

DNA Computing: Challenges and Application

<https://doi.org/10.3991/ijim.v11i2.6564>

Samir Abou El-Seoud
British University in Egypt, Cairo, Egypt
samir.abouelseoud@bue.edu.eg

Reham Mohamed*
British University in Egypt, Cairo, Egypt
reham.mohamed@bue.edu.eg

Samy Ghoneimy
British University in Egypt, Cairo, Egypt
samy.ghoneimy@bue.edu.eg

Abstract—much of our scientific, technological, and economic future depends on the availability of an ever-increasing supply of computational power. However, the increasing demand for such power has pushed electronic technology to the limit of physical feasibility and has raised the concern that this technology may not be able to sustain our growth in the near future. It became important to consider an alternative means of achieving computational power. In this regard, DNA computing was introduced based on the usage of DNA and molecular biology hardware instead of the typical silicon based technology. The molecular computers could take advantage of DNA's physical properties to store information and perform calculations. These include extremely dense information storage, enormous parallelism and extraordinary energy efficiency. One of the main advantages that DNA computations would add to computation is its self - parallel processing while most of the electronic computers now use linear processing. In this paper, the DNA computation is reviewed and its state of the art challenges and applications are presented. Some of these applications are those require fast processing, at which DNA computers would be able to solve the hardest problems faster than the traditional ones. For example, 10 trillion DNA molecules can fit in one cubic centimeter that would result in a computer that holds 10 terabytes of data. Moreover, this work focuses on whether a large scale molecular computer can be built.

Keywords—DNA Computing, Quantum Computation, Biocomputer, Molecular Computation.

1 Introduction

The amount of information gathered on the molecular biology of DNA over the last 40 years is almost overwhelming in scope. However, for this work we would avoid

getting bogged down in too much biochemical and biological details. So we will concentrate only on the information most relevant to DNA role in computing.

1.1 The structure of DNA and its role in the cell

Deoxyribonucleic acid (abbreviated to DNA) is the molecular structure which provides the genetic instructions used in the development and for the functioning of a cell of all known living organisms. The three-dimensional structure of DNA, first proposed by James D. Watson and Francis H. C. Crick (Watson J, Crick F 1953) [2] about 50 years ago, consists of two long helical strands that are coiled around a common axis forming a double helix. The double-helical structure of DNA as shown in figure (a), one of nature's most magnificent constructions, is critical to the phenomenon of heredity, the transfer of genetically determined characteristics from one generation to the next.

The main role of DNA molecules is the long-term storage of information. DNA is often compared to a set of blueprints, or a code, since it contains the instructions needed to construct other components of cells, such as proteins molecules. Chemically, DNA consists basically of two long polymers of simple units called nucleotides. Each nucleotide consists of a sugar (deoxyribose) with, attached to it, a phosphate and one of four types of molecules called bases. These four types of nucleotides bases are: Adenine (A), Thymine (T), Guanine (G), and Cytosine (C). It is the sequence of these four bases along the backbone that encodes information in an alphabet called the genetic code. The DNA information is read using that genetic code. Nucleotides can be stuck together as base-to-base. These base-pairings are specific: that is, adenine will bind with thymine and only thymine; thymine will bind with adenine and only adenine; cytosine will bind with guanine and only guanine; guanine will bind with cytosine and only cytosine.



Fig. 1. The double helix of DNA

This means that we can bind together two chains of nucleotides base-to-base, so long as the chains are complementary: that is, where one has adenine (A), the other has thymine (T), and where one has guanine (G), the other has cytosine (C), as shown in the diagram. Therefore every DNA sequence has a natural complement. For example if sequence S is as follows in Figure (2):

C	G	A	G	C	T	A	T	T
G	C	T	C	G	A	T	A	A

Fig. 2. Example for DNA sequence

CGAGCTATT, its complement, S', is GCTCGATAA. Both S and S' will come together (or hybridize) to form double stranded DNA. The double-helix comes about this way. Those two strands entwine in the famous double-helix fashion. Because the bonds between bases are hydrogen bonds, they are fairly weak and easy to break, making it quite easy to "unwind" DNA into its two complementary strands. Some DNA binding enzymes can recognize specific base pairing patterns that identify particular regulatory regions of genes. Since cells use proteins (enzymes) to make other molecules like sugars or fats, DNA indirectly directs the synthesis of many small molecules as well as proteins. DNA also contains a coded set of instructions about when various proteins are to be made and in what quantities. Recently many researchers have found that the information contained in the base sequence of DNA molecules opens the possibility of using them as computational building blocks to store non-biological data. Moreover, it can pack that information into far smaller packages than is possible with solid state memory or the densest hard drive platters. Hence, in principle if we substitute the DNA bases A, C, G and T for 0's and 1's i.e. hardware, and substitute enzymes for software we have a DNA computing system. The fundamental set of operations that are needed is also provided by DNA. Those molecules can split, evolve and replicate themselves which provides us a huge set of operation like adding, ORing, etc.

2 Silicon vs DNA computing

(Adleman L. 1994)[1], has realized that the enzyme DNA polymerase, which replicates strands of DNA, works like a machine. As it moves down a strand, the polymerase creates a new one. The process is similar to the concept of Turing Machine: a device that would read a single bit of data off tape, performs some action on that data based on instructions stored in the machine, and then writes the result in the spot of the original data. DNA computing thus uses DNA strands to store information and uses properties of DNA to perform operations (interactions). Hence, a small test tube of DNA strands suspended in a solution could yield billions of such operations simultaneously, thus achieve a speed - in theory - faster than today's fastest supercomputers.

Adleman demonstrated experimentally that DNA can be used in finding a solution to the famous directed Hamiltonian Path (HP) problem using standard laboratory techniques. The problem is a variant of the so-called ‘traveling salesman problem’ (TSP): Given N cities, find a path visiting each and every city only once starting and ending at given point. For N cities there are $(N-1)!/2$ possible paths. As the number of cities grows, the number of possible path combinations increases rapidly. With the von Neumann computer, one simple method would be to set up a search tree, measure each complete branch sequentially, and keep the shortest one. Improvements could be made with better search algorithms (for larger N), such as pruning the search tree when one of the branches you are measuring is already longer than the best candidate. However, the basic solution would be to first generate all possible paths and then search the entire list. For example, if there are 9 cities, there are about 180,000 possible paths, and 17 cities would have about 200 trillion paths. To solve the problem for larger numbers of cities, brute force attempts to calculate all paths would quickly overwhelm even the fastest supercomputer. Also, the entire list of routes for a 20 city problem, for example, could theoretically take about 45×10^{15} Bytes of memory ($18!$ routes with 7 byte words) This one would take two years on a 100 MIPS supercomputer, just to generate all paths (assuming one instruction cycle to generate each city in every path). However, if using DNA computing was possible, this method becomes feasible. 10^{15} is just a nanomole of material. Also, routes no longer have to be searched through sequentially. Operations can be done all in parallel.

DNA, with its unique data structure and ability to perform many parallel operations, allows us to look at a computational problem from a different point of view (Ryu, W 2007) [3]. Silicon-based computers typically handle operations in a sequential manner. Although multi-processor computers and modern CPUs incorporate some parallel processing, the architecture is basically von Neumann. Computer instructions are handled extremely fast but basically sequential. DNA computers, however, are non-von Neumann stochastic machines that approach computation in a different way from ordinary computers.

Typically, increasing performance of silicon computing means faster clock cycles (and larger data paths), where the emphasis is on the speed of the CPU and not on the size of the memory. For DNA computing, though, the power comes from the memory capacity and parallel processing. For example, if we consider the read and write rates of DNA, DNA can be replicated at a rate of about 500 base pairs a second. Biologically this is quite fast and assuming low error rate, an impressive achievement. However, this corresponds to only 1000 bits/sec, which is quite slow when compared to the data throughput of an average hard drive. But if we allow many copies of the replication enzymes to work on DNA in parallel, the number of DNA strands increases exponentially (2^n after n iterations). With each additional strand, the data rate increases by 1000 bits/sec. So after 10 iterations, the DNA is being replicated at a rate of about 1Mbit/sec; and after 30 iterations it increases to 1000 Gbits/sec. This is beyond the sustained data rates of the fastest hard drives.

Adleman’s demonstration is a simple version of TSP, for only $N = 7$. Nevertheless, his work is significant for a number of reasons (Ryu, W 2007) [3]. • It was the first example of computation at a molecular level. • It illustrates the potential of using

DNA to solve a class of problems that is difficult or impossible to solve if using traditional computing methods. • It demonstrates unique aspect of DNA as a data structure. • It demonstrates that computing with DNA can work in a massively parallel fashion.

2.1 DNA as a data structure

The data density of DNA is impressive. As a string of binary data is encoded with 1's and 0's, a strand of DNA is encoded with the four bases (nucleotides), represented by the letters A, T, C, and G. These bases are spaced every 0.35 nanometers along the DNA molecule, giving DNA a remarkable data density of nearly 18 Mbits per inch. In two dimensions, if you assume one base per square nanometer, the data density is over one million Gbits per square inch. Compare this to the data density of a typical high performance hard drive, which is about 7 Gbits per square inch - a factor of over 100,000 smaller.

Another important property of DNA structure is its double stranded nature. This makes DNA a unique data structure for computation and can be exploited in many ways. Error correction is one example. Errors in DNA happen due to many factors. Occasionally, DNA enzymes simply make mistakes, for example inserting a T for a G. DNA can also be damaged by thermal energy and UV energy from the sun. If the error occurs in one of the strands of double stranded DNA, repair enzymes can restore the proper DNA sequence by using the complement strand as a reference. In this sense, double stranded DNA is similar to a "RAID 1 array", where data is mirrored on two drives, allowing data to be recovered from the second drive if errors occur on the first. In biological systems, this facility for error correction means that the error rate can be quite low. For example, in DNA replication, there is one error for every 10⁹ copied bases, i.e. an error rate of 10⁻⁹, which is obviously extremely low.

2.2 DNA operations in parallel

In the cell, DNA is modified biochemically by a variety of enzymes, which are tiny protein machines that read and process DNA according to nature's design. There is a wide variety and number of these "operational" proteins, which manipulate DNA on the molecular level. For example, there are enzymes that cut DNA and enzymes that paste it back together. Other enzymes function as copiers and others as repair units. Molecular biology, Biochemistry, and Biotechnology have developed techniques that allow us to perform many of these cellular functions in the test tube. Just like a CPU has a basic suite of operations like addition, bit-shifting, logical operators, etc. that allows it to perform many complex calculations, DNA has cutting, copying, pasting, repairing, and many others. Also, as mentioned above, in the test tube enzymes do not work sequentially on one DNA at a time. Rather, many copies of the enzyme can work on many DNA molecules simultaneously.

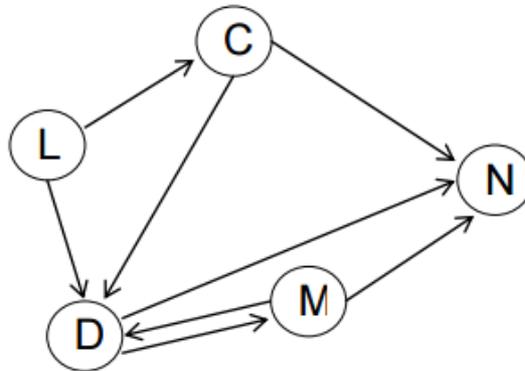


Fig. 3. TSP (Travelling salesman problem)

3 Adleman's Experiment

The method Adleman used to solve directed Hamiltonian Path problem in Figure 3, using the DNA technology, is basically the brute force approach. He first generated all the possible paths and then selected the correct path. Specifically, the method has the following steps (Ryu, W 2007) [3]:

1. Generate all possible routes.
2. Select paths that start with the proper city and end with the final city.
3. Select paths with the correct number of cities.
4. Select paths that contain each city only once.

All of the above steps were accomplished with standard molecular biology techniques.

3.1 Generate all possible routes

Encode city names in short DNA sequences, and then encode the paths by connecting the city sequences for which routes exist. DNA can simply be treated as a string of data. For example, for the case when number of cities $N=5$, each city can be represented by a string of six bases as follows:

L-GCTACG C-CTAGTA D-TCGTAC M-CTACGG N-ATGCCG

The entire path can be encoded by simply stringing together these DNA sequences that represent specific cities. For example, the route L - C - D - M - N would simply be GCTACGCTAGTATCGTACCTACGGATGCCG, or equivalently it could be represented in double stranded form with its complement sequence. Synthesizing short single stranded DNA is now a routine process, so encoding the city names is straightforward. The molecules can be made by a machine called a DNA synthesizer.

Paths can then be produced from the city encodings by linking them together in proper order. To accomplish this we can take advantage of the fact that DNA hybridizes with its complimentary sequence. For example, you can encode the routes between cities by encoding the compliments of the second half (last three letters) of the departure city and the first half (first three letters) of the arrival city. For example the route between M (CTACGG) and N (ATGCCG) can be made by taking the second half of the coding for M (CGG) and the first half of the coding for N (ATG). This gives CGGATG. By taking the complement of this you get, GCCTAC, which not only uniquely represents the route from M to N, but will connect the DNA representing M and N by hybridizing itself to the second half of the code representing M (...CGG) and the first half of the code representing N (ATG...). An example is shown below in figure 4.

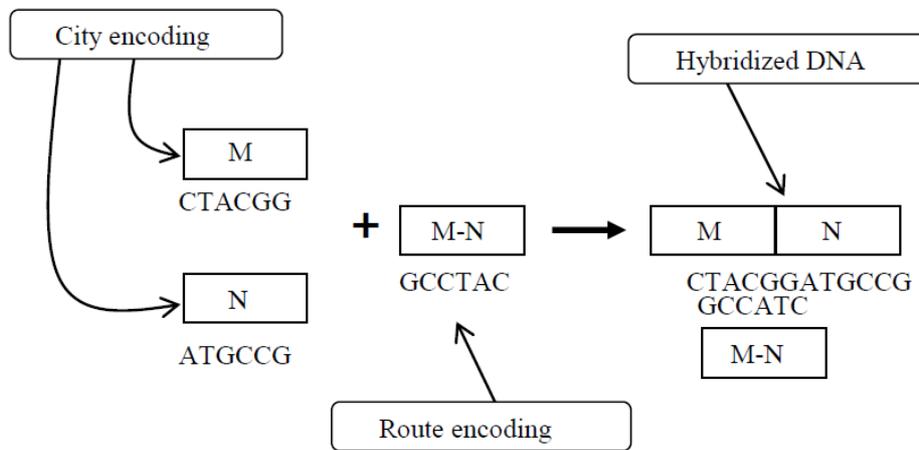


Fig. 4. An example to show the generation of all possible routes

Random paths can be made by mixing city encodings with the route encodings. Finally, the DNA strands can be connected together by an enzyme called ligase. What we are left with are strands of DNA representing paths with a random number of cities and random set of routes. For example as shown in figure 5:

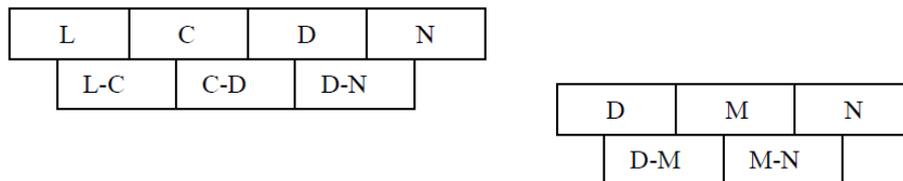


Fig. 5. Examples for route generation

Using an excess of DNA encodings, we can be confident that we have all possible combinations routes between cities including the correct one.

3.2 Select paths that start and end with the correct cities

Selectively copy and amplify only the section of the DNA that starts with L and ends with N by using the Polymerase Chain Reaction. To accomplish this we can use technique called Polymerase Chain Reaction (PCR), which allows us to produce many copies of a specific sequence of DNA. PCR is an iterative process that cycles through a series of copying events using an enzyme called polymerase. Polymerase will copy a section of single stranded DNA starting at the position of a primer. A primer is a short piece of DNA complimentary to one end of a section of the DNA that we are interested in. By selecting primers at the start and end of the section of DNA we want to amplify, the polymerase using this as a template for replication, amplifies the DNA between these primers, doubling the amount of DNA containing this sequence. After many paths of PCR, the DNA we are working on is amplified exponentially. So to amplify the paths that start and stop with our cities of interest, L and N, we use primers that are complimentary to L and N. We then end up with a test tube full of double stranded DNA of various lengths (i.e. with various number of cities), encoding only paths that start with L and end with N.

3.3 Select paths that contain the correct number f cities

Sort the DNA by length and select the DNA whose length corresponds to 5 cities. To accomplish this Adelman used a technique called Gel Electrophoresis(GE), which is a common procedure used to resolve the size of DNA. The basic principle behind Gel Electrophoresis is to force DNA through a gel matrix by using an electric field. DNA is a negatively charged molecule under most conditions, so if placed in an electric field it will be attracted to the positive potential. Since the charge density of DNA is constant (charge per length) long pieces of DNA move as fast as short pieces when suspended in a fluid. The gel is made up of a polymer that forms a meshwork of linked strands. As the DNA is forced to thread its way through the tiny spaces between these strands, it slows down at different rates depending on its size i.e. length. This means that after running a gel we end up with a series of DNA bands, with each band corresponding to a certain length. The band of interest can then simply cut out to isolate DNA of a specific length. In this case the DNA that was 30 base pairs long (5 cities with each city encoded with 6 base pairs of DNA) would be isolated. In the end the test tube will contain all paths that each starts at L and ends at N, and which have a total of exactly 5 cities encoded

3.4 Select paths that have a complete set of cities

Successively filter the DNA molecules by city, one city at a time. Since the DNA we start with contains five cities, we will be left with strands that encode each city once.

DNA containing a specific sequence can be purified from a sample of mixed DNA by a technique called affinity purification. This is accomplished by attaching the compliment of the sequence in question to a substance like a magnetic bead. The

beads are then mixed with the DNA. DNA, which contains the sequence we are after then hybridizes with the complement sequence on the beads. These beads can then be retrieved and the DNA isolated.

So we now affinity purify five times, using a different city complement for each run. For example, for the first run we use L'-beads (the compliment of L) to filter out DNA sequences which contain the encoding for L. (which should be all the DNA because of step 3). The next run we use D'-beads, and then C'-beads, M'-beads, and finally N'-beads. The order isn't important. If a path is missing a city, then it will not be filtered out during one of the runs and will be removed from the candidate pool. What we are left with are the paths that start in L, visit each city once, and end in N. This is exactly what we are looking for. If the answer exists it would be retrieved at this step.

4 Conclusion to Adleman's Experiment

Adleman's experiment has not really solved a problem, but it has shown that DNA and the procedures mentioned above can be used to solve a mathematical problem. However, there are two major shortcomings preventing a large scaling up of his computation. The complexity of the traveling salesman problem simply doesn't disappear when applying a different method of solution - it still increases exponentially. For his method, what scales exponentially is not the computing time, but rather the amount of DNA. Unfortunately this places some hard restrictions on the number of cities that can be solved. Researchers have pointed out that if using Adelman's method to solve a 200 city HP problem the amount of DNA required would be utterly impossible (Ryu, W 2007) [3], (Paun, G 1998) [4]. Another factor that places limits on his method is the error rate for each operation. Since these operations are not deterministic but stochastic in nature, each step contains statistical errors, limiting the number of iterations we can do successively before the probability of producing an error becomes greater than producing the correct result.

The above demonstration of DNA computing has used a simple unsophisticated brute force algorithm. In the future, it is expected that the formalism of DNA computing will become refined. Assuming more efficient algorithms were developed, perhaps one day DNA will overtake conventional computation and set a new record in TSP problem.

5 DNA Advantages and Limitations

DNA has many advantages that make it the future of data computing. It has the ability to perform million operations simultaneously as adleman's experiment shows, the experiment was done at 1,014 operations per second, a rate of 100 Teraflops while the fastest supercomputer in the world runs at just 35.8 Teraflops (Srivastava 2003) which makes the performance rate of DNA computing is so high. DNA computing also allows the most efficient use of memory and space by holding massive amounts of data in very small spaces. For example, 10 trillion DNA molecules can fit in one

cubic centimeter that would result in a computer that holds 10 terabytes of data. Moreover, it provides the ability of parallel processing.

On the other hand, there are some limitations and issues while using DNA computing. One main issue is its accuracy as there is a possibility for errors like mismatching pairs and it is more dependent on the used enzymes' accuracy (Parker 2003). As (Adam 2008) mentioned, DNA computing works with massive data which means the probability of error increases exponentially. In addition to that, it needs intensive resources of human and mechanical intervention between steps because of the parallel operations that consumes a lot of time (hours or days). Also, one of the fundamental issues that are still open according to (Babaei, M. 2013) [15] is that the DNA cryptography does not have a mature mathematical background like other cryptographic systems. Moreover, it has an implementation problem. As mentioned by (Ignatova et al. 2008), that is because of the cost of the labs and biological experiments that have to be done to generate a reliable DNA cryptography system.

6 DNA and Cryptography Systems

DNA computing nowadays works as one of the most important cryptosystems because of its exclusive parallelism and massive data density characteristics. Most of the cryptosystems are based on mathematical equations. The harder these mathematical equations, the harder the attacker can have access to the system. There are many cryptographic systems are used nowadays. One of the most known cryptographic systems is the DES which can be used to produce 64-bit cipher text from 64-bit plain text by using 56-bit secret key. However, using the parallel processing of DNA computing is more advanced than what DES is doing. As (Cui et al. 2009) [16], using a secret key and efficient DNA method was able to break the DES key within a day. The same goes for the RSA cryptographic method which has a reliability problem as it is based on produced factoring large numbers and another problem on breaking it as (Xiao et al. 2012) [16] mentioned.

OTP (One Time Pad) (Hirabayashi et al. 2009; Tantau 2011) is one of the algorithms that are still secure and reliable even while comparing to DNA computing. OTP is a private key encryption algorithm which is completely secure theoretically but practically, it has some challenges in the generation and distribution of keys. The idea of OTP is that the encryption of each bit or character in the plain text is done by a modular addition which means that each bit or character would get a bit or a character from a random key generator. This randomness makes it impossible to break the cipher text in theory. But in practical, to have this kind of security and reliability, each secret key shouldn't be used more than one time in the algorithm which is considered one of the main disadvantages of using OTP. One other disadvantage is that each Pad sequence should be truly random and unpredictable and the length of this Pad sequence should be at least equal to the number of bits in the message.

(Babaei, M. 2013) [15] have developed DNA encryption algorithm which is also combined with the chaos theory. The developed algorithm works for both the text and image encryption. This algorithm combined between the advantages of using DNA of

storing a very large amount of data and the capability of performing enormous parallel reactions, while using the chaos theory. The developed algorithm added some advances to improve the OTP (One Time Pad) algorithm which is considered as a reliable data encryption algorithm while it has some drawbacks that have been considered while developing this algorithm. The proposed algorithm includes a logistic chaotic map as an input of OTP algorithm to overcome these drawbacks. Security analysis, tests and results showed the high reliability of the developed algorithm in image encryption. Concerning the text encryption, the developed algorithm has introduced a chaotic selection between either the DNA strands or the OTP DNA strands.

(Aich et al 2015) [14] have developed a two stage DNA sequence based encryption algorithm while the shared key is generated by a DNA sequence. By the advancement of technology, the traditional cryptography systems couldn't achieve the required level of security to face the advanced attackers. So, DNA was one of the solutions for this problem because of its ability to contain larger information set than any other traditional technique. This algorithm provides more secure, scalable and robust than the traditional cryptosystems because of the two stages of encryption used. The cipher text is generated by encrypting the plain text on two stages. The first stage is by using a randomly generated key which works as an OTP (One Time Pad) key. This way, the attacker would face a problem predicting the random numbers used for the key generation and this would make the system more secure. The other stage is by generating the DNA sequence (by using binary DNA coded scheme) symmetric key which would be used in the decryption as well. The two staged encrypted text and the cipher text are sent over two separate secure channels. The cipher text is also transferred as a DNA sequence to generate the final cipher text. In the developed algorithm, the DNA is used as a carrier data which makes the system more secure as it makes it too difficult for the attacker to guess and have access to the symmetric key while facing the biological difficulty of the DNA sequence. Despite of being more robust, secure and scalable, one of the main challenges of this algorithm is the computational complexity it may produce.

As (Goswami, P. S. 2016) [18] mentioned in his paper, DNA cryptography has been developed, studied and explored more in the last years. As the paper mentioned, most of the studies and research done on DNA cryptography is centralized on DNA sequences used for binary data encoding. Despite the complexity of the DNA cryptography and being in the first stages of development, it is expected that using DNA in cryptography would be a great advance for information security. Using this cryptography system makes use of the advantages of the classical cryptography systems while reducing their risks and restrictions. The randomness of the generated DNA sequences adds many advances and ease to the cryptography. For example, there is no need to send long keys. Instead, keys can be a DNA sequence which has a unique identification number.

7 Advances in DNA Computing

Since Adelman's original work [1] several researchers have come up with number of papers on DNA computation. (Boneh et al 1995) [13] using the same principles as

Adelman's show that it can be applied to breaking the Data Encryption Standard (DES). (R. J. Lipton 1995) [9] showed how to use DNA to solve more general problems. Namely to find satisfying assignments answer for arbitrary (directed) contact networks, which includes the important case of arbitrary formulas. A "knight problem" with the condition that the knights must not attack each other on a 3×3 chessboard was interpreted as an SAT problem using DNA (Faulhammer, et al 2000). (Braich et al .2000) [6] attacked a 20-variable satisfiability (SAT) problem by devising a tool that used glass tubes containing DNA molecules under an electric field at an appropriate temperature. The tic-tac-toe game was examined by Stojanovic and Stefanovic using molecular computing (Stojanovic, M.N., et al 2003) [7]. Once more the traveling salesman problem with DNA molecules encoding numerical values, were studied by (Lee et al. 2004) [8]. Others have explored the potential of DNA computing in biotechnology and medicine. Benenson group, (Benenson, Y., et al 2004) [9], showed that DNA computing can be used in medical diagnosis and drug delivery. This was done by synthesizing DNA-computing drugs in vitro with restriction enzymes. DNA computing has also been used in proving theorems. (Guo et al 2005) [10] proved Cook's theorem to be correct with DNA-based computing. Cook's theorem is that, if one algorithm for a non-deterministic-in-polynomial-time (NP) complete or an NP-hard problem can be developed, then other problems will be solved by means of reduction to that problem. Also (Boneh et al, 1996) [12] have shown how DNA based computer to be used to solve satisfiability problem for Boolean circuits and optimization problems directly without solving several decisions problems. These papers and others have generally been on results of various classical computation models using DNA computation. As have been noticed by researchers (Boneh et al, 1996) [12] most of the results are of the form: Given enough strands of DNA and certain biological operations, one can simulate some classic model of computation efficiently.

However, there are a number of issues that are currently under investigation in research. Two of these issues are the number of biological steps such computation would take, and how many DNA strands it uses. (Boneh et al, 1996) [12] have rated the results of DNA computations on these two parameters. This is similar to the way parallel algorithms are rated. For example, Adelman's HP problem has biological steps of $O(n)$ and used $n!$ strands (n is the number of vertices). Also the set of operations used is another factor to be considered (Boneh et al, 1996) [12]. A computation in DNA is just a sequence of test tubes. Each test tube contains many strands of DNA that encode certain computation. Each subsequent test tube is created from earlier ones by some biological operation. For example, Adelman has used mainly 3 operations Extract (extract strands with given substring), Length (separate strands by length), and Amplify (PCR used to make copies of strands). Another issue to be considered, and foremost one, is that of errors. DNA molecules are known to be fragile, they break easily. It is essential to obtain a better understanding of the type of errors which may occur and to come up with further techniques for error recovery.

8 Conclusion

Advances in biotechnology are happening at a rate similar to the advances made in the silicon industry. Considering all the attention that DNA has garnered, it is not hard to imagine that one day we might have the tools and talent to produce a small integrated desktop machine that uses DNA as a computing substrate along with set of designer enzymes. Perhaps it will not be used to do things that traditional computers are good at, but it might be used in the study of fields such as encryption, genetic algorithms, language systems, and more.

9 References

- [1] Adleman, L. (1994) "Molecular computation of solutions to combinatorial problems". *Science* 266, 1021–1024. <https://doi.org/10.1126/science.7973651>
- [2] Watson J, Crick F (1953) "Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid" *Nature* 171 (4356): 737–8. <https://doi.org/10.1038/171737a0>
- [3] Ryu, W (2007) "DNA Computing" *ars technical*, <http://arstechnica.com/reviews>
- [4] Paun, G (1998) "Computing with Bio-Molecules" Springer-Verlag
- [5] Faulhammer, D., Cukras, A.R., Lipton, R.J., Landweber, L.F. (2000) "Molecular computation: RNA solutions to chess problems". *PNAS* 97, 1385–1389. <https://doi.org/10.1073/pnas.97.4.1385>
- [6] Braich, R.S., Chelyapov, N., Johnson, C., Rothmund, P.W.K., Adleman, L. (2002). "Solution of a 20-variable 3-SAT problem on a DNA computer". *Science* 296, 499–502. <https://doi.org/10.1126/science.1069528>
- [7] Stojanovic, M.N., Stefanovic, D. (2003). "A deoxyribozyme-based molecular automaton". *Nat. Biotechnol.* 21, 1069–1074. <https://doi.org/10.1038/nbt862>
- [8] Lee, J.Y., Shin, S.-Y., Park, T.H., Zhang, B.-T. (2004). "Solving traveling salesman problems with DNA molecules encoding numerical values". *BioSystems* 78, 39–47. <https://doi.org/10.1016/j.biosystems.2004.06.005>
- [9] Benenson, Y., Gil, B., Ben-Dor, U., Adar, R., Shapiro, E. (2004). "An autonomous molecular computer for logical control of gene expression." *Nature* 429, 423–429. <https://doi.org/10.1038/nature02551>
- [10] Guo, M., Chang, W.-L., Ho, M., Lu, J., Cao, J. (2005). "Is optimal solution of every NP-complete or NP-hard problem determined from its characteristic for DNA-based computing". *BioSystems* 80, 71–82. <https://doi.org/10.1016/j.biosystems.2004.10.003>
- [11] R. J. Lipton (1995) "Using DNA to solve NP-Complete problems". *Science*, 268:542-545 <https://doi.org/10.1126/science.7725098>
- [12] D. Boneh, C. Dunworth, R. J. Lipton, J. Sgall (1996) "On the Computational power of DNA" *J. Discrete Applied Maths.*, Special Issue on Computational Molecular Biology, 71, 79-94 [https://doi.org/10.1016/s0166-218x\(96\)00058-3](https://doi.org/10.1016/s0166-218x(96)00058-3)
- [13] D. Boneh, C. Dunworth, R. J. Lipton (1995) "Breaking DES Using a Molecular Computer" Technical Report CS-TR 489-95, Princeton University.
- [14] Aich, Asish, et al. "A Symmetric Key Cryptosystem Using DNA Sequence with OTP Key." *Information Systems Design and Intelligent Applications*. Springer India, 2015. 207-215. https://doi.org/10.1007/978-81-322-2247-7_22

- [15] Babaei, M. (2013). A novel text and image encryption method based on chaos theory and DNA computing. *Natural computing*, 12(1), 101-107. <https://doi.org/10.1007/s11047-012-9334-9>
- [16] Cui G, Li C, Li H, Li X (2009) DNA computing and its application to information security field. In: International conference on natural computation, pp 148–152 <https://doi.org/10.1109/icnc.2009.27>
- [17] Xiao G, Lu M, Qin L, Lai X (2012) New field of cryptography: DNA cryptography. *Chinese Sci Bull* 51(12):1413–1420 <https://doi.org/10.1007/s11434-006-2012-5>
- [18] Goswami, P. S. (2016). Cryptographic Scheme using the Biological Properties of DNA-RNA-A Review. *American Journal Of Advanced Computing*, 3(2).

10 Authors

Samir ABOU EL-SEOUD is with Faculty of Informatics and Computer Science, British University in Egypt, Egypt, (Samir.abouelseoud@bue.edu.eg).

Reham MOHAMED (corresponding author) is with Faculty of Informatics and Computer Science, British University in Egypt, Cairo, Egypt (reham.mohamed@bue.edu.eg).

Samy GHONEIMY is with Faculty of Informatics and Computer Science, British University in Egypt, Cairo, Egypt (samy.ghoneimy@bue.edu.eg).

This article is a revised version of a paper presented at the BUE International Conference on Sustainable Vital Technologies in Engineering and Informatics, held Nov 07, 2016 - Nov 09, 2016, in Cairo, Egypt. Article submitted 22 December 2016. Published as resubmitted by the authors 10 March 2017.