REPORT

DNA STORAGE

DNA Fountain enables a robust and efficient storage architecture

Yaniv Erlich[1,2,3]* and Dina Zielinski[1]

DNA is an attractive medium to store digital information. Here we report a storage strategy, called DNA Fountain, that is highly robust and approaches the information capacity per nucleotide. Using our approach, we stored a full computer operating system, movie, and other files with a total of 2.14×10^6 bytes in DNA oligonucleotides and perfectly retrieved the information from a sequencing coverage equivalent to a single tile of Illumina sequencing. We also tested a process that can allow 2.18×10^{15} retrievals using the original DNA sample and were able to perfectly decode the data. Finally, we explored the limit of our architecture in terms of bytes per molecule and obtained a perfect retrieval from a density of 215 petabytes per gram of DNA, orders of magnitude higher than previous reports.

Humanity is currently producing data at exponential rates, creating a demand for better storage devices. DNA is an excellent medium for data storage, owing to its demonstrated information density of petabytes of data per gram, high durability, and evolutionarily optimized machinery to faithfully replicate this information (*1*, *2*). Recently, a series of proof-of-principle experiments has demonstrated the value of DNA as a storage medium (*3*–*9*).

To better understand its potential, we explored the Shannon information capacity (*10*, *11*) of DNA storage (*12*). This measure sets a tight upper bound on the amount of information that can be reliably stored in each nucleotide. In an ideal world, the information capacity of each nucleotide could reach 2 bits, as there are four possible options. However, DNA encoding faces several practical limitations. First, not all DNA sequences are created equal (*13*, *14*). Biochemical constraints dictate that DNA sequences with high GC content or long homopolymer runs (e.g., AAAAAA…) are undesirable, as they are difficult to synthesize and prone to sequencing errors. Second, oligonucleotide (hereafter "oligo") synthesis, polymerase chain reaction (PCR) amplification, and decay of DNA during storage can induce uneven representation of the oligos (*7*, *15*). This might result in dropout of a small fraction of oligos that will not be available for decoding. In addition to biochemical constraints, oligos are sequenced in a pool and necessitate indexing to infer their order, which further limits the number of available nucleotides for encoding information. Quantitative analysis shows that the biochemical constraints reduce the coding potential of each nucleotide to 1.98 bits. After combining the expected dropout rates and barcoding demand, the overall Shannon information capacity of a DNA storage device is ~1.83 bits per nucleotide for a range of practical architectures (*12*) (figs. S1 to S5 and tables S1 to S3).

Previous studies of DNA storage realized about half of the Shannon information capacity of DNA molecules. In addition, most of the previous studies reported challenges in perfect information retrieval (Table 1). For example, two previous studies attempted to address oligo dropout by dividing the original file into overlapping segments so that each input bit is represented by multiple oligos (*4*, *6*). However, this repetitive coding procedure generates a loss of information content and is poorly scalable (fig. S6). In both cases, these studies reported small gaps in the retrieved information (*4*, *6*). Other studies explored the use of Reed-Solomon (RS) code on small blocks of the input data to recover dropouts (*5*, *7*). Although these studies were able to perfectly retrieve the data, they were still far from realizing the capacity. Moreover, testing this strategy on a large file size highlighted difficulties in decoding the data due to local correlations and large variations in the dropout rates within each protected block (*7*), which is a known issue of blocked RS codes (*16*, *17*). Only after employing a complex multistep procedure and high sequencing coverage was the study able

[1]New York Genome Center, New York, NY 10013, USA. [2]Department of Computer Science, Fu Foundation School of Engineering, Columbia University, New York, NY 10027, USA. [3]Center for Computational Biology and Bioinformatics (C2B2), Department of Systems Biology, Columbia University, New York, NY 10027, USA.
*Corresponding author. Email: yaniv@cs.columbia.edu

Table 1. Comparison of DNA storage coding schemes and experimental results. For consistency, the table describes only schemes that were empirically tested with pooled oligo synthesis and high-throughput sequencing data. The schemes are presented chronologically on the basis of publication date. Coding potential is the maximal information content of each nucleotide before indexing or error correcting. Redundancy denotes the excess of synthesized oligos to provide robustness to dropouts. Error correction/detection denotes the presence of error-correction or -detection code to handle synthesis and sequencing errors (RS, Reed-Solomon codes). Full recovery indicates whether all information was recovered without any error. Net information density indicates the input information in bits divided by the number of synthesized DNA nucleotides (excluding adapter annealing sites). Realized capacity is the ratio between the net information density and the Shannon capacity of the channel. Physical density is the actual ratio of the number of bytes encoded and the minimal weight of the DNA library used to retrieve the information. This information was not available for studies by Bornholt *et al*. (*6*) and Blawat *et al*. (*7*), as indicated by the dashes. See (*12*) for more information.

Parameter	Church *et al*. (*3*)	Goldman *et al*. (*4*)	Grass *et al*. (*5*)	Bornholt *et al*. (*6*)	Blawat *et al*. (*7*)	This work
Input data (Mbytes)	0.65	0.75	0.08	0.15	22	2.15
Coding potential (bits/nt)	1	1.58	1.78	1.58	1.6	1.98
Redundancy	1	4	1	1.5	1.13	1.07
Robustness to dropouts	No	Repetition	RS	Repetition	RS	Fountain
Error correction/detection	No	Yes	Yes	No	Yes	Yes
Full recovery	No	No	Yes	No	Yes	Yes
Net information density (bits/nt)	0.83	0.33	1.14	0.88	0.92	1.57
Realized capacity	45%	18%	62%	48%	50%	86%
Number of oligos	54,898	153,335	4,991	151,000	1,000,000	72,000
Physical density (Pbytes/g)	1.28	2.25	25	–	–	214

to rescue a sufficient number of oligos. Taken together, these results inspired us to seek a coding strategy that can better utilize the information capacity of DNA storage devices while showing higher data-retrieval reliability.

We devised a strategy for DNA storage, called DNA Fountain, that approaches the Shannon capacity while providing robustness against data corruption. Our strategy harnesses fountain codes (*18*, *19*), which have been developed for reliable and effective unicasting of information over channels that are subject to dropouts, such as mobile TV (*20*). In our design, we carefully adapted the power of fountain codes to overcome both oligo dropouts and the biochemical constraints of DNA storage. Our encoder works in three steps (Fig. 1) (*12*): First, it preprocesses a binary file into a series of nonoverlapping segments of a certain length. Next, it iterates over two computational steps: Luby transform and screening. The Luby transform sets the basis for fountain codes. Basically, it packages data into any desired number of short messages, called droplets, by selecting a random subset of segments from the file using a special distribution (fig. S7) and adding them bitwise together under a binary field. The droplet contains two pieces of information: a data payload part that holds the result of the addition procedure and a short, fixed-length seed. This seed corresponds to the state of the random-number generator of the transform during the droplet creation and allows the decoder algorithm to infer the identities of the segments in the droplet. Theoretically, it is possible to reverse the Luby transform using a highly efficient algorithm by collecting any subset of droplets as long as the accumulated size of droplets is slightly bigger than the size of the original file. For DNA Fountain, our algorithm applies one round of the transform in each iteration to create a single droplet. Next, the algorithm moves to the droplet screening stage. This stage is not part of the original fountain code design and allows us to completely realize the coding potential of each nucleotide. In screening, the algorithm translates the binary droplet to a DNA sequence by converting {00,01,10,11} to {A,C,G,T}, respectively. Then, it screens the sequence for the desired biochemical properties of GC content and homopolymer runs. If the sequence passes the screen, it is considered valid and added to the oligo design file; otherwise, the algorithm simply trashes the droplet. Since the Luby transform can create any desired number of droplets, we keep iterating over the droplet creation and screening steps until a sufficient number of valid oligos are generated. In practice, we recommend 5 to 10% more oligos than input segments (*12*). Searching for valid oligos scales well with the size of the input file and is economical for various oligo lengths within and beyond current synthesis limits (*12*) (table S4).

We used DNA Fountain to encode a single compressed file of 2,146,816 bytes in a DNA oligo pool. The input data were in the form of a tarball that packaged several files, including a complete graphical operating system of 1.4 Mbytes, a movie, and other files (*12*) (Fig. 2A and fig. S8). We split the input tarball into 67,088 segments of 32 bytes and iterated over the steps of DNA Fountain to create valid oligos. Each droplet was 38 bytes (304 bits): 4 bytes of the random-number generator seed, 32 bytes for the data payload, and 2 bytes for an RS error-correcting code, to reject erroneous oligos in low-coverage conditions. With this seed length, our strategy supports encoding files of up to 500 Mbytes (*12*). The DNA oligos had a length of 304/2 = 152 nucleotides (nt) and were screened for homopolymer runs of ≤3 nt and GC content of 45 to 55%. We instructed DNA Fountain to generate 72,000 oligos, yielding a redundancy of 72,000/67,088 – 1 = 7%. We selected this number of oligos due to the price structure offered by the manufacturer, allowing us to maximize the number of oligos per dollar. Finally, we added upstream and downstream annealing sites for Illumina adapters, making our final oligos 200 nt long (Fig. 2B and fig. S9). Encoding took 2.5 min on a single central processing unit (CPU) of a standard laptop. We achieved an information density of 1.57 bits/nt, only 14% from the Shannon capacity of DNA storage and 60% more than previous studies with a similar scale of data (Table 1).

Sequencing and decoding the oligo pool fully recovered the entire input file with zero errors

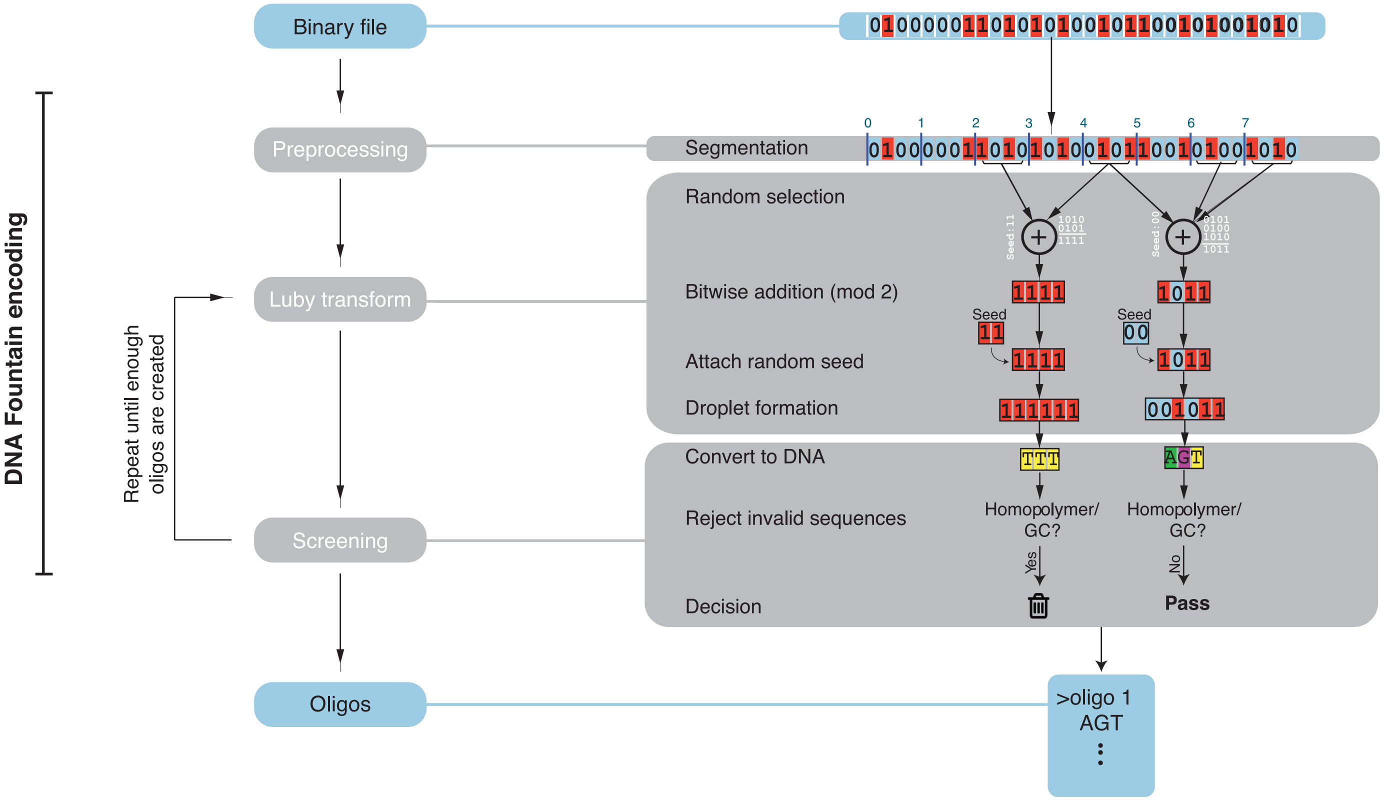


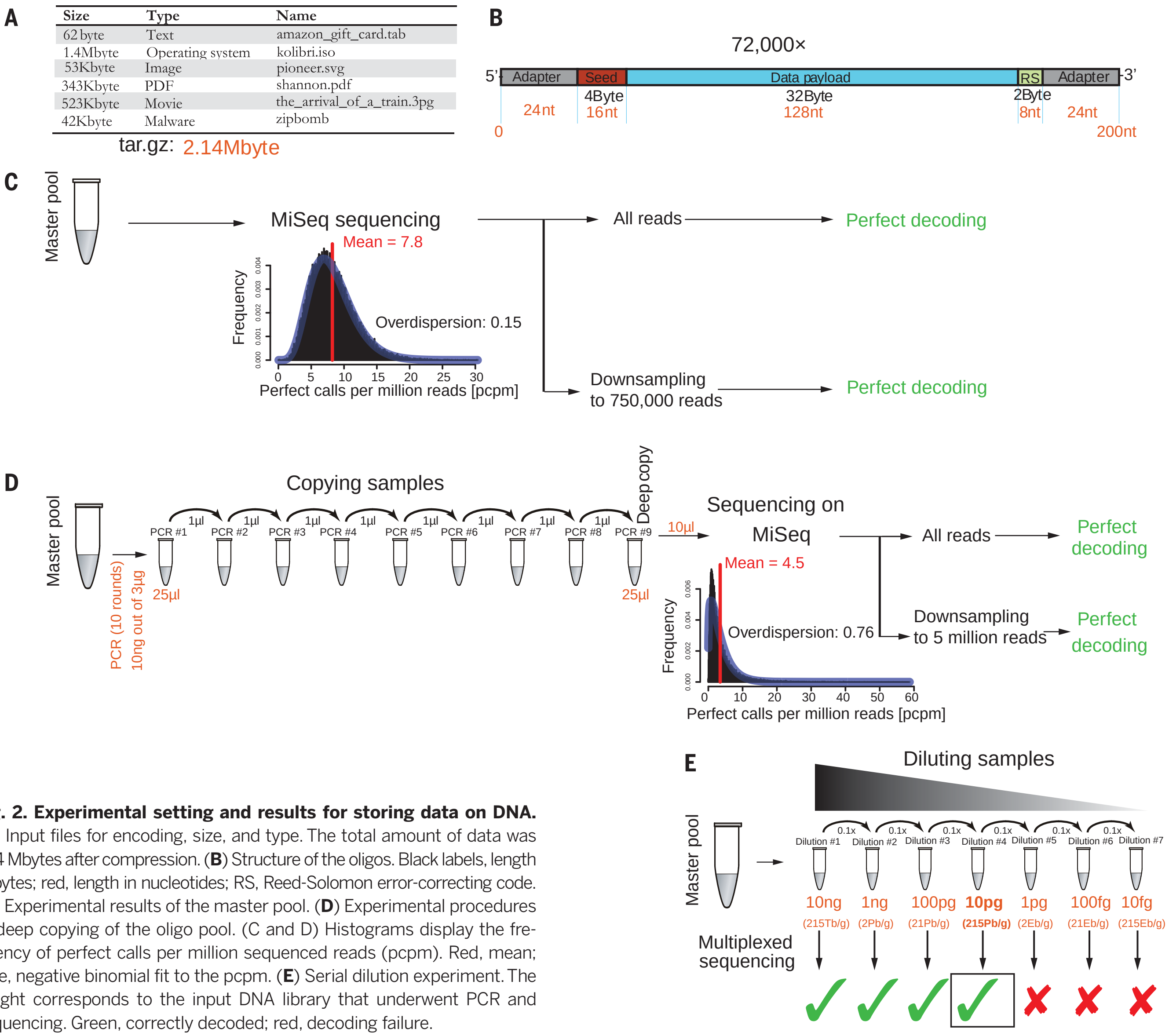
Fig. 1. DNA Fountain encoding. (**Left**) Three main algorithmic steps. (**Right**) Example with a small file of 32 bits. For simplicity, we partitioned the file into eight segments of 4 bits each. The seeds are represented as 2-bit numbers and are presented for display purposes only. See (*12*) for the full details of each algorithmic step.

(Fig. 2C). To retrieve the information, we PCR-amplified the oligo pool and sequenced the DNA library on one Illumina MiSeq flow cell with 150 paired-end cycles, which yielded 32 million reads. We employed a preprocessing strategy that prioritizes reads that are more likely to represent high-quality oligos (*12*). Because not all oligos are required for the decoding due to redundancy, this procedure reduces the exposure to erroneous oligos. The decoder scans the reads, recovers the binary droplets, rejects droplets with errors based on the RS code, and employs a message-passing algorithm to reverse the Luby transform and obtain the original data (*12*).

In practice, decoding took ~9 min with a Python script on a single CPU of a standard laptop (movie S1). The decoder recovered the information with 100% accuracy after observing only 69,870 oligos out of the 72,000 in our library (fig. S10). To further test the robustness of our strategy, we down-sampled the raw Illumina data to 750,000 reads, equivalent to one tile of an Illumina MiSeq flow cell. This procedure resulted in 1.3% oligo dropout from the library. Despite these limitations, the decoder was able to perfectly recover the original 2.1 Mbytes in 20 of 20 random down-sampling experiments. These results indicate that beyond its high information density, DNA Fountain also reduces the amount of sequencing required for data retrieval, which is beneficial when storing large-scale information.

DNA Fountain can also perfectly recover the file after creating a deep copy of the sample. One of the caveats of DNA storage is that each retrieval of information consumes an aliquot of the material. Copying the oligo library with PCR is possible, but this procedure introduces noise and induces oligo dropout. To further test the robustness of our strategy, we created a deep copy of the file by propagating the sample through nine serial PCR amplifications (Fig. 2D). The first PCR reaction used 10 ng of material out of the 3-μg master pool. Each subsequent PCR reaction consumed 1 μl of the previous PCR reaction and employed 10 cycles in each 25-μl reaction. We sequenced the final library using one run on the Illumina MiSeq.

Overall, this recursive PCR reflects one full arm of an exponential process that theoretically could generate 300 × 25^9 × 2 = 2.28 quadrillion copies of the file by repeating the same procedure with each aliquot (fig. S11). As expected, the quality of the deep copy was substantially worse than the initial experiment with the master pool. The average coverage per oligo dropped from an average of 7.8 perfect calls for each oligo per million reads (pcpm) to 4.5 pcpm in the deep copy. In addition, the deep copy showed much higher skewed representation with a negative binomial overdispersion parameter (1/size) of 0.76 compared to 0.15 in the master pool. Despite the lower quality, the DNA Fountain decoder



Size	Type	Name
62 byte	Text	amazon_gift_card.tab
1.4Mbyte	Operating system	kolibri.iso
53Kbyte	Image	pioneer.svg
343Kbyte	PDF	shannon.pdf
523Kbyte	Movie	the_arrival_of_a_train.3pg
42Kbyte	Malware	zipbomb

tar.gz: 2.14Mbyte

Fig. 2. Experimental setting and results for storing data on DNA. (**A**) Input files for encoding, size, and type. The total amount of data was 2.14 Mbytes after compression. (**B**) Structure of the oligos. Black labels, length in bytes; red, length in nucleotides; RS, Reed-Solomon error-correcting code. (**C**) Experimental results of the master pool. (**D**) Experimental procedures of deep copying of the oligo pool. (C and D) Histograms display the frequency of perfect calls per million sequenced reads (pcpm). Red, mean; blue, negative binomial fit to the pcpm. (**E**) Serial dilution experiment. The weight corresponds to the input DNA library that underwent PCR and sequencing. Green, correctly decoded; red, decoding failure.

was able to fully recover the file without a single error with the full sequencing data. After down-sampling the sequencing data to five million reads, resulting in an approximate dropout rate of 1.0%, we were able to perfectly recover the file in 10 of 10 trials. These results suggest that with DNA Fountain, DNA storage can be copied virtually an unlimited number of times while preserving the data integrity of the sample.

Next, we explored the maximal achievable physical density using DNA Fountain. The pioneering study by Church *et al.* predicted that DNA storage could theoretically achieve an information density of 680 Pbytes (P: peta-; 10^{15}) per gram of DNA, assuming the storage of 100 molecules per oligo sequence (*3*). However, previous DNA storage experiments have never tested the maximal density of their storage scheme. To find the maximal physical density, we sequentially diluted our library by seven orders of magnitude from 10 ng to 10 fg of DNA (Fig. 2E). Under a perfect synthesis process (no synthesis errors and/or fragmented DNA molecules), the first dilution (10 ng) corresponds to ~10^6 copies per oligo and a density of ~200 Tbytes/g, whereas the last dilution corresponds to ~1 copy per oligo and ~200 Ebytes/g (E: exa; 10^{18}). We PCR-amplified all libraries using an identical strategy to keep all conditions uniform and sequenced the libraries using two multiplexed Illumina rapid runs, which yielded a similar number of reads and quality metrics (*12*).

We were able to perfectly retrieve the information from a physical density of 215 Pbytes/g. This density is over two orders of magnitude higher than previous reports with a comparable number of oligos and is close to the theoretical prediction by Church *et al.* At this density, the input weight of the library was 10 pg and each oligo was represented by ~1300 molecules, on average (table S5). We observed a 4% dropout rate, close to the limit of our decoder. For the lower input weights, the libraries had substantially more oligo dropouts, ranging from 62% for the 1-pg condition (~2 Ebytes/g) to 87% for the 10-fg condition (~200 Ebytes/g). A more aggressive error-correcting capability than DNA Fountain is unlikely to dramatically improve the physical density. We tested decoding of the low-weight libraries (<10 pg) under the unrealistic assumption of a decoder that can correct any number of indels and substitutions as long as a very short stretch (15 nt) of the read is still intact (*12*). Even this aggressive error correction failed to bring the dropout rates of the 1-pg library below 30%. Therefore, these results suggest that the current design approaches the maximal physical density permitted by the stochastic bottleneck induced by PCR amplification of highly complex libraries using a small number of DNA molecules.

To summarize, in this work, we reported an efficient and robust coding strategy that enables virtually unlimited data retrieval and high physical density while approaching the Shannon capacity of DNA storage closer than any previous design. We tested our framework with a relatively large file compared with those used in previous studies and were able to perfectly recover the data under various tests. Implementing our approach in concert with long-term preservation techniques, such as DNA embedding in silica beads (*5*), might require further fine-tuning of the redundancy levels. We expect that such fine-tuning can benefit from the high flexibility of the DNA Fountain framework, which allows determination of virtually any redundancy level without changing the software or affecting the decoding time.

Moving forward, practical implementation of DNA storage will require addressing the high cost of DNA synthesis, which was \$3500/Mbyte in this study. However, these costs reflect the relatively low throughput of manufacturing high-quality oligos for traditional synthetic biology applications that are sensitive to errors (*21*). This is not the case for DNA storage. As shown in our experiments, strong coding-theoretic tools can enable perfect decoding of the data from conditions that are well below the initial quality and quantity of the oligo manufacturer while still approaching the information capacity. Therefore, we envision that the cost issue of DNA storage could be addressed by two complementary approaches, the first of which is continuous improvements to the DNA synthesis chemistry, which have been estimated to exponentially reduce the cost by one to two orders of magnitude per decade (*4*). A second complementary approach to achieve cost reduction could rely on exploring quick-and-dirty oligo synthesis methods that consume less machine time and fewer reagents and, therefore, are more cost-effective. For example, previous results showed that a ~sixfold reduction in the time for the coupling reaction (60 to 10 s) in maskless array synthesis technology results in decreased reaction efficiency, from ~99 to 97.5% per cycle (*22*). The lower efficiency would exponentially decrease the number of valid oligos in the pool to a few percent ($0.975^{152\text{nt}} = 0.02$). But a robust and highly flexible coding strategy, such as DNA Fountain, might handle these imperfections with small tweaks to the redundancy levels or the error-detecting code. Taken together, we hypothesize that a substantial cost reduction could be achieved by finding the sweet spot that optimally combines the highest chemical throughput and the highest possible rate (bits per nucleotide) of the coding design. By exploiting these two complementary strategies for more cost-effective synthesis, DNA might become an economically viable solution for long-term, high-latency storage.

REFERENCES AND NOTES

1. M. R. Wallace, *Kybernetes* **7**, 265–268 (1978).
2. C. Bancroft, T. Bowler, B. Bloom, C. T. Clelland, *Science* **293**, 1763–1765 (2001).
3. G. M. Church, Y. Gao, S. Kosuri, *Science* **337**, 1628 (2012).
4. N. Goldman *et al.*, *Nature* **494**, 77–80 (2013).
5. R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, W. J. Stark, *Angew. Chem. Int. Ed.* **54**, 2552–2555 (2015).
6. J. Bornholt *et al.*, in *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems* (Association for Computing Machinery, 2016), pp. 637–649.
7. M. Blawat *et al.*, *Procedia Comput. Sci.* **80**, 1011–1022 (2016).
8. S. H. T. Yazdi *et al.*, *IEEE Trans. Mol. Biol. Multi-Scale Commun.* **1**, 230–248 (2015).
9. S. M. Yazdi, Y. Yuan, J. Ma, H. Zhao, O. Milenkovic, *Sci. Rep.* **5**, 14138 (2015).
10. C. E. Shannon, *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **5**, 3–55 (2001).
11. D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms* (Cambridge Univ. Press, 2002).
12. See supplementary materials.
13. J. J. Schwartz, C. Lee, J. Shendure, *Nat. Methods* **9**, 913–915 (2012).
14. M. G. Ross *et al.*, *Genome Biol.* **14**, R51 (2013).
15. Y. Erlich *et al.*, *Genome Res.* **19**, 1243–1253 (2009).
16. J. W. Byers, M. Luby, M. Mitzenmacher, *IEEE J. Sel. Areas Comm.* **20**, 1528–1540 (2002).
17. U. Demir, O. Aktas, in *2006 International Symposium on Computer Networks* [Institute of Electrical and Electronics Engineers (IEEE), 2006], pp. 264–269.
18. M. Luby, in *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science* (IEEE, 2002), pp. 271–280.
19. D. J. C. MacKay, *IEEE Proc. Commun.* **152**, 1062–1068 (2005).
20. T. Stockhammer, A. Shokrollahi, M. Watson, M. Luby, T. Gasiba, in *Handbook of Mobile Broadcasting: DVB-H, DMB, ISDB-T, AND MEDIAFLO* (CRC Press, 2008), pp. 239–280.
21. S. Kosuri, G. M. Church, *Nat. Methods* **11**, 499–507 (2014).
22. C. Agbavwe *et al.*, *J. Nanobiotechnology* **9**, 57 (2011).

ACKNOWLEDGMENTS

Y.E. holds a Career Award at the Scientific Interface from the Burroughs Wellcome Fund. This study was supported by a gift from A. Heafy and P. Heafy to the Erlich lab. We thank P. Smibert and M. Stoeckius for technical assistance with oligo design, N. Abe and S. Pescatore for sequencing operations, A. Gordon for creating the movie and the DNA Fountain website, and S. Zaaijer and N. Sanjana for comments and discussions. Code is available at https://github.com/TeamErlich/dna-fountain. Sequencing data are available at www.ebi.ac.uk/ena/data/view/PRJEB19305 and www.ebi.ac.uk/ena/data/view/PRJEB19307.

SUPPLEMENTARY MATERIALS

www.sciencemag.org/content/355/6328/950/suppl/DC1
Materials and Methods
Figs. S1 to S11
Tables S1 to S4
References (*23*–*38*)
Movie S1

12 September 2016; accepted 9 February 2017
10.1126/science.aaj2038