DNA Based Computers: Molecular Programming and Perspectives

Saurabh Sindhu
Department of Computer Science, CRM Jat College, Hisar, Haryana, India

Divya Sindhu
Department of Computer Science, CRM Jat College, Hisar, Haryana, India

Abstract - Cells of living organisms are 'natural nanocomputers' which read and rewrite DNA all the time and perform all the processing for the organisms' activities. Computer scientists along with molecular biologist and biochemists are exploring the potential for computation of biological molecules such as deoxyribonucleic acid (DNA) and ribonucleic acid (RNA), which are information carrying molecules in cell. DNA molecules have the potential to perform calculations many times faster than the most powerful supercomputers. DNA based computation exploits the properties of the DNA as a quaternary logic with the advantages of better storage, better accuracy and shorter time as compared to binary logic used in conventional silicon-based machines. Adleman first time announced the arrival of computers based on biochemical operations and showed that a large class of difficult and computationally hard problems could be solved by mixing solutions in test tubes in a molecular biology laboratory. DNA based computation is an innovative approach and in near future, DNA might be integrated into a computer chip to construct 'Biochip' for making the computer even faster. In this article, various features of DNA computer and operation of its processing unit are discussed.

Keywords: DNA computing, Enzymes, Nucleotides, Logic gates, Parallel computing.

I. INTRODUCTION

The electronic computers are used to store, manipulate and communicate the information and to perform complex calculations. It is capable of receiving information and processes it in accordance with variable procedural instructions, and displays this information in visual, graphic form. Microprocessors used in electronic computers are continuously being improved/updated for their speed and miniaturization. Today's computers are millions of times more powerful than their ancestors in the 40's and 50's. Microprocessors made of silicon will eventually reach their physical limits beyond which our current technology could not venture. To build next generation microprocessors, new material will be required to produce faster computing speeds. An elegant alternative to the classical silicon-based microprocessors is DNA computing [1]. DNA can store vast amounts of information encoded as sequences of the nucleotides i.e, adenine (A), cytosine (C), guanine (G) and thymine (T) and this capacity can be used in computing. The electronic computers use only two digits that are 0 and 1, known as binary digits, whereas a DNA strand contains four-letter alphabets A, C, G and T which can hold large information than earlier type of computers. It is sure that the union of two of science's most fertile fields, molecular biology and computer science is to produce some remarkable offsprings in the form of DNA computing [2, 3].

DNA computing is a form of computing which uses DNA, biochemistry and molecular biology, instead of the traditional silicon-based computer technologies. This field was initially developed by Leonard Adleman of the University of Southern California, in 1994 [4]. Adleman invented a method for solving a small instance of a Directed Hamiltonian Path (DHP) Problem by an *in vitro* DNA-recombination assay which he performed experimentally using hybridization, several agarose-gel separations and PCR by handling DNA sequences in a test tube. Adleman demonstrated use of DNA as a form of computation which solved the seven-point Hamiltonian path problem (the traveling salesman problem; TSP), whose solution required finding a path from start to end going through all the points (cities) only once. As the number of cities increases, the problem becomes more difficult until its solution is beyond analytical analysis altogether, at which point it requires brute force search methods. TSPs with a large number of cities quickly become computationally expensive, making them impractical to solve on even the latest super-computer. Adleman's demonstration only involved seven cities, making it in some sense a trivial problem that can easily be solved by inspection. Since the initial Adleman experiments, advances have been made and various Turing machines have been constructed [5, 6].

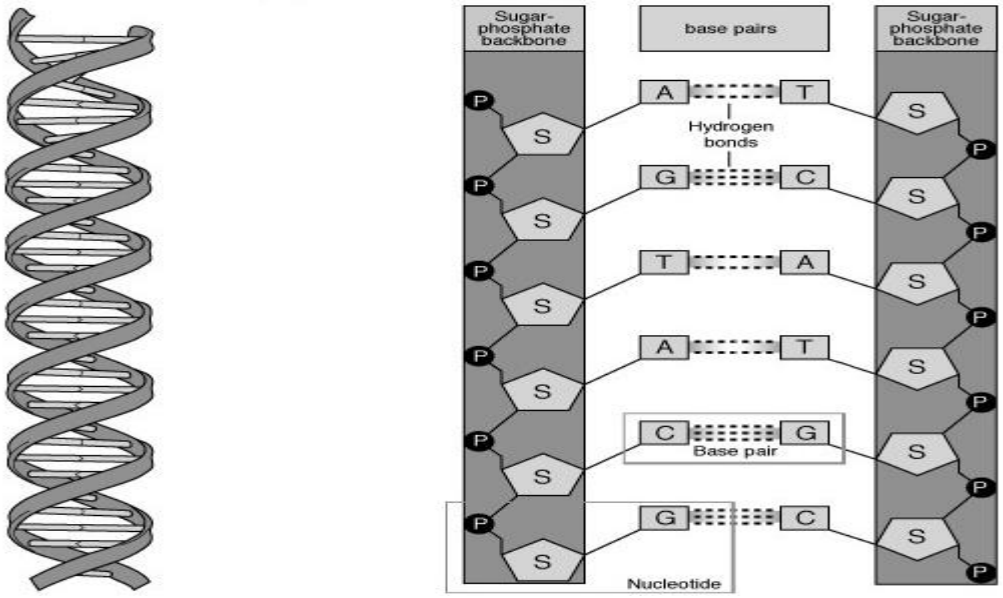
In conventional computers, a CPU has a basic suite of operations like addition, bit-shifting, logical operators (AND, OR, NOT NOR), etc. that allow it to perform even the most complex calculations. DNA has cutting, copying, pasting, repairing and many other functions. In the test tube, enzymes do not function sequentially; working on one DNA at a time, rather, many copies of the enzyme can work on many DNA molecules simultaneously. Therefore, DNA computing is fundamentally similar to parallel computing in that it takes advantage of the many different molecules of DNA to try many different possibilities at once in a massively parallel fashion [7]. 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 [8]. 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 [9]. In this article, we represent an overview about DNA molecules, which make the way for this sort of innovative computing model.

II. STRUCTURE OF DNA

DNA (deoxyribose nucleic acid) was recognized as the most important biological molecule of cellular organisms, wherein it serves to store genetic information. In living organisms, the genetic information is transferred from the parents to the progeny organisms by a faithful replication of its parental DNA molecules. DNA is located on the chromosomes. Its proper replication and transmission to next progeny is crucial for maintaining the genotype characteristics [10]. The ability to store billions of data as nucleotide base sequences is an important feature of the DNA.

DNA is a polymer of nucleotides consisting of adenine, guanine, cytosine and thymine bases, commonly abbreviated to A, G, C and T, respectively. The DNA double helix is stabilized primarily by two forces: hydrogen bonds between nucleotides and base-stacking interactions among nucleotide bases (Figure 1). Both strands of the double-stranded structure store the same biological information. Each purine (guanine and adenine) base is the Watson-Crick (WC) complement of a unique pyrimidine (cytosine and thymine) base (and vice versa) - adenine and thymine form a complementary pair, as do guanine and cytosine. The chemical ties between the two WC pairs are different; C and G pair through three hydrogen bonds, while A and T pair through two hydrogen bonds. The different nucleotides i.e., ATP, GTP, CTP and TTP are polymerized by the enzyme DNA polymerase using a template of the parent strand. The arrangement of nucleotides varies from one organism to the other and different organisms have different arrangement of nucleotides. The discrete segments of DNA coding for particular protein/polypeptide are called genes. Each person's genome has about three billion bases, having the capability to encode about 100,000 genes and these coding regions make up only about 10% of human's genome.

The biological information encoded in genes is made available by gene expression [8, 11]. The information is transferred from DNA to mRNA by the process of transcription and this information is further translated into the protein by making use of the ribosomes. The hereditary information present in the nucleotide sequence is maintained intact by complex metabolism involving both replication and repair functions. Occasionally, mistakes may occur and it may result in alteration of a particular nucleotide in DNA. The change of even single base pair in particular gene of the DNA structure could lead to mutation and it may result in morphological and functional changes. These changes in DNA composition could result in evolution of new species.



A **B**

Figure 1. Double helical structure of DNA. A: Two separate strands of DNA bond together and bonding occurs by the pairwise attraction of bases; Base Adenine (A) bonds with Thymine (T) (with two hydrogen bonds) and Guanine (G) bonds with Cytosine (C) (with three hydrogen bonds). B: The pairs (A, T) and (G, C) are known as Watson-Crick complementary base pairs. Each strand has, according to chemical convention, a 5' and a 3' end. Thus a hypothetical DNA molecule having the following base sequences will have complementary bases as shown below:

GCATGAGTACGTACAAGTGTCCGAATGGCCAATG
CGTACTCATG CATGTTCACAGGCTTACCGGTTAC

The technique by which precise order of nucleotides in a piece of DNA (on full chromosomes or entire genomes) can be determined is termed as DNA sequencing [12]. Recently, automated DNA sequencing machines are capable of identifying 10,000 nucleotide base pairs per day and have become commercially available. The detection and analysis of sequencing reactions in second generation sequencing techniques 454 FLX and Solexa as well as in 3rd generation advanced sequencing techniques including PacBio SMRT and nanopore sequencing [13], is carried out by instruments controlled by computers. Different bioinformatics tools are used in these sequencing techniques like alignment tools, variation detection tools and assembly tools. The advent of rapid DNA sequencing methods has greatly accelerated biological and medical research and discovery. To encode information using DNA, one can choose an encoding scheme mapping the original alphabet onto strings over (A, C, G, T), and proceed to synthesize the obtained information-encoding strings as DNA single strands. Knowledge of DNA sequences has become indispensable for basic biological research and in numerous applied fields such as diagnostic, biotechnology, forensic biology, virology and biological systematics.

III. MOLECULAR COMPUTING OF DNA

Biological computers used to produce input, output and software are all composed of DNA [2, 14], while DNA-manipulating enzymes are used as hardware (Figure 2). DNA-based *in vitro* biocomputer systems have been mainly implemented in test tubes where well-designed species have been assembled and their emergent computational behaviour was observed. Like the most computers, the RNA device operates on a simple system of Boolean logic wherein it can be programmed to respond to the commands AND, OR, NAND and NOR. By combining the RNA components in certain ways, it showed different types of logic gates circuit elements common to any computer [15]. For example, an AND gate produces an output only when its inputs detect the presence of both drugs, while a NOR gate produces an output only when neither drug is detected.

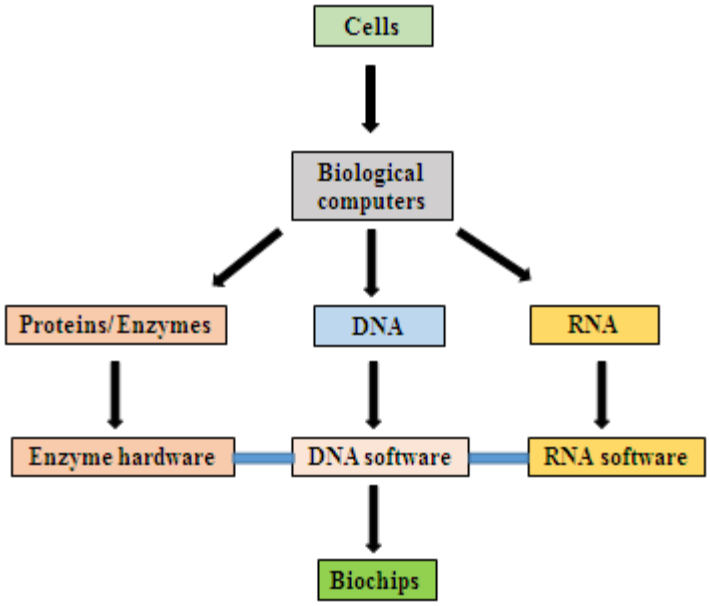


Figure 2. Biological cells work as nanocomputers in which DNA acts as software and the various enzymes involved in DNA manipulation act as hardware.

While DNA-based networks have relied heavily on the primary DNA sequence as information carrier [16], *in vivo* systems adapted existing mechanisms for biological regulation, in particular transcriptional [17] and post transcriptional regulatory links, and generally adhered to logic circuits as the guiding model of computation. Most biological regulation interactions can be classified as either activating or inhibitory [18].

DNA based devices have been addressed for most proposed models of DNA computation [19]. The existing models of DNA computation are based on various properties or combinations of the biological operations, which can be done on DNA sequences in a test tube to program the DNA computer. The bio-operations listed below are used to write programmes which receive a tube containing DNA strands as input and return as output either 'yes' or 'no' or a set of tubes [20, 21].

(i) Synthesis/Replication: synthesis of a desired DNA strand using the DNA polymerase enzymes
(ii) Melting: breaking apart of the two DNA strands having complementary sequences by heating of the solution
(iii) Annealing/merging: bonding together of two DNA strands having complementary sequences by pouring into two test tubes and by cooling of the solution
(iv) Amplification: making copies of the DNA strands by using Polymerase Chain Reaction (PCR)
(v) Separation: separating the DNA strands using agarose gel electrophoresis
(vi) Extraction: extracting those strands containing a given pattern by using affinity purification
(vii) Cutting: cutting of the DNA double strands with restriction enzymes
(viii) Ligation: pasting of the DNA strands with complementary sticky ends using enzyme ligase
(ix) Substitution: insertion, deletion or substitution of the DNA nucleotide bases by mutagenesis
(x) Digestion/destroying: DNA strand could be digested by particular exonucleases
(xi) Detection: confirming the presence/absence of DNA in a given test tube

A computation will consists of a succession of bio-operations [22], such as cutting and pasting DNA strands, separating DNA sequences by length, extracting DNA sequences containing a given pattern, or making copies of DNA strands. The DNA strands representing the output of the computation can then be read out and decoded. DNA computing is based on the idea that molecular biology processes can be used to perform arithmetic and logic operations on information encoded as DNA strands. Nevertheless, some simple dynamic programming techniques can be used to approximately determine base pairings in a secondary structure of a oligonucleotide DNA sequence. Among these techniques, the Nussinov-Jacobson (NJ) folding algorithm is one of the simplest and most widely used schemes.

The theoretical work on DNA computing consists of designing potential experiments for solving various problems by means of DNA manipulation. Descriptions of such experiments include the Satisfiability Problem, breaking the Data Encryption Standard (DES), expansions of symbolic determinants, matrix multiplication, graph

connectivity and knapsack problem using dynamic programming, road coloring problem, exascale computer algebra problems and simple Horn clause computation. To this aim, various aspects of the implementability of DNA computing have been experimentally investigated and the effects of good encodings on the solutions of Adleman's problem were addressed. Moreover, the experimental gap between the design and assembly of unusual DNA structures was pointed out and joining as well as rotating data with molecules was reported. The concatenation with PCR was studied, evaluation of simple Boolean formulas was started and ligation experiments in computing with DNA were conducted.

IV. ADVANTAGES OF DNA COMPUTING OVER CONVENTIONAL COMPUTING

DNA performs parallel operations while conventional, silicon-based computers typically handle operations sequentially [1, 14]. Conventional computers operate linearly, taking on tasks one at a time. A modern CPU basically repeats the same "fetch and execute (fetches an instruction and the appropriate data from main memory and executes it) cycle" over and over again. This process is repeated many times in a row and very fast. Transistor-based computers typically handle operations in a sequential manner. 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. While in DNA based computing, performance improves due to memory capacity and parallel processing [2, 3]. Parallel computing allows DNA to solve complex mathematical problems in hours, whereas it might take electrical computers months/years to complete them. Every DNA molecule could act as a small processor on nano-scale and the number of such processors per volume would be potentially enormous. In an *in vitro* assay, we could handle easily with about 10^{18} processors working in parallel.

DNA computing is another way to surmount certain limitations of traditional silicon-based computing. Computing with DNA offers the advantage of massive degrees of miniaturization and parallelism over conventional silicon-based machines. For example, a square centimeter of silicon can currently support around a million transistors, whereas current manipulation techniques can handle to the order of 10^{20} strands of DNA. The information density could go up to 1 bit/nm^3. The size problem of traditional computers is removed by making the processors as small as a molecule. This technique makes circuits thousand times smaller than traditional silicon based computers. Cells (living parts) of organisms are ingredients for computation. These provide the basic idea of computing, as these tiny parts are complete machines and perform all the processing for the organisms activities [23]. It can overcome on two major limitations of silicon-based traditional computers: storage capacity and processing speed. In this technique, different enzymes and amino acids of DNA are used to solve particular problem.

The key advantage of DNA computing is that it will make computers smaller than any computer that has come before them, while at the same time holding more data. The computing power of a teardrop-sized DNA computer, using the DNA logic gates, will be more powerful than the world's most powerful supercomputer. More than 10 trillion DNA molecules can fit into an area no larger than 1 cubic centimeter. With this small amount of DNA, a computer would be able to hold 10 terabytes of data and perform 10 trillion calculations at a time. By adding more DNA, more calculations could be performed. DNA computers have the potential to take computing to new levels, picking up where Moore's Law leaves off.

Although the elementary operations (electrophoretic separation, ligation, PCR-amplifications) would be slow compared to electronic computers, their parallelism would strongly prevail, so that in certain models the number of operations per second could be of order 10^{18} operations per second, which is at least 100,000 times faster than the fastest supercomputers existing today. The energy efficiency of DNA computing is quite high and it can perform 10^{19} operations per Joule. This is about a billion times more energy efficient than today's electronic devices.

Thus, there are several advantages of using DNA instead of silicon: (i) DNA computers are many times smaller than today's computers; (ii) The large supply of DNA makes it a cheap resource; (iii) As long as there are cellular organisms, there will always be a supply of DNA; (iv) Unlike the toxic materials used to make traditional microprocessors, DNA biochips can be made cleanly; (v) It is based on parallel data processing; (vi) DNA computer has big data storage capacity; (vii) It has low power consumption.

Perhaps the greatest advantage of DNA over electronic circuits is that it can interact with biochemical environment. Computing with molecules involves recognizing the presence or absence of certain molecules and so a natural application of DNA computing is to bring such programmability into the realm of environmental biosensing, or delivering medicines and therapies inside living organisms.

V. DNA LOGIC GATES

Logic gates are a vital part of how computer carries out functions that you command it to do. These gates convert binary code moving through the computer into a series of signals that the computer uses to perform operations. Currently, logic gates interpret input signals from silicon transistors and convert these signals into an output signal

that allows the computer to perform complex functions. In DNA computers, these DNA logic gates rely on DNA code, instead of using electrical signals to perform logical operations. These gates are actually tiny DNA processing centers that detect specific fragments of the genetic blueprint as input and then splice together the fragments to form a single output. For example, a genetic gate called the 'And gate' links two DNA inputs by chemically binding them so they are locked in an end-to-end structure. Ogihara and Ray [24] suggested that these logic gates might be combined with DNA microchips to create a breakthrough in DNA computation.

- **NOT Gate:** The NOT gate is always known as an inverter. It is the simplest DNA-based logic gate in which one input is supplied to the gate and the output is the corresponding complementary sequence. As the output should evaluate "true" only in the presence of a "false" input, the base mixture provided to the gate contains the representative "true" sequence.
- **XOR Gate:** The XOR gate is designed to give "true" if only one of the input sequences evaluates "true." With binary inputs, while XOR is defined as evaluating "true" if input values are opposite. In DNA-based logic gates, the XOR gate is the simplest design as there is no base chain needed to be provided to the gate.
- **OR Gate:** The OR gate gives "true" if at least one of the gate inputs is "true". Introducing the "false" chain as DNA-based enforcement of the XOR Gate. (a) One true and one false input results in a double-strand chain representing reality valuation. On the contrary, (b) two true inputs and (c) two false inputs do not create a double-strand chain. The base mixture will need at least one of the inputs be "true" to create a double-strand chain.
- **NAND Gate:** The NAND gate gives "true" when inputs are not both "true," in the other meaning, if there is at least one input is false. The DNA-based NAND logic gate is the same as the OR gate except the base sequence contains the sequence representing "true" rather than "false." Thus, at least one of the inputs must be "false" in order to create a double–strand chain.

The DNAzymes (also known as deoxyribozymes) catalyzes a reaction when interacting with the appropriate input, such as a matching oligonucleotide and are used to build logic gates analogous to digital logic in silicon; however, DNAzymes are limited to 1-, 2-, and 3-input gates with no current implementation for evaluating statements in series. The DNAzyme logic gate changes its structure when it binds to a matching oligonucleotide and the fluorogenic substrate it is bonded to is cleaved free. While other materials can be used, most models use a fluorescence-based substrate because it is very easy to detect, even at the single molecule limit [25]. The amount of fluorescence can then be measured to tell whether or not a reaction took place. The DNAzyme that changes is then "used," and cannot initiate any more reactions. Because of this, these reactions take place in a device such as a continuous stirred-tank reactor, where old product is removed and new molecules added. Two commonly used DNAzymes are named E6 and 8-17.

VI. MODELS AND FORMATS OF DNA COMPUTATION

In order to define a general-purpose DNA-based molecular computer, a lot of theoretical work has been done on generalizing Adleman's approach that could also be implemented by an *in vitro* system. Adleman's model was generalized by Lipton [26) and showed how his model can encompass solutions to other NP-complete problems. The other model is by splicing operation and vigorously followed by many researchers using formal language theory. It is shown that the generative power of finite extended splicing systems is equal to that of Turing Machines.

Markov chains: A Markov chain, named after Andrey Markov, is a random stochastic process that undergoes transitions from one state to another on a state space [27]. It possesses a property that is usually characterized as "memorylessness": the probability distribution of the next state depends only on the current state and not on the sequence of events that preceded it. This specific kind of "memorylessness" is called the Markov property. Markov chains have many applications as statistical models of real-world processes. The term "Markov chain" refers to the sequence of random variables such a process moves through, with the Markov property defining serial dependence only between adjacent periods (as in a "chain"). It can thus be used for describing systems that follow a chain of linked events, where what happens next depends only on the current state of the system.

Various algorithms performing computations over Markov chains have been developed. These determine sequence power of the transition matrix of a Markov chain and their properties of convergence. Some other algorithms help enable to estimate this limit. These also allow the computation of a limit using DNA computation. The states and the transition probabilities have been encoded using strands of DNA for generating paths of the Markov chain [27].

Yang and Yang [28] introduced another model called as Sticker model. In this model, it is essentially easier to create an initial data pool covering answers at first place followed by a series of selection process to destroy the incorrect ones. The surviving DNA sequences are read as the solutions to the problem. But, algorithms are limited to the problem size. As the number of parameters in the studied problem grows, the algorithm becomes impossible

owing to the tremendous initial data pool size. The solution sequences are built in parts to satisfy one clause in a step and eventually solve the whole Boolean formula after a number of steps. The size of data pool grows from one sort of molecule to the number of solution assignments [28]. Unlike previous models, the sticker mode has a memory that can both read and write to, and employs reusable DNA. Also there is a proposal about the tendency of DNA structures to self-assemble as a computational tool.

For effective computation by these models, it is a common feature of all the proposed implementations that the biological operations to be used are assumed to be error-free. An operation central to and frequently employed in most models is the extraction of DNA strands containing a certain sequence (known as removal by DNA hybridization). The most important problem with this method is that extraction is not 100% efficient and may at times inadvertently remove strands that do not contain the specified sequence. Especially for a large problem, the number of extractions required may run into hundreds, or even thousands resulting a high probability of incorrect hybridization.

Thus, a novel error-resistant model of DNA computation was proposed by Gibbons et al. [29] that obviates the need for hybridization extraction within the main body of the computation. Like previous models, this model is particularly effective for algorithmic description. The authors have given DNA algorithms for 3-vertex-colorability problem, Permutations Problem, Hamiltonian Path Problem, the Subgraph isomorphism problem, and the Maximum clique and maximum independent set problem. There are two general formats in which complex combinatorial sets of DNA molecules may be manipulated either in (i) solution-phase format (in solution or (ii) solid-phase format (attached to a surface).

The solid-phase format possesses many important advantages over the solution-phase format.

(i) Facilitated sample handling: With the DNA molecules attached to a support, the experimental manipulations are very simple. They are addition of a solution to the support and removal (washing) to a solution from the support. These steps are readily automated.
(ii) Decreased losses during sample handling
(iii) Reduction of interference between oligonucleotides
(iv) Solid-phase chemistry permits facile purification of the DNA molecules at every step of the process.

Recently, novel technologies and algorithms have been developed on DNA computation which will contribute to greater understanding and application in biological systems. These approaches/ technologies include:

Molecular nanotechnology: DNA sequences and structures have been discovered with new functional properties for preventing the expression of harmful genes. Bioinformaticians design rigid DNA structures that serve as scaffolds for the organization of matter at the molecular scale and can build simple DNA-computing devices, diagnostic machines and DNA motors. The integration of biological and engineering advances offers great potential for therapeutic and diagnostic applications [30].

Autonomous DNA computation: A one-pot autonomous DNA computation machine is proposed that is based on photochemical gate transition. In this machine, photoligation via 5-carboxyvinyldeoxyuridene (CVU) containing oligodeoxynucleotides and photocleavage via carbazole - modified oligodeoxynucleotides, were employed. The binary digit additions are carried out by one time irradiation at 366 nm in the single test tube. The fluorescence readout by the DNA chip was in good agreement with the correct answer of binary digit additions [31].

Clustering approaches: Clustering is regarded as a consortium of concepts and algorithms that are aimed at revealing a structure in highly dimensional data and arriving at a collection of meaningful relationships in data and information granules [32]. DNA computation has also been used for developing clustering techniques. This is very useful while dealing with huge data sets, unknown number of clusters and encountering a heterogeneous character of available data. It was shown as the essential components of the clustering technique through the corresponding mechanisms of DNA computation.

VII. THE FUTURE AND PITFALLS OF DNA COMPUTING

DNA molecules have also been extensively used as a highly versatile construction material to build nanomechanical devices [33] such as DNA machines [34], motors [35, 36], walkers [37, 38] and robots [39]. The significance of research on DNA computing is two-fold: it is the first demonstrable use of DNA molecules for representing information and also the first attempt to deal with an NP-complete problem. But still much more work has to be done to develop error-resistant and scalable laboratory computations. Designing experiments that are likely to be successful in the laboratory and algorithms that proceed through polynomial-sized volumes of DNA is the need of the hour. It is unlikely that DNA computers will be used for tasks like word processing, but they may ultimately find a niche market for solving large-scale intractable combinatorial problems. The goal of automating, miniaturizing and integrating them into a general-purpose desktop DNA computer may take much longer time.

Enzyme based DNA computers are usually of the form of a simple Turing machine; there is analogous hardware, in the form of an enzyme, and software in the form of DNA. Benenson, et al. [40] demonstrated a DNA computer using the FokI enzyme and expanded on their work by going on to show automata that diagnose and react to prostate cancer: under expression of the genes PPAP2B and GSTP1 and an over expression of PIM1 and HPN [41]. Their automata evaluated the expression of each gene, one gene at a time and on positive diagnosis then released a single strand DNA molecule (ssDNA) that is an antisense for MDM2. MDM2 is a repressor of tumor suppressor protein 53 [42]. On negative diagnosis, it was decided to release a suppressor of the positive diagnosis drug. A limitation of this implementation is that two separate automata are required, one to administer each drug. The entire process of evaluation until drug release took around an hour to complete. This method also requires transition molecules as well as the FokI enzyme. The requirement for the FokI enzyme limits application *in vivo*, at least for use in "cells of higher organisms" [43]. The 'software' molecules in the form of DNA can be reused in this case.

DNAzymes are popular because they allow cleaving of a substrate in any arbitrary location [44]. The E6 DNAzymes were used to build the MAYA I [45] and MAYA II [46] machines, respectively. Stojanovic et al. [47] has also demonstrated logic gates using the 8-17 DNAzyme. While these DNAzymes have been demonstrated to be useful for constructing logic gates, they are limited by the need for a metal cofactor to function, such as Zn^{2+} or Mn^{2+} and thus are not useful *in vivo* [25, 48].

However, there are many technological hurdles to overcome. The idea of using DNA to solve computational problems is certainly intriguing and elegant, and DNA does provide a massive parallelism far beyond what is available on existing silicon-based computers. The fundamental problem is that, the function of 2^n is exponential whether it counts time or molecules. It has been estimated that Adleman's Hamiltonian path problem, if enhanced to 50 or 100 cities, would require tonnes of DNA. The minimum amount of required DNA for Lipton's SAT method needs a few grams of DNA molecules for 70 variables. If this is increased to 100 variables, the minimum DNA requirement will be of millions of kilograms. Thus, it is imperative to bring forth new revolutionary ideas to make this notion of DNA-based computing to work realistically.

VIII. CONCLUSION AND PERSPECTIVES

The apparent ease with which DNA hybridization can be formalized made Adleman's invention
very attractive to researchers in the fields of computer science and discrete mathematics. Numerous architectures for DNA based computing have since been proposed [49-51]. It is expected that DNA will soon exhibit all sorts of eccentric behavior that cannot be conveniently incorporated into formal descriptions. However, DNA may very well have potential for soft computing applications that can tolerate or even benefit from components which do not overly conform to formal rules. DNA computer may be used only to solve a specific problem and it can be used only once, in one problem case.

The advantages of DNA computers, at least for the time being, are outweighed by the difficulties of using them to do anything useful. There is more theory than practical success at the moment, though they have enormous potential for applications in medicine, farming, food and fingerprint recognition, and forensic science [52]. DNA programmes have already been put to medical uses such as diagnosing of tuberculosis. Other DNA programmes for medical applications target white blood cell lymphocyte, which are defined by the presence or absence of certain cell markers and so, can be naturally detected with true/false Boolean logic.

Thus, the field of DNA computation remains alive and promising, even as new challenges emerge. Most important among these are the uncertainty because of the DNA chemistry in the computational results and the exponential increase in number of DNA molecules necessary to solve problems of interesting size. Despite these issues, definite progress has been made both in quantifying errors and in development of new protocols for more efficient and error-tolerant DNA computation.

REFERENCES

[1] Kumar, S.N., "A proper approach on DNA based computer", American Journal of nanomaterials 3(1), 1–14 (2015).

[2] Tagore, S., Bhattacharya, S., Islam, M.A. and Islam, M.L., "DNA computation: Applications and Perspectives", Journal of Proteomics and Bioinformatics 3(7), 234–243 (2010).

[3] Bridget, M., Oludele, A., Olusola, M. and Andy, E., "A survey of DNA computing", Journal of Research in Humanities and Social Science 1, 25–33 (2013).

[4] Adleman, L.M., "Molecular computation of solutions to combinatorial problems", Science 266 (5187), 1021–1024. doi:10.1126/science.7973651 (1994).

[5] Boneh, D., Dunworth, C., Lipton, R.J. and Sgall, J.Í., "On the computational power of DNA", Discrete Applied Mathematics 71, 79–94 (1996).

[6] Kari, L., Gloor, G. and Sheng, Y., "Using DNA to solve the Bounded Post Correspondence Problem", Theoretical Computer Science 231(2), 192–203 (2000).

[7] Lewin, D.I., "DNA computing", Computing in Science and Engineering 4(3), 5–8 (2002).
[8] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walters, P., "Molecular Biology of the Cell", Fourth Edition. New York and London: Garland Science (2002).
[9] Benenson, Y., "Biocomputers: from test tubes to live cells", Molecular Biosystems 5, 675–685 (2009).
[10] Krieger, M., Scott, M.P, Matsudaira, P.T., Lodish, H.F., Darnell, J.E., Lawrence, Z., Kaiser, C. and Berk, A., "Structure of nucleic acids, Section 4.1", Molecular Cell Biology. New York: W.H. Freeman and Co. (2004).
[11] Jacob, F. and Monod, J., "Genetic regulatory mechanisms in the synthesis of proteins", Journal of Molecular Biology 3, 318–356 (1961).
[12] Sanger, F., "Sequences, sequences, and sequences", Annual Review of Biochemistry 57, 1–28 (1988).
[13] Shendure, J. and Hanlee, J., "Next generation DNA sequencing", Nature Biotechnology 26, 1135–1145 (2008).
[14] Abd El-Bary, A.E.M., AbdelRassoul, R. and El-Ebshihy, A.M., "Using DNA computing in classification", International Journal of Computer Applications 133(11), 38–44 (2016).
[15] Uckun, F.M., "Role of defective RNA processing in the pathogenesis of pediatric diseases", Pediatric Therapeutics 1, pp. e101 (2011).
[16] Butt, A.M. and Ahmed, A., "Mutator: tool for the introduction of custom position based mutations in protein and nucleotide sequences", Journal of Proteomics and Bioinformatics 2, 344–348 (2009).
[17] Ling, J., "Translation of human genome", Biochemistry and Analytical Biochemistry 1, pp. 101e (2011).
[18] Yun, Y., Conforti, L., Muganda, P. and Sankar, J., "Nanomedicine-based synthetic biology", Journal of Nanomedicine and Biotherapeutics Discovery 1, 102e (2011).
[19] Faulhammer, D., Lipton, R.J. and Landweber, L.F., "Fidelity of enzymatic ligation for DNA computation", Journal of Computer Biology 7, 839–848 (2000).
[20] Aoi, Y., Yoshinobu, T., Tanizawa, K., Kinoshita, K. and Iwashaki, H., "Ligation errors in DNA computation", Biosystems 52, 181–187 (1999).
[21] Fu, B. and Beigel, R., "Length bounded molecular computing", Biosystems 52, 156–163 (1999).
[22] Kari, L., "DNA computing: arrival of biological mathematics", The Mathematical Intelligencer 19(2), 9–22 (1997).
[23] Landweber, L.F. and Kari, L., "The evolution of cellular computing: nature's solution to a computational problem", Biosystems 52, 3–13 (1999).
[24] Ogihara, M. and Ray, A., "DNA computation on a chip", Nature 403, 143–144 (2000).
[25] Weiss, S., "Fluorescence spectroscopy of single biomolecules", Science 283 (5408), 1676–1683 (1999).
[26] Lipton, R.J., "Using DNA to solve NP-complete problems", Princeton University (1995).
[27] Cardona, M., Colomer, M.A., Conde, J., Miret, J.M., Miró, J. and Zaragoza, A., "Markov chains: computing limit existence and approximations with DNA", Biosystems 81, 261–266 (2005).
[28] Yang, C.N. and Yang, C.B., "A DNA solution of SAT problem by a modified sticker model", Biosystems 81, 1–9 (2005).
[29] Gibbons, A., Amos, M. and Hodgson, D., "DNA computation", Current Opinion in Biotechnology 8, 103–106 (1997).
[30] Condon, A., "Designed DNA molecules: principles and applications of molecular nanotechnology", Natural Review in Genetics 7, 565–575 (2006).
[31] Ogasawara, S., Ami, T. and Fujimoto, K., "Autonomous DNA computation machine based on photochemical gate transition", Journal of American Chemical Society 130, 10050–10051 (2008).
[32] Bakar, R.B., Watada, J. and Pedrycz, W., "DNA approach to solve clustering problem based on a mutual order", Biosystems 91, 1–12 (2008).
[33] Li, T., Lohmann, F. and Famulok, M., "Interlocked DNA nanostructures controlled by a reversible logic circuit", Nature Communications 5, 4940. DOI: 10.1038/ncomms5940 (2014).
[34] Yurke, B., Turberfield, A.J., Mills, Jr. A.P., Simmel, F.C. and Neumann, J.L., "A DNA-fuelled molecular machine made of DNA", Nature 406, 605–608 (2000).
[35] Chen, Y. and Mao, C., "Putting a brake on an autonomous DNA nanomotor", Journal of American Chemical Society 126, 8626–8627 (2004).
[36] Wickham, S.F., Bath, J., Katsuda, Y., Endo, M., Hidaka, K., Sugiyama, H. and Turberfield, A.J., "A DNA-based molecular motor that can navigate a network of tracks", Nature Nanotechnology 7, 169–173 (2012).
[37] Shin, J.S. and Pierce, N.A., "A synthetic DNA walker for molecular transport", Journal of American Chemical Society 126, 10834–10835 (2004).
[38] Omabegho, T., Sha, R. and Seeman, N.C., "A bipedal DNA brownian motor with coordinated legs", Science 324, 67–71 (2009).
[39] Douglas, S.M., Bachelet, I. and Church, G.M., "A logic-gated nanorobot for targeted transport of molecular payloads", Science 335, 831–834 (2012).
[40] Benenson, Y., Paz-Elizur, T., Adar, R., Keinan, E., Livneh, Z. and Shapiro, E., "Programmable and autonomous computing machine made of biomolecules", Nature 414 (6862): 430–434 (2001).
[41] Benenson, Y., Gil, B., Ben-Dor, U., Adar, R., and Shapiro, E., "An autonomous molecular computer for logical control of gene expression" Nature 429 (6990), 423–429 (2004).
[42] Bond, G.L., Hu, W. and Levine, A.J., "MDM2 is a central node in the p53 pathway: 12 years and counting", Current Cancer Drug Targets 5(1), 3–8 (2005).
[43] Kahan, M., Gil, B., Adar, R. and Shapiro, E., "Towards molecular computers that operate in a biological environment", Physical: Nonlinear Phenomena 237(9), 1165–1172 (2008).
[44] Santoro, S.W. and Joyce, G.F., "A general purpose RNA-cleaving DNA enzyme", Proceedings of the National Academy of Sciences 94(9), 4262–4266 (1997).
[45] Stojanovic, M.N. and Stefanovic, D., "A deoxyribozyme-based molecular automaton", Nature Biotechnology 21(9), 1069–1074 (2003).
[46] MacDonald, J., Li, Y., Sutovic, M., Lederman, H., Pendri, K., Lu, W., Andrews, B.L., Stefanovic, D. and Stojanovic, M.N., "Medium scale integration of molecular Logic Gates in an automaton", Nano Letters 6(11), 2598–2603 (2006).
[47] Stojanovic, M.N., Mitchell, T.E. and Stefanovic, D., "Deoxyribozyme-based Logic gates", Journal of the American Chemical Society 124(14): 3555–3561 (2002).
[48] Cruz, R.P.G., Withers, J.B. and Li, Y., "Dinucleotide junction cleavage versatility of 8-17 Deoxyribozyme", Chemistry and Biology 11, 57–67 (2004).
[49] Braich, R.S., Chelyapov, N., Johnson, C., Rothemund, P.W.K. and Adleman, L., "Solution of a 20-variable 3-SAT problem on a DNA computer", Science 296, 492–502 (2002).

[50] Seelig, G., Soloveichik, D., Zhang, D.Y. and Winfree, E., "Enzyme-free nucleic acid logic circuits", Science 314 (5805), 1585–1588 (2006).
[51] Church, G.M., Gao, Y. and Kosuri, S., "Next-generation digital information storage in DNA", Science 337 (6102), 1628 (2012).
[52] Sindhu, D. and Sindhu, S., "Biological computers: Their application in gene data mining and protein engineering", International Journal of Technical Research 4(3), 15–21 (2015).