

DNA Computing: A Review

Russell Deaton

*Electrical Engineering
The University of Memphis
Memphis, TN, USA
rjdeaton@memphis.edu*

Max Garzon

*Computer Science
The University of Memphis
Memphis, TN, USA
garzonm@hermes.msci.memphis.edu*

John Rose

*Electrical Engineering
The University of Memphis
Memphis, TN, USA
jrose@lear.csp.ee.memphis.edu*

D. R. Franceschetti

*Physics
The University of Memphis
Memphis, TN, USA
dfrncsch@memphis.edu*

S. E. Stevens, Jr.

*Microbiology and Molecular Cell Sciences
The University of Memphis
Memphis, TN, USA
estevens@cc.memphis.edu*

Abstract. DNA computing holds out the promise of important and significant connections between computers and living systems, as well as promising massively parallel computations. Before these promises are fulfilled, however, important challenges related to errors and practicality have to be addressed. On the other hand, new directions toward a synthesis of molecular evolution and DNA computing might circumvent the problems that have hindered development, so far.

Keywords: DNA Computing, Molecular Evolution, Combinatorial Optimization, Evolutionary Optimization

1. Introduction

DNA-based computing is at the intersection of several threads of research. The information-bearing capability of DNA molecules is a cornerstone of modern theories of genetics and molecular biology[1]. The information in a DNA molecule is contained in the sequence of nucleotide bases, which hydrogen bond in a complementary fashion to form double-stranded molecules from single-stranded oligonucleotides[2]. Various aspects of life inspired early results in computer science in the 1950's (J. von Neumann's universal constructor and computer[3], S. Ulam's models of growth using cellular automata[4]). A second development occurred in the early 1970's with J. Holland's computational implementation of fundamental biological mechanisms, such as genetic operations (splicing, recombination and mutation) and evolution[5]. Finally, a third stage inaugurated by L. Adleman's 1994 proof of concept[6] that recombinant properties of real DNA can actually use massive parallelism to solve problems appropriately encoded into single DNA strands. Much work[7, 8, 9, 10, 11], ranging from the highly speculative to the realistic kind has been stimulated by this seminal paper.

In this paper, the state of DNA computing that has developed from Adleman's paper is reviewed. The purpose is not to produce a comprehensive bibliography of the field. For this, readers are referred to [12]. Rather, the purpose is to summarize the main trends in the field, to identify the key challenges for researchers, and to speculate on future developments.

2. A Molecular Biology Primer

Macromolecules of nucleic acids are the prime conveyers of genetic information[1]. They are composed of nucleotide building blocks. In DNA, the nucleotides are the purines adenine (A) and guanine (G), and the pyrimidines thymine (T) and cytosine (C). Single-stranded DNA molecules, or oligonucleotides, are formed by connecting the nucleotides together with phosphodiester bonds. The single strands of DNA can form a double-stranded molecule when the nucleotides hydrogen bond to their Watson-Crick complements, $\overline{A} \equiv T$ and $\overline{G} \equiv C$, and *vice versa*. Oligonucleotides bind in an antiparallel way with respect to the chemically distinct ends, 5' and 3', of the DNA molecule. In the DNA helix, the intertwined strands are complementary, and one strand serves as the template for the replication of the other. The hydrogen bonding, or base pairing, of one oligonucleotide to another is called hybridization.

Ideally, oligonucleotide hybridizations occur only between Watson-Crick complements. Depending upon the conditions under which the hybridization is done, however, base pairs that are not Watson-Crick complements, or mismatched base pairs, can occur[13]. In addition, oligonucleotides can hybridize in various alignments that are shifted from the designed one. The effect of the reaction conditions is characterized as the hybridization stringency. In general, as the reaction temperature of the hybridization is increased, the stringency increases. The temperature at which half the population of perfectly matched oligonucleotide hybrids will have dissociated into single strands is called the melting temperature, T_m (°C). The melting temperature is determined from curves of UV absorbance versus temperature, and can be interpreted as the fraction of single strands versus temperature[14]. Under conditions of low stringency, oligonucleotides can hybridize with more mismatched base pairs and over shorter lengths than under conditions of high stringency.

The stoichiometric equation for hybridization of two arbitrary oligonucleotides, x_i and x_j , is



where x_jx_i represents the hybridized oligonucleotides. In a DNA computation, there are many reactions like Eq. 1. The direction of the reaction in Eq. 1 is determined by the sign of the change in the Gibb's free energy (G) of the reaction,

$$\Delta G = \Delta G^\circ + RT \log Q. \quad (2)$$

where ΔG° is the free energy change under standard conditions of concentration and pressure, R is the gas constant, and T is the temperature. The concentration factor, Q , is,

$$Q = \frac{[x_i x_j]}{[x_i][x_j]}, \quad (3)$$

where $[]$ indicates mole fractions, and therefore, Q is dimensionless. The reaction will be driven towards chemical equilibrium, where the rates to the left and right of \rightleftharpoons in Eq. 1 are equal, and $\Delta G = 0$. The condition for chemical equilibrium, $\Delta G = 0$, translates to

$$\Delta G^\circ = RT \log K, \quad (4)$$

where K is the equilibrium constant, which is given by

$$K = \frac{[x_i x_j]_{eq}}{[x_i]_{eq}[x_j]_{eq}}, \quad (5)$$

where $[]_{eq}$ indicates the equilibrium mole fractions. The reactions will proceed to a greater or lesser degree according to the size of their free energy changes, and the relative concentrations of reaction products will be related to the equilibrium constant. The primary energetic factor for hybridization is not the energy of the hydrogen bonding between nucleotide bases, but is the nearest neighbor stacking energies[15]. These base stacking energies must be measured, and are not unique. Nevertheless, from an energetic point of view, they are the parameters of choice to determine the potential for hybridization between oligonucleotides. Recently measured values for these parameters are given in [16].

In molecular biology and the cell, enzymes play an important role as catalysts of reactions, and transformers of information encoded in the DNA molecules. Several enzymes have played an important role in DNA computing, and as tools for the manipulation and transformation of DNA molecules, will continue to do so.

Restriction enzymes arise in natural systems as part of the cell's defense mechanisms, in which they cut up DNA that is not protected by specific chemicals, or methylated. Typically, a restriction enzyme recognizes a specific nucleotide sequence in a double-stranded DNA molecule, attaches itself in the vicinity of that sequence, and cuts the DNA molecule at that point. Another enzyme is ligase, which aids the formation of the phosphodiester linkages in double-stranded DNA. Other useful enzymes are exonucleases and endonucleases which chop DNA into its component nucleotides from the end of strand, or location internal to a strand, respectively.

An important technique in molecular biology is the Polymerase Chain Reaction (PCR). The enzyme polymerase synthesizes a DNA strand in the 5' to 3' direction on a DNA template. PCR is used for amplification of specific base sequences. In the basic technique, double-stranded DNA is separated, and short oligonucleotide primers are hybridized to the target. Of course, this necessitates that part of the desired sequence be known for determination of the primers. Then, new DNA is synthesized from the primers using a polymerase. At this point, the amount of DNA has doubled. Then, additional primers are added to delineate the other end of the target sequence, and the entire process is repeated. Because of its repetitive nature, PCR has been automated, and is capable of generating a huge number of copies of the target sequence.

Gel electrophoresis is a technique used to separate and visualize DNA molecules based on their size. It exploits the fact that the DNA molecules are electrically charged. The gel is a matrix of molecules with holes of varying size. Under an applied electric field, DNA molecules will move a distance that depends on their size, with smaller molecules moving farther. By comparison to standards, the sizes of molecules under analysis are determined. With appropriate cutting of molecules and standards, DNA molecules can be sequenced with this technique.

Target sequences are extracted from a reaction mixture with a magnetic bead separation technique. The complement of the target sequence is attached to a bead, and then, many duplicates are mixed with the solution containing the target. Strands containing the target hybridize with the sequences in the bead, and are extracted with a magnetic field.

For additional detail on biochemistry and molecular biology, the readers are referred to several excellent reference texts[1, 2].

3. Beginnings

The computational capability of living systems has intrigued researchers for years. Primarily, the focus has been on implementing aspects of living systems in computational devices. Examples are cellular automata, genetic algorithms, artificial neural networks, and artificial life. The argument has been that universal computational devices are capable of simulating the behavior of physical, living systems through appropriate programming. Therefore, the direction of innovation has been from biology to computer science. Until Adleman[6], the implementation of computational algorithms in biological systems had been extremely rare, and usually, presented as a theoretical suggestion, rather than an actual implementation.

An early proposal for implementation of a computer in DNA was [17]. Though motivated by the existence of logically irreversible automata, the insight of this work is valuable for its connection of computation and thermodynamics. With the goal of minimizing energy dissipation, Bennett proposed a DNA-based computer that operates near thermal equilibrium. He also realized the transformative power of enzymes, and foresaw their potential role in a DNA-based computation. In a DNA-based computer, the computation is realized by a chain of reactions. At each step, equilibrium is achieved between reactants and products, and therefore, the forward and backward reactions occur with identical rates. Bennett looked upon this equilibrium mechanism as a means of achieving thermodynamic, as well as logical, reversibility. The trick in Bennett's proposed DNA computer, as well as their current realizations, is to insure that the reaction path corresponding to the desired computation is followed. This is an important point concerning DNA-based computers. As opposed to solid state based computers, DNA computers are much more intimately connected to the underlying chemistry. This makes digital-style computations difficult to control and validate, but as Bennett realized, also makes a new style of computation possible.

Another proposal was Head's for implementation of regular languages in DNA, which he termed a splicing system[18]. Specifically, a splicing system is the set of double-stranded DNA molecules that are generated as the result of restriction enzyme activity represented as a language over the four symbol alphabet of nucleic acid pairs. The importance of this work is the realization of the computational power of enzymatic activity coupled with the information capacity of DNA. Splicing systems represent a model for DNA computation that is a part of formal language theory[19], and would provide a more direct path to DNA computers for implementation of traditional algorithms. The main question is whether real biological computers based on these theories would bring anything beyond digital computers to DNA computation, or are practical. A partial answer to the latter question is that simple splicing systems can be implemented in the lab[20].

4. Adleman's Breakthrough

Adleman[6] ignited the interest in DNA computing with an actual laboratory demonstration of the computational power inherent in DNA and molecular biology operations. He implemented an algorithm for the solution of a hard, combinatorial problem, the directed Hamiltonian path problem (HPP), which is NP-complete[21]. In HPP, the goal is to find a path through a directed graph that starts and ends at specified vertices, and visits each

vertex in the graph once, and only once. There is overwhelming evidence that the problem is impossible to solve by conventional digital computers by means other than a brute-force algorithm. The algorithm is reproduced below[6]:

1. Generate random paths through the graph.
2. Keep only those paths that begin with v_{in} and end with v_{out} .
3. If the graph has n vertices, then keep only those paths that enter exactly n vertices.
4. Keep only those paths that enter all the vertices of the graph at least once.
5. If any paths remain, say "Yes"; otherwise, say "No."

Adleman's insight was to find a way to encode the problem into DNA molecules and turn this algorithm into a feasible search using available biotechnology. Step 1 was implemented with DNA hybridization reactions. The vertices and edges of the graph were encoded in oligonucleotides of DNA so that upon hybridization and ligation, molecules were formed that represented paths through the graph. An example encoding of a graph is given below:

Vertex	Oligonucleotide Encoding	Edge	Edge Encoding
0	5' TACTCATATGGGGTTATACG 3'	0 → 1	3' CCAATATGCGAGGCGGACC 5'
1	5' CTCCGCCTGGGCTTAGCTTA 3'	1 → 2	3' CGAATCGAATCTAGGAGACA 5'
2	5' GATCCTCTGTTTCCTCAGCT 3'	2 → 3	3' AAGGAGTCGACCGAGGTGAA 5'
3	5' GGCTCCACTTACTCTCTTGT 3'	3 → 4	3' TGAGAGAACAATACCCGATC 5'
4	5' TATGGGCTAGCGGTCCGGTT 3'	4 → 5	3' GCCAGGCCAACGGGAACATC 5'
5	5' GCCCTTGTAGTCTCGGGTCC 3'	5 → 6	3' AGAGCCCAGGAAGGACATTG 5'
6	5' TTCCTGTA ACTTGCCCTCTAA 3'	6 → 0	3' AACGGAGATTATGAGTATAC 5'

Because of the massive number of oligonucleotides in the reaction (approximately 3×10^{13} copies of each), the hybridization reactions performed a massively parallel generation of all possible paths in the graph. Therefore, it is the huge number of trials and matches that take place in parallel by the DNA hybridization reactions that represent the search power of a DNA computation. After the hybridizations, a molecule representing the Hamiltonian path, if any, would exist in the reaction mixture. Subsequent steps were to extract and identify this molecule.

To implement step 2 of the algorithm, PCR was performed with primers representing the input and output vertices (v_0 and v_6 , respectively). In addition, the oligonucleotides representing the input and output vertices were blunt ended to prevent cyclic repetition of paths. Step 3 was done with gel electrophoresis to extract molecules of the proper length. Step 4 was implemented with successive affinity purification using magnetic beads with sequences corresponding to each vertex in the graph. In the final step, the products of step 4 were amplified, and run on a gel. If any molecules remained, then, they were a representation of the Hamiltonian path. If no molecules were detected on the final gel, then, no Hamiltonian path existed.

Adleman's success produced an enthusiastic response from both the technical community, and popular press. Admittedly, the graph that Adleman solved was almost trivial, but that was not the real significance of what he did, and certainly, not responsible for the excitement generated. The exciting thing is that his result opened a window between biological and computational systems, which offers the immediacy and feasibility absent in prior proposals. The promise was not only of more powerful computer systems based on DNA, but moreover, of new insight into life itself.

5. Promise

Lipton[22] modified and extended Adleman's technique to SATISFIABILITY and other NP-complete problems. The problem-specific technique of Adleman was extended to more general Boolean operations, though theoretically, not in the lab. A DNA computation was classified into five simple operations that could be done on a test tube of DNA:

1. Synthesis of a large numbers of oligonucleotides.
2. Annealing (*i. e.*, hybridization) of oligonucleotides to produce double-stranded DNA molecules.
3. Extraction of molecules containing a given sequence of bases.
4. Detection of any molecules in the tube.
5. Amplification of all the DNA in the tube.

Another paper suggested that a DNA computer based on Adleman's result could be used to implement an associative memory with a capacity larger than that of the human brain[23]. Though not implemented in the lab, the suggested scheme is interesting because it takes advantage of the chemistry of the DNA hybridization reactions. Rather than requiring that perfect Watson-Crick complements hybridize to form the result of the computation, the associative memory capitalizes on the fact that a given oligonucleotide, depending on reaction conditions, can hybridize with any number of other oligonucleotides that are chemically close to it.

After Adleman, many early papers focused on constructions of DNA computers which showed equivalence to traditional algorithmic models of computation, *e. g.* Turing Machines[24, 25, 26], boolean circuits[27], parallel models of computation[28] *etc...* While useful for understanding the limits on the power of DNA computation, to our knowledge, none of these suggestions were ever successfully or completely implemented in the lab.

6. Challenges

There are two main challenges to DNA computing[29]. The first is to cope with the uncertainty that the DNA chemistry produces in the results of the computation. DNA chemistry is not an exact process. In fact, it can even be described as messy. The consequence of this is that DNA computers designed to produce exact, algorithmic results, often, failed to do so. This is evident in the early failures to reproduce Adleman's results, and the focus at the 1997 DIMACS meeting[9] on "making it work in the lab." The second challenge is scaling DNA computers so that they will work on problems of interesting size. As pointed out early on by Hartmanis[30], problems of significant size might require a volume of DNA that would fill the Pacific ocean, or weigh as much as the Earth.

The importance of practical implementation, and the potential for errors in a DNA computation have been realized from the very beginning. Adleman[6] used random 20-mer encodings for the oligonucleotides in his experiment. Since there are 4^{20} different 20-mers, it could be argued that the likelihood of unwanted hybridizations is small. Adleman [6], however, points out that as the size of the problem grows, particular attention must be paid to errors and the formation of "pseudopaths", *i. e.*, DNA strands which appear to be valid solutions, but actually are false positives. Lipton[22], as well, realized the criticality of implementation and errors. While suggestions for algorithms have been plentiful, experimental results are lacking, especially on the important issues of errors and practicality.

Errors can be introduced at any number of steps in a DNA computation. Care must be taken that the starting material (oligonucleotides, enzymes, *etc...*) is of good quality and free of contaminants[31]. Non-Watson-Crick base pairs are possible in hybridized DNA, and can introduce false positives and negatives[32]. The fidelity of PCR is an issue[33], as is the efficiency and reliability of extraction techniques[34, 35]. All of these effects are produced

by the fundamental chemical nature of a DNA computer[31], and the sensitive dependence of the computational results on the chemical conditions. A DNA computer is a chemical computer in a much more immediate sense than a silicon device. Multiple outcomes to hybridization reactions can and do occur. The concentration of different species, and the environment, especially temperature, are critical. Different graphs can produce transport and concentrations problem that are unique to the graph being solved[36]. Enzyme reactions are often incomplete, and depend on time, concentration, and temperature. Molecular biology protocols and estimates of experimental parameters are adequate for molecular biology applications, but may be inadequate for applications that require faithful, reproducible, and precise computational results. Therefore, practical issues should dominate any discussion of DNA computing, and though slow and expensive, progress is being made.

7. Progress

A preliminary analysis of hybridization errors appears in [37], where it is assumed that undesirable hybridizations occur if complementary substrands x of a certain minimum sticking length $|x| := k$ appeared in the encoding. If x repeats within the same strand, the intended hybridization will occur at the wrong place, thereby reducing the efficiency of the reaction (i.e., number of molecules available for extraction). If its complement \bar{x} occurs in a different strand, pseudopaths may be created that may eventually end up, upon extraction, in a false positive path.

Other work has shown that random encodings in general are inadequate to prevent hybridization errors[32]. At first, the propensity of oligonucleotides to hybridize was characterized with the Hamming distance between oligonucleotides, and encodings generated to be some minimum Hamming distance apart under the operation of Watson-Crick complementation are less error-prone than random encodings[38]. Using the Hamming distance and estimates of the melting temperature[13], the reaction temperature could be estimated for better performance, and an upper bound on the number of reliable encodings is the Hamming bound[38].

The Hamming distance, however, is inadequate as it fails to account for all possible mishybridization mechanisms, particularly the likelihood of hybridization between oligonucleotides that are shifted relative to each other, and hybridization of a oligonucleotide with itself (hairpin), a definite possibility in solution[39]. Therefore, a new metric has been proposed to account for hybridization propensity between oligonucleotides in arbitrary alignment[39]. This metric, termed the H-distance, captures some of the chemical constraints on hybridization, while allowing a more abstract and manageable analysis. Preliminary analysis has revealed complex, but as of yet, unexplained structure in the space of DNA hypercubes, and has presented the possibility of an error-preventing coding theory for DNA computing. Currently, the best methods for generation of good encodings for DNA computation are based on evolutionary techniques, such as genetic algorithms[40].

Hybridization errors can produce errors during the extraction process[34, 35]. An early proposal was to apply PCR at several stages of the process to overwhelm the number of bad strands with good strands[35]. This paper also proposed an encoding procedure to reduce the chance of false negatives. Errors in hybridization of primers to template can also affect the fidelity of PCR[33]. Restricting the alphabet over which information is encoded in oligonucleotides to two bases would alleviate hairpin loops which are a severe type of error, usually producing a false negative[41].

Recent results on extraction efficiency show that no better than 3 - 28 % recovery rates could be expected[42]. Other, more efficient and faithful extraction techniques have been proposed[43], but it is evident that no agreed upon solution to the problem of errors and efficient implementation has emerged, though these problems are much better understood[14, 44].

In other developments, advanced PCR techniques[45, 46, 47] have been employed to reduce the dependence on error-prone hybridizations, though hybridization cannot be eliminated. Promising results have been obtained with these PCR-based techniques with regard to fidelity, speed, and efficiency. Other promising results[48] are based on new surface technology from molecular biology[49]. By affixing oligonucleotides to a solid support and optically sensing hybridized pairs, problems with extraction, detection, and efficiency should be alleviated. A group[48] has made progress in the exploration of this approach, and DNA “chips” would seem to be a path into the future for practical implementations[50].

8. Future

The question is: are the problems facing practical, controllable, and reliable DNA computing insurmountable? There are indications[51] that the design of oligonucleotides and molecular biology protocols for error-tolerant computing is as difficult as the original combinatorial problems for which the technique first showed such promise. If this turns out to be the case, then, success may depend on compilation of oligonucleotide collections and protocols which enable successful applications. Current protocols were not developed for computational applications. It may be that with current technology and understanding, a precise enough characterization of DNA chemistry and physics is not possible for building DNA computers. On the other hand, continued exploration of current protocols and techniques may eventually result in a new body of methods that are specifically adapted to computing with DNA[52]. Another problem is the lack of a “killer app” [53]. Part of this is certainly associated with the problems that DNA computing has experienced, and the hope is that once the problems are resolved, the applications will come. It has already been suggested[53] that the “killer app” might be the application of DNA computing to problems in molecular biology.

Several people[54, 47, 55] have mentioned that nature apparently has either solved or avoided the error and efficiency problems that have confronted DNA computing. Nature uses changes in molecular sequence (what we have so far called an error) to introduce variation and change in a population[54, 47], and has produced solutions to evolutionary problems without an exponential explosion in the amount of DNA required[55]. Therefore, a new direction for DNA computing is the exploration of evolutionary paradigms and the links between DNA computing and molecular evolution. Several applications for evolutionary computing with DNA have been proposed[54, 56]. The connections between DNA computing and molecular evolution are the subject of an upcoming workshop[57], and several techniques from molecular evolution, such as *in vitro* evolution[58] and DNA shuffling[59], have already been explored[47].

Given the difficulties with implementing traditional algorithms in DNA and their potential for evolutionary-style computation, DNA computers apparently follow Michael Conrad’s trade-off principle[60]: “a computing system cannot at the same time have high programmability, high computational efficiency, and high evolutionary adaptability.” He describes programmability as the ability to communicate programs to the computing system exactly with a finite alphabet and finite number of steps. The efficiency of a computing system is defined as the ratio of the number of interactions in the system that are used for computing and the total number of interactions possible in the system, and the evolutionary adaptability is defined as the ability of the system to change in response to uncertainty.

The problem with programming DNA computers is that it is difficult to communicate the program exactly. To do this, the free energies of formation (Eq. 2) for every possible hybridization alignment between oligonucleotides and the resulting equilibrium concentrations (Eq. 5) would have to be calculated. A similar problem from computational biology is the determination of all possible alignments of two sequences[61]. An alignment involves the insertion of spaces in arbitrary locations so that two sequences, which may be of different initial length, are the same final length[62]. For each possible alignment between two

sequences, a score is computed based on whether each column in the alignment contains a match, a mismatch, or a space. The similarity between two sequences is the maximum score over all alignments. This alignment problem from molecular biology has much in common with the problem of determining a good encoding for a DNA computation. Unlike the traditional alignment problem, however, to measure the strength of a hybridization potential, we have to measure similarity under Watson-Crick complementation. The cost is associated, in the case of perfect Watson-Crick complements, with the thermodynamic parameters (ΔG) of the nearest neighbor base stacks[16]. Mismatched base pairs can also be assigned a thermodynamic cost, as well as dangling ends and bulge loops[14]. The number of alignments, however, between two sequences is exponential[61]. Therefore, for large problems, the number of interactions (*i. e.* alignments) makes enumeration impossible, and by Conrad's[60] definition, a DNA computer is not effectively programmable. Efficiency can be sacrificed by reducing the size of the problem, using less of the possible oligonucleotide interactions, and therefore, making the number alignments that need to be checked for the problem smaller. In this case, the degree of programmability increases. Alternatively, programmability could be sacrificed in favor of computational efficiency and evolutionary adaptability. Conrad's trade-off principle[60] may be suitable theoretical framework to characterize the trade-offs in DNA computing.

9. Conclusion

The field of DNA computing 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 computing. In addition, new paradigms based on molecular evolution have emerged from molecular biology to inspire new directions in DNA computing. As has been the case in the recent development of "hot" new fields, only further work will allow the determination of the proper scope and niche of DNA computing.

References

- [1] J. D. Watson, N. H. Hopkins, J. W. Roberts, J. A. Steitz, and A. M. Weiner, *Molecular Biology of the Gene*. Menlo Park, CA: The Benjamin/Cummings Publishing Co., Inc, fourth ed., 1987.
- [2] L. Stryer, *Biochemistry*. New York: W. H. Freeman and Company, fourth ed., 1995.
- [3] J. von Neumann, *Theory of Self-Reproducing Automata*. Urbana, IL: University of Illinois Press, third ed., 1966.
- [4] S. Ulam, "On some mathematical problems connected with patterns of growth of figures," in *Essays on cellular automata* (A. E. Burks, ed.), pp. 219–231, Chicago: U. of Illinois Press, 1972.
- [5] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press, 1992.
- [6] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, pp. 1021–1024, 1994.
- [7] DIMACS, *Proceedings of the First Annual Meeting on DNA Based Computers*, (Providence, RI), American Mathematical Society, 1996. DIMACS Proc. Series No. 27.
- [8] DIMACS, *Preliminary Proceedings of the Second Annual Meeting on DNA Based Computers*, (Providence, RI), American Mathematical Society, 1997. DIMACS Proc. Series.

- [9] DIMACS, *Preliminary Proceedings of the 3rd DIMACS Workshop on DNA Based Computers*, (Providence, RI), American Mathematical Society, 1997. Philadelphia, PA., June 23-27.
- [10] IEEE, *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, 1997. Indianapolis, IN, April 13-16.
- [11] AAAI, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, 1997. Stanford University, July 13-16, 1997.
- [12] J. M. H. Dassen, "A bibliography of molecular computation and splicing systems." <http://www.wi.LeidenUniv.nl/~jdassen/dna.html>.
- [13] J. Sambrook, E. F. Fritsch, and T. Maniatis, *Molecular Cloning: A Laboratory Manual*. Cold Spring Harbor Laboratory Press, second ed., 1989.
- [14] J. G. Wetmur, "Physical chemistry of nucleic acid hybridization," in *Preliminary Proceedings of the 3rd DIMACS Workshop on DNA Based Computers* [9], pp. 1-25. Philadelphia, PA., June 23-27.
- [15] P. N. Borer, B. Dengler, I. Tinoco, Jr., and O. C. Uhlenbeck, "Stability of ribonucleic acid double-stranded helices," *J. Mol. Biol.*, vol. 86, pp. 843-853, 1974.
- [16] J. SantaLucia, Jr., H. T. Allawi, and P. A. Seneviratne, "Improved nearest-neighbor parameters for predicting DNA duplex stability," *Biochemistry*, vol. 35, pp. 3555-3562, 1996.
- [17] C. H. Bennett, "Logical reversibility of computation," *IBM J. Res. Dev.*, vol. 17, pp. 525-532, 1973.
- [18] T. Head, "Formal language theory and DNA: An analysis of the generative capacity of specific recombination behaviors," *Bull. Math. Biology*, vol. 49, pp. 737-759, 1987.
- [19] R. Freund, L. Kari, and G. Paun, "DNA computation based on splicing : the existence of universal computers." Technical Report 185-2/FR-2/95, 1995. TU Wien.
- [20] E. Laun and K. J. Reddy, "Wet splicing systems," in *Preliminary Proceedings of the 3rd DIMACS Workshop on DNA Based Computers* [9], pp. 115-126. Philadelphia, PA., June 23-27.
- [21] M. R. Garey and D. S. Johnson, *Computers and Intractability*. New York: Freeman, 1979.
- [22] R. J. Lipton, "DNA solution of hard computational problems," *Science*, vol. 268, pp. 542-545, 1995.
- [23] E. B. Baum, "Building an associative memory vastly larger than the brain," *Science*, vol. 268, pp. 583-585, 1995.
- [24] D. Beaver, "Molecular computing." Technical Report TR95-001, January 1995. Penn State University.
- [25] D. Boneh, C. Dunworth, and J. Sgall, "On the computational power of DNA," *Discrete Applied Mathematics*, vol. 71, pp. 79-94, 1996.
- [26] P. W. K. Rothmund, "A DNA and restriction enzyme implementation of Turing machines," in *Proceedings of the First Annual Meeting on DNA Based Computers* [7]. DIMACS Proc. Series No. 27.
- [27] M. Ogihara and A. Ray, "Simulating Boolean circuits on a DNA computer." Technical Report TR 631, August 1996. University of Rochester.
- [28] J. H. Reif, "Local parallel biomolecular computation," in *Preliminary Proceedings of the 3rd DIMACS Workshop on DNA Based Computers* [9], pp. 243-264. Philadelphia, PA., June 23-27.
- [29] M. Ogihara and A. Ray, "DNA-based parallel computation by "Counting"," in *Preliminary Proceedings of the 3rd DIMACS Workshop on DNA Based Computers* [9], pp. 265-274. Philadelphia, PA., June 23-27.
- [30] J. Hartmanis, "On the weight of computations," *Bulletin of the European Association for Theoretical Computer Science*, vol. 55, pp. 136-138, 1995.

- [31] N. C. Seeman, H. Wang, B. Liu, J. Qi, X. Li, X. Yang, F. Liu, W. Sun, Z. Shen, R. Sha, C. Mao, Y. Wang, S. Zhang, T. Fu, S. Du, J. E. Mueller, Y. Zhang, and J. Chen, "The perils of polynucleotides: The experimental gap between the design and assembly of unusual DNA structures," in *Preliminary Proceedings of the Second Annual Meeting on DNA Based Computers* [8], pp. 191–205. DIMACS Proc. Series.
- [32] R. Deaton, R. C. Murphy, M. Garzon, D. R. Franceschetti, and S. E. Stevens Jr., "Good encodings for DNA-based solutions to combinatorial problems," in *Preliminary Proceedings of the Second Annual Meeting on DNA Based Computers* [8], pp. 159–171. DIMACS Proc. Series.
- [33] P. D. Kaplan, G. Cecci, and A. Libchaber, "Dna-based molecular computation: template-template interactions in PCR," in *Preliminary Proceedings of the Second Annual Meeting on DNA Based Computers* [8], pp. 131–140. DIMACS Proc. Series.
- [34] R. Karp, C. Kenyon, and O. Waarts, "Error-resilient DNA computation," in *Proceedings of the 7th ACM-SIAM symposium on discrete algorithms*, pp. 458–467, ACM Press/SIAM, 1996.
- [35] D. Boneh, C. Dunworth, J. Sgall, and R. J. Lipton, "Making DNA computers error resistant," in *Preliminary Proceedings of the Second Annual Meeting on DNA Based Computers* [8], pp. 102–110. DIMACS Proc. Series.
- [36] S. A. Kurtz, S. R. Mahaney, J. S. Royer, and J. Simon, "Active transport in biological computing," in *Preliminary Proceedings of the Second Annual Meeting on DNA Based Computers* [8], pp. 111–122. DIMACS Proc. Series.
- [37] E. B. Baum, "DNA sequences useful for computation," in *Preliminary Proceedings of the Second Annual Meeting on DNA Based Computers* [8], pp. 122–127. DIMACS Proc. Series.
- [38] R. Deaton, M. Garzon, J. A. Rose, D. R. Franceschetti, R. C. Murphy, and S. E. S. Jr., "Reliability and efficiency of a DNA based computation," *Phys. Rev. Lett.*, vol. 80, pp. 417–420, 1998.
- [39] M. Garzon, R. Deaton, P. Neathery, R. C. Murphy, S. E. Stevens Jr., and D. R. Franceschetti, "A new metric for DNA computing," in *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pp. 479–490, AAAI, 1997. Stanford University, July 13-16, 1997.
- [40] R. Deaton, R. C. Murphy, M. Garzon, D. R. Franceschetti, and S. E. Stevens Jr., "Genetic search of reliable encodings for dna-based computation," in *Late Breaking Papers*, (Stanford University), First Genetic Programming Conference, 1996.
- [41] K. U. Mir, "A restricted genetic alphabet for DNA computing," in *Preliminary Proceedings of the Second Annual Meeting on DNA Based Computers* [8], pp. 128–130. DIMACS Proc. Series.
- [42] J. Khodor and D. K. Gifford, "The efficiency of sequence-specific separation of DNA mixtures for biological computing," in *Preliminary Proceedings of the 3rd DIMACS Workshop on DNA Based Computers* [9], pp. 26–34. Philadelphia, PA., June 23-27.
- [43] J. Chen and D. H. Wood, "A new DNA separation technique with low error rate," in *Preliminary Proceedings of the 3rd DIMACS Workshop on DNA Based Computers* [9], pp. 43–56. Philadelphia, PA., June 23-27.
- [44] A. J. Hartemink and D. K. Gifford, "Thermodynamic simulation of deoxyoligonucleotide hybridization for DNA computation," in *Preliminary Proceedings of the 3rd DIMACS Workshop on DNA Based Computers* [9], pp. 15–25. Philadelphia, PA., June 23-27.
- [45] M. Arita, A. Suyama, and M. Hagiya, "A heuristic approach for Hamiltonian Path Problem with molecules," in *Genetic Programming 1997: Proceedings of the Second Annual Conference* [11], pp. 457–461. Stanford University, July 13-16, 1997.
- [46] N. Morimoto, M. Arita, and A. Suyama, "Solid phase DNA solution to the Hamiltonian path problem," in *Preliminary Proceedings of the 3rd DIMACS Workshop on DNA Based Computers* [9], pp. 83–92. Philadelphia, PA., June 23-27.

- [47] P. D. Kaplan, D. S. Thaler, and A. Libchaber, "Parallel overlap assembly of paths through a directed graph," in *Preliminary Proceedings of the 3rd DIMACS Workshop on DNA Based Computers* [9], pp. 127–141. Philadelphia, PA., June 23-27.
- [48] Q. Liu, Z. Guo, A. E. Condon, R. M. Korn, M. G. Lagally, and L. M. Smith, "A surface based approach to DNA computation," in *Preliminary Proceedings of the Second Annual Meeting on DNA Based Computers* [8], pp. 206–216. DIMACS Proc. Series.
- [49] M. Chee, R. Yang, E. Hubbell, A. Berno, X. C. Huang, D. Stern, J. Winkler, D. J. Lockhart, M. S. Morris, and S. P. A. Fodor, "Accessing genetic information with high-density DNA arrays," *Science*, vol. 274, pp. 610–614, 1996.
- [50] A. G. Frutos, Q. Liu, A. J. Thiel, A. M. W. Sanner, A. E. Condon, L. M. Smith, and R. M. Corn, "Demonstration of a word design strategy for DNA computing on surfaces," *Nucleic Acids Research*, vol. 25, p. 4748, 1997.
- [51] M. Garzon, R. Deaton, P. Neather, L. F. Nino, and M. Wittner, "Encoding genomes for DNA computing." submitted to GP-98.
- [52] R. C. Murphy, R. Deaton, S. E. Stevens Jr., D. R. Franceschetti, J. A. Rose, and M. Garzon, "A new algorithm for DNA based computation," in *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, pp. 207–212, IEEE, 1997. Indianapolis, IN, April 13-16.
- [53] L. Landweber and R. J. Lipton, "DNA²DNA computations: A potential "Killer App"," in *Preliminary Proceedings of the 3rd DIMACS Workshop on DNA Based Computers* [9], pp. 59–68. Philadelphia, PA., June 23-27.
- [54] R. Deaton, R. C. Murphy, J. A. Rose, M. Garzon, D. R. Franceschetti, and S. E. Stevens Jr., "A DNA based implementation of an evolutionary search for good encodings for dna computation," in *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, pp. 267–272, IEEE, 1997. Indianapolis, IN, April 13-16.
- [55] W. P. C. Stemmer, "The evolution of molecular computation," *Science*, vol. 270, p. 1510, 1995.
- [56] R. Deaton, M. Garzon, J. A. Rose, D. R. Franceschetti, R. C. Murphy, and S. E. S. Jr., "DNA based artificial immune system for self-nonsel discrimination," in *The Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, 1997. Orlando, FL, Oct. 12-15.
- [57] L. Landweber, "DIMACS workshop on nucleic acid selection and computing." <http://dimacs.rutgers.edu/Workshops/NucleicAcid/>.
- [58] A. D. Ellington and J. W. Szostak, "In-vitro selection of RNA molecules that bind specific ligands," *Nature*, vol. 346, pp. 818–822, 1990.
- [59] W. P. Stemmer, "DNA shuffling by random fragmentation and reassembly: in vitro recombination for molecular evolution," *Proc. Natl. Acad. Sci.*, vol. 91, pp. 1057–1051, 1994.
- [60] M. Conrad, "The price of programmability," in *The Universal Turing Machine: A Half-Century Survey* (R. Herken, ed.), pp. 261–282, Wien: Springer-Verlag, 1995.
- [61] M. S. Waterman, *Introduction to Computational Biology*. New York: Chapman and Hall, first ed., 1995.
- [62] J. Setubal and J. Meidanis, *Introduction to Computational Molecular Biology*. Boston: PWS Publishing Co., 1997.