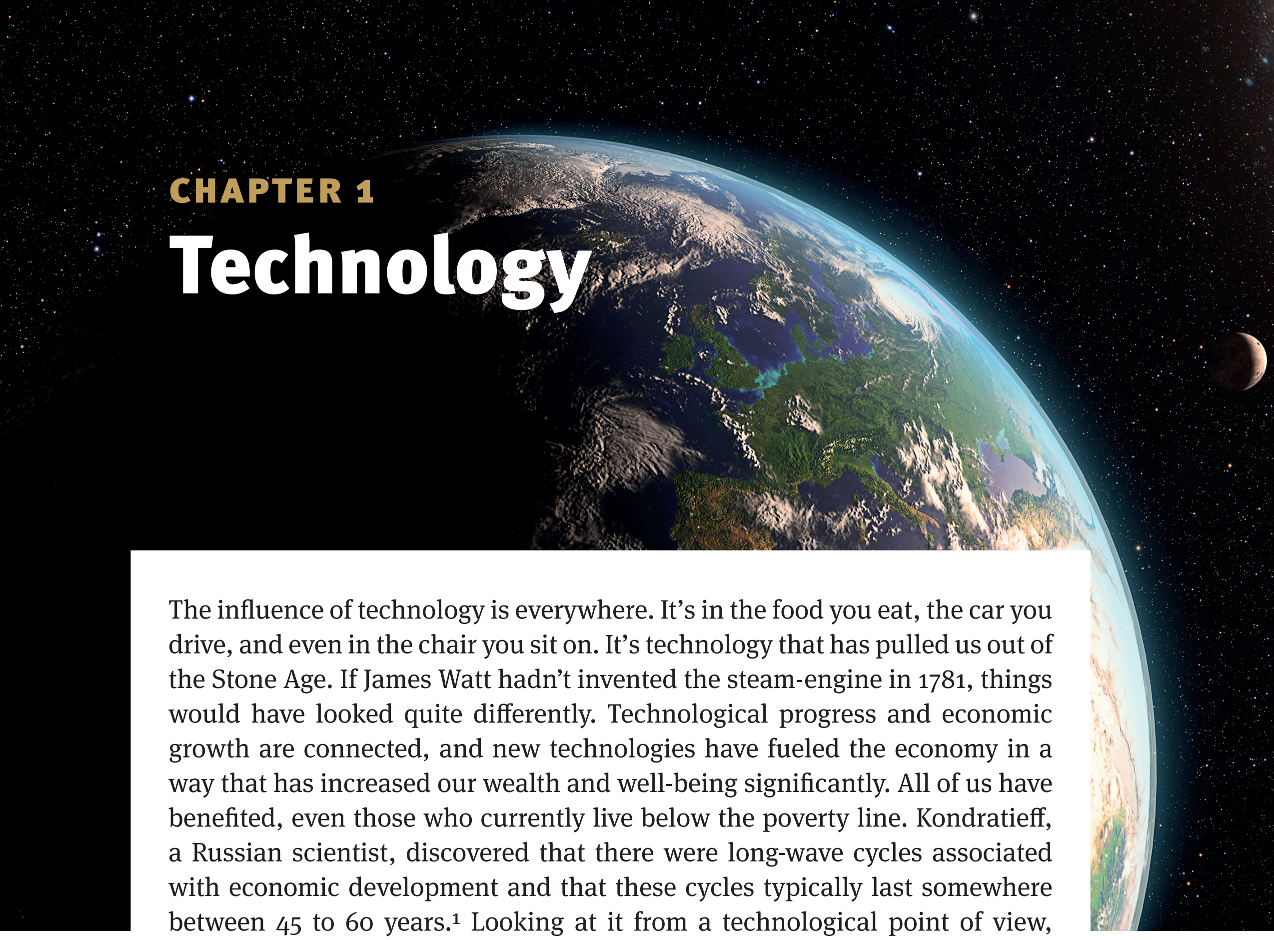
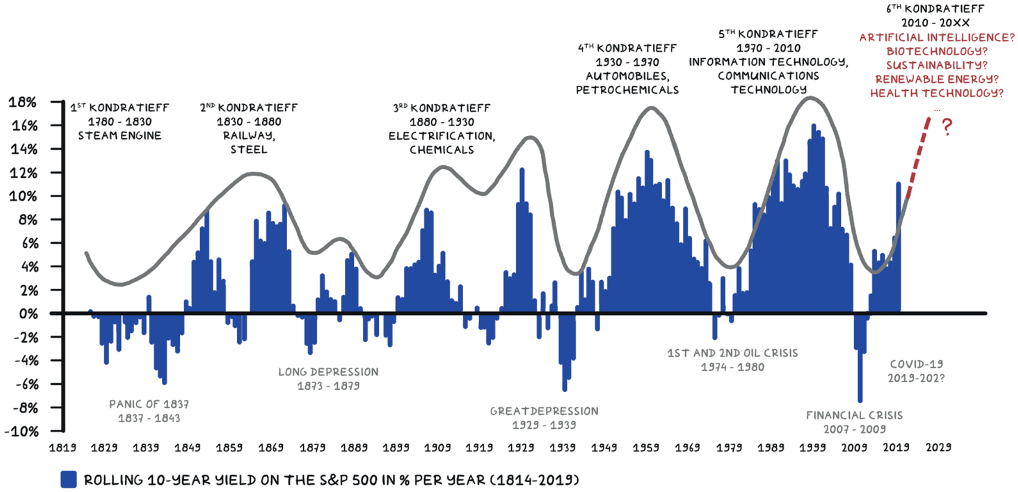


CHAPTER 1

Technology

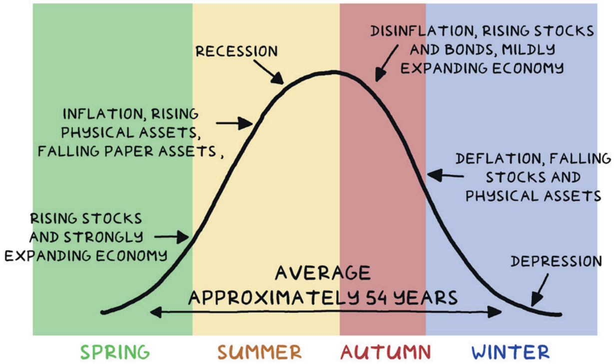
The influence of technology is everywhere. It's in the food you eat, the car you drive, and even in the chair you sit on. It's technology that has pulled us out of the Stone Age. If James Watt hadn't invented the steam-engine in 1781, things would have looked quite differently. Technological progress and economic growth are connected, and new technologies have fueled the economy in a way that has increased our wealth and well-being significantly. All of us have benefited, even those who currently live below the poverty line. Kondratieff, a Russian scientist, discovered that there were long-wave cycles associated with economic development and that these cycles typically last somewhere between 45 to 60 years.[1] Looking at it from a technological point of view, humanity has already gone through as many as five waves (F1.1a).



ⓐ

1 (Kondratieff, 1926) (Kondratieff & Stolper, 1935)

ⓑ



F1.1 ⓐ **Kondratieff long-waves.** ⓑ **Idealized Kondratieff Cycle.**
Source: ⓐ Adapted from: (Allianz Global Investors, 2010). ⓑ Adapted from: (Prechter, 2002).

Each wave can be divided into four phases (F1.1b): spring (improvement), summer (prosperity), fall (recession), and winter (depression). Looking at F1.1a, we can see that historically, each new wave of innovation has started after a great depression. This isn't strange, as in times of despair people try to find new ways to move forward. For us the last recession took place around 2008: the financial crisis. In 2020, the COVID-19 pandemic, also known as coronavirus disease, has added fuel to the fire. We are currently riding what is being called the sixth Kondratieff wave, or, from a slightly different perspective – the **Fourth Industrial Revolution.**[2] To find out what it's all about, we need to understand what it thrives on. To do that, we need to step back in time, just briefly, and look at what happened in the previous wave: the age of information and communication technology, otherwise known as the rise of computer power.

1.1 Riding the wave of raw computer power

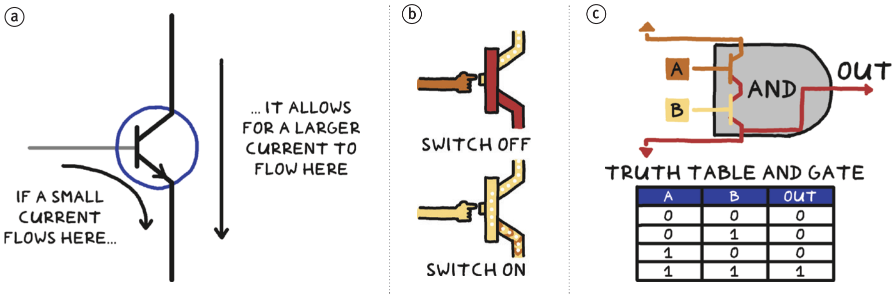
In the 1930s, digital computers had not yet been invented, and words like *processor* or *data storage* were simply unheard of. Calculations were mostly done by hand, and only a few simple, specific-purpose mechanical machines existed. Alan Turing and John von Neumann, both mathematicians with broad interests, were two major contributors to the founding of computer science.

In 1936, Turing developed a theoretical mathematical model of computation called a **Turing Machine.**[3] At the time it was brilliant, so brilliant in fact, that

2 The steam engine led to the first Industrial Revolution, electricity to the second, and information and communication technology to the third.
3 A Turing Machine, which consists of, in theory, a tape for storage, a head for reading and/or writing the tape, a state register to determine the current state of the machine, and a table of instructions. Taking into account the state the machine is in, the instructions indicate what to do next, for example, move the head or write a symbol. Turing showed that a universal Turing

today's computers work in much the same way: they are general machines that can run all kinds of programs; their memory contains instructions as well as data. During the Second World War, Turing[4] considered how the theoretical Turing Machine could be approximated by a physical machine. It was Von Neumann, however, in 1945, who described an architecture for an electronic digital computer,[5] putting Turing's theoretical ideas into practice.[6] What we now call the Von Neumann architecture is at the heart of every microprocessor. Turing, who worked with Von Neumann earlier, helped design one of the first stored-program digital computers at the Victoria University of Manchester in 1948: The Manchester Mark 1. In those days, a computer would fill up an entire room. However, things started to change rapidly when it became possible to scale down the most essential part of a processor: the transistor (F1.2a).

A **transistor** can be viewed as a switch (F1.2b). It holds information, a state, when it blocks electrons or lets them through, thereby creating a position *0* or *1*. By combining transistors, one can create **logic gates**. These logic modules have inputs and outputs. For example an *AND* gate (F1.2c) requires all inputs to be *1* (for example, input A and B) to create a *1* at the output. When basic logic gates are combined, one can perform arithmetical functions like adding, subtracting, and multiplying. It is these units that form the building blocks of a computer.



F1.2 ⓐ **Transistor schematic.** ⓑ **Transistor function.** ⓒ **Logic AND gate.**

To advance technology, the main challenge for the semiconductor industry is to make the processors, currently made from silicon, more powerful and energy efficient. To achieve this, they aim to make the most crucial part, the transistor, increasingly smaller. It goes without saying that looking back over

Machine that emulates all Turing Machines by storing data, as well as instructions on the tape, could be built (Turing, 1937).

4 At that time, Turing was codebreaking the Enigma code for the British Secret Intelligence Service. For inspiration see the movie: The Imitation Game (2014).

5 (Von Neumann & Alexander, 1945)

6 The architecture consists of things such as a Store (memory), Control (CPU), an ALU (Arithmetic Logic Unit), and of course, input and output mechanisms. Nowadays, the ALU has become part of the CPU (Central Processing Unit).

the last 50 years, they have been quite successful. In less than 50 years, they have accomplished an increase from 3500 transistors on a single dice (chip) to over eleven billion transistors.[7] The reality is that just one state-of-the-art smartphone now has more computing power than was available on the entire planet back in the 1950s.

Gordon Moore, the unassuming billionaire and co-founder of Intel Corporation, described the way the growth of transistors would emerge, nowadays referred to as **Moore's law**. Moore described the trend in a paper in 1965, where he stated that transistors in a dense integrated circuit would roughly double every 18-24 months.[8] For 50 years, the law has been ridiculously accurate.

Despite all kinds of technological challenges, the semiconductor industry has been keeping up with Moore's law by using various new innovative materials and designs.[9] During an interview in 2007, Moore stated that there would be an end to his law, going on to quote Stephen Hawking, who visited Intel in 2005. When Stephen Hawking was asked what the fundamental limits to microelectronics are, he said "the speed of light and the atomic nature of matter."[10] Eventually, atomic limitations and things such as heat dissipation, current leakage, and thermal noise will prevent silicon-based integrated circuits from approaching quantum mechanics, something we will explain further shortly.

Nowadays, we already see a slowing down of Moore's law, and the semiconductor industry is looking for new ways to enhance silicone with **nanotechnology**.[11] In 2020, we are still scaling down the production node of chips, and the large-scale production of 5nm (nanometer) chips has already begun, large scale 4nm production is planned to take off in 2023.[12] To provide some explanation, in general, the smaller the node, the smaller the transistor and the spacing between features. Using new materials, scientists can even create transistors as small as 1nm,[13] but the question remains if it would be possible and commercially cost-effective to get billions of such tiny transistors on a

7 For example, back 1972, an 8008-processor had 3500 transistors. In 2020 an Apple A14 Bionic processor (5nm) had 11,8 billion.

8 (Moore, 1965)

9 For example, materials like hafnium, strained silicon, high-k metal, and designs like 3D tri-gate transistors.

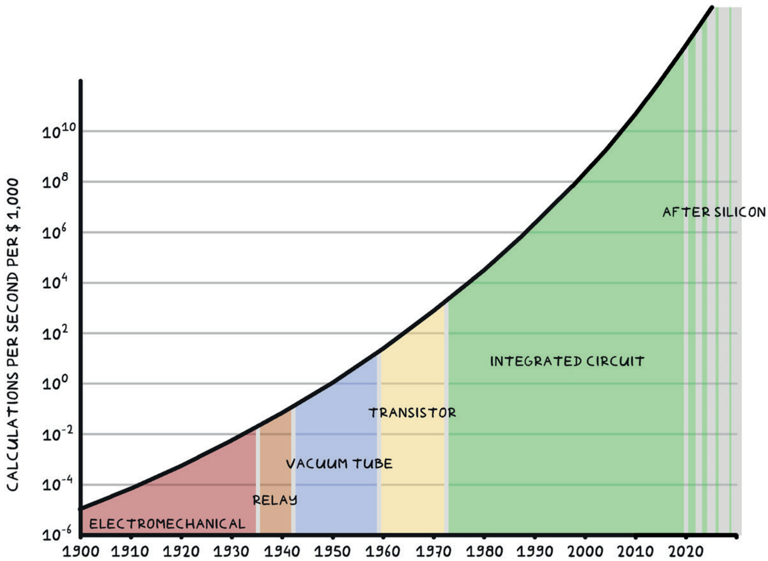
10 Moira Gunn conducted the interview, at Intel's twice-annual technical conference on September 18, 2007 (Forit, 2007).

11 There are high hopes to use Carbon Nano Tubes (CNTs, see section 1.4 for more information) as the connection between transistors. CNTs are only one nanometer wide and could increase the clock speed of processors, something that has barely risen over the past decade. It is, however, difficult to properly connect the CNTs with the transistors on a large scale. Stacking circuits in 3D could also offer temporary relief but stacking circuits too high could result in unwanted high temperatures, which would burn out the circuitry.

12 Based on the future R&D plans of TSMC. Also, the test production of 3nm expects to start in 2022; it will, however, take some time before large scale production will become possible.

13 (Desai et al., 2016)

single chip. One way or another, it's expected that enhancing silicone with nanotechnology won't be enough to keep us moving forward. The result is that the exponential growth of Moore's law will flatten out. But would this mean the end of the exponential growth of computing? No, it wouldn't. This is because Moore's law isn't a law of physics but rather a law of economics. Futurist Ray Kurzweil already noticed this by looking at the increase of computer power from another perspective and introducing **The Law of Accelerating Returns**.[14] This law says that at a price/performance level, the growth of computer power is exponential.



F1.3 **Kurzweil and The Law of Accelerating Returns.**
Source: Adapted from: (Ray Kurzweil, 2001).

Kurzweil believes that we will keep shifting from one technology to another. Just like when we moved from vacuum tubes to transistors to integrated circuits. We have changed from one technology to another by using technology itself to create the next. Kurzweil claims that Moore's law is just another phase in technological development that will eventually run out of steam and that we will shift to another paradigm that will further exponentially increase the price/performance ratio of computer power. If you look at the data, it sure looks like he's right (F1.3).

Moore's law is losing momentum, and without having a real alternative in place, in theory our economy could eventually be impacted significantly. In the last decades, our economy has been built upon computer power, and it is the speed at which computers have gained more power that has been a major

14 (Ray Kurzweil, 2001) (R. Kurzweil, 2005)

driver behind our economic growth. But who will buy a new smartphone or laptop when they know it isn't going to be any more powerful than last year's model? Yes, Silicon Valley (aptly named after the material itself) would run out of steam, and we need a replacement if we want to keep up the pace. The hunger for more powerful computers isn't only an economic hunger but also a necessity if we want to solve complex problems, something we will address further on. But what will the post-silicon era look like?[15] To explore in which way we might be heading we will address four developments that could set the stage for the future of computing: quantum computers, DNA computers, chemical computers, and photonics.[16] Further on, when we discuss Artificial Intelligence, we will also introduce neuromorphic computers.

Quantum computers

As shown in F1.2, a transistor can be viewed as a switch. It holds information as it blocks or passes through electrons, thereby creating a position *0* or *1*. However, when transistors become just a few atoms thick, we move to the quantum level. At this level, electrons don't obey the rules as we know them. They can pass through the transistor, even when it is in a blocked state. This process, called quantum tunneling, makes it impossible to create a silicon processor that will do what we expect it to do. As a consequence, the node limit of commercial silicon processors will probably range from 2nm – 3nm by using nano enhancements. In **quantum computers** we use the specific properties that only occur on a quantum level to our benefit. To understand how they work you will need to know a bit of quantum mechanics, in particular, two properties: superposition and entanglement.

In standard computers information is stored in bits, which can be *0* or *1*. In quantum computers, information is stored in **qubits**, which can also be *0* and *1*. However, qubits in **quantum superposition** are *0* and *1* at the same time, as long as they are not observed; otherwise, they would collapse to a *0* or *1*.[17] This might be difficult to get your head around, so let's make it more practical: imagine you are one out of a hundred guests at a party. To find out who you could have a pleasant conversation with, you could shake hands with all of the guests, one at a time, and then decide who to speak with. In quantum computing, you would be shaking hands with all 99 people at once, and the moment you ask yourself the question who would be the best conversation partner, you instantly end up shaking hands with only that person. In terms of a typical computer that looks at a database, this means that at

15 Some expectations are that the post-silicon era will be on the biomolecular level (Adleman, 1994).

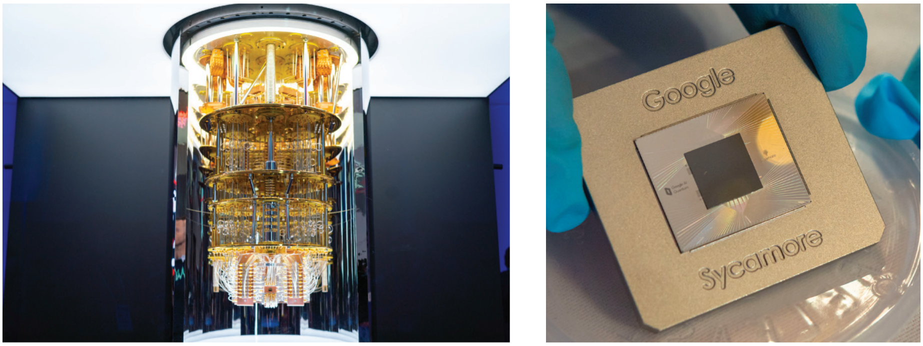
16 Another type of development is a molecular computer. Because the size of molecules needed exceeds the current size of transistors in silicon, it shouldn't be expected that molecular computers will replace silicon-based processors.

17 The Heisenberg uncertainty principle says that the position and speed of a particle (or the energy and time of a wave) is uncertain.

worst, it needs to test every entry. A quantum computer can easily handle this because it can look at all the entries at the same time, massively managing information in parallel. With enough qubits, you could easily have something that could compete with a billion-core machine made out of standard silicon processors.[18]

The second valuable property used in quantum computers is **quantum entanglement**, which is a special state of superposition. When two qubits are entangled, they connect to each other instantly, no matter how far they are apart.[19] This means that when observing one qubit, you can directly deduce the properties of its partners without having to look. Through entanglement, we can build quantum logic gates out of qubits, the building blocks of a computer.

ⓐ ⓑ



F1.4 ⓐ Quantum computer: IBM-Q SystemOne. ⓑ The Sycamore processor containing 54 qubits made by Google.

Sources: ⓐ IBM/Andrew Lindemann. ⓑ DPA Picture Alliance / Alamy Stock.

Currently quantum computers are still in the experimental stage. Building a fully working quantum computer that solves real problems is extremely difficult. The difficulty of quantum computers has to do with two things. First, to get down to the quantum properties that are needed, the computer needs to be supercooled to get it into a superconducting state, a state where electrons can move freely without resistance. Secondly, the time a qubit can hold its state (the coherence time) is currently very short. So, making a complex program results in the whole thing collapsing before coming up with an answer. Thus, quantum computers will not be found in any laptop soon and probably never

18 For example, if you have four bits, you can create 16 possibilities (2^4). A quantum computer with 54 qubits, can be in 2^{54}=18,014,398,509,481,984 positions at once! That's like doing well over 18,000 trillion calculations at the same time, and when set up correctly, the right answer will come up as soon as you ask the question.

19 Einstein famously described this as "spooky action at a distance" back in 1930 (Nikolic, 2012). The property of entanglement can also be used to create encrypted data channels; one day, these could be used to build a quantum version of the Internet (Yu et al., 2020).

will. Quantum computers are mostly used for special purposes, and they look a bit like they did back in the 1950s, where rooms full of equipment were needed to build a single computer (F1.4a).

Because of the incredible potency of quantum computers, companies like IBM,[20] Google, IonQ, D-Wave and Rigetti are investing heavily in their development. Being first to market would not only make you super-rich and extremely powerful, but it would also change the world in the same way the first computer did when it was introduced back in the 1950s. Quantum computers will boost advanced climate modeling, biotech, simulation of complex chemistry, and things like Artificial Intelligence. They could easily hack security keys in a matter of seconds instead of years.

In 2019, Google had a major breakthrough. They claimed to be the first to reach **quantum computer supremacy**,[21] which means demonstrating that you can solve a problem no classical computer feasibly can. Their quantum computer, called Sycamore (F1.4b), demonstrated this by performing a calculation in three minutes and 20 seconds where the world's supercomputer Summit would have needed 10,000 years. IBM responded to this claim by stating that an ideal simulation of the same task can be performed on a classical system in two and a half days with far greater fidelity.[22] So besides Google's quantum computer being a huge step forward, it isn't the supremacy some might hope for, it's more of a quantum advancement. A real breakthrough that enables the use of quantum computing for practical applications could occur in just a matter of years, but it's more likely to be decades away. For now, quantum computing's performance is expected to grow exponentially, just like Moore's law did.

DNA computers

DNA is spectacular. Almost every single cell in our body has DNA in its center (the nucleus) that contains information not only about how to build that cell but also the complete blueprint of how to build a human.[23]

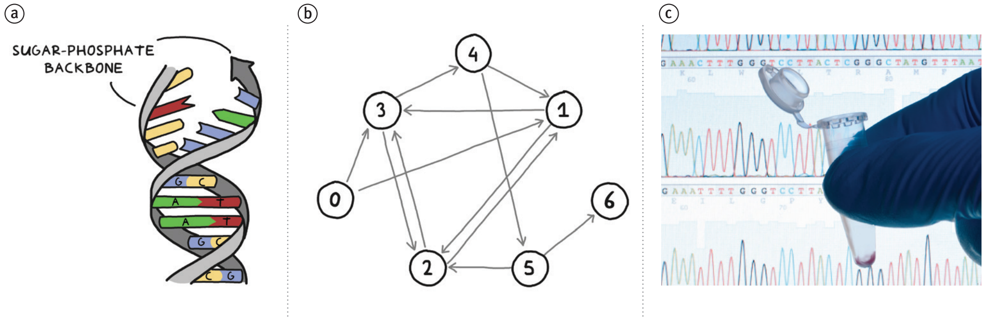
DNA can be found in animals and plants and is stored in a code that consists of four chemical bases: adenine (A), guanine (G), cytosine (C), and thymine (T). Our human DNA is like a string of text that is about 6.4 billion characters long, where the 3.2 billion bases A, G, C, and T are combined in a unique sequence. Can you imagine that all this information is available in almost every cell of your body? DNA is an incredibly compact information carrier!

20 If you want to build your own quantum programs and test them on a real quantum computer, check out https://quantumexperience.ng.bluemix.net/qx/experience.

21 (Arute et al., 2019)

22 (Pednault, Gunnels, Maslov, & Gambetta, 2019)

23 All humans share 99% of the same DNA.



F1.5 ⓐ **DNA double helix structure.** ⓑ **Hamiltonian path problem.** ⓒ **The pink substance in the test tube is DNA that can store an incredible amount of data.**

Sources: ⓑ Adapted from: (Adleman, 1994). ⓒ Damerau / Shutterstock.

In human DNA, bases pair up to form units called base pairs (A with T and G with C). Each pair is attached to a sugar-phosphate backbone, and together they form something a bit like a ladder that is called a double helix (F1.5a). This is an essential property because DNA can replicate itself using one strand of the double helix when the cell divides to create an exact copy. Copies of DNA can be made exponentially; one creates two, two creates four, four creates eight, etc. Besides that, DNA stays stable for a very long time, especially when it is kept in a cold, dark, and dry space. The current record of sequencing DNA is on an ancient horse that lived 700,000 years ago.[24]

So how can we use these amazing properties of DNA to our benefit? First, you can make complex computations using DNA (not computations on DNA, **DNA computers** are an entirely different concept compared to the regular computers we use today). In 1994, Leonard Adleman demonstrated that DNA computing was possible.[25] He solved the traveling salesman problem, also known as the Hamiltonian path problem. The objective was to find the most effective flight route through seven cities connected by 14 one-way flights (F1.5b). It can easily be solved by pencil and paper for a small number of cities. However, it explodes when a large number of cities are considered. Even supercomputers can't solve it. The way Adleman solved the problem was by using parallelism. He created massive amounts of short DNA fragments that represented each city. Each end of the DNA fragments is sticky, meaning that they become stuck to one another in long sequences. Mixing everything together in a test-tube, it took less than an hour to create every possible route. Then, through controllable chemical processes, the DNA sequence that started and ended with the same city was filtered out, followed by the ones that had the correct number of stops. The result was DNA strands that contained the most optimal solutions to the problem.

24 (Orlando et al., 2013)
25 (Adleman, 1994)

The application of DNA computing is still being explored. It is useful for cracking even the toughest combinational problems through massive parallelism by using trillions of DNA strands. The computing, however, is slow and it can take weeks to complete the chemical processes needed to extract solutions. DNA computing will, therefore, not be an alternative to silicon-based computing anytime soon. An application of DNA computing is medical diagnosis and treatment. In this case, DNA computers can already (on a very small scale) identify cancer cells and release a drug straight into them.[26]

Another property of DNA is that it stores information in a really compact way (F1.5C) and that it remains effective for a very long time. Besides that, DNA itself is cheap, clean, and very energy-efficient. The future of using DNA as a medium for long-term archiving looks bright.

In 2017, Yaniv Erlich and Dina Zielinski successfully developed and applied a new storage strategy, called the DNA Fountain. They were able to store a full computer operating system, a movie, and other files on DNA and obtained a perfect retrieval from a density of 215 Petabyte per gram of DNA.[27] In other words, they could store 215 million Gigabytes of data in just a single gram of DNA! Erlich: “We need about ten tons of DNA to store all the world’s data. That’s something you could fit on a semi-trailer.”[28]

DNA is perfect for long-term archiving, it is not going to be obsolete in 10,000 years, and you can make a copy of an entire data center for one dollar in one hour. But writing DNA is expensive, and reading data from DNA is slow (measured in minutes, hours, and days instead of milliseconds). So, all that considered, our next-generation hard drive will probably have no DNA in it.

Chemical computers and Photonics

If you take into consideration that temperature and coherence are still a challenge for quantum computers and that DNA computers are really slow, what else is out there that may be an interesting bet to get ahead in creating more computer power?

First, let us take a look at **chemical computers**, which are based on Belousov-Zhabotinsky (BZ) reactions, developed by Lee Cronin.[29] A BZ-reaction happens when certain chemicals are stirred, resulting in a short flash when the color of the reaction changes. It is like an oscillating computer clock that ticks

26 (Benenson, Gil, Ben-Dor, Adar, & Shapiro, 2004)
27 (Erlich & Zielinski, 2017)
28 (Cornish, 2018)
29 (Cronin et al., 2019)