DNA Data Storage and Hybrid Molecular-Electronic Computing

Douglas Carmean[2] Luis Ceze[1] Georg Seelig[1] Kendall Stewart[1] Karin Strauss[2] Max Willsey[1]

***Abstract*— Moore's Law may be slowing, but our ability to manipulate molecules is improving faster than ever. DNA could provide alternative substrates for computing and storage as existing ones approach physical limits. In this paper, we explore the implications of this trend in computer architecture.**

We present a computer systems prospective on molecular processing and storage, positing a hybrid molecular-electronic architecture that plays to the strengths of both domains. We cover the design and implementation of all stages of the pipeline: encoding, DNA synthesis, system integration with digital microfluidics, DNA sequencing (including emerging technologies like nanopores), and decoding. We first draw on our experience designing a DNA-based archival storage system, which includes the largest demonstration to date of DNA digital data storage of over 3 billion nucleotides encoding over 400MB of data. We then propose a more ambitious hybrid-electronic design that uses a molecular form of near-data processing for massive parallelism. We present a model that demonstrates the feasibility of these systems in the near future.

We think the time is ripe to consider molecular storage seriously and explore system designs and architectural implications.

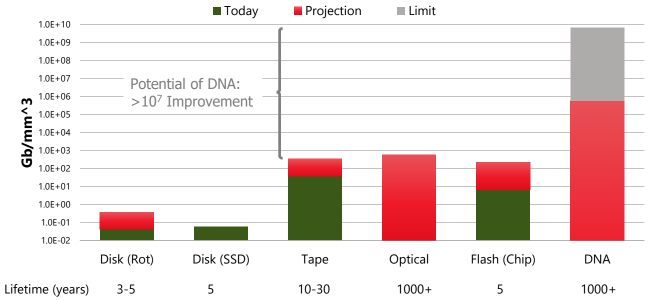


Fig. 1: Comparing DNA with mainstream storage media.

I. Introduction

Exponentially growing data poses a significant challenge to the landscape of current storage technologies. If we are to store and make use of the world's information, we need fundamentally denser and cheaper storage technologies. We believe going to the molecular level is inevitable, as also observed by Zhirnov et al [1].

Synthetic DNA is an attractive storage medium for many reasons: its theoretical information density of about 10^{18} B/mm^3 is 10^7 times denser than magnetic tape (Figure 1), it can potentially last for thousands of years, and it will never go obsolete since we will always be interested in reading DNA for health purposes. The biotechnology industry has developed the basic tools to manipulate DNA, including writing and reading DNA, which can now be leveraged and improved for digital data storage. Importantly, there is rapid exponential progress in DNA reading and writing, arguably surpassing Moore's law [2] (though in the analysis provided in this paper, we chose to model sequencing and synthesis rates that are achievable today). Given the current trends in data production and the rapid progress of DNA manipulation technologies, we believe the time is ripe to make DNA-based storage and computing systems a reality.

In this paper we articulate a vision towards an end-to-end system for archival and retrieval, discuss the challenges in building it, and consider additional applications once it is built. The key challenge is scaling throughput and cost of DNA synthesis and sequencing orders of magnitude beyond the needs of the life sciences industry. Others challenges include system integration, fluidics automation, reliable interfaces between electronics and wet system components, stable preservation, and random access of data stored in molecular form.

Molecular data storage creates opportunities for near-data processing; for example, pattern matching and search could be performed directly on the molecular representation. Adleman [3] noted that DNA's stable double-stranded structure comes with a simple computational primitive: matching single-stranded molecules will stochastically ”bump into each other” in solution. Adleman used this property to compute a solution to the Hamiltonian path problem, pioneering the field of DNA computing. While this area of work has advanced rapidly over the last two decades, the path to large-scale systems remains unclear.

Motivated by progress in DNA data storage, we envision a hybrid molecular-electronic architecture that combines the strengths of molecular and conventional electronics. This approach takes advantage of DNA as both a storage medium and computing substrate. It promises to achieve nearly unlimited bandwidth: data and processing units float free in solution, so computation can diffuse through data and effectively occur everywhere simultaneously. We call this phenomenon *near-molecule processing*. This property effectively breaks the fixed *capacity*/*bandwidth* ratio on typical storage devices in traditional systems, making it especially promising for data-intensive applications such as content-based media search.

In the remainder of this paper, we provide background in Section II, discuss hybrid molecular-electronic systems in general in Section III, and then present our work on

[1] Douglas Carmean and Karin Strauss are with Microsoft

[2] Luis Ceze, Max Willsey, Kendall Stewart and Georg Seelig are with University of Washington

DNA data storage in detail in Section IV. In Section V, we propose a new hybrid molecular-electronic system for image similarity search and model its feasibility. Finally, we conclude with a discussion of future technology trends that impact the design space of these systems in Section VI.

II. Background

DNA's potential as a substrate for molecular computation and storage has been the subject of research for over two decades, dating back to Adleman's exploration of combinatorial problems [3] and Baum's proposal for a massive DNA-based database with associative search capability [4].

A. DNA structure

DNA molecules are biopolymers consisting of a sequence of *nucleotides*. Each nucleotide can have one of four *bases*: adenine (A), cytosine (C), guanine (G), or thymine (T). A single DNA molecule (also called an *oligonucleotide* or *oligo* for short) consists of a sequence of bases, written with initials, e.g., AGTATC. The direction is significant: as normally written, the left end is called the $5'$ ("five prime") end, and the right end is called the $3'$ ("three prime") end.

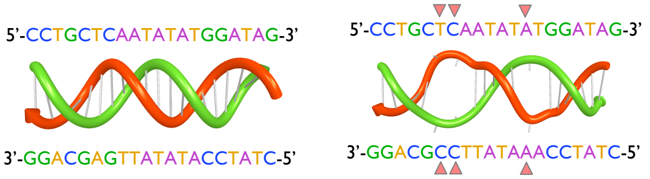
Two oligos can come together to form a double-stranded *duplex*, where bases are paired with their *complement*: A with T, and C with G. The two oligos in a duplex run in opposite directions, therefore a sequence will be fully complementary with its *reverse complement*. This is illustrated in Figure 2a: each base on the $5'$ end of the upper strand is paired with a complementary base on the $3'$ end of the lower strand.

The process of duplex formation is called *hybridization*. Two complementary strands suspended in solution will eventually form a stable structure that requires energy to break. Fully complementary sequences will form the famous double helix structure (Figure 2a).

Two sequences do not have to be fully complementary to hybridize (Figure 2b). Partially hybridized structures are less thermodynamically stable, and occur less frequently at higher solution temperatures. Given two strands, the solution temperature at which 50% of the strands have formed a duplex at equilibrium is called the duplex's *melting temperature*. A higher melting temperature indicates a more stable duplex. The number of unpaired bases is not necessarily related to melting temperature [5]. For instance, changing the mismatched A on the lower strand of Figure 2b to a mismatched G raises the melting temperature to 44.6°C, despite the fact that the number of unpaired bases remains the same. Melting temperature for a pair of sequences can be calculated precisely by thermodynamic simulation software such as NUPACK [6]. In addition to temperature, one can also control hybridization via pH or ionic strength of the solution.

B. DNA writing (synthesis) and reading (sequencing)

DNA *synthesis* is the process of making arbitrary DNA molecules from a specification. One of the most established methods is based on phosphoramidite chemistry due to Caruthers [7]. The method uses ”protected” monomers (individual nucleotides) to prevent the formation of a long homopolymer chain. Removing the protecting group is done with an acid solution. The synthesis cycles works by: (1) incorporating a chosen nucleotide into an existing polymer; (2) strengthening the bond via oxidation; (3) washing out excess monomers; (4) deprotecting the last added base; (5) repeat. DNA synthesis can be made very parallel via an array-based control of either deposition of the next base or localized removal of the protecting group.


5’-CCTGCTCAATATATGGATAG-3’
3’-GGACGAGTTATATACCTATC-5’
5’-CCTGCTCAATATATGGATAG-3’
3’-GGACGCCTTATAAACCTATC-5’

(a) A fully hybridized duplex with complementary sequences. Melting temperature = 66.5°C.

(b) A partially hybridized duplex with mismatched sequences (indicated with red arrows). Melting temperature = 42.5°C.

Fig. 2: DNA molecules can form double-stranded duplexes even when their sequences are not fully complementary, but these structures are less stable and thus have lower melting temperatures.

There are several technologies for DNA synthesis [8]. Enzymatic synthesis is a potential alternative to phosphoramidite chemistry. In this process, engineered enzymes incorporate bases in a controllable fashion without a template. This method promises to be cheaper, faster and cleaner (water-based, as opposed to needing to use solvents). Making synthesis scale requires a control mechanism to select which bases to add to which sequences. This is often called array-based synthesis: sequences are seeded on a surface and reagents flow in succession to add bases in a cyclic fashion. There are several basic technologies, from which three are most commonly used: electrochemical and light-based arrays, which selectively deblock sequences and adds the same base to all deblocked sequences simultaneously; and deposition-based arrays that use inkjet to selectively deposit bases where they are to be added.

The most commercially-adopted DNA sequencing platform today is based on image processing and the concept called sequencing by synthesis. Single-stranded DNA sequences are attached to a substrate and complementary bases with fluorescent markers are attached one by one to individual sequences (yet, in parallel for all sequences). The spatial fluorescence pattern created by the fluorescent markers is captured in an image, which is then processed and fluorescent spots correlated to individual bases in the sequences. The fluorescent markers are then chemically removed, leaving complementary bases behind and setting up the next base in the sequence to be recognized. Scaling such technology to higher throughputs will depend on more precise optical setups and improvements in image processing, and once optical resolution limits are reached, this style of sequencing will probably no longer be appropriate.

Another DNA sequencing solution that has been gaining momentum is nanopore technology. The cornerstone of nanopore technology is to capture DNA molecules and force them through a nanoscale pore which causes small fluctuations in electrical current depending on the passing DNA. The main challenges in using nanopore devices for DNA storage are controlling the high error rates resulting from sensing these minute current fluctuations, which may require heavy signal processing and more precise sensors, and increasing the density of nanopores on a physical substrate, as well as solving problems with clogging and endurance of pores.

C. Brief History of DNA Data Storage

The general idea of using DNA as storage of synthetic information has been around since at least the mid 1960s, when Norbert Wiener suggested the idea of ”genetic” memory for computers. In the past 6 years, work from Harvard [8] and the European Bioinformatics Institute [9] showed that progress in modern DNA manipulation methods could make it both possible and practical soon. Many research groups, including group at ETH Zurich, University of Illinois at Urbana-Champaign, and Columbia University are working on this problem. Our own group at the University of Washington and Microsoft holds the world record for the amount of data successfully stored in and retrieved from DNA: over 500 megabytes as of June 2018.

D. DNA-based computation

The kinetics of DNA hybridization enable more than just a lookup operation. For instance, partial hybridization can implement “fuzzy matching”, where the query and target do not have to be entirely complementary, and the “fuzziness” can be controlled by varying the temperature [5]. This property can be leveraged to perform distance computations, which we discuss further in Section III.

More recently, researchers have shown that hybridization reactions can form complex cascades called *strand displacement* reactions, which can be used to implement general purpose computations, including boolean circuits [10] and neural networks [11].

Beyond hybridization, evolution has led to a variety of enzymes for processing DNA, including cutting, joining, replication, and editing. These enzymes can be used to create even more complex circuits.

III. Hybrid Molecular-Electronic Systems

A hybrid molecular-electronic system aims to leverage the best properties of each domain (Figure 3). As with any heterogeneous system, the strengths and weaknesses of each domain raise a series of critical design questions. In this section, we discuss challenges and trade-offs pertaining to physical constraints, communication, storage, and computation. We discuss these dimensions in general, and we provide examples of how they guided design decisions in the systems presented in Section IV and Section V.

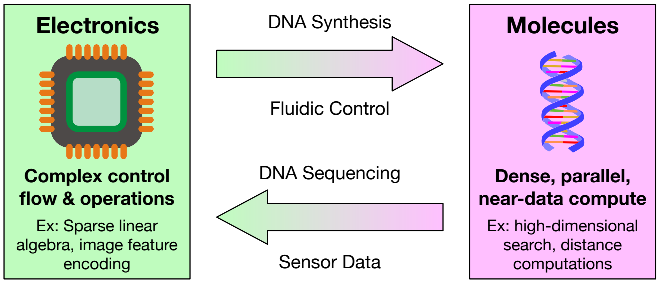

Electronics
Complex control flow & operations
Ex: Sparse linear algebra, image feature encoding
DNA Synthesis
Fluidic Control
DNA Sequencing
Sensor Data
Molecules
Dense, parallel, near-data compute
Ex: high-dimensional search, distance computations

Fig. 3: Hybrid eletronic-molecular architecture. Benefits of electronic and molecular components. Different applications may better fit the strengths of either domain. The arrows show ways of getting data from electronic to molecular components and vice-versa.

A. Physical Constraints

Molecular systems are unique because they require the storage and manipulation of various solutions, including mixing, splitting, diluting, and incubating them. Architects must take care to ensure that a system is physically realizable. Adleman’s famous DNA-based algorithm for solving the Hamiltonian path problem [3] provides a cautionary tale: the amount of DNA required grows exponentially with the graph size. A system is not feasible if modestly sized problems require swimming pools or oceans of DNA. The systems presented in this paper, however, demonstrate that some applications require only a small reaction volume and are thus feasible.

Since we are trying to build computer systems, physical manipulation also necessitates automation. The various steps of preparing, operating on, and analyzing samples are typically done by humans in a wetlab. Microfluidic technology could provide the needed automation, but it is not yet advanced enough to support a practical computer system. Some instances of the technology are not flexible enough, and those that are remain error-prone and difficult to program [12]. Furthermore, programming these hybrid molecular-electronic systems will require intertwined control code, sample manipulation, data analysis, and conventional computation; these challenges remain to be explored.

B. Communication Considerations

How to move information between domains is a primary concern for any heterogeneous system, and it is especially important for hybrid molecular-electronic systems, where communication can be expensive.

There are many ways to communicate from the electronic domain to the molecular domain. DNA synthesis adds new molecules representing data to the system. Physical manipulation also adds data: the choice of which samples to combine determines the behavior of the system. Changes to the environment (e.g., temperature, humidity) can also control the system by influencing chemical properties.

Getting data from the molecular domain back into the electronic domain varies as well. Some operations may obtain enough data from a simple sensor reading: for example, fluorescent markers can indicate the presence of a particular substance or the occurrence of a reaction. DNA sequencing provides even more information by reconstructing the exact sequence of bases from a sample.

The cost of getting data into and out of molecular components is a crucial consideration. The extreme density and parallelism afforded by the molecular domain is of limited use if the interface is a bottleneck. An efficient hybrid system would send a relatively small amount of information to the molecular domain, where lots of work would be done in parallel, and return a relatively small amount of information again to the electronic domain. In this respect, hybrid molecular-electronic systems are similar to heterogeneous systems with hardware accelerators.

C. *Storage Considerations*

Molecular computation is based on strand interaction, so having some information already in the molecular domain would reduce the amount of data that crosses the interface. Bandwidth into and out of the molecular domain is limited, so an existing database in molecular form could greatly improve performance at execution time. Information stored in DNA is dense and long-lived, so this molecular preprocessing could be done out-of-band with actual execution.

The nature of molecular interactions may lead to destructive reads of edits of molecular information. We envision getting around this potential issue via periodic molecular amplification like polymerase chain reaction (PCR) or re-synthesis. Re-amplification of the entire molecular database could lead to errors accumulating over time (e.g., polymerase errors are estimated to be 10^{-6} to 10^{-5} per base). If resorting to re-synthesis, it is important to include only data that was read out and not the entire database, which would possibly oversubscribe the electronic domain with excessive data volumes.

D. *Computation Considerations*

When it comes to computation, our goal is to harness the best of both the electronic and molecular systems. Electronic platforms can be highly general and precise; they can perform a wide variety of operations exactly as specified. No molecular platforms exist today that match the generality and precision of electronic systems, but they may offer orders of magnitude improvements in performance and/or energy efficiency.

Computationally, the main benefit of molecular systems is that certain computations can be performed in a massively parallel fashion. For example, the systems presented below use hybridization to search for exact and approximate matches. Since both query and data are in solution and there are multiple copies of both (DNA synthesis naturally creates multiple copies of the molecules), the search operation is entirely parallel. We refer to this molecular version of near-data processing as *near-molecule processing*.

Interestingly, the latencies of these parallel operations do not change with the size of the dataset: performing an operation on a few items takes as long as doing it over trillions. This "constant-time" performance is offset by a large overhead; operations could take on the order to hours to complete. As such, it may only be profitable to perform such operations in molecular form when the dataset is above a certain offload break-even size. As with electronic systems, this break-even point also determines the granularity of communication between the two domains.

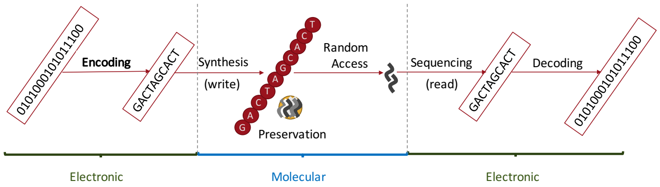


Fig. 4: Overview of DNA-based data storage.

IV. DNA Data Storage

A DNA storage system (Figure 4) takes digital data as input, synthesizes DNA molecules to represent that data, and stores them in a a physical container or pool. To read data back, the system selects molecules from the pool, amplifies them using polymerase chain reaction (a standard molecular biology protocol), and sequences them back to digital data. One can think of a DNA data storage system as a key-value store, in which input data is associated with a key, and read operations identify the key they wish to recover.

A. *Requirements for End-to-End DNA-based Archival Storage*

The requirements of a storage system are low read/write latency, high throughput (bits/s), random access and reliability. DNA manipulation latency is significantly higher than electronics. However, write and read throughput (bits/second) can be competitive. This makes DNA-based storage a good fit for archival purposes, where latency is not critical if throughput is high enough. For example, current archival storage services quote access times in minutes to hours and sometimes service-level agreements (SLA) specify times in the order of a day. But to be competitive with other commercial systems, a DNA archival storage system will need to offer throughputs of about 1 GB/s in a few years.

B. *Encoding and Synthesis*

Writing to DNA storage involves encoding binary data as DNA nucleotides and synthesizing the corresponding molecules. Synthesizing and sequencing DNA is far from perfect (errors on the order 1% per position), hence we need a robust error correction scheme. This process involves two non-trivial steps. First, the trivial encoding from binary into the four DNA nucleotides (A, C, T, G) produces problematic sequences such as long stretches of repeated letters. We avoid that with a rotating code [9] and randomization using one-time pads [13]. Second, DNA synthesis technology effectively manufactures molecules one nucleotide at a time, so it cannot synthesize molecules of arbitrary length without error. Based on current efficiency of synthesis methods and

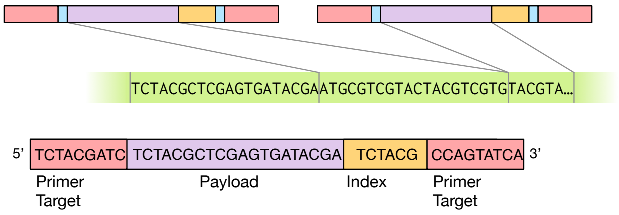


Fig. 5: Layout of a DNA strand for data storage. The primer regions in both extremities are used to both enable molecular amplification and to map molecules to the object [13], [14]. The index region is necessary when reassembling the right order of payloads, since molecular storage does not have fixed 3D physical structure across data items.

technologies, a reasonably efficient strand length for DNA synthesis is about 150 nucleotides (a couple hundred bits of information). The write process therefore splits input data into small blocks which correspond to separate DNA sequences.

Because DNA molecules do not offer spatial organization like traditional storage media, we must explicitly include addressing information in the DNA molecule. Figure 5 shows the layout of an individual DNA strand in our system. Each strand contains a payload, which is a substring of the input data to encode. An address includes both a key identifier and an index into the input data (to allow data longer than one strand). At each end of the strand, special primer sequences [13], [14] – which correspond to the key identifier according to a hash function — allow for efficient sequencing during read operations.

Splitting data into smaller strands requires a coding method that provisions information for later reassembly. Previous work [9] overlapped multiple small blocks, but our experimental and simulation results show this approach to sacrifice too much density for little gain. Our coding scheme embeds indexing information within each block and uses a Reed Solomon-based outer coding scheme [15]. Such coding methods provision what we refer to as *logical redundancy*. Note that DNA synthesis makes many copies of each sequences, and hence also naturally offers *physical redundancy*, in the form of multiple copies of each sequence (on the order of hundreds of millions). Overheads in addressing and error correction can be amortized with longer strands, but because of diminishing returns and higher errors in longer synthesis processes, it is not advantageous to go beyond 500-1,000 nucleotides (Figure 6).

C. Random Access

Random access is fundamental because it is not practical to have to sift through a vast data archive to retrieve a desired data item. Our design allows for random access by using polymerase chain reaction (PCR). The read process first determines the primers for the given key (analogous to a hash function) and synthesizes them into new DNA molecules. Then, rather than applying sequencing to the entire pool of stored molecules, we first apply PCR to the pool using these primers. PCR amplifies the strands in the pool whose primers match the given ones, creating many copies of those strands. To recover the file, we now take a sample of the resulting pool, which contains a large number of copies of all the relevant strands but only a few other irrelevant strands. Sequencing this sample therefore returns the data for the relevant key rather than all data in the system. As a side note, while PCR amplification is not even (i.e., there may be bias) and may amplify undesired strands, it is not a problem for DNA data storage because of underlying error tolerance of the encoding/decoding schemes.

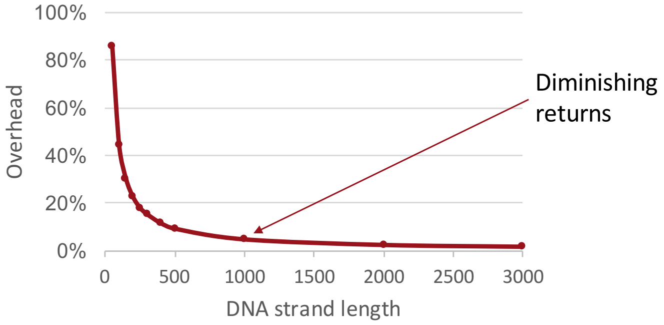


Fig. 6: Overheads as function of strand length.

While PCR-based random access [16], [14], [13] is a viable implementation, we don't believe it is practical to put all data in a single pool. We instead envision a "library" of pools offering spatial isolation. We estimate each pool to contain about 1TB of data. An address then maps to both a pool location and a PCR primer. This design is analogous to a magnetic tape storage library, where robotic arms are used to retrieve tapes. A production DNA-based storage system would require the use of microfluidic automation to perform the necessary reactions. Tape libraries offer random access by robotic movement of cartridges and fast-forwarding to specific tape segments. The equivalent in DNA would be physically isolated "containers" with DNA, along with some form of molecular selection prior to sequencing and decoding. While PCR is the mechanism we have focused on so far, one can also use magnetic-bead based and other DNA random access methods.

D. Reading and Decoding

Reading back the data involves selecting the appropriate pool where the data of interest is stored, retrieving a sample, and *sequencing* the DNA. No matter the DNA sequencing method, the result is a large number of *reads*. Recall that each unique strand is replicated many times in the sequenced sample, so the result will contain many reads for each unique DNA sequence. The decode process will then have to use this physical redundancy to cope with errors introduced by synthesis and sequencing.

The decoder operates in three basic stages: The first step is to cluster noisy reads by similarity to collect all available reads that likely correspond to a unique originally stored DNA sequence. To do so, we employ an algorithm that leverages the input randomization done during encoding. The next step is to processes each cluster to recover the original sequence using a variant of the Bitwise Majority Alignment algorithm (BMA) [17] adapted to support insertions, deletions, and substitutions. Finally, the bits are recovered by using a Reed-Solomon (RS) code to correct errors and erasures.

We have used an Illumina NextSeq instrument that implements this technology to read over 200MB of encoded data so far. We have re-sequenced the data several times, which brings the total of digital data read from DNA to the equivalent of well over 1GB. Sequencing error rates have been reasonably low, typically below 1%, and has not prevented us from decoding any files. The largest commercial nanopore DNA sequencing device to which we have access contains about 2,000 nanopores and delivers error rates of about 12.5%, after recent improvements in its chemistry. Despite this high error rate, we have been able to decode a file read with this platform.

E. *Our results so far*

Our work so far demonstrates an end-to-end approach toward the viability of DNA data storage with large-scale random access. Although we have only reported on the initial 35 files and 200MB of data [13], we have so far encoded, stored, retrieved, and successfully recovered about 40 distinct files totaling about 400MB of data in more than unique 25 million DNA oligonucleotides synthesized by Twist Bioscience (over 3 billion nucleotides in total). Our results represent an advance of more than an order of magnitude over prior work. Our dataset focused on technologically advanced data formats and historical relevance, including the universal declaration of human rights in over 100 languages, a high-definition music video of the band OK Go, and a CropTrust database listing seeds stored in the Svalbard Global Seed Vault.

We demonstrated our random access methodology based on selective PCR amplification, for which we designed and validated a large library of primers, and randomly accessed arbitrarily chosen items from our whole pool with zero-byte error. Moreover, we developed a novel coding scheme that dramatically reduces the sequencing reads per DNA sequence required for error-free decoding to about 6*x*, while maintaining levels of logical redundancy comparable to the best prior codes. Finally, we further stress-tested our coding approach by successfully decoding a file using the more error-prone nanopore sequencing.

V. Near-molecule processing

Most computer systems consist of a few processors surrounded by memory. To perform computation, the processor must load data from memory, operate on it, and write it back. Even parallel processors and GPUs still have to load all of the relevant data before doing computation. As applications become bandwidth-bound, instead of compute-bound, researchers have sought to bring compute closer to the data [18].

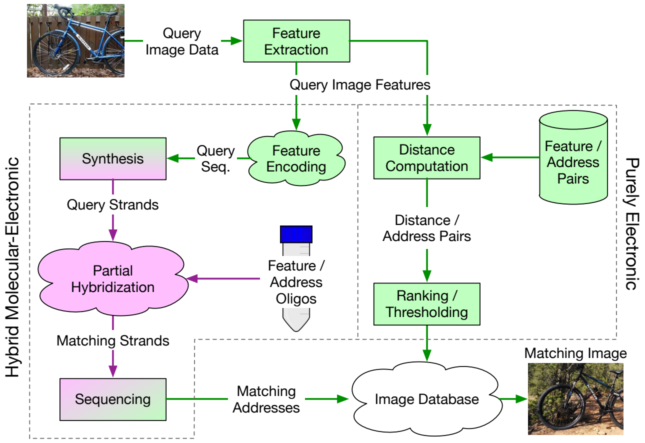


Fig. 7: Stack diagram for a hybrid and a purely electronic content-based image retrieval system. Electronic components are green; molecular components are pink.

In the molecular setting, we can take advantage of nature to perform massively parallel near-molecule computation. If we can formulate the operation and data such that the result we want is thermodynamically favorable, the operation will diffuse through solution and happen everywhere simultaneously. Random access through hybridization and PCR as discussed in the previous section is an example of this. The query strand "searches" for target in the entire dataset, all at once. However, random access in electronic systems does not scan the entire dataset, so molecular retrieval does not offer any performance gain.

Here we explore a more compelling case for near-molecular processing. We describe a DNA-based hybrid system for content-based image retrieval which we call MASS (molecular accelerated similarity search). MASS relies on a biomolecular mechanism for "fuzzy matching" that has not yet been demonstrated but we believe is feasible. The purpose of this section is to, given a new molecular mechanism, discuss how to design a feasible hybrid molecular-electronic system. We also explore the practicality of such a system with a model of its latency, necessary reaction volume, and scalability.

A. *Molecular Accelerated Similarity Search*

Similarity search is a mechanism for searching a large dataset for objects similar to some given query. We focus on a particular instance of this problem, content-based image retrieval (CBIR), the task of finding images that appear visually similar to a given query image. CBIR systems power real-world applications such as Google's reverse image search. Figure 7 shows a stack diagram for our proposed CBIR implementation and a purely electronic one.

The first step in building a CBIR system is to extract visual features from each image in the database. Visual features are usually real-valued numbers that represent the activity of some filter applied to the image. These can be hand-engineered features like scale-invariant feature transform (SIFT) [19], or learned features such as intermediate layer activations from a deep neural network [20]. Pairs of feature vectors can be compared using familiar functions such as Euclidean or cosine distance. To find images that are visually similar to a query, the system searches for image feature vectors within some distance of the query's feature vector.

Ordinarily, such searches could be accelerated by partitioning the data into a tree-like data structure. However, when feature vectors are high-dimensional, partitioning schemes become no better than a linear search. This phenomenon is popularly known as "the curse of dimensionality". Real systems overcome this limitation by using approximation schemes that reduce the amount of data to be sifted through, at the cost of potentially missing similar images [21], [22], [23].

Ultimately, a high-recall CBIR system must examine a large part of the dataset. This provides an opportunity for MASS to outperform its purely electronic counterparts by using the near-molecular compute afforded by the molecular domain.

B. Architecture

Much like our DNA data storage system described in Section IV, the database of the MASS system consists of DNA strands that associate a primer with some data. Instead of mapping an address (the primer) to some data, the strands in the MASS database map encoded feature vectors to the address of the image in some other database. So the MASS system deals with the image feature vectors and the *addresses* of the images instead of the actual image data.

Figure 7 shows the lifetime of single query in the MASS system. The input to the system is a query feature vector which is encoded into a string of bases and then synthesized into (many copies of) a *query strand*. The query strands are then combined with a small sample of the database in the *reaction vessel*. In the reaction, the query strands will partially hybridize with matching targets, performing the similarity search with massive parallelism. The matching targets can then be sequenced, yielding the addresses of the similar images. The images themselves can then be retrieved from a different database downstream, potentially a DNA data store like ours described in Section IV.

The encoder is a critical part of the system that we leave unspecified. We believe that it is possible to encode feature vectors into DNA such that their similarity correlates with partial hybridization efficiency, but this remains to be demonstrated in future work.

The following section will describe the protocol for the molecular search in detail. We will also introduce a simple analytical model that relates the important quantities describing the protocol. The model will let us predict the systems latency and physical feasibility.

C. Modeling a Hybrid Molecular-Electronic System

Figure 8 shows the equations that comprise the model. Parameters marked with syn refer to the synthesized query, rxn refers to the reaction vessel, and seq refers to the solution that actually gets sequenced. They will be introduced as they become relevant, but Figure 8 shows a summary of all model parameters and their values in a potential design.

Param	Value	Description
l_{syn}	100 bases	Length of query strand
r_{syn}	.02 b/s	Synthesis rate in bases per second
t_{syn}	83 min	Synthesis latency
n_{rxn}	1e16	Number of unique strands in reaction
c_{rxn}	10	Copies of each unique strand in reaction
v_{rxn}	1.7 ml	Volume of reaction
ρ_{rxn}	100 μM	Concentration of strands in reaction
n_{seq}	10,000	Number of unique strands sequenced
c_{seq}	40	Copy number required for sequencing
l_{seq}	100 bases	Length of strands that get sequenced
r_{seq}	226 b/s	Sequencing rate in bases per second
t_{seq}	2 min	Sequencing latency

$$t_{\mathsf{syn}} = l_{\mathsf{syn}} / r_{\mathsf{syn}} \tag{1}$$

$$v_{\mathsf{rxn}}\rho_{\mathsf{rxn}} = n_{\mathsf{rxn}}c_{\mathsf{rxn}} \tag{2}$$

$$t_{\mathsf{seq}} = \frac{c_{\mathsf{seq}} l_{\mathsf{seq}} n_{\mathsf{seq}}}{r_{\mathsf{seq}}} \tag{3}$$

Fig. 8: Parameters for the content-based retrieval model and equations that describe their relationships.

1) Query Synthesis: The protocol starts by synthesizing many copies of the query strand, which represents the encoded feature vector of the query image. DNA synthesis makes many copies of a strand at once, so the latency is proportional to the length of the strand, not the number of copies. Synthesizing many copies helps ensure that partial hybridization happens quickly and allows us to perform PCR.

To get the latency of synthesis, we model the the length of the synthesized strand l_{syn} and the rate of synthesis r_{syn}. Equation 1 shows how to calculate the latency of query synthesis.

2) Reaction: The reaction vessel initially contains a sample from the database (see Figure 7). The rxn parameters describe this sample of target strands in the reaction vessel, *not* the synthesized query strands.

The number of *unique* targets in the reaction vessel is n_{rxn}, and each unique strand is replicated c_{rxn} times. This replication factor c_{rxn} is also called the *copy number*. There are $n_{\mathsf{rxn}}c_{\mathsf{rxn}}$ total target strands in the reaction. These are stored at some concentration ρ_{rxn}, which determines the volume v_{rxn}; Equation 2 shows this relation.

Because the targets come from a sample of the database, the reaction has the same concentration (ρ_{rxn}) and number of unique targets (n_{rxn}) as the database. The number of unique targets determines the capacity of the system; the reaction is effectively searching over n_{rxn} unique image feature vectors in parallel.

Once the query strands are added to the database sample, partial hybridization binds the query to similar targets. These targets can be retrieved with a procedure similar to the one used in DNA data storage (Section IV). For example, PCR can amplify strands that hybridized, leaving the reaction vessel dominated by target image features that were similar (and bound to) the query.

3) Sequencing: Once the reaction vessel is dominated by similar target strands, we take a sample of the vessel to avoid unnecessary sequencing. We sample such that we only sequence c_{seq} of each unique strand.

The number of unique strands (n_{seq}) to be sequenced, their copy number (c_{seq}), and their length (l_{seq}) together determine the number of bases to be sequenced. This and the sequencing rate r_{seq} determine the latency (Equation 3).

Note that the amount to be sequenced is not dependent on the size of the dataset, n_{rxn}. Unlike electronic systems, whose time-to-solution is proportional to the size of the dataset, the molecular system instead depends on *the size of the result*. This is the fundamental benefit provided by the near-molecule computation.

For massive datasets on the order of trillions of images or more, the number of images similar to a given query could be quite large, so controlling the number of desired results (those that end up getting sequenced, n_{seq}) independently of the dataset size n_{rxn} is crucial to maintain good performance. To that end, the temperature of the reaction vessel can be raised or lowered to get more or fewer similar results.

D. Model Instantiation

Using the model in Figure 8, we can derive the latencies (t_{syn} and t_{seq}) and capacity of the system (n_{rxn}). The remaining model parameters are constrained by either the biomolecular protocol or technology limits.

1) Protocol Constraints: We choose the reaction concentration $\rho_{\mathsf{rxn}} = 100\mu$M, a common concentration for synthetic DNA [24]. We choose the reaction copy number $c_{\mathsf{rxn}} = 10$. PCR is incredibly specific, we have observed it working when the copy number is as low as 5.

We chose length of the synthesized query, l_{syn}, to be 100 bases. We believe that this is sufficient to encode feature vectors given a dimensionality reduction. The length of the target strands that get sequenced, l_{seq}, is 160 bases. This allocates 100 bases for the encoded feature vector and 60 bases for the address. At a density of 1 bit per base [13], 60 bases is sufficient to uniquely address $n_{\mathsf{rxn}} = 1\text{e}16$ images.

2) Technological Constraints: Sequencing and synthesis are expected to get exponentially faster, improving at a rate exceeding Moore's Law [2]. However, we chose to model sequencing and synthesis rates that are achievable today.

We draw the synthesis rate for our model, r_{syn}, from recent literature proposing a method to synthesize a base every 50 seconds [25]. Recall that synthesis time Equation 1 is proportional only to the length of the strand, not the copy number. Synthesis of a single unique strand is already commercially available on the scale of millimoles, which is well above the amount we require.

Note that we are assuming the existence of a large database of potentially up to 10 quadrillion of *unique* targets. This is beyond the capability of DNA synthesis today. Today's technology can synthesize many unique strands of DNA at once, on the order of millions [13], but making a database that references 10 quadrillion images would only become feasible with further advancements.

3) System Capability: Plugging the above constraints into the model yields a synthesis latency of t_{syn} of 83 minutes and a sequencing latency t_{seq} of 2 minutes. These are of course rough estimations due to the coarse granularity of our model. The partial hybridization and PCR reactions would take on the order of hours. The bottlenecks are clearly DNA synthesis and the reactions, not sequencing.

If we plug in a dataset size (equal to the number of unique strands in the reaction, n_{rxn}) of 10^{16}, the model shows we only require a reaction volume of 1.7 mL. Modeling other systems is outside the scope of this paper, but we believe that MASS would be competitive with or outperform purely electronic systems at this scale.

VI. Discussion

Both synthesis and sequencing need to be lower cost and higher throughput than they are today for DNA data storage and computing to succeed. The gap in both dimensions is daunting, estimated to be about 6 orders of magnitude, but it is important to note that, when used for data storage, DNA synthesis and sequencing have different requirements than for life sciences. First, when storing data, control over the sequences to be synthesized allows for the use of smart error correction to tolerate error rates orders of magnitude higher than those required for life sciences applications. Second, storage applications can tolerate completely missing sequences as well as contamination. Third, data storage needs very few copies of each sequence, compared to the much higher life sciences requirements. Higher synthesis and sequencing density implies simultaneously higher throughput and lower costs, so it will be key to a practical, large-scale end-to-end DNA storage system.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful feedback on the manuscript, and MISL members for input on the research and feedback on how to present this work. This work was partially support by Microsoft, the National Science Foundation and DARPA under the Molecular Informatics Program.

Author Bios

Douglas Carmean is a Distinguished Engineer at Microsoft. His current work explores new architectures on futures device technology. Carmean holds a BS in electrical and electronics engineering from Oregon State University.

Luis Ceze is a Professor at the Paul G. Allen School of Computer Science and Engineering at the University of Washington. His research focuses on the intersection between computer architecture, programming languages, machine learning and biology. His current focus is on approximate computing for efficient machine learning and DNA-based data storage. He co-directs the Molecular Information Systems Lab (MISL), the Systems and the Architectures and Programming Languages for Machine Learning lab (SAMPL). He received his Ph.D. in Computer Science from UIUC and his M.Eng. and B.Eng. from USP, Brazil. He is a Senior Member of IEEE and ACM.

Georg Seelig is an associate professor in the Paul G. Allen School of Computer Science & Engineering and the Department of Electrical and Computer Engineering at the University of Washington. He is an adjunct associate professor in Bioengineering. Seelig holds a PhD in physics from the University of Geneva in Switzerland and did postdoctoral work in synthetic biology and DNA nanotechnology at Caltech.

Kendall Stewart is a third-year PhD student at the University of Washington. Her focus is on designing practical systems that leverage unique properties of unconventional devices. She is currently working on architectures for high-throughput parallel computing using synthetic DNA.

Karin Strauss is a Senior Researcher at Microsoft and an Affiliate Professor in the Allen School for Computer Science and Engineering at University of Washington. Her research lies at the intersection of computer architecture, systems, and biology. Lately, her focus has been on DNA data storage. In the past, she has studied other emerging memory technologies and hardware accelerators for machine learning, among others. Previously, she worked for AMD Research, and before that she got her Ph.D. in 2007 from the Department of Computer Science at University of Illinois, Urbana-Champaign. She is a Senior Member of IEEE and ACM.

Max Willsey is a Ph.D. student at the Paul G. Allen School for Computer Science & Engineering at the University of Washington. He received his B.S. in Computer Science from Carnegie Mellon University (2016). He is a recipient of a Qualcomm Innovation Fellowship and an NSF Graduate Research Fellowship honorable mention. His research interests are in programming languages and computer architecture with applications in biology.

References

[1] Victor Zhirnov, Reza M. Zadegan, Gurtej S. Sandhu, George M. Church, and William L. Hughes. Nucleic acid memory. *Nature Materials*, 15(4):366–370, 2016.

[2] Rob Carlson. Time for new dna synthesis and sequencing cost curves, 2014. https://synbiobeta.com/time-new-dna-synthesis-sequencing-cost-curves-rob-carlson.

[3] L M Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, November 1994.

[4] E B Baum. Building an associative memory vastly larger than the brain. *Science*, 268(5210):583–585, April 1995.

[5] Sotirios A Tsaftaris, A K Katsaggelos, T N Pappas, and T E Papoutsakis. DNA-based matching of digital signals. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages V–581–4. IEEE, 2004.

[6] Joseph N Zadeh, Conrad D Steenberg, Justin S Bois, Brian R Wolfe, Marshall B Pierce, Asif R Khan, Robert M Dirks, and Niles A Pierce. NUPACK: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, 32(1):170–173, January 2011.

[7] Marvin H Caruthers. The Chemical Synthesis of DNA/RNA: Our Gift to Science. *THE JOURNAL OF BIOLOGICAL CHEMISTRY*, 288(2):1420–1427, 2013.

[8] George M Church, Yuan Gao, and Sriram Kosuri. Next-Generation Digital Information Storage in DNA. *Science*, 337(6102):1628–1628, September 2012.

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

[10] David Yu Zhang and Georg Seelig. Dynamic DNA nanotechnology using strand-displacement reactions. *Nature Chemistry*, 3(2):103–113, February 2011.

[11] Lulu Qian, Erik Winfree, and Jehoshua Bruck. Neural network computation with DNA strand displacement cascades. *Nature*, 475(7356):368–372, July 2011.

[12] Nathan Blow. Microfluidics: The great divide. *Nature Methods*, 6(9):683–686, 2009.

[13] Lee Organick, Siena Dumas Ang, Yuan-Jyue Chen, Randolph Lopez, Sergey Yekhanin, Konstantin Makarychev, Miklos Z. Racz, Govinda Kamath, Parikshit Gopalan, Bichlien Nguyen, Christopher Takahashi, Sharon Newman, Hsing-Yeh Parker, Cyrus Rashtchian, Kendall Stewart, Gagan Gupta, Robert Carlson, John Mulligan, Douglas Carmean, Georg Seelig, Luis Ceze, and Karin Strauss. Random access in large scale dna data storage. *Nature Biotechnology*, 2018.

[14] James Bornholt, Randolph Lopez, Douglas M Carmean, Luis Ceze, Georg Seelig, and Karin Strauss. A DNA-Based Archival Storage System. In *ASPLOS '16: Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems*. Microsoft Research, ACM, March 2016.

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

[16] S M Hossein Tabatabaei Yazdi, Yongbo Yuan, Jian Ma, Huimin Zhao, and Olgica Milenkovic. A Rewritable, Random-Access DNA-Based Storage System. *Scientific Reports*, 5(1):1763, September 2015.

[17] Andrew Batu, Tugkan; Kannan, Sampath; Khanna, Sanjeev; Mcgregor. Reconstructing Strings from Random Traces. *Symposium A Quarterly Journal In Modern Foreign Literatures*, 2004(Soda):910–918, 2004.

[18] Rajeev Balasubramonian and Boris Grot. Near-Data Processing [Guest editors' introduction]. *IEEE Micro*, 36(1):4–5, 2016.

[19] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 2004.

[20] Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. pages 157–166, 2014.

[21] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM.

[22] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, January 2008.

[23] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP International Conference on Computer Vision Theory and Applications*, 2009.

[24] Integrated dna technologies. `https://www.idtdna.com`. Accessed: 2017-08-11.

[25] Matej Sack, Kathrin Hölz, Ann-Katrin Holik, Nicole Kretschy, Veronika Somoza, Klaus-Peter Stengele, and Mark M. Somoza. Express photolithographic dna microarray synthesis with optimized chemistry and high-efficiency photolabile groups. *Journal of Nanobiotechnology*, 14(1):14, Mar 2016.