DNA-Based Storage: Trends and Methods

S. M. Hossein Tabatabaei Yazdi, Han Mao Kiah, Eva Garcia-Ruiz, Jian Ma, Huimin Zhao, and Olgica Milenkovic, *Senior Member, IEEE*

(Invited Paper)

***Abstract*—We provide an overview of current approaches to DNA-based storage system design and of accompanying synthesis, sequencing and editing methods. We also introduce and analyze a suite of new constrained coding schemes for both archival and random access DNA storage channels. The analytic contribution of our work is the construction and design of sequences over discrete alphabets that avoid pre-specified address patterns, have balanced base content, and exhibit other relevant substring constraints. These schemes adapt the stored signals to the DNA medium and thereby reduce the inherent error-rate of the system.**

***Index Terms*—Constrained and error-control coding, DNA-based storage, DNA synthesis and sequencing.**

I. Introduction

DESPITE the many advances in traditional data recording techniques, the surge of Big Data platforms and energy conservation issues have imposed new challenges to the storage community in terms of identifying extremely high volume, non-volatile and durable recording media. The potential for using macromolecules for ultra-dense storage was recognized as early as in the 1960s, when the celebrated physicists Richard Feynman outlined his vision for nanotechnology in the talk "There is plenty of room at the bottom." Among known macromolecules, DNA is unique in so far that it lends itself to implementations of non-volatile recoding media of outstanding integrity (one can still recover the DNA of species extinct for more than 10,000 years) and extremely high storage capacity (a human cell, with a mass of roughly 3 picograms, hosts DNA encoding 6.4 GB of information). Building upon the rapid growth of biotechnological systems for DNA synthesis and sequencing, two laboratories recently outlined architectures for archival DNA based storage in [1], [2]. The first architecture achieved a density of 700 TB/gram, while the second approach raised the density to 2 PB/gram. The success of the later method was largely attributed to the use of three elementary coding schemes, Huffman coding (a fixed-to-variable length entropy coding/compression method), differential coding (encoding the differences of consecutive symbols or the difference between a sequence and a given template) and single parity-check coding (encoding of a single symbol indicating the parity of the string). More recent work [3] extended the coding approach used in [2] by replacing single parity-check codes with Reed-Solomon codes [4].

All the aforementioned approaches have a number of drawbacks, including the lack of partial access to data – i.e., one has to reconstruct the whole sequence in order to read even one base – and the unavailability of rewrite mechanisms. Moving from a read only to a random access, rewritable memory requires a major paradigm shift in the implementation of the DNA storage system, as one has to append unique addresses to constituent storage DNA blocks that will not lead to erroneous cross-hybridization with the information encoded in the blocks; avoid using overlapping DNA blocks for increased coverage and subsequent synthesis, as they prevent efficient rewriting; and ensure low synthesis (write) and sequencing (read) error rates of the DNA blocks. To overcome these and other issues, the authors recently proposed a (hybrid) DNA rewritable storage architecture with random access capabilities [6]. The new DNA-based storage scheme encompasses a number of coding features, including constrained coding, ensuring that DNA patterns prone to sequencing errors are avoided; prefix synchronized coding, ensuring that blocks of DNA may be accurately accessed without perturbing other blocks in the DNA pool; and low-density parity-check (LDPC) coding for classically stored redundancy combating rewrite errors [7].

The shared features of current DNA-based storage architectures are depicted in Figure 1. The green circles denote the source and media, while the blue circles denote processing methods applied on the source and media. The processes of Encoding and DNA Encoding add controlled redundancy into the original source of digital information or into the DNA blocks, respectively. This redundancy can be use to combat synthesis (write) and sequencing (access and read) errors [8]–[10]. Synthesis is the biochemical process of creating physical double-stranded DNA strings that reliably represent the encoded data strings. Synthesis thereby also creates the storage media itself – the DNA blocks. Storage refers to some means of storing the DNA strings, and it represents a communication channel that transfers information from one point in time to another. In rewritable architectures [6], the Editing module encompasses the process of creating mutations in the stored DNA strings (by deleting one or multiple substrings and

Manuscript received July 6, 2015; revised October 20, 2015; accepted November 30, 2015. Date of publication March 2, 2016; date of current version April 5, 2016. This work was supported in part by the NSF STC Class 2010 CCF 0939370 grant and in part by the Strategic Research Initiative (SRI) Grant conferred by the University of Illinois, Urbana-Champaign. The associate editor coordinating the review of this paper and approving it for publication was S. M. Moser.

S. M. H. T. Yazdi and O. Milenkovic are with the Department of Electrical and Computer Engineering, University of Illinois, Urbana-Champaign, Champaign, IL 61801 USA (e-mail: milenkov@illinois.edu).

H. M. Kiah is with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore.

E. Garcia-Ruiz and H. Zhao are with the Department of Chemical and Biomolecular Engineering, Carl R. Woese Institute for Genomic Biology, University of Illinois, Urbana-Champaign, Champaign, IL USA.

J. Ma is with the Department of Bioengineering, Carl R. Woese Institute for Genomic Biology, University of Illinois, Urbana-Champaign, Champaign, IL USA.

Digital Object Identifier 10.1109/TMBMC.2016.2537305

2332-7804 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

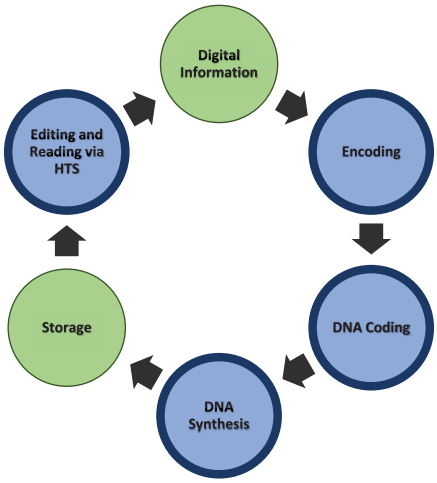

Fig. 1. Block Diagram of Prototypical DNA-Based Storage Systems. A classical information source is encoded (converted into ASCII or some specialized word format, potentially compressed, and represented over a four letter alphabet); subsequently, the strings over four-letter alphabets are encoded using standard and DNA-adapted constrained and/or error-control coding schemes. The DNA codewords are synthesized, with potential undesired mutations (errors) added in the process, and stored. When possible, rewriting is performed via classical DNA editing methods used in synthetic biology. Sequencing is performed either through Sanger sequencing [5], if short information blocks are accessed, or via High Throughput Sequencing (HTS) techniques, if large portions of the archive are selected for readout.

potentially inserting other strings), while the Reading module refers to DNA sequencing that retrieves the content of selected DNA storage blocks and subsequent decoding.

In order to understand how errors occur during the read and write process, we start our exposition by describing state-of-the art synthesis, sequencing and editing methods (Sections II–IV). We then proceed to discuss how synthesis, sequencing and editing methods are used in various DNA-storage paradigms (Section V), and the accompanying coding techniques identified with different types of synthesis and sequencing errors. New constrained coding techniques for rewritable and random access systems, and their relationship to classical codes for magnetic and optical storage, are described in Section VI.

Given the semi-tutorial and interdisciplinary nature of this manuscript, we refer readers with a limited background in synthetic biology to the Appendix for a glossary of terms used throughout the paper.

II. DNA Sequence Synthesis

De novo DNA synthesis is a powerful biotechnological process that enables the creation of DNA sequences without pre-existing templates. Synthesis tools have a myriad of applications in different research areas, ranging from traditional molecular biology to emerging fields of synthetic biology, nanotechnology and data storage. Vaguely speaking, most technologies for large-scale DNA synthesis rely on the assembly of pools of oligonucleotide building blocks into increasingly larger DNA fragments. The current high cost and small throughput of de novo synthesis of these building blocks represents the main limitation for widespread implementations of DNA synthesis systems: as an example, oligo synthesis methods via *phosphoramidite column-based synthesis*, described in subsequent sections, may cost as much as $0.15 per nucleotide [11]. The maximum length of the produced oligostrings lies in the range 100–200 nts [11]. Hence, the synthesis of long DNA oligos using hundreds of building blocks can cost anywhere from hundreds to thousands of US dollars. Therefore, it is imperative to develop new, high-quality, robust, and scalable DNA synthesis technologies that offer synthetic DNA at significantly more affordable prices. This is in particular the case for massive DNA-based storage systems, which may potentially require billions of nucleotides.

Among the most promising synthesis technologies is the so called *microarray-based* synthesis method; more than ten-to-hundreds of thousands oligos can be synthesized per one microarray, in conjunction with a decrease in the reagent consumption. For large scale DNA synthesis projects, the price of microarray-based synthesis is roughly $0.001 per nucleotide [11], [12]. Similarly to the case of phosphoramidite column-based synthesis, the length of microarray synthesized oligos usually does not exceed 200 nt. However, oligos synthesized in microarrays typically suffer from higher error rates than those generated by phosphoramidite column methods. Nevertheless, microarrays are the preferred synthesis tool for generating customized DNA-chips and for performing gene synthesis. Many projects are underway to bridge the gap between these two extremes, hight-cost and high-accuracy and low-cost, low-accuracy strategies and hence reduce the limitations of the corresponding methods [13], [14].

To provide a better understanding of the basic principles of DNA-based storage and the limitations that need to be overcome in the writing process, we first describe different DNA synthesis methods from nucleotides to larger DNA molecules. We then discuss recent techniques that aim to improve the quality and reliability of the synthesized sequences.

A. Chemical Oligonucleotide Synthesis

Chemical synthesis of single stranded DNA originated more than 60 years ago, and since the 1950's, when the first oligonucleotides were synthesized [15]–[17], four different chemical methods have been developed. These methods are named after the major reagents included in the process, and include i) H-phosphonate; ii) phosphodiester; iii) phosphotriester; and iv) phosphite triester/phosphoramidite. A detailed description of these methods may be found in [18], [19], and here we only briefly review the advantages and disadvantages of these methods.

The H-phosphonate method was first described in [16], and it derives its name from the use of H-phosphonates nucleotides as building blocks. This approach was later refined in [20], [21], where the H-phosphonate chemistry was improved to synthesize deoxyoligonucleotides on a solid support by using different oligo coupling (stitching) agents that expedite the reactions. The phosphodiester method was introduced in [17], [22]. Unfortunately, the approach had one major drawback – the linkages between nucleotides were unprotected during the elongation step of the oligonucleotide chain, which allowed for the creation of branched oligonucleotides. The phosphotriester approach was first published in the 50s [15] and later improved by Letsinger [23], [24] and Reese [25] using different

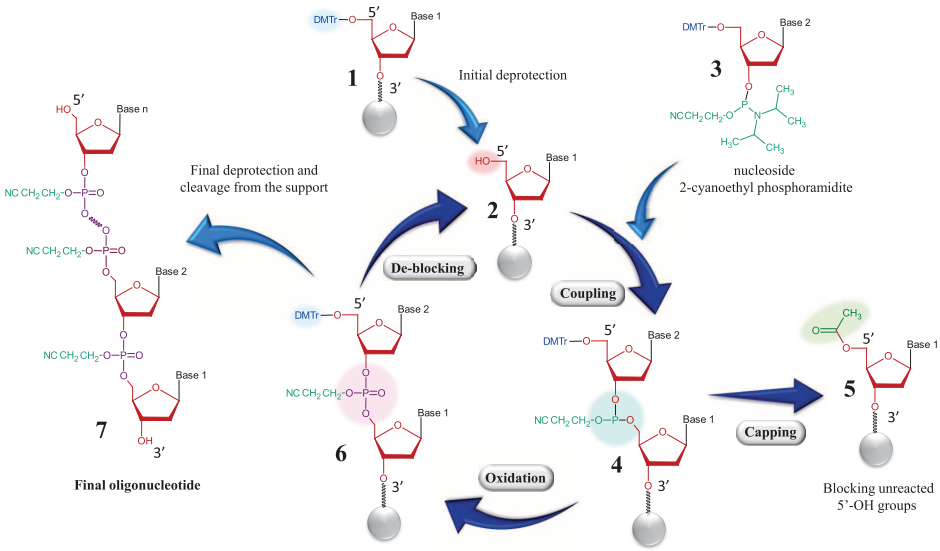

Fig. 2. The Main Steps of Column-Based Oligo Synthesis of Section II-B1. The first step in DNA synthesis cycle is the deprotection of the support-bound nucleoside at the 5' terminal end (1, highlighted in blue) by removal of the DMTrgroup. This step lead a nucleoside with a 5' OH group (2, highlighted in red). During the coupling step an activated nucleoside (3) react with the 5' OH group of the support-bound nucleoside (2) generating a dinucleotide phosphoramidite (4) (formation of phosphitetriester, highlighted in green-blue). In the capping step, unreacted 5' OH are blocked by acetylation (5, highlighted in green) to prevent further chain extension. In the last step of the cycle the unstable phosphitetriester (in green-blue) is oxidized to phosphate linkage (6, highlighted in purple) which is more stable in the chemical conditions of the following synthesis steps. The cycle is repeated for each nucleoside addition. After the last step of the synthesis of the entire oligonucleotide, the final product needs to be cleavage from the solid support and deprotect the 5' terminal end. In red is the pentose of the nucleoside, in blue the dimethoxytrityl (DMTr) protecting group, in green-blue the 2-cyanoethyl phosphoramiditegroup, and in purple the phosphate group. Grey spheres represent the solid support in which the growing oligo is attached. Circles highlight the group that is modified in each step.

reagents to protect the phosphate group in the internucleotide linkages. This approach prevented the formation of branched oligonucleotides. Nevertheless, all the previously described methods and variants thereof proved to be inefficient and time consuming.

In the mid-seventies, a major advantage in synthesis technology was reported by Letsinger [26], solving in part a number of problems associated with other existing methods. His method was termed the *phosphite triester approach*. The basic idea behind the approach was that the reagent phosphorochloridite reacts with nucleotides faster than its chloridate counterpart used in previous approaches. In addition to expediting their underlying reactions, bifunctional phosphorodichloridites unfortunately also produced undesirable side products such as symmetric dimers. A modified method that precluded the drawback of side products was developed by Caruthers *et al.* [27]. The authors of [27] used a different type of nucleoside phosphites that were more stable, reacted faster, and produced higher yields of the desired dinucleoside phosphite. The resulting method was named *phosphoramidite synthesis*. Another important contribution includes the technology described in [28], where the use of stable and easy-to-prepare phosphoramidites facilitated the automation of oligo synthesis in solid-phase, making it the method-of-choice for chemical synthesis.

B. Oligo Synthesis Platforms

1) Column-Based Oligo Synthesis: The standard phosphoramidite oligonucleotide synthesis operates via stepwise addition of nucleotides to the growing chain which is immobilized on a solid support (Figure 2). Each addition cycle consists of four chemical steps: i) de-blocking; ii) coupling or condensation; iii) capping; and iv) oxidation [18]. At the beginning of the synthesis process, the first nucleotide, which is attached to a solid substrate, is completely protected at all of its active sites. Therefore, to make a reaction possible and include a second nucleotide, it is necessary to remove the dimethoxytrityl (DMT) protecting group from the 5'-end by addition of an acid solution. The removal of the DMT group generates a reactive 5'-OH group (De-blocking step). Subsequently, a coupling step is performed via condensation of a newly activated DMT-protected nucleotide and the unprotected 5'-OH group of the substrate-bound growing oligostrand through the formation of a phosphite triester link (Coupling or Condensation step). After the coupling step, some unprotected 5'-OH groups may still exist and react in later stages of additions of nucleotides leading to oligos with *deletion and bursty deletion errors*. To mitigate this problem, a capping reaction is performed by acetylation of the unreactive nucleotides (Capping step). Finally, the unstable phosphite triester linkage is oxidized to a more stable phosphate linkage using an iodine solution (Oxidation step). The cycle is repeated iteratively to obtain an oligonucleotide of the desired sequence composition. At the end of the synthesis, the oligonucleotide sequence is deprotected, and cleaved from the support to obtain a completely functional unit.

2) Array-Based Oligo Synthesis: In the 90 s, Affymetrix developed a method for chemical synthesis of different polymers combining photolabile protecting groups and photolithography [29], [30]. The Affymetrix solution uses a photolithographic mask to direct UV light in a targeted manner, so as to selectively deprotect and activate 5' hydroxyl groups of

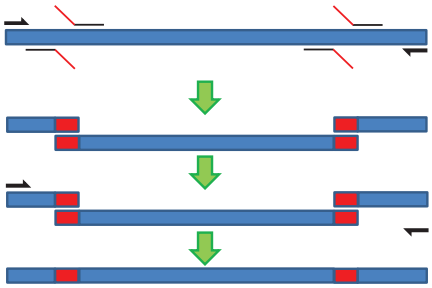

Fig. 3. Rewriting (Deletion and Insertion Edits) via gBlocks. This method is used when edits of relatively short length are required, as it is cost efficient and simple. Primers corresponding to unique contexts in the encoded DNA are used to access the edit region, which is subsequently cleaved and replaced by the gBlock.

nucleotides that should react with the nucleotide to be incorporated in the next step. The mask is designed to expose specific sites on the microarray to which new nucleotides will be added, with others sites being masked. Once synthesis is completed, the oligos are released from the array support and recovered as a complex mixture (pool) of sequences.

A number of other, related methods have been developed for the purpose of synthesizing oligostrands on microarrays [31]. For instance, the method developed by Agilent uses Ink-jet-based printing, where with high precision, picoliters of each incorporated nucleotide and activator can be spotted (deposited) at specific sites on an array. This ink-jet method mitigates the need for using photolithography masks [32]. In an alternative method commercialized by NimbleGen Systems, the photolithography masks are superseded by a virtual mask that is combined with digital programmable mirrors to activate specific locations on the array [33], [34]. CustomArray (former CombiMatrix) developed a technology in which thousands of microelectrodes control acid production by an electrochemical reaction to deprotect the growing oligo at a desired spot [35]. In addition, oligo synthesis is implemented within a multichamber microfluidic device coupled to a digital optical device that uses light to produce acid in the chambers [36]. Masking and printing errors may introduce both *substitution* and *insertion and deletion* errors, and when multiple sequences are synthesized simultaneously, the error patterns within different sequences may be correlated, depending on the location of their synthesis spots.

Both solid-phase and microarray technologies exhibit a number of challenges that need to be overcome to reduce error rates and increase throughout. Side reactions such as depurination [37], [38] and reaction inefficiencies during the stepwise addition of nucleotides [18], [19] reduce the desired yield, and generate errors in the sequence especially when synthesizing long oligostrands. In particular, these processing problems introduce both *substitution* and *insertion and deletion* errors. Thus, a purification step is usually necessary to identify and discard undesirable erroneous sequences. High-performance liquid chromatography and polyacrylamide gel electrophoresis can be used to eliminate truncated products, but both are expensive and time-consuming, and single insertions and deletions or substitution errors in the sequence often cannot be removed. Nevertheless, by optimizing chemical reaction and conditions the fidelity can be increased [38].

3) Complex Strand and Gene Synthesis: Traditionally, to generate DNA fragments of length several hundred nucleotides, a set of shorter length oligostrands is fused together by either using ligation-based or polymerase-based reactions. Ligation-based approaches usually rely on thermostable DNA ligases that ligate phosphorylated overlapping oligos in high stringency conditions [39]. In polymerase-based approaches (Polymerase cycling assembly - PCA) oligos with overlapping regions are used to generate progressively longer double-stranded sequences [40]. After assembly, synthesized sequences need to be PCR amplified, cloned, and verified, thus increasing the cost of production. Another approach developed by Gibson *et al.* [41] exploits yeast in vivo recombination to assemble a set of more than 30 oligos together with a plasmid, all in one step. The same group also synthesized the mouse mitochondrial genome from 600 overlapping oligos using an isothermal assembly method [42].

Although microarray synthesis reduces the price of oligonucleotides, there are two major challenges that still hamper its use. First, hundreds of thousands of oligonucleotides can be made on a single microarray, but each oligo is produced in very small amounts. Second, the oligostrands are cleaved from the array all at once as a large heterogeneous pool which subsequently leads to difficulties in sequence assembly and cross-hybridization. A number of strategies have been recently developed to solve these problems. For example, PCR amplification increases the concentration of the oligos before assembly that combined with hybridization selection reduces the incorporation of oligonucleotides containing undesirable synthesis errors [43]. A modification of this approach, based on hybridization selection embedded in the assembly process and coupled with the optimization of oligo design and assembly conditions was reported in [44]. Still, large pools of oligos (>10000) increase difficulties in sequence assembly. Two different strategies have been described where subpools of oligos involved in a particular assembly were isolated, thus partially avoiding cross-hybridization. Kosuri *et al.* [45] used predesigned barcodes to amplify subpools of oligos, and in a second step removed the barcodes by digestion. In another approach, the microarray was physically divided in sub-arrays that enabled performing amplification and assembly separately in each microwell [46].

4) Error Correction: Despite having elaborate biochemical error removal processes in place, some residual errors tend to remain in the synthesized pool and additional errors arise during the assembly phase. A number of error-correction strategies have been reported in the literature [11], [13], [14]. Many of the current error-removal techniques rely on DNA mismatch recognition proteins. Denaturation and re-hybridization steps lead to double-stranded DNA with mismatches between erroneous bases and the corresponding correct bases. The disrupted sites are recognized and/or cleaved by mismatch recognition proteins. MutS is a protein that binds unpaired bases and small DNA loops (i.e., small unmatched substrings in DNA that protrude from the double helix). After denaturation and re-hybridization, MutS detects and binds the mismatched

regions that are later removed by gel electrophoresis. This strategy reduces the error-rate to 1 nucleotide per 10 Kb [47]. "Consensus shuffling" is a variation of the MutS method where mismatch-containing pieces are captured by column-immobilized MutS proteins, and error-free fragments are eluted [48]. In other variations of this method, two homologs of MutS immobilized in cellulose columns can reduce the error rate to 0.6 nucleotides per Kb at a very low cost [49]. On the other hand, in the MutHLS approach, MutS binds unpaired bases, while the protein MutL links the MutH endonuclease to the MutS bound sites that cleave the erroneous heteroduplexes. The correct sequences are recovered by gel electrophoresis [50]. Similarly, resolvases [51] and single-strand nucleases [52], [53] may also be used to recognize and cleave mismatched sites in DNA heteroduplexes. It is worth pointing out that CEL endonuclease, its commercial version Surveyor^{TM} nuclease (Transgenomic, Inc.) or a commercial CEL-based enzymatic cocktail, ErrASE, that recognizes and nicks at the base-substitution mismatch, is commonly used in practice due to its broad substrate specificity; it can reduce the error rate up to 1 nucleotide per 9.6 Kb [54], [55].

The introduction of Next Generation Sequencing (NGS) platforms as high throughput purification methods opened new possibilities for error-free DNA synthesis. Matzas *et al.* [56] combined a next-generation pyrosequencing platform with a robotic system to image and pick beads containing sequence-verified oligonucleotides. The estimated error rate using this approach is 1 nucleotide error per 21 Kb. One limitation of this method is that the "pick-and-place" recovery system is not accurate enough, due to the small size of clonal beads, to satisfy the increasing demand for long length DNA strands (involving 10^4 building blocks) [57]. A new NGS-based method was recently announced, where specific barcoded primers were used to amplify only those oligos with the correct sequence [58], [59]. Similarly, a new method termed "sniper cloning" has been reported in [57]. There, NGS platform beads containing sequence-verified oligonucleotides are recovered by "shooting" a laser pulse. This laser technology enables cost-effective, high throughput selective separation of correct fragments without cross-contamination.

As a parting note, we observe that even single substitution errors in the synthesis process may be detrimental for applications in biological and medical research. This is *not the case* for DNA-based storage systems, where the DNA strands are used as storage media which may have a non-negligible error rate. Synthesis errors may be easily combated through the introduction of carefully designed parity-checks of the information strings, as will be discussed in subsequent sections.

III. DNA Editing

Once desired information is stored in DNA by synthesizing properly encoded heteroduplexes, it may be rewritten using classical *DNA editing* techniques. DNA editing is the process of adding very specific point mutations (often with the precision of a few nucleotides) or deleting and inserting DNA substrings at tightly controlled locations. In the latter case, one needs to synthesize readily usable short-to-medium length DNA fragments.

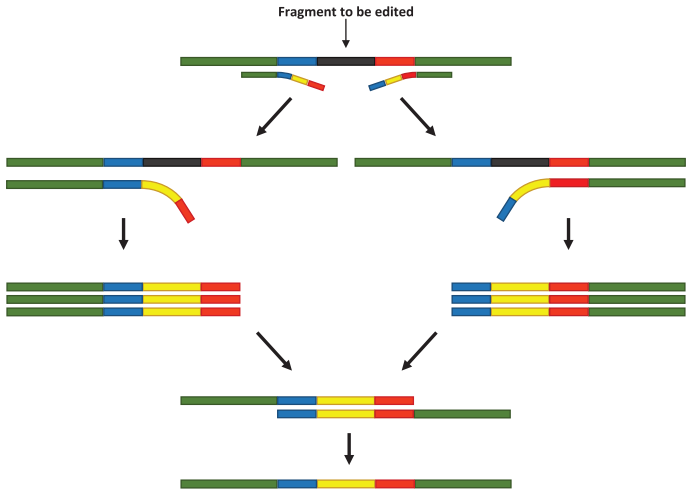

Fig. 4. Rewriting (Deletion and Insertion Edits) via OEPCR. OEPCR allows for incorporating customized sequence changes via primers used in amplification reactions. As the primers have terminal complementarity, two separate DNA fragments may be amplified and fused into a single sequence without using restriction endonuclease sites. Overlapping fragments are fused together in an extension reaction and PCR amplified.

For this purpose, two techniques are commonly used: gBlocks Gene Fragments [60] (see Integrated DNA Techologies) as building blocks for insertion and deletion edits, and Overlap-Extension PCR (OEPCR) [61] as a means of adding the mutated blocks.

gBlocks are double-stranded, precisely content-controlled DNA blocks that may be used for applications as diverse as gene construction, PCR and qPCR control, recombinant antibody research, protein engineering, CRISPR-mediated genome editing and general medical research [62] (see Figure 3 for an illustration). They are usually constructed at very low cost (fraction of a dollar) using *gene fragments libraries*, i.e., pools of short DNA strings that contain up to 18 consecutive bases of type N (any nucleotide) or K (Keto). The libraries and library products are carefully tested for correct length via capillary electrophoresis, and for sequence composition via mass spectrometry; consensus protocols are used in the final verification stage to reduce any potential errors. The last stage, and additional quality control testing ensures that at least 80% of the generated pool contains the desired string. For strings with complex secondary structure, this percentage may be significantly lower. This calls for controlling the secondary structure of the products whenever the applications allows for it. Such is the case for DNA-based storage, and methods for designing DNA codewords with no secondary structure (predicted to the best extent possible via combinatorial techniques) were described in [63].

DNA substring editing is frequently performed via specialized PCR reactions. Of particular use in DNA rewriting is the process of OEPCR, illustrated in Figure 4. In OEPCR, one uses two primers to flank two ends of the string to be edited. For fragment deletion (splicing), the flanking primers act like zippers that need to join over the segment to be sliced. Furthermore, the primer at the end to be joined is designed so that it has an overhanging part complementary to the overhanging part of the other primer. Via controlled hybridization, the DNA strands

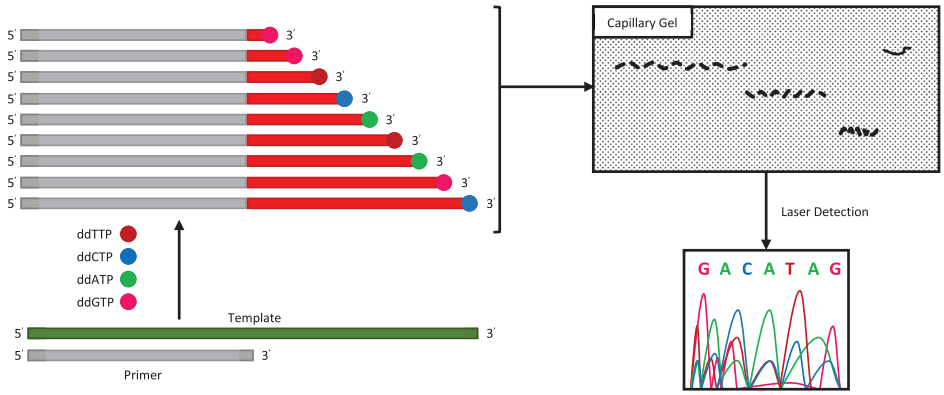

Fig. 5. Main Steps of the Sanger Sequencing Protocol. In the first step, a pool of DNA fragments is sequenced via synthesis. Synthesis terminates whenever chemically inactive versions of nucleotides (dd*TP) are incorporated into the growing chains. These inactive nucleotides are fluorescently labeled to uniquely determine their bases. In the second step, the fragments are sorted by length using capillary gel methods. The terminal step involves reading the last bases in the fragments using laser systems.

are augmented by a DNA insert that is also complementary to the underlying DNA strand. Upon completion of this extension, classical PCR amplification is performed for the elongated sequence primers and the inserted overlapping fragments of the sequences are fused. Note that this method does not requiring restriction sites or enzymes. OEPCR is mostly used to insert oligonucleotides of lengths longer than 100 nucleotides. In OEPCR the sequence being modified is used to make two modified strands with the mutation at opposite ends, using the method outlined above. After denaturation, the strands are mixed, leading to different hybridization products. Of all the products, only one will allow for polymerase extension via the introduction of a primer – the heterodimer without overlap at the 5' end. The duplex created by the polymerase is denatured once again and another primer is hybridized to the created DNA strand, introducing a sequence contained in the first primer. DNA replication consequently results in an extended sequence containing the desired insert.

IV. DNA Sequencing

The goal of DNA sequencing is to read the DNA content, i.e., to determine the exact nucleotides and their order in a DNA molecule. Such information is critical for understanding both basic biology and human diseases as well as for developing nature-inspired computational platforms.

Sanger *et al.* [64] first developed sequencing methods to sequence DNA based on chain termination (see Figure 5 for an illustration). This technique, which is commonly referred to as Sanger sequencing, has been widely used for several decades and it is still being used routinely in numerous laboratories. The automated and parallelized approaches of Sanger sequencing directly led to the success of the Human Genome Project [65] and the genome sequencing projects of other important model organisms for biomedical research (e.g., mouse [66]). The availability of these entire genomes has provided scientists with unprecedented opportunities to make novel discoveries for genome architecture and genome function, trajectory of genome evolution, and molecular bases of phenotypic variation and disease mechanisms.

However, in the past decade, the development of faster, cheaper, and higher-throughput sequencing technologies has dramatically expanded the reach of genomic studies. These "next-generation sequencing" (NGS) technologies, as opposed to Sanger sequencing which is considered as first-generation, have been one of the most disruptive modern technological advances. In general, the NGS technologies have several major differences when compared to Sanger sequencing (see Figure 5). First, electrophoresis is no longer needed for reading the sequencing output (i.e., substring lengths) – it is now typically detected directly. Second, more straightforward library preparations that do not use DNA clones have become a critical part of sequencing workflow. Third, tremendously large number of sequencing reactions are generated in parallel with ultra-high throughput. A demonstration of the significant NGS technology development is the cost reduction. Around the year 2001, the cost of sequencing a million base-pairs was about $5,000; but it only costs about $0.05 in mid 2015 (http://www.genome.gov/sequencingcosts/). In other words, it will cost less than $5,000 to sequence an entire human genome with 30× coverage. This cost keeps dropping every few months due to new developments in sequencing technology. However, a clear shortcoming of NGS versus Sanger technologies has been data quality. The read lengths are much shorter and the error rate is higher as compared to Sanger sequencing. For instance, the read length from Illumina sequencing platforms ranges from 50 base pairs (bps) to 300 bps, making subsequent genome assembly extremely difficult, especially for genomes with a large proportion of repetitive elements/substrings. The error-rates of latest Illumina sequencing platforms, such as HiSeq 2500 are less than 1%, and the errors are highly non-uniformly distributed along the sequenced reads: the terminal 20% of nucleotides have orders of magnitude higher error-rates than the remaining 80% of initial bases.

The first NGS platform was introduced by 454 Life Sciences (acquired by Roche in 2007). Although Roche will shut down 454 in 2016, 454 platforms have made significant contributions to both NGS technology development and biological applications, including the first full genome of a human individual using NGS [67]. The 454 platform utilizes pyrosequencing.

Briefly, pyrosequencing operates as follows. DNA samples are first fragmented randomly. Then each fragment is attached to a bead and emulsion PCR is used to make each bead contain many copies of the initial fragment. The sequencing machine contains numerous picoliter-volume wells, each containing a bead. In pyrosequencing, luciferase is used to produce light, initiated by pyrophosphate when a nucleotide is incorporated at each cycle during sequencing. One drawback of 454 sequencing is that multiple incorporation events occur in homopolymers. Therefore, as the length of a homopolymer is reflected by the light intensity, a number of sequencing errors arise in connection with homopolymers. We remark that such errors were accounted for in a number of DNA-storage implementations, even those using other sequencing platforms which typically do not introduce homopolymer errors.

The SOLiD planform, developed by Applied Biosystems (merged with Invitrogen to become Life Technologies in 2008), was introduced in 2007. SOLiD uses sequencing by ligation; i.e., unlike 454, DNA ligase is used instead of polymerase to identify nucleotides. During sequencing, a pool of possible oligonucleotides of a certain length are labeled according to the sequenced position. These oligonucleotides are ligated by DNA ligase for matching sequences. Before sequencing, the DNA is amplified using emulsion PCR. Each of the resulting beads contains single copies of the same DNA molecule. The output of SOLiD is in color space format, an encoded form of the nucleotide sequences with four colors representing 16 combinations of two adjacent bases.

The most frequently used sequencing platform so far has been Illumina. It's sequencing technology was developed by Solexa, which was acquired by Illumina in 2007. The method is mainly based on reversible dye-terminators that allow the identification of nucleotide bases when they are introduced into DNA strands. DNA samples are first randomly fragmented and primers are ligated to both ends of the fragments. They are then attached on the surface of the flow cell and amplified – in a process also known under the name bridge amplification – so that local clonal DNA colonies, called "DNA clusters", are created. To determine each nucleotide base in the fragments, sequencing by synthesis is utilized. A camera takes images of the fluorescently labeled nucleotides to enable base calling. Subsequently, the dye, along with the terminal 3' blocker, is removed from the DNA to allow for the next cycle to begin with multiple iterations.The most frequently encountered errors in Illumina data are simple substitution errors. Much less common are deletion and insertion errors, and there is an indication that sequencing error rates are higher in regions in which there are homopolymers exceeding lengths $15-20$ [68]. Substitution errors arise when nucleotides are incorporated at different positions in the fragments of a cluster during the same cycle. They are also caused by clusters from more than one DNA fragment, resulting in mixed signals during the base calling step. Illumina sequencers have been used in numerous NGS applications, ranging from whole-genome sequencing, whole-exome sequencing, to RNA sequencing, ChIP sequencing and others. The Illumina HiSeq 2500 system can generate up to 2 billion single-end reads (in 250 bp) per flow cell with 8 lanes. The recently announced HiSeq 4000 system can produce up to 5 billion single-end reads per flow cell.

In addition, several other types of sequencing technologies have been developed in recent years, with the Pacific Biosciences (PacBio) single-molecule real time (SMRT) technology and the Oxford Nanopore's nanopore sequencing systems being the most promising ones. In SMRT, no amplification is needed and the sequencer observes enzymatic reaction in real time. It is also sometimes referred as "third-generation sequencing" because it does not require any amplification prior to sequencing. The most significant advantage of PacBio data is the much longer read length as compared to other NGS technologies. SMRT can achieve read lengths exceeding 10 Kbases, making it more desirable for finishing genome assemblies. Another advantage is speed – run times are much faster. However, the cost of PacBio sequencing is fairly high, amounting to a few dollars per million base-pairs. Furthermore, SMRT error rates are significantly higher than those of Illumina sequencers and the throughput is much lower as well.

Oxford Nanopore is considered another third-generation technology. Its approach is based on the readout from eletrical signals when a single-stranded DNA sequence passes through a nanoscale hole made from proteins or synthetic materials. The DNA passing through the nanopore would change its ion current, allowing the sequencing process to recognize nucleotide bases. Oxford Nanopore has developed a hand-held device called MinION, which has been available to early users. MinION can generate more than 150 million bases per run. However, the error rate is significantly higher than other technologies and it is still being improved. Some of the errors were identified in [10] as *asymmetric errors*, caused by two bases creating highly similar current impulse responses.

Significant challenges of NGS still remain, in particular data analysis problems arising due to short read length. One major step after having the sequencing reads is to assemble reads into longer DNA fragments. Most of these assemblers follow a multi-stage procedure: correcting raw read errors, constructing contigs (i.e. contiguous sequences obtained via overlapping reads), resolving repeats, and connecting contigs into scaffolds using paired-end reads. Most de novo assemblers utilize the de Bruijn graph (DBG) data structure to represent large number of input short reads. EULER [69] pioneered the use of DBG in genome assembly. In recent years, several NGS assemblers (such as Velvet [70], ALLPATHS-LG [71], SOAPdenovo [72], ABySS [73], SGA [74]) have shown promising performances.

V. Archival DNA-Based Storage

A. The Church-Gao-Kosuri Implementation

The first large-scale archival DNA-based storage architecture was implemented and described in the seminal paper of Church *et al.* [1]. In the proposed approach, user data was converted to a DNA sequence via a symbol-by-symbol mapping, encoding each data bit 0 into A or C, and each data bit 1 into T or G. Which of the two bases is used for encoding a particular bit is determined by a runlenghth constraint, i.e., one base is chosen randomly as long as it prohibits homopolymer runs of length greater than three. Furthermore, the choice of one of the two bases enables control of the GC content and secondary structure within the DNA data blocks.

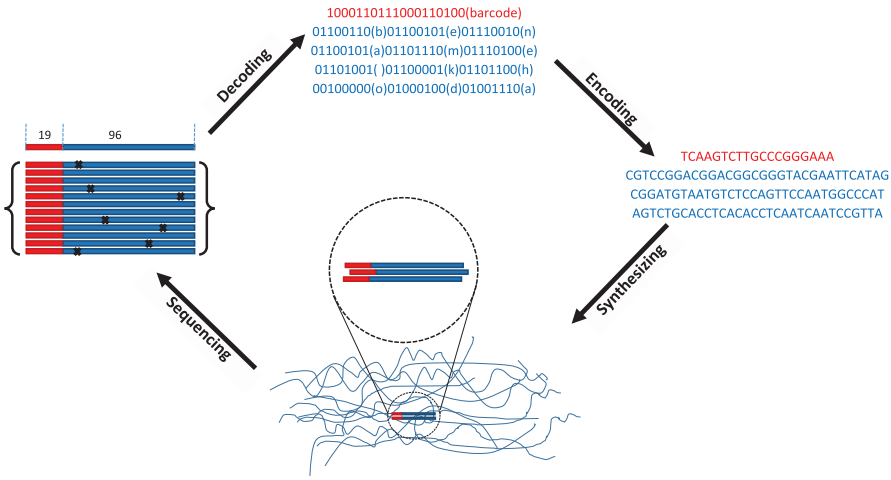

Fig. 6. The Method of Church *et al.* [1]. A chosen text file is converted to ASCII format using 8 bits, for each symbol. Blocks of bits are subsequently encoded into DNA using a 1 bit-per-oligonucleotide encoding. The entire 5.27 Mb html file amounted to 54, 898 oligonucleotides and was synthesized and eluted from a DNA microchip. After amplification – common primer sequences of the blocks are not shown – the library was sequenced using an Illumina platform. Individual reads with the correct barcode and length were screened for consensus, and then converted back into bits comprising the original file.

To illustrate the feasibility of their approach, the authors of [1] encoded in DNA a HTML file of size 5.27 MB. The file included 53, 426 words, 11 JPG images and one Java Script file. In order to eliminate the need for long synthetic DNA strands that are hard to assemble, the file was converted into 54, 898 blocks of length 159 oligonucleotides. Each block contained 96 information oligonucleotides, 19 oligonucleatides for addressing, and 22 oligonucleotides for a common sequence used for amplification and sequencing. The 19 oligonucleotide addresses corresponded to binary encodings of consecutive integers, starting from $00\ldots001$.

The oligonucleotide library was synthesized using Inkjet printed, high-fidelity DNA microchips [38], described in Section II. To encode the data, the library was first amplified by limited-cycle PCR, and then sequenced on a single lane of an Illumina HiSeq system, as described in Section IV. Because synthesis and sequencing errors occurred with low frequency, the DNA blocks were correctly decoded using their own encodings and decoded copies of overlapping blocks. As a result, only 10 bit errors were observed within the 5.27 million encoded bits, i.e., the reported system error rate was less than 2×10^{-6}.

The architecture of the Church-Gao-Kosuri DNA-based encoding system is illustrated in Figure 6.

Encoding example: We provide next an example for the encoding algorithm proposed by Church-Gao-Kosuri [1]. The text of choice is "ferential DN".

- First, each symbol is converted into its 8 bit ASCII format. The encoding results in a binary string of length $12 \times 8 = 96$ of the following form:

$$\overbrace{01100110}^{\text{f}}\overbrace{01100101}^{\text{e}}\overbrace{01110010}^{\text{r}}\overbrace{01100101}^{\text{e}}\overbrace{01101110}^{\text{n}}$$
$$\overbrace{01110100}^{\text{t}}\overbrace{01101001}^{\text{i}}\overbrace{01100001}^{\text{a}}\overbrace{01101100}^{\text{l}}\overbrace{00100000}^{\text{(space)}}$$
$$\overbrace{01000100}^{\text{D}}\overbrace{01001110}^{\text{N}}.$$

- Second, a unique 19 bits barcode is prepended to the binary string for the purpose of DNA block identification: here, we assume that the barcode is 10001101110001 10100. This results in a binary string of length $19 + 96 = 115$, namely:

$$\overbrace{1000110111000110100}^{\text{barcode}}0110011001100101011100100110010101101110011101000110100101100001011000010110110000010000001000100010011110.$$

- Third, every bit 0 is converted into A or C and every bit 1 into T or G. This conversion is performed randomly, while disallowing homopolymer runs of length greater than three. The scheme also asks for balancing the GC content and controlling the secondary structure. For instance, the following DNA code generated from the example binary text satisfies all the aforementioned conditions:

TAACGTCTTGCCCGGAGAAATGAATTCAT
TCATATATGTCAGAATTCATAGCGGATGTA
ATGTCTACGTCTCATAGGCCCATAGTCTG
CCACTACACCATACATAACTCCGTTA.

Finally, two primers of length 22 nucleotide (nt) are added to both ends of the DNA block. The forward primer is CTACACGACGCTCTTCCGATCT, while the backward primer is just the reverse complement of the forward primer, AGATCGGAAGAGCGGTTC AGCA. Hence, the encoded DNA codeword is of length $22 + 115 + 22 = 159$ nt, and reads as:

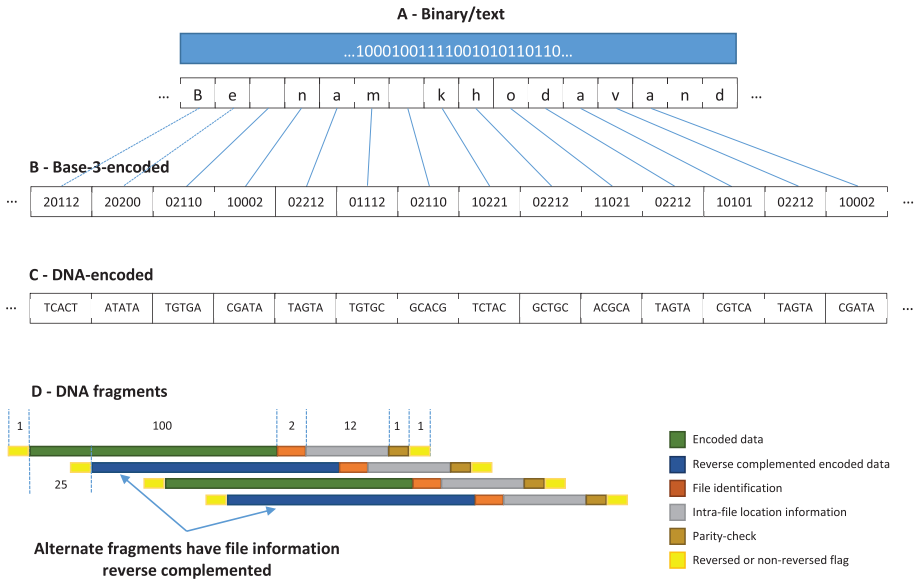

Fig. 7. The Goldman *et al.* Encoding Method. The method uses ASCII and differential coding, Huffman compression, four-fold coverage, reverse complementation of alternate data blocks and single parity-check coding.

forward

CTACACGACGCTCTTCCGATCT TAAC
GTCTTGCCCGGAGAAATGAATTCATTC
ATATATGTCAGAATTCATAGCGGATGTA
ATGTCTACGTCTCATAGGCCCATAGTCT
GCCACTACACCATACATAACTCCGTTA
AGATCGGAAGAGCGGTTCAGCA.

backward

B. The Goldman et al. Method

To encode the digital information into a DNA sequence, Goldman *et al.* [2] started with a binary data set (see Figure 7). The binary file representation was obtained via ASCII encoding, using one byte per symbol (Step A). Each byte was subsequently converted into 5 or 6 ternary digits (trits) via an optimal Huffman code for the underlying distribution of the particular dataset used. The compressed file comprised 5.2×10^6 information bits (Step B). Each trit was then used to select one out of three DNA oligonucleotides differing from the last encoded oligonucleotide. This form of differential coding ensures that there are no homopolymer runs of any length greater than one (Step C). Finally, the resulting DNA string was partitioned into segments of length 100 oligonucleotides, each of which has the property that it overlaps in 75 bases with each adjacent segment (Step D). This overlap ensures $4\times$ coverage for each base. In addition, alternate segments of length 100 were reverse complemented. Indexing information, along with 2 trits for file identification, 12 trits for intra-file location information (which can be used to encode up to 3^{14} unique segment locations), one parity-check and one additional base are appended to both ends to indicate whether the entire fragment was reverse complemented or not. The resulting fragment lengths of the constituent encodings amounted to 153, 335 oligos of length 117.

As an experiment, Goldman *et al.* [2] encoded a digital data file of size 739 KB with an estimated Shannon information of 5.2×10^6 bits into DNA. Their file included all 154 of Shakespear's sonnets (ASCII text), a classic scientific paper (PDF format), a medium-resolution color photograph of the European Bioinformatics Institute (JPEG 2000 format), and a 26-s excerpt from Martin Luther King's 1963 'I have a dream'speech (MP3 format). The encoded strings were synthesized by an updated version of Agilent Technologies. For each sequence, 1.2×10^7 copies were created, with 1 base error per 500 bases, and sequenced on an Illumina HiSeq 2000 system, and decoded successfully. After several postprocessing steps, the original data was decoded with 100% accuracy.

The architecture of the Goldman *et al.* DNA-based encoding system is illustrated in Figure 7.

Encoding example: We present next a short example of the encoding algorithm introduced by Goldman *et al.* [2]. The text to be encoded is "Birney and Goldman".

- First, we apply Huffman coding base 3 to compress the data, resulting in

$$S_1 = \overbrace{20100}^{\text{B}}\overbrace{20210}^{\text{i}}\overbrace{10101}^{\text{r}}\overbrace{00021}^{\text{n}}\overbrace{20001}^{\text{e}}\overbrace{222111}^{\text{y}}\overbrace{02212}^{\text{(space)}}$$
$$\overbrace{01112}^{\text{a}}\overbrace{00021}^{\text{n}}\overbrace{22100}^{\text{d}}\overbrace{02212}^{\text{(space)}}\overbrace{222212}^{\text{G}}\overbrace{02110}^{\text{o}}\overbrace{02101}^{\text{l}}$$
$$\overbrace{22100}^{\text{d}}\overbrace{11021}^{\text{m}}\overbrace{01112}^{\text{a}}\overbrace{00021}^{\text{n}}.$$

- Let $n = len\,(S_1) = 92$, which equals 10102 in base 3. Hence, we set $S_2 = 00000000000000010102$ (an encoding of length 20) and $S_3 = 0000000000000$ (an encoding of length 13). Therefore,

$S_4 = S_1 S_3 S_2 = 2010020210101010002120001222111021212011120002122100022122222120211002101221001102101112000210000000000000000000000000000010102,$

of total length $92 + 13 + 20 = 125$.

- Applying differential coding to S_5 according to the table

	text		
previous	0	1	2
A	C	G	T
C	G	T	A
G	T	A	C
T	A	C	G

results in an encoding of S_4 with the initial state A that reads as

$S_5 =$ TAGTATATCGACTAGTACAGCGTAGCATCTCGCAGCGAGATACGCTGCTACGCAGCATGCTGTGAGTATCGATGACGAGTGACTCTGTACAGTACGTACGTACGTACGTACGTACGTACGACTAT.

- Since $len\,(S_5) = 125$, there are two DNA blocks F_0 and F_1 of length 100 overlapping in exactly 75 bps, i.e.,

$F_0 =$ TAGTATATCGACTAGTACAGCGTAGCATCTCGCAGCGAGATACGCTGCTACGCAGCATGCTGTGAGTATCGATGACGAGTGACTCTGTACAGTACGTACG

and

$F_1 =$ CATCTCGCAGCGAGATACGCTGCTACGCAGCATGCTGTGAGTATCGATGACGAGTGACTCTGTACAGTACGTACGTACGTACGTACGTACGACTAT.

Moreover, the odd-numbered DNA blocks are reverse complemented so that

$F_1 =$ ATAGTCGTACGTACGTACGTACGTACGTACGTACTGTACAGAGTCACTCGTCATCGATACTCACAGCATGCTGCGTAGCAGCGTATCTCGCTGCGAGATG.

- The file identification for the text equals 12. This gives $ID_0 = ID_1 = 12$. The 12 trits intra-file location for F_0 equals $intra_0 = 000000000000$ and for F_1, it equals $intra_1 = 000000000001$. The parity check P_i for block F_i is the sum of the bits at odd locations in $ID_i intra_i$ taken mod 3. Thus, $P_0 = P_1 = 1 + 0 + 0 + 0 + 0 + 0 + 0 \overset{\text{mod}3}{\equiv} 1$. By appending $IX_i = ID_i intra_i P_i$ to F_i we get,

$F_0' =$ TAGTATATCGACTAGTACAGCGTAGCATCTCGCAGCGAGATACGCTGCTACGCAGCATGCTGTGAGTATCGATGACGAGTGACTCTGTACAGTACGTACG AT ACGTACGTACGT C

and

$F_1' =$ ATAGTCGTACGTACGTACGTACGTACGTACGTACTGTACAGAGTCACTCGTCATCGATACTCACAGCATGCTGCGTAGCAGCGTATCTCGCTGCGAGATG AT ACGTACGTACG A G.

- In the last step, we prepend A or T and append C or G to even and odd blocks, respectively. The resulting DNA codewords equals

$F_0'' =$ A TAGTATATCGACTAGTACAGCGTAGCATCTCGCAGCGAGATACGCTGCTACGCAGCATGCTGTGAGTATCGATGACGAGTGACTCTGTACAGTACGTACGATACGTACGTACG TC G

and

$F_1'' =$ T ATAGTCGTACGTACGTACGTACGTACGTACGTACTGTACAGAGTCACTCGTCATCGATACTCACAGCATGCTGCGTAGCAGCGTATCTCGCTGCGAGATGATACGTACGTACGAG C.

C. The Grass et al. Method

As it is apparent from the previous exposition, the Church-Gao-Kosuri and Goldman *et al.* methods did not implement error-correction schemes that go beyond single parity-check coding of fragments. Additional error-correction was accomplished via four-fold coverage. Nevertheless, with the relatively low synthesis and sequencing accuracies of the proposed platforms, the lack of advanced error-correction solutions may be a significant disadvantage. Furthermore, additional errors may arise due to "aging" of the media, as there are no best practices for physically storing the DNA strings to maximize their stability over long periods of time.

In [3], the authors addressed both these issues by implementing a specialized error-correcting scheme and by outlining best practices for DNA media maintainance. Their experiments show that by only combining these two approaches, one should be able to store and recover information encoded in the DNA from the Global Seed Vault (at 18 8C) for hundreds of thousands of years.

The steps applied in [3] for encoding text onto DNA include:

- *Grouping*: Every two characters are mapped to three elements in $\mathbb{F}\,(47)$, the finite field of size 47, via base conversion from 256^2 to 47^3. This results in B information arrays of dimension $m \times k$ information blocks, with elements in $\mathbb{F}\,(47)$. The information arrays are denoted by M_b, with $b \in \{1, \ldots, B\}$ (see Figure 8 for the notation and for an illustration of the grouping). Hence, each block M_b corresponds to a vector of length k, with elements in $\mathbb{F}\,(47^m)$.

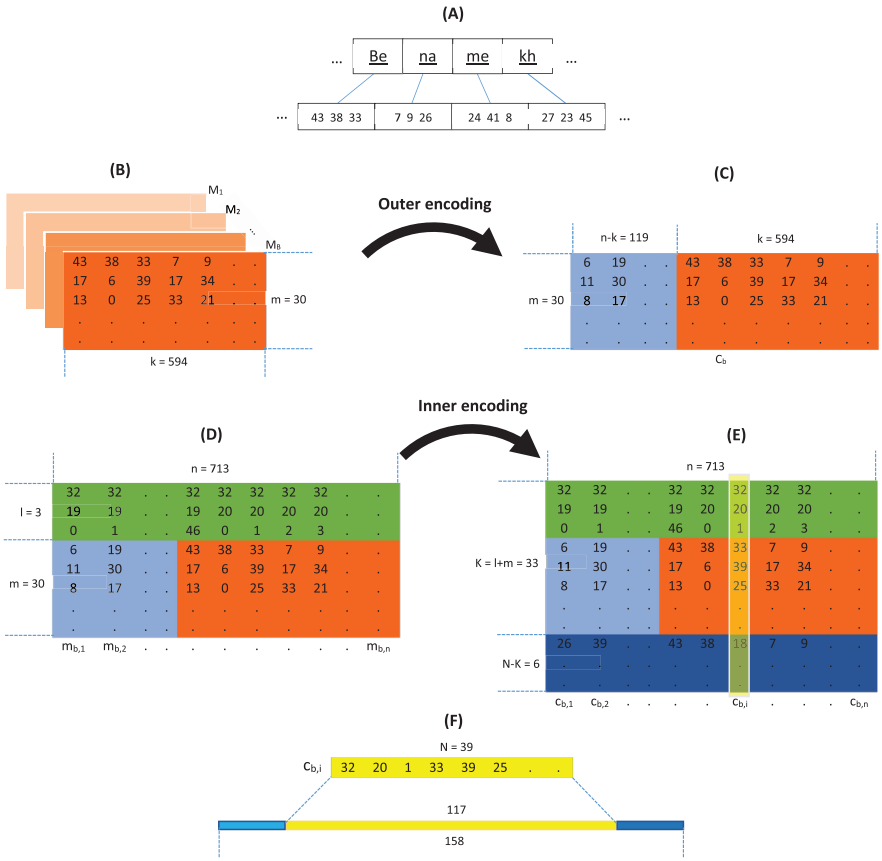

Fig. 8. The Grass *et al.* Method. The method involves DNA text conversion, arraying (grouping) and encoding method.

- *Outer Encoding*: Each block M_b of size $m \times k$ is encoded row by row using a (n, k) Reed-Solomon (RS) code over $\mathbb{F}(47)$ to obtain a codeword block C_b of size $m \times n$, where n and k are the lengths of codewords and information blocks, respectively. To uniquely identify each column in C_b, one has to append l elements in $\mathbb{F}(47)$ to each column. This produces vectors $m_{b,1}, \ldots, m_{b,n}$ of length $K = l + m$ each.
- *Inner Encoding*: Each vector $m_{b,i}$ is mapped to a vector of length N by using another (N, K) RS code over $\mathbb{F}(47)$ to obtain the codewords $c_{b,i}$.
- *Mapping to DNA Strings*: Each element in $c_{b,i}$ is converted to a DNA string of length 3 so that no homopolymers of length three or longer appear. This process results in a DNA strings of length $3N$. To complete the mapping and encoding, two fixed primers are attached to both ends of each created DNA string and used for rapid sequencing.

To experimentally test their method, the authors started with 83 KB of uncompressed text containing the Swiss Federal Charter from 1291 and the English translation of the Methods of Archimedes. This information was encoded into 4991 DNA oligos of length 158. Each of the oligostrings comprised 117 "information" nucleotides. The sequences were synthesized using the CustomArray electrochemical microarray technology described in the previous sections, with a total price of 2, 500 USD. In the process of information retrieval, custom PCR was combined with sequencing on the Illumina MiSeq platform.

The individual decay rates of different DNA strands are mostly influenced by the storage temperature and the water concentration of the DNA storage environment. Four different dry storage technologies for DNA were tested: pure solid-state DNA, DNA on a Whatman FTA filter card, DNA on a biopolymeric storage matrix and DNA encapsulated in silica. Among the tested methods, DNA encapsulated in silica appears to offer the most durable storage format, as silica has the lowest water concentration and it separates DNA molecules from the environment through an inorganic layer. Therefore, the quality of preservation is not affected by environmental humidity, which is important since unlike low temperature (e.g. permafrost) and absence of light, humidity is relatively hard to control. DNA storage systems within silica substrates have the further advantages of exceptional stability against oxidation and photoresistance, provided that an additional titania layer is added to silica.

VI. Random Access and Rewritable DNA-Based Storage

Although the techniques described in [1]–[3] provided a number of solutions for DNA storage, they did not address one important issue: accurate partial and random access to data. In all the cited methods, one has to reconstruct the whole text in order to read or retrieve the information encoded even in a few bases, as the addressing methods used only allow for determining the position of a read in a file, but cannot ensure

precise selection of reads of interest due to potential undesired cross-hybridization between the primers and parts of the information blocks. Moreover, all current designs support read-only storage. Adapting the archival storage solutions to address random access and rewriting appears complicated, due to the storage format that involves reads of length 100 bps shifted by 25 bps so as to ensure four-fold coverage of the sequence. In order to rewrite one base, one needs to selectively access and modify four consecutive reads.

The drawbacks of the archival architectures were addressed in [6], where new coding-theoretic methods were introduced to allow for rewriting and controlled random access.

A. The Yazdi et al. Method

To overcome the aforementioned issues, Yazdi *et al.* [6] developed a new, random-access and rewritable DNA-based storage architecture based on DNA sequences endowed with specialized address strings that may be used for selective information access and encoding with inherent error-correction capabilities. The addresses are designed to be mutually uncorrelated, which means that for a set of addresses $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_M\}$, each of length n, and any two distinct addresses $\mathbf{a}_i, \mathbf{a}_j \in \mathcal{A}$, no prefix of $\mathbf{a}_i$ of length $\leq n-1$ appears as a suffix of $\mathbf{a}_j$.

Information is encoded into DNA blocks of length $L = 2n + ml$. The ith block, B_i, is flanked at both ends by two unique addresses, one of which, say $\mathbf{a}_i$, of length n, is used for encoding. The remainder of the block is divided into m sub-blocks $\text{sub}_{i,1}, \ldots, \text{sub}_{i,m}$, each of length l. Encoding of the block B_i is performed by first dividing the classical digital information stream into m non-overlapping segments and then mapping them to integers $x_1, \ldots, x_m$, respectively. Then, each x_j, for $1 \leq j \leq m$, is encoded into a DNA sub-block $\text{sub}_{i,j}$ of length l using an algorithm, named $\text{ENCODE}_{\mathbf{a}_i,l}(x_j)$, introduced in [6] and described in detail in the next section. The algorithm represents an extension of *prefix-synchronized coding* methods [75] (see Figure 9 for an illustration). Given that the addresses in $\mathcal{A}$ are chosen to be mutually uncorrelated and at large Hamming distance from each other, no a_i appears as a subword in any DNA block, except at one flanking end of the ith block. This feature enables highly sensitive random access and accurate rewriting using the DNA editing techniques described in Section III.

To experimentally test their scheme, Yazdi *et al.* [6] used the introductory pages of five universities retrieved from Wikipedia, amounting to a total size of 17KB in ASCII format. The text was encoded into 32 DNA blocks of length $L = 1000$ bps. To facilitate addressing, they constructed a set of 32 pairs of mutually uncorrelated addresses and used 32 of them for encoding. The addresses used for encoding $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_{32}\}$ were each of length $n = 20$ bps. Different words in the text were counted and tabulated in a dictionary. Each word in the dictionary was converted into a binary sequence of length 21. Groups of six consecutive words in the file were grouped and mapped to binary strings of length $6 \times 21 = 126$. The binary sequences were then translated into DNA sub-blocks of length $l = 80$ bps using $\text{ENCODE}_{(\cdot)}(\cdot)$. Next, $m = 12$ sub-blocks of length 80 bps each were adjoined

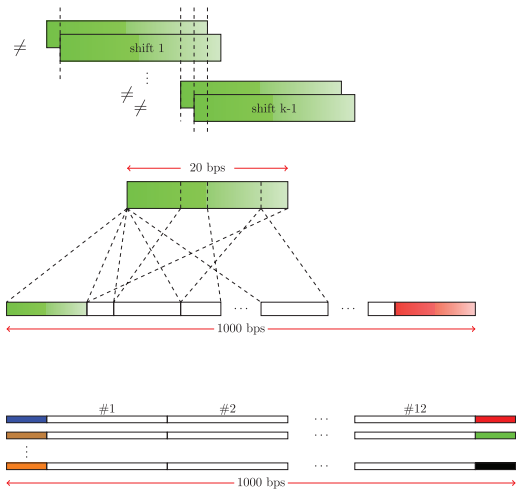

Fig. 9. Data format and encoding for the random access, rewritable architecture of [6].

to form a DNA string of length $12 \times 80 = 960$ bps. To complete the encoding, each string of length 960 bps was equipped with two unique primers of length 20 bps at its ends, forming a DNA block of length $L = 20 + 960 + 20 = 1000$ bps[1]. The resulting DNA sequences were synthesized by IDT [60], at the price of $149 per 1000 bps.

To test the rewriting method, all 32 linear 1000 bps fragments were mixed, and the information in three blocks was rewritten in the DNA encoded domain using both gBlocks and OEPCR editing techniques, described in Section III. The rewritten blocks were selected, amplified and Sanger sequenced to verify that selection and rewriting were performed with 100% accuracy.

Encoding example: We illustrate next the encoding and decoding procedure described in [6] for the short address string $\mathbf{a} = \text{ACCTG}$, which can easily be verified to be self-uncorrelated (i.e., no prefix of the sequence equals a suffix of the sequence). For the sequence of integers $G_{n,1}, G_{n,2}, \ldots, G_{n,7}$, the construction of which will be described in detail in Section VI-D, one can verify that

$$\left(G_{n,1}, G_{n,2}, \ldots, G_{n,7}\right) = (3, 9, 27, 81, 267, 849, 2715)\,.$$

Here, n denotes the length of the address string, which in this case equals five. The algorithm $\text{ENCODE}_{\mathbf{a},8}(550)$ produces

$$\begin{aligned}
550 &= 0 \times G_{5,7} + 550 \\
&\Rightarrow \text{ENCODE}_{\mathbf{a},8}(550) = \underline{\text{C}}\text{ENCODE}_{\mathbf{a},7}(550) \\
550 &= 0 \times G_{5,6} + 550 \\
&\Rightarrow \text{ENCODE}_{\mathbf{a},7}(550) = \underline{\text{C}}\text{ENCODE}_{\mathbf{a},6}(550) \\
550 &= 2 \times G_{5,5} + 0 \times G_{5,4} + 16 \\
&\Rightarrow \text{ENCODE}_{\mathbf{a},6}(550) = \underline{\text{AA}}\text{ENCODE}_{\mathbf{a},4}(16), \\
16 &= 0 \times 3^3 + 1 \times 3^2 + 2 \times 3^1 + 1 \times 3^0 \\
&\Rightarrow \text{ENCODE}_{\mathbf{a},4}(16) = \underline{\text{ATCT}}, \\
&\Rightarrow \text{ENCODE}_{\mathbf{a},8}(550) = \underline{\text{CCAAATCT}}
\end{aligned}$$

[1]Two different addresses were used to terminate one sequence because of DNA synthesis issues, as having one long repeated string at both flaking ends lead to undesired secondary structures.

When running $\text{DECODE}_{\mathbf{a}}(X)$ on the encoded output $X = \underline{\text{CC}}\text{AAATCT}$, the following steps are executed:

$$\begin{aligned}
&\Rightarrow \text{DECODE}_{\mathbf{a}}(\underline{\text{C}}\text{CAAATCT}) = 0 \times G_{5,7} \\
&\quad + \text{DECODE}_{\mathbf{a}}(\text{CAAATCT}) \\
&\Rightarrow \text{DECODE}_{\mathbf{a}}(\underline{\text{C}}\text{AAATCT}) = 0 \times G_{5,6} \\
&\quad + \text{DECODE}_{\mathbf{a}}(\text{AAATCT}), \\
&\Rightarrow \text{DECODE}_{\mathbf{a}}(\underline{\text{AA}}\text{ATCT}) = 2 \times G_{5,5} + 0 \times G_{5,4} \\
&\quad + \text{DECODE}_{\mathbf{a}}(\text{ATCT}) \\
&\Rightarrow \text{DECODE}_{\mathbf{a}}(\underline{\text{ATCT}}) = 16 \\
&\Rightarrow \text{DECODE}_{\mathbf{a}}(\text{CCAAATCT}) = 2 \times \text{G}_{5,5} + 16 = 550
\end{aligned}$$

B. Address Design and Constrained Coding

To encode information on a DNA media, Yazdi *et al.* [6] first designed a set $\mathcal{A}$ of address sequences, each of length n, that satisfies a number of constraints. These constraints make the codewords suitable for selective random access; given the address set $\mathcal{A}$, they also constructed a code $\mathcal{C}_{\mathcal{A}}(\ell)$ of length ℓ and provided efficient methods to encode and decode messages to codewords in $\mathcal{C}_{\mathcal{A}}(\ell)$. In their experiment, Yazdi *et al.* chose $n = 20$ and $\ell = 80$ and stored twelve data subblocks of length 80, each corresponding to the codewords in $\mathcal{C}_{\mathcal{A}}(\ell)$, and flanked these subblocks with two address sequences to obtain a datablock of length 1000 bps.

In Section VI-C, we describe the design constraints for the address sequences and relate these constraints to previously studied concepts such as *running digital sums* and *sequence correlation*. In Section VI-D, we describe the desired properties of $\mathcal{C}_{\mathcal{A}}(\ell)$ and present the encoding schemes developed by Yazdi *et al.* based on *prefix-synchronized schemes* described by Morita *et al.* [76].

C. Constrained Coding for Address Sequences

Constrained coding serves two purposes in the design of address sequences. First, it ensures that DNA patterns prone to sequencing errors are avoided. Second, it allows DNA blocks to be accurately accessed, amplified and selected without perturbing other blocks in the DNA pool. We remark that while these constraints apply to address primer design, they indirectly govern the properties of the fully encoded DNA information blocks. Specifically, we require the address sequences to satisfy the following constraints:

(C1) *Constant GC content (close to 50%) for all the prefixes of the sequences of sufficiently long length*. DNA strands with 50% GC content are more stable than DNA strands with lower or higher GC content and have better coverage during sequencing. Since encoding user information is accomplished via prefix-synchronization, it is important to impose the GC content constraint on the addresses as well as their prefixes, as the latter requirement ensures that all fragments of encoded data blocks are balanced as well. Given $D > 0$, we define a sequence to be *D-GC-prefix-balanced* (D-GCPB) if for all prefixes (including the sequence itself), the difference between the number of G and C bases and the number of A and T bases is at most D. A set of address sequences is D-GCPB if all sequences in the set are D-GCPB.

(C2) *Large mutual Hamming distance*. This reduces the probability of erroneous address selection. Recall that the Hamming distance between two strings of equal length equals the number of positions at which the corresponding symbols disagree. Given $d > 0$, we design our set of sequences such that the Hamming distance between any pair of distinct sequences is at least d.

(C3) *Uncorrelatedness of the addresses*. This imposes the restriction that prefixes of one address do not appear as suffixes of the same or another address. The motivation for this new constraint comes from the fact that addresses are used to provide unique identities for the blocks, and that their substrings should therefore not appear in "similar form" within other addresses. Here, "similarity" is assessed in terms of hybridization affinity. Furthermore, long undesired prefix-suffix matches may lead to assembly errors in blocks during joint sequencing. Most importantly, uncorrelated sequences may be jointly avoided via simple and efficient coding methods. Hence, one can ensure that address sequences only appear at the flanking ends of the blocks and nowhere else in the encoding.

(C4) *Absence of secondary (folding) structure for the address primers*. Such structures may cause errors in the process of PCR amplification and fragment rewriting.

As observed by Yazdi *et al.*, constructing addresses that simultaneously satisfy the constraints C1–C4 and determining bounds on the largest number of such sequences is prohibitively complex [6]. To mitigate this problem, Yazdi *et al.* used a *semi-constructive* address design approach, in which balanced error-correcting codes are designed independently, and subsequently expurgated so as to identify a large set of mutually uncorrelated sequences. The resulting sequences are subsequently tested for secondary structure using *mfold* and *Vienna* [77].

In the same paper, Yazdi *et al.* observed that if one considers the constraints individually or one focuses on certain proper subsets of constraints, it is possible to construct families of codes whose size grow exponentially with code length. To demonstrate this, Yazdi *et al.* borrowed concepts from other areas in coding theory. We provide an overview of these techniques in what follows.

Running Digital Sums. An important criteria for selecting block addresses is to ensure that the corresponding DNA primer sequences have prefixes with a GC content approximately equal to 50%, and that the sequences are at large pairwise Hamming distance. Due to their applications in optical storage, codes that address related issues have been studied in a slightly different form under the name of *bounded running digital sum* (BRDS) codes [78], [79]. A detailed overview of this coding technique may be found in [78].

Fix an integer $D > 0$. A binary sequence $\mathbf{a}$ has a D-bounded running digital sum (D-BRDS) if for any prefix of $\mathbf{a}$ (including $\mathbf{a}$ itself), the number of zeroes and the number of ones differ by at most D. A set $\mathbf{A}$ of binary sequences is D-BRDS if all sequences in $\mathbf{A}$ have D-BRDS. A 1-BRDS set $\mathbf{A}$ with minimum

distance $2d$ may be obtained from a binary code with distance d via the following theorem.

Theorem 1 ([79, Thm 2]): If a binary unrestricted code of length n, size M and minimum distance d exists, then a 1-BRDS set of length $2n$ and minimum distance $2d$ and size M exists.

Hence, it follows from the Gilbert-Varshamov bound that there exists a 1-BRDS set of length $2n$ and minimum distance $2d$ whose size is at least $2^n / \left(\sum_{j=0}^{d-1} \binom{n}{j}\right)$.

A set of DNA sequences over $\{\mathtt{A}, \mathtt{T}, \mathtt{G}, \mathtt{C}\}$ may then be constructed in a straightforward manner by mapping each 0 into one of the bases $\{\mathtt{A}, \mathtt{T}\}$, and 1 into one of the bases $\{\mathtt{G}, \mathtt{C}\}$. In other words, a D-BRDS set of length n and size M yields a D-GCPB set of sequences of size M. For $0 < d \leq n$, $D > 0$, let $M_1(n, d; D)$ denote the maximum size of a D-GCPB set of sequences of length n and minimum distance d. Furthermore, for $q > 0$, let $A_q(n, d)$ denote the maximum size of a q-ary code with minimum distance d.

Applying Theorem 1 and the simple mapping above, we have the following estimates for the size of codes satisfying C1 and C2.

Theorem 2: Fix $0 < d \leq n$, $D = 1$. Then

$$A_2(n/2, d/2) \leq M_1(n, d; 1) \leq A_4(n, d). \tag{1}$$

Sequence Correlation. We describe next the notion of autocorrelation of a sequence and introduce the related notion of mutual correlation of sequences. It was shown in [80] that the autocorrelation function is the crucial mathematical concept for studying sequences avoiding forbidden words (strings) and subwords (substrings). In order to accommodate the need for selective retrieval of a DNA block without accidentally selecting any undesirable blocks, we find it necessary to also introduce the notion of mutually uncorrelated sequences.

Let X and Y be two words, possibly of different lengths, over some alphabet of size $q > 1$. The correlation of X and Y, denoted by $X \circ Y$, is a binary string of the same length as X. The i-th bit (from the left) of $X \circ Y$ is determined by placing Y under X so that the leftmost character of Y is under the i-th character (from the left) of X, and checking whether all characters in the overlapping segments of X and Y are identical. If they are identical, the i-th bit of $X \circ Y$ is set to 1, otherwise, it is set to 0. For example, for $X = \mathtt{GTAGTAG}$ and $Y = \mathtt{TAGTAGCC}$, $X \circ Y = 0100100$, as depicted below.

Note that in general, $X \circ Y \neq Y \circ X$, and that the two correlation vectors may be of different lengths. In the example above, we have $Y \circ X = 00000000$. The autocorrelation of a word X equals $X \circ X$.

In the example below, $X \circ X = 1001001$.

```
X =  G T A G T A G
Y =  T A G T A G C C                0
        T A G T A G C C             1
          T A G T A G C C           0
            T A G T A G C C         0
              T A G T A G C C       1
                T A G T A G C C     0
                  T A G T A G C C   0
```

Definition 1: A sequence X is self-uncorrelated if $X \circ X = 10 \ldots 0$. A set of sequences $\{X_1, X_2, \ldots, X_m\}$ is termed mutually uncorrelated if each sequence is self-uncorrelated and if all pairs of distinct sequences satisfy $X_i \circ X_j = 0 \ldots 0$ and $X_j \circ X_i = 0 \ldots 0$.

The notion of mutual uncorrelatedness may be relaxed by requiring that only sufficiently long prefixes do not match sufficiently long suffixes of other sequences. Sequences with this property, and at sufficiently large Hamming distance, eliminate undesired address cross-hybridization during selection.

Mutually uncorrelated codes were studied by many authors under a variety of names. Levenshtein first introduced them in 1964 under the name 'strongly regular codes' [81], suggesting that the codes are interesting for synchronisation applications. Inspired by the use of distributed sequences in frame synchronisation applications by van Wijngaarden and Willink [82], Bajić and Stojanović [83] recently independently rediscovered mutually uncorrelated codes using the term 'cross-bifix-free' (see also [84]–[86] for recent papers and the references therein). The maximum size of a set of mutually uncorrelated code has been determined up to a constant factor by Blackburn [86]. We state his result below.

Theorem 3: Let $M_2(n)$ be the maximum size of a set of mutually uncorrelated sequences of length n. Then

$$\frac{3 \cdot 4^n}{4en}(1 - o(1)) \leq M_2(n) \leq \frac{4^n}{en}(1 + o(1)).$$

We point to an interesting construction by Bilotta *et al.* [84] and provide a simple modification to obtain a set of sequences satisfying C1 and C3. To do so, we introduce a simple combinatorial object called a *Dyck word*. A Dyck word is a binary string consisting of m zeroes and m ones such that no prefix of the word has more zeroes than ones.

By definition, a Dyck word necessarily starts with a one and ends with a zero. Consider a set $\mathcal{D}$ of Dyck words of length $2m$ and define the following set of words of length $2m + 1$,

$$\mathcal{A} \triangleq \{1\mathbf{a} : \mathbf{a} \in \mathcal{D}\}.$$

Bilotta *et al.* demonstrated that $\mathcal{A}$ is a mutually uncorrelated set of sequences.

A Dyck word has *height* at most D if for any prefix of the word, the difference between the number of ones and the number of zeroes is at most D. In other words, a Dyck word has height at most D if it has D-BRDS. Let $\mathrm{Dyck}(m, D)$ denote the number of Dyck words of length $2m$ and height at most D. de Bruijn *et al.* [87] proved that for fixed values of D,

$$\mathrm{Dyck}(m, D) \sim \frac{4^m}{D+1} \tan^2\left(\frac{\pi}{D+1}\right) \cos^{2m}\left(\frac{\pi}{D+1}\right). \tag{2}$$

Here, $f(m) \sim g(m)$ means that $\lim_{m \to \infty} f(m)/g(m) = 1$.

As with Bilotta *et al.*, we observe that if we prepend Dyck words of length $2m$ and height at most D by 1, we obtain a mutually uncorrelated $(D+1)$-BRDS set of binary words of length $2m + 1$. As before, we map 0 and 1 into $\{\mathtt{A}, \mathtt{T}\}$ and $\{\mathtt{C}, \mathtt{G}\}$, respectively, and obtain a mutually uncorrelated $(D+1)$-GCPB set of sequences.

Theorem 4: Let $M_3(n, D)$ be the maximum size of a mutually uncorrelated D-GCPB set of sequences of length n. If n is odd and $D \geq 2$, then

$$M_3(n, D) \geq \frac{2^{n-1}}{D} \tan^2\left(\frac{\pi}{D}\right) \cos^{n-1}\left(\frac{\pi}{D}\right)(1 + o(1)).$$

As already pointed out, it is an open problem to determine the largest number of address sequences that jointly satisfy the constraints C1 to C4. We conjecture that the number of such sequences is exponential in n, since the number of words that satisfy C1+C2, C3, or C1+C3 separately is exponential (see Theorems 2, 3, 4). Furthermore, the number of words that avoid secondary structures was also shown to be exponentially large by Milenkovic and Nashyap [63].

D. Prefix-Synchronized DNA Codes

Thus far, we described how to construct address sequences that may serve as unique identifiers of the blocks they are associated with. We also pointed out that once such address sequences are identified, user information has to be encoded so as to *avoid* the appearance of any of the addresses, sufficiently long substrings of the addresses, or substrings similar to the addresses in the resulting codewords.

Specifically, for a fixed set $\mathcal{A}$ of address sequences of length n, we define the set $\mathcal{C}_{\mathcal{A}}(\ell)$ to be the set of sequences of length ℓ such that each sequence in $\mathcal{C}_{\mathcal{A}}(\ell)$ does not contain any string belonging to $\mathcal{A}$. Therefore, by definition, when $\ell < n$, the set $\mathcal{C}_{\mathcal{A}}(\ell)$ is simply the set of strings of length ℓ. Our objective is then to design an efficient encoding algorithm (one-to-one mapping) to encode a set $\mathcal{I}$ of messages into $\mathcal{C}_{\mathcal{A}}(\ell)$. For the sake of simplicity, we let $\mathcal{I} = \{0, 1, 2, \ldots, |\mathcal{I}| - 1\}$ and as is usual with constrained coding, we hope to maximize $|\mathcal{I}|$.

Clearly, $|\mathcal{I}| \leq |\mathcal{C}_{\mathcal{A}}(\ell)|$ and hence, it is of interest to determine the size of $\mathcal{C}_{\mathcal{A}}(\ell)$. In the case, when $\mathcal{A}$ is a set of mutually uncorrelated strings, Yazdi *et al.* [6] proved the following theorem.

Theorem 5: Suppose that $\mathcal{A}$ is a set of M mutually uncorrelated sequences of length n over the alphabet $\{\mathtt{A}, \mathtt{T}, \mathtt{C}, \mathtt{G}\}$. Define $F(z) = \sum_{\ell=0}^{\infty} |\mathcal{C}_{\mathcal{A}}(\ell)| z^{\ell}$. Then

$$F(z) = \frac{1}{1 - 4z + Mz^n}. \tag{3}$$

We make certain observations on (3). When M is fixed, it is easy to show that $F(z) = 1/(1 - 4z + Mz^n)$ has only one pole with radius less than one for sufficiently large n. Furthermore, if R^{-1} is the pole of F, we can show that $1/4 < R^{-1} < 1/(4 - \epsilon(n))$ with $\epsilon(n) = o(1)$. Here, the asymptotic is computed with respect to n. In other words, for the case where M is fixed, the size of $\mathcal{C}_{\mathcal{A}}(\ell)$ is at least $(4 - \epsilon(n))^{\ell}(1 - o(1))$ (here, asymptotic is computed with respect to ℓ).

In the case where $\mathcal{A}$ contains a single address $\mathbf{a}$, Morita *et al.* proposed efficient encoding schemes into $\mathcal{C}_{\{\mathbf{a}\}}(\ell)$ in the context of prefix-synchronized codes [76]. Based on the scheme of Morita *et al.*, Yazdi *et al.* developed another encoding method that encodes messages into $\mathcal{C}_{\mathcal{A}}(\ell)$ where $\mathcal{A}$ contains more than one address. In this scheme, Yazdi *et al.* assume that $\mathcal{A}$ is mutually uncorrelated and all sequences in $\mathcal{A}$ end with the same base, which we assume without loss of generality to be $\mathtt{G}$. We then pick an address $\mathbf{a} \triangleq (a_1, a_2, \ldots, a_n) \in \mathcal{A}$ and define the following entities for $1 \leq i \leq n - 1$,

$$\bar{A}_i = \{\mathtt{A}, \mathtt{C}, \mathtt{T}\} \setminus \{a_i\},$$
$$\mathbf{a}^{(i)} = (a_1, a_2, \ldots, a_i).$$

In addition, assume that the elements of $\bar{A}_i$ are arranged in increasing order, say using the lexicographical ordering $\mathtt{A} \prec \mathtt{C} \prec \mathtt{T}$. We subsequently use $\bar{a}_{i,j}$ to denote the j-th smallest element in A_i, for $1 \leq j \leq \left|\bar{A}_i\right|$. For example, if $\bar{A}_i = \{\mathtt{C}, \mathtt{T}\}$, then $\bar{a}_{i,1} = \mathtt{C}$ and $\bar{a}_{i,2} = \mathtt{T}$.

Next, we define a sequence of integers $G_{n,1}, G_{n,2}, \ldots$ that satisfies the following recursive formula

$$G_{n,\ell} = \begin{cases} 3^{\ell}, & 1 \leq \ell < n, \\ \sum_{i=1}^{n-1} \left|\bar{A}_i\right| G_{n,\ell-i}, & \ell \geq n. \end{cases}$$

For an integer $\ell \geq 0$ and $y < 3^{\ell}$, let $\theta_{\ell}(y) = \{\mathtt{A}, \mathtt{T}, \mathtt{C}\}^{\ell}$ be a length-ℓ ternary representation of y. Conversely, for each $W \in \{\mathtt{A}, \mathtt{T}, \mathtt{C}\}^{\ell}$, let $\theta^{-1}(W)$ be the integer y such that $\theta_{\ell}(y) = W$. We proceed to describe how to map every integer $\{0, 1, \ldots, G_{n,\ell} - 1\}$ into a sequence of length ℓ in $\mathcal{C}_{\mathcal{A}}(\ell)$ and vice versa. We denote these functions as $\textsc{Encode}_{\mathbf{a},\ell}$ and $\textsc{Decode}$, respectively.

The steps of the encoding and decoding procedures are listed in Algorithm 1 and the correctness of the algorithm was demonstrated by Yazdi *et al.* [6].

Theorem 6: Let $\mathcal{A}$ be a set of mutually uncorrelated sequences that ends with the same base. Then $\textsc{Encode}_{\mathbf{a},\ell}$ is a one-to-one map from $\{0, 1, \ldots, G_{n,\ell} - 1\}$ to $\mathcal{C}_{\mathcal{A}}(\ell)$ and for all $x \in \{0, 1, \ldots, G_{n,\ell} - 1\}$, $\textsc{Decode}_{\mathbf{a}}(\textsc{Encode}_{\mathbf{a},\ell}(x)) = x$.

In their experiment, Yazdi *et al.* found a set $\mathcal{A}$ of $M = 32$ address sequences of length $n = 20$ and used this method to encode information into $\mathcal{C}_{\mathcal{A}}(\ell = 80)$. In this instance, the value of $G_{20,80} = 1.56 \times 10^{38} \geq 126$ bits, while the size of $\mathcal{C}_{\mathcal{A}}(80)$ is $1.462 \times 10^{48} \geq 159$ bits.

The previously described $\textsc{Encode}_{\mathbf{a},\ell}(x)$ algorithm imposes no limitations on the length of a prefix used for encoding. This feature may lead to unwanted cross hybridization between address primers used for selection and the prefixes of addresses encoding the information. One approach to mitigate this problem is to "perturb" long prefixes in the encoded information in a controlled manner. For small-scale random access/rewriting experiments, the recommended approach is to first select all prefixes of length greater than some predefined threshold. Afterwards, the first and last quarter of the bases of these long prefixes are used unchanged while the central portion of the prefix string is cyclically shifted by half of its length.

For example, for the address primer $\mathbf{a} = \mathtt{ACTAACTG}$ $\mathtt{TGCGACTGATGC}$, if the prefix $\mathbf{a}^{(16)} = \mathtt{ACTAACTGT}$ $\mathtt{GCGACTG}$ appears as a subword, say $\mathbf{p}$, in $X = \textsc{Encode}_{\mathbf{a},\ell}(x)$ then X is modified to X' by mapping $\mathbf{p}$ to

Algorithm 1. Encoding and decoding

$X = \text{ENCODE}_{\mathbf{a},\ell}(x)$
begin
1 if $(\ell \geq n)$
2 $t \leftarrow 1$;
3 $y \leftarrow x$;
4 while $\left(y \geq \left|\bar{A}_t\right| G_{n,\ell-t}\right)$
5 $y \leftarrow y - \left|\bar{A}_t\right| G_{n,\ell-t}$;
6 $t \leftarrow t + 1$;
7 end;
8 $a \leftarrow \lfloor y/G_{n,\ell-t} \rfloor$;
9 $b \leftarrow y \bmod G_{n,\ell-t}$;
10 return $\mathbf{a}^{(t-1)}\bar{a}_{t,a+1}$ $\text{ENCODE}_{\mathbf{a},\ell-t}(b)$;
11 else
12 return $\theta_\ell(y)$;
13 end;
end;

$x = \text{DECODE}_{\mathbf{a}}(X)$
begin
1 $\ell = \text{length}(X)$;
2 $X = X_1 X_2 \ldots X_\ell$;
3 if $(\ell < n)$
4 return $\theta^{-1}(X)$;
5 else
6 find (s,t) such that $\mathbf{a}^{(t-1)}\bar{a}_{t,s} = X_1 \ldots X_t$;
7 return $\left(\sum_{i=1}^{t-1} |\bar{A}_i| G_{n,\ell-i}\right) + (s-1)G_{n,\ell-t} + \text{DECODE}_{\mathbf{a}}(X_{t+1} \ldots X_\ell)$;
8 end;
end;

$\mathbf{p}' = \texttt{ACTAATGCCTGGACTG}$. This process of shifting is illustrated below:

$$X = \ldots \overbrace{\texttt{ACTGT}\underbrace{\texttt{GCGACT}}_{\text{cyclically shift by 3}}\texttt{GATGC}}^{\mathbf{p}} \ldots$$
$$\Downarrow \qquad\qquad \Downarrow$$
$$X' = \ldots \underbrace{\texttt{ACTGT}\overbrace{\texttt{ACTGCG}}\texttt{GATGC}}_{\mathbf{p}'} \ldots$$

For an arbitrary choice of the addresses, this scheme may not allow for unique decoding $\text{ENCODE}_{\mathbf{a},\ell}(x)$. However, there exist simple conditions that can be checked to eliminate primers that do not allow this transform to be "unique". Given the address primers created for our random access/rewriting experiments, we were able to uniquely map each modified prefix to its original prefix and therefore uniquely decode the readouts.

As a final remark, we would like to point out that prefix-synchronized coding also supports error detection and limited error-correction. Error-correction is achieved by checking if each substring of the sequence represents a prefix or "shifted" prefix of the given address sequence and making proper changes when needed.

E. Error-Control Coding for DNA Storage

Based on the discussion of error mechanisms in DNA synthesis and sequencing, it is apparent that most errors follow into the following categories:

- *Substitution errors introduced during synthesis.* These errors may be addressed using many classical coding schemes, such as Reed-Solomon and Low-Density Parity-Check coding methods [7]. One nontrivial problem associated with substitution errors introduced during the synthesis phase arises after high-throughput sequencing. In this case, errors in the synthesized sequences propagate through a number of reads produced during sequencing, and hence correspond to a previously unknown class of burst errors. The authors addressed this issue in a companion paper [8], [9], where they introduced the notion of *DNA profile codes*, which have the property that they can correct combinations of sequencing and synthesis errors in reads, in addition to missing coverage (i.e., missing read errors).
- *Single deletion errors introduced during synthesis.* Isolated single deletion errors may be corrected by using Levenshtein-Tenengolts codes [88], directly encoded into the DNA string. It also appears possible to extend the DNA profile coding paradigm to encompass deletion and insertion errors incurred during synthesis, although no results in this directions were reported.
- *Substitution and coverage errors introduced during sequencing.* These errors may be handled in a similar manner as substitution errors introduced during synthesis, provided that they are used with the correct sequencing platform (i.e., Illumina). For the third generation sequencing platforms - PacBio and Oxford Nanopore - only one specialized error-correction procedure was reported so far [10], addressing problems arising due to overlapping impulse responses of two out of four bases (see Figure 10).

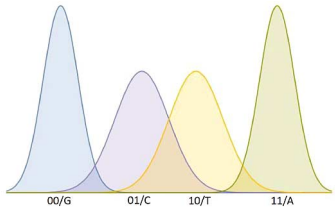

Fig. 10. Impulse response of prototypical solid state nanopore sequencers.

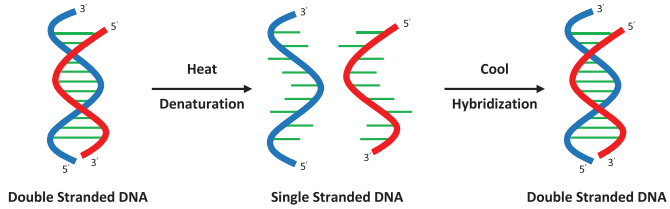

Fig. 11. Principles of DNA Denaturation and Hybridization.

It remains an open problem to design codes that efficiently combine all the constraints imposed by address design considerations and at the same provide robustness to both synthesis and sequencing errors.

APPENDIX

- **Bases** A, T, G **and** C: Nucleotides, the building units of DNA, include one out of four possible bases, A (adenine), G (guanine), C (cytosine), and T (thymine). With a slight abuse of meaning, we alternatively use the terms nucleotides and bases, and express DNA sequence lengths in nucleotides or base pairs, as formally defined next.
- **Base pairs**: A measure of length of DNA using the number of nucleotide pairs.

- **Capillary Electrophoresis**: Capillary electrophoresis is a technique that separates ions based on their electrophoretic mobility, observed when applying a controlled voltage.
- **Clone**: A section of DNA that has been inserted into a vector molecule, such as a plasmid, and then replicated to form many identical copies.
- **Coverage (of a sequencing experiment)**: The average number of reads that contains a base at a particular position in the DNA string to be sequenced. Simply put, $n\times$coverage implies that on average, each base in the DNA string is "observed" n times.
- **De novo**: From scratch, without a template, anew.
- **Deoxinucleotides**: Components of DNA, containing the phosphate, sugar and organic base; when in the triphosphate form, they are the precursors required by DNA polymerase for DNA synthesis (i.e., ATP, CTP, GTP, TTP).
- **DNA microarray**: A DNA microarray (also commonly known as DNA chip or biochip) is a collection of microscopic DNA spots containing relatively short DNA fragments termed probes, attached to a solid surface.
- **DNA Hybridization**: DNA Hybridization is the process of combining two complementary (in the Watson-Crick sense) single-stranded DNA or RNA molecules and allowing them to form a single double-stranded molecule through base pairing (see Figure 11).
- **Dye-terminators**: Labeled versions of dideoxyribonucleotide triphosphates (ddNTPs), "defective" nucleotides used in Sanger sequencing.
- **Enzyme**: Enzymes are biological molecules (proteins) that accelerate, or catalyze, chemical reactions.
- **GC-content**: GC-content is the percentage of nitrogenous bases on a DNA molecule that are either G (guanine) or G (cytosine).
- **Heteroduplex**: A heteroduplex is a double-stranded (duplex) molecule of nucleic acid originated through the genetic recombination of single complementary strands derived from different sources, such as from different homologous chromosomes or even from different organisms.
- **Homologs**: Two chromosomes or fragments from chromosomes from a particular pair, containing the same genetic loci in the same order.
- **Homopolymers**: Sequences of identical bases in DNA strings.
- **In vivo recombination**: Recombination is the process of combining genetic (DNA) material from multiple sources to create new sequences. In vivo recombination refers to recombination performed inside a living cell (in vivo).
- **Ligase**: An enzyme that catalyzes the process of joining two molecules through the formation of new chemical bonds.
- **Luciferase**: An oxidative enzyme used to provide luminescence in natural or controlled biological environments.
- **Nucleotide**: A nucleotide is one of the structural units of DNA and RNA. It comprises a sugar, phosphate group and a base.
- **Oligonucleotide (short strand of nucleotides)**: A relatively short sequence of nucleotides, usually synthesized to match a region where a mutation is known to occur.
- **Polymerase chain reaction (PCR)**: Polymerase chain reaction (PCR) is a laboratory technique used to amplify DNA sequences. The method involves using short DNA sequences called primers to select the portion of the genome to be amplified. The temperature of the sample is repeatedly raised and lowered to help a DNA replication enzyme copy the target DNA sequence. The technique can produce a billion copies of the target sequence in just a few hours.
- **Primer**: A primer is a strand of short nucleic acid sequences that serves as a starting point for DNA synthesis.
- **Protein**: Proteins are large biological molecules, or macromolecules, consisting of one or more long chains of amino acid residues.
- **Read**: DNA fragmet created during the sequencing process.
- **Sequence assembly**: Sequence assembly refers to aligning and merging fragments of a much longer DNA sequence in order to reconstruct the original sequence.
- **Symmetric dimer**: A chemical structure formed from two symmetric units.

REFERENCES

[1] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, vol. 337, no. 6102, p. 1628, 2012.

[2] N. Goldman *et al.*, "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, no. 7435, pp. 77–80, 2013.

[3] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark, "Robust chemical preservation of digital information on DNA in silica with error-correcting codes," *Angew. Chem. Int. Ed.*, vol. 54, no. 8, pp. 2552–2555, 2015.

[4] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, 1960.

[5] S. C. Schuster, "Next-generation sequencing transforms today's biology," Nat. Methods, vol. 5, no. 1, pp. 16–18, 2008.

[6] S. Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic, "A rewritable, random-access DNA-based storage system," *Sci. Rep.*, vol. 5, no. 14138, 2015, doi: 10.1038/srep14138.

[7] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. IT-8, no. 1, pp. 21–28, Jan. 1962.

[8] H. M. Kiah, G. J. Puleo, and O. Milenkovic, "Codes for DNA sequence profiles," in *Proc. IEEE Int. Symp. Inf. Theory*, 2015, pp. 814–818.

[9] H. M. Kiah, G. J. Puleo, and O. Milenkovic, "Codes for DNA storage channels," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2014, pp. 1–5.

[10] R. Gabrys, H. M. Kiah, and O. Milenkovic, "Asymmetric Lee distance codes for DNA-based storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2015, pp. 909–913.

[11] S. Kosuri and G. M. Church, "Large-scale de novo DNA synthesis: Technologies and applications," *Nat. Methods*, vol. 11, no. 5, pp. 499–507, 2014.

[12] J. Tian, K. Ma, and I. Saaem, "Advancing high-throughput gene synthesis technology," *Mol. BioSyst.*, vol. 5, no. 7, pp. 714–722, 2009.

[13] S. Ma, N. Tang, and J. Tian, "DNA synthesis, assembly and applications in synthetic biology," *Curr. Opin. Chem. Biol.*, vol. 16, no. 3, pp. 260–267, 2012.

[14] S. Ma, I. Saaem, and J. Tian, "Error correction in gene synthesis technology," *Trends Biotechnol.*, vol. 30, no. 3, pp. 147–154, 2012.

[15] A. Michelson and A. R. Todd, "Nucleotides—Part XXXII: Synthesis of a dithymidine dinucleotide containing a 3': 5'-internucleotidic linkage," *J. Chem. Soc.*, pp. 2632–2638, 1955, doi: 10.1039/JR9550002632.

[16] R. Hall, A. Todd, and R. Webb, "644. Nucleotides—Part XLI: Mixed anhydrides as intermediates in the synthesis of dinucleoside phosphates," *J. Chem. Soc.*, pp. 3291–3296, 1957, doi: 10.1039/JR9570003291.

[17] P. T. Gilham and H. G. Khorana, "Studies on polynucleotides—Part I: A new and general method for the chemical synthesis of the C5 internucleotidic linkage. Syntheses of deoxyribo-dinucleotides1," *J. Amer. Chem. Soc.*, vol. 80, no. 23, pp. 6212–6222, 1958.

[18] S. Roy and M. Caruthers, "Synthesis of DNA/RNA and their analogs via phosphoramidite and H-phosphonate chemistries," *Molecules*, vol. 18, no. 11, pp. 14268–14284, 2013.

[19] C. B. Reese, "Oligo-and poly-nucleotides: 50 years of chemical synthesis," *Org. Biomol. Chem.*, vol. 3, no. 21, pp. 3851–3868, 2005.

[20] B. C. Froehler, P. G. Ng, and M. D. Matteucci, "Synthesis of DNA via deoxynudeoside H-phosphonate intermediates," *Nucleic Acids Res.*, vol. 14, no. 13, pp. 5399–5407, 1986.

[21] P. J. Garegg, I. Lindh, T. Regberg, J. Stawinski, R. Strömberg, and C. Henrichson, "Nucleoside H-phosphonates—Part III: Chemical synthesis of oligodeoxyribonucleotides by the hydrogenphosphonate approach," *Tetrahedron Lett.*, vol. 27, no. 34, pp. 4051–4054, 1986.

[22] H. Khorana, W. Razzell, P. Gilham, G. Tener, and E. Pol, "Syntheses of dideoxyribonucleotides," *J. Amer. Chem. Soc.*, vol. 79, no. 4, pp. 1002–1003, 1957.

[23] R. L. Letsinger and V. Mahadevan, "Oligonucleotide synthesis on a polymer support1, 2," *J. Amer. Chem. Soc.*, vol. 87, no. 15, pp. 3526–3527, 1965.

[24] R. L. Letsinger and K. K. Ogilvie, "Nucleotide chemistry—Part XIII: Synthesis of oligothymidylates via phosphotriester intermediates," *J. Amer. Chem. Soc.*, vol. 91, no. 12, pp. 3350–3355, 1969.

[25] C. Reese and R. Saffhill, "Oligonucleotide synthesis via phosphotriester intermediates: The phenyl-protecting group," *Chem. Commun.*, no. 13, pp. 767–768, 1968, doi: 10.1039/C19680000767.

[26] R. L. Letsinger and W. B. Lunsford, "Synthesis of thymidine oligonucleotides by phosphite triester intermediates," *J. Amer. Chem. Soc.*, vol. 98, no. 12, pp. 3655–3661, 1976.

[27] S. Beaucage and M. Caruthers, "Deoxynucleoside phosphoramidites—A new class of key intermediates for deoxypolynucleotide synthesis," *Tetrahedron Lett.*, vol. 22, no. 20, pp. 1859–1862, 1981.

[28] N. Sinha, J. Biernat, J. McManus, and H. Köster, "Polymer support oligonucleotide synthesis—Part XVIII1.2: Use of β-cyanoethyi-N, N-dialkylamino-/N-morpholino phosphoramidite of deoxynucleosides for the synthesis of DNA fragments simplifying deprotection and isolation of the final product," *Nucleic Acids Res.*, vol. 12, no. 11, pp. 4539–4557, 1984.

[29] S. P. Fodor, L. Read, M. Pirrung, L. Stryer, A. T. Lu, and D. Solas, "Light-directed, spatially addressable parallel chemical synthesis," *Science*, vol. 251, p. 251, 1991.

[30] A. C. Pease, D. Solas, E. J. Sullivan, M. T. Cronin, C. P. Holmes, and S. Fodor, "Light-generated oligonucleotide arrays for rapid DNA sequence analysis," *Proc. Nat. Acad. Sci.*, vol. 91, no. 11, pp. 5022–5026, 1994.

[31] X. Gao, E. Gulari, and X. Zhou, "In situ synthesis of oligonucleotide microarrays," *Biopolymers*, vol. 73, no. 5, pp. 579–596, 2004.

[32] T. R. Hughes *et al.*, "Expression profiling using microarrays fabricated by an ink-jet oligonucleotide synthesizer," *Nat. Biotechnol.*, vol. 19, no. 4, pp. 342–347, 2001.

[33] S. Singh-Gasson *et al.*, "Maskless fabrication of light-directed oligonucleotide microarrays using a digital micromirror array," *Nat. Biotechnol.*, vol. 17, no. 10, pp. 974–978, 1999.

[34] E. F. Nuwaysir *et al.*, "Gene expression analysis using oligonucleotide arrays produced by maskless photolithography," *Genome Res.*, vol. 12, no. 11, pp. 1749–1755, 2002.

[35] A. L. Ghindilis *et al.*, "Combimatrix oligonucleotide arrays: Genotyping and gene expression assays employing electrochemical detection," *Biosens. Bioelectron.*, vol. 22, no. 9, pp. 1853–1860, 2007.

[36] D. S. Kong, P. A. Carr, L. Chen, S. Zhang, and J. M. Jacobson, "Parallel gene synthesis in a microfluidic device," *Nucleic Acids Res.*, vol. 35, no. 8, p. e61, 2007.

[37] J. W. Efcavitch and C. Heiner, "Depurination as a yield decreasing mechanism in oligodeoxynucleotide synthesis," *Nucleosides Nucleotides Nucleic Acids*, vol. 4, nos. 1–2, p. 267, 1985.

[38] E. M. LeProust *et al.*, "Synthesis of high-quality libraries of long (150mer) oligonucleotides by a novel depurination controlled process," *Nucleic Acids Res.*, vol. 38, no. 8, pp. 2522–2540, 2010.

[39] L.-C. Au, F.-Y. Yang, W.-J. Yang, S.-H. Lo, and C.-F. Kao, "Gene synthesis by a LCR-based approach: High-level production of leptin-l54 using synthetic gene in Escherichia coli," *Biochem. Biophys. Res. Commun.*, vol. 248, no. 1, pp. 200–203, 1998.

[40] W. P. Stemmer, A. Crameri, K. D. Ha, T. M. Brennan, and H. L. Heyneker, "Single-step assembly of a gene and entire plasmid from large numbers of oligodeoxyribonucleotides," *Gene*, vol. 164, no. 1, pp. 49–53, 1995.

[41] D. G. Gibson, "Synthesis of DNA fragments in yeast by one-step assembly of overlapping oligonucleotides," *Nucleic Acids Res.*, vol. 37, no. 20, pp. 6984–6990, 2009.

[42] D. G. Gibson, H. O. Smith, C. A. Hutchison III, J. C. Venter, and C. Merryman, "Chemical synthesis of the mouse mitochondrial genome," *Nat. Methods*, vol. 7, no. 11, pp. 901–903, 2010.

[43] J. Tian *et al.*, "Accurate multiplex gene synthesis from programmable DNA microchips," *Nature*, vol. 432, no. 7020, pp. 1050–1054, 2004.

[44] A. Y. Borovkov *et al.*, "High-quality gene assembly directly from unpurified mixtures of microarray-synthesized oligonucleotides," *Nucleic Acids Res.*, vol. 38, no. 19, p. e180, 2010.

[45] S. Kosuri *et al.*, "Scalable gene synthesis by selective amplification of DNA pools from high-fidelity microchips," *Nat. Biotechnol.*, vol. 28, no. 12, pp. 1295–1299, 2010.

[46] J. Quan *et al.*, "Parallel on-chip gene synthesis and application to optimization of protein expression," *Nat. Biotechnol.*, vol. 29, no. 5, pp. 449–452, 2011.

[47] P. A. Carr, J. S. Park, Y.-J. Lee, T. Yu, S. Zhang, and J. M. Jacobson, "Protein-mediated error correction for de novo DNA synthesis," *Nucleic Acids Res.*, vol. 32, no. 20, p. e162, 2004.

[48] B. F. Binkowski, K. E. Richmond, J. Kaysen, M. R. Sussman, and P. J. Belshaw, "Correcting errors in synthetic DNA through consensus shuffling," *Nucleic Acids Res.*, vol. 33, no. 6, p. e55, 2005.

[49] W. Wan *et al.*, "Error removal in microchip-synthesized DNA using immobilized muts," *Nucleic Acids Res.*, vol. 42, no. 12, p. gku405, 2014, doi: 10.1093/nar/gku405.

[50] J. Smith and P. Modrich, "Removal of polymerase-produced mutant sequences from PCR products," *Proc. Nat. Acad. Sci.*, vol. 94, no. 13, pp. 6847–6850, 1997.

[51] M. Fuhrmann, W. Oertel, P. Berthold, and P. Hegemann, "Removal of mismatched bases from synthetic genes by enzymatic mismatch cleavage," *Nucleic Acids Res.*, vol. 33, no. 6, p. e58, 2005.

[52] B. J. Till, C. Burtner, L. Comai, and S. Henikoff, "Mismatch cleavage by single-strand specific nucleases," *Nucleic Acids Res.*, vol. 32, no. 8, pp. 2632–2641, 2004.

[53] C. A. Oleykowski, C. R. B. Mullins, A. K. Godwin, and A. T. Yeung, "Mutation detection using a novel plant endonuclease," *Nucleic Acids Res.*, vol. 26, no. 20, pp. 4597–4602, 1998.

[54] I. Saaem, S. Ma, J. Quan, and J. Tian, "Error correction of microchip synthesized genes using surveyor nuclease," *Nucleic Acids Res.*, vol. 40, no. 3, p. gkr887, 2011, doi: 10.1093/nar/gkr887.

[55] P. R. Dormitzer *et al.*, "Synthetic generation of influenza vaccine viruses for rapid response to pandemics," *Sci. Transl. Med.*, vol. 5, no. 185, p. 185ra68, 2013.

[56] M. Matzas *et al.*, "High-fidelity gene synthesis by retrieval of sequence-verified DNA identified using high-throughput pyrosequencing," *Nat. Biotechnol.*, vol. 28, no. 12, pp. 1291–1294, 2010.

[57] H. Lee *et al.*, "A high-throughput optomechanical retrieval method for sequence-verified clonal DNA from the NGS platform," *Nat. Commun.*, vol. 6, no. 6073, 2015, doi: 10.1038/ncomms7073.

[58] H. Kim *et al.*, "'Shotgun DNA synthesis' for the high-throughput construction of large DNA molecules," *Nucleic Acids Res.*, vol. 40, no. 18, p. gks546, 2012, doi: 10.1093/nar/gks546.

[59] J. J. Schwartz, C. Lee, and J. Shendure, "Accurate gene synthesis with tag-directed retrieval of sequence-verified DNA molecules," *Nat. Methods*, vol. 9, no. 9, pp. 913–915, 2012.

[60] H. Packer, "CRISPR and Cas9 for flexible genome editing," Technical report, 2014 [Online]. Available: www.idtdna.com/pages/products/genes/gblocks-gene-fragments/decoded-articles/decoded/2013/12/13/crispr-and-cas9-forflexible-genome-editing.

[61] R. Higuchi, B. Krummel, and R. Saiki, "A general method of in vitro preparation and specific mutagenesis of DNA fragments: Study of protein and DNA interactions," *Nucleic Acids Res.*, vol. 16, no. 15, pp. 7351–7367, 1988.

[62] R. Jansen *et al.*, "Identification of genes that are associated with DNA repeats in prokaryotes," *Mol. Microbiol.*, vol. 43, no. 6, pp. 1565–1575, 2002.

[63] O. Milenkovic and N. Kashyap, "On the design of codes for DNA computing," in *Coding and Cryptography*. New York, NY, USA: Springer, 2006 pp. 100–119.

[64] F. Sanger, S. Nicklen, and A. R. Coulson, "DNA sequencing with chain-terminating inhibitors," *Proc. Nat. Acad. Sci. USA*, vol. 74, no. 12, pp. 5463–5467, Dec. 1977.

[65] E. S. Lander *et al.*, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, no. 6822, pp. 860–921, Feb. 2001.

[66] R. H. Waterston *et al.*, "Initial sequencing and comparative analysis of the mouse genome," *Nature*, vol. 420, no. 6915, pp. 520–562, Dec. 2002.

[67] D. A. Wheeler *et al.*, "The complete genome of an individual by massively parallel DNA sequencing," *Nature*, vol. 452, no. 7189, pp. 872–876, Apr. 2008.
[68] M. G. Ross *et al.*, "Characterizing and measuring bias in sequence data," *Genome Biol.*, vol. 14, no. 5, p. R51, 2013.
[69] P. A. Pevzner, H. Tang, and M. S. Waterman, "An Eulerian path approach to DNA fragment assembly," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 17, pp. 9748–9753, Aug. 2001.
[70] D. R. Zerbino and E. Birney, "Velvet: Algorithms for de novo short read assembly using de Bruijn graphs," *Genome Res.*, vol. 18, no. 5, pp. 821–829, May 2008.
[71] S. Gnerre *et al.*, "High-quality draft assemblies of mammalian genomes from massively parallel sequence data," *Proc. Nat. Acad. Sci. USA*, vol. 108, no. 4, pp. 1513–1518, Jan. 2011.
[72] R. Li *et al.*, "De novo assembly of human genomes with massively parallel short read sequencing," *Genome Res.*, vol. 20, no. 2, pp. 265–272, Feb. 2010.
[73] J. T. Simpson, K. Wong, S. D. Jackman, J. E. Schein, S. J. Jones, and I. Birol, "ABySS: A parallel assembler for short read sequence data," *Genome Res.*, vol. 19, no. 6, pp. 1117–1123, Jun. 2009.
[74] J. T. Simpson and R. Durbin, "Efficient de novo assembly of large genomes using compressed data structures," *Genome Res.*, vol. 22, no. 3, pp. 549–556, Mar. 2012.
[75] E. Gilbert, "Synchronization of binary messages," *IRE Trans. Inf. Theory*, vol. IT-6, no. 4, pp. 470–477, Sep. 1960.
[76] H. Morita, A. J. van Wijngaarden, and A. Han Vinck, "On the construction of maximal prefix-synchronized codes,"*IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 2158–2166, Nov. 1996.
[77] J.-M. Rouillard, M. Zuker, and E. Gulari, "Oligoarray 2.0: Design of oligonucleotide probes for DNA microarrays using a thermodynamic approach," *Nucleic Acids Res.*, vol. 31, no. 12, pp. 3057–3062, 2003.
[78] G. D. Cohen and S. Litsyn, "Dc-constrained error-correcting codes with small running digital sum," *IEEE Trans. Inf. Theory*, vol. 37, no. 3, pp. 949–955, May 1991.
[79] M. Blaum, S. Litsyn, V. Buskens, and H. C. van Tilborg, "Error-correcting codes with bounded running digital sum," *IEEE Trans. Inf. Theory*, vol. 39, no. 1, pp. 216–227, Jan. 1993.
[80] L. J. Guibas and A. M. Odlyzko, "Maximal prefix-synchronized codes," *SIAM J. Appl. Math.*, vol. 35, no. 2, pp. 401–418, 1978.
[81] V. Levenshtein, "Decoding automata, invariant with respect to the initial state," *Problemy Kibernet*, vol. 12, pp. 125–136, 1964.
[82] A. J. De Lind Van Wijngaarden and T. J. Willink, "Frame synchronization using distributed sequences," *IEEE Trans. Commun.*, vol. 48, no. 12, pp. 2127–2138, Dec. 2000.
[83] D. Bajić and J. Stojanović, "Distributed sequences and search process," in *Proc. IEEE Int. Conf. Commun.*, 2004, vol. 1, pp. 514–518.
[84] S. Bilotta, E. Pergola, and R. Pinzani, "A new approach to cross-bifix-free sets," *IEEE Trans. Inf. Theory*, vol. 6, no. 58, pp. 4058–4063, Jun. 2012.
[85] Y. M. Chee, H. M. Kiah, P. Purkayastha, and C. Wang, "Cross-bifix-free codes within a constant factor of optimality," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4668–4674, Jul. 2013.
[86] S. R. Blackburn, "Non-overlapping codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4890–4894, Sep. 2015.
[87] N. de Bruijn, D. Knuth, and S. Rice, "The average height of planted plane trees," *Graph Theory and Computing*, R. C. Read, Ed. Stanford, CA, USA: Stanford University of California Department of Computer Science, 1972, p. 15.
[88] R. Varshamov and G. Tenenholtz, "A code for correcting a single asymmetric error," *Automat. Telemekh.*, vol. 26, no. 2, pp. 288–292, 1965.

S. M. Hossein Tabatabaei Yazdi received the Bachelor's degree in electrical engineering from Sharif University of Technology, Tehran, Iran, and the Master's degree in electrical and computer engineering from Texas A&M university, College Station, TX, USA. He is currently pursuing the Ph.D. degree at the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, IL, USA. His research interests include bioinformatics and constrained coding.

Han Mao Kiah received the Ph.D. degree in mathematics from Nanyang Technological University, Singapore, in 2014. From 2014 to 2015, he was a Postdoctoral Research Associate with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Champaign, IL, USA. Currently, he is a Lecturer with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. His research interests include combinatorial design theory, coding theory, and enumerative combinatorics.

Eva Garcia-Ruiz received the B.Sc degree in biology from the University of Alcala, Alcalá de Henares, Spain, in 2004, and the Ph.D. degree in microbiology from Complutense University, Madrid, Spain, in 2012. From January 2005 to July 2006, she did an internship at the Basic Research Center of Merck, Sharp & Dohme of Spain (MSD), Madrid, Spain. From November 2012 to July 2013, she was a Postdoctoral Research Fellow with the Institute of Catalysis and Petrochemistry, Spanish Council for Scientific Research (CSIC), Spain. Since July 2013, she has been a Postdoctoral Research Associate with the Department of Chemical and Biomolecular Engineering, University of Illinois (UIUC), Champaign, IL, USA. Her research interests include development and application of synthetic biology tools to engineer microorganism for production of added-value chemicals.

Jian Ma received the B.S. and M.S. degrees in computer science from Fudan University, Shanghai, China, in 2000 and 2003, respectively, and the Ph.D. degree in computer science from Pennsylvania State University, State College, PA, USA, in 2006. After his postdoc training with Dr. David Haussler at UC Santa Cruz, Santa Cruz, CA, USA, he joined the Department of Bioengineering, University of Illinois at Urbana-Champaign, Champaign, Il, USA, in 2009, as an Assistant Professor and became an Associate Professor in 2015. He joined the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, as an Associate Professor in January 2016. His research interests include computational genomics.

Huimin Zhao received the B.S. degree in biology from the University of Science and Technology of China, Hefei, China, in 1992, and the Ph.D. degree in chemistry from California Institute of Technology, Pasadena, CA, USA, in 1998. He is the Centennial Endowed Chair Professor of Chemical and Biomolecular Engineering, and a Professor of Chemistry, Biochemistry, Biophysics, and Bioengineering with the University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA. Prior to joining UIUC in 2000, he was a Project Leader with the Industrial Biotechnology Laboratory of the Dow Chemical Company. He has authored and coauthored over 220 research articles and 20 patents. His research interests include the development and applications of synthetic biology tools to address society's most daunting challenges in human health and energy, and in the fundamental aspects of enzyme catalysis and gene regulation.

Olgica Milenkovic (M'04–SM'12) received the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA and is currently a Professor with the Electrical Engineering Department, University of Illinois at Urbana-Champaign, Champaign, IL, USA. In 2012, she was named a Center for Advanced Studies (CAS) Associate, and in 2013 she became a Willet Scholar. Her research interests include bioinformatics, coding theory, compressive sensing, and social sciences. In 2015, she was named the Distinguished Lecturer of the IEEE Information Theory Society. She served on the Editorial Board for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the IEEE TRANSACTIONS ON INFORMATION THEORY. She was a recipient of the NSF CAREER Award, the DARPA Young Faculty Award, and the Dean's Excellence in Research Award.