nature biotechnology

Random access in large-scale DNA data storage

Lee Organick[1], Siena Dumas Ang[2], Yuan-Jyue Chen[2], Randolph Lopez[3], Sergey Yekhanin[2], Konstantin Makarychev[2,5], Miklos Z Racz[2,5], Govinda Kamath[2,5], Parikshit Gopalan[2,5], Bichlien Nguyen[2], Christopher N Takahashi[1], Sharon Newman[1,5], Hsing-Yeh Parker[2], Cyrus Rashtchian[2], Kendall Stewart[1], Gagan Gupta[2], Robert Carlson[2], John Mulligan[2], Douglas Carmean[2], Georg Seelig[1,4], Luis Ceze[1] & Karin Strauss[2]

Synthetic DNA is durable and can encode digital data with high density, making it an attractive medium for data storage. However, recovering stored data on a large-scale currently requires all the DNA in a pool to be sequenced, even if only a subset of the information needs to be extracted. Here, we encode and store 35 distinct files (over 200 MB of data), in more than 13 million DNA oligonucleotides, and show that we can recover each file individually and with no errors, using a random access approach. We design and validate a large library of primers that enable individual recovery of all files stored within the DNA. We also develop an algorithm that greatly reduces the sequencing read coverage required for error-free decoding by maximizing information from all sequence reads. These advances demonstrate a viable, large-scale system for DNA data storage and retrieval.

Storing digital data using synthetic DNA requires information to be encoded into nucleotide sequences and the corresponding molecules to be synthesized and stored in an appropriate environment. To extract the stored information, one has to sequence the DNA and decode it back into digital data. Here, we provide an end-to-end DNA storage workflow (**Fig. 1a**). We focus on scaling up data volumes and solving the associated challenges. Specifically, we address the need to access data selectively, rather than in bulk, to minimize the amount of sequencing required to recover the desired stored data.

For many years, high cost and low throughput have limited the applications of DNA data as a storage medium[1,2]. Recently, various groups have observed that the biotechnology industry has made substantial progress and DNA data storage is nearing practical use[3–10]. However, most prior DNA data storage efforts sequenced and decoded the entire amount of stored information, with no random access[3–7]. However, this type of redundant sequencing becomes impractical as the amount of data increases (**Fig. 1b**,**c**). Being able to selectively access only part of the written information (e.g., retrieving only one image from a collection) is therefore necessary to make DNA data storage viable, but so far accessing part of stored information has only been demonstrated on a small scale[8–10]. Our work demonstrates that PCR-based random access can be scaled up to reliably extract files of widely varying size and complexity from a DNA pool three orders of magnitude larger than those used in prior random access experiments.

Both DNA synthesis and sequencing are highly error-prone[11]. It is not unusual to observe aggregate insertion, deletion, and substitution rates at ~0.01 errors/base. Even complete loss of specific data strands can occur during library synthesis or amplification. Prior work has shown that it is possible to recover data even from such noisy conditions if proper encoding schemes are used. Although efforts have been made to minimize the amount of logical redundancy (i.e., the amount of additional information encoded) required for complete data recovery at a given error rate, existing approaches rely on a high degree of sequencing redundancy (i.e., having many copies of each sequence and deep sequencing coverage).

Here, we present a coding algorithm that explicitly reduces sequencing redundancy, hence requiring fewer sequencing resources and, in turn, fewer physical copies of any given molecule to fully recover the stored data. Our scheme tolerates aggressive settings of uneven low coverage and high coordinate error rates of insertions, deletions, and substitutions (**Supplementary Note 1** and **Supplementary Table 1**), while maintaining a logical density (bits per nucleotide) competitive with previously proposed schemes (**Fig. 1d**, **Supplementary Note 2**, and **Supplementary Table 2**). As DNA data storage technology matures, the goals of increasing throughput and lowering costs will likely drive coordinate error rates in the DNA data storage channel even higher than the current value.

To investigate challenges associated with increasing DNA data storage size, we created a large DNA library of modern data types, such as high-definition video, images, audio, and text. These included the "Universal Declaration of Human Rights" in over 100 languages (doi:10.1080/13642989808406748; http://www.ohchr.org/EN/UDHR/Pages/UDHRIndex.aspx), a high-definition music video of the band "OK Go" (https://www.youtube.com/watch?v=qybUFnY7Y8w), and a CropTrust database of the seeds stored in the Svalbard Global Seed Vault (https://www.nordgen.org/sgsv).

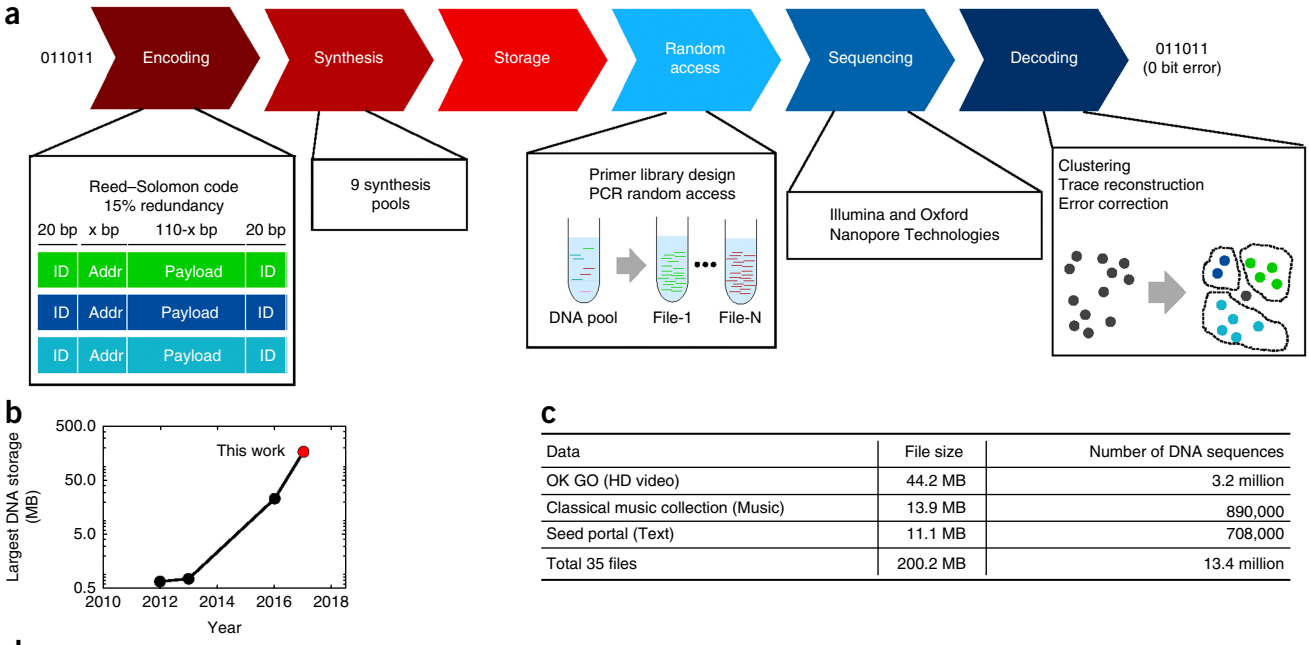
RESULTS

Coding method and random-access primer design

We encoded 35 files ranging from 29 KB to over 44 MB, totaling over 200 MB of unique (compressed) data (**Fig. 1c** lists a few examples;

[1]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, Washington, USA. [2]Microsoft Research, Redmond, Washington, USA. [3]Department of Bioengineering Department, University of Washington, Seattle, Washington, USA. [4]Department of Electrical Engineering, University of Washington, Seattle, Washington, USA. [5]Present addresses: VMware, Palo Alto, California, USA (P.G.); Stanford University, Stanford, California, USA (G.K. and S.N.); Northwestern University, Evanston, Illinois, USA (K.M.); Princeton University, Princeton, New Jersey, USA (M.Z.R.). Correspondence should be addressed to L.C. (luisceze@cs.washington.edu) or K.S. (kstrauss@microsoft.com).

Received 13 July 2017; accepted 11 January 2018; published online 19 February 2018; corrected after print 6 March 2018; doi:10.1038/nbt.4079



c

Data	File size	Number of DNA sequences
OK GO (HD video)	44.2 MB	3.2 million
Classical music collection (Music)	13.9 MB	890,000
Seed portal (Text)	11.1 MB	708,000
Total 35 files	200.2 MB	13.4 million

d

	Data size	Sequencing technology	Subsampled to low coverage	Coverage	Bits per base including primers	Bits per base excluding primers	Random access
Church *et al.*[3]	0.65 MB	Illumina	No	3,000x	0.60	0.83	No
Goldman *et al.*[4]	0.63 MB	Illumina	Yes	51x	0.19	0.29	No
Grass *et al.*[5]	0.08 MB	Illumina	No	372x	0.86	1.16	No
Bornholt *et al.*[9]	0.15 MB	Illumina	Yes	40x	0.57	0.85	Yes
Erlich and Zielinski[7]	2.11 MB	Illumina	Yes	10.5x	1.18	1.55	No
Blawat *et al.*[6]	22 MB	Illumina	No	160x	0.89	1.08	No
This work	200.2 MB	Illumina	Yes	5x	0.81	1.10	Yes
Yadzi *et al.*[10]	0.003 MB	Nanopore	No	200x	1.71	1.74	Yes
This work	0.033 MB	Nanopore	Yes	36x / 80x	0.81	1.10	Yes

Figure 1 Overview of the DNA data storage workflow and stored data. (**a**) The encoding process maps digital files into a large set of 150-nucleotide DNA sequences, including Reed–Solomon code redundancy to overcome errors in synthesis and sequencing. The resulting collection of sequences is synthesized by Twist Bioscience. The random access process starts with amplifying a subset of the sequences corresponding to one of the files using PCR. The amplified pools are sequenced using either sequencing by synthesis (Illumina NextSeq) or nanopore sequencing (Oxford Nanopore Technologies). Finally, sequencing reads are decoded using our clustering, consensus and error correction algorithms. (**b**) We encoded a total of 200 MB of data, about an order of magnitude more than prior work. (**c**) Example files encoded within these 200 MB of data. (**d**) A comparison to prior work shows that our coding scheme has similar logical redundancy, but requires lower sequencing coverage to recover files.

Supplementary Note 3 and **Supplementary Table 3** provide the full list). We added 15% logical redundancy for robust error correction to 33 of our files and 25% to the other two, resulting in an additional 32.2 MB of data encoded in DNA. For DNA synthesis, we segmented each input file into a large number of oligonucleotides, each containing the same PCR primer target sequences that form a unique file ID. Moreover, each strand also includes a unique, strand-specific address to order strands within a file. The resulting synthetic DNA library contains 13,448,372 unique DNA sequences of lengths ranging from 150 to 154 bases, synthesized using Twist Bioscience's oligo pool services in a total of nine synthesis pools. Our resulting combined pool of about 2 billion bases represents an increase of about an order of magnitude in the amount of information stored in and retrieved from DNA, relative to prior work[6].

Achieving robust random access in a large DNA data storage system requires effective PCR primers to reliably amplify a specific file without crosstalk. We thus devised a framework for designing a primer library with thousands of pairs of orthogonal primers (file IDs). Our design method (**Fig. 2a,i**, **Supplementary Note 4**, and **Supplementary Fig. 1**) optimizes primers for several properties: avoidance of secondary structure formation and primer–dimer formation, absence of long stretches of homopolymers, melting temperature constrained to a narrow range (55–60 °C), and a minimum of 30% of their sequence unique compared to other primers. To increase the stringency of sequence orthogonality, we used the basic sequence alignment program BLAST to screen out primers with long stretches of similar sequences[12]. Before incorporating selected primer sequences into the library files, we tested primer performance on a pool of 3,240 synthetic "mini-files" ranging from 1 to 200 103-mers. We successfully accessed and sequenced up to 48 mini-files from this pool in a one-pot multiplex PCR experiment (**Fig. 2a**, **ii**), validating our primer library design approach (**Supplementary Note 5** and **Supplementary Fig. 2**).

Next, we created a coding scheme to convert digital information to DNA sequences and back to digital information. Similar to prior

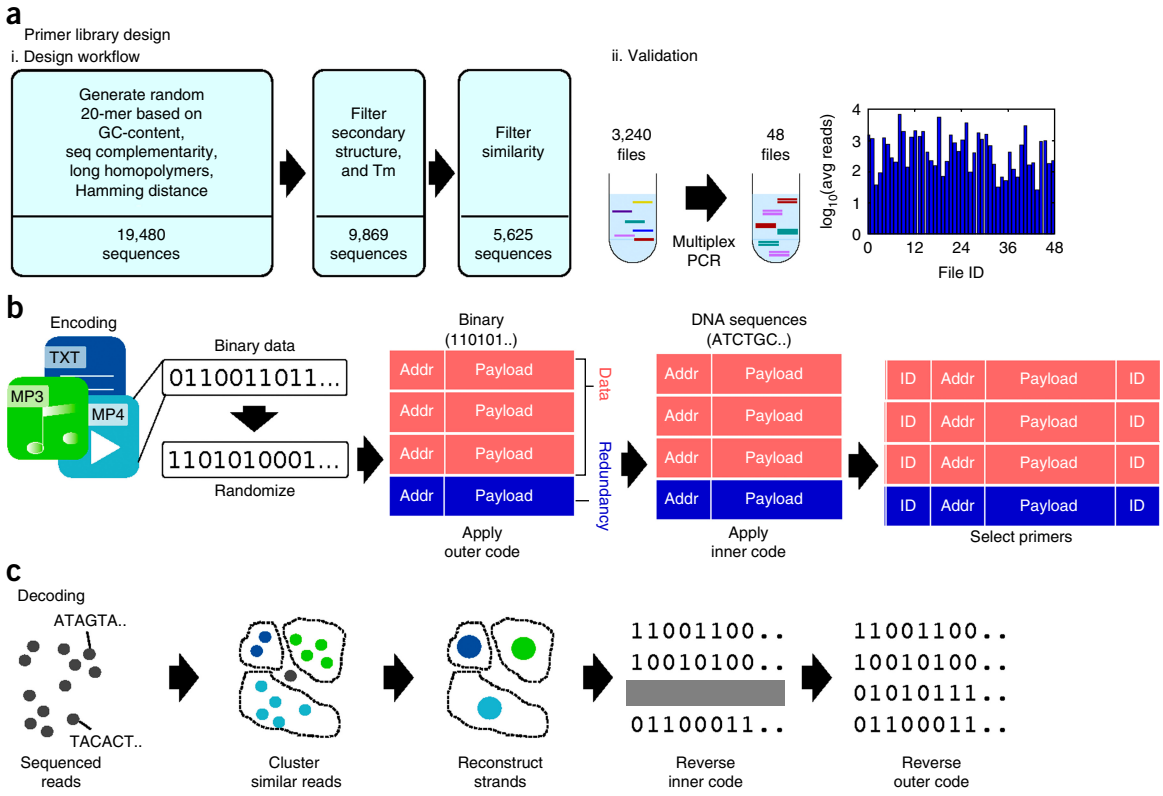


Figure 2 Design of random access primers and coding algorithm. (**a**, **i**) We designed a primer library for our PCR-based random access method using an *in silico* process. Starting with a set of random 20-mers, the sequences keep mutating until they satisfy all the design criteria, which include their GC-content, the absence of long sequence-complementarities, absence of long stretches of homopolymers, and a minimum Hamming distance of 6 bases from other primers. The preselected sequence set (19,480 sequences) is then filtered by melting temperature and a set that is as diverse as possible, that is, has low similarity between the sequences, is selected. (**a**, **ii**) The resulting set of candidate primers is then validated experimentally by synthesizing a pool of about 100,000 strands containing sets of size 1 to 200 DNA sequences each, surrounded by one of the 3,240 candidate primer pairs, and then randomly selecting 48 of those pairs for amplification. The product is sequenced, and sequences with each of the 48 primer pairs appear among sequencing reads, albeit at different relative proportions when normalized to the number of sequences in each set. (**b**) Our encoding process starts by randomizing data to reduce chances of secondary structures, primer–payload non-specific binding, and improved properties during decoding. It then breaks the data into fixed-size payloads, adds addressing information (Addr), and applies outer coding, which adds redundant sequences using a Reed–Solomon code to increase robustness to missing sequences and errors. The level of redundancy is determined by expected errors in sequencing and synthesis, as well as DNA degradation. Next, it applies inner coding, which ultimately converts the bits to DNA sequences. The resulting set of sequences is surrounded by a primer pair chosen from the library based on (low) level of overlap with payloads. (**c**) The decoding process starts by clustering reads based on similarity, and finding a consensus between the sequences in each cluster to reconstruct the original sequences, which are then decoded back to digital data.

work[5,6], our approach employs concatenated codes with Reed–Solomon (RS) as the outer code (**Fig. 2b**). (However, unlike most earlier work, we used very long codes (length up to 65,536) to handle large variations in the number of errors between code words.) Input data are then randomized by XOR with a pseudo-random sequence. Randomization facilitates coping with errors by breaking multi-bit repeats (e.g., 00000000) and ensures that the DNA sequences we produce are dissimilar, which makes decoding less computationally costly.

The encoder first partitions the randomized digital file into multiple blocks, up to a megabyte in size. We represent each block by a matrix M with up to ten rows and up to 55,000 columns, where every matrix cell carries a 16-bit value. Next, we encode each row of M with a Reed–Solomon code to obtain a larger matrix M′ that extends M by appending redundant columns. Every column of M′ is later converted into a DNA sequence of length 110 (114 for File 33; **Supplementary Note 3** and **Supplementary Table 3**). When Reed–Solomon redundancy is set to 15%, 87% of the DNA sequences carry raw input data (systematic RS coordinates), while 13% carry redundant data used for error correction (redundant RS coordinates).

The conversion of columns of M′ to DNA sequences involves representing a column in base 4, appending a prefix with address information (block index and column index), breaking the column into consecutive fragments of size three each, treating the content of each fragment as a number between 0 and 63 written in base four, representing this number in base three to obtain a fragment of size four, putting the new fragments together, and applying a rotating code[4] to turn a base-three representation into a base-four representation that eliminates homopolymers.

Finally, all DNA sequences are appended with 20-base PCR primer targets selected from the primer library on both ends to allow random access to the file (**Supplementary Note 6** and **Supplementary Figs. 3** and **4**). Resulting DNA sequences are synthesized into DNA strands, which can then be preserved using a variety of methods, and later selected via random access.

Our proposed random-access approach and associated primer design and conflict detection methodology scales to physically isolated pools of several terabytes each (**Supplementary Note 5**). In dehydrated spots, these would measure on the order of one millimeter, which in turn can be organized in dense arrays. Such a system would be orders of magnitude denser than tape.

When servicing a read request, we retrieve and rehydrate the DNA. Sequencing these DNA strands produces a collection of noisy reads, which do not necessarily include all original DNA sequences; sequences may be lost by sampling, storing, retrieving and preparing the DNA for sequencing. Sequences belonging to a specific file are obtained by aligning and filtering based on the primer sequence and length. Frugality with respect to coverage was a key consideration when designing our decoding approach. Therefore, we do not require reads to be the correct length[5], and we do not filter out reads with errors in their primer region[7]. Instead, noisy reads whose length is within 20 nucleotides of the original length are selected and passed to the decoder.

The decoder operates in four stages (**Fig. 2c**). First, it clusters noisy reads by similarity, based on their entire content, not just the addresses[8], to collect all available reads that likely correspond to one of the unique DNA sequences originally stored. To do so, we employ an algorithm that leverages the input randomization done during encoding. At a high level, we initially consider each noisy read a separate cluster and iteratively merge clusters based on random representatives, leveraging the fact that noisy reads of any specific DNA sequence are similar and noisy reads of different DNA sequences are dissimilar. Our algorithm runs in time that is close to linear in the input size and utilizes a series of filters to avoid unnecessary and slow edit distance computations. Using a locality-sensitive hashing scheme for edit distance, we compare only a small subset of representatives. We also use a lightweight check based on a binary embedding to further filter pairs. If a pair of representatives passes these two tests, edit distance determines whether the clusters are merged. A less computationally efficient, but functionally equivalent alternative, approach to clustering that uses off-the-shelf software is discussed in **Supplementary Note 7**.

The second stage of the decoder then processes each cluster to recover the original sequence. This stage, which we call trace reconstruction, uses a variant of the Bitwise Majority Alignment algorithm (BMA)[13], adapted to support insertions, deletions, and substitutions. The algorithm follows BMA in that pointers for noisy reads are maintained and moved from left to right, and at every step of the process the next symbol of the original sequence is estimated via a plurality vote. For the noisy reads that agree with plurality, the pointer is moved to the right by 1 (hypothesizing that the read had the correct symbol at the respective position), just like in BMA. But for the samples that do not agree with plurality, the algorithm tries to decide what the reason for the disagreement is: is it due to a deletion, an insertion, or a substitution? The classification of mismatches is done by looking at the context around the symbol under consideration in the noisy read. Once this is estimated, the pointers are then moved to the right accordingly.

In the third stage, the decoder unwinds the no-homopolymer representation to obtain matrices M corresponding to different blocks. In each recovered matrix some columns may be missing (erasures), and others may contain errors. In stage four, we decode the outer Reed–Solomon (RS) code to correct errors and erasures in rows of matrices M′ and invert randomization. Successful decoding is possible if for each row of each matrix M′ the 'used error resilience' ratio $\frac{2*(\#\text{errors})+(\#\text{erasures})}{\#\text{redundant RS coordinates}}$ is at most 1.

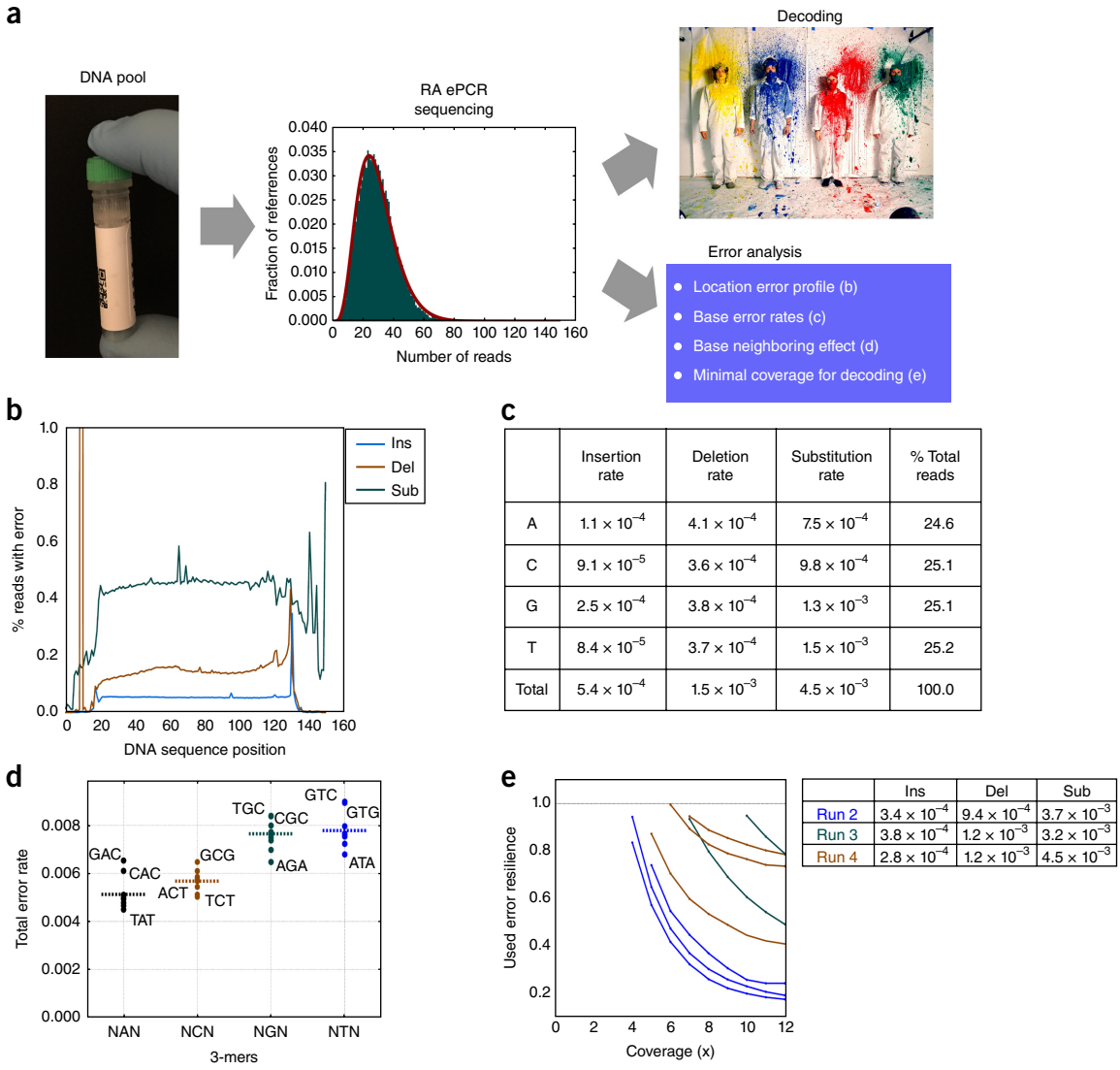
Error analysis and decoding from Illumina sequencing

We received nine synthesized DNA pools periodically over several months. In each case, we immediately individually amplified every file in the pool using random-access emulsion PCR, attached Illumina sequencing primers and adapters, and then sequenced the files for a total of ten sequencing runs. We have aggregated about 723 million reads of more than 13 million distinct synthetic DNA sequences. The mean coverage (i.e., number of reads for a given DNA sequence) across the data set was 53.8 reads with a s.d. of 48.7 reads. We observed considerable variance across files, ranging from a mean of 6.7 reads with s.d. of 3.4 reads, to a mean of 298.6 reads with s.d. of 139.6 reads. For most files, the empirical coverage distribution was reasonably well-approximated by a gamma distribution with matching mean and variance (**Fig. 3a**, center).

The sequencing information serves two purposes: (1) error analysis of processes related to DNA manipulation, including synthesis, random access, and sequencing, when used in conjunction with knowledge of the encoded DNA sequences; (2) and decoding of data stored in DNA and analysis of code resilience, that is, its ability to recover the information under the observed error regime. The decoding process uses only information that would be available at read time in a real storage scenario, that is, no knowledge of the encoded DNA sequences, other than the information received from sequencing, is used.

The error analysis in **Figure 3b** (**Supplementary Note 8**) reveals an average error rate per position of 0.6%. Substitutions were the most prominent type (0.4%), twice as likely as deletions (0.2%) and ten times as common as insertions (0.04%). In some files, specific positions showed higher error rates owing to systematic errors in the reading or writing processes. Also, primer target regions (first and last 20 positions) suffered from fewer errors because of the nature of PCR, which favors amplification of perfect primers and primer target regions. A clear exception is the spike at position 9, which was caused by a single primer sequence with an error at that position. However, in all cases, error rates in the primer region were low enough to associate most reads coming from sequencing to the sequenced files via sequence alignment. Analysis of the non-primer region is shown in **Figure 3c**,**d**. **Figure 3c** shows the percent breakdown of errors per base type, highlighting that insertion and substitution errors were biased toward certain base types. **Figure 3d** also shows variation of error rates across differing neighboring base types.

Next, we proceeded to decode the data. In practice, current preparation and sequencing technologies yield some unusable reads owing to their length being outside the acceptable range or too low-confidence in base calling (6.5% on average in our experiments). We expect this number to improve as sequencing technology and wet lab protocols mature. We randomly subsampled the usable sequencing reads for each individual file, gradually increasing the number of reads supplied to the decoder. We were able to recover all 200 MB of data (zero-byte difference when compared to the original digital data) stored in the DNA with median coverage of only 5 reads per DNA sequence, with different files ranging from 4 to 14 reads per DNA sequence. If we include unusable reads in the calculation, the median goes up to 6.2 reads per DNA sequence. This is half as much as the minimum coverage ever reported in decoding digital data from DNA (**Fig. 1d**). The impact is lower cost because decoding from lower coverages allows for a larger number of different DNA sequences to be read with the same sequencing kit. To understand the effect of coverage on our ability to decode files with no bit errors, we supplied the decoder with increasing coverage of reads, and measured the 'used error resilience' for several of our files (**Fig. 3e**). As expected, the ratio decreased with



	Insertion rate	Deletion rate	Substitution rate	% Total reads
A	1.1×10^{-4}	4.1×10^{-4}	7.5×10^{-4}	24.6
C	9.1×10^{-5}	3.6×10^{-4}	9.8×10^{-4}	25.1
G	2.5×10^{-4}	3.8×10^{-4}	1.3×10^{-3}	25.1
T	8.4×10^{-5}	3.7×10^{-4}	1.5×10^{-3}	25.2
Total	5.4×10^{-4}	1.5×10^{-3}	4.5×10^{-3}	100.0

	Ins	Del	Sub
Run 2	3.4×10^{-4}	9.4×10^{-4}	3.7×10^{-3}
Run 3	3.8×10^{-4}	1.2×10^{-3}	3.2×10^{-3}
Run 4	2.8×10^{-4}	1.2×10^{-3}	4.5×10^{-3}

Figure 3 Experimental error analysis and decoding, sequencing using Illumina's NextSeq. (**a**) Overview of the experimental analysis workflow. A file of interest is randomly accessed (RA) via ePCR amplification from a stored pool and sequenced, resulting in a distribution of sequencing reads from which the file is decoded to recover its original bit content, and a file-specific error profile is generated. Decoding image is a snapshot from the high-definition video by Ok Go that was encoded in the DNA. (**b**) Per-position average read error profile averaged over the first 150 positions of all 13 million strands and their corresponding reads. There is a spike to almost 15% at position 9 caused by an error in the primer region of a single file. (**c**) Error rates and number of reads for different nucleotide types in the payload region. Almost half of the insertions are associated with type G and about a third of the substitutions are associated with type T. Deletions are evenly distributed. (**d**) Error rates depend not only on nucleotide type but also on the type of neighboring nucleotides. Each dot corresponds to a 3-mer in the payload region colored according to the central base: black for A, red for C, green for G, and blue for T. The horizontal bars represent the weighted mean of the dots in that particular column, as not all 3-mer appear the same number of times. Again, types G and T are associated with higher error rates. (**e**) Estimating the minimal coverage required for decoding. Each curve corresponds to a different file, each color corresponds to a different sequencing run, and numbers in the legend correspond to the average insertion, deletion, and substitution errors for the corresponding sequencing run. Redundant information is more scarce at lower coverages, resulting in higher 'used error resilience'.

coverage because the total number of errors and erasures decreases with extra read information.

DNA assembly for nanopore sequencing

To further stress-test our decoding algorithm, we sequenced two files (32 KB and 1.3 KB) using the Oxford Nanopore Technologies (ONT) MinION sequencer (**Fig. 4**). The compactness and potential for scalability makes nanopore-based sequencing an intriguing option for integration in future stand-alone DNA data storage systems. A key advantage is a very long read length of potentially thousands of nucleotides; however, with current technology, only a limited number of reads can be obtained from a single sequencing flow cell. To best utilize these characteristics, we developed a protocol to concatenate multiple oligonucleotides into longer reads (**Fig. 4a**,**b**). Using this approach, we successfully recovered a 32-KB file sequenced with nanopore technology at a coverage of 36× and a 1.3-KB file at a coverage of 80× despite a high coordinate error rate of ~12%, computed using exhaustive minimum edit distance. We observed that reads of incorrect length constituted over 88% of all reads, and ignoring those reads makes recovering the files impossible even at maximal available

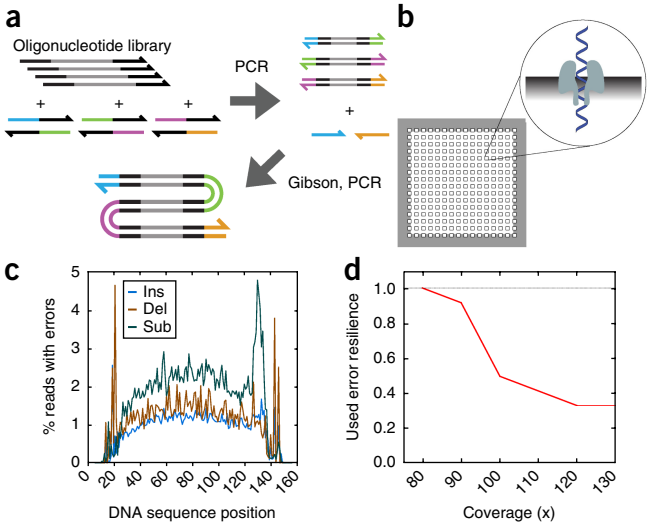


Figure 4 Sequencing using Oxford Nanopore Technologies' MinION. Overview of DNA data storage workflow using Oxford Nanopore sequencing. (**a**) A file of interest is amplified using PCR with primers containing complementary overhang sequences. Subsequently, amplification products are mixed in a Gibson assembly reaction and amplified using primers corresponding to the unique overhangs present at the 5′ and 3′ of the Gibson assembly product. (**b**) Amplicon consisting of concatenated oligonucleotides is sequenced using the MinION and thousands of reads are generated. (**c**) Per-position average read error profile averaged over all 88 strands of a 1.3-KB file and their corresponding 2D reads. Error rates are higher than in Illumina-sequenced reads. (**d**) Estimating the minimal coverage required for decoding. Higher error rates are offset by higher coverage, making decoding the original stored data possible.

coverages of 74× and 147×, respectively, indicating the importance of using as many reads as possible (**Fig. 4d**), a unique feature of our proposed decoder. Results above bode well for building an integrated, scalable DNA data storage system that is tolerant of the high error rates that could accumulate over millennia.

DISCUSSION

Given the current trends in data production and the rapid pace of progress of DNA manipulation technologies, DNA data storage has the potential to complement or eventually replace tape, the densest commercially available storage medium for archival storage.

The global demand for synthetic DNA in 2015 was estimated to be 4.8 billion bases of single-stranded oligos and ~1 billion bases of longer double-stranded oligos, or just under 6 Gigabases in total (http://www.synthesis.cc/synthesis/2016/03/on_dna_and_transistors). To provide a sense of scale, the size of the largest known eukaryotic genome is about 149 Gigabases[14]. The first practical 'DNA drive' should have a throughput of at least a few kilobytes per second. At the coding density demonstrated here, this is a few kilobases per second, or the equivalent of the entire synthetic DNA industry annual production in just 2 weeks. Clearly, synthetic DNA production will have to increase to meet this goal. We contend this is attainable because the synthetic DNA needed for data storage can be significantly more error prone than DNA used by life sciences, and very few copies per sequence are required. This is due to error-correcting algorithms such as the one described in this paper.

Even at kilobyte-per-second throughput, a DNA drive can be interesting because of the long-term durability and relevance DNA can offer to the preservation of high value-per-bit data. However, large-scale, deployed storage technologies today offer throughputs of hundreds of megabits per second, which will be more challenging to match. At this point, even DNA sequencing technologies, which are currently capable of reading megabases per second, will require improvement. The cost per bit offered by current storage devices is also much lower than what is possible with DNA today. Luckily, both DNA synthesis and sequencing technologies use array-based designs, which are readily replicable and amenable to scaling. This scaling increases the number of DNA sequences that can be read or written at a time, simultaneously increasing throughput and decreasing costs. Additional cost reductions can be obtained by optimizing fluidic delivery and exploiting large-scale efficiencies in the chemistry of reagents. We expect to see substantial activity in these areas in the upcoming years.

This paper describes large-scale random access, low redundancy, and robust encoding and decoding of information stored in DNA, as well as a notable increase in the volume of data stored (200 MB, the largest synthetic DNA pool available to date). To encourage more work in this area, we will be making samples available to select groups interested in DNA data storage.

METHODS

Methods, including statements of data availability and any associated accession codes and references, are available in the online version of the paper.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

We would like to thank B. Peck, P. Finn, S. Chen, A. Stewart, B. Arias, and E. Leproust from Twist Bioscience for supplying the DNA, suggesting protocol refinements, and offering input to our data analysis. We also thank J. Bornholt, K. D'Silva, and A. Levskaya for their help in the early stages of this project, and Y. Chou for her help in preparing samples for distribution. This work was supported in part by a sponsored research agreement by Microsoft, NSF award CCF-1409831 to L.C. and G.S. and by NSF award CCF-1317653 to G.S.

AUTHOR CONTRIBUTIONS

L.O., Y.J.C., and R.L. designed protocols and performed experiments. S.Y., S.D.A., K.M., M.Z.R., C.R., and P.G. designed and implemented the encoding and decoding pipeline. S.D.A., M.Z.R., G.K., Ke.S., and C.N.T., collected and analyzed data. B.N., C.N.T., S.N., G.G., H.Y.P., R.C., and J.M. assisted in designing and evaluating experiments. D.C., G.S., L.C., and Ka.S. designed experiments, analyzed data and supervised the work.

COMPETING FINANCIAL INTERESTS

The authors declare competing financial interests: details are available in the online version of the paper.

1. Neiman, M.S. On the molecular memory systems and the directed mutations. *Radiotekhnika* **6**, 1–8 (1965).
2. Cox, J.P.L. Long-term data storage in DNA. *Trends Biotechnol.* **19**, 247–250 (2001).
3. Church, G.M., Gao, Y. & Kosuri, S. Next-generation digital information storage in DNA. *Science* **337**, 1628 (2012).
4. Goldman, N. *et al.* Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature* **494**, 77–80 (2013).
5. Grass, R.N., Heckel, R., Puddu, M., Paunescu, D. & Stark, W.J. Robust chemical preservation of digital information on DNA in silica with error-correcting codes. *Angew. Chem. Int. Ed.* **54**, 2552–2555 (2015).
6. Blawat, M. *et al.* Forward error correction for DNA data storage. *Procedia Comput. Sci.* **80**, 1011–1022 (2016).
7. Erlich, Y. & Zielinski, D. DNA Fountain enables a robust and efficient storage architecture. *Science* **355**, 950–954 (2017).

8. Yazdi, S.M.H.T., Yuan, Y., Ma, J., Zhao, H. & Milenkovic, O. A rewritable, random-access DNA-based storage system. *Sci. Rep.* **5**, 14138 (2015).
9. Bornholt, J. *et al.* in *Proc. Int. Conf. ASPLOS*. 637–649 (ACM, 2016).
10. Yazdi, S.M.H.T., Gabrys, R. & Milenkovic, O. Portable and error-free DNA-based data storage. *Sci. Rep.* **7**, 5011 (2017).
11. Kosuri, S. & Church, G.M. Large-scale de novo DNA synthesis: technologies and applications. *Nat. Methods* **11**, 499–507 (2014).
12. Xu, Q., Schlabach, M.R., Hannon, G.J. & Elledge, S.J. Design of 240,000 orthogonal 25mer DNA barcode probes. *Proc. Natl. Acad. Sci. USA* **106**, 2289–2294 (2009).
13. Batu, T., Kannan, S., Khanna, S. & McGregor, A. Reconstructing strings from random traces. *Proc. Fifteenth Annu. ACM-SIAM SODA'04*. **2004**, 910–918 (2004).
14. Pellicer, J., Fay, M.F. & Leitch, I.J. The largest eukaryotic genome of them all? *Bot. J. Linn. Soc.* **164**, 10–15 (2010).

ONLINE METHODS

Selectively amplifying DNA. Whenever we received a pool of synthetic DNA, we rehydrated the pool in 1× TE buffer and used the following protocol to amplify each file individually (please see **Supplementary Fig. 5** for more workflow information):

Mix 10 ng of ssDNA pool (1 µL) with 1 µL of 100 µM of the forward primer (with a 25 nucleotide random overhang, see **Supplementary Note 9** for rationale and **Supplementary Fig. 5** for a simple schematic) and 1 µL of 100 µM of the reverse primer (with no overhang), 25 µL of 2× Kapa HiFi enzyme mix, 20 µL of molecular grade water, and 2 µL of 1.25 mg/mL acetylated bovine serum albumin. All primers were ordered from Integrated DNA Technologies. This 50 µL mix was then mixed with the 300 µL oil surfactant mixture detailed in the EURx ePCR kit. The resulting mixture was then attached to a benchtop vortexer in a refrigerator and vortexed for 5 min at the highest setting.

After vortexing, the now milky-appearing product was split evenly into three PCR tubes and placed in a thermocycler with the following protocol: (1) 95 °C for 3 min, (2) 98 °C for 20 s, (3) 62 °C for 20 s, (4) 72 °C for 15 s, (5) go to step 2 a varying number of times depending on the proportion of the pool being amplified, (6) 72 °C for 30 s. The reaction was then purified according to the instructions in the EURx ePCR kit. Total yield typically ranged between 30 ng and 1 µg because ePCR yield is directly proportional to the size of the file. The reverse micelles that make up the emulsion should all theoretically have the same amount of primer and one strand of DNA, so the larger the file, the greater the proportion of micelles have targeted strands. Recall that regardless of file size, 10 ng of the pool was used (see **Supplementary Table 3** for the percent of the amplified pool each file comprised). This resulted in ~80 k copies of each strand present at the start of each ePCR reaction.

When necessary, qPCR was performed to determine the ideal number of cycles to amplify a file according to the following recipe: mix 1 ng of ssDNA pool (1 µL) with 0.5 µL of 10 µM of the forward primer (with no overhang) and 0.5 µL of 10 µM of the reverse primer (with no overhang), 10 µL of 2× Kapa HiFi enzyme mix, 7 µL of molecular grade water, and 1 µL of 20× Eva Green. The thermocycling protocol was: (1) 95 °C for 3 min, (2) 98 °C for 20 s, (3) 62 °C for 20 s, (4) 72 °C for 15 s, then repeat steps 2–4 as needed.

After amplification with ePCR, the length of the dsDNA products was confirmed with a Qiaxcel fragment analyzer, with sample concentration measured by Qubit 3.0 fluorometer.

Ligation of amplified DNA files for sequencing. After ePCR, amplified products were ligated to the Illumina sequencing adapters with a modified version of Illumina TruSeq Nano ligation protocol and TruSeq ChIP Sample Preparation protocol. Briefly, samples were first converted to blunt ends with the ERP2 reagent and directions provided in the Illumina TruSeq Nano kit, then purified with AMPure XP beads according to the TruSeq ChIP protocol. An 'A' nucleotide was added to the 3′ends of the blunt DNA fragments with the TruSeq Nano's A-tailing ligase and protocol, followed by ligation to the Illumina sequencing adapters with the TruSeq Nano reagents and protocol. We then cleaned the samples with Illumina sample purification beads and enriched the sample using PCR to yield enough product for sequencing. The length of enriched products was confirmed using a Qiaxcel bioanalyzer.

Sample preparation for sequencing. When multiple separate samples were prepared for sequencing, these samples were mixed proportionally (e.g., a 10,000 oligonucleotide file to be sequenced with a 500,000 file would comprise 1.96% of the DNA material in this mix). The mixed sample was then prepared for sequencing by following the NextSeq System Denature and Dilute Libraries Guide. The sequencing sample was loaded into the sequencer at 1.3 pM, with a 10 to >20% PhiX spike-in as a control (PhiX is a reliable, adaptor-ligated, well-characterized genomic DNA sample provided by Illumina).

Sequencing with Oxford Nanopore Technologies MinION. First, we used PCR to amplify an oligonucleotide library with primers containing orthogonal overhang sequences. Then, we combined the amplified products into one Gibson assembly reaction where each overhang allowed for multiple library members to be concatenated. Finally, we used PCR to amplify the resulting concatenated product with primers that hybridize to each respective end of the assembly product. Using this approach, we generated a 3-fragment and a 6-fragment assembly for a 1.3-KB and a 32-KB file, respectively.

To amplify the original file and add the overhangs, a 100-fold diluted sample of ssDNA library was amplified using a KAPA SYBR FAST qPCR kit with the following thermal profile: (1) 95 °C for 3 min, (2) 98 °C for 20 s, (3) 69 °C for 20 s, (4) 72 °C for 20 s. The total number of cycles of steps 2–4 was determined by monitoring the fluorescence of the qPCR instrument as the amplification reached the plateau phase. Each amplification reaction was performed separately with primers containing distinct overhang regions necessary for a subsequent Gibson assembly reaction. Overhang sequences were designed using the NUPACK[15] design module to avoid secondary structure formation. After amplification, each reaction was purified using Agencourt AMPure XP. Subsequently, amplification products mixed at equal molar ratio were added to NEB Gibson assembly master mix (1:1 volume ratio) and incubated at 50 °C for 30 min.

Upon AMPure XP clean-up, the ligated product was amplified using the same qPCR protocol described above. Amplification was performed with primers corresponding to unique overhang sequences present at the 5′ and 3′ ends of the DNA. After amplification, a DNA band corresponding to the expected size was gel-extracted from a 2% agarose gel and quantified by a Qubit 3.0 fluorometer. The final product had the expected size corresponding to the number of fragments and overhangs used in the assembly.

Sequencing sample preparation of the 1.3-KB file was performed according to the Oxford Nanopore Technologies (ONT) Amplicon (R9) protocol for 2D sequencing. Metrichor sequencing metrics indicated 37,478 reads with workflow successful exit status out of 130,573 total reads. Sequencing sample preparation of the 32-KB file was performed according to the Oxford Nanopore Technologies (ONT) Amplicon (R9.4) protocol for $1D^2$ sequencing. ONT Albacore basecalling software yielded 57,012 $1D^2$ reads. In both cases, these reads were then successfully decoded into the original digital file.

Code availability. Supplementary Note 7 provides details on how to reproduce the clustering results using off-the-shelf software, as the custom code is proprietary.

Life Sciences Reporting Summary. Further information on experimental design is available in the **Life Sciences Reporting Summary**.

Data availability. We have made available two files that enable the reproduction of the key parts of our decoding pipeline. MANIFEST describes the content and its use. The first file (id20.fastq.gz) is a FASTQ file containing reads associated with a single file and the second (id20.refs.txt.gz) contains a list of references corresponding to these reads. The data can be used freely and are available via http://misl.cs.washington.edu/data and https://github.com/uwmisl/data-nbt17.

15. Zadeh, J.N. *et al.* NUPACK: analysis and design of nucleic acid systems. *J. Comput. Chem.* **32**, 170–173 (2011).

Erratum: Random access in large-scale DNA data storage

Lee Organick, Siena Dumas Ang, Yuan-Jyue Chen, Randolph Lopez, Sergey Yekhanin, Konstantin Makarychev, Miklos Z Racz, Govinda Kamath, Parikshit Gopalan, Bichlien Nguyen, Christopher N Takahashi, Sharon Newman, Hsing-Yeh Parker, Cyrus Rashtchian, Kendall Stewart, Gagan Gupta, Robert Carlson, John Mulligan, Douglas Carmean, Georg Seelig, Luis Ceze & Karin Strauss
***Nat. Biotechnol.* 36, 242–248 (2018); published online 19 February 2018; corrected after print 6 March 2018**

In the version of this article initially published, the references in the reference list were in the wrong order; the references have been renumbered as follows: 3 as 2; 5 as 3; 6 as 8; 7 as 9; 8 as 11; 9 as 6; 10 as 12; 11 as 5; 12 as 13; 13 as 7; 16 as 10; and no. 2, "Hoch, J.A. & Losick, R. Panspermia, spores and the *Bacillus subtilis* genome. *Nature* **390**, 237–238 (1997)," has been deleted. In addition, on p.242, end of paragraph 2, the citation in "experiments[7]" has been deleted. The errors have been corrected in the HTML and PDF versions of the article.