Towards Migration-Free "Just-In-Case" Data Archival for Future Cloud Data Lakes using Synthetic DNA

Eugenio Marinelli
Eurecom
France
eugenio.marinelli@eurecom.fr

Yiqing Yan
Eurecom
France
yiqing.yan@eurecom.fr

Virginie Magnone
IPMC
France
magnone@ipmc.cnrs.fr

Charlotte Dumargne
IPMC
France
dumargne@ipmc.cnrs.fr

Pascal Barbry
IPMC
France
barbry@ipmc.cnrs.fr

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

Raja Appuswamy
Eurecom
France
raja.appuswamy@eurecom.fr

ABSTRACT

Given the growing adoption of AI, cloud data lakes are facing the need to support cost-effective "just-in-case" data archival over long time periods to meet regulatory compliance requirements. Unfortunately, current media technologies suffer from fundamental issues that will soon, if not already, make cost-effective data archival infeasible. In this paper, we present a vision for redesigning the archival tier of cloud data lakes based on a novel, obsolescence-free storage medium–synthetic DNA. In doing so, we make two contributions: (i) we highlight the challenges in using DNA for data archival and list several open research problems, (ii) we outline OligoArchive-DSM (OA-DSM)–an end-to-end DNA storage pipeline that we are developing to demonstrate the feasibility of our vision.

PVLDB Reference Format:
Eugenio Marinelli, Yiqing Yan, Virginie Magnone, Charlotte Dumargne, Pascal Barbry, Thomas Heinis, and Raja Appuswamy. Towards Migration-Free "Just-In-Case" Data Archival for Future Cloud Data Lakes using Synthetic DNA. PVLDB, 16(8): 1923 - 1929, 2023.
doi:10.14778/3594512.3594522

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://gitlab.eurecom.fr/marinele/oligoarchive-columnar.

1 INTRODUCTION

Over the past few years, driven by the popularity of analytics and AI, we have witnessed a rapid growth in the popularity of cloud data lakes like GCS, ADLS, and S3. Naturally, an enormous amount of effort is being made in advancing various aspects of data management for data lakes, from standardizing open file formats, to supporting transaction updates in addition to "just-in-time" optimizations for BI/AI analytics. However, as these data lakes continue to accumulate more data, it is becoming increasingly more important to focus on yet another task that has historically received much less attention, one related to "just-in-case" data archival.

Enterprises have always had the need to archive data in order to meet "just-in-case" safety, legal and regulatory compliance requirements. Today, over 80% of data stored is archival in nature [20]. Archival data is the fastest growing data segment with over 60% cumulative annual growth rate [33]. As enterprises migrate to the cloud, cloud data lakes will soon be (if not already) in need of archival storage technologies that can provide high-density, low-cost storage of such data for decades without degradation.

Unfortunately, current media technologies are unable to meet these requirements as they have fundamental limitations. First, all media technologies suffer from decay due to a limited lifetime (around 5 years for HDD and 30 years for tape) [13]. Cloud vendors, in contrast, will be tasked with preserving archival data for much longer duration. For instance, a recent survey of enterprise executives that manage Petabyte-sized data lakes have reported archive retention periods of several decades. Second, current media technologies tightly couple the storage medium with the technology to read data off the medium. This coupling leads to *media obsolescence*, with data stored in an older medium no longer being readable by newer readers. For instance, Linear Tape Open (LTO) drives are backwards compatible only with two tape generations for read and one generation for write.

Both aforementioned hardware issues necessitate expensive, cumbersome, periodic remastering, where data is migrated from an older generation of archival media to a newer one. This problem is particularly acute for tape. Although tape media can last a few decades, obsolescence issues caused by LTO compatibility effectively reduce the remastering window to 5–7 years. A recent article summarized the financial impact of such data migration on the movie industry, where several independent productions are no longer being archived on tape due to rising migration costs [30]. Given the rate of growth of cloud data lakes, it is inevitable that cloud vendors will face similar problems in the near future. This has prompted researchers to investigate the use of new storage media that has radically different density and durability characteristics compared to HDD and tape [29, 34]. We believe that time is ripe for the data management community to join this effort and take on the challenges related to "just-in-case" data archival in addition to "just-in-time" data analytic.

In this paper, we make two contributions. First, we present our vision for migration-free data archival for cloud data lakes

Proceedings of the VLDB Endowment, Vol. 16, No. 8 ISSN 2150-8097.
doi:10.14778/3594512.3594522

of the future using a novel, obsolescence-free, biological medium–synthetic DNA. We show how synthetic DNA opens up new avenues of research by highlighting several open challenges that need to be addressed to make long-term data archival durable and cost effective. As our second contribution, we take the first steps towards concretely realizing the aforementioned vision by presenting *OligoArchive-DSM(OA-DSM)*, an end-to-end pipeline for data archival using synthetic DNA. In doing so, we set the stage for a new line of research in "just-in-case" data archival on synthetic DNA by (i) showing how database-inspired techniques can be used in the context of DNA storage to reduce read/write costs compared to other state-of-the-art (SOTA) approaches, and (ii) making the OA-DSM framework and related data publicly available.

2 BACKGROUND & VISION

In order to make long-term data archival cost effective, it is necessary to eliminate the non-scalable, expensive, periodic data migration procedure. Recently, several new initiatives have emerged from both industry and academia in an effort to develop new long-term storage technologies that can overcome the media decay and obsolescence issues faced by contemporary media.

2.1 Obsolescence-Free Storage

Analog medium. Historically, analog media like microfilm and paper have been used by libraries and museums for protecting journals across several decades [8, 31]. More recently, film has been used for the preservation of the Declaration of Children's Rights document in collaboration with the UN in the Arctic World Archive [31]. The advantage of analog media is the longevity. For instance, LE-500 rated microfilms and ISO 9706 rated archival paper, are designed to last 500 years. Some analog media, like paper, also does not have obsolescence issues, as any scanning technology can be used to read data off the medium. However, others still suffer from obsolescence as they require dedicated readers that are customized to the media technology which can become obsolete. A major disadvantage with analog media is its density, as all current analog technologies have density much lower than tape (O(KB-MB) per unit for paper/microfilm and O(GB) per unit for film).

Optical medium. On the optical front, optical discs (like BluRay, Panasonic Archival Disc, etc.) have been used for data archival by cloud-scale systems in production [29]. Recent efforts, like Microsoft Project Silica [2], are pushing the limits of optical storage by using femtosecond lasers to create layers of three-dimensional, nanoscale deformations in quartz glass. Data is read back by shining polarized light through the glass and analyzing the retrieved image to decode back digital data. Albeit being in its nascency, project Silica has demonstrated the feasibility and durability of glass by storing 76GB of data. While optical media provides much higher density than analog media, their resistance to obsolescence is unclear, as they still require dedicated readers to retrieve data.

Biological medium. On the biological front, a medium that has received a lot of attention recently is synthetic Deoxyribo Nucleic Acid (DNA). DNA is a macro-molecule that is composed of four submolecules called *nucleotides (nts)* (Adenine (A), Guanine (G), Cytosine (C), Thymine (T)). DNA used for data storage is a single-stranded sequence of these nts. In order to use DNA as an archival medium, digital data is mapped from its binary form into a quaternary sequence of nts using an encoding algorithm. Once encoded, the nt sequence is used to manufacture actual DNA molecules, also referred to as *oligonucleotides* (oligos), through a chemical process called *synthesis* that assembles the DNA one nt at a time. Data stored in DNA is read back by *sequencing* the DNA, which essentially reads out the nt composition of each oligo to produce strings called *reads*, and then decoding the information back from the reads into the original binary form.

DNA possesses several key advantages over current storage technologies. First, it is a three-dimensional storage medium with a capacity of storing 1 Exabyte/mm^3 which is $10^8\times$ higher than tape [13]. Second, DNA is very durable and can last millennia when stored at room temperature under proper conditions [12]. Third, the technologies used for storing data in DNA (synthesis) and reading data from DNA (sequencing) have eternal relevance, as there will always be the need to synthesize and sequence DNA for biological applications. Further, as a storage medium, DNA is decoupled from the reader (sequencer), as DNA can be read by any sequencing platform. Hence, DNA does not suffer from media obsolescence.

2.2 Vision and Challenges

The aforementioned advances in obsolescence-free media clearly paint the vision of a paradigm shift that is under way in replacing the migration-based, tape-based archival infrastructure today with passive, migration-free archival in the future. Given its benefits, synthetic DNA is clearly a promising candidate as an archival medium. However, the use of DNA creates three major challenges: (i) media encoding and decoding, (ii) metadata archival, and (iii) format obsolescence.

Media Encoding. There are several challenges in designing encoders for DNA storage. First, there are several biological constraints that must be respected during encoding to ensure that DNA molecules can be synthesized and sequenced: (i) Experiments have demonstrated that DNA sequences that have a high number of repeated nts, also known as homopolymers (e.g. TTTT) create problems for sequencing [14]. Thus, the encoder must avoid long homopolymer repeats; (ii) Oligos with a low GC-ratio (fraction of Cs and Gs in the oligo) are known to be unstable, while those with a high GC-ratio are known to have higher melting temperatures and create problems for synthesis and sequencing [14]. Thus, the encoder must maintain GC-ratios in a well-defined range.

Second, current synthesis processes cannot synthesize oligos longer than a few hundred nts. Thus, as a single oligo can not store more than a few hundred bits at best, it is necessary to fragment the data and encode it across several oligos. As DNA molecule itself has no addressing, it is necessary to add addressing information explicitly in the oligo during encoding in order to be able to reorder the oligos later during decoding. Thus, SOTA approaches reserve a few bits per oligo to encode this indexing information.

Third, as mentioned earlier, synthesis and sequencing are error prone. There can be insertion errors, where extra nts are added to the original oligo resulting in a sequenced read being longer than the oligo, deletion errors where nts are deleted resulting in shorter reads, and substitution errors. DNA storage also suffers from a *coverage bias* [10]. The average number of reads per oligo generated after sequencing is called *coverage*. Coverage bias refers to the fact that some oligos can be covered by multiple reads, and others can be completely missing as they are not covered by any reads. Coverage bias happens due to the fact DNA synthesis itself creates multiple copies of each DNA molecule. On top of this "physical" redundancy, DNA is also amplified using library preparation steps (like Polymerase Chain Reaction) before sequencing. This amplification creates multiple copies of each synthesized DNA molecule, adding further redundancy. As these amplification procedures are stochastic, different DNA molecules get copied at different rates, leading to coverage bias.

Metadata archival. In order to provide reliable data storage on DNA despite such errors, encoders need to add additional redundancy to data in the form of parity bits generated by using error control coding techniques. Error-control logic often uses additional metadata, like parity check matrices, in order to derive these parity bits from data. As this metadata is required to decode the data back from DNA, it cannot be stored on DNA itself. Typically, with contemporary media like tape, such error-control metadata is stored in the tape reader where the encoding/decoding logic is also implemented. However, as we mentioned earlier, one of the key benefits of DNA–its obsolescence free nature–is due to the separation of the reader (sequencers) from the media itself (DNA). This raises the question of how and where should this auxiliary metadata be stored. One possibility is to store metadata and data separately, with the former on tape and latter on DNA. However, decades of experience from the digital preservation domain argues against this separation and favors self-contained information storage that physically and logically groups together related data and metadata [32]. Thus, it is necessary to develop tiered storage strategies for passive archival of data and metadata.

Format obsolescence. Data stored over long time periods suffers not only from media obsolescence (where data stored on a storage media can no longer be read), but also format obsolescence (data, even if it can be read back from the media, cannot be understood as it is in an "extinct" file format). While DNA solves the media obsolescence problem, it does not solve format obsolescence. Modern data lakes use sophisticated file formats that are optimized for fast querying of data, like Arrow, Parquet, Iceberg, and DeltaLake. The open source nature of these formats and associated tooling is certainly a step towards eliminating format obsolescence. However, they are far from ideal as archival storage formats, as they are continuously evolving without a focus on ensuring backwards compliant data access (future query engines being able to access data stored in older format versions).

In order to use DNA as an archival medium, it is necessary to overcome the three aforementioned challenges. In this paper, we take the first steps towards this goal by presenting OligoArchive-DSM (OA-DSM), an encoding/decoding pipeline for DNA storage that solves the first challenge, in Section 3. After presenting OA-DSM, we outline our future work that must be done to overcome the last two challenges, among others, in Section 5.

3 DESIGN

Several researchers have proposed encoding solutions to demonstrate the feasibility of using DNA as a digital storage medium [4, 7, 9, 11, 15, 18, 19, 23, 28]. In order to ensure reliable data storage despite errors listed in Section 2, all SOTA encoding methods rely on two functionalities: (i) error control coding and (ii) consensus calling. During the write pipeline, input data bits are grouped into blocks (Fig 2(a)), and each block is encoded using error-correction codes to generate parity bits. Each block of data and parity bits is then converted into a set of oligos (Fig 2(b)).

During the read pipeline, the sequencer produces noisy reads. As several reads could correspond to a single oligo, SOTA methods use a consensus caller whose goal is to cluster similar reads and infer the original sequences from each cluster [5, 24, 25]. It is important to note here that these consensus sequences will not be error-free, accurate reproductions of original oligos. Some sequences will be inferred correctly by consensus, but others could have insertion, deletion, or substitution errors, or could go missing due to coverage bias. Hence, it is the job of the error-control decoder to use the additional parity bits to recover original input data despite these errors. In order to provide reliable data storage on DNA, SOTA approaches rely on using a significant amount of redundancy in both writing (in the form of parity bits generated by error control coding) and reading pipelines (in the form of very high sequencing coverage). The added redundancy has the undesirable side effect of amplifying the read/write cost as we show in Section 4. Thus, efficient handling of errors is crucial to reducing overall cost. In the rest of this section, we provide an overview of the OA-DSM write and read pipelines, and explain why the use of a database-inspired columnar oligo layout in OA-DSM can provide substantial reduction in read/write cost.

3.1 OA-DSM Write Pipeline

Figure 1 shows the OA-DSM data writing pipeline. The input to the write pipeline is a stream of bits. Thus, any binary file can be stored using this pipeline. The first few steps in OA-DSM are similar to SOTA pipelines. The input data is grouped into blocks of size 256,000 bits. Each block of input is then randomized to improve the accuracy of read clustering in the data decoding stage as explained in Section 3.2. After randomization, error correction encoding is applied to protect the data against errors. We use Low-Density Parity Check (LDPC) codes [17] with a block size of 256,000 bits. Prior work has demonstrated that such a large-block-length LDPC code is resilient to errors caused by synthesis and sequencing [9].

The LDPC encoded bit sequence is fed as input to the *DSM-oligo-encoder* which converts bits into oligos. OA-DSM differs from SOTA approaches in this encoding process. SOTA approaches design each oligo as a random collection of nts and map each block of data to a group of oligos, one oligo at a time (Figure 2(b)). As a result of this encoding, a group of oligos becomes the unit of recovery; before data can be decoded, the entire group of oligos must be reassembled by consensus. This is the reason why SOTA pipelienes strictly separate consensus calling from decoding and perform consensus calling first.

The key idea in OA-DSM is to integrate decoding and consensus into a single step, where the error-correction provided by decoding is used to improve consensus accuracy, and the improved accuracy in turn reduces the burden on error correction, thereby providing a synergistic effect. In order to do this, the DSM-oligo-encoder designs oligos using composable building blocks called *motifs*. Each motif is itself a short DNA sequence that obeys all the biological constraints enforced by synthesis and sequencing. In order to perform the conversion of bits into motifs, the DSM-oligo-encoder

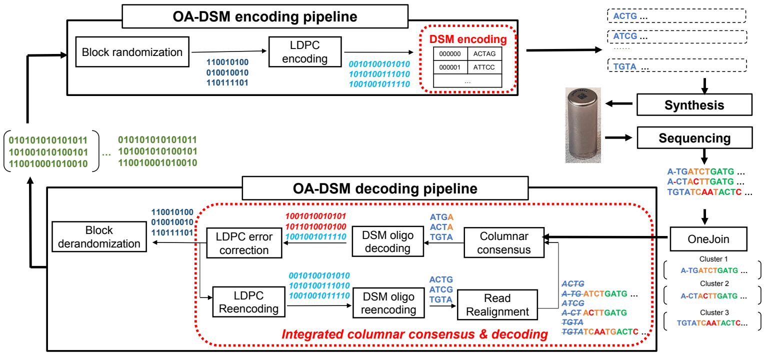

Figure 1: OA-DSM data writing pipeline (top) showing encoding and storage in an encapsulated container like Imagene DNAShell™. OA-DSM reading pipeline (bottom) showing decoding. The blocks in red are unique to OA-DSM (versus SOTA).

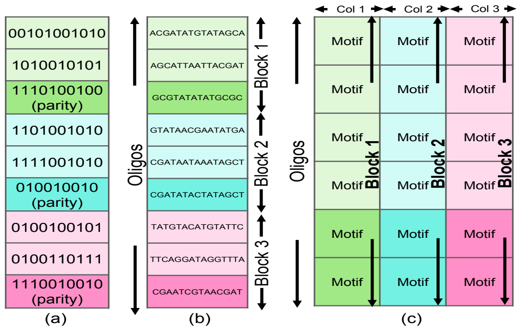

Figure 2: Comparison of SOTA versus OA-DSM layout of oligos. The figures show input data being grouped into blocks with associated parity (a), each block being mapped to multiple oligos with SOTA approaches (b), and each block being mapped to a column of motifs with OA-DSM (c).

maintains an associative array with a 30-bit integer key and a 16nt motif value. This array is built by enumerating all possible motifs of length 16nt (AAA, AAT, AAC, AAG, AGA...) and eliminating motifs that fail to meet a given set of biological constraints (up to two homopolymer repeats and GC content in the range 0.25 to 0.75 based on our prior experimental work [25]). With these constraints, using 16nt motifs, out of 4^{16} possible motifs, we end up with 1,405,798,178 that are valid. By mapping each motif to an integer in the range 0 to 2^{30}, we can encode 30-bits per motif. Thus, at the motif level, the encoding density is 1.875 bits/nt.

The second major difference of our approach to SOTA is the layout of motifs across oligos which is reminiscent of Decomposition Storage Model (DSM) adopted by modern analytical database engines. If we view oligos as rows, then the traditional approach of encoding one oligo at a time by using nts (Figure 2(b)) can be seen as a "row"-based encoding. Instead of this approach, OA-DSM-encoder uses the motifs generated from an error-control coded data block to build oligos by adding a new column at a time as shown in Figure 2(c). This process of column-at-a-time encoding is repeated until the oligos reach a configurable number of columns after which the process is reset to generate the next batch of oligos again from the first column. The generated oligos can then be synthesized to produce DNA molecules that archive data.

There are two subtle aspects of OA-DSM design that we would like to mention explicitly. First, while we organize motifs in a columnar fashion, it is also possible to organize nts in a columnar fashion. As we show later, columnar organization enables us to integrate consensus calling and error-control decoding as we decode column at a time. Central to this approach is the ability to identify where a column ends and a new column begins. With columnar organization of motifs, we can use edit similarity-based alignment algorithms to identify these boundaries even if there are errors in motifs. With single nts, this is not possible, as a single insertion or deletion error can result in complete misalignment of columns. Second, while the figure shows all columns as being of the same size, a small subtlety in the practical implementation is the distinction between the first column and the rest. As we need to index the oligos to enable reordering during decoding, the first column of motifs is generated by using a 15-bit address and a 15-bit data to generate a 30-bit integer. Thus, the first LDPC encoded block is decomposed into 15-bit integers. However, from the second column, there is no need to add addressing information. Thus, rest of the LDPC blocks are decomposed into 30-bit integers. As a result, all columns except the first encode 2 LDPC blocks, while the first column encodes only 1 LDPC block. Note that with 15-bit addresses, we can address up to 32,768 oligos. In ongoing work, we are using OA-DSM to develop a block-addressed, randomly-accessible, DNA file system. Similar to traditional file systems, OA-DSM allows us to view a column like a disk block, and a collection of columns like an extent. The 15-bit address here provides intra-extent addressing. Extents themselves will be addressed separately using a separate mechanism. We explicitly mention this here to clarify that OA-DSM can scale to much larger oligo pools. But for the rest of this paper, we focus on columnar design and consensus.

3.2 OA-DSM Read Pipeline

As mentioned before, data stored in DNA is read back by sequencing the DNA to produce reads, which are noisy copies of the original oligos that can contain insertion, deletion, or substitution errors. As each oligo can be covered by multiple reads, the first step in decoding is clustering to group related reads together. In prior work, we developed an efficient clustering technique based on edit similarity joins [24, 25] that exploits the fact that due to randomization during encoding, reads corresponding to the same original oligo are "close" to each other despite errors and "far" from the reads related to other oligos. The output of this algorithm is a set of clusters, each corresponding to some unknown original oligo.

After the clustering stage, other SOTA methods apply consensus in each cluster followed by decoding in two separate phases. In OA-DSM, we exploit the motif design and columnar layout of oligos to iteratively perform consensus and decoding in an integrated fashion as shown in Figure 1. Unlike other approaches, OA-DSM processes the reads one column at a time. Thus, the first step is columnar consensus which takes as input the set of reads and produces one column of motifs. The choice of consensus algorithm is orthogonal to OA-DSM design. We use an alignment-based bitwise majority algorithm we developed previously for consensus [25], as we found this to provide accuracy comparable to other state-of-the-art trace reconstruction solutions [5]. The motifs obtained from consensus are then fed to the *DSM-oligo-decoder* which is the inverse of the encoder, as it maps the motifs into their 30-bit values. Note here that despite consensus, the inferred motifs can still have errors. These wrong motifs will result in wrong 30-bit values. These errors are fixed by the *LDPC-decoder*, which takes as input the 30-bit values corresponding to one LDPC block and produces as output the error-corrected, randomized input bits. These input bits are then derandomized to produce the original input bits for that block.

As mentioned earlier, SOTA methods do not use the error-corrected input bits during decoding. OA-DSM, in contrast, uses these bits to improve accuracy as shown in the bottom part of the integrated columnar consensus in Figure 1. The error-corrected bits produced by the LDPC-decoder are reencoded again by passing them through the LDPC-encoder and DSM-oligo-encoder. This once again produces a column of motifs as it would have been done during input processing. The correct column of motifs is used to realign reads so that the next round of columnar decoding starts at the correct offset. The intuition behind this realignment is as follows. An insertion or deletion error in the consensus motifs will not only affect that motif, but also all downstream motifs also due to a variation in length. For instance, if we look at the example in Figure 1, we see a deletion error in read $A - TGAT..$ which should have been $ACTGATCT....$ This results in the first motif being incorrectly interpreted as $ATGA$ (instead of $ACTG$, and second motif as $TCTG$ (instead of $ATCT$). Thus, an error early in consensus keeps propagating. Without a knowledge of the correct motif, there is no way to fix this error. But in OA-DSM, by reencoding the error-corrected bits, we get the correct motifs. By aligning these motifs against the reads, we can ensure that consensus errors do not propagate. Note here that such realignment is only possible because we use motifs, as two sequences can be aligned accurately only if they are long enough to identify similar subsequences. Thus, columnar layout without motifs, or with just nts, would not make realignment possible. Similarly, integrating consensus and decoding is possible only because of the columnar layout, as the SOTA layout that spreads a LDPC block across several oligos cannot provide incremental reconstruction.

4 EVALUATION

In this section, we will present the results from our experimental evaluation of the OA-DSM pipeline. Due to lack of space, we present only results from our real wetlab experiments that validate the OA-DSM pipeline and demonstrate the feasibility of our vision here. A more in-depth evaluation and simulation studies can be found in an extended version of the paper [26]. The core components of the OA-DSM pipeline shown in Figure 1 have been implemented in C++17. All experiments were run on a server equipped with a 12-core Intel(R) CPU and 128GB of RAM.

To validate OA-DSM, we used the TPC-H dbgen utility to generate a compressed, TPC-H database archival file of 1.2MB. We limited the size to 1.2MB to limit the cost of actual synthesis. Using OA-DSM configured with 30% LDPC redundancy, we encoded the archive file to generate 44376 oligos, with each oligo of length 160nts (length chosen to optimize synthesis cost). The oligos were synthesized by Twist Biosciences. We sequenced the synthesized oligos using Oxford Nanopore PromethION platform generating approximately 43 million noisy reads.

Table 1: OA-DSM vs. SOTA rd/wt costs: RS-RL [28], LDPC [9], Fountain+RS [15]

	OA-DSM	LDPC-30%	RS+RLL	Fountain+RS
Read Cost	2.10	4.46	4.50	6.8
Write Cost	0.70	0.78	0.92	0.65

In order to test end-to-end decoding, we first used the full 43M read dataset as input to the decoding pipeline. Unsurprisingly, we were able to achieve full data reconstruction, given the ability of OA-DSM to handle much lower coverage levels and higher error rates. In order to stress test our decoding pipeline and identify the minimum coverage that allows fully reconstruction of data, we repeated the decoding experiment on smaller readsets which were derived by randomly sampling a fraction of reads from the 43M read dataset. In doing so, we found that OA-DSM was able to perform full recovery using just 200K reads, which corresponds to a coverage of 4×. At this coverage, nearly 3500 out of 44376 reference oligos were completely missing. However, the LDPC code and columnar decoding were able to successfully recover data. As further reduction in coverage led to data loss, we validate 4× as the minimum coverage OA-DSM can handle with our wetlab experiment.

We will now present a comparison of OA-DSM with SOTA approaches [9, 15, 28] in terms of reading and writing cost. We use the definition of read/write cost as introduced in prior work [9]. Writing cost is defined as $\frac{\#nts-in-oligos}{\#bits}$, where the numerator is the product of the number of oligos and the oligo length, and the denominator is the input data size. Thus, higher the redundancy and encoding overhead, higher the write cost. The reading cost is defined by $\frac{\#nts-in-reads}{\#bits}$. The numerator is the sum total of all read lengths, and denominator is the input size. Thus, higher the coverage required, higher the read cost. Table 1 shows the read and write cost for OA-DSM and other SOTA algorithms. Computing the costs for minimum coverage of 4×, we get a read cost of 3.37 nts/bit, and a write cost of 0.72 nts/bit. For OA-DSM, we compute these costs based on the values from our wet-lab experiment. For SOTA approaches, we use the values as reported by prior work [9].

There are several observations to be made. First, let us compare the OA-DSM with row-based SOTA approach that also uses LDPC (by S. Chandak et al. [9]). Both these cases use the same LDPC encoder configured with 30% redundancy. The cost reported here is for 1% error rate in both cases. Clearly, the OA-DSM approach has both a lower write and read cost. The difference in write cost can be explained due to the fact that in the row-based LDPC approach, the authors also added additional redundancy in each oligo in the form of markers which they used in their decoder. OA-DSM is able to achieve 100% data reconstruction using the same LDPC encoder at a much lower coverage level without such markers as demonstrated by the lower read cost.

Comparing OA-DSM with the other two efficient encoders (large-block Reed-Solomon coding by Organick et al. [28] and fountain codes by Erlich et al. [15]), we see that OA-DSM provides substantially better read cost, but slightly worse write cost than fountain coding approach. As we mentioned earlier, we can further improve the write cost for OA-DSM using several approaches. First, the OA-DSM results in Table 1 were obtained with a 30% redundancy based on its ability to handle even 12% error rate. For lower error rates (less than 1%), as was the case with the Fountain coding work, even 10% redundancy would be able to fully restore data. Second, as mentioned in Section 3, scaling the motif set by using longer motifs (17nt and 33 bits) could allow us to increase bit-level density further from 1.87 bits/nt to over 1.9 bits/nt. These two changes would lead to further reduction in write cost without any adverse effect on the read cost. We leave open these optimizations to future work.

5 DISCUSSION & CONCLUSION

As cloud data lakes grow in popularity, it is becoming increasingly important to scale the archival practices used by these systems. In this work, we argued that the current tape-driven, migration-based archival suffers from fundamental problems which will soon make cost-efficient data archival infeasible. Given recent advances in the design of obsolescence-free storage media, we believe that time is ripe to investigate the impact of such media on the archival tier of data lakes. In this work, we considered synthetic DNA–one such medium that has received a lot of attention recently. We presented OA-DSM, an end-to-end pipeline for DNA data archival that uses a database-inspired, columnar data organization. We showed how such an approach enables the integration of consensus calling and decoding to reduce read/write costs.

While OA-DSM solves the first challenge related to archiving data on DNA mentioned in Section 2, there are other challenges that we do not address in this work. Some of these challenges are out of context for data management research. For instance, current DNA synthesis and sequencing procedures enable writing and reading data from DNA at the rate of Kilobases/second and Megabases/second with a latency of several hours. A large and highly collaborative "DNA storage alliance" has developed around this topic and rapid advances in automation are being investigated for scaling throughput and latency. However, there are several other research directions that are particularly relevant to the data management community.

Metadata archival. In order to recover the data stored in DNA, the OA-DSM decoder needs additional metadata (LDPC matrices and derandomization seed). As the decoder will be run in the future, it will be necessary to save this metadata. As described in Section 2, this metadata should be preferably archived together with the data to ensure completely passive archival. In prior work, we developed a solution that enables data archival on analog media, like archival paper or microfilm by converting digital data into printable, visual barcodes [3]. In ongoing work, we are developing a tiered solution, where data will be stored on DNA with OA-DSM, and metadata will be stored on analog media. The two can then be stored together as a unit passively, completely eliminating the need for migration.

Format obsolescence. As mentioned in Section 2, DNA solves the media obsolescence problem but not format obsolescence. Museums and archives have long suffered from this problem as culturally significant digital data that is stored in databases cannot be preserved over long duration due to proprietary file formats. To circumvent this issue, the SOTA approach for long-term database preservation is to extract data from databases, convert it into a textual representation based on CSV and XML [27], and archive the text file. Unfortunately, the switch from binary to text leads to severe data bloat and is not suitable for large cloud data lakes. Thus, more work is required to understand the applicability of open-source, binary file formats like Parquet, Arrow, Deltalake, and Iceberg, as the basis for long-term archival in terms of forward compatibility, conformance to SQL standards, and their ability to archive application logic expressed in stored procedures, SQL queries, and views which provide the context in which data is accessed.

New synthesis techniques and consensus calling. At its current price point, DNA storage is six orders of magnitude more expensive than tape. The key bottleneck when it comes to DNA storage cost is the phosphoramidite synthesis chemistry that is used for manufacturing DNA. Recently, innovative solutions have emerged in order to dramatically reduce the writing cost based on enzymatic DNA synthesis [21, 22]. While these techniques have the potential to reduce cost by several orders of magnitude, they are highly error prone. As a result, novel consensus algorithms that can decode oligos from highly noisy reads are required. In prior work, we showed that the consensus calling problem can be modeled as a database edit similarity join problem [24]. Thus, we are developing accurate similarity join algorithms for enzymatically-synthesized DNA archives that can scale well while providing very high accuracy [35].

Abstractions for DNA storage. SOTA work on DNA storage has focused on using an object interface to DNA storage [28] which has several limitations. First, only large objects can be efficiently supported as the storage of small objects would incur very high overhead in terms of indexing. Second, objects with non-uniform sizes can lead to data recovery issues due to coverage bias–a small object would correspond to few sequences compared to a large object, and during library preparation, the small objects can be covered by fewer reads than large objects leading to data loss [10]. Tape archives, in contrast, support other abstractions to facilitate data access. At the lowest level, tape exposes storage as a sequence of blocks. Tape also provides the distinction between an index partition and a data partition. Together, these abstractions are used to build higher-level abstractions, like hierarchical file systems, to enable searching and indexing. The columnar design of OA-DSM suggests a natural extension to the block interface, with each column acting like a disk block and a collection of columns acting like an extent. Thus, we are extending OA-DSM to provide block-based access interface to data stored in DNA.

Uncertain data management over DNA storage. Unique to DNA storage is the fact that sequencing DNA not only provides reads but also quality scores, also called Phred scores, that represent the probability of each nt in the read being correct. Current research primarily focuses on using DNA as a precise storage media. An interesting avenue of research is developing techniques that can map phred scores back to higher-level constructs, like attributes or tuples, and use them with uncertain data models [1] for answering queries with error estimates. Doing so will transform DNA into an approximate storage medium [6, 16].

ACKNOWLEDGMENTS

This work was funded by the European Union's HE research and innovation programme projects OligoArchive (Grant No. 863320, Glaciation (Grant No. 101070141), SYCLOPS (Grant No. 101092877), and EIC Transition project MOSS (Grant No. 101058035).

REFERENCES

[1] Charu C. Aggarwal and Philip S. Yu. 2009. A Survey of Uncertain Data Algorithms and Applications. *IEEE Transactions on Knowledge and Data Engineering* 21, 5 (2009), 609–623.

[2] Patrick Anderson, Richard Black, Ausra Cerkauskaite, Andromachi Chatzieleftheriou, James Clegg, Chris Dainty, Raluca Diaconu, Austin Donnelly, Rokas Drevinskas, Alexander Gaunt, Andreas Georgiou, Ariel Gomez Diaz, Peter G. Kazansky, David Lara, Sergey Legtchenko, Sebastian Nowozin, Aaron Ogus, Douglas Phillips, Ant Rowstron, Masaaki Sakakura, Ioan Stefanovici, Benn Thomsen, and Lei Wang. 2018. Glass: A New Media for a New Era?. In *HotStorage*.

[3] Raja Appuswamy and Vincent Joguin. 2021. Universal Layout Emulation for Long-Term Database Archival. In *CIDR*.

[4] R. Appuswamy, Kevin Lebrigand, Pascal Barbry, Marc Antonini, Oliver Madderson, Paul Freemont, James MacDonald, and Thomas Heinis. 2019. OligoArchive: Using DNA in the DBMS storage hierarchy. In *CIDR*.

[5] Tuundefinedkan Batu, Sampath Kannan, Sanjeev Khanna, and Andrew McGregor. 2004. Reconstructing Strings from Random Traces. In *SODA*.

[6] Callista Bee, Yuan-Jyue Chen, Melissa Queen, David Ward, Xiaomeng Liu, Lee Organick, Georg Seelig, Karin Strauss, and Luis Ceze. 2021. Molecular-level similarity search brings computing to DNA data storage. *Nature Communications* 12 (08 2021), 4764.

[7] Meinolf Blawat, Klaus Gaedke, Ingo Hutter, Xiao-Ming Chen, Brian Turczyk, Samuel Inverso, Benjamin W. Pruitt, and George M. Church. 2016. Forward Error Correction for DNA Data Storage. *Procedia Comput. Sci.* 80, C (2016).

[8] Dana M. Caudle, Cecilia M. Schmitz, and Elizabeth J. Weisbrod. 2013. Microform Not extinct yet: Results of a long-term microform use study in the digital age. *Library Collections, Acquisitions, and Technical Services* 37, 1 (2013).

[9] Shubham Chandak, Kedar Tatwawadi, Billy Lau, Jay Mardia, Matthew Kubit, Joachim Neu, Peter Griffin, Mary Wootters, Tsachy Weissman, and Hanlee Ji. 2019. Improved read/write cost tradeoff in DNA-based data storage using LDPC codes. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing*.

[10] Yuan-Jyue Chen, Christopher N Takahashi, Lee Organick, Callista Bee, Siena Dumas Ang, Patrick Weiss, Bill Peck, Georg Seelig, Luis Ceze, and Karin Strauss. 2020. Quantifying molecular bias in DNA data storage. *Nature communications* 11, 1 (2020), 1–9.

[11] George M. Church, Yuan Gao, and Sriram Kosuri. 2012. Next-Generation Digital Information Storage in DNA. *Science* 337, 6102 (2012).

[12] Dominique Clermont, Sylvain Santoni, Safa Saker, Maite Gomard, Eliane Gardais, and Chantal Bizet. 2014. Assessment of DNA Encapsulation, a New Room-Temperature DNA Storage Method. *Biopreservation and Biobanking* 12, 3 (2014), 176–183.

[13] Semiconductor Research Corporation. 2018. 2018 Semiconductor Synthetic Biology Roadmap. https://www.src.org/program/grc/semisynbio/ssb-roadmap-2018-1st-edition_e1004.pdf.

[14] Juliane C. Dohm, Claudio Lottaz, Tatiana Borodina, and Heinz Himmelbauer. 2008. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Research* 36, 16 (2008).

[15] Yaniv Erlich and Dina Zielinski. 2017. DNA Fountain enables a robust and efficient storage architecture. *science* 355, 6328 (2017), 950–954.

[16] Giulio Franzese, Yiqing Yan, Giuseppe Serra, Ivan D'Onofrio, Raja Appuswamy, and Pietro Michiardi. 2021. Generative DNA: Representation Learning for DNA-based Approximate Image Storage. In *International Conference on Visual Communications and Image Processing*.

[17] Robert Gallager. 1962. Low-density parity-check codes. *IRE Transactions on information theory* 8, 1 (1962), 21–28.

[18] Nick Goldman, Paul Bertone, Siyuan Chen, Christophe Dessimoz, Emily M LeProust, Botond Sipos, and Ewan Birney. 2013. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *nature* 494, 7435 (2013), 77–80.

[19] Robert N Grass, Reinhard Heckel, Michela Puddu, Daniela Paunescu, and Wendelin J Stark. 2015. Robust chemical preservation of digital information on DNA in silica with error-correcting codes. *Angewandte Chemie International Edition* 54, 8 (2015), 2552–2555.

[20] Intel. [n. d.]. Cold Storage in the Cloud: Trends, Challenges, and Solutions. White Paper.

[21] Howon Lee, Daniel J. Wiegand, Kettner Griswold, Sukanya Punthambaker, Honggu Chun, Richie E. Kohman, and George M. Church. 2020. Photon-directed multiplexed enzymatic DNA synthesis for molecular digital data storage. *Nature Communications* 11, 1 (2020).

[22] Henry H Lee, Reza Kalhor, Naveen Goela, Jean Bolot, and George M Church. 2019. Terminator-free template-independent enzymatic DNA synthesis for digital information storage. *Nature communications* 10, 1 (2019), 1–12.

[23] Dehui Lin, Yasamin Tabatabaee, Yash Pote, and Djordje Jevdjic. 2022. Managing Reliability Skew in DNA Storage. In *ISCA*.

[24] Eugenio Marinelli and Raja Appuswamy. 2021. OneJoin: Cross-architecture, scalable edit similarity join for DNA data storage using oneAPI. In *ADMS*.

[25] Eugenio Marinelli, Eddy Ghabach, Yiqing Yan, Thomas Bolbroe, Omer Sella, Thomas Heinis, and Raja Appuswamy. 2022. *Digital Preservation with Synthetic DNA*.

[26] Eugenio Marinelli, Yiqing Yan, Virginie Magnone, Marie-Charlotte Dumargne, Pascal Barbry, Thomas Heinis, and Raja Appuswamy. 2022. OligoArchive-DSM: Columnar Design for Error-Tolerant Database Archival using Synthetic DNA. *bioRxiv* (2022).

[27] Library of Congress. 2015. SIARD (Software Independent Archiving of Relational Databases) Version 1.0. https://www.loc.gov/preservation/digital/formats/fdd/fdd000426.shtml. [Online; accessed 28-May-2021].

[28] Lee Organick, Siena Dumas Ang, Yuan-Jyue Chen, Randolph Lopez, Sergey Yekhanin, Konstantin Makarychev, Miklos Z Racz, Govinda Kamath, Parikshit Gopalan, Bichlien Nguyen, et al. 2018. Random access in large-scale DNA data storage. *Nature biotechnology* 36, 3 (2018), 242–248.

[29] Kestutis Patiejunas. [n. d.]. Freezing Exabytes of Data at Facebook's Cold Storage.

[30] Marty Perlmutter. 2017. The Lost Picture Show. https://tinyurl.com/y9woh4e3.

[31] PIQL. 2020. UNICEF deposits Child Convention in AWA. https://www.piql.com/unicef-deposits-child-convention-in-awa/.

[32] Simona Rabinovici-Cohen, Mary Baker, Roger Cummings, Samuel Fineberg, and John Marberg. 2011. Towards SIRF: Self-contained information retention format. 15.

[33] Horison Information Strategies. 2015. Tiered Storage Takes Center Stage. Report.

[34] Wenrui Yan, Jie Yao, Qiang Cao, Changsheng Xie, and Hong Jiang. 2018. ROS: A Rack-Based Optical Storage System with Inline Accessibility for Long-Term Data Preservation. *ACM Trans. Storage* 14, 3, Article 28 (nov 2018), 26 pages. https://doi.org/10.1145/3231599

[35] Yiqing Yan, Nimesh Pinnamaneni, Sachin Chalapati, Conor Crosbie, and Raja Appuswamy. 2023. Scaling Logical Density of DNA storage with Enzymatically-Ligated Composite Motifs. *bioRxiv* (2023).