



DNA archival storage, a bottom up approach

Omer S. Sella
University of Cambridge
oss22@cam.ac.uk

Amir Apelbaum

Thomas Heinis
Imperial College London
t.heinis@imperial.ac.uk

Jasmine Quah
Imperial College London
jasmine.quah17@imperial.ac.uk

Andrew W. Moore
University of Cambridge
Andrew.Moore@cl.cam.ac.uk

ABSTRACT

DNA is an attractive medium for storage of digital data in the cloud, owing to a form factor several orders of magnitude smaller than any other. Key to storing binary data in synthetic DNA is the translation between the binary representation of digital data to the quaternary domain of DNA. This translation must adhere to constraints imposed by the synthesis and sequencing processes used to write and read respectively. A technological advance in either process changes the constraints and renders current encoding schemes obsolete. In this paper we present a recipe for taking constraints and producing an appropriate encoding scheme. Such a mechanism allows moving the encoding in lockstep with the technological advances in the underlying processes. We further show a method to understand trade-offs in constraints for a given overhead of bits needed to meet such constraints.

1 INTRODUCTION

Archival storage refers to storage of digital data for decades with low read request frequency. The dominating metrics for archival storage are \$ per gigabyte and durability. If the life expectancy of magnetic tape is 25 years, then in the span of 60 years data stored on tape will have been refreshed or migrated at least three times, that is, three reads and writes for maintenance regardless of any user interaction. Deoxyribonucleic acid (DNA) has proven as a highly durable molecule and so using it to store data has the potential to avoid data migration and refresh for much longer. With a theoretical amount of terabytes per inch cubed (i.e. density) that beats any current medium, it is a leading candidate to replace

hard disk drives (HDDs) and magnetic tape. A unique advantage of DNA is replication of the molecules using polymerase chain reaction (PCR) – a strictly biochemical process, also known as amplification. Using PCR, DNA could be replicated quickly without the need to read it and write it explicitly. The building blocks of synthetic DNA are four nucleotides, or bases, marked as A,C,T,G, and so "synthesis" refers to any (current or future) process that allows one to assemble DNA from these bases. Similarly "sequencing" refers to any mechanism that used to produce a sequence of bases from a DNA molecule. Ideally, there would be no restrictions over the sequential combinations of A,C,T,G, i.e.: all would be equally possible to synthesise and sequence but this is not the case. Constraints over the content of the sequence arise from either synthesis, sequencing or PCR. Some are defined in terms of G-C content, which is defined to be the ratio between the count of G and C instances in the sequence divided by the sequence length and include:

G-C limits Limits on G-C content from above and below.

G-C variation refers to variation in G-C content between adjacent DNA stretches. While some variance is expected, limits on it may be imposed.

Other common constraints are related to repetitions such as:

Homopolymers Subsequences that are made of repetitions of a single base. The maximal number of consecutive appearances allowed is called maximum run length. Homopolymers violate a more general type of subsequences to avoid, i.e.: **short tandem repeats (STRs)**. These are made from multiple adjacent repetitions of a short subsequence. Sequences that contain STRs are prone to PCR slippage, a fault in PCR that causes mutations such as insertions and deletions [23, 25]. Since PCR may be used in either synthesis or sequencing, constraints over small repetitions may arise from both. We list maximum run length independently since certain sequencing methods struggle specifically with discerning long runs of the same base, however, our treatment of them is the same.

In addition, any mapping of binary data to DNA that contains a form of reserved subsequences for addressing or other operations, e.g.: [4], needs to assure such reserved subsequences do not appear in the encoded data.



This work is licensed under a Creative Commons Attribution International 4.0 License.

HotStorage '21, July 27–28, 2021, Virtual, USA

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8550-3/21/07.

<https://doi.org/10.1145/3465332.3470880>

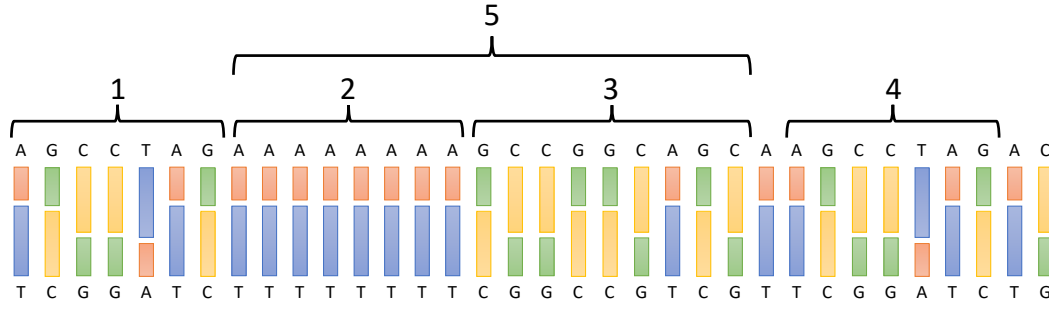


Figure 1: 1. The sequence AGCCTAG is a primer that acts as a "start" marker. 2. Multiple repetitions of A, G-C content below 25% 3. G-C content over 65% 4. Primer appears in the data. 5. Adjacent subsequences with high G-C content variation between them.

Figure 1 illustrates some of these constraints. Synthesis is open to the public through various providers, for example [24] and [1], and lay out specific G-C limits, G-C variation, and other possible limitations. Violation of these constraints may result in failures or errors in the synthesised DNA, incur higher cost or simply be rejected by the service provider. With the target users drawn from the life science disciplines, advancements in synthesis include the relaxation of such limitations, and at the same time performing a set of Quality Assurance (QA) tests to make sure the DNA produced matches the one ordered with a low probability of error. While this seems to align with the desire to store arbitrary digital data with arbitrarily low bit error rate, it actually keeps the cost of synthesis at several cents per base, too high for DNA archival storage to become a commodity. Sequencing technologies also target higher accuracy, sometimes through multiple sequencing attempts. Unlike the life science use-case, digital data allows us complete freedom in encoding the binary data into DNA sequences, avoiding synthesis difficulties and QA as much as possible in DNA and picking up whatever errors are left using error correcting codes. These are lots of moving parts when it comes to designing an archival storage system, and the encoding needs to change accordingly.

In this work we do not bring a single encoding from binary to DNA but rather the machinery to compile an encoding from a set of constraints. This machinery and the compiled encoding impose no restrictions on the data to be stored and at the same time:

- (1) Take known difficulties that stem from a given biochemical process and treat them as constraints to obey.
- (2) Avoid any reserved subsequence in the encoded DNA.
- (3) Lay out a practical way to assess constraints trade-offs in either sequencing or synthesis in terms of coding overhead.

2 PRELIMINARIES

A DNA molecule is made from two chains of bases depicted in either the top or bottom of figure 1. These chains, or **single-strand DNA (ssDNA)**, are base paired and complementary, i.e.: they are positioned such that A in one strand is hydrogen bonded to T in the other. Similarly G is bonded to C. One ssDNA completely determines the second ssDNA [21]. So as an information carrying medium, we may only decide on the bases making one such strand. Table 1 shows a mapping of bit pairs to bases most common to the literature, with an added notation that serves our analysis and construction.

A	C	T	G
00	01	11	10
0	0	1	1
0	1	1	0

Table 1: A possible mapping between bit pairs and nucleotides.

2.1 Synthesis

The write process for constructing DNA molecules out of bases is DNA synthesis, in which a ssDNA is constructed [16]. Current methods for synthesising DNA often synthesise sets of molecules at once in an array, with each set clustered in a spot on a hard surface. A ssDNA is attached to a glass support on one side and is extended one base at a time, effectively growing the strand according to the prescribed digital sequence. The addition of a base could be done by a process similar to ink-jet printing, or by guiding nucleotides using an electrochemical process [12], [19]. Non-binding of a nucleotide to a designated strand results in a deletion. **Primers** are specific short ssDNA sequences. A primer could

be used for addressing in the synthesis array, or as a known subsequence to identify in QA. When the construction of a ssDNA is complete, a primer is used to start the PCR process that generates the complementary ssDNA. The result is a double-strand DNA molecule (or plainly, DNA).

2.2 Sequencing

Reading is possible through any form of DNA sequencing following amplification of the material. The source molecule is replicated, or amplified, many times over using PCR [10]. As before, PCR requires a primer to start the reaction. Flanking the data by primers is meant to amplify the data from start to finish. Incident of the flanking primers in the data may result in fragments of the DNA being amplified. Once amplified the DNA could be sequenced [22]. Sequencing is enabled by one of several processes, e.g. [14, 15, 18], each with their own error characteristics and constraints.

3 RELATED WORK

Custom modulation schemes for mapping bit sequences to nucleotide sequences are used to satisfy constraints in existing work. For example, mapping each bit to a single nucleotide (0 = A/C, 1 = G/T) [7] or using a base-3 Huffman encoding with a rotating code [2, 5, 6, 20]. The rotating code in [6] could be viewed as a finite state machine (FSM) encoding, and so our work continues it. Reed-Solomon (RS) codes were used in [5, 11] and XOR encoding in [6]. While the input to these error-correcting codes (ECCs) may conform with all constraints, the output may still violate them. This problem is addressed in FountainDNA [9], where the Luby transform repeatedly generates pseudo-random coded data until a satisfactory output is reached.

Architectures have been proposed for building a random access DNA key-value storage system. One such architecture maps keys to pairs of PCR primers flanking data in a strand [6], [20]. To retrieve a key-value pair, the relevant primer is used with PCR to amplify the corresponding strands in the pool and read it. OligoArchive [4] goes one step further, performing common SQL operations on data in its DNA form. A comprehensive survey of storing digital data using DNA is presented in [13].

4 ANALYSIS AS A CONSTRAINED CHANNEL

Some constraints are solvable individually using a mechanism such as scrambling or 8/10b coding [3]. As a concurrent set of constraints, however, encoding into DNA requires treatment as a constrained channel. The approach we take follows that of Mackay [17].

One first needs to decide on a symbol length L to start the analysis. A symbol is then defined to be any sequence of L

bases. Given two symbols S_1 and S_2 their concatenation:

$$\mathbf{S} = S_1 \frown S_2 \quad (1)$$

is simply the sequence obtained by concatenating S_2 to S_1 . Although arbitrarily large lengths could work, it is better if L is smaller than:

- Maximum run length.
- Length of the shortest subsequence to avoid.

Intuitively, if a symbol may already contain a violation we would have to work harder to end up with a violation free encoding, whereas if a violation occurs due to concatenation of symbols we would only have to worry about the concatenation site shown in figure 3.

In this work we chose to focus on a symbol size of 4, but the methods are applicable for any symbol size. Next we enumerate the symbols and form a connectivity matrix A such that $A(i, j) = 1$ if $S_i \frown S_j$ conforms with all constraints and 0 otherwise. Figure 2 shows three matrices each representing a single constraint corresponding to homopolymers with run length ≥ 6 , reserved subsequences, and G-C limits of 15% – 85%. Non-white pixels correspond to $A(i, j) = 0$. Note that we do not incorporate a G-C variance constraint into our example, because representing it with $L = 4$ is too limiting. Let M_N be the number of distinguishable sequences $S_0 \frown S_1 \dots \frown S_{N-1}$ made from concatenating N symbols. Then the capacity of this constrained channel could be defined by:

$$C = \lim_{n \rightarrow \infty} \frac{1}{N} \log(M_N) \quad (2)$$

An approximation of this quantity may be used to compare between different synthesis options, and could be obtained by counting paths through the graph defined by A or by using the largest eigenvalue [17]. The approach we adopt is more practical, i.e.; we will show how to use A to obtain an estimate on the practical number of bits per symbol needed to avoid all constraints that A represents.

We begin by representing the symbols S_1 and S_2 graphically so that certain symmetries of A would translate to a visual meaning. Figure 3 shows two symbols and their concatenation site. Suppose we were willing to reserve a single bit from S_2 depicted in red in figure 3, setting its value according to the rest of bits in order to avoid violations. This gives rise to a new connectivity matrix, A' , between symbols S_1 of four bases on the vertical axis, as before, and symbols S_2 of the form $RxDfE$ on the horizontal axis, where $D, F, E \in \{A, C, T, G\}$ are bases, $x \in \{0, 1\}$ is a single bit and R is a reserved bit. Such a connectivity matrix is depicted in Figure 2. For given i and j , the value of $A'(i, j)$ is 0 if the reserved bit R can be chosen from $\{0, 1\}$ to conform with all the constraints in $S_1 \frown RxDfE$. This happens if $\max(A(S_1, 0xDfE), A(S_1, 1xDfE)) = 1$.

Since the horizontal axis of the connectivity matrix in figure

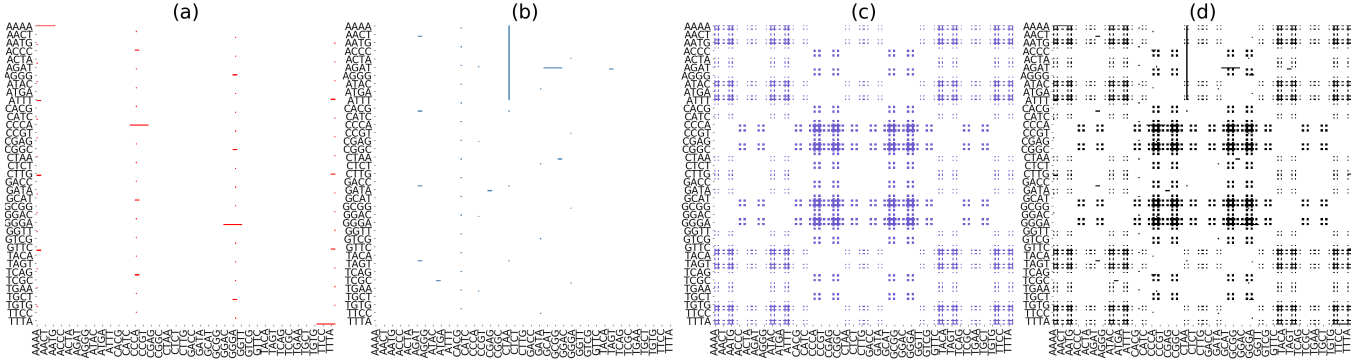


Figure 2: (a) Homopolymers. In this case repetitions of length 6 or more. (b) Primer and reserved subsequences. The longer vertical line corresponds to A[AGCT][AGCT][AGCT]CTAG. (c) G-C content limits 15% – 85%. (d) All previous constraints concurrently.

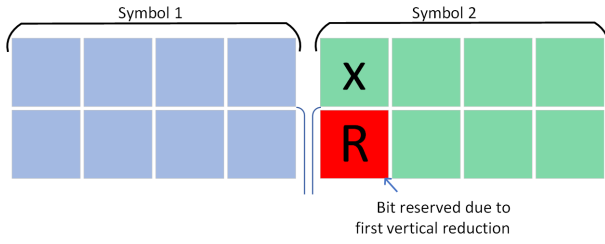


Figure 3: S_1 , S_2 and first bit reserved (red).

2 represents S_2 , halving A vertically gives rise to two connectivity matrices A_0 and A_1 . A_0 is a connectivity matrix between 4 base symbols S_1 and symbols of the form $0xDFE$. Similarly A_1 is a connectivity matrix for S_1 and $1xDFE$. A' is then obtained by the coordinate-wise maximum between A_0 and A_1 .

We may repeat the process until a violation free matrix is obtained, i.e., a connectivity matrix with no 0's or until all bits of S_2 have been consumed for conflict resolution without a full-connectivity matrix in hand. The latter may indicate that at least one symbol already violates the original set of constraints. We chose to reserve bits from S_2 alone. While reserving bits from S_1 yields a similar analysis, its use for the construction in the next section requires more delicate conditions. With this more constructive way, all constraints represented by the matrix (d) in figure 2 can be eliminated by reserving no less than three bits from S_2 . If we were to change, for example, the G-C limits in matrix (c) in figure 2 from 15% – 85% to 35% – 65% then five bits from S_2 would be required.

We could use this analysis to answer the following question: *given a number of reserved bits, what constraints could be met?*. As an example, figure 5 shows what combinations of low

and high G-C content could be handled for 1, 3, 5 and 7 bits out of a four bases long S_2 .

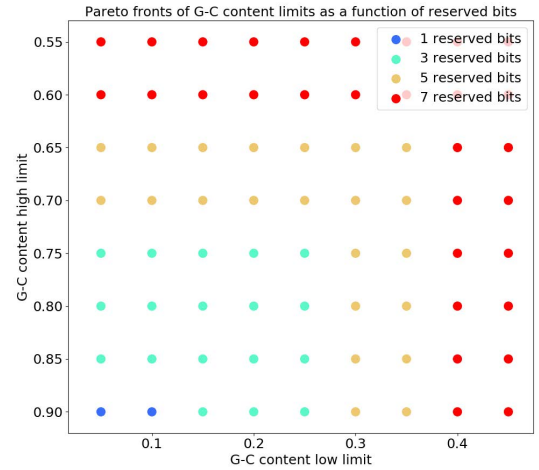


Figure 5: G-C limits 5% – 85% are at the same Pareto front as 25% – 75%, i.e. both require 3 reserved bits, while 35% – 65% require 5.

5 ENCODING USING A FINITE STATE MACHINE

We now use our analysis in the previous section to encode a binary sequence into a sequence of bases. Following our example of 4 bases long symbols, and the constraints depicted in figure 2, we reserve three bits from S_2 according to our convention.

Suppose we somehow managed to concatenate a sequence of symbols $S = S_0 \frown S_1 \dots \frown S_{k-1}$ and would like to extend it by one more. Since, in our example, we reserve the leftmost three bits of the next symbol, we take the next five bits from

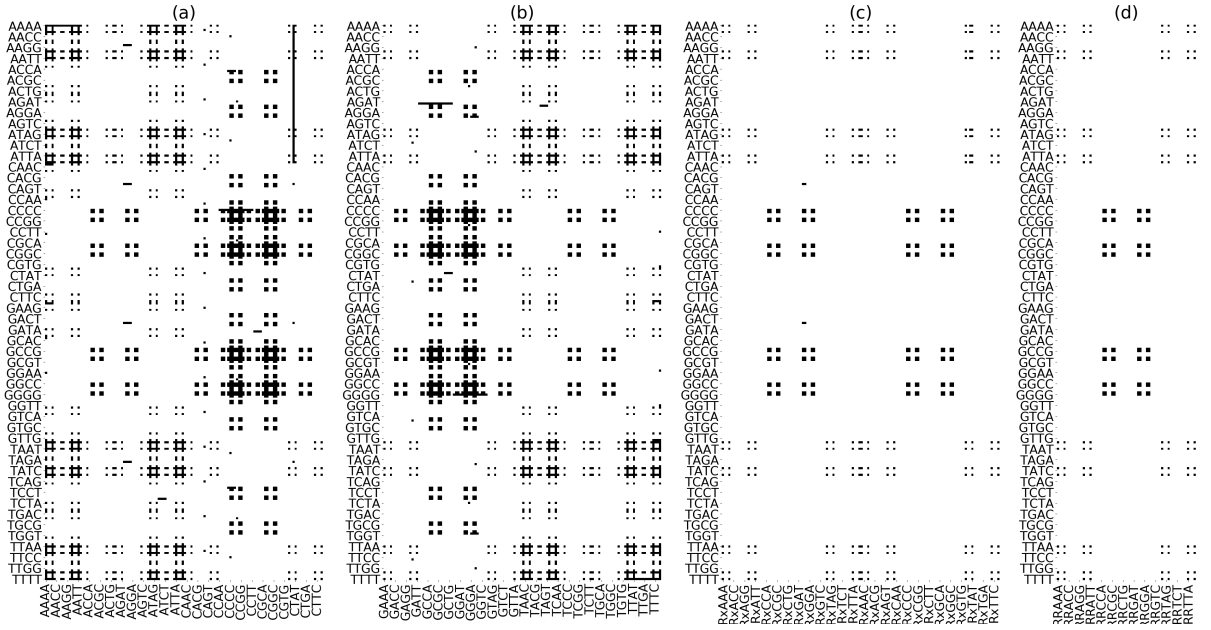


Figure 4: After halving the connectivity matrix in 2 we have two connectivity matrices (a) and (b) that are limited to subsets of symbols for S_2 , with no limitation on S_1 . By taking the coordinate-wise minimum, we obtain a connectivity matrix (c) that allows three arbitrary bases and another arbitrary bit, i.e.: x , but keep one bit reserved to resolve constraints, i.e.: R . Halving further and taking the coordinate-wise minimum yields connectivity matrix (d) in which S_2 is comprised of three arbitrary bases and two reserved bits. Since (c) has a certain symmetry about the vertical centre, the resulting step is not constraint free. A third step (not shown) yields a matrix that reflects full connectivity, but requires three reserved bits, one arbitrary bit and two arbitrary bases.

the binary input and translate them in a consistent order into bWZ where $W, Z \in A, C, T, G$ and b is a single bit.

OBSERVATION 5.1 (VIOLATION RESOLUTION BITS). *There exists a choice of three bits R_0, R_1, R_2 such that $S_k = \begin{bmatrix} R_0 & R_2 \\ R_1 & b \end{bmatrix} WZ$ could be appended to S_{k-1} .*

This is true because our stopping criterion for repeated reduction of the original connectivity matrix, reserving consecutive bits while we reduce it, was reaching a violation free matrix. This process could be described by a finite state machine (FSM). To do so we need to define the set of states S , inputs $\mathcal{A}$ and outputs $\mathcal{O}$. At state s and input a , the FSM moves to state s' and emits the output o .

In our example, the set of states S and the outputs $\mathcal{O}$ are defined as 4 bases long symbols. An input is any 2.5 bases long sequence bWZ . The output o , as well as the next state s' is defined as the symbol formed from concatenating the bits $R_0R_1R_2$ to the input sequence. For this to be well defined and deterministic, we need to make sure that the choice of $R_0R_1R_2$ is the same every time the input sequence is bWZ and the machine is at state s .

6 CONCLUSION AND FURTHER WORK

We have shown a systematic way to construct a constraint driven encoding from binary sequences to DNA. The resulting encoding conforms with the set of constraints dictated by the underlying processes. Additionally, the example in figure 5 measures the impact of relaxing or tightening a specific constraint. The same reasoning applies to other constraints. These tools are valuable in designing a DNA storage system because the technology around digital storage in DNA is advancing. We used the binary connectivity matrix, but one could assign cost to violations (instead of 0) and change the construction to choose reserved bits that minimise this cost. Finite state machines are used in encoding of convolutional error correcting codes, and our formulation of the encoding as an FSM is intended to level up and leverage our constraints driven encoding to an error correcting code that handles deletions and insertions as well. Further work includes a choice mechanism for the values of the reserved bits. Our current work involves using the Viterbi algorithm for decoding, which is in fact an existing instrument in biological sequence alignment [8].

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive comments and helpful suggestions. This work was partially funded by the European Union's Horizon 2020 research and innovation programme, project OligoArchive (grant agreement No 863320).

REFERENCES

- [1] URL: <https://www.genescrypt.com>.
- [2] Towards practical, high-capacity, low-maintenance information storage in synthesized dna. *Nature*, 494(7435):77–80, 2013.
- [3] Ieee standard for ethernet amendment 2: Physical layer specifications and management parameters for 100 gb/s operation over backplanes and copper cables. *IEEE Std 802.3bj-2014 (Amendment to IEEE Std 802.3-2012 as amended by IEEE Std 802.3bk-2013)*, pages 1–368, 2014. doi:10.1109/IEEESTD.2014.6891095.
- [4] Raja Appuswamy, Kevin Lebrigand, Pascal Barbry, Marc Antonini, Oliver Madderson, Paul Freemont, James MacDonald, and Thomas Heinis. Oligoarchive: Using dna in the dbms storage hierarchy. In *Bienal Conference on Innovative Data Systems Research, CIDR*, volume 19, 2019. URL: <http://www.eurecom.fr/en/publication/5783/download/data-publi-5783.pdf>.
- [5] Meinolf Blawat, Klaus Gaedke, Ingo Huetter, Xiao-Ming Chen, Brian Turczyk, Samuel Inverso, Benjamin W Pruitt, and George M Church. Forward error correction for dna data storage. *Procedia Computer Science*, 80:1011–1022, 2016. URL: <https://doi.org/10.1016/j.procs.2016.05.398>.
- [6] James Bornholt, Randolph Lopez, Douglas M Carmean, Luis Ceze, Georg Seelig, and Karin Strauss. Toward a dna-based archival storage system. *IEEE Micro*, 37(3):98–104, 2017. URL: <https://homes.cs.washington.edu/~bornholt/papers/dnastorage-topicks17.pdf>.
- [7] George M Church, Yuan Gao, and Sriram Kosuri. Next-generation digital information storage in dna. *Science*, 337(6102):1628–1628, 2012.
- [8] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.
- [9] Yaniv Erlich and Dina Zielinski. Dna fountain enables a robust and efficient storage architecture. 355(6328):950–954, 2017.
- [10] Lilit Garibyan and Nidhi Avashia. Research techniques made simple: polymerase chain reaction (pcr). *The Journal of investigative dermatology*, 133(3):e6, 2013.
- [11] Robert N. Grass, Reinhard Heckel, Michela Puddu, Daniela Paunescu, and Wendelin J. Stark. Robust Chemical Preservation of Digital Information on DNA in Silica with Error-Correcting Codes. *Angewandte Chemie International Edition*, 54(8):2552–2555, 2 2015. URL: <http://doi.wiley.com/10.1002/anie.201411378>, doi:10.1002/anie.201411378.
- [12] Reinhard Heckel, Gediminas Mikutis, and Robert N Grass. A characterization of the dna data storage channel. *arXiv preprint arXiv:1803.03322*, 2018.
- [13] Thomas Heinis. Survey of information encoding techniques for dna. *arXiv preprint arXiv:1906.11062*, 2019. URL: <https://arxiv.org/abs/1906.11062>.
- [14] Illumina. Technology Spotlight: Illumina ® Sequencing. Technical report.
- [15] Miten Jain, Hugh E. Olsen, Benedict Paten, and Mark Akeson. The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community. *Genome Biology*, 17(1):239, 12 2016. URL: <http://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-1103-0>, doi:10.1186/s13059-016-1103-0.
- [16] Sriram Kosuri and George M Church. Large-scale de novo dna synthesis: technologies and applications. *Nature methods*, 11(5):499, 2014. URL: http://arep.med.harvard.edu/pdf/Kosuri_Church_2014.pdf.
- [17] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003. URL: <http://www.inference.org.uk/mackay/itila/book.html>.
- [18] Michael L. Metzker. Sequencing technologies the next generation, 1 2010. URL: www.nature.com/reviews/genetics, doi:10.1038/nrg2626.
- [19] Melissa B Miller and Yi-Wei Tang. Basic concepts of microarrays and potential applications in clinical microbiology. *Clinical microbiology reviews*, 22(4):611–633, 2009. URL: <https://cmr.asm.org/content/cmrv22/4/611.full.pdf>.
- [20] Lee Organick, Siena Dumas Ang, Yuan-Jyue Chen, Randolph Lopez, Sergey Yekhanin, Konstantin Makarychev, Miklos Z Racz, Govinda Kamath, Parikshit Gopalan, Bichlien Nguyen, et al. Random access in large-scale dna data storage. *Nature biotechnology*, 36(3):242, 2018. URL: <https://www.nature.com/articles/nbt.4079>.
- [21] Jane B Reece, Lisa A Urry, Michael Lee Cain, Steven Alexander Wasserman, Peter V Minorsky, Robert B Jackson, et al. *Campbell biology*. Number s 1309. Pearson Boston, 2014.
- [22] Fred Sanger and Alan R Coulson. A rapid method for determining sequences in dna by primed synthesis with dna polymerase. *Journal of molecular biology*, 94(3):441–448, 1975. URL: <https://www.sciencedirect.com/science/article/pii/0022283675902132>, doi:https://doi.org/10.1016/0022-2836(75)90213-2.
- [23] Deepali Shinde, Yinglei Lai, Fengzhu Sun, and Norman Arnheim. Taq dna polymerase slippage mutation rates measured by pcr and quasi-likelihood analysis:(ca/gt) n and (a/t) n microsatellites. *Nucleic acids research*, 31(3):974–980, 2003.
- [24] twistbioscience. URL: <https://www.twistbioscience.com/>.
- [25] Enrique Viguera, Danielle Canceill, and S Dusko Ehrlich. Replication slippage involves dna polymerase pausing and dissociation. *The EMBO journal*, 20(10):2587–2595, 2001.