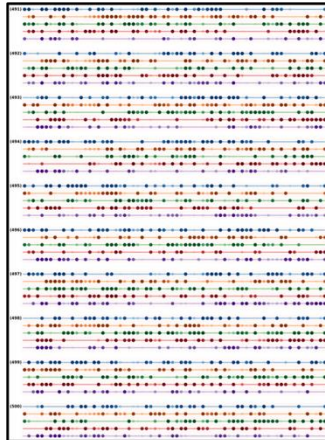
Well 461~470



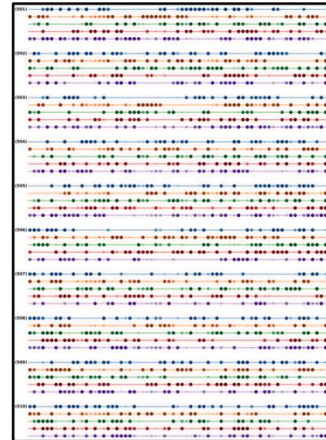
Well 471~480



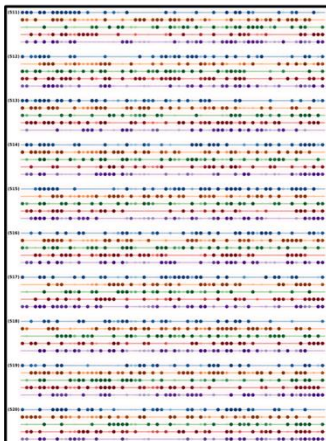
Well 481~490



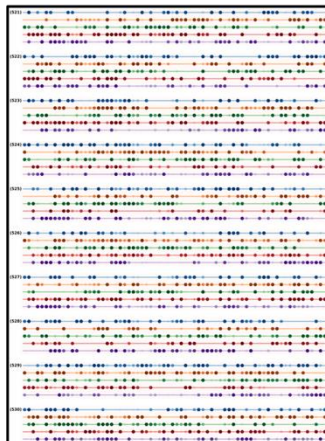
Well 491~500



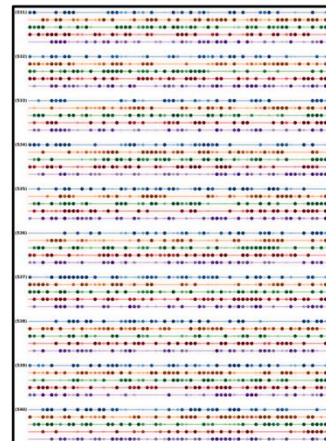
Well 501~510



Well 511~520



Well 521~530



Well 531~540

Fig. S50. Retrieved epi-bit streams of the panda image storage experiment, well 451-540.

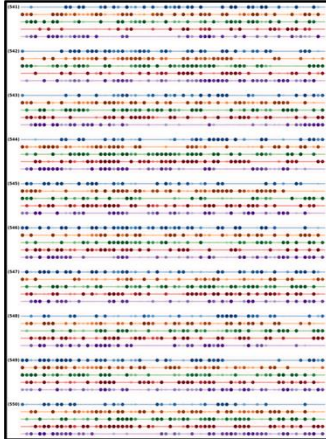


Panda

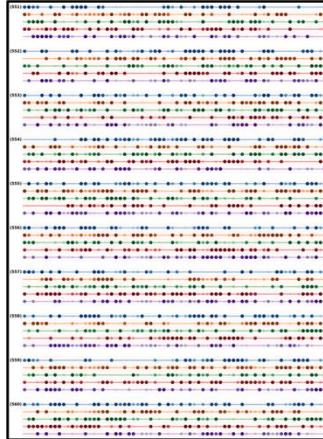
Compressed, error corrected

252504 bits (30.82KB)

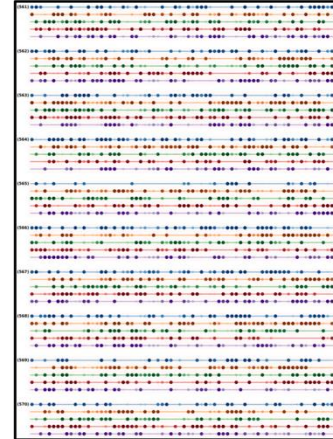
Well 541 to 630 (total 756 wells)



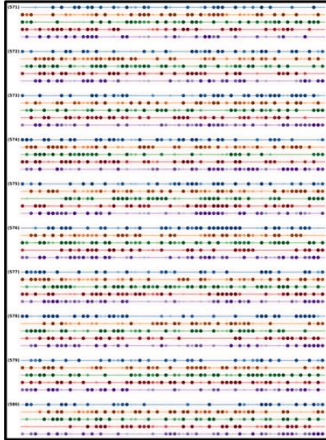
Well 541~550



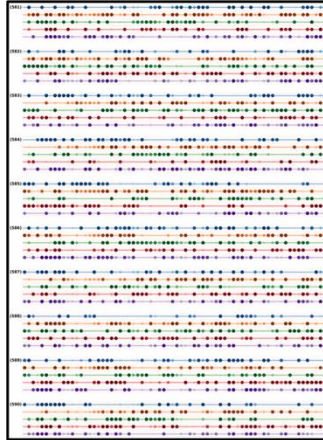
Well 551~560



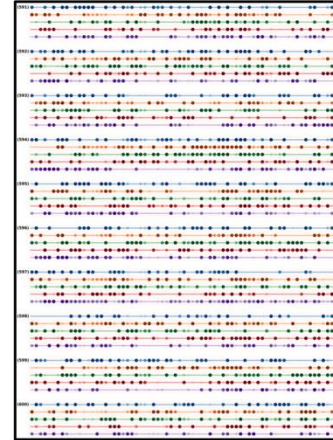
Well 561~570



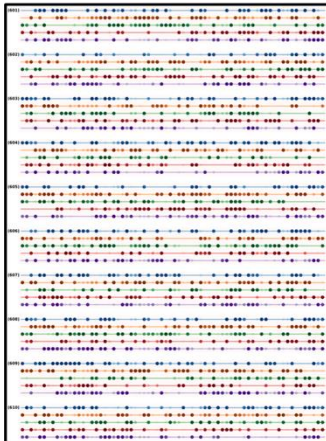
Well 571~580



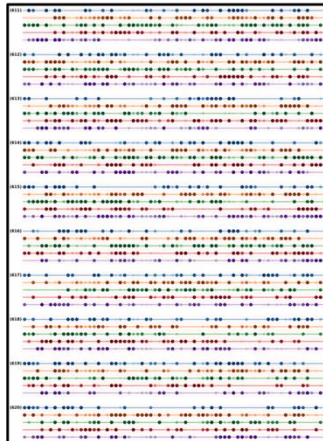
Well 581~590



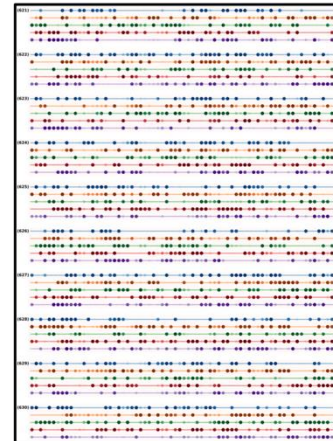
Well 591~600



Well 601~610



Well 611~620



Well 621~630

Fig. S51. Retrieved epi-bit streams of the panda image storage experiment, well 541-630.

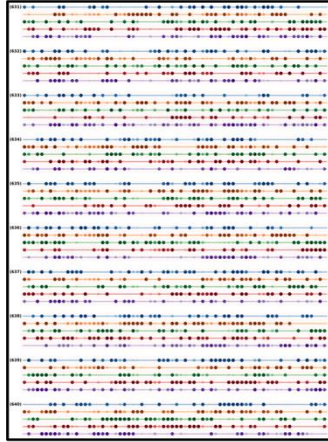


Panda

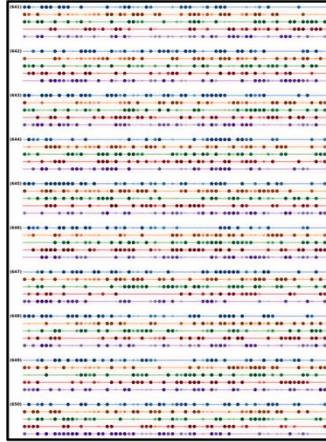
Compressed, error corrected

252504 bits (30.82KB)

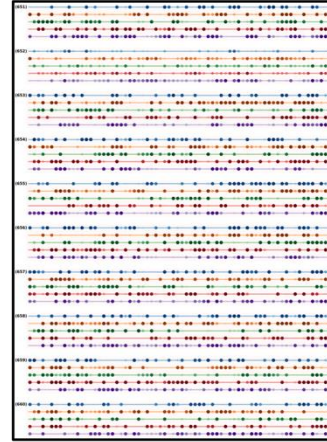
Well 631 to 720 (total 756 wells)



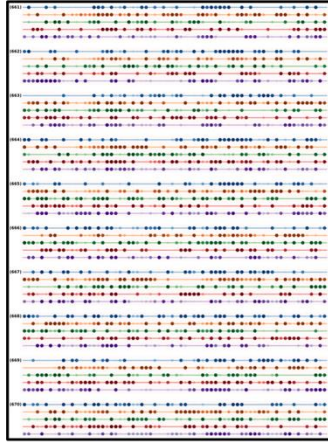
Well 631~640



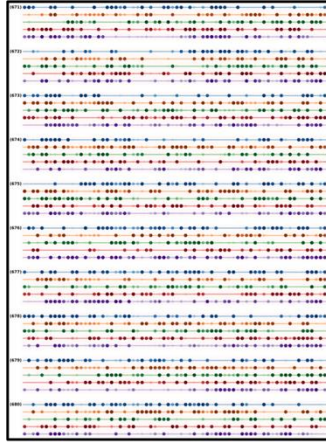
Well 641~650



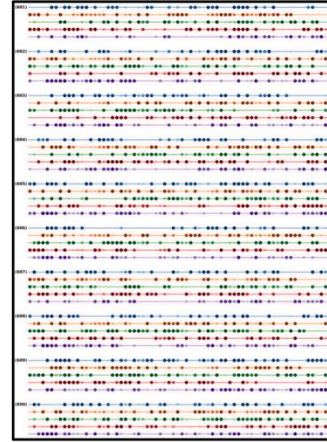
Well 651~660



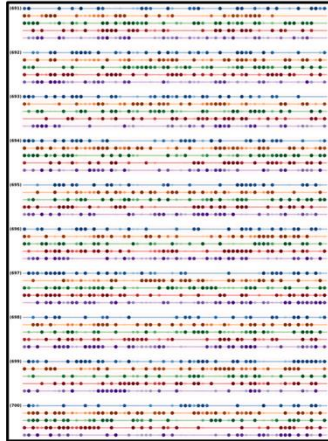
Well 661~670



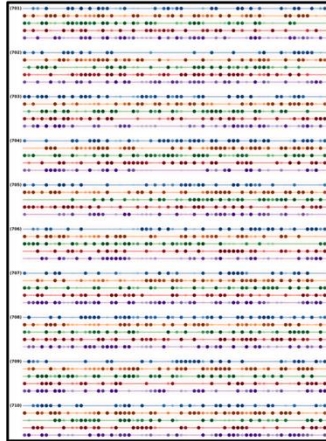
Well 671~680



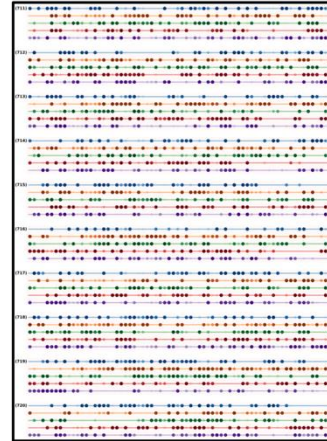
Well 681~690



Well 691~700



Well 701~710



Well 711~720

Fig. S52. Retrieved epi-bit streams of the panda image storage experiment, well 631-720.



Panda

Compressed, error corrected

252504 bits (30.82KB)

Well 721 to 756 (total 756 wells)

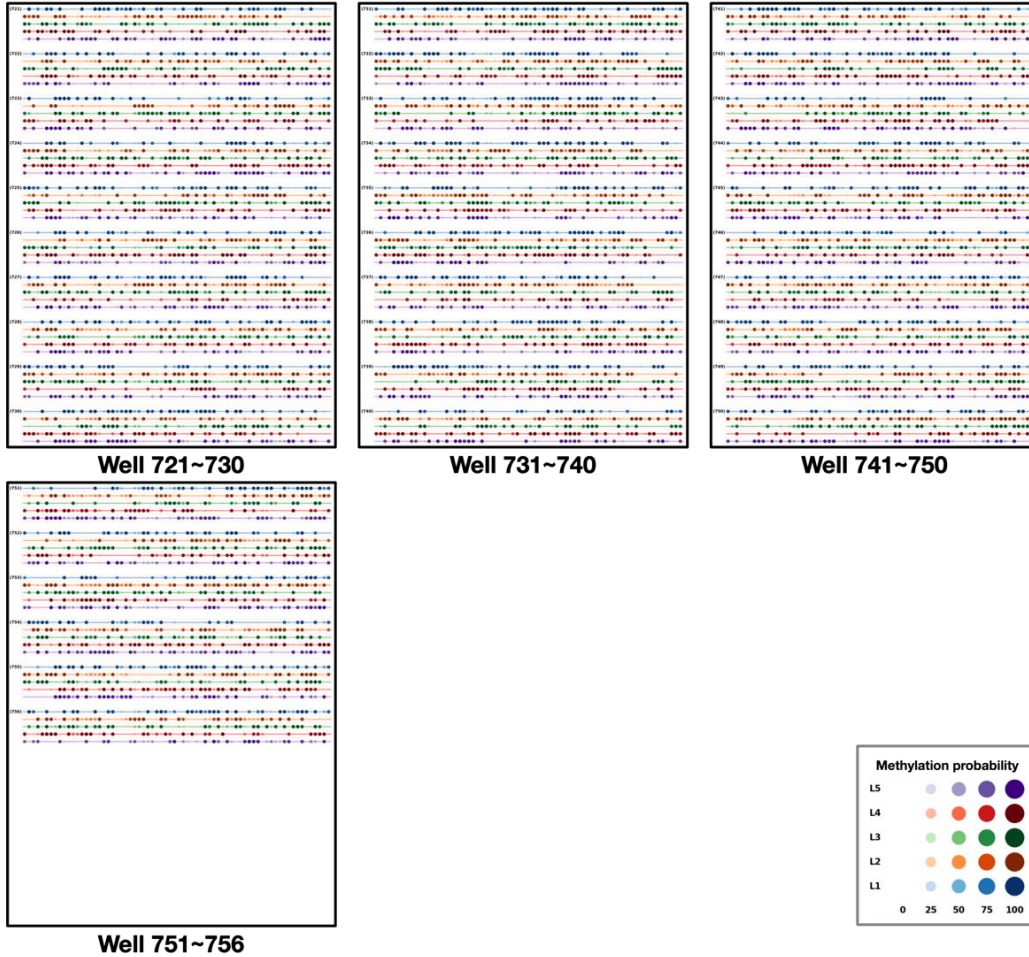


Fig. S53. Retrieved epi-bit streams of the panda image storage experiment, well 721-756.

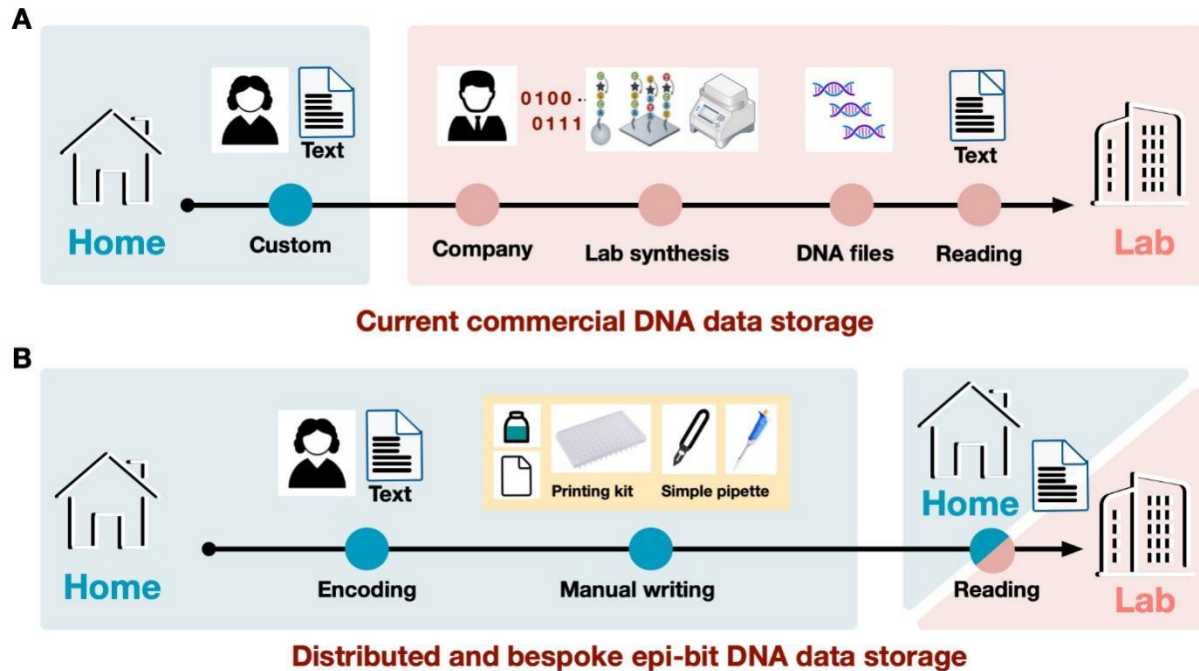


Fig. S54. Comparison of iDNAdrive (A) and *de novo* synthesis based centralized DNA storage (B). In the centralized storage mode, most of experimental processes are performed in professional facilities. The epi-bit based iDNAdrive framework offers an alternative, distributed storage mode, with which the users can write their personal data into DNA molecules on their own, such that the data is undisclosed until sequencing.

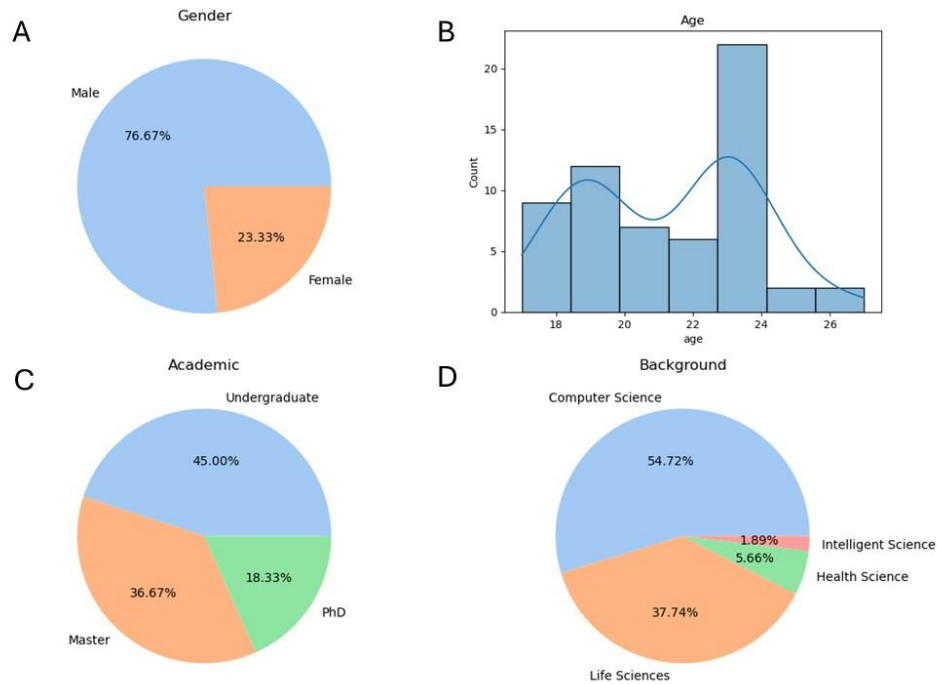


Fig. S55. The demographic analysis of 60 student volunteers involved in iDNAdrive. (A) The gender distribution, with blue representing males (76.67%) and orange representing females (23.33%). **(B)** The age distribution. **(C)** The education stage distribution. Blue represents PhD students (45.00%), orange represents master's students (36.67%), and green represents undergraduate students (18.33%). **(D)** The academic background distribution. Blue represents those with a computer science background (54.72%), orange represents those with a life sciences background (37.74%), green represents those with a health sciences background (5.66%), and red represents those with an intelligent sciences background (1.89%).

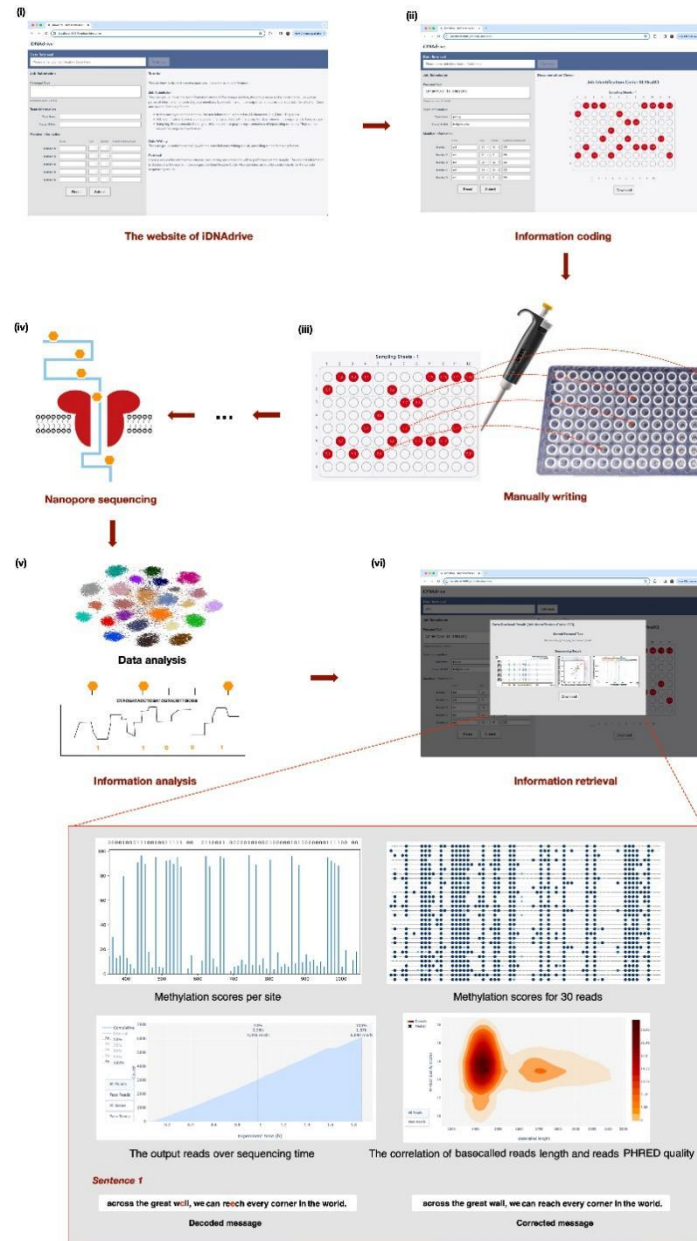


Fig. S56. The experimental pipeline of iDNAdrive. The users firstly upload the text they want to store to the iDNAdrive webserver (step i). Accordingly, the sampling schemes are generated in the form of lattice diagrams, together with a unique passcode (step ii). Then, the users perform sample additions based on the schemes and store the information by epi-bit transfer (step iii). The epi-bit information can be recovered from the DNA samples by nanopore sequencing (steps iv and v). Finally, the users retrieve the stored message on the iDNAdrive webserver by entering the passcode (vi). Also returned to the users are a few graphs showing the decoding process and sequencing quality control statistics.

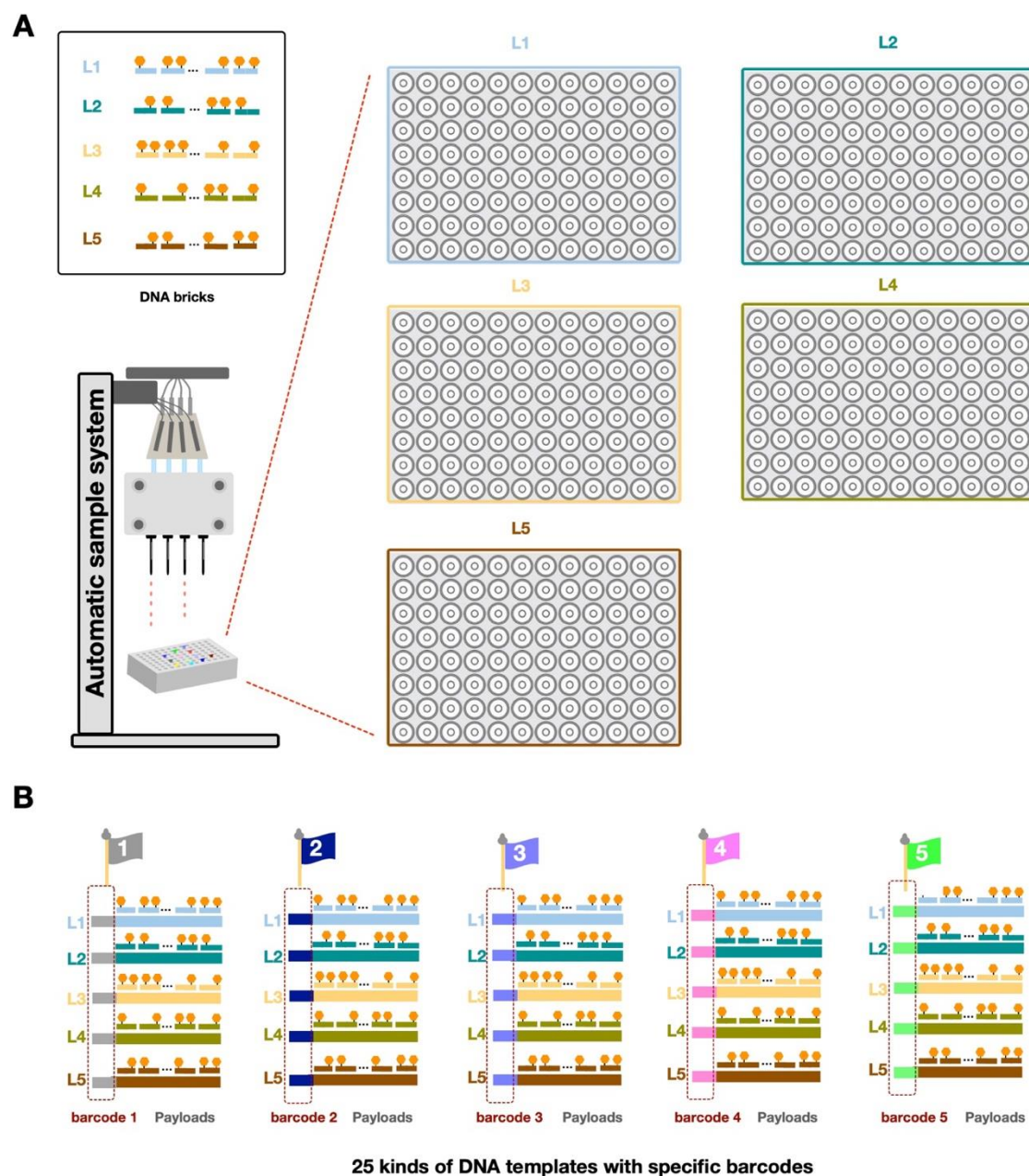


Fig. S57. Preparation of the epi-pit data writing kit and the illustrations of DNA templates used in iDNAdrive. (A) An automatic liquid handling platform was employed to dispense the full set of 700 DNA bricks (corresponding to templates L1-L5) into the writing kits. (B) 25 kinds of DNA templates with specific barcodes were used in iDNAdrive.

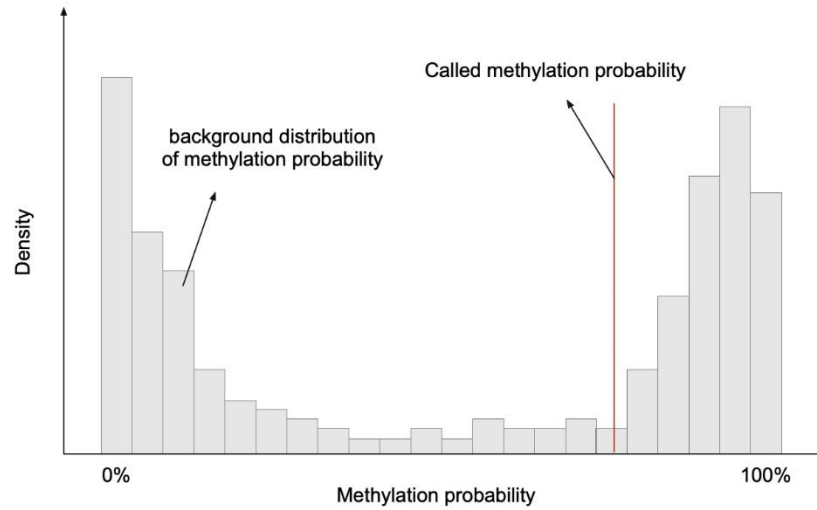
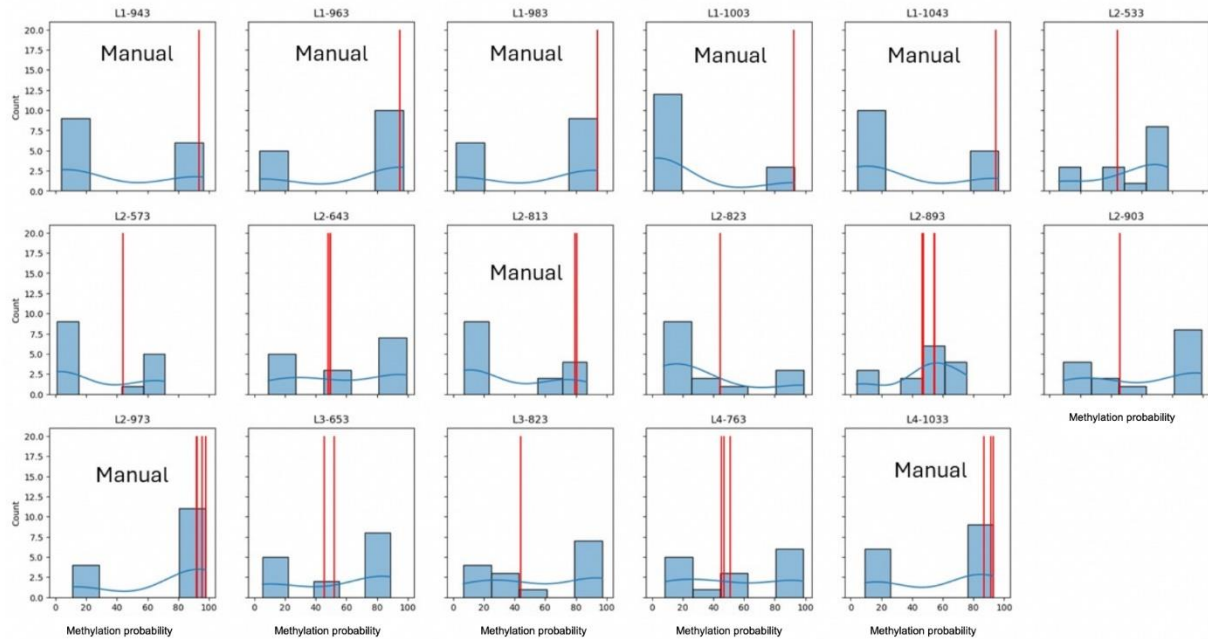
A**B**

Fig. S58. Modes of 0-to-1 substitution errors in iDNAdrive. (A) Schematic of the error mode. All predicted methylation probabilities at an epi-bit site on a template form a background bimodal distribution. The red line(s) shows the called methylation probability at the epi-bit site in a specific storage realization. (B) Error modes of all 0-to-1 substitution errors occurred in iDNAdrive. Suspicious manual errors are annotated in the subplots. The error is deemed highly likely to be caused by manual sampling mistakes if the background distribution is clearly bi-modal and the called methylation probability resides in the peak of high probabilities.

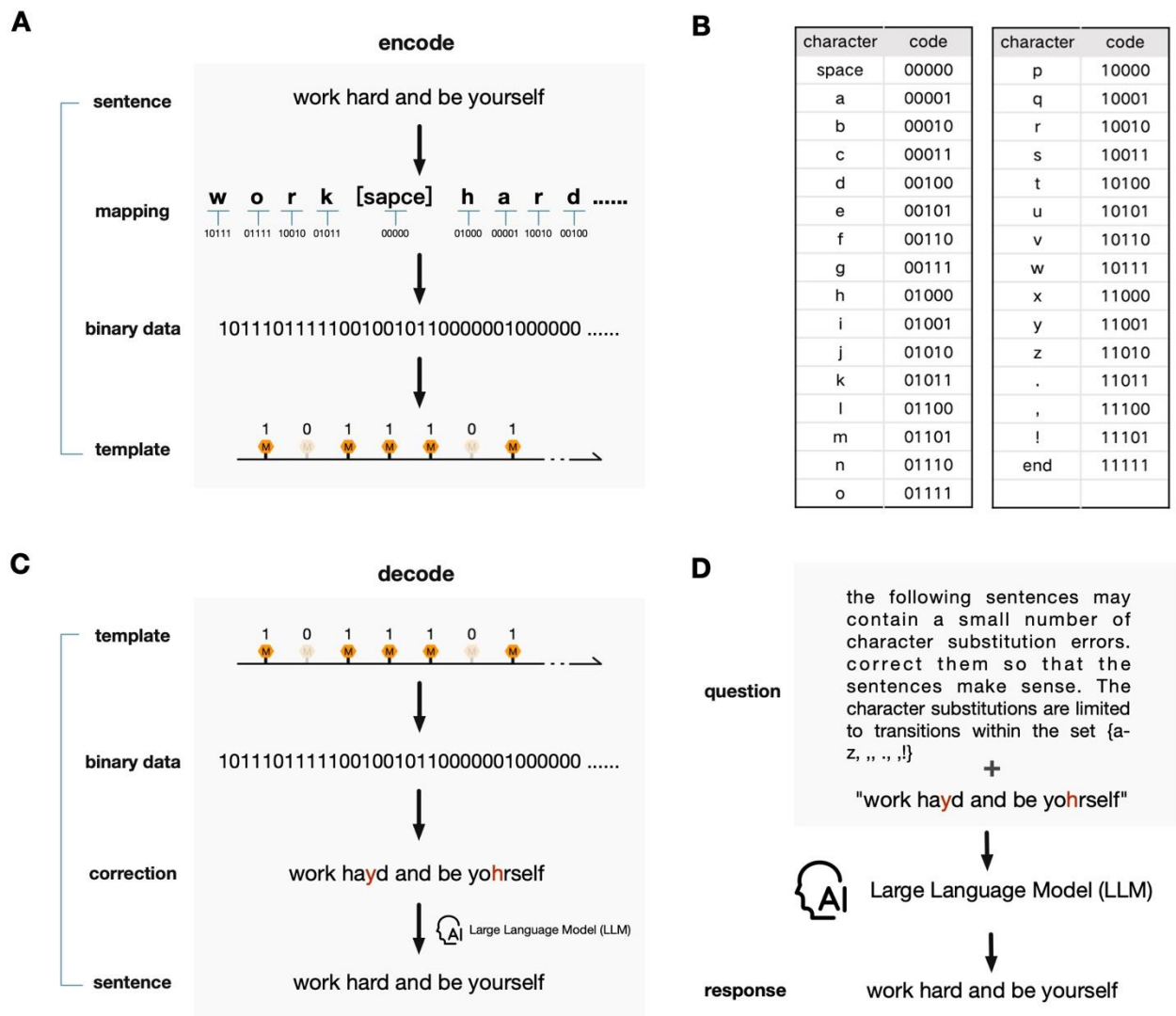


Fig. S59. Encoding, decoding and spelling error correction in iDNAdrive. (A) and (B) Message encoding. Each allowed character is encoded by 5 two-binary numbers. (C) and (D) Message decoding and error correction. The errors can be corrected by AI assisted spelling error correction.

Sentence 1		across the great wall, we can reach every corner in the world.
	✓	across the great wcll, we can reech every corner in the world.
	✗	across the great wall, we can reach every corner in the world.
Sentence 2		a strong man will struggle with the storms of fate.
	✓	a strong man will stbugglm withathe storms gf fate.
	✗	a strong man will struggle with the storms of fate.
Sentence 3		success grows out of struggles to overcome difficulties.
	✓	success grows outlof s.ruogles to overcome diffmculties.
	✗	success grows out of struggles to overcome difficulties.
Sentence 4		fought it out in the debris, now we know that life will change.
	✗	fought it out in the decris, now we know that lmfe will change.
	✗	fought it out in the crisis, now we know that life will change.
Sentence 5		from robots to rockets, build the future you dream of.
	✓	from robots to roakeds, bild the guture yku dream og.
	✗	from robots to rockets, build the future you dream of.
Sentence 6		may your love soar on the wings of a dove in flight at pku
	✗	lay your love soap on .he wingspof a dove if flmghd at pku
	✗	lay your love upon the wings of a dove in flight at play.
Sentence 7		entities should not be multiplied unnecessarily.
	✓	entities should nmt be oultipliud unnecessarily.
	✗	entities should not be multiplied unnecessarily.
Sentence 8		it is well known that peking university is the best in china!!!
	✓	it is well known thad pgking univessity is the best in china!!!
	✗	it is well known that peking university is the best in china!!!
Sentence 9		let life be beautiful like summer flowers and death like autumn
	✓	let life be beautiful .kkm stmmur flowers and deat[] like autumn
	✗	let life be beautiful like summer flowers and death like autumn
Sentence 10		have nothing to lose conquer from within easier said than done
	✗	have nothing to nmse c*nq!er from within easier saidathan done
	✗	have nothing to lose can conquer from within; easier said than done.
Sentence 11		blessed be the lord god from now and unto all eternity.
	✓	blessed be the lord god from now aod unto all eterniuy.
	✗	blessed be the lord god from now and unto all eternity.
Sentence 12		curiosity is your compass. explore the cutting edge!
	✓	curiosity is youv compass. explore the cutpajg edge!
	✗	curiosity is your compass. explore the cutting edge!
Sentence 13		science and tech, your mind can unlock incredible things.
	✓	science andu*ech* ,mur oind canaunlock incredible dhings.
	✗	science and tech, your mind can unlock incredible things.
Sentence 14		college is just the beginning,the adventure of your life awaits
	✓	college is just thg begknning,the adventura kf loub life awaits
	✗	college is just the beginning,the adventure of your life awaits
Sentence 15		tech transforms lives. be the spark of change.
	✓	tech transforms livec. be the spark of chaige.
	✗	tech transforms lives. be the spark of change.

Fig. S60. The recovered texts in iDNAdrive. GPT-4 was used for spelling error correction.

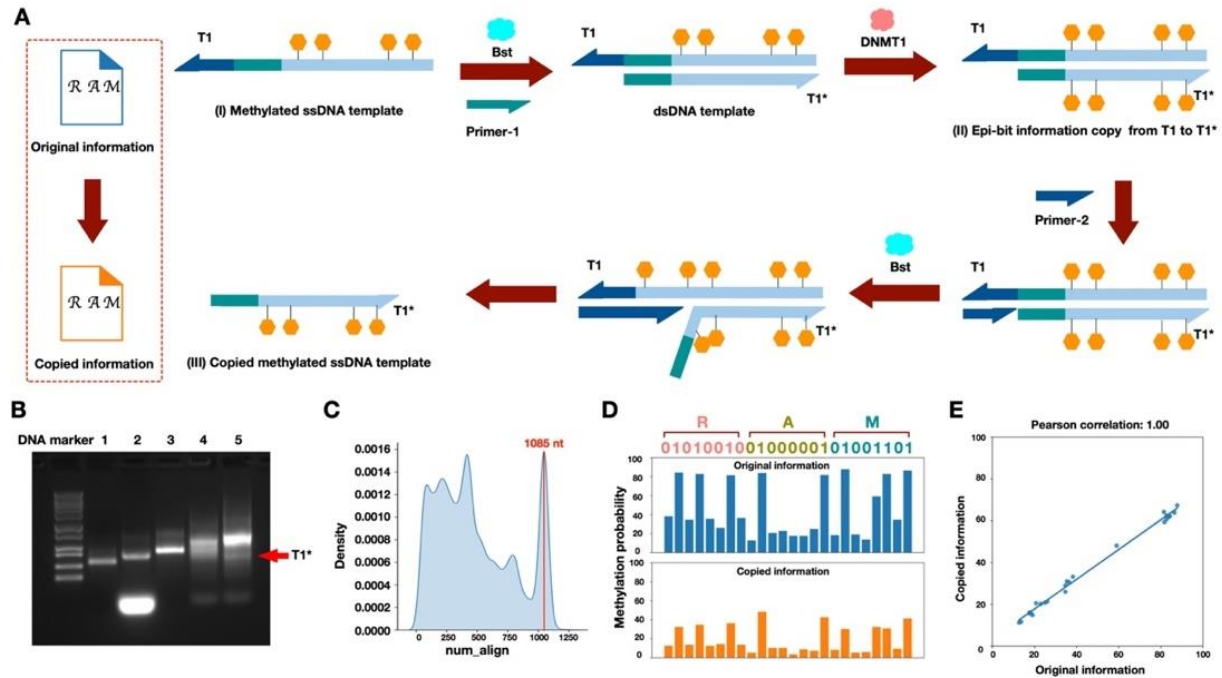


Fig. S61. Epi-bit information copy. (A) Schematic of the epi-bit information copy. (B) Gel results of the information copy. Lane 1: T1; lane 2: T1 + bricks; lane 3: T1 + bricks (Kit recovery); lane 4: T1 + bricks (Kit recovery) + primer-1 + bst + primer-2; lane 5: T1 + bricks (Kit recovery) + primer-1 + bst + primer-2 + bst. (C) The distribution of sequencing read lengths. (D) The comparison of called methylation probabilities along the original templates and the copied templates. (E) The correlation of epi-bit methylation probabilities between the original and the copied information.

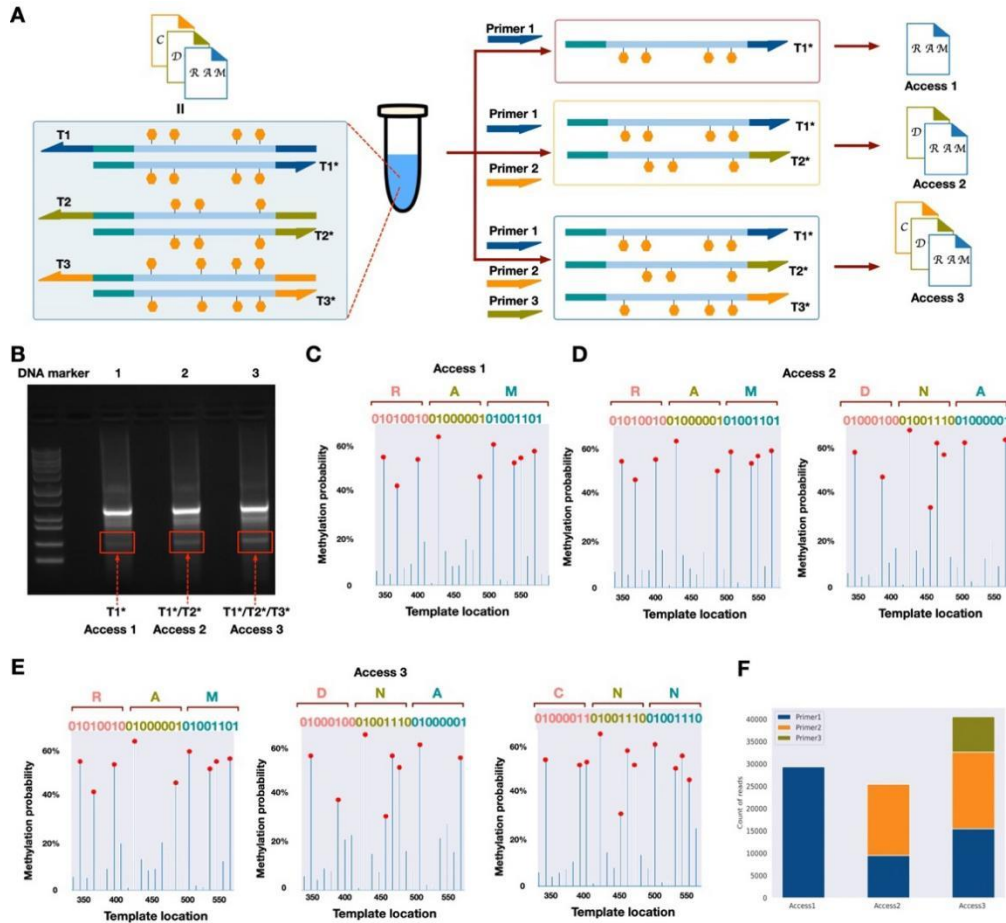


Fig. S62. Random access of epi-bit information. (A) Schematic of the epi-bit random access. (B). Gel results of the random access. Lane 1: Accessing epi-bit information on T1 through Primer 1; lane 2: Accessing epi-bit information on T1 and T2 through Primers 1 and 2; lane 3: Accessing epi-bit information on T1-T3 through Primers 1, 2 and 3. (C) to (E). The information decoded from retrieved reads in the three random access experiments. (F) The proportions of reads retrieved in the three random access experiments that aligned to Templates T1*-T3*.

Supplementary Tables

Table S1. List of alternative unconventional DNA storage schemes

Researches	Years	Data writing	De novo/Synthesis-free writing	Information bits	Total size of data stored	Data reading
This work	2023	Enzymatic methylation transfer	Synthesis-free	DNA modification	275000 bits	Nanopore
Roquet et al. bioRxiv.	2021	Assembling pre-fabricated DNA fragments	Synthesis-free	DNA fragments	195840 bits	Nanopore
Fan et al. Nat. Biotechnol.	2021	Mirror-image <i>Pfu</i> DNA polymerase	De novo	L-DNA sequence	4368 bits	Chemical sequencing
Mayer et al. Angew. Chem. Int. Ed.	2016	Chemical modification transformations	Synthesis-free	DNA modification	1380 bits	Illumina
Sadremomtaz et al. Nat. Commun.	2023	<i>In vitro</i> CRISPR base-editing reactions	Synthesis-free	DNA mutation	1250 bits	Nanopore
Lee et al. Sci. Adv.	2022	Poly(l-lactic-co-glycolic acid)s (PLGAs)	De novo	Polymers	896 bits	Mass spectrum
Zhang et al. Nat. Commun.	2019	DNA origami cryptography (DOC)	Synthesis-free	Braille-like streptavidin patterns	512 bits	Atomic force microscopy
Xu et al. ACS Appl. Mater. Interfaces	2023	Assembly of DNA blocks	Synthesis-free	DNA sequence	288 bits	Illumina
Lee et al. Nat. Commun.	2020	α -hydroxy acids, poly (phenyllactic-co-lactic acid) (PcL)	De novo	Polymers	128 bits	Mass spectrometry
Tabatabaei et al. Nat. Commun.	2020	DNA punch cards	Synthesis-free	Nicks of double-stranded DNA	126 bits	Illumina
Chen et al. Nano Lett.	2019	DNA carriers with attached DNA hairpins	Synthesis-free	DNA hairpin	112 bits	Solid-State Nanopores
Lee et al. Nat. Commun.	2019	Enzymatic DNA synthesis	De novo	Transitions between non-identical nucleotides	96 bits	Illumina and Nanopore
Doroschak et al. Nat. Commun.	2020	Molecular tag mixture	Synthesis-free	DNA molecular bits (molbits)	96 bits	Nanopore
Ouahabi et al. Nat. Commun.	2017	Automated phosphoramidite chemistry	De novo	Polymers	64 bits	Mass spectrum
Kawabe et al. Nat. Commun.	2023	Polymerase DNA amplification	De novo	Supernumerary DNA (Xenonucleic acid (XNA))	8 bits	Nanopore
Cao et al. Sci. Adv.	2020	Phosphoramidite chemistry	De novo	Polymers	6 bits	Nanopore
Wang et al. PNAS	2023	DNA strand displacement	Synthesis-free	DNA mutations	4 bits	Illumina

Table S2. Error analysis for epi-bit storage

Error source		Error pattern	Solution (experimental, algorithmic)
Writing	DNMT1 methylation efficiency	Random	DNMT1 concentration More efficient DNMT1
	DNMT1's inefficiency at certain sequences	Context dependent	Brick design Site selection*
Reading	Overall methylation calling errors	Random	Sequencing depth* Better algorithm for methylation calling PacBio sequencing
	Methylation calling at specific sequence contexts	Context dependent	Brick design Site specific thresholding* Site selection*
	Alignment of templates	Random barcode switch	Dynamic clustering* Better ligation protocols for DNA barcodes

* These methods were explored in the study.

Table S3. The DNA sequences used in the experiments

Name	Sequence
Brick strands complementary to template 960	
960-1	CCACAATCTTGCGTTAGGTGGAACCTCTCCAGCTTCGGATA GGAAGCA
960-2	TGACTAGAAGTCGTGTCGGCCATT
960-3	AGTCATACTGTCGCCGGTGCAGTC
960-4	AACTAGAACTTCGTCTGTTGAGCGC
960-5	AACAATTCTTGCGATTAGGCGGCC
960-6	CAACAACCTTGACGGAGTTGGCTTC
960-7	TATGTATTACTCGTCACGGAATGC
960-8	CTCTCAATTCTCGCATAAGACGTG
960-9	ACCTCTATAGACGATGGTGAGCTC
960-10	CACTCATATCACGTCAATGGCAGG
960-11	GTAATACTCTTCGGTGGAGCCGAC
960-12	CCACCAACAATCGTATTGTGGTCA
960-13	CGCTTGCTTAACGGCAAGGCTAAT
960-14	ACCTCTACTGTCGTAGAGGCAGAT
960-15	CACACTCCATTCTGAAGTGGAGGTT
960-16	ATTATTCAAGGCGTCGAAGCCGCT
960-17	CTACATACTTGCGTAGTAGCACGG
960-18	CAATAACTCTCCGGTTAGGTGAGC
960-19	CCATATTCCATCGCCGTGGAGTAG
960-20	CCTCACAATTACGTTAGTGCATGA
960-21	CTCTCGTGTAACGAAGGTGTCACA
960-22	ACTTCCATTAGCGGTCGTGTCTAC
960-23	ACATCCGATGTCGATGAGGTATCC
960-24	ATCAATTAAGGCGACTGTTCCGCG
960-25	CGCATTCGGACCGTAATGGTAATC
960-26	CAAGAGTATTGCGCAGTTGCCTCA
960-27	CACTAACTACACGTAGCTGTGGTG
960-28	CTACAGACGTTCTGAAGAGGTTTAC
960-29	CCAATACCTGACGGACATCGCTAG
960-30	AACTAAGTGGCCGTATCGGCATCT
960-31	ACACCTATATCCGTTACGTCAACC
960-32	CCTAATCTAAGCGGTATTGCAGCG
960-33	TCCGTGATCAACGTTGTGGTTGGA
960-34	CTTCTCTACTCCGGATGTGAGAAG
960-35	CTCCATTGAGACGTGAATGCATCG
960-36	CCGGAGCAACACGTATTGGTTACT
960-37	CAGCAGAGTTCCGTATAGGTTCCA
960-38	ACTAACTTATTCGATGCGGCCGAGCTCAAGACCTACGTGGA GGTCTAT
Bar-1	AACCGAGCAATCAGCAATAGCGTCTTCCGCAGTCTCTCAT

M960-2	TGACTAGAAGT/i5MedC/GTGTCCGGCCATT
M960-3	AGTCATACTGT/i5MedC/GCCGGTGCAGTC
M960-4	AACTAGAACTT/i5MedC/GTCGTTGAGCGC
M960-5	AACAATTCTTG/i5MedC/GATTAGGC/GGCC
M960-6	CAACAACCTTGA/i5MedC/GGAGTTGGCTTC
M960-7	TATGTATTACT/i5MedC/GTCACGGAATGC
M960-8	CTCTCAATTCT/i5MedC/GCATAAGACGTG
M960-9	ACCTCTATAGA/i5MedC/GATGGTGAGCTC
M960-10	CACTCATATCA/i5MedC/GTCAATGGCAGG
M960-11	GTAATACTCTT/i5MedC/GGTGGAGCCGAC
M960-12	CCACCAACAAT/i5MedC/GTATTGTGGTCA
M960-13	CGCTTGCTTAA/i5MedC/GGCAAGGCTAAT
M960-14	ACCTCTACTGT/i5MedC/GTAGAGGCAGAT
M960-15	CACACTCCATT/i5MedC/GAAGTGGAGGTT
M960-16	ATTATTCAAGG/i5MedC/GTCGAAGCCGCT
M960-17	CTACATACTTG/i5MedC/GTAGTAGCACGG
M960-18	CAATAACTCTC/i5MedC/GGTTAGGTGAGC
M960-19	CCATATTCCAT/i5MedC/GCCGTGGAGTAG
M960-20	CCTCACAATTA/i5MedC/GTTAGTGCATGA
M960-21	CTCTCGTGTA/i5MedC/GAAGGTGTCACA
M960-22	ACTTCCATTAG/i5MedC/GGTCGTGTCTAC
M960-23	ACATCCGATGT/i5MedC/GATGAGGTATCC
M960-24	ATCAATTAAGG/i5MedC/GACTGTTCCGCG
M960-25	CGCATTCGGAC/i5MedC/GTAATGGTAATC
M960-26	CAAGAGTATTG/i5MedC/GCAGTTGCCTCA
M960-27	CACTAACTACA/i5MedC/GTAGCTGTGGTG
M960-28	CTACAGACGTT/i5MedC/GAAGAGGTTTCAC
M960-29	CCAATACCTGA/i5MedC/GGACATCGCTAG
M960-30	AACTAAGTGGC/i5MedC/GTATCGGCATCT
M960-31	ACACCTATATC/i5MedC/GTTACGTCAACC
M960-32	CCTAATCTAAG/i5MedC/GGTATTGCAGCG
M960-33	TCCGTGATCAA/i5MedC/GTTGTGGTTGGA
M960-34	CTTCTCTACTC/i5MedC/GGATGTGAGAAG
M960-35	CTCCATTGAGA/i5MedC/GTGAATGCATCG
M960-36	CCGGAGCAACA/i5MedC/GTATTGGTTACT
M960-37	CAGCAGAGTTC/i5MedC/GTATAGGTTCCA
960 primer and fluorescent strands	
960-pcr1	TGCTTCCTATCCGAAGCTG
960-pcr2	ACTAACTTATTCGATGCGGCC
960-fam1	FAM-AACTAGAACTTCGTCGTTGAGCGC
960-fam2	FAM-AACAATTCTTGCGATTAGGCGGCC
1200 primer and fluorescent strands	
1200-pcr1	GTGTGTCCTCATGCTAATAG
1200-pcr2	CTCGGCCGCATCGAATAA
1200-fam1	FAM-AATACACTTAGGCGGTGAGCTCCT

1200-fam2	FAM-CTTCCTAGAGGATAGTCGTCCAGA
The strands of Glu I experiments	
a1	TGACTGATG/i5MedC/G/i5MedC/ATAGG-BHQ
a2	TGACTGATGCG/i5MedC/ATAGG-BHQ
a1-without BHQ	TGACTGATG/i5MedC/G/i5MedC/ATAGG
a2-without BHQ	TGACTGATGCG/i5MedC/ATAGG
b1	FAM-CCTATGCG/i5MedC/ATCAGTCA
b2	FAM-CCTATG/i5MedC/G/i5MedC/ATCAGTCA
Brick strands complementary to template 1200	
1200-1	CCACTACTTCGCTTGTAGTGGATGTCCGCTATTAGCATGAG GACACAC
1200-2	CTTCCTAGAGGATAGTCGTCCAGA
1200-3	AATACACTTAGGCGGTGAGCTCCT
1200-4	CACGTGCAAATGACCAATGGCGAT
1200-5	GTCGACTATAACTGGAACGGTGAT
1200-6	GTCGCGTCCAATCTTGTGGATGAC
1200-7	CTATCGAGCCAATATTGTATGCAT
1200-8	CCACAACGTATTTCGCCTGGTGGCC
1200-9	CAGTGGCGTCAATTGTACTAGTCA
1200-10	CACCTCATCGTTCAAGTGGTAGAG
1200-11	ACAATCAGCGCTCATTGTGGCTAG
1200-12	CTTCTGACAGCGTTAGAGGACTCA
1200-13	CCTAATTCTGCGTAGTAGGCAGAC
1200-14	ACAATAATCGGCGTGTTGGACCT
1200-15	CCGAATGGAATTCGCATGGTCATC
1200-16	AGTTCTACCTCGTGCGAGCATGTA
1200-17	CCTCCATTATTAGGCGAGGAGATC
1200-18	CCTACCTTCATTAAGGCGTTATGA
1200-19	CCATTGTTCAATTTCTACGGACAGA
1200-20	ACAGCAATCAATTGGTTGCGCTGG
1200-21	GAAGCTACTGAGTGTAGGCGACTT
1200-22	CCTTGAAACTCAGTGTTGGTCGTT
1200-23	CACAACACTTGTGGTCAAGCGCT
1200-24	CACGCCACATTATTGGTGGACGTA
1200-25	CACGACAACATCTTGGTGGTCGAT
1200-26	AATTCGGACCACACTTGCCGATAG
1200-27	CTCTCGTCAATTATGAGGCGAGAC
1200-28	GTTCTACGAGTCATGGCGTTCAGG
1200-29	CACATTCGCAATGACGCGGTACCA
1200-30	ACTTACGCGATTTATCGCATGTGG
1200-31	AATAATTCGTCCGCACGCTAGTGG
1200-32	TCTTATTCCGAGTGCGATGGAGGA
1200-33	ACCTCAACCGTCAGCGCATGATTG
1200-34	CTACGATGGCGTACGAATTCAGTC
1200-35	CCACTATTACGATCGGTGGTGCAA

1200-36	GGCGGTTATGGCGAGTATTGCGAC
1200-37	TTCGCTTCAGTCGTCTAGTCCGTC
1200-38	TACCCGACAAACGCAGCACGTACT
1200-39	CCAACGCCTATCGTGAATCGATGG
1200-40	TTTGCTCGTTACGGGCCGCAAGAT
1200-41	GAGTCACGAGGCGTATCGTTGTAG
1200-42	AACATCCACGACGGCGTCAGGCTT
1200-43	AATAACTACGGCGCCGCTCTAGTG
1200-44	ATCGATTACGTGTGCGGAGTCGCG
1200-45	TTCGGAATCGAGTCCGTAGCCGGT
1200-46	ACATCTGCTAACTCTGTGGAGTTG
1200-47	CATACCATTGAGGTTGCTGCAA
1200-48	CTCGGCCGCATCGAATAAGTTAGTCTCTTATACGAGCATTC GGCGAAG
M1200-4	CA/i5MedC/GTGCAAATGACCAATGGCGAT
M1200-6	GTCG/i5MedC/GTCCAATCTTGTGGATGAC
M1200-8	CCACAA/i5MedC/GTATTCGCCTGGTGGCC
M1200-18	CCTACCTTCATTAAGG/i5MedC/GTTATGA
M1200-20	ACAGCAATCAATTGGTTG/i5MedC/GCTGG
M1200-22	CCTTGAAACTCAGTGTGGT/i5MedC/GTT
M1200-24	CA/i5MedC/GCCACATTATTGGTGGA/i5MedC/GTA
M1200-25	CA/i5MedC/GACAACATCTTGGTGGT/i5MedC/GAT
M1200-26	AATT/i5MedC/GGACCACACTTGC/i5MedC/GATAG
M1200-34	CTACGATGG/i5MedC/GTA/i5MedC/GAATTCAGTC
M1200-38	TACC/i5MedC/GACAAA/i5MedC/GCAGCA/i5MedC/GTACT
Barcode and linker strands	
960code-1	CCAGCTTCGGATAGGAAGCACTGGAGACTGCAACAATTAA
960code-2	CCAGCTTCGGATAGGAAGCAAACGCGGTCAATCCTACAAT
960code-3	CCAGCTTCGGATAGGAAGCAGTTATGGTTCTCTTACGACT
960code-4	CCAGCTTCGGATAGGAAGCAACTGCGGTCCATCACTAAGA
960code-5	CCAGCTTCGGATAGGAAGCAATATCAGCCTAGCGAGTACA
960code-6	CCAGCTTCGGATAGGAAGCAATAAGAATGCGATAACCTCC
960code-7	CCAGCTTCGGATAGGAAGCACGACTGAATCCACGACTACT
960code-8	CCAGCTTCGGATAGGAAGCATAGTAAGCCTGTATCTCCTC
960code-9	CCAGCTTCGGATAGGAAGCACGACACAATGCCTCTGACTT
960code-10	CCAGCTTCGGATAGGAAGCAATTGCATCCTCGTCTGTTAC
960code-11	CCAGCTTCGGATAGGAAGCACGGAGTCGATCTGCAAGTGT
960code-12	CCAGCTTCGGATAGGAAGCAGGTCCGCCGTCCATTACTTC
960code-13	CCAGCTTCGGATAGGAAGCACTCACCTGGCTAATGCTTGG
960code-14	CCAGCTTCGGATAGGAAGCATAGAACAGTATGCACCTCCG
960code-15	CCAGCTTCGGATAGGAAGCACGCGCAATAGTAGCAGGTG
960code-16	CCAGCTTCGGATAGGAAGCATCCACTATCCGAATGAGCTA
960code-17	CCAGCTTCGGATAGGAAGCATATAGAAGTCATCCTGTGCG
960code-18	CCAGCTTCGGATAGGAAGCATTGTAAGAGTAGTGCGCGGA
960code-19	CCAGCTTCGGATAGGAAGCATGCTAAGTGCGGTTGGTGTA

960code-20	CCAGCTTCGGATAGGAAGCATGCTATGCTGGATTGTACTG
960code-21	CCAGCTTCGGATAGGAAGCAGCGTCTTGTTATTGCTTCAC
960code-22	CCAGCTTCGGATAGGAAGCAGGCCGACCAGACGAAGGATA
960code-23	CCAGCTTCGGATAGGAAGCAGCTATTCAGGACCACGCATC
960code-24	CCAGCTTCGGATAGGAAGCACTGCTAATGAACGCCAACGG
960code-25	CCAGCTTCGGATAGGAAGCATAACCAATAGTCGCTGAGAA
barcode-1	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTTAATTGTTGCAGTCTCCAG
barcode-2	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGATTGTAGGATTGACCGCGTT
barcode-3	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGAGTCGTAAGAGAACCATAAC
barcode-4	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTCTTAGTGATGGACCGCAGT
barcode-5	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTGTACTCGCTAGGCTGATAT
barcode-6	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGGGAGGTTATCGCATTCTTAT
barcode-7	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGAGTAGTCGTGGATTCAAGTCG
barcode-8	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGGAGGAGATACAGGCTTACTA
barcode-9	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGAAGTCAGAGGCATTGTGTCG
barcode-10	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGGTAACAGACGAGGATGCAAT
barcode-11	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGACACTTGCAGATCGACTCCG
barcode-12	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGGAAGTAATGGACGGCGGACC
barcode-13	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGCCAAGCATTAGCCAGGTGAG
barcode-14	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGCGGAGGTGCATACTGTTCTA
barcode-15	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGCACCTGCTACTATTGCCGCG
barcode-16	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTAGCTCATTCCGATAGTGGA
barcode-17	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGCGCACAGGATGACTTCTATA
barcode-18	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTCCGCGCACTACTCTTACAA
barcode-19	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTACACCAACCGCACTTAGCA

barcode-20	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGCAGTACAATCCAGCATAGCA
barcode-21	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGGTGAAGCAATAACAAGACGC
barcode-22	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTATCCTTCGTCTGGTCGGCC
barcode-23	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGGATGCGTGGTCCTGAATAGC
barcode-24	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGCCGTTGGCGTTCATTAGCAG
barcode-25	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTTCTCAGCGACTATTGGTTA
barcode-26	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGACATAGTAAGCCAAGCAAGT
barcode-27	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTGTGTGCCGATGATGATAAC
barcode-28	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGCCAACCTTATGCTCACGAAT
barcode-29	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGGGTTCCTAGCCAGTCGTTCTG
barcode-30	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGCTGGTATGGTCAATTCGAGT
barcode-31	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGCTTAGATCACACCTCAGTTG
barcode-32	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGCACACAAGATGCTATTCGCT
barcode-33	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGCCAACAGGCATATTCTAGGA
barcode-34	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTGAATACGGTTGCAGAGAAG
barcode-35	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGAGACGAAGCTCAGATGGCCT
barcode-36	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTACCGACTACTGGAGGAGAG
barcode-37	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGAGGCCAGGAGCAAGAACAGT
barcode-38	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGAGTTGATGCACCACCGTAGG
barcode-39	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGATGCTTCGCGTATATGGTTA
barcode-40	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGAAGAGATGGTGTGAGGCGTA
barcode-41	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTCCTTGTATTCTAGTGACGC
barcode-42	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTTGATGGTATGACTGCTGTG

barcode-43	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGGTAGGCAGTATGTTAGGCGA
barcode-44	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGGGCAGCAGTGGATGTCTTAA
barcode-45	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTATCTCACCTTCGCTACGCA
barcode-46	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTGGAGACAGGTAGAACATGG
barcode-47	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGCAAGGAAGAGAGTTACAAGC
barcode-48	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGGAATAAGGATGGTGTCTCAA
barcode-49	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTCAACTACTCCAGGTCCATA
barcode-50	ATGAGAGACTGCGGAAGACGCTATTGCTGATTGCTCGGTTG TAACCTAAGCCTAATCAGTTCTGGATTCTTGCTGGTCA
code-1	GAACACATTACCATCTAAACCTGGAGACTGCAACAATTAA
code-2	GAACACATTACCATCTAAACAACGCGGTCAATCCTACAAT
code-3	GAACACATTACCATCTAAACGTTATGGTTCTCTTACGACT
code-4	GAACACATTACCATCTAAACACTGCGGTCCATCACTAAGA
code-5	GAACACATTACCATCTAAACATATCAGCCTAGCGAGTACA
code-6	GAACACATTACCATCTAAACATAAGAATGCGATAACCTCC
code-7	GAACACATTACCATCTAAACCGACTGAATCCACGACTACT
code-8	GAACACATTACCATCTAAACTAGTAAGCCTGTATCTCCTC
code-9	GAACACATTACCATCTAAACCGACACAATGCCTCTGACTT
code-10	GAACACATTACCATCTAAACATTGCATCCTCGTCTGTTAC
code-11	GAACACATTACCATCTAAACCGGAGTCGATCTGCAAGTGT
code-12	GAACACATTACCATCTAAACGGTCCGCCGTCCATTACTTC
code-13	GAACACATTACCATCTAAACCTCACCTGGCTAATGCTTGG
code-14	GAACACATTACCATCTAAACTAGAACAGTATGCACCTCCG
code-15	GAACACATTACCATCTAAACCGCGGCAATAGTAGCAGGTG
code-16	GAACACATTACCATCTAAACTCCACTATCCGAATGAGCTA
code-17	GAACACATTACCATCTAAACTATAGAAGTCATCCTGTGCG
code-18	GAACACATTACCATCTAAACTTGTAAGAGTAGTGCGCGGA
code-19	GAACACATTACCATCTAAACTGCTAAGTGCGGTTGGTGTA
code-20	GAACACATTACCATCTAAACTGCTATGCTGGATTGTACTG
code-21	GAACACATTACCATCTAAACGCGTCTTGTTATTGCTTCAC
code-22	GAACACATTACCATCTAAACGGCCGACCAGACGAAGGATA
code-23	GAACACATTACCATCTAAACGCTATTCAGGACCACGCATC
code-24	GAACACATTACCATCTAAACCTGCTAATGAACGCCAACGG
code-25	GAACACATTACCATCTAAACTAACCAATAGTCGCTGAGAA
code-26	GAACACATTACCATCTAAACACTTGCTTGGCTTACTATGT
code-27	GAACACATTACCATCTAAACGTTATCATCATCGGCACACA
code-28	GAACACATTACCATCTAAACATTCTGTGAGCATAAGGTTGG
code-29	GAACACATTACCATCTAAACCGAACGACTGGCTAGGAACC

code-30	GAACACATTACCATCTAAACACTCGAATTGACCATAACCAG
code-31	GAACACATTACCATCTAAACCAACTGAGGTGTGATCTAAG
code-32	GAACACATTACCATCTAAACAGCGAATAGCATCTTGTGTG
code-33	GAACACATTACCATCTAAACTCCTAGAAATATGCCTGTTGG
code-34	GAACACATTACCATCTAAACCTTCTCTGCAACCGTATTCA
code-35	GAACACATTACCATCTAAACAGGCCATCTGAGCTTCGTCT
code-36	GAACACATTACCATCTAAACCTCTCCTCCAGTAGTCGGTA
code-37	GAACACATTACCATCTAAACACTGTTCTTGCTCCTGGCCT
code-38	GAACACATTACCATCTAAACCCTACGGTGGTGCATCAACT
code-39	GAACACATTACCATCTAAACTAACCATATACGCGAAGCAT
code-40	GAACACATTACCATCTAAACTACGCCTCACACCATCTCTT
code-41	GAACACATTACCATCTAAACGCGTCACTAGAATACAAGGA
code-42	GAACACATTACCATCTAAACCACAGCAGTCATACCATCAA
code-43	GAACACATTACCATCTAAACTCGCCTAACATACTGCCTAC
code-44	GAACACATTACCATCTAAACTTAAGACATCCACTGCTGCC
code-45	GAACACATTACCATCTAAACTGCGTAGCGAAGGTGAGATA
code-46	GAACACATTACCATCTAAACCCATGTTCTACCTGTCTCCA
code-47	GAACACATTACCATCTAAACGCTTGTAACCTCTCTTCCTTG
code-48	GAACACATTACCATCTAAACTTGAGACACCATCCTTATTC
code-49	GAACACATTACCATCTAAACTATGGACCTGGAGTAGTTGA
code-50	GAACACATTACCATCTAAACTGACCAGCAAGAATCCAGAA
2000-82	AACTAACACTGTGAATACAGACAGAACCAGGCACCACAGA AAAGAAAATAAGGACCGAGA
Primers for templates L1 to L5	
L-pcr5	CCGATAGTCACATAAGGAGG
L-pcr3	CTGTTAGGATTCTATCCGCG
Brick strands complementary to 3 and 5 ends of templates L1 to L5	
L-1	CCTCCTTATGTGACTATCGG
L-2	TCCACGTAATCGAGTTGTAG
L-3	GGCATCACTCTACAATGAAT
L-4	CATGCAAGCTGGATATCCAT
L-5	TTCATGCACCATCCGCATCA
L-6	GACGATCTCAACACACTGTC
L-7	TATACATCCGTCAGTCAGAG
L-8	AATTGATTGGAGCTGATGCA
L-9	TGTAACGAACAGGTCCGTGT
L-10	TACTGTTCTCCATCATAGGT
L-11	AGTTGGTTGTCTCCTGGTCC
L-12	ACTTGGCCTAGCCAGTTATT
L-13	AGTTCAAGGAGAACTCGGTC
L-14	GAATATGTGTATCACAACCG
L-15	CGACGCTAACTATATGCGGC
L-16	CCGGTGTGCCGGTAATATCT
L-17	TGTTGAACATATGGCTCTTG
L-18	GAAGTGCAAGCACGTAATGG

L-19	TCCTGTCATGGCACAGAAGA
L-20	ACCGAGCATCGTTGCTGTCA
L-21	CTGAGGTTAAGTACAGTTGC
L-22	GAGTAGGAGTAAGTAGGACG
L-23	TGCCGTATAGCGGTAGTTGT
L-24	CGATACCATTGGTAGATAGG
L-25	ATGAGGACACTCCATTTCGCA
L-26	GGTAGTCTGTTAATACACCG
L-27	AACCGAACAATCCGGTAACA
L-28	GCGCACTTCAGCGATGAAGG
L-29	TCAGCTCAGAATGGACAGTC
L-30	CTGTTAGGATTCTATCCGCG
Brick strands complementary to template L1	
L1-1	TATCCGAATCCTGGCGAAGG
L1-2	GACTCGTCATCTGGCGGACA
L1-3	TTAACGAAGGTTGACGCTTG
L1-4	GTCACGGTATTGTGCGGTAT
L1-5	TGCACGAATTGATCCGAGTG
L1-6	GTTACGACACATGCCGTGTA
L1-7	GTATCGAAGACAGACGAAGT
L1-8	GAGACGTAATTAGGCGACAT
L1-9	TGTACGATCATAAGCGCAGA
L1-10	CAGCCGTAGGAATTCGCTTC
L1-11	ACAACGGAGACCTGCGACAG
L1-12	TATGCGAGTATAGTCGCTGA
L1-13	ATATCGCTAACCAGCGCATG
L1-14	TGTTTCGTATGTGATCGCAGA
L1-15	ACTGCGGAGTGAAGCGCTCA
L1-16	AATGCGAAGGATCACGTTCT
L1-17	GTGGCGGAGAGTTCCGTCTA
L1-18	CCAGCGTTCATTGCCGCTAA
L1-19	GAATCGTAAGGTACCGTCTG
L1-20	ATAACGGACATTACCGTCCA
L1-21	TCCACGCTAACTTACGTGCT
L1-22	TAGGCGTATAGTTGCGGAAG
L1-23	ACCACGATCATGGTCGGTAC
L1-24	TAATCGAGACTGGCCGATAG
L1-25	ATCACGATACTGCACGCACT
L1-26	TCTTCGTTGACTAGCGGCTA
L1-27	TGAACGTCTAGGTACGAGGT
L1-28	TGTCCGCCTATACTCGGACT
L1-29	AGTTCGTCATACTACGCTGA
L1-30	TGCTCGACATATAACGTCCA
L1-31	AGGCCGCTATGCAGCGGTAT
L1-32	ATAACGTGTCACATCGACTT

L1-33	CTAGCGACTCTGTACGAGCC
L1-34	ACTGCGTGATACCACGATCT
L1-35	TAGCCGATAACCTACGGAGC
M01-L1-1	TATC/i5MedC/GAATCCTGGCGAAGG
M01-L1-2	GACT/i5MedC/GTCATCTGGCGGACA
M01-L1-3	TTAA/i5MedC/GAAGGTTGACGCTTG
M01-L1-4	GTCA/i5MedC/GGTATTGTGCGGTAT
M01-L1-5	TGCA/i5MedC/GAATTGATCCGAGTG
M01-L1-6	GTTA/i5MedC/GACACATGCCGTGTA
M01-L1-7	GTAT/i5MedC/GAAGACAGACGAAGT
M01-L1-8	GAGA/i5MedC/GTAATTAGGCGACAT
M01-L1-9	TGTA/i5MedC/GATCATAAGCGCAGA
M01-L1-10	CAGC/i5MedC/GTAGGAATTCGCTTC
M01-L1-11	ACAA/i5MedC/GGAGACCTGCGACAG
M01-L1-12	TATG/i5MedC/GAGTATAGTCGCTGA
M01-L1-13	ATAT/i5MedC/GCTAACCAGCGCATG
M01-L1-14	TGTT/i5MedC/GTATGTGATCGCAGA
M01-L1-15	ACTG/i5MedC/GGAGTGAAGCGCTCA
M01-L1-16	AATG/i5MedC/GAAGGATCACGTTCT
M01-L1-17	GTGG/i5MedC/GGAGAGTTCCGTCTA
M01-L1-18	CCAG/i5MedC/GTTCATTGCCGTAA
M01-L1-19	GAAT/i5MedC/GTAAGGTACCGTCTG
M01-L1-20	ATAA/i5MedC/GGACATTACCGTCCA
M01-L1-21	TCCA/i5MedC/GCTAACTTACGTGCT
M01-L1-22	TAGG/i5MedC/GTATAGTTGCGGAAG
M01-L1-23	ACCA/i5MedC/GATCATGGTCGGTAC
M01-L1-24	TAAT/i5MedC/GAGACTGGCCGATAG
M01-L1-25	ATCA/i5MedC/GATACTGCACGCACT
M01-L1-26	TCTT/i5MedC/GTTGACTAGCGGCTA
M01-L1-27	TGAA/i5MedC/GTCTAGGTACGAGGT
M01-L1-28	TGTC/i5MedC/GCCTATACTCGGACT
M01-L1-29	AGTT/i5MedC/GTCATACTACGCTGA
M01-L1-30	TGCT/i5MedC/GACATATAACGTCCA
M01-L1-31	AGGC/i5MedC/GCTATGCAGCGGTAT
M01-L1-32	ATAA/i5MedC/GTGTACATCGACTT
M01-L1-33	CTAG/i5MedC/GACTCTGTACGAGCC
M01-L1-34	ACTG/i5MedC/GTGATACCACGATCT
M01-L1-35	TAGC/i5MedC/GATAACCTACGGAGC
M10-L1-1	TATCCGAATCCTGG/i5MedC/GAAGG
M10-L1-2	GACTCGTCATCTGG/i5MedC/GGACA
M10-L1-3	TTAACGAAGGTTGA/i5MedC/GCTTG
M10-L1-4	GTCACGGTATTGTG/i5MedC/GGTAT
M10-L1-5	TGCACGAATTGATC/i5MedC/GAGTG
M10-L1-6	GTTACGACACATGC/i5MedC/GTGTA
M10-L1-7	GTATCGAAGACAGA/i5MedC/GAAGT

M10-L1-8	GAGACGTAATTAGG/i5MedC/GACAT
M10-L1-9	TGTACGATCATAAG/i5MedC/GCAGA
M10-L1-10	CAGCCGTAGGAATT/i5MedC/GCTTC
M10-L1-11	ACAACGGAGACCTG/i5MedC/GACAG
M10-L1-12	TATGCGAGTATAGT/i5MedC/GCTGA
M10-L1-13	ATATCGCTAACCAG/i5MedC/GCATG
M10-L1-14	TGTTCTGTATGTGAT/i5MedC/GCAGA
M10-L1-15	ACTGCGGAGTGAAG/i5MedC/GCTCA
M10-L1-16	AATGCGAAGGATCA/i5MedC/GTTCT
M10-L1-17	GTGGCGGAGAGTTC/i5MedC/GTCTA
M10-L1-18	CCAGCGTTCATTGC/i5MedC/GCTAA
M10-L1-19	GAATCGTAAGGTAC/i5MedC/GTCTG
M10-L1-20	ATAACGGACATTAC/i5MedC/GTCCA
M10-L1-21	TCCACGCTAACTTA/i5MedC/GTGCT
M10-L1-22	TAGGCGTATAGTTG/i5MedC/GGAAG
M10-L1-23	ACCACGATCATGGT/i5MedC/GGTAC
M10-L1-24	TAATCGAGACTGGC/i5MedC/GATAG
M10-L1-25	ATCACGATACTGCA/i5MedC/GCACT
M10-L1-26	TCTTCGTTGACTAG/i5MedC/GGCTA
M10-L1-27	TGAACGTCTAGGTA/i5MedC/GAGGT
M10-L1-28	TGTCCGCCTATACT/i5MedC/GGACT
M10-L1-29	AGTTCGTCATACTA/i5MedC/GCTGA
M10-L1-30	TGCTCGACATATAA/i5MedC/GTCCA
M10-L1-31	AGGCCGCTATGCAG/i5MedC/GGTAT
M10-L1-32	ATAACGTGTCACAT/i5MedC/GACTT
M10-L1-33	CTAGCGACTCTGTA/i5MedC/GAGCC
M10-L1-34	ACTGCGTGATACCA/i5MedC/GATCT
M10-L1-35	TAGCCGATAACCTA/i5MedC/GGAGC
M11-L1-1	TATC/i5MedC/GAATCCTGG/i5MedC/GAAGG
M11-L1-2	GACT/i5MedC/GTCATCTGG/i5MedC/GGACA
M11-L1-3	TTAA/i5MedC/GAAGGTTGA/i5MedC/GCTTG
M11-L1-4	GTCA/i5MedC/GGTATTGTG/i5MedC/GGTAT
M11-L1-5	TGCA/i5MedC/GAATTGATC/i5MedC/GAGTG
M11-L1-6	GTTA/i5MedC/GACACATGC/i5MedC/GTGTA
M11-L1-7	GTAT/i5MedC/GAAGACAGA/i5MedC/GAAGT
M11-L1-8	GAGA/i5MedC/GTAATTAGG/i5MedC/GACAT
M11-L1-9	TGTA/i5MedC/GATCATAAG/i5MedC/GCAGA
M11-L1-10	CAGC/i5MedC/GTAGGAATT/i5MedC/GCTTC
M11-L1-11	ACAA/i5MedC/GGAGACCTG/i5MedC/GACAG
M11-L1-12	TATG/i5MedC/GAGTATAGT/i5MedC/GCTGA
M11-L1-13	ATAT/i5MedC/GCTAACCAG/i5MedC/GCATG
M11-L1-14	TGTT/i5MedC/GTATGTGAT/i5MedC/GCAGA
M11-L1-15	ACTG/i5MedC/GGAGTGAAG/i5MedC/GCTCA
M11-L1-16	AATG/i5MedC/GAAGGATCA/i5MedC/GTTCT
M11-L1-17	GTGG/i5MedC/GGAGAGTTC/i5MedC/GTCTA

M11-L1-18	CCAG/i5MedC/GTTCATTGC/i5MedC/GCTAA
M11-L1-19	GAAT/i5MedC/GTAAGGTAC/i5MedC/GTCTG
M11-L1-20	ATAA/i5MedC/GGACATTAC/i5MedC/GTCCA
M11-L1-21	TCCA/i5MedC/GCTAACTTA/i5MedC/GTGCT
M11-L1-22	TAGG/i5MedC/GTATAGTTG/i5MedC/GGAAG
M11-L1-23	ACCA/i5MedC/GATCATGGT/i5MedC/GGTAC
M11-L1-24	TAAT/i5MedC/GAGACTGGC/i5MedC/GATAG
M11-L1-25	ATCA/i5MedC/GATACTGCA/i5MedC/GCACT
M11-L1-26	TCTT/i5MedC/GTTGACTAG/i5MedC/GGCTA
M11-L1-27	TGAA/i5MedC/GTCTAGGTA/i5MedC/GAGGT
M11-L1-28	TGTC/i5MedC/GCCTATACT/i5MedC/GGACT
M11-L1-29	AGTT/i5MedC/GTCATACTA/i5MedC/GCTGA
M11-L1-30	TGCT/i5MedC/GACATATAA/i5MedC/GTCCA
M11-L1-31	AGGC/i5MedC/GCTATGCAG/i5MedC/GGTAT
M11-L1-32	ATAA/i5MedC/GTGTACACAT/i5MedC/GACTT
M11-L1-33	CTAG/i5MedC/GACTCTGTA/i5MedC/GAGCC
M11-L1-34	ACTG/i5MedC/GTGATACCA/i5MedC/GATCT
M11-L1-35	TAGC/i5MedC/GATAACCTA/i5MedC/GGAGC
Brick strands complementary to template L2	
L2-1	GTCTCGTCCTTGTCGAGGA
L2-2	TTGCCGGTTGACTACGCCAA
L2-3	CTAACGGCTTGGATCGTTAA
L2-4	ACACCGCCTTAATGCGATTG
L2-5	CTGACGATTGCTCTCGTACC
L2-6	CACTCGACTATGGTCGATAT
L2-7	CAGTCGGCTCTTGCCGCATT
L2-8	GAAGCGCATACTTGCGCTCT
L2-9	GGTCCGTAACATCTCGGATG
L2-10	TCAACGTACTCATACGTCTG
L2-11	TTACCGTGGCCTCTCGCCAT
L2-12	CTGGCGTGATACTACGCCTG
L2-13	AGTACGGCCTGATGCGTCTC
L2-14	TAGCCGAATGGCCTCGTGAT
L2-15	TCTCCGTCCATTAACGTATG
L2-16	CAGCCGTTATTATGCGGATG
L2-17	AGTTCGTGGTGAACCGTTGC
L2-18	CACACGGTCATTACGGTGT
L2-19	ACAGCGCACCATGGCGGTTA
L2-20	GCAGCGTTAGTCCTCGCTCT
L2-21	TAATCGCAGCACCTCGATAG
L2-22	GAAGCGGAACCTAACCGCTGA
L2-23	GGATCGACTGTACACGATGG
L2-24	TCTACGAAGGAATTCGAGCT
L2-25	CCAGCGATAGAACACGCTCC
L2-26	AATGCGTCTCAAGGCGCAGA

L2-27	TATCCGCCACTAGCCGGAAG
L2-28	AGTACGGAATAATCCGGTAG
L2-29	AACACGAGTTAGGTCGTCAT
L2-30	AGGTCGAAGCTACTCGAATA
L2-31	TTGACGCTGACCAACGTACC
L2-32	ACAACGTAAGACTACGTAAC
L2-33	ATATCGCATTGATGCGCTAA
L2-34	CATGCGGAATGAACCGTAGC
L2-35	AATACGTAGTGCACCGAGCC
M01-L2-1	GTCT/i5MedC/GTCCTTGTCGAGGA
M01-L2-2	TTGC/i5MedC/GGTTGACTACGCCAA
M01-L2-3	CTAA/i5MedC/GGCTTGGATCGTTAA
M01-L2-4	ACAC/i5MedC/GCCTTAATGCGATTG
M01-L2-5	CTGA/i5MedC/GATTGCTCTCGTACC
M01-L2-6	CACT/i5MedC/GACTATGGTCGATAT
M01-L2-7	CAGT/i5MedC/GGCTCTTGCCGCATT
M01-L2-8	GAAG/i5MedC/GCATACTTGCGCTCT
M01-L2-9	GGTC/i5MedC/GTAACATCTCGGATG
M01-L2-10	TCAA/i5MedC/GTACTCATACTGTCTG
M01-L2-11	TTAC/i5MedC/GTGGCCTCTCGCCAT
M01-L2-12	CTGG/i5MedC/GTGATACTACGCCTG
M01-L2-13	AGTA/i5MedC/GGCCTGATGCGTCTC
M01-L2-14	TAGC/i5MedC/GAATGGCCTCGTGAT
M01-L2-15	TCTC/i5MedC/GTCCATTAAACGTATG
M01-L2-16	CAGC/i5MedC/GTTATTATGCGGATG
M01-L2-17	AGTT/i5MedC/GTGGTGAACCGTTGC
M01-L2-18	CACA/i5MedC/GGTCATTACAGGTGT
M01-L2-19	ACAG/i5MedC/GCACCATGGCGGTTA
M01-L2-20	GCAG/i5MedC/GTTAGTCCTCGCTCT
M01-L2-21	TAAT/i5MedC/GCAGCACCTCGATAG
M01-L2-22	GAAG/i5MedC/GGAACTAACCGCTGA
M01-L2-23	GGAT/i5MedC/GACTGTACACGATGG
M01-L2-24	TCTA/i5MedC/GAAGGAATTCGAGCT
M01-L2-25	CCAG/i5MedC/GATAGAACACGCTCC
M01-L2-26	AATG/i5MedC/GTCTCAAGGCGCAGA
M01-L2-27	TATC/i5MedC/GCCACTAGCCGGAAG
M01-L2-28	AGTA/i5MedC/GGAATAATCCGGTAG
M01-L2-29	AACA/i5MedC/GAGTTAGGTCGTCAT
M01-L2-30	AGGT/i5MedC/GAAGCTACTCGAATA
M01-L2-31	TTGA/i5MedC/GCTGACCAACGTACC
M01-L2-32	ACAA/i5MedC/GTAAGACTACGTAAC
M01-L2-33	ATAT/i5MedC/GCATTGATGCGCTAA
M01-L2-34	CATG/i5MedC/GGAATGAACCGTAGC
M01-L2-35	AATA/i5MedC/GTAGTGCACCGAGCC
M10-L2-1	GTCTCGTCCTTGTC/i5MedC/GAGGA

M10-L2-2	TTGCCGGTTGACTA/i5MedC/GCCAA
M10-L2-3	CTAACGGCTTGGAT/i5MedC/GTTAA
M10-L2-4	ACACCGCCTTAATG/i5MedC/GATTG
M10-L2-5	CTGACGATTGCTCT/i5MedC/GTACC
M10-L2-6	CACTCGACTATGGT/i5MedC/GATAT
M10-L2-7	CAGTCGGCTCTTGC/i5MedC/GCATT
M10-L2-8	GAAGCGCATACTTG/i5MedC/GCTCT
M10-L2-9	GGTCCGTAACATCT/i5MedC/GGATG
M10-L2-10	TCAACGTACTCATA/i5MedC/GTCTG
M10-L2-11	TTACCGTGGCCTCT/i5MedC/GCCAT
M10-L2-12	CTGGCGTGATACTA/i5MedC/GCCTG
M10-L2-13	AGTACGGCCTGATG/i5MedC/GTCTC
M10-L2-14	TAGCCGAATGGCCT/i5MedC/GTGAT
M10-L2-15	TCTCCGTCCATTAA/i5MedC/GTATG
M10-L2-16	CAGCCGTTATTATG/i5MedC/GGATG
M10-L2-17	AGTTCGTGGTGAAC/i5MedC/GTTGC
M10-L2-18	CACACGGTCATTCA/i5MedC/GGTGT
M10-L2-19	ACAGCGCACCATGG/i5MedC/GGTTA
M10-L2-20	GCAGCGTTAGTCCT/i5MedC/GCTCT
M10-L2-21	TAATCGCAGCACCT/i5MedC/GATAG
M10-L2-22	GAAGCGGAACATAAC/i5MedC/GCTGA
M10-L2-23	GGATCGACTGTACA/i5MedC/GATGG
M10-L2-24	TCTACGAAGGAATT/i5MedC/GAGCT
M10-L2-25	CCAGCGATAGAACA/i5MedC/GCTCC
M10-L2-26	AATGCGTCTCAAGG/i5MedC/GCAGA
M10-L2-27	TATCCGCCACTAGC/i5MedC/GGAAG
M10-L2-28	AGTACGGAATAATC/i5MedC/GGTAG
M10-L2-29	AACACGAGTTAGGT/i5MedC/GTCAT
M10-L2-30	AGGTCAAGCTACT/i5MedC/GAATA
M10-L2-31	TTGACGCTGACCAA/i5MedC/GTACC
M10-L2-32	ACAACGTAAGACTA/i5MedC/GTAAC
M10-L2-33	ATATCGCATTGATG/i5MedC/GCTAA
M10-L2-34	CATGCGGAATGAAC/i5MedC/GTAGC
M10-L2-35	AATACGTAGTGCAC/i5MedC/GAGCC
M11-L2-1	GTCT/i5MedC/GTCCTTGTC/i5MedC/GAGGA
M11-L2-2	TTGC/i5MedC/GGTTGACTA/i5MedC/GCCAA
M11-L2-3	CTAA/i5MedC/GGCTTGGAT/i5MedC/GTTAA
M11-L2-4	ACAC/i5MedC/GCCTTAATG/i5MedC/GATTG
M11-L2-5	CTGA/i5MedC/GATTGCTCT/i5MedC/GTACC
M11-L2-6	CACT/i5MedC/GACTATGGT/i5MedC/GATAT
M11-L2-7	CAGT/i5MedC/GGCTCTTGC/i5MedC/GCATT
M11-L2-8	GAAG/i5MedC/GCATACTTG/i5MedC/GCTCT
M11-L2-9	GGTC/i5MedC/GTAACATCT/i5MedC/GGATG
M11-L2-10	TCAA/i5MedC/GTACTCATA/i5MedC/GTCTG
M11-L2-11	TTAC/i5MedC/GTGGCCTCT/i5MedC/GCCAT

M11-L2-12	CTGG/i5MedC/GTGATACTA/i5MedC/GCCTG
M11-L2-13	AGTA/i5MedC/GGCCTGATG/i5MedC/GTCTC
M11-L2-14	TAGC/i5MedC/GAATGGCCT/i5MedC/GTGAT
M11-L2-15	TCTC/i5MedC/GTCCATTAA/i5MedC/GTATG
M11-L2-16	CAGC/i5MedC/GTTATTATG/i5MedC/GGATG
M11-L2-17	AGTT/i5MedC/GTGGTGAAC/i5MedC/GTTGC
M11-L2-18	CACA/i5MedC/GGTCATTCA/i5MedC/GGTGT
M11-L2-19	ACAG/i5MedC/GCACCATGG/i5MedC/GGTTA
M11-L2-20	GCAG/i5MedC/GTTAGTCCT/i5MedC/GCTCT
M11-L2-21	TAAT/i5MedC/GCAGCACCT/i5MedC/GATAG
M11-L2-22	GAAG/i5MedC/GGAACATAAC/i5MedC/GCTGA
M11-L2-23	GGAT/i5MedC/GACTGTACA/i5MedC/GATGG
M11-L2-24	TCTA/i5MedC/GAAGGAATT/i5MedC/GAGCT
M11-L2-25	CCAG/i5MedC/GATAGAACA/i5MedC/GCTCC
M11-L2-26	AATG/i5MedC/GTCTCAAGG/i5MedC/GCAGA
M11-L2-27	TATC/i5MedC/GCCACTAGC/i5MedC/GGAAG
M11-L2-28	AGTA/i5MedC/GGAATAATC/i5MedC/GGTAG
M11-L2-29	AACA/i5MedC/GAGTTAGGT/i5MedC/GTCAT
M11-L2-30	AGGT/i5MedC/GAAGCTACT/i5MedC/GAATA
M11-L2-31	TTGA/i5MedC/GCTGACCAA/i5MedC/GTACC
M11-L2-32	ACAA/i5MedC/GTAAGACTA/i5MedC/GTAAC
M11-L2-33	ATAT/i5MedC/GCATTGATG/i5MedC/GCTAA
M11-L2-34	CATG/i5MedC/GGAATGAAC/i5MedC/GTAGC
M11-L2-35	AATA/i5MedC/GTAGTGCAC/i5MedC/GAGCC
Brick strands complementary to template L3	
L3-1	TGTCCGGAGCTTCACGCTGA
L3-2	ACTACGTACTATTCCGTGCA
L3-3	GCTCCGAACTAGACCGATCT
L3-4	CACACGCATGTTACCGCTGT
L3-5	GATGCGGAATCTTCCGACCT
L3-6	GACTCGAAGAATGTCGACAA
L3-7	TAACCGACAGTAATCGGTAG
L3-8	CTCTCGTCTAGTGACGGCAC
L3-9	CTACCGATGTTGCTCGATAC
L3-10	AGATCGATCTGCCTCGAAGA
L3-11	CTATCGAGTGAACTCGCCTC
L3-12	GGTACGCAAGGAAGCGTGAT
L3-13	CAACCGTAATCCTACGAACT
L3-14	GAAGCGTCTGCTAGCGCTTA
L3-15	GCTTCGTATTCTAACGTCAG
L3-16	CAATCGGTGGTTACCGTTCT
L3-17	CAACCGCCTACTCTCGGTTG
L3-18	AATTCGGAGATAGCCGGCCT
L3-19	AGTGCGCAGGATGCCGCTAA
L3-20	ATGTCGAAGGCATTCTGTTGT

L3-21	CCTACGCTACTCAGCGGAAC
L3-22	GTTCCGGTAAGAGTCGTCTA
L3-23	ACAGCGCAGTAATGCGAAGT
L3-24	AGTTCGCCATTAGTCGAACC
L3-25	GGTGCGATATTAATCGGCTT
L3-26	GGTGCGAATGATTGCGCACA
L3-27	CTCTCGAATGATATCGTCTA
L3-28	GATCCGGTTGTAGCCGGTCT
L3-29	CACACGGAGTTCACCGTGTC
L3-30	GTAACGATAGATACCGTTGC
L3-31	TGAGCGGCAGCTCACGGTTA
L3-32	AGATCGGTATGGTCCGCAGA
L3-33	TGAGCGGAAGGTTCCGGTAT
L3-34	ATTGCGTACTCATCCGCATA
L3-35	GCATCGAATAAGGTCGCTGA
M01-L3-1	TGTC/i5MedC/GGAGCTTCACGCTGA
M01-L3-2	ACTA/i5MedC/GTACTATTCCGTGCA
M01-L3-3	GCTC/i5MedC/GAACTAGACCGATCT
M01-L3-4	CACA/i5MedC/GCATGTTACCGCTGT
M01-L3-5	GATG/i5MedC/GGAATCTTCCGACCT
M01-L3-6	GAAT/i5MedC/GAAGAATGTGACAA
M01-L3-7	TAAC/i5MedC/GACAGTAATCGGTAG
M01-L3-8	CTCT/i5MedC/GTCTAGTGACGGCAC
M01-L3-9	CTAC/i5MedC/GATGTTGCTCGATAC
M01-L3-10	AGAT/i5MedC/GATCTGCCTCGAAGA
M01-L3-11	CTAT/i5MedC/GAGTGAACCTCGCCTC
M01-L3-12	GGTA/i5MedC/GCAAGGAAGCGTGAT
M01-L3-13	CAAC/i5MedC/GTAATCCTACGAACCT
M01-L3-14	GAAG/i5MedC/GTCTGCTAGCGCTTA
M01-L3-15	GCTT/i5MedC/GTATTCTAACGTCAG
M01-L3-16	CAAT/i5MedC/GGTGGTTACCGTTCT
M01-L3-17	CAAC/i5MedC/GCCTACTCTCGGTTG
M01-L3-18	AATT/i5MedC/GGAGATAGCCGGCCT
M01-L3-19	AGTG/i5MedC/GCAGGATGCCGCTAA
M01-L3-20	ATGT/i5MedC/GAAGGCATTCGTTGT
M01-L3-21	CCTA/i5MedC/GCTACTCAGCGGAAC
M01-L3-22	GTTC/i5MedC/GGTAAGAGTCGTCTA
M01-L3-23	ACAG/i5MedC/GCAGTAATGCGAAGT
M01-L3-24	AGTT/i5MedC/GCCATTAGTCGAACC
M01-L3-25	GGTG/i5MedC/GATATTAATCGGCTT
M01-L3-26	GGTG/i5MedC/GAATGATTGCGCACA
M01-L3-27	CTCT/i5MedC/GAATGATATCGTCTA
M01-L3-28	GATC/i5MedC/GGTTGTAGCCGGTCT
M01-L3-29	CACA/i5MedC/GGAGTTCACCGTGTC
M01-L3-30	GTAA/i5MedC/GATAGATACCGTTGC

M01-L3-31	TGAG/i5MedC/GGCAGCTCACGGTTA
M01-L3-32	AGAT/i5MedC/GGTATGGTCCGCAGA
M01-L3-33	TGAG/i5MedC/GGAAGGTTCCGGTAT
M01-L3-34	ATTG/i5MedC/GTACTCATCCGCATA
M01-L3-35	GCAT/i5MedC/GAATAAGGTCGCTGA
M10-L3-1	TGTCCGGAGCTTCA/i5MedC/GCTGA
M10-L3-2	ACTACGTACTATTC/i5MedC/GTGCA
M10-L3-3	GCTCCGAAC TAGAC/i5MedC/GATCT
M10-L3-4	CACACGCATGTTAC/i5MedC/GCTGT
M10-L3-5	GATGCGGAATCTTC/i5MedC/GACCT
M10-L3-6	GACTCGAAGAATGT/i5MedC/GACAA
M10-L3-7	TAACCGACAGTAAT/i5MedC/GGTAG
M10-L3-8	CTCTCGTCTAGTGA/i5MedC/GGCAC
M10-L3-9	CTACCGATGTTGCT/i5MedC/GATAC
M10-L3-10	AGATCGATCTGCCT/i5MedC/GAAGA
M10-L3-11	CTATCGAGTGA ACT/i5MedC/GCCTC
M10-L3-12	GGTACGCAAGGAAG/i5MedC/GTGAT
M10-L3-13	CAACCGTAATCCTA/i5MedC/GAACT
M10-L3-14	GAAGCGTCTGCTAG/i5MedC/GCTTA
M10-L3-15	GCTTCGTATTCTAA/i5MedC/GTCAG
M10-L3-16	CAATCGGTGGTTAC/i5MedC/GTTCT
M10-L3-17	CAACCGCCTACTCT/i5MedC/GGTTG
M10-L3-18	AATTCGGAGATAGC/i5MedC/GGCCT
M10-L3-19	AGTGCGCAGGATGC/i5MedC/GCTAA
M10-L3-20	ATGTCGAAGGCATT/i5MedC/GTTGT
M10-L3-21	CCTACGCTACTCAG/i5MedC/GGAAC
M10-L3-22	GTTCCGGTAAGAGT/i5MedC/GTCTA
M10-L3-23	ACAGCGCAGTAATG/i5MedC/GAAGT
M10-L3-24	AGTTCGCCATTAGT/i5MedC/GAACC
M10-L3-25	GGTGCGATATTAAT/i5MedC/GGCTT
M10-L3-26	GGTGCGAATGATTG/i5MedC/GCACA
M10-L3-27	CTCTCGAATGATAT/i5MedC/GTCTA
M10-L3-28	GATCCGGTTGTAGC/i5MedC/GGTCT
M10-L3-29	CACACGGAGTTCAC/i5MedC/GTGTC
M10-L3-30	GTAACGATAGATAC/i5MedC/GTTGC
M10-L3-31	TGAGCGGCAGCTCA/i5MedC/GGTTA
M10-L3-32	AGATCGGTATGGTC/i5MedC/GCAGA
M10-L3-33	TGAGCGGAAGGTTTC/i5MedC/GGTAT
M10-L3-34	ATTGCGTACTCATC/i5MedC/GCATA
M10-L3-35	GCATCGAATAAGGT/i5MedC/GCTGA
M11-L3-1	TGTC/i5MedC/GGAGCTTCA/i5MedC/GCTGA
M11-L3-2	ACTA/i5MedC/GTACTATTC/i5MedC/GTGCA
M11-L3-3	GCTC/i5MedC/GAACTAGAC/i5MedC/GATCT
M11-L3-4	CACA/i5MedC/GCATGTTAC/i5MedC/GCTGT
M11-L3-5	GATG/i5MedC/GGAATCTTC/i5MedC/GACCT

M11-L3-6	GACT/i5MedC/GAAGAATGT/i5MedC/GACAA
M11-L3-7	TAAC/i5MedC/GACAGTAAT/i5MedC/GGTAG
M11-L3-8	CTCT/i5MedC/GTCTAGTGA/i5MedC/GGCAC
M11-L3-9	CTAC/i5MedC/GATGTTGCT/i5MedC/GATAC
M11-L3-10	AGAT/i5MedC/GATCTGCCT/i5MedC/GAAGA
M11-L3-11	CTAT/i5MedC/GAGTGAAC/i5MedC/GCCTC
M11-L3-12	GGTA/i5MedC/GCAAGGAAG/i5MedC/GTGAT
M11-L3-13	CAAC/i5MedC/GTAATCCTA/i5MedC/GAACT
M11-L3-14	GAAG/i5MedC/GTCTGCTAG/i5MedC/GCTTA
M11-L3-15	GCTT/i5MedC/GTATTCTAA/i5MedC/GTCAG
M11-L3-16	CAAT/i5MedC/GGTGGTTAC/i5MedC/GTTCT
M11-L3-17	CAAC/i5MedC/GCCTACTCT/i5MedC/GGTTG
M11-L3-18	AATT/i5MedC/GGAGATAGC/i5MedC/GGCCT
M11-L3-19	AGTG/i5MedC/GCAGGATGC/i5MedC/GCTAA
M11-L3-20	ATGT/i5MedC/GAAGGCATT/i5MedC/GTTGT
M11-L3-21	CCTA/i5MedC/GCTACTCAG/i5MedC/GGAAC
M11-L3-22	GTTC/i5MedC/GGTAAGAGT/i5MedC/GTCTA
M11-L3-23	ACAG/i5MedC/GCAGTAATG/i5MedC/GAAGT
M11-L3-24	AGTT/i5MedC/GCCATTAGT/i5MedC/GAACC
M11-L3-25	GGTG/i5MedC/GATATTAAT/i5MedC/GGCTT
M11-L3-26	GGTG/i5MedC/GAATGATTG/i5MedC/GCACA
M11-L3-27	CTCT/i5MedC/GAATGATAT/i5MedC/GTCTA
M11-L3-28	GATC/i5MedC/GGTTGTAGC/i5MedC/GGTCT
M11-L3-29	CACA/i5MedC/GGAGTTCAC/i5MedC/GTGTC
M11-L3-30	GTAA/i5MedC/GATAGATAC/i5MedC/GTTGC
M11-L3-31	TGAG/i5MedC/GGCAGCTCA/i5MedC/GGTTA
M11-L3-32	AGAT/i5MedC/GGTATGGTC/i5MedC/GCAGA
M11-L3-33	TGAG/i5MedC/GGAAGGTTC/i5MedC/GGTAT
M11-L3-34	ATTG/i5MedC/GTACTCATC/i5MedC/GCATA
M11-L3-35	GCAT/i5MedC/GAATAAGGT/i5MedC/GCTGA
Brick strands complementary to template L4	
L4-1	CTAACGTTACCAACGAGTC
L4-2	TTGGCGCTTGATAGCGTAGG
L4-3	GATCCGTGGTAGAACGGTTC
L4-4	ATTACGTAGCAGAGCGTATA
L4-5	TATACGAGGCTCCACGCCAG
L4-6	GTTCCGGAGTTATCCGGTAG
L4-7	AATTCGTCAATTAGCGTTGA
L4-8	TTGACGACAACAATCGCCAC
L4-9	TTACCGTATTACTTCGTGAC
L4-10	TTGACGTGCCTTCACGCTCC
L4-11	ATATCGAACACTGTCGTGAA
L4-12	CATTCGTCATTGAACGAATC
L4-13	GAGGCGGATGACTCCGTTAC
L4-14	CTGCCGTGCACACACGTTCA

L4-15	AGACCGTCTTATTGCGGTGA
L4-16	GTTGCGCTGAGACACGCAAC
L4-17	TACTCGTGTAGTACCGAGTG
L4-18	GTCACGAGCTATCTCGGTGT
L4-19	CAATCGAAGCATTACGCAAT
L4-20	TGTCCGTAACTCACGACAA
L4-21	CAACCGATTACTATCGTAGT
L4-22	ATAGCGGTCAAGGTCGCTCT
L4-23	CTCACGCAATGAGCCGTGGT
L4-24	AGAGCGAGAGTGCTCGAACC
L4-25	GAGACGTAGGTGTACGACTA
L4-26	GCATCGTCTCCTAACGCTAT
L4-27	CCTACGAGTCAGTACGTGTG
L4-28	CATTCGCCAGTTGGCGACTC
L4-29	GTTCCGTGCTGCCACGTTAC
L4-30	AGGTCGCTCTACAGCGCTCT
L4-31	GACACGGTATGCACCGTCCA
L4-32	CAGACGTATAGACACGTCAA
L4-33	TGGACGAGTTGTGCCGTATG
L4-34	GATGCGTAGCAATACGATAC
L4-35	CATACGTAAGTCTACGGTGA
M01-L4-1	CTAA/i5MedC/GTTCACCAACGAGTC
M01-L4-2	TTGG/i5MedC/GCTTGATAGCGTAGG
M01-L4-3	GATC/i5MedC/GTGGTAGAACGGTTC
M01-L4-4	ATTA/i5MedC/GTAGCAGAGCGTATA
M01-L4-5	TATA/i5MedC/GAGGCTCCACGCCAG
M01-L4-6	GTTC/i5MedC/GGAGTTATCCGGTAG
M01-L4-7	AATT/i5MedC/GTCAATTAGCGTTGA
M01-L4-8	TTGA/i5MedC/GACAACAATCGCCAC
M01-L4-9	TTAC/i5MedC/GTATTACTTCGTGAC
M01-L4-10	TTGA/i5MedC/GTGCCTTCACGCTCC
M01-L4-11	ATAT/i5MedC/GAACACTGTCTGTGAA
M01-L4-12	CATT/i5MedC/GTCATTGAACGAATC
M01-L4-13	GAGG/i5MedC/GGATGACTCCGTTAC
M01-L4-14	CTGC/i5MedC/GTGCACACACGTTCA
M01-L4-15	AGAC/i5MedC/GTCTTATTGCGGTGA
M01-L4-16	GTTG/i5MedC/GCTGAGACACGCAAC
M01-L4-17	TACT/i5MedC/GTGTAGTACCGAGTG
M01-L4-18	GTCA/i5MedC/GAGCTATCTCGGTGT
M01-L4-19	CAAT/i5MedC/GAAGCATTACGCAAT
M01-L4-20	TGTC/i5MedC/GTTAACTCACGACAA
M01-L4-21	CAAC/i5MedC/GATTACTATCGTAGT
M01-L4-22	ATAG/i5MedC/GGTCAAGGTCGCTCT
M01-L4-23	CTCA/i5MedC/GCAATGAGCCGTGGT
M01-L4-24	AGAG/i5MedC/GAGAGTGCTCGAACC

M01-L4-25	GAGA/i5MedC/GTAGGTGTACGACTA
M01-L4-26	GCAT/i5MedC/GTCTCCTAACGCTAT
M01-L4-27	CCTA/i5MedC/GAGTCAGTACGTGTG
M01-L4-28	CATT/i5MedC/GCCAGTTGGCGACTC
M01-L4-29	GTTC/i5MedC/GTGCTGCCACGTTAC
M01-L4-30	AGGT/i5MedC/GCTCTACAGCGCTCT
M01-L4-31	GACA/i5MedC/GGTATGCACCGTCCA
M01-L4-32	CAGA/i5MedC/GTATAGACACGTCAA
M01-L4-33	TGGA/i5MedC/GAGTTGTGCCGTATG
M01-L4-34	GATG/i5MedC/GTAGCAATACGATAC
M01-L4-35	CATA/i5MedC/GTAAGTCTACGGTGA
M10-L4-1	CTAACGTTACCAA/i5MedC/GAGTC
M10-L4-2	TTGGCGCTTGATAG/i5MedC/GTAGG
M10-L4-3	GATCCGTGGTAGAA/i5MedC/GGTTC
M10-L4-4	ATTACGTAGCAGAG/i5MedC/GTATA
M10-L4-5	TATACGAGGCTCCA/i5MedC/GCCAG
M10-L4-6	GTTCCGGAGTTATC/i5MedC/GGTAG
M10-L4-7	AATTCGTCAATTAG/i5MedC/GTTGA
M10-L4-8	TTGACGACAACAAT/i5MedC/GCCAC
M10-L4-9	TTACCGTATTACTT/i5MedC/GTGAC
M10-L4-10	TTGACGTGCCTTCA/i5MedC/GCTCC
M10-L4-11	ATATCGAACACTGT/i5MedC/GTGAA
M10-L4-12	CATTCGTCATTGAA/i5MedC/GAATC
M10-L4-13	GAGGCGGATGACTC/i5MedC/GTTAC
M10-L4-14	CTGCCGTGCACACA/i5MedC/GTTCA
M10-L4-15	AGACCGTCTTATTG/i5MedC/GGTGA
M10-L4-16	GTTGCGCTGAGACA/i5MedC/GCAAC
M10-L4-17	TACTCGTGTAGTAC/i5MedC/GAGTG
M10-L4-18	GTCACGAGCTATCT/i5MedC/GGTGT
M10-L4-19	CAATCGAAGCATTA/i5MedC/GCAAT
M10-L4-20	TGTCCGTTAACTCA/i5MedC/GACAA
M10-L4-21	CAACCGATTACTAT/i5MedC/GTAGT
M10-L4-22	ATAGCGGTCAAGGT/i5MedC/GCTCT
M10-L4-23	CTCACGCAATGAGC/i5MedC/GTGGT
M10-L4-24	AGAGCGAGAGTGCT/i5MedC/GAACC
M10-L4-25	GAGACGTAGGTGTA/i5MedC/GACTA
M10-L4-26	GCATCGTCTCCTAA/i5MedC/GCTAT
M10-L4-27	CCTACGAGTCAGTA/i5MedC/GTGTG
M10-L4-28	CATTCGCCAGTTGG/i5MedC/GACTC
M10-L4-29	GTTCCGTGCTGCCA/i5MedC/GTTAC
M10-L4-30	AGGTCGCTCTACAG/i5MedC/GCTCT
M10-L4-31	GACACGGTATGCAC/i5MedC/GTCCA
M10-L4-32	CAGACGTATAGACA/i5MedC/GTCAA
M10-L4-33	TGGACGAGTTGTGC/i5MedC/GTATG
M10-L4-34	GATGCGTAGCAATA/i5MedC/GATAC

M10-L4-35	CATACGTAAGTCTA/i5MedC/GGTGA
M11-L4-1	CTAA/i5MedC/GTTCACCAA/i5MedC/GAGTC
M11-L4-2	TTGG/i5MedC/GCTTGATAG/i5MedC/GTAGG
M11-L4-3	GATC/i5MedC/GTGGTAGAA/i5MedC/GGTTC
M11-L4-4	ATTA/i5MedC/GTAGCAGAG/i5MedC/GTATA
M11-L4-5	TATA/i5MedC/GAGGCTCCA/i5MedC/GCCAG
M11-L4-6	GTTC/i5MedC/GGAGTTATC/i5MedC/GGTAG
M11-L4-7	AATT/i5MedC/GTCAATTAG/i5MedC/GTTGA
M11-L4-8	TTGA/i5MedC/GACAACAAT/i5MedC/GCCAC
M11-L4-9	TTAC/i5MedC/GTATTACTT/i5MedC/GTGAC
M11-L4-10	TTGA/i5MedC/GTGCCTTCA/i5MedC/GCTCC
M11-L4-11	ATAT/i5MedC/GAACACTGT/i5MedC/GTGAA
M11-L4-12	CATT/i5MedC/GTCATTGAA/i5MedC/GAATC
M11-L4-13	GAGG/i5MedC/GGATGACTC/i5MedC/GTTAC
M11-L4-14	CTGC/i5MedC/GTGCACACA/i5MedC/GTTCA
M11-L4-15	AGAC/i5MedC/GTCTTATTG/i5MedC/GGTGA
M11-L4-16	GTTG/i5MedC/GCTGAGACA/i5MedC/GCAAC
M11-L4-17	TACT/i5MedC/GTGTAGTAC/i5MedC/GAGTG
M11-L4-18	GTCA/i5MedC/GAGCTATCT/i5MedC/GGTGT
M11-L4-19	CAAT/i5MedC/GAAGCATTA/i5MedC/GCAAT
M11-L4-20	TGTC/i5MedC/GTTAACTCA/i5MedC/GACAA
M11-L4-21	CAAC/i5MedC/GATTACTAT/i5MedC/GTAGT
M11-L4-22	ATAG/i5MedC/GGTCAAGGT/i5MedC/GCTCT
M11-L4-23	CTCA/i5MedC/GCAATGAGC/i5MedC/GTGGT
M11-L4-24	AGAG/i5MedC/GAGAGTGCT/i5MedC/GAACC
M11-L4-25	GAGA/i5MedC/GTAGGTGTA/i5MedC/GACTA
M11-L4-26	GCAT/i5MedC/GTCTCCTAA/i5MedC/GCTAT
M11-L4-27	CCTA/i5MedC/GAGTCAGTA/i5MedC/GTGTG
M11-L4-28	CATT/i5MedC/GCCAGTTGG/i5MedC/GACTC
M11-L4-29	GTTC/i5MedC/GTGCTGCCA/i5MedC/GTTAC
M11-L4-30	AGGT/i5MedC/GCTCTACAG/i5MedC/GCTCT
M11-L4-31	GACA/i5MedC/GGTATGCAC/i5MedC/GTCCA
M11-L4-32	CAGA/i5MedC/GTATAGACA/i5MedC/GTCAA
M11-L4-33	TGGA/i5MedC/GAGTTGTGC/i5MedC/GTATG
M11-L4-34	GATG/i5MedC/GTAGCAATA/i5MedC/GATAC
M11-L4-35	CATA/i5MedC/GTAAGTCTA/i5MedC/GGTGA
Brick strands complementary to template L5	
L5-1	GGTTCGCTTAATAGCGGAAG
L5-2	TTACCGTGTGAATCCGGTCT
L5-3	ACTGCGCCATTGGCCGAATT
L5-4	GCTCCGATCTGAATCGACCT
L5-5	AGTACGGTTGTGACCGGACA
L5-6	GATCCGTTACCATTCGTAAC
L5-7	TGTGCGAGAATCTGCGCATA
L5-8	GTTGCGCTATGGAGCGAGTA

L5-9	ACTCCGCTCACTAGCGATAT
L5-10	GGCACGAGTAGTGTCTGAATC
L5-11	GGATCGCACAGATGCGTGGT
L5-12	TTCTCGAGTCTTCTCGTATA
L5-13	CAGTCGAGTCTCTACGGAAC
L5-14	CCATCGCCAACAGGCGATTG
L5-15	AGGTCGGCTGTAGCCGATTA
L5-16	AGCACGTGGAACTTCGGTGC
L5-17	CAGCCGTCTTATAACCGGCAC
L5-18	AAGGCGCTCCTCACCGATAT
L5-19	ATTGCGCCAAGTAGCGCATA
L5-20	CTCACGGTTGCCACCGTTGA
L5-21	ACAACGATTATGGACGTTCT
L5-22	ACAGCGACAAGATCCGCTCA
L5-23	TGGACGAGATTCTCTCGGACC
L5-24	ATGCCGTGTTTCATGCGGCTT
L5-25	TGTTTCGTCCTGTACGAATG
L5-26	TCCACGCTGTTATGCGTAGT
L5-27	AGGTCGTGAGCATAACGATTG
L5-28	GAAGCGATTGGACTCGCTCC
L5-29	CAAGCGACAGATGTCTGCATA
L5-30	TCATCGATTCCATGCGATGA
L5-31	TGCTCGACCTAAGTCGACTT
L5-32	GTTCCGATCTGTTCCGCAGA
L5-33	CCACCGCTGAATTGCGGAAG
L5-34	TGTACGGTAGATAACGCTAG
L5-35	CAGGCGTGTCTTAACGCAGT
M01-L5-1	GGTT/i5MedC/GCTTAATAGCGGAAG
M01-L5-2	TTAC/i5MedC/GTGTGAATCCGGTCT
M01-L5-3	ACTG/i5MedC/GCCATTGGCCGAATT
M01-L5-4	GCTC/i5MedC/GATCTGAATCGACCT
M01-L5-5	AGTA/i5MedC/GGTTGTGACCGGACA
M01-L5-6	GATC/i5MedC/GTTACCATTTCGTAAC
M01-L5-7	TGTG/i5MedC/GAGAATCTGCGCATA
M01-L5-8	GTTG/i5MedC/GCTATGGAGCGAGTA
M01-L5-9	ACTC/i5MedC/GCTCACTAGCGATAT
M01-L5-10	GGCA/i5MedC/GAGTAGTGTCTGAATC
M01-L5-11	GGAT/i5MedC/GCACAGATGCGTGGT
M01-L5-12	TTCT/i5MedC/GAGTCTTCTCGTATA
M01-L5-13	CAGT/i5MedC/GAGTCTCTACGGAAC
M01-L5-14	CCAT/i5MedC/GCCAACAGGCGATTG
M01-L5-15	AGGT/i5MedC/GGCTGTAGCCGATTA
M01-L5-16	AGCA/i5MedC/GTGGAACCTTCGGTGC
M01-L5-17	CAGC/i5MedC/GTCTTATAACCGGCAC
M01-L5-18	AAGG/i5MedC/GCTCCTCACCGATAT

M01-L5-19	ATTG/i5MedC/GCCAAGTAGCGCATA
M01-L5-20	CTCA/i5MedC/GGTTGCCACCGTTGA
M01-L5-21	ACAA/i5MedC/GATTATGGACGTTCT
M01-L5-22	ACAG/i5MedC/GACAAGATCCGCTCA
M01-L5-23	TGGA/i5MedC/GAGATTCCTCGGACC
M01-L5-24	ATGC/i5MedC/GTGTTCATGCGGCTT
M01-L5-25	TGTT/i5MedC/GTCACTGTACGAATG
M01-L5-26	TCCA/i5MedC/GCTGTTATGCGTAGT
M01-L5-27	AGGT/i5MedC/GTGAGCATAACGATTG
M01-L5-28	GAAG/i5MedC/GATTGGACTCGCTCC
M01-L5-29	CAAG/i5MedC/GACAGATGTTCGCATA
M01-L5-30	TCAT/i5MedC/GATTCCATGCGATGA
M01-L5-31	TGCT/i5MedC/GACCTAAGTCGACTT
M01-L5-32	GTTC/i5MedC/GATCTGTTCCGCAGA
M01-L5-33	CCAC/i5MedC/GCTGAATTGCGGAAG
M01-L5-34	TGTA/i5MedC/GGTAGATAACGCTAG
M01-L5-35	CAGG/i5MedC/GTGTCTTAACGCAGT
M10-L5-1	GGTTCGCTTAATAG/i5MedC/GGAAG
M10-L5-2	TTACCGTGTGAATC/i5MedC/GGTCT
M10-L5-3	ACTGCGCCATTGGC/i5MedC/GAATT
M10-L5-4	GCTCCGATCTGAAT/i5MedC/GACCT
M10-L5-5	AGTACGGTTGTGAC/i5MedC/GGACA
M10-L5-6	GATCCGTTACCATT/i5MedC/GTAAC
M10-L5-7	TGTGCGAGAATCTG/i5MedC/GCATA
M10-L5-8	GTTGCGCTATGGAG/i5MedC/GAGTA
M10-L5-9	ACTCCGCTCACTAG/i5MedC/GATAT
M10-L5-10	GGCACGAGTAGTGT/i5MedC/GAATC
M10-L5-11	GGATCGCACAGATG/i5MedC/GTGGT
M10-L5-12	TTCTCGAGTCTTCT/i5MedC/GTATA
M10-L5-13	CAGTCGAGTCTCTA/i5MedC/GGAAC
M10-L5-14	CCATCGCCAACAGG/i5MedC/GATTG
M10-L5-15	AGGTCGGCTGTAGC/i5MedC/GATTA
M10-L5-16	AGCACGTGGAACCTT/i5MedC/GGTGC
M10-L5-17	CAGCCGTCTTATAC/i5MedC/GGCAC
M10-L5-18	AAGGCGCTCCTCAC/i5MedC/GATAT
M10-L5-19	ATTGCGCCAAGTAG/i5MedC/GCATA
M10-L5-20	CTCACGGTTGCCAC/i5MedC/GTTGA
M10-L5-21	ACAACGATTATGGA/i5MedC/GTTCT
M10-L5-22	ACAGCGACAAGATC/i5MedC/GCTCA
M10-L5-23	TGGACGAGATTCCT/i5MedC/GGACC
M10-L5-24	ATGCCGTGTTTCATG/i5MedC/GGCTT
M10-L5-25	TGTTTCGTCCTGTGTA/i5MedC/GAATG
M10-L5-26	TCCACGCTGTTATG/i5MedC/GTAGT
M10-L5-27	AGGTCGTGAGCATA/i5MedC/GATTG
M10-L5-28	GAAGCGATTGGACT/i5MedC/GCTCC

M10-L5-29	CAAGCGACAGATGT/i5MedC/GCATA
M10-L5-30	TCATCGATTCCATG/i5MedC/GATGA
M10-L5-31	TGCTCGACCTAAGT/i5MedC/GACTT
M10-L5-32	GTTCCGATCTGTTC/i5MedC/GCAGA
M10-L5-33	CCACCGCTGAATTG/i5MedC/GGAAG
M10-L5-34	TGTACGGTAGATAA/i5MedC/GCTAG
M10-L5-35	CAGGCGTGTCTTAA/i5MedC/GCAGT
M11-L5-1	GGTT/i5MedC/GCTTAATAG/i5MedC/GGAAG
M11-L5-2	TTAC/i5MedC/GTGTGAATC/i5MedC/GGTCT
M11-L5-3	ACTG/i5MedC/GCCATTGGC/i5MedC/GAATT
M11-L5-4	GCTC/i5MedC/GATCTGAAT/i5MedC/GACCT
M11-L5-5	AGTA/i5MedC/GGTTGTGAC/i5MedC/GGACA
M11-L5-6	GATC/i5MedC/GTTACCATT/i5MedC/GTAAC
M11-L5-7	TGTG/i5MedC/GAGAATCTG/i5MedC/GCATA
M11-L5-8	GTTG/i5MedC/GCTATGGAG/i5MedC/GAGTA
M11-L5-9	ACTC/i5MedC/GCTCACTAG/i5MedC/GATAT
M11-L5-10	GGCA/i5MedC/GAGTAGTGT/i5MedC/GAATC
M11-L5-11	GGAT/i5MedC/GCACAGATG/i5MedC/GTGGT
M11-L5-12	TTCT/i5MedC/GAGTCTTCT/i5MedC/GTATA
M11-L5-13	CAGT/i5MedC/GAGTCTCTA/i5MedC/GGAAC
M11-L5-14	CCAT/i5MedC/GCCAACAGG/i5MedC/GATTG
M11-L5-15	AGGT/i5MedC/GGCTGTAGC/i5MedC/GATTA
M11-L5-16	AGCA/i5MedC/GTGGAACCT/i5MedC/GGTGC
M11-L5-17	CAGC/i5MedC/GTCTTATAC/i5MedC/GGCAC
M11-L5-18	AAGG/i5MedC/GCTCCTCAC/i5MedC/GATAT
M11-L5-19	ATTG/i5MedC/GCCAAGTAG/i5MedC/GCATA
M11-L5-20	CTCA/i5MedC/GGTTGCCAC/i5MedC/GTTGA
M11-L5-21	ACAA/i5MedC/GATTATGGA/i5MedC/GTTCT
M11-L5-22	ACAG/i5MedC/GACAAGATC/i5MedC/GCTCA
M11-L5-23	TGGA/i5MedC/GAGATTCCT/i5MedC/GGACC
M11-L5-24	ATGC/i5MedC/GTGTTTCATG/i5MedC/GGCTT
M11-L5-25	TGTT/i5MedC/GTCACTGTA/i5MedC/GAATG
M11-L5-26	TCCA/i5MedC/GCTGTTATG/i5MedC/GTAGT
M11-L5-27	AGGT/i5MedC/GTGAGCATA/i5MedC/GATTG
M11-L5-28	GAAG/i5MedC/GATTGGACT/i5MedC/GCTCC
M11-L5-29	CAAG/i5MedC/GACAGATGT/i5MedC/GCATA
M11-L5-30	TCAT/i5MedC/GATTCCATG/i5MedC/GATGA
M11-L5-31	TGCT/i5MedC/GACCTAAGT/i5MedC/GACTT
M11-L5-32	GTTC/i5MedC/GATCTGTTC/i5MedC/GCAGA
M11-L5-33	CCAC/i5MedC/GCTGAATTG/i5MedC/GGAAG
M11-L5-34	TGTA/i5MedC/GGTAGATAA/i5MedC/GCTAG
M11-L5-35	CAGG/i5MedC/GTGTCTTAA/i5MedC/GCAGT

Supplementary references

1. Ceze, L., Nivala, J. & Strauss, K. Molecular digital data storage using DNA. *Nat Rev Genet* **20**, 456–466 (2019).
2. 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).
3. Verardo, D. et al. Multiplex enzymatic synthesis of DNA with single-base resolution. *Sci. Adv.* **9**, eadi0263 (2023).
4. Kant, C. et al. Large area inkjet-printed OLED fabrication with solution-processed TADF ink. *Nat. Commun.* **14**, 7220 (2023).
5. Nguyen, B. H. et al. Scaling DNA data storage with nanoscale electrode wells. *Sci. Adv.* **7**, eabi6714 (2021).
6. Erlich Y. & Zielinski D. DNA Fountain enables a robust and efficient storage architecture. *Science* **355**, 950-954 (2017).
7. Roquet, N. et al. DNA-based data storage via combinatorial assembly. *bioRxiv* 2021.04.20.440194 (2021).
8. Meiser, L. C. et al. Reading and writing digital data in DNA. *Nat. Protoc.* **15**, 86-101(2020).
9. Lee, H. H. et al. Terminator-free template-independent enzymatic DNA synthesis for digital information storage. *Nat. Commun.* **10**, 2383 (2019).
10. Zhou, Y., Dong, J. & Wang, Q. Fabricating higher-order functional DNA origami structures to reveal biological processes at multiple scales. *NPG Asia Mater* **15**, 25 (2023).
11. Sun, F. et al. Mobile and Self-Sustained Data Storage in an Extremophile Genomic DNA. *Advanced Science* **10**, 2206201 (2023).
12. Yuen, Z. W. S. et al. Systematic benchmarking of tools for CpG methylation detection from nanopore sequencing. *Nat. Commun.* **12**, 3438 (2021).
13. Adam, S. et al. DNA sequence-dependent activity and base flipping mechanisms of DNMT1 regulate genome-wide DNA methylation. *Nat. Commun.* **11**, 3723 (2020).
14. Nanoporetech: Megalodon. <https://github.com/nanoporetech/megalodon>.
15. Koch, J. et al. A DNA-of-things storage architecture to create materials with embedded memory. *Nature Biotechnology* **38**, 39-43 (2020).
16. Niu, K., Chen, K., Lin, J. R. & Zhang, Q. T. Polar codes: Primary concepts and practical decoding algorithms. *IEEE Communications Magazine* **52**, 192-203 (2014).

17. Church, G. M, Gao, Y. & Kosuri, S. Next-generation digital information storage in DNA. *Science* **337**, 1628-1628 (2012).
18. Goldman, N. et al. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature* **494**, 77–80 (2013).
19. Winston, C. et al. Combinatorial PCR method for efficient, selective oligo retrieval from complex oligo pools. *ACS Synthetic Biology* **11**, 1727-1734 (2022).
20. Organick L. et al. Probing the physical limits of reliable DNA data retrieval. *Nat Commun.* **11**, 616 (2020).
21. Dong Y., Sun F., Ping Z., Ouyang Q. & Qian L. DNA storage: research landscape and future prospects. *National Science Review* **7**, 1092-1107 (2020).
22. Wood, R. J., McKelvie, J. C., Maynard-Smith, M. D. & Roach, P. L. A real-time assay for CpG-specific cytosine-C5 methyltransferase activity. *Nucleic Acids Research* **38**, e107(2010).
23. Sun, Y. et al. A novel restriction endonuclease Glal for rapid and highly sensitive detection of DNA methylation coupled with isothermal exponential amplification reaction. *Chem. Sci.* **9**, 1344-1351(2018).
24. Zadeh, J. N. et al. NUPACK: Analysis and Design of Nucleic Acid Systems. *Journal of Computational Chemistry* **32**, 170–173 (2011).
25. Stoiber, M. H. et al. De novo identification of DNA modifications enabled by genome-guided nanopore signal processing. <http://biorxiv.org/content/early/2017/04/10/094672> (2016).
26. Simpson, J. “Nanopolish” Github, <https://github.com/jts/nanopolish> (2022).

Acknowledgements to the student volunteers:

Baolin Li, Bojun Xie, Boyuan Yang, Yinghao Liang, Chengyu Fan, Chenqi Zhao, Chenying Wang, Chun-Kai Fan, Dengke Hu, Guangxin Zhang, Guangyue Peng, Hsiu-Yuan Huang, Junbo Zheng, Lean Wang, Haohua Luo, Qingwen Yang, Rongao Yuan, Ruidong Zhu, Runkai Chen, Xuan Liu, Sijia Yao, Siqi Wang, Qian Lin, Siyue Li, Tiaojiao Yan, Wanting Liu, Wei He, Weimin Xiong, Wenchun Jing, Wenzhen Yu, Xiangwei Deng, Yujie Chen, Xiaoming Wang, Zhihao Zhang, Xinxiaobao Xie, Yanjin He, Yanru He, Yao Zhao, Ye Wang, Yecheng Xue, Yifan Zhang, Yiliang Zhao, Yimang Xu, Yining Shi, Yisheng Lin, Yiyang Liao, Yiyang Zhang, Yiming Chen, Yuanshi Huang, Yuanyi Zhang, Yue Fang, Yuhanyan Song, Yusong Lei, Yuyang Han, Yuyang Zhang, Zeyan Yin, Zhen Li, Zichen Song, Zikang Gong, Ziteng Li.