

# DNA Computing

Bachelor's Thesis, 12.10.2019

Author:

PANU LAPPALAINEN

Supervisor(s):

JUSSI TOPPARI



UNIVERSITY OF JYVÄSKYLÄ  
DEPARTMENT OF PHYSICS

© 2019 Panu Lappalainen

Julkaisu on tekijänoikeussäännösten alainen. Teosta voi lukea ja tulostaa henkilökohtaista käyttöä varten. Käyttö kaupallisiin tarkoituksiin on kielletty. This publication is copyrighted. You may download, display and print it for Your own personal use. Commercial use is prohibited.

## Tiivistelmä

Lappalainen, Panu

DNA Computing (DNA-laskenta)

Kandidaatintutkielma

Fysiikan laitos, Jyväskylän yliopisto, 2019, 33 sivua

Tässä kirjallisuuskatsauksessa käsitellään DNA:n mahdollisuuksia tietokonelaskennassa. DNA on erinomainen rinnakkaislaskennassa laskentanopeutensa ja energiatehokkuutensa puolesta, mutta sen hitaan valmisteluajan ja yleisen vaikean käsiteltävyyden takia se ei kilpaile nykyaikaisten tietokoneiden kanssa useimmilla aloilla, joskin DNA:sta tehdyille logiikkaportteille on löydetty käyttötarkoituksia virusten havaitsemisessa. DNA:n syntetisointi ja sekvensointi (kirjoittaminen ja lukeminen) on hidasta, mutta DNA:lla on korkea informaatiotiheys ja se säilyy mahdollisesti tuhansia vuosia. Tämä tekee DNA:sta hyödyllisen pitkäaikaisarkistointitarkoituksiin.

Avainsanat: DNA-laskenta, rinnakkaislaskenta, DNA-logiikkaportit, DNA-tiedontalennusväline, DNA-turvallisuus



## Abstract

Lappalainen, Panu

DNA Computing

Bachelor's thesis

Department of Physics, University of Jyväskylä, 2019, 33 pages.

In this literary review the potential of DNA in computing is discussed. DNA is great for parallel computation in terms of performance speed and energy efficiency, but due to a long initial preparation time and general difficulty in usage, it is no competition to modern computers in most areas. However, logic gates made of DNA have found potential use in virus detection. The synthesis and sequencing (that is, writing and reading) of DNA is slow, but DNA has very high information density and lasts potentially thousands of years, making it useful for long-term archiving purposes.

Keywords: DNA computing, parallel computation, DNA logic gates, DNA data storage, DNA security



# Contents

<b>Tiivistelmä</b>	<b>3</b>
<b>Abstract</b>	<b>5</b>
<b>1 Introduction</b>	<b>9</b>
1.1 About This Topic . . . . .	9
1.2 What is DNA? . . . . .	9
1.3 Origins of DNA Computing . . . . .	11
<b>2 Theoretical Background</b>	<b>13</b>
2.1 Binary and Quaternary Bases . . . . .	13
2.2 Principles of Logic Gates . . . . .	14
2.3 Limitations of Digital Data Storage . . . . .	15
<b>3 Research</b>	<b>17</b>
3.1 Molecular Logic Gates . . . . .	17
3.2 Logic Gates from DNA . . . . .	18
3.3 Nontraditional Computing . . . . .	20
3.4 Data Storage in DNA . . . . .	23
<b>4 Discussion</b>	<b>25</b>
4.1 Benefits of Parallel Computation . . . . .	25
4.2 Superior Data Storing Capabilities of DNA . . . . .	26
4.3 Risks . . . . .	26
<b>5 Conclusion</b>	<b>29</b>
<b>References</b>	<b>29</b>

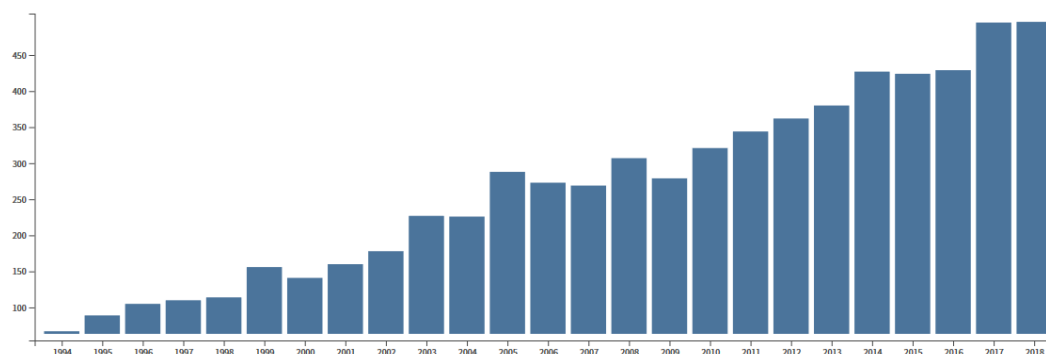




# 1 Introduction

## 1.1 About This Topic

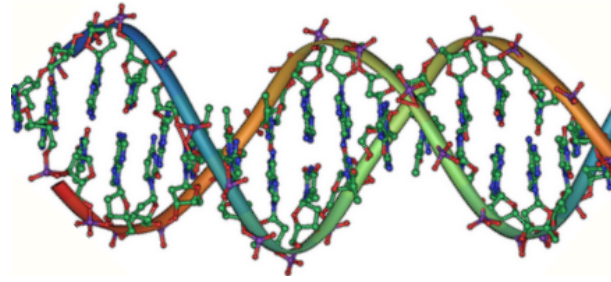
*DNA* (deoxyribonucleic acid) is genetic material, containing a double helix with base pairs, a property that makes DNA not only excellent in storing information, but also programmable. DNA computing has been developed since 1994, and many discoveries have been made for use of DNA in logic gates, parallel computation and information storage platforms. DNA computing has shown a healthy growth in interest over the years (Figure 1). This thesis is a literary review of the many aspects of the usage of DNA in computing and briefly introduces the theory behind it.



**Figure 1.** Number of publications per year on the topic of “DNA computing”. Web of Science [1] (data retrieved with keywords “dna computing”).

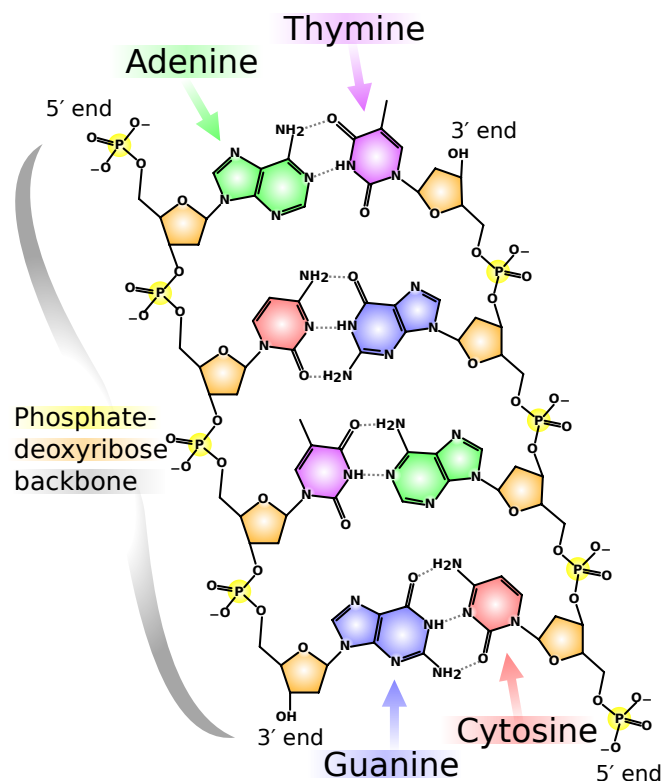
## 1.2 What is DNA?

The information transmitted from parents to offspring is stored in genes, sequences of DNA, which are contained in chromosomes. DNA consists of a chain of tiny organic structural components called *nucleotides* arranged in a double helix (depicted in Figure 2). Each nucleotide is made up of a phosphate group, a deoxyribose sugar and a *nucleobase*. There are four different nucleobases: adenine (abbreviated as A), cytosine (C), guanine (G) and thymine (T). These one-letter abbreviations are also used to refer to the entire nucleotide containing the abbreviated nucleobase.



**Figure 2.** The double helix of DNA. Wikimedia Commons.

Each nucleobase forms a *base pair* with another nucleobase with hydrogen bonds. Adenine and thymine are capable for making two hydrogen bonds, while cytosine and guanine are capable of making three. [2] Thus, adenine will always pair with thymine to form an A-T base pair, and cytosine will always pair with guanine to form a C-G base pair (Figure 3).



**Figure 3.** The chemical structure of the four base pairs of DNA. Adenine forms two hydrogen bonds with thymine, and guanine forms three hydrogen bonds with cytosine. Wikimedia Commons.

### 1.3 Origins of DNA Computing

The creation of the field of DNA computing is often attributed to Leonard Adleman, who in 1994 used DNA to solve an instance of the directed Hamiltonian path problem [3]. In graph theory, a *graph* is a set of points called *vertices* connected by lines called *edges*. A sequence of edges is called a *path*. A *Hamiltonian path problem* is an NP-complete decision problem (where “NP” stands for “nondeterministic polynomial time”, that is, no efficient algorithm exists for solving it). In it, the existence of a path visiting every vertex exactly once (or lackthereof) is determined. Adleman took advantage of the base-pairing properties of DNA and assigned single DNA strands to each edge and each vertex. Both strand halves of each edge were complementary to a half of a vertex it connected to. The molecules formed a long double strand, thus revealing the Hamiltonian path.



## 2 Theoretical Background

### 2.1 Binary and Quaternary Bases

The *base*, also known as radix, is the number of unique digits used to represent numbers (not to be confused with *nucleobases* of DNA, where “base” refers to chemical basicity, low acidity). Modern Western people are used to the base-10 *decimal* numeral system, likely originating from the total number of fingers typical to our species. Other base systems have been used throughout history, such as the Mayan base-20 *vigesimal* or the Babylonian base-60 *sexagesimal* system [4].

The 10 unique digits in the decimal system run from 0 through 9. The base-2 *binary* system only has two digits, 0 and 1, which is convenient in digital electronic circuitry used in modern computers relying on voltages. The digit 0 can be defined as “off”, or - for practical reasons - a voltage below a threshold value (in this work henceforth referred to as “low”). The digit 1 can be defined as “on”, or a voltage above a threshold value (henceforth referred to as “high”). In the genetic code in DNA, the A, C, G and T nucleotides can be thought of as the four unique “digits” of a base-4 *quaternary* system. Table 1 demonstrates the relations between different bases, including a base-3 *ternary* system, which will be relevant later in Section 3.4.

**Table 1.** The relations between bases 2, 3, 4 and 10, where base-4 is represented with both the numerical digits from 0 through 3 and the abbreviations of DNA nucleotides in alphabetical order. Leading zeroes are not shown.

Base	Name	Value										
2	Binary	0	1	10	11	100	101	110	111	1000	1001	1010
3	Ternary	0	1	2	10	11	12	20	21	22	100	101
4	Quaternary	0	1	2	3	10	11	12	13	20	21	22
		A	C	G	T	CA	CC	CG	CT	GA	GC	GG
10	Decimal	0	1	2	3	4	5	6	7	8	9	10

## 2.2 Principles of Logic Gates

Conjunction (AND), disjunction (OR) and negation (NOT) gates are the three most basic logic gates, and the building blocks of all logic. Apart from the NOT gate, each of the logic gates has two inputs and one output. The inputs and outputs deal with binary variables that can have two different values. In logics and mathematics these two values are defined as true and false, or as “high” and “low” in electronics. These can be represented with the binary system as 1 and 0, respectively.

The simplest logic gate is the NOT gate, which inverts a single input value; 1 becomes 0 or 0 becomes 1. Other logic gates have two inputs, hence there are four possible combinations. The AND gate produces an output of 1 if and only if both of its inputs are 1 – that is, when the first *and* the second input is 1, hence the name. For comprehensibility, these values can be arranged in a *truth table*, as shown in Table 2. The OR gate produces 1 if one or both of its inputs are 1.

**Table 2.** The AND gate truth table for binary variables **A** and **B**.

input		output
<b>A</b>	<b>B</b>	<b>A AND B</b>
0	0	0
0	1	0
1	0	0
1	1	1

All these logic gates may be combined; connecting an AND and a NOT gate in series will produce 1 if one or neither of the inputs is 1, but 0 when both are 1. This type of gate is called the NAND gate (as in NOT-AND), or the NOR gate for OR and NOT combination. The XOR gate is an exclusive OR gate, returning 1 if one and only one of the inputs is 1; its inverse is the XNOR gate.

All operations can be obtained by combining AND, OR and NOT gates. NOR and NAND gates are sometimes called “universal logic gates”, since Charles Sanders Peirce showed in 1880 that using either one of these gates alone, all other gates can be reproduced. [5]

Different combinations of logic gates can be used to solve arithmetic operations. A simple example with two logic gates is a half-adder, which calculates the sum of two binary inputs. Both of these inputs connect to both of the two logic gates, thus

producing two separate outputs. One output is the sum (S) that incorporates an XOR gate, and the other one the carry (C) that incorporates an AND gate (Table 3).

**Table 3.** The truth table of a half adder calculating a binary operation  $\mathbf{A+B}$ , where S (sum) is the *ones digit* and C (carry) is the *tens digit*. When both  $\mathbf{A}$  and  $\mathbf{B}$  are one, the value will overflow and “carry” to the tens digit ( $1+1=10$ ).

input		output	
$\mathbf{A}$	$\mathbf{B}$	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

### 2.3 Limitations of Digital Data Storage

Over 10 000 years ago early Mesopotamians used clay tablets to store information, some of which have survived to this day. Around 4000 years ago Egyptians were using papyrus rolled in scrolls up to 40 metres long. While papyrus scrolls rapidly deteriorated, clay tablets could be baked hard and stored in libraries for hundreds of years, surpassing in longevity even paper, which has since become the standard writing platform. [6]

Compact disc (CD) is an early modern example of a digital optical disc data storage format with a physical recordable spiral. The surface of the disc is susceptible to scratches and thus vulnerable to data loss. Although CD is still used, today a more common storage format is flash memory, used in e.g. USB flash drives such as memory sticks. A major problem with most of modern data storage formats is the relatively short lifetime of the storage device, with many factors such as temperature affecting the lifespan. With proper handling, CDs and its more sophisticated extensions like DVDs and Blu-Ray discs are estimated to have a “shelf life” of 100–200 years [7]. The lifetime of flash memory is difficult to determine, but due to charge leaks from flash cells over time, most estimates are around 10 years of data life expectancy [8].

When it comes to information density, a standard A4-sized writing paper holds around 500 words, or about 25 000 bits. With dimensions of 210 mm  $\times$  297 mm and thickness of 0.1 mm, it has an information density of about 4 bits per mm<sup>3</sup>. CDs have an information density of about 10<sup>5.5</sup>, DVDs about 10<sup>6.5</sup> and Blu-Ray discs

about  $10^7$  bits per  $\text{mm}^3$ . Even commercial flash memory has an information density of only around  $10^{9.5}$  bits per  $\text{mm}^3$ . [9]

Thus, due to the low information density and aforementioned short lifetime, most modern data storage formats are unsuitable for long term archiving purposes.



## 3 Research

### 3.1 Molecular Logic Gates

A molecule can perform a logical operation to yield a single output, like a logic gate. The inputs of this molecular logic gate may be either physical or chemical, and its output is typically optical, interpreted with spectroscopy such as transmittance or absorbance.

The foundation of computers, transistor–transistor logic (TTL) operates with electricity as both inputs and outputs. It is fed with a power supply of  $(5.00 \pm 0.25)$  V. For TTL gate inputs “low” is defined as voltage between 0 and 0.8 volts and “high” as voltage between 2 and 5 volts. For outputs the “low” is defined as voltage between 0 and 0.5 volts and “high” as voltage between 2.7 and 5 volts. [10]

In 2000 Silva and McClenaghan [11] constructed a molecular half-adder with parallel XOR and AND gates, using  $\text{H}^+$ - and  $\text{Ca}^{2+}$  ions as input signals, such that “low” input corresponded to  $10^{-9.5}$  M for  $\text{H}^+$  and below  $10^{-9}$  M for  $\text{Ca}^{2+}$ , and “high” input corresponded to  $10^{-6}$  M for  $\text{H}^+$  and  $10^{-2.3}$  M for  $\text{Ca}^{2+}$ . The output of the XOR gate was measured in transmittance at 390 nm and the AND gate as fluorescence at 419 nm. With the arrangements they used (Table 4), both gates were compatible in terms of chemical inputs, optical outputs and power supplies with minimal interference with each other.

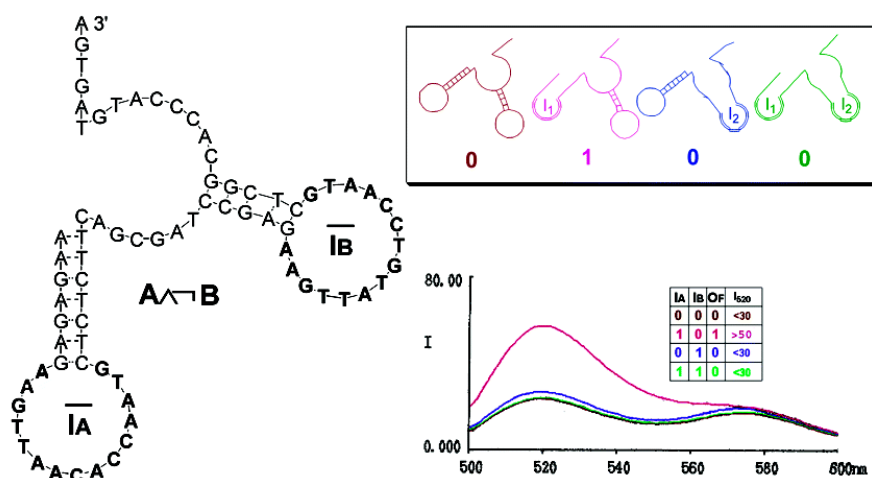
**Table 4.** Simplified version of the “Table 1” of Silva and McClenaghan’s 2000 article, depicting the truth tables of the three of their logic gates operated separately. (The units are irrelevant for the purpose of this work.) [11]

$\text{H}^+$ input	$\text{Ca}^{2+}$ input	XNOR output		XOR output		AND output	
		Extinction		Transmittance		Fluorescence	
0	0	high	13000	low	5	low	4
0	1	low	3100	high	49	low	10
1	0	low	4100	high	39	low	9
1	1	high	11400	low	7	high	100

### 3.2 Logic Gates from DNA

For the purposes of a DNA logic gate, in 2004 Okamoto, Tanaka and Saito [12] developed an artificial nucleobase Methoxybenzodeazaadenine ( $^{\text{MD}}\text{A}$ ) that formed stable base pairs with the natural nucleobases cytosine (C) and thymine (T). They executed the DNA logic gate by hole transport through short DNA duplexes consisting of “input strands” and “logic gate strands”. Previously DNA hole transport mediated by  $\pi$  stack had been prone to oxidative damage to guanine bases, but the artificial  $^{\text{MD}}\text{A}/\text{T}$  base pair behaved as a good mediator and was not oxidatively decomposed, according to the article. In their experiment, the input strand containing “input pyrimidines” with “logic bases” hybridized the logic gate strand with opposite logic bases. As the logic gate was designed to observe output signal when hole transport occurred, it was triggered by photoirradiation with 312 nm transilluminator at 0 °C. Its efficiency is defined by the ratio of oxidation strand cleavage. The principle works well for complicated combinatorial logic circuits.

Other methods have also been used in the research of DNA logic gates. In 2002, Stojanovic et al. [13] created NOT, AND and XOR gates by combining molecular beacon stem-loops with hammerhead-type deoxyribozymes, operating with fluorescence, with oligonucleotides as both inputs and output. They listed multiple reasons for choosing oligonucleotides, mainly their speed, recognition and stability. They defined two oligonucleotides as inputs (where their presence indicated 1 and absence 0) and a cleaved product oligonucleotide as the output.



**Figure 4.** Basic concept of logic gates (ANDNOT depicted) based on molecular beacon stem-loops with hammerhead-type deoxyribozymes, by Stojanovic et al. [13]

Stojanovic et al. constructed the NOT gate through substitution of a nonconserved loop in the deoxyribozyme with a beacon stem loop complementary to the input. The deoxyribozyme was inactive in a complex with the input oligonucleotide, and the gate was active with a closed loop and inactive with an open loop. An AND gate was constructed through attachment of two loops complementary to the input oligonucleotides to the 5'- and 3'-ends of the deoxyribozyme. This correctly resulted the deoxyribozyme to be active only if both inputs were present.

For the more complicated XOR gate Stojanovic et al. combined an ANDNOT gate with a complementary ANDNOT gate, ie.

$$\mathbf{A XOR B} = (\mathbf{A AND (NOT B)}) OR (\mathbf{B AND (NOT A)}),$$

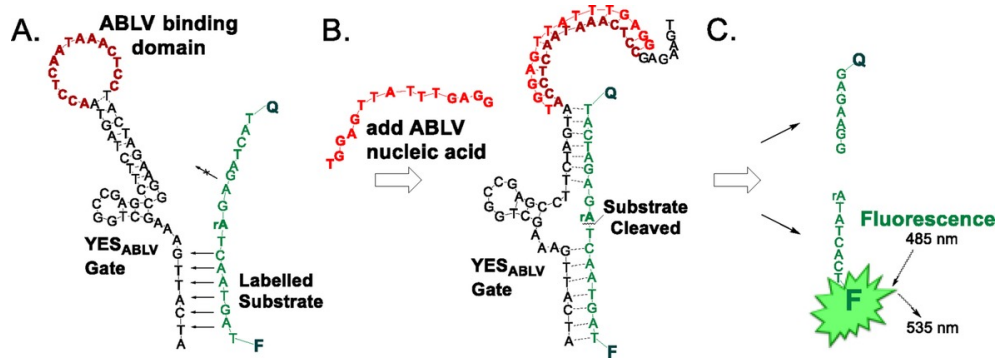
where **A** and **B** are the inputs. They claimed the method to “suffice to generate any Boolean function, subject only to practical constraints of specific detection and [their] future ability to serially connect the gates”, as the universal NAND gate could be constructed.

As a brief example of a vastly different approach, Tam et al. [14] used one- and two-photon excitations to create a reversible DNA logic gate platform in 2016. Their rationale was that “light would be an ideal signal input for exogenous control of the performance of DNA logic gates because it is a clean, non-invasive, and non-contact source”, with no waste being accumulated to interfere with the working efficiency. In their highly technical article they describe their demonstration of the use of “two different wavelength ranges of excitation light as inputs to remotely trigger the responses of the self-assembled DNA devices”.

When it comes to utilizing the DNA logic gates in their fullest potential, Vijaykumar and Macdonald [15] developed DNA logic gate automation in 2017 for detection of Rabies and other lyssaviruses. They arranged deoxyribozyme-based logic gate networks into visual displays, being automatically able to discriminate between seven different genotypes of Lyssaviruses. An ANDNOT gate was used to prevent non-specific activation across genotypes, whilst the large diversity of Lyssavirus sequence populations were detected with mixed-base logic gates.

According to Vijaykumar and Macdonald, immediate activation of biosensors is not always desirable, so they experimented with networks of molecular logic gates arranged into visual displays that can be embedded within a biosensor. They constructed a YES gate, which binds to a labelled substrate, but does not cleave until a nucleic acid is added (Figure 5). The output fluorescence was used in a

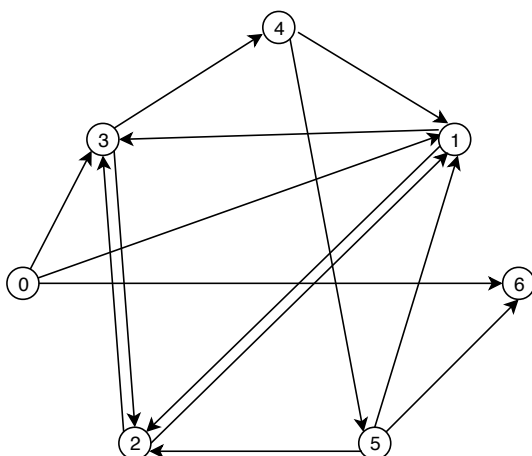
5x7 dot matrix circuit that visually displays the first letter of the viral genotype name and its genotype number.



**Figure 5.** Basic concept of the deoxyribozyme-based YES logic gate for detecting Australian bat lyssavirus (ABLV) by Vijaykumar and Macdonald [15]: A. The gate cannot initially cleave with the labelled substrate. B. The added 15-nucleotide ABLV nucleic acid binds to the stem-loop region and activates the gate. C. The gate cleaves and accumulates fluorescence over time.

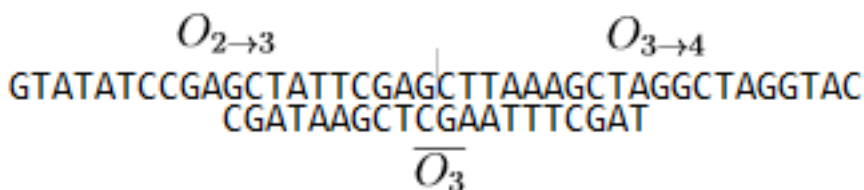
### 3.3 Nontraditional Computing

As briefly discussed in Section 1.3, Leonard Adleman [3] used the very nature of DNA in finding a Hamiltonian path in a seven-vertex directed graph shown in Figure 6. In graph theory, the number-labelled “points” of a graph are called *vertices* and the lines connecting them *edges*. Arrows indicate that the edges are *directed*, one-way. A Hamiltonian path exists should there be a path that visits every vertex once and only once. Vertex 0 was defined as the starting point and vertex 6 as the ending point for the sake of “convenience in this exposition”.



**Figure 6.** Recreation of the directed graph used in Leonard Adleman's experiment in 1994 containing a Hamiltonian path  $0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4, 4 \rightarrow 5, 5 \rightarrow 6$ . [3]

Each vertex  $i$  was associated with a 20-nucleotide long sequence of DNA, or *20-mer*, and denoted as  $O_i$ . Each edge  $i \rightarrow j$  was associated with a 20-mer oligonucleotide  $O_{i \rightarrow j}$  whose first half was complementary to the second half  $O_i$  and the second half complementary to the first half of  $O_j$ . (Vertices 0 and 6 were the starting and ending points, thus edges connecting to them included the entirety of their 20-mer.) The oligonucleotides of the edges,  $O_{i \rightarrow j}$ , were mixed in single ligation reaction with the oligonucleotides of the vertices,  $\overline{O_i}$ . Figure 7 illustrates the binding of one part of chain (edges  $2 \rightarrow 3$  and  $3 \rightarrow 4$ ). With electrophoresis, the longest chain formed was found to be 140 base pairs long. Since each vertex consists of 20 base pairs, this proves that the Hamiltonian path between the seven vertices exists. [3]

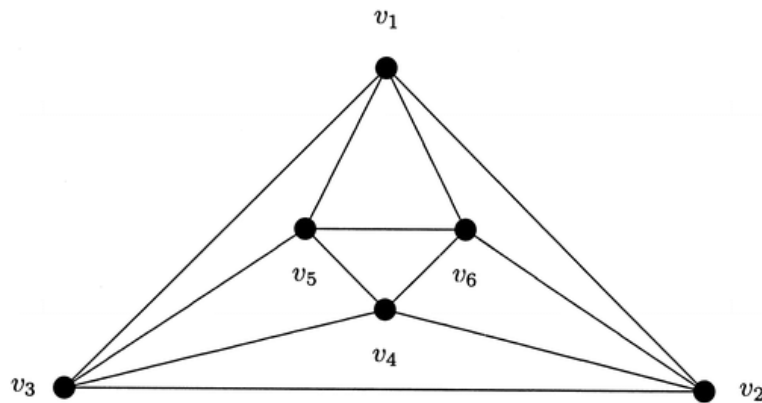


**Figure 7.** A recreation of the illustration in Adleman's original 1994 article: the second half of the oligonucleotide  $O_{2 \rightarrow 3}$  of the edge  $2 \rightarrow 3$  binding to its complement, the first half of the oligonucleotide  $\overline{O_3}$  of the vertex 3, and the first half of the oligonucleotide  $O_{3 \rightarrow 4}$  of the edge  $3 \rightarrow 4$  binding to its complement, the second half of  $\overline{O_3}$ . [3]

DNA has since been demonstrated to be able to be used to solve other mathematical problems as well, such as in 2001 by Liu et al. [16] to solve a graph colouring

problem for an octahedron (Figure 8). In the graph colouring problem the task is to find a way to colour the vertices of the graph in a such way that adjacent vertices never have the same colour, using the minimum number of colours (the “chromatic number” of a graph) [17]. The problem has well-known algorithms for solving, but since the problem is NP-complete, no efficient algorithm exists for it. Liu et al. designed a nondeterministic algorithm to solve it using DNA; the following is a simplification of the algorithm described in their article:

For a graph  $G$  with  $N$  vertices, each possible colouring scheme is represented by a rearrangement of  $N$  elements. A bit set to  $i$  represents that the corresponding vertex is assigned the  $i$ th colour. From the complete data pool, all rearrangements in which repeated numbers are present in the positions of adjacent vertices are eliminated. The remaining data pool is sorted and the rearrangements with the minimum number of colours will reveal the chromatic number.



**Figure 8.** The octahedron used for the graph colouring problem by Liu et al. [16]

Each bit in the rearrangement is represented by the name  $N_v$  and the colour  $C_v$  of the vertex  $v$ . Thus, in a DNA molecule representing a six-digit rearrangement every other section is a name section and every other a colour section, with an extra name section  $N_7$  at the end for PCR amplification. The length of  $N_i$  was 20 base pairs and the length of  $C_j$  was  $(10+5j)$  base pairs. Hence each DNA representation was 230–380 base pairs long.

The data pool was constructed using parallel overlap assembly. All 36 fragment oligonucleotides were mixed together for thermal cycling, during which the name strings in one oligonucleotide got annealed to the complementary strings of the next oligonucleotide. After all combinations got built, the mixture was PCR amplified

with restriction enzymes. The colouring of adjacent vertices was prevented with six sequential restriction operations of dividing the data into separate test tubes and cutting the banned colours.

The results of the final PCR can be read from the electrophoresis of the products. Since in the data structure, the shortest length of DNA represents the chromatic numbers, the lowest band in the electrophoresis is the chromatic number of the graph. Its length was 260 base pairs, thus the chromatic number is 3. Finding out the colouring of each vertex in the graph is more complicated, as the answer DNA must be extracted and sequenced.

### 3.4 Data Storage in DNA

DNA has been successfully recovered from a 60 000 year old mammoth in ice [18]. This shows that DNA has the potential to store information for millennia, as sometimes even samples that were not carefully prepared have survived. Clearly, with the correct preparation, information stored in DNA can outlast many of the modern storage devices.

As discussed in Section 2.1, there are four different possible nucleobases in a nucleotide of DNA. This means that theoretically DNA can encode 2 bits per nucleotide, or  $10^{18}$  bytes per gram of single-stranded DNA. This, however, is practically impossible due to synthesis and reading errors associated with the biological material. In 2012 Church et al. [9] saved a book, 11 images and a JavaScript program into DNA. The aforementioned items were converted into a 5.27 megabit bitstream and were encoded onto 54 898 oligonucleotides of 159 nucleotides. The encoding was done one bit per base and reached an information density of 5.5 petabits/mm<sup>3</sup>. In their work, each oligonucleotide consisted of a 96-nucleotide data block, a 19-nucleotide “address” code and flanking 22-nucleotide common sequences for amplification and sequencing, with the data eventually being recovered with 10 bit errors.

Church’s team used adenine or cytosine to represent zeroes, and guanine or thymine to represent ones, which led to sequencing error prone long stretches of the same base. In 2013 Goldman et al. [18, 19] used a more complex cipher to encode in DNA all 154 of Shakespeare’s sonnets, 26-second excerpt from the audio recording of Martin Luther King’s 1963 “I have a dream” speech, an image, a scientific paper and the code used in the study. They converted the data into base-3 with a Huffman code, and converted the result into DNA by replacing each digit with one of the

three nucleobases that were different to the previous one. This assured that the entire DNA sequence contained no consecutive same bases. In addition, the method created 100-nucleotide segments with 75-nucleotide overlaps, essentially causing each nucleotide to be repeated four times, two of which were reverse complemented for added data security. The team reported a storage density of 2.2 petabits per gram and 100% reconstruction of the original files.



## 4 Discussion

### 4.1 Benefits of Parallel Computation

Because the method used by Adleman (discussed in Section 3.3) utilizes the unique base-pairing properties of DNA, it is fundamentally different from traditional logic-gate based computation. The advantage of using DNA to solve this type of problems is *parallel computation*, a property of being able to “take multiple paths simultaneously” instead of trying each possible path individually (making it ideal for solving NP-complete problems). This specific type of problem solving is practically impossible for traditional TTL-gate computers.

Adleman’s entire 1994 Hamiltonian path solving experiment took “seven lab days” [3]. According to Adleman, if the ligation of two DNA molecules is considered a single operation, about  $10^{14}$  operations were executed. In computing, a common measure of computer performance is *floating point operations per second* (flops). The best modern supercomputers, such as the *Summit* by IBM [20, 21], can reach  $10^{17}$  flops.

To get a very rough estimate for the performance in Adleman’s 1994 experiment, I estimated “seven lab days” to be equal to  $7 \times 8$  hours, thus giving *the entire experiment* a performance of

$$\frac{10^{14} \text{ operations}}{7 \times 8 \text{ h}} \approx 10^8 \text{ operations/s.} \quad (1)$$

Albeit a rough estimate, this does show that for a small number of operations, DNA is slow compared to modern supercomputers. However it should be noted that the “seven lab days” included the entire work, from which the ligation was only one part of. The amount of substance used is directly proportional to the operations completed, and scaling up this experiment (i.e. increasing the amount of DNA used) would not significantly impact the ligation time, thus increasing the performance.

Since the hydrolysis of a single molecule of adenosine triphosphate provides the Gibbs free energy ( $\delta G = -8 \text{ kcal mol}^{-1}$ ), 1 joule of energy is sufficient for  $2 \times 10^{19}$

operations [22]. Energy efficiency of computers is often measured in *flops per watt* (which is essentially *operations per joule*), for which the current supercomputer records stand at  $10^{11}$  flops/W [23], and at the time of Adleman's 1994 experiment,  $10^9$  operations per joule [3]. It should also be noted that due to the second law of thermodynamics, the theoretical maximum of irreversible operations is  $34 \times 10^{19}$  operations per joule, to which the energy efficiency of DNA comes very close to.

## 4.2 Superior Data Storing Capabilities of DNA

As discussed in Section 2.3, modern digital data storage does not last very long, generally surviving no longer than  $10^2$  years [7, 8]. DNA, on the other hand, has at best survived for  $10^4$  years in the wilderness, and with proper handling can undoubtedly reach the same or longer age [18]. With two orders of magnitude longer age, DNA is superior to any modern storage device in terms of data life expectancy.

When it comes to information density, DNA also flourishes. Modern flash memory at best has an information density of below  $10^{10}$  bits per  $\text{mm}^3$ . While information density of  $10^{13}$  bits per  $\text{mm}^3$  has been reached with quantum holography, current records of information density in DNA are at  $10^{16}$  bits per  $\text{mm}^3$ , surpassing rival data storing formats by three orders of magnitude. [9, 19]

## 4.3 Risks

DNA, being a biological material, is susceptible to radiation damage. Ultraviolet radiation of the sun is the most significant part of sunlight that causes skin cancer in humans. In order to achieve effective radioprotection, the mechanisms of the interaction between radiation and biomolecules need to be better understood. Czapla-Masztafiak et al. [24] made progress in this field in 2015 with research that provides information about the damage types and the changes in electronic structure around the phosphorus atoms associated with each damage type. The research was conducted using x-ray spectroscopy to investigate the structure of DNA lesions caused by irradiation by UVA light and protons.

Computer viruses are a common nuisance in modern computer softwares. In 2017, Ney et al. [25] exposed privacy and security problems associated with commonly used open-source bioinformatics tools for DNA sequencing, which were found to contain security vulnerabilities such as little input sanitization and overflowable static buffers.

They demonstrated that the sequencing process of DNA can be exploited to reveal sensitive information or to inject data. By synthesising a DNA strand containing a malicious computer code, “a virus”, encoded in the bases of the DNA strand, they were even able to gain remote control of the computer processing the DNA during sequencing. The carelessness in security in many bioinformatics tools might be due to a lack of widespread DNA viruses, at least for now.



## 5 Conclusion

DNA shows prowess in energy-efficient parallel computation and potential in logic gates, but is likely not surpassing modern computers in terms of every day use. DNA computing is based on biological and chemical reactions, whereas TTL-technology is based on electricity, physical reactions. Despite DNA being able to perform huge amounts of parallel computations, its initial preparation is still very time consuming, even for small problems. A computer must not only be able to quickly solve big calculations, but small ones too. However, DNA logic gates might be useful for biological work, such as detecting viruses.

In terms of data storage, writing information in DNA and especially reading information from DNA is slow, surpassed in speed by all electronic digital data storage. Despite slow writing and reading, data stored in DNA exceeds all competitors in terms of information density and life expectancy, making it suitable for long-term archiving purposes, in which the two aforementioned properties are crucial, but speed less so.



## References

- [1] Clarivate Analytics. *Web of Science*. URL: <https://apps.webofknowledge.com> (visited on 09/25/2019).
- [2] J. B. Reece et al. *Campbell Biology Ninth Edition*. San Francisco: Pearson, 2011.
- [3] L. M. Adleman. “Molecular Computation of Solutions to Combinatorial Problems”. *Science* 66 (1994), pp. 1021–1023.
- [4] I. Bello, J. R. Britton, and A. Kaul. *Topics in Contemporary Mathematics Ninth Edition*. Belmont: Cengage Learning, 2009.
- [5] C. J. W. Kloesel. *Writings of Charles S. Peirce*. Bloomington and Indianapolis: Indiana University Press, 1989.
- [6] Harry Ransom Center - University of Texas at Austin. *Early Writing*. 2015. URL: <http://www.hrc.utexas.edu/educator/modules/gutenberg/books/early/> (visited on 12/10/2018).
- [7] The CD Information Center. *Longevity of CD/DVD Media*. 2016. URL: <http://www.cd-info.com/archiving/longevity/index.html> (visited on 09/08/2018).
- [8] National Instruments. *Understanding Life Expectancy of Flash Storage*. 2018. URL: <http://www.ni.com/product-documentation/10126/en/> (visited on 09/08/2018).
- [9] G. M. Church, Y. Gao, and S. Kosuri. “Next-Generation Digital Information Storage in DNA”. *Science* 337 (2012), pp. 1628–1629.
- [10] T. R. Kuphaldt. *Lessons in Electric Circuits - Logic Signal Voltage Levels*. 2001. URL: <https://www.allaboutcircuits.com/textbook/digital/chpt-3/logic-signal-voltage-levels/> (visited on 05/21/2018).
- [11] A. P. de Silva and N. D. McClenaghan. “Proof-of-Principle of Molecular-Scale Arithmetic”. *Journal of American Chemical Society* 122 (2000), pp. 3965–3966.

- [12] A. Okamoto, K. Tanaka, and I. Saito. “DNA Logic Gates”. *Journal of American Chemical Society* 126 (2004), pp. 9458–9463. DOI: 10.1021/ja047628k.
- [13] M. N. Stojanovic, T. E. Mitchell, and D. Stefanovic. “Deoxyribozyme-Based Logic Gates”. *Journal of American Chemical Society* 114 (2002), pp. 3555–3561. DOI: 10.1021/ja016756v.
- [14] D. Y. Tam et al. “A Reversible DNA Logic Gate Platform Operated by One- and Two-Photon Excitations”. *Angewandte Chemie International Edition* 55 (2016), pp. 174–168. DOI: 10.1002/anie.201507249.
- [15] P. Vijayakumar and J. Macdonald. “A DNA Logic Gate Automaton for Detection of Rabies and Other Lyssaviruses”. *Chemphyschem : a European journal of chemical physics and physical chemistry* 18(13) (2017), pp. 1735–1741. DOI: 10.1002/cphc.201700072.
- [16] Y. Liu et al. “DNA Solution of a Graph Coloring Problem”. *Journal of Chemical Information and Modeling* 42 (3) (2002), pp. 424–528. DOI: 10.1021/ci010016o.
- [17] E. W. Weisstein. *Chromatic Number, MathWorld—A Wolfram Web Resource*. URL: <http://mathworld.wolfram.com/ChromaticNumber.html> (visited on 10/02/2019).
- [18] E. Young. “Synthetic double-helix faithfully stores Shakespeare’s sonnets”. *Nature/News* (2013). DOI: 10.1038/nature.2013.12279. (Visited on 09/25/2019).
- [19] N. Goldman et al. “Towards practical, high-capacity, low-maintenance information storage in synthesized DNA”. *Nature* 494 (2013), pp. 77–80. DOI: 10.1038/nature11875.
- [20] IBM. *IBM Power System AC922*. 2018. URL: <https://www.ibm.com/us-en/marketplace/power-systems-ac922/details> (visited on 09/08/2018).
- [21] TOP500. *Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband*. 2018. URL: <https://www.top500.org/system/179397> (visited on 09/08/2018).
- [22] J. D. Watson et al. *Molecular Biology of the Gene, ed 3*. Menlo Park, CA: Benjamin/Cummings, 1987.
- [23] GREEN500. *June 2018*. 2018. URL: <https://www.top500.org/green500/lists/2018/06/> (visited on 09/08/2018).



- [24] J. Czapla-Masztafiak et al. “Investigating DNA Radiation Damage Using X-Ray Absorption Spectroscopy”. *Biophysical Journal* 110 (2016), pp. 1304–1311. DOI: 10.1016/j.bpj.2016.01.031.
- [25] P. Ney et al. “Computer Security, Privacy, and DNA Sequencing: Compromising Computers with Synthesized DNA, Privacy Leaks, and More”. *USENIX Security Symposium*; 26 (2017), pp. 765–779.