The Emergence of the Discipline of Biomolecular Computation in the US

John H. Reif

Department of Computer Science, Duke University

reif@cs.duke.edu

Received 10 Febuary 2002

Abstract This paper provides a description of the recent evolution in the US of an emerging technology known as *DNA Computation* or more generally as *Biomolecular Computation* from its early stages to its development and extension into other areas such as nanotechnology, emerging as a viable sub-discipline of science and engineering.

§1 Introduction

Biomolecular Computation (also known as *DNA Computation*) is computation done at the molecular scale, using biotechnological techniques. Our aim is to briefly overview the history this emerging technology in the US from its early stages to its curent status as viable sub-discipline of biomolecular science and engineering, highlighting the key developments in the US that impacted the field, and those that are likely to have impact in the future. Since this paper is contained in a volume covering the parallel efforts of research teams of a number of other nations, the paper has been written to emphasize the prior and on-going work within the US. An extensive prior survey is given by Reif[72], and other surveys include Kari[41], Kari and Landweber[42], Rozenberg and Salomaa[79] and Paun et al[67], the latter two emphasizing the formal language aspects of the discipline. The technical level of the discussion is limited to insure readability for the broadest possible audience. The paper first very briefly overviews the basic enabling technologies and intellectual ideas that Biomolecular Computation relies upon in biochemistry, computer science and other related fields;

these include the basic recombinant DNA operations used to amplify and edit DNA strands, as well as basic computation concepts such as that of an abstract computational model, universal computation models, algorithms, and research bounded computations. It then briefly traces the origins of Biomolecular Computation in the US from its first incarnation, beginning as a theoretical concept, and its first small scale experimental demonstrations. Then the paper describes the recent emergence of this field as a discipline with a central theme but with many divergent methodologies and goals. The paper discusses challenges and recent successes in regard to improvements in scalability, as well as the rather divergent techniques used to achieve the increasing scales. It discusses the recent grow of a diversity of applications, many that go beyond the original concept of simply executing computations. We point out some of these applications which we feel are posed to have considerable impact on the biotechnologies and nanotechnologies that enabled the field. These include: (i) design software (e.g. for design of DNA databases containing diverse sets of DNA words and for design of DNA nanostructures), (ii) simulation software (e.g. for DNA hybridization and melting temperature prediction as well as for simulation of the self-assembly of DNA nanostructures), (iii) improvements to biochemical protocols: e.g. for low error synthesis and separation operations, and (iv) improvements to biotechnologies for genomics (e.g. hybrid DNA databases with expressed or genomic DNA tagged with synthetic DNA informational tags). Other applications in nanotechnology include: 2D and 3D patterning of DNA nanostructures such as positioning of nanoelectronics on 2D and 3D DNA lattice substrates, and molecular robotics. Also described are possible impacts on other fields such as intelligence (e.g., DNA cryptography and DNA steganography). The diversity of applications and techniques motivates the suggestion that the field be viewed as encompassing an emerging discipline of biomolecular science and engineering. Furthermore, side benefits of this discipline to education in the US are overviewed, in particular to the training of a new generation of scientists with uniquely interdisciplinary skills spanning the traditional areas in the biological and chemical sciences, as well is in computer science and mathematics.

§2 Enabling Technologies

An *enabling technology* for a field is a technology that provides key capabilities. Biomolecular Computation relies upon a number of basic enabling technologies and intellectual ideas in biochemistry, computer science and other

related fields. We have deliberately limited the technical level of the discussion so as to insure readability for the broadest possible audience, but will assume the reader is familiar with basic definitions and terminology used in biotechnology and computer science. Nevertheless, we will provide a brief discussion of these enabling technologies described in the next two subsections.

Enabling Biotechnology and Molecular Biology. The most basic properties of DNA are well known, but we wish to emphasize that in comparison to conventional data storage media, it is *far more compact*, providing in principle more than 10^{21} bits of information per gram of dehydrated DNA. The *recombinant DNA operations* for detection, amplification, and editing DNA and RNA are also well known, but here it must be emphasized that DNA and RNA have by far the most extensive and well tested set of such editing operations. This often provides the experimenter using DNA and RNA a considerable advantage and experimental design flexibility as compared to other chemistries (e.g., say as compared to carbon nanotechnology). The selective *hybridization* of single stranded DNA (ssDNA) into double stranded DNA (dsDNA) depends on base sequences and other parameters is the basis for many of these recombinant DNA operations, and there are a number of semi-empirical models for predicting the resulting melting temperatures that provide excellent confirmation with experimental tests. While many traditional recombinant DNA operations were solution-based, there are now a number of competing 2D surface-based chemistries. DNA arrays (also known as expression arrays) as well as cell sorters provide capabilities for fast output of hybridization assays. Also, automation (e.g., quantitative PCR cyclers), have provided for the further scaling of Biomolecular Computations.

Enabling Information Technologies and Theoretical Concepts from Computer Science. The discipline of Computer Science provides more than the obvious well developed information technologies such as the conventional electronic computer and attendant software - it provides some key intellectual ideas key to the development of a theory of Biomolecular Computation. An *abstract machine model* is a precisely defined mathematical model of computation, e.g. the Turing Machine (with an input tape and storage tape and finite state transitioning). *Universal computation* is the ability of a single abstract machine to simulate any arbitrary computation on a conventional computational model. An *algorithm* is a precise specification of a procedure for solving a problem. In the context of a Biomolecular Computation, an algorithm might be considered to be a laboratory protocol, abstracted to be executable on a given abstract

machine model. The *computational complexity* of a problem is determined by the resources consumed by the least costly algorithm to solve the problem. Resources for a Biomolecular Computation are time and the number and length of DNA strands required.

The Interaction of Theory and Practice. The strong interaction of theory and practice is made clear in the history of DNA computing in the US briefly recounted in the following Sections 3 4, and 5 where we deliberately relate the discussions of theoretical progress with our descriptions of experimental progress and vise versa, to illustrate the interplay between theory and practice in the field. The theoretical work in the US is based on abstract models for the biochemical reality, and in many cases the knowledge of this biochemical reality has been obtained from interaction with the experimentalists. The theoreticians have often taken the task of modeling experimental reality seriously and have in many cases developed a software component, that provides capability for computer automated design and computer simulation. The experimentalists have relied in a number of cases on collaboration with theorists for DNA word encodings. Theory research in this area also has lately stressed algorithm development over universality results and lower bounds, and this has led to new experimental protocols.

§3 The Early History of Biomolecular Computation in the US

We now briefly trace the origins of Biomolecular Computation in the US in its first incarnations, beginning as a theoretical concept, and its first small scale experimental demonstrations, and immediately subsequent theoretical work.

Theoretical Underpinnings of Biomolecular Computation: Head's splicing systems. Predating any other work on Biomolecular Computation are the pioneering 1987 and 1992 papers of Head[34, 35], which used formal language techniques (Lindenmayer systems) to provide the first abstract models of Biomolecular Computations. His *splicing* model represented DNA strands as strings over a finite alphabet and specified a set of editing operations on these strings to model recombinant DNA operations such as cutting and appending. Much of the subsequent work in splicing has continued to emphasize formal language techniques over experimental techniques, and has been done in Europe and Japan, so is outside the domain of this discussion centered on US work.

The Adleman Experiment and Its Impact. The ground-breaking experiment of Adleman[1)] demonstrated the use of recombinant DNA techniques to solve a small seven node instance of the combinatorial search problem known as the Hamiltonian graph problem, which is to find a path in a graph that visits all nodes exactly once. The experimental protocol used many of the elements common to a number of current Biomolecular Computations: (i) it used a sequence of short oligonucleotides to encode information (in this case names of nodes), (ii) it used a self-assembly technique to generate a combinatorial library of DNA strands representing possible solutions(in this case, these are paths) to the combinatorial problem, (iii) is used a series of subsequent recombinant DNA operations to insure the proper conditions(in this case, to insure the path visits all nodes exactly once), and (iv) it provided a means for output (in this case via jell electrophoresis). Adleman's experiment essentially defined the experimental field of Biomolecular Computation. Its impact was extensive, inspiring a host of further work, both experimental and also theoretical.

Scaling Issues. The initial work of Adleman may have caused unreasonable expectations about the goals and capabilities of Biomolecular Computations. One possible goal of Biomolecular Computation might be to solve large combinatorial search problems; mimicking the above approach, we might: generate a combinatorial library of DNA strands, where each distinct DNA strand encodes a possible solution of the problem, and then execute further recombinant DNA operations to separate out those strands encoding a correct solution. This approach is however not scalable, since in general the number of distinct DNA strands can grow exponentially with the size of the problem statement. The molecular scale of data storage is quickly swamped by the numbers of DNA strands required for problem instances of large size. Hence, the use of this approach to solve very large combinatorial search problems appears not to be feasible. Nevertheless, combinatorial search problems, such as moderate size instances of the SAT problem, discussed below in Section 4, provide a useful test problems for demonstrating Biomolecular Computation techniques. Furthermore, small to medium scale SAT problems, if solved entirely in the biolomolecular domain, may have important applications such as to executing Boolean queries in Biomolecular Databases, as described in Section 6.

§4 Experimental Demonstrations of Biomolecular Computations

Here we overview some of the experimental work in the area since the original Adleman experiment.

4.1 Experiments Solving SAT Problems

The *Satisfiability Problem (SAT)* is a combinatorial search problem posed as follows: given a Boolean formula of n Boolean variables, find a truth assignment to the variables so as to satisfy the formula. (In the formulations of SAT discussed below, the formula is presented in conjunctive normal form as an AND of a series of clauses, each clause consisting of at most 3 variables or their complements.) SAT is of a class of problems (NP complete problems) which computer scientists view as unlikely to have a fast solution on a conventional computer. In spite of the known scaling issues describe above in Section 3 for such combinatorial search problems, the SAT problem has been used as a test example for research groups wishing to demonstrate their techniques for Biomolecular Computation. Generally, these methods involve the design of a combinatorial library of synthetic DNA strands, each of with a base sequence with n words, each word associated with a Boolean variable. As discussed below, these groups in the US have used a wide variety of methods to do this word encoding and a wide variety of biochemical methods to separate out or identify the strands encoding an assignment of variables satisfying the formula. In some cases, each of these words also encode a possible truth value for the Boolean variable, while in other cases the truth value for the Boolean variable is indicated by hybridization of a complementary strands of ssDNA.

• **Surface-Based Experimental Approaches.** A group in University of Wisconsin[55, 56, 18, 25, 89] (the research team included a number of experimentalists (Smith, Corn and Lagally) as well as a computer scientist (Condon) who designed the word encoding) have used surface chemistry techniques to solve a 4 variable instance of SAT. In their experiments, they affix the DNA stands encoding all possible Boolean variable assignments onto 2D surface, and then proceed to apply restriction enzymes to destroy those DNA strands that do not satisfy the Boolean formula. A fluorescent optical readout is used to identify the remaining DNA strands that encode satisfying assignments for the given Boolean formula.

• **An Experimental Approach using RNA.** A group at Princeton[20, 23, 24] used a combination of DNA and RNA techniques to solve a 9 variable instance of a SAT problem related to the *knights* problem in chess. They employed evo-

lutionary methods to determine a set of RNA sequences used to eliminate those strands that do not satisfy the Boolean formula. This work was a collaboration between a computer scientist, Lipton, and experimentalist, Landweber, who had expertise in evolutionary methods.

• **Separation Based Experimental Approaches.** Adleman's group[15] utilized hybridization of short *sticker* strands of ssDNA onto DNA words to encode variable assignments of Boolean variables. In their protocols, they need to separate out specified DNA strands - namely, those that have stickers encoding satisfying assignments of the Boolean formula. This lead to an extensive research effort by Adleman to develop technology for exquisitely sensitive and error-resistant separation of a small set of specified molecules from a large combinatorial set of molecules, a topic that has many other diverse biotechnology and security applications. Recently, in a major breakthrough[78] was achieved by Adleman's group, who applied these techniques, using automated gel electrophloresis repeated the separation steps, to solve a 20 variable instance of SAT.

4.2 Biomolecular Computations Executing Arithmetic

A group led by Bancroft[30] demonstrated execution of the first single bit addition operations in recombinant DNA, but their protocol did not allow for chaining of further arithmetic steps operations. A group led by Rubin[83] gave an experimental demonstration of reversible arithmetic operations with chaining of the output of arithmetic operations into the inputs to further operations. The remaining challenge here is to extend the scale to allow for multibit arithmetic operations to be executed in massive parallel fashion.

4.3 Biomolecular Computations Executing Multiple Steps Without Outside Mediation

In all the above work, the Biomolecular Computations required considerable outside mediation; each step of the computations generally requires multiple laboratory operations to be executed, either by an individual or via automation. One interesting challenge is to develop methods that operate multiple steps autonomously, without outside mediation. This will be termed here *autonomous computation*. As a a concrete example, we discuss the execution of multiple steps of a finite state machine. (Recall a *finite state machine* is an abstract computational machine that has a finite number of states and has a fixed

set of rules that determine how it can transition between these states, accept inputs and produce outputs.) The execution of multibit addition can be viewed as an example of a finite state transition system, but the above cited methods require multiple laboratory operations per step of computation. As detailed below, some other rather diverse methods can be applied to realize autonomous computation:

• **Cellular Computation.** A novel approach to Biomolecular Computation being investigated in the US is to utilize a microorganisms such as bacteria. The idea is to modify the regulatory feedback systems used in cellular metabolism, so as to program behavior that represents computation, by re-engineering the regulatory feedback systems used in cellular metabolism. The inputs and outputs to the computation may be via protein metabolism, modifications of the chemistry nearby the cell or optical changes (e.g., light reception and/or fluorescence output). This approach first appeared in a science fiction short story of Bear[9]. Knight[47] gave a design for finite state transitioning and logic gates, using cellular processing and subsequently made some small scale experimental demonstrations. Ji[37] and Kazic[46] also describe methods for executing Cellular Computations. Unfortunately, re-engineering cellular regulatory feedback systems is an extremely challenging task at this time and the result can easily kill the cell rather elicit the extended behavior. The number of steps that can be executed without outside mediation is at this time limited to the small number of states that can be so modified.

• **Whiplash PCR.** This is a technique originally developed by a Japanese group (Hagiya et al[31]) to execute finite state transitions, using a series of hybridizations between the end segments of a DNA strand and the interior of the same strand to do editing and processing of data encoded by the DNA strand. Winfree[93] at Caltech has proposed improvements[93] of Whiplash PCR techniques methods to allow them to solve SAT problems. Further improvements of Sakamoto et al[84] provided for methods for making a bounded number of multiple steps without outside mediation, using a sequence of hybridizations with decreasing melting temperatures. The number of steps that can be executed without outside mediation is limited to a small number due to the bounded range (of approximately at most 25 C) of feasible melting temperatures, and the requirement that each step have a distinguishable melting temperature difference (of approximately at least 2 C).

• **Autonomous computation using restriction enzymes and ligase.** An

Israeli group led by Shapiro[12] demonstrated a simple autonomous computation using restriction enzymes and ligase applied to dsDNA to execute the state transitions of a small finite automaton. Since the yield would geometrically decrease with the number of steps, the number of autonomous steps possible may be quite limited.

Self-Assembled Nanostructures. In this approach, a computation is executed by the self-assembly of DNA nanostructures(see surveys of[73, 76]) from component ssDNA strands. The self-assembly occurs in multiple stages. In the first stage, DNA nanostructures known as *DNA tiles* self-assemble from component ssDNA strands as well possibly as input DNA strands encoding computational inputs. These DNA tiles have ends with protruding ssDNA known as *pads*. In further stages, these DNA tiles self-assemble by hybridization at their pads into 1D, 2D, or 3D lattices. The structure of these lattices is determined by the tile pads, and so the programming of the computation essentially amounts to determining the set of tiles and their pads. Surveys of recent work in the self-assembly of DNA lattices are given by Reif[73, 76]. The enabling technology here is that of DNA nanostructures, pioneered by Seeman[27, 87]. The idea of using the self-assembly of DNA tiles is due to Winfree[90, 91, 97]. The DNA tiles use strand crossovers[52] known as Holliday junctions to stabilize their structure. DNA tiles come in many varieties, but the most commonly used are known as DX and TX tiles: the DX tiles[95] consist of two dsDNA that have Holliday junctions, and the TX tiles[48] consist of three dsDNA that have Holliday junctions. Winfree and Seeman[94, 86] gave the first demonstrations of 2D DNA tile lattices, which were visualized by atomic force microscopy[58]. Mao and Seeman in collaboration with LaBean and Reif[59] gave the first experimental demonstration of a computation executed by a the self-assembly of DNA nanostructures. That computation was a cumulative exclusive OR of a sequence of bits, equivalent to the carry sum computation required in integer addition. This seems the most promising of the known methods for executing multiple steps without outside mediation, since the number of steps that may be executed has no bound, and the programming is quite straightforward. For example, self-assembly techniques[49] may be used to execute a variety of computations, such as multibit addition, in massive parallel fashion.

§5 Theoretical Work in Biomolecular Computation

Initial Theoretical Work following Adleman's Experiment: Abstract Computational Models. One immediate consequence of the Adleman experiment was the development (Lipton[54, 15]) of the first abstract computational models of Biomolecular Computations and algorithms in these models to solve SAT. The later model[15] was the basis the separation-based experimental approach to solve SAT developed by Adleman's group.

Decreasing the Volume Used in Combinatorial Search. In all the experimental methods for solving SAT described above, the number of steps grows as a polynomial function of the number n of Boolean variables, but the volume grows exponentially with the input, that is as c^n for some constant $c \geq 2$. A number of methods have been proposed to decrease the search space and the volume (which is proportional to the total number of strands used). Instead of initially generate a very large volume containing all possible solutions, an alternative heuristic approach of iteratively refining the solution space. was first suggested by Hagiyal[31]. Ogihara and Ray[64] also proposed a method which utilized a known method to somewhat reduce the constant base c of the exponential growth rate of the search space required to solve a SAT problem.

Universal Computation Results. Theoretical researchers have been able to show that in a wide variety of abstract computational models for Biomolecular Computations (including various splicing models and other models for recombinant DNA operations) provide for universal computation simulating any conventional computation. While these list of these results are too numerous to be listed here, we briefly mention three example results:

- Kari et al[45] proved the universality of sticker systems for DNA computation.
- Winfree[90, 91] proved that the self-assembly of DNA nanostructures allows for universal computation. This work used techniques similar to those used by Berger[13] to show the universality of certain domino tiling problems.
- Certain cell recombination processes[50, 51] were shown[43, 44] to allow for universal computation. This work was a collaboration between a theoretician (Kari) and an experimentalist (Landweber), and combined detailed modeling of recombination processes in the cell with theoretical techniques derived from formal language theory - a dual approach now also being pursued by other interdisciplinary groups[69, 81].

Resource Bounded Biomolecular Computations: PSPACE and Parallel Computation Results. The above cited universal computation results have limited practical utility since some resource (generally strand length) inevitably

grows unboundedly. Recall that the key resources of Biomolecular Computations include number of steps, strand length and number of strands. Theoretical researchers have also investigated the power of abstract computational models for Biomolecular Computations which have restrictions on these resources. A few definitions will be required, including of language classes which we will view for simplicity as computation classes. A *polynomial* function of n grows at rate $n^{O(1)}$ and a *polylog* function grows at rate $\log^{O(1)} n$. *PSPACE* is a class of computations executed by conventional computers with a polynomial bound on the memory (as a function of the input size n), but no such bound on number of computational steps. *NC* is a class of computations executed by parallel computers with a polynomial number of processors in polylog time. Reif[70] (and independently a number of other authors[11, 26]) proved that Biomolecular Computations restricted to polynomial strand length can solve PSPACE problems. Reif[70] also proved that these Biomolecular Computations can also execute NC computations in polylog time, and Ogihara and Ray[65] subsequently extended these results to improved parallel circuit evaluation.

Theoretical Work in self-assembly of DNA nanostructures. The key resources of Biomolecular Computations using the self-assembly of DNA nanostructures also include the size and depth of the resulting assembled DNA lattices. Reif[70] also proved that the self-assembly of DNA nanostructures with assemblies of polynomial size and polylog depth can be used to execute NC computations. The 1D tilings used in the experimental demonstrations[49, 59] were derived from the theoretical 1D tiling assemblies of Reif[70] and[49]. The number of tiles required in a self-assembly is also a resource; lower and upper bounds for various assembly problems are given in Adleman[4], Rothemund et al[77] and Adleman et al[5]. Jonoska contributed to the theory of DNA self-assembly by proposing methods for creating 3D structures[40] as well as a method for solving a combinatorial search problem known as graph coloring[39]. Lagoudakis and LaBean[57] proposed a 2D DNA self-assembly for SAT, which uses parallel construction of multiple self-assembling 2D DNA lattices to solve the problem.

5.1 Mathematical Models and Software

DNA Hybridization and Melting Temperature Prediction. One byproduct of the work in Biomolecular Computations has been an extensive work in mathematical models for DNA hybridization thermodynamics simulation and prediction of melting temperatures, extending prior work in this area. While

this literature is too extensive to cite completely, For example, the work of[32] is notable since it resulted in the software BIND that has been used by a number of groups working in Biomolecular Computation.

DNA Word Design. This is the problem of designing of a library of short short oligonucleotides (DNA words) for information storage. Word design is crucial to error control in BMC. Ideally, a good word design will minimize unwanted secondary structure, and minimize mismatching, by maximizing binding specificity. Note that there are conflicting requirements on word design for BMC: as strand length decreases (which is desirable), the difference between distinct words of information decreases (which is not desirable). Adleman[1] and Lipton[54] first suggested the use of random strings for word design, noting that DNA strings are non-degenerate with high likelihood. In addition to randomization, the range of methods used have included the use of combinatorial designs, error correcting codes, greedy search, and evolutionary search. While this literature is too extensive to cite completely in this article, we note that a four-base mismatch word design of[32] was used in the surface-based chemistry experiments of[25], and similar techniques were employed in word designs by Lynx, Inc. Extensions to word designs for some much larger DNA libraries are described in Reif[75].

Error Correction Techniques. In addition ot the use of optimized DNA word designs to decrease errors, a number of papers[16, 22, 80] propose error correction techniques for Biomolecular Computations based on improved protocols and probabilistic analysis;[74] also discusses the use of improved error-resilient biotechnology techniques inspired by classical information theory techniques (error-correcting codes and vector-quantization).

Computer Simulations of DNA Tiling. Winfree[92, 97] made kinetic simulations of computing by self-assembly of tiling lattices, using a stochastic model of assembly from single component tiles.

§6 Applications of Biomolecular Computations

There are a number of applications of Biomolecular Computations that we feel may have revolutionary impact.

Application of Biomolecular Computation on Genomic Databases and Information Technology. Information technology is being currently used to do various tasks of genomic and DNA database processing, some of which can be very computationally intensive. Furthermore, they require the transforma-

tion (via a variety of biotechnology techniques such as sequencing) of genomic information to a digital medium, which can be very lengthy and error prone. We feel that the development of a system for storage, processing and retrieval of genetic information and material is an ideal application area for Biomolecular Computation. In this approach, the DNA needs not be transformed to digital medium. Instead the molecular database system will take as input human genomic DNA or reverse transcript cDNA obtained from mRNA. The input DNA strands will be fragmented and tagged with artificially synthesized DNA strands. These *information tags* would encode essential information and key properties (e.g., organism, cell type, date of origin, etc.) about the DNA strand. This information will be represented by a sequence of distinct DNA words, each encoding variables over a small domain. (Also, various sophisticated tagging techniques have been developed by the biotechnology industry for expression analysis and differential expression analysis: these include the SAGE tagging of Genzyme Molecular Oncology, Inc. and the randomized tagging techniques of Lynx Therapeutics, Inc.) Such a resulting *Biomolecular Database* would be capable of containing a vast store of genomic DNA obtained from many organisms, individuals, cell types and at many distinct dates. Associative searches in such a Biomolecular Database would provide the ability to retrieve material that might be used for experimental purposes. The ultra-compact nature of DNA and the ability to make selective associate searches via a variety of biotechnology methods makes this approach attractive. Baum[7] was the first to describe the use of DNA databases to execute associative searches in a database, but this was not tested until recently. We[75] have made experimental construction of large scale databases of synthetic DNA, and have tested the use of a variety of techniques (including cell sorting and PCR) for executing associative search on these databases. Also, we[74] have developed methods for limiting the errors in these searches. In principle the Biomolecular Computing techniques described above for solving SAT problems can be applied to execute Boolean and relational queries such Biomolecular Databases. The Biomolecular Database system would then have the capability of retrieval of subsets of the stored genetic material, specified by associative queries on the tags and/or the attached genomic DNA strands, as well as logical selection queries on the tags of the database.

Applications to Intelligence and Security. A number of proposals have been made for applying Biomolecular Computation to break conventional cryptographic systems such as DES(Boneh et al[14]) and RSA(Beaver[10]). However,

these proposed methods suffer from the scalability issues discussed in Section 3. Nevertheless, the ultra-compact storage of information by DNA provides potentially revolutionary applications in the area of intelligence and security. Here we give two examples: DNA-based steganography and DNA-based one-time pad cryptography

• **DNA-based steganography.** *Steganography* is a class of techniques that hide secret messages within other messages. In a steganography system, the original plaintext is not actually encrypted but is instead hidden within other data. In the Bancroft[19] procedure for DNA-based steganography, a message encoded in plaintext by DNA strands is appended by flanking *secret key* primer sequences known only to the sender and the intended recipient. The resulting *secret message* DNA strands are then hidden by mixing them within many other additional *distracter* DNA strands. Given knowledge of the secret key primer sequences, the secret message DNA strands can be separated out by a number of possible known recombinant DNA separation methods. We[28] developed some entropy-based methods that might be used to break such DNA-based steganography systems, but also suggested information theoretic methods that may be used to enhance its security.

• **DNA-based one-time pad cryptography.** We[28] also gave some procedures for DNA-based cryptography based on one-time-pads that are in principle unbreakable. Practical applications of cryptographic systems based on one-time-pads are limited in conventional electronic media, by the size of the one-time-pad; however DNA provides a much more compact storage media, and an extremely small amount of DNA suffices even for huge one-time-pads. One of our procedures for DNA-based cryptography require a exclusive OR computation simply requires a logical bit-wise XOR operations that has been executed by the using a self-assembly of DNA tiles into 1D lattices[59].

Applications to Nanotechnology:

• **Selective Positioning of Molecules via Patterned DNA lattices.** While DNA tiling Self-Assemblies have demonstrated computation, we feel a further key application will be to create patterned 2D and 3D lattices which can be used as superstructures for controlled positioning of other nanostructures such as molecular electronic components and molecular motors. A number of molecular electronic components have been have constructed and tested, including molecular wires and diodes (Reed and Tour[21]). The positioning of these is a key challenge. Alivisatos[6] and Mirkin[61, 62] have discussed the use of attached

DNA strands for positioning molecular objects. The creation of a regular 2D DNA lattices has already been demonstrated, as discussed above. These DNA tiles composing these tiling lattices can be designed to have protruding strands of ssDNA for selective self-positioning of molecular objects tagged with complementary strands. The key remaining challenge is the creation of highly patterned DNA lattices. These can in principle be constructed by the use of tile sets that are chosen to create a given 2D pattern, but these remain to be experimentally demonstrated.

• **DNA Nanorobotics.** Nanomechanical devices built of DNA have been developed using two distinct approaches: (i) Seeman's group[60] used rotational transitions of dsDNA confirmations between the B-form (right handed) to the Z-form (left-handed) controlled by an ionic effector molecules and extended this technique to be DNA sequence dependant[60, 98]. (ii) Yurke and Turberfield[99, 100, 88] demonstrated a series of DNA nanomechanical devices that used a *fuel* DNA strands acting as a hybridization catalyst to generate a sequence of motions in another *tweezers* strand of DNA; the two strands of DNA bind and unbind with the overhangs to alternately open and shut the tweezers. In principle, either of these DNA nanomechanical devices can be incorporated into self-assembled DNA lattices. Possible applications include the use of the induced movements to hold state information and to sequence between distinct confirmations. These capabilities might be used to selectively control nanofabrication stages. Also, the size or shape of the lattice may be programmed through the control of such sequence-dependent devices and this might be used to execute a series of foldings (similar to Japanese paper folding techniques) of the DNA lattice to form a variety of geometries. However, these DNA nanomechanical devices have some key restrictions that restrict their use. (a) They can only execute one type of motion (rotational or translational). It seems feasible to re-design these DNA nanomechanical devices to have both translational and rotational motion, but this still needs to be done. (b) Also, these prior DNA devices require environmental changes such as temperature cycling or bead treatment of biotin-streptavidin beads to make the machine being cycled. A key challenge is then to make an *autonomous* DNA nanomechanical device that executes cycles of motion (either rotational or translational or both) without environmental changes.

§7 Conclusion

Due to scaling issues discussed in Section 3, in our opinion Biomolecular

Computation will not be able to improve on conventional computers for traditional computational-intensive tasks such as the solution of large combinatorial search problems. We nevertheless feel that Biomolecular Computation will likely have revolutionary impact in a number of areas, including genomic databases and information technology, intelligence (e.g., security and cryptography), and nanotechnology (e.g., molecular electronic and molecular robotics). We expect that the impact of Biomolecular Computation to these areas will be due to the the leveraged advantage of combining DNA technology (e.g., properties of DNA as an ultra-compact storage medium, the selectivity of DNA hybridization, and to the ability to use recombinant DNA and other biotechnologies to controllably manipulate DNA) with computer science techniques. The impact of the field will also be seen in the multidisciplinary training of students; this field provides a reserve of personal trained in areas that span tow of the most quickly evolving disciplines (biotechnology and computer science).

Acknowledgment This work was supported by DARPA/AFSOR Contract F30602-01-2-0561, NSF ITR Grant EIA-0086015, DARPA/NSF Grant CCR-9725021.

References

1) Adleman, L., *Molecular Computation of Solution to Combinatorial Problems*, Science, **266**, 1021–7pt24, (1994).

2) Adleman, L., *On Constructing a Molecular Computer*, Dept of CS, U.S.C., Available via anonymous ftp from ftp.usc.edu /pub/csinfo/papers/adleman/molecular_computer·ps, (1995).

3) Adleman, L.M., P.W.K. Rothemund, S. Roweis, E. Winfree, *On Applying Molecular Computation To The Data Encryption Standard*, 2nd Annual DIMACS Meeting on DNA Based Computers, DIMACS Workshop, Princeton, June 10-12, 1996 (Volume 44 in DIMACS). Laura F. Landweber and Eric B. Baum, editors. American Mathematical Society, pp. 31-44, 1998. Published as Leonard M. Adleman, Paul W. K. Rothemund, Sam Roweis, Erik Winfree. "On Applying Molecular Computation to the Data Encryption Standard." Journal of Computational Biology, 6(1): 53-63, 1999.

4) Leonard Adleman, Toward a mathematical theory of self-assembly, USC Tech Report, January 2000.

5) Adleman,L., Q. Cheng, A. Goel, M. Huang, D. Kempe, P. M. de Moisset and P. W. K. Rothemund, Combinatorial Optimization Problems in Self-Assembly, *Thirty-Fourth Annual ACM Symposium on Theory of Computing*, Montréal, Québec, Canada, May 19-21, 2002.

6) Alivisatos, A.P., K.P. Johnsson, X. Peng, T.E. Wilson, C.J. Loweth, M. P. Bruchez Jr., P. G. Schultz, *Organization of 'nanocrystal molecules' using DNA*, Nature, bf 382,

609–611, August 1996.

7) Baum, E. B. How to build an associative memory vastly larger than the brain, Science pp 583-585, Vol. 268, (April 28,1995).

8) Baum, E. B. *DNA Sequences Useful for Computation*, 2nd Annual DIMACS Meeting on DNA Based Computers, Princeton University, June 1996.

9) G. Bear, *Blood Music*, Analog, (June, 1983). Also, appearing in Nanodreams, (Ed. by E Elliott), Baen Pub., (1995). Also, appearing as a full novel as *Blood Music*, Ace Pub., (1985). *Blood Music*, Analog, (june, 1983).

10) Beaver, D. *Factoring: The DNA Solution*, Advances in Cryptology, Asia Crypt94 Proceedings, Lecure Notes in Computer Science, (1994), Springer Verlag,

11) Beaver, D. *Computing with DNA*, J. Comp. Biol., **2**, 1–7, (1995).

12) Benenson, Y., T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, and E. Shapiro, Programmable and autonomous computing machine made of biomolecules. Nature, 414, 430 - 434, (2001).

13) Berger, R. *The Undecidability of the Domino Problem*, Memoirs of the American Mathematical Society, **66**, (1966). ,

14) Boneh, D., C. Dunworth, R. Lipton, *Breaking DES Using a Molecular Computer*, Princeton CS Tech-Report number CS-TR-489-95 (1995). ,

15) Boneh, D., C. Dunworth, R. Lipton, J. Sgall, *On the Computational Power of DNA*, Discrete Applied Math, (Dec 96)

16) Boneh, D., and R. Lipton, *Making DNA Computers Error Resistant*, Princeton CS Tech-Report CS-TR-491-95 Also in 2nd Annual DIMACS Meeting on DNA Based Computers,, Princeton University, June 1996.

17) R. S. Braich, C. Johnson, P. W.K. Rothemund, D. Hwang, N. Chelyapov and L. M. Adleman, Solution of a Satisfiability Problem on a Gel-Based DNA Computer, Proceedings of the 6th International Conference on DNA Computation in the Springer-Verlag Lecture Notes in Computer Science series, 2000.

18) Cai, W., A. Condon, R.M. Corn, Z. Fei, T. Frutos, E. Glaser, Z. Guo, M.G. Lagally, Q. Liu, L.M. Smith, and A. Thiel, *The Power of Surface-Based Computation*, Proc. First International Conference on Computational Molecular Biology (RECOMB97), January, 1997. CRFCC+96

19) Clelland, C. T., V. Risca, and C. Bancroft. Hiding messages in DNA microdots. Nature 399:533-4 (1999).

20) Cukras, A., D. Faulhammer, R. Lipton, L. Landweber, *Chess games: A model for RNA-based computation*, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

21) CRR+99 J. Chen, M. A. Reed, A. M. Rawlett, and J. M. Tour, Observation of a Large On-Off Ratio and Negative Differential Resistance in an Electronic Molecular Switch, Science, 286, 1550 (1999).

22) Chen, K., E. Winfree. "Error Correction in DNA Computing: Misclassification and Strand Loss." In Proceedings of the Fifth Annual Meeting on DNA Based Computers, held at MIT, June 14-15, 1999.

23) Faulhammer, D., Cukras, A. R., Lipton, R. J. and L. F. Landweber (2000) Molecular Computation: RNA Solutions to Chess Problems. Proc. Natl. Acad. Sci. USA. 97:1385-1389.

24) Faulhammer, D., Lipton, R. J. and L. F. Landweber (2000) When the knight falls: On constructing an RNA computer. In DNA Based Computers V, E. Winfree, D. Gifford, eds. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 54: 1-7.

25) Frutos, A.G., A.J. Thiel, A.E. Condon, L.M. Smith, R.M. Corn, *DNA Computing at Surfaces: 4 Base Mismatch Word Design*, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., (June, 1997).

26) Fu, B., R. Beigel, *Length bounded molecular computing*, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

27) Fu, T.-J., and N.C. Seeman, *DNA Double Crossover Structures*, Biochemistry, **32**, 3211–3220, (1993).

28) A. Gehani, T. H. LaBean, and J.H. Reif, DNA-based Cryptography, 5th DIMACS Workshop on DNA Based Computers, MIT, June, 1999. To appear in DNA Based Computers, V, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (ed. E. Winfree), American Mathematical Society, 2000. GR98a

29) Guarnieri, F., and C. Bancroft, *Use of a Horizontal Chain Reaction for DNA-Based Addition*, Proceedings of the 2nd Annual DIMACS Meeting on DNA Based Computers., June 10-12, 1996, American Mathematical Society, Providence, RI (in press), (1996).

30) Guarnieri, F., Fliss, M., and C. Bancroft, *Making DNA add*, Add. Science, **273**, 220–223, (1996).

31) Hagiya, M., M. Arita, D. Kiga, K. Sakamoto, and S. Yokoyama, *Towards Parallel Evaluation and Learning of Boolean μ-Formulas with Molecules*, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., (June, 1997).

32) Hartemink, A.J., D.K. Gifford *Thermodynamic Simulation of Deoxyoligonucleotide Hybridize for DNA Computation*, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., (June, 1997).

33) Hartemink, A., David Gifford, J. Khodor, *Automated constraint-based nucleotide sequence selection for DNA computation*, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

34) Head, T., *Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors*, Bull. Math. Biology, **49**, 737–759, (1987).

35) Head, T., *Splicing schemes and DNA*, In: *Lindenmayer Systems: Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology*, Ed. by G.Rozenberg and A.Salomaa, Springer-Verlag, 371-383, (1992). Also appears in: *Nanobiology*, **1**, 335–342, (1992).

36) Head, T., *Splicing System and Molecular Processes*, ICEC'97 Special Session on DNA Based Computation, Indiana, April, 1997.

37) Ji, S., *The cell as a DNA-based molecular computer*, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

38) Jonoska, N, S. Karl, M. Saito, *Three dimensional DNA structures in computing*, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

39) Jonoska, N., and S.A. Karl, *A Molecular Computation of the Road Coloring Problem*, 2nd Annual DIMACS Meeting on DNA Based Computers, Princeton University, June 1996.

40) Jonoska, N., and S.A. Karl, *Creating 3-Dimensional Graph Structures with DNA*, 3nd Annual DIMACS Meeting on DNA Based Computers, University of Pens., (June 1997).

41) Kari, L., *DNA computers: tomorrow's reality*, Tutorial in the Bulletin of EATCS, no.59, pp.256-266, (1996).

42) Kari, L. and Landweber, L. F. Computing with DNA. Methods in Molecular Biology. 1999.

43) Kari, L., and L. Landweber. 1999. Computational Power of Gene Rearrangement. Proceedings of the Fifth International Meeting on DNA Based Computers. Massachusetts Institute of Technology: 14 June 1999-15 June 1999. E. Winfree, and D. Gifford. Boston: 203-213.

44) Lila Kari, Jarkko Kari, Laura F. Landweber: Reversible Molecular Computation in Ciliates. Jewels are Forever 1999: J. Karhumäki, H. Maurer, G. Paun, G. Rozenberg (eds.): Jewels are forever, contributions to Theoretical Computer Science in honor of Arto Salomaa, Springer Verlag, 353-363, 1999.

45) L.Kari, G.Paun, G.Rozenberg, A.Salomaa, S.Yu, *DNA computing, sticker systems, and universality*, Acta Informatica, **35:** 401-420, (1998).

46) Kazic, T., *After the Turing machine: A metamodel for molecular computing*, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998). KS97

47) Knight,T. F., G.J. Sussman, *Cellular Gate Technology*, MIT Artificial Intelligence Labratory, July 1997.

48) LaBean, T. H., Yan, H., Kopatsch, J., Liu, F., Winfree, E., Reif, J.H. and Seeman, N.C., The construction, analysis, ligation and self-assembly of DNA triple crossover complexes, Journal of American Chemestry Society 122, 1848-1860 (2000).

49) LaBean, T. H., E. Winfree, J. H. Reif, Experimental Progress in Computation by Self-Assembly of DNA Tilings, 5th International Meeting on DNA Based Computers(DNA5), MIT, Cambridge, MA, (June, 1999). DIMACS Series in Discrete Mathematics and Theoretical Computer Science, ed. E. Winfree, to appear American Mathematical Society, 2000. L97

50) Landweber, L., and L. Kari. 1998. The Evolution of Cellular Computing: Nature's Solution to a Computational Problem. Late Breaking Papers at the Genetic Programming 1998 Conference, Proceedings of the 3rd Annual Genetic Programming Conference. 22 July 1998-25 July 1998. J.R. Koza. San Francisco, CA: Morgan Kaufmann Publishers. 700-708.

51) Landweber, L., and L. Kari. 1999. Universal Molecular Computation in Ciliates. Proceedings of the DIMACS Workshop on Evolution as Computation. Princeton University: 11 January 1999-11 January 1999. Published in L. Landweber, E. Winfree, R. Lipton, and S. Freeland. Evolution As Computation. New York: Springer-Verlag. 51-60. 1999

52) Li, X.J., X.P. Yang, J. Qi, and N.C. Seeman, *Antiparallel DNA Double Crossover Molecules as Components for Nanoconstruction*, J. Am. Chem. Soc., **118**, 6131–6140, (1996).

53) Lipton, R.J. *DNA Solution of Hard Computational Problems*, Science, **268**, 542–845, (1995).

54) Lipton, R. *Speeding Up Computations via Molecular Biology*, Princeton University Draft, (1994).

55) Liu, Q., Z. Guo, A.E. Condon, R.M. Corn, M.G. Lagally, and L.M. Smith, *A Surface-Based Approach to DNA Computation,*, Proc. 2nd Annual Princeton Meeting on DNA-Based Computing, June 1996.

56) Liu, Q., A.J. Thiel, A.G. Frutos, R.M. Corn, L.M. Smith, *Surface-Based DNA Computation: Hybridize and Destruction*, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., (June, 1997).

57) Lagoudakis, M. G., T. H. LaBean, 2D DNA Self-Assembly for Satisfiability, 5th International Meeting on DNA Based Computers(DNA5), MIT, Cambridge, MA, (June, 1999). DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol.44, American Mathematical Society, ed. E. Winfree, (1999).

58) Mao, W. Sun and N.C. Seeman, Designed Two-Dimensional DNA Holliday Junction Arrays Visualized by Atomic Force Microscopy, J. Am. Chem. Soc. 121, 5437-5443 (1999).

59) C. Mao, LaBean, T.H. Reif, J.H., Seeman, Logical Computation Using Algorithmic Self-Assembly of DNA Triple-Crossover Molecules, Nature, vol. 407, Sept. 28 2000, pp. 493-495; C. Erratum: Nature 408, 750-750(2000).

60) Mao, W. Sun, Z. Shen and N.C. Seeman, A DNA Nanomechanical Device Based on the B-Z Transition, Nature 397, 144-146 (1999).

61) Mirkin, C.A., R.L. Letsinger, R.C. Mucic, J.J. Storhoff, *A DNA-based Method for Rationally Assembling Nanoparticles Into Macroscopic Materials*, Nature, **382**, 607–611, August 1996.

62) Mirkin, C. A. *Programming the Assembly of Two- and Three-Dimensional Architectures with DNA and Nanoscale Inorganic Building Blocks*, Inorg. Chem., 2000, 39, 2258-2272.

63) Morimoto, N., M.A.A. Suyama, *Solid Phase DNA Solution to the Hamiltonian Path Problem*, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., (June, 1997).

64) Ogihara, M., and A. Ray, *Breadth first search 3SAT algorithms for DNA computer*, Technical Report TR-629, Department of Computer Science, University of Rochester, (July 1996).

65) Ogihara, M., and A. Ray, *Simulating Boolean circuits on a DNA computer*, 1st Annual International Conference On Computational Molecular Biology (RECOMB97), Santa Fe, New Mexico, January 1997. OR97c

66) Orlian, M., F. Guarnieri, C. Bancroft, *Parallel Primer Extension Horizontal Chain Reactions as a Paradigm of Parallel DNA-Based Computation*, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., (June, 1997).

67) G. Paun, G. Rozenberg, A. Salomaa: DNA Computing - New Computing Paradigms, Texts in Theoretical Computer Science - An EATCS Series, Springer Verlag, 1998.

68) Pirrung, M. C.; Connors, R. V. Montague-Smith, M. P. Odenbaugh, A. L. Walcott, N. G. Tollett, J. J. The arrayed primer extension method for DNA microchip analysis. Molecular computation of satisfaction problems, J. Am. Chem. Soc., 122, 1873 (2000).

69) D.M. Prescott, A. Ehrenfeucht, and G. Rozenberg, Molecular operations for DNA processing in hypotrichous ciliates, European Journal of Protistology 37, 241-260, 2001.

70) Reif, J., *Parallel Molecular Computation: Models and Simulations*, Seventh Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA95), ACM,Santa Barbara,213–223,June 1995. Published in *Algorithmica*, special issue on Computational Biology, Vol. 25, No. 2, 142-176, 1999.

71) Reif, J.H., *Local Parallel Biomolecular Computation*, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., June 23-26, 1997. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, H. Rubin and D. H. Wood, editors. American Mathematical Society, Providence, RI, vol. 48, 1999, pp. 217-254.

72) Reif, J.H., Paradigms for Biomolecular Computation, First International Conference on Unconventional Models of Computation, Auckland, New Zealand, January 1998. Published in Unconventional Models of Computation, edited by C.S. Calude, J. Casti, and M.J. Dinneen, Springer-Verlag, New York, January 1998, pp 72-93.

73) J.H. Reif, T. H. LaBean, and Seeman, N.C., Challenges and Applications for Self-Assembled DNA Nanostructures, Invited paper, Proc. Sixth International Workshop on DNA-Based Computers, Leiden, The Netherlands, June, 2000. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Edited by A. Condon and G. Rozenberg. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg, vol. 2054, 2001, pp. 173-198.

74) J.H. Reif and T. H. LaBean, Computationally Inspired Biotechnologies: Improved DNA Synthesis and Associative Search Using Error-Correcting Codes and Vector-Quantization, Sixth International Meeting on DNA Based Computers (DNA6), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Leiden, The Netherlands, (June, 2000) ed. A. Condon. Springer-Verlag volume in Lecture Notes in Computer Science, (2001).

75) J. H. Reif, T. H. LaBean, M. Pirrung, V. Rana, B. Guo, K. Kingsford, and G. Wickham, Experimental Construction of Very Large Scale DNA Databases with Associative Search Capability, Seventh International Meeting on DNA Based Computers (DNA7), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Tampa, FL, June 11-13, 2001.

76) J. H. Reif, DNA Lattices: A Programmable Method for Molecular Scale Patterning and Computation, Special issue on Bio-Computation, Computer and Scientific Engineering Journal of the Computer Society. 2002.

77) P. W. K. Rothemund, E. Winfree. "The Program-Size Complexity of Self-Assembled Squares." *Thirty-Second Symposium on Theory of Computing,* May 21-23, 2000.

78) Ravinderjit,B., N. Chelyapov, C. Johnson, P. W. K. Rothemund, L. Adleman, "Solution of a 20 variable 3-SAT problem on a molecular computer", Science, to appear,2002

79) G. Rozenberg and A. Salomaa, DNA Computing: new ideas and paradigms, LNCS 1644, 106-118, Springer Verlag , 1999.

80) Sam Roweis, Erik Winfree. "On the reduction of errors in DNA computation." Journal of Computational Biology, 6(1): 65-75, 1999.

81) G. Rozenberg, Gene Assembly in Ciliates: Computing by Folding and Recombination, in: A Half-Century of Automata Theory – Celebration and Inspiration (A. Salomaa, D. Wood, S. Yu, eds.), World Scientific, New Jersey, 93-130, 2001.

82) Roweis, S., E. Winfree, R. Burgoyne, N.V. Chelyapov, M.F. Goodman, P.W.K. Rothemund, L. M. Adleman, *A Sticker Based Architecture for DNA Computation*, 2nd Annual DIMACS Meeting on DNA Based Computers, Princeton University, June 10-12, 1996 (Volume 44 in DIMACS). Laura F. Landweber and Eric B. Baum, editors. American Mathematical Society, pp. 1-29, 1998. Published as Sam Roweis, Erik Winfree, Richard Burgoyne, Nickolas V. Chelyapov, Myron F. Goodman, Paul W. K. Rothemund, Leonard M. Adleman. "A Sticker-Based Model for DNA Computation." Journal of Computational Biology, 5(4): 615-629, 1998.

83) Rubin, H., J. Klein, T. Leete, *A biomolecular implementation of logically reversible computation with minimal energy dissipation*, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

84) Sakamoto, K., D. Kiga, K. Komiya, H. Gouzu, S. Yokoyama, S. Ikeda, H. Sugiyama, M. Hagiya, *State transitions by molecules*, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

85) Seeman, N., F. Liu, C. Mao, X. Yang, L. Wenzler, E. Winfree, *DNA nanotechnology: Control of 1-D and 2-D arrays and the construction of a nanomechanical device*, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

86) Nadrian C. Seeman, Furong Liu, Chengde Mao, Xiaoping Yang, Lisa A. Wenzler, Ruojie Sha, Weiqiong Sun, Zhiyong Shen, Xiaojun Li, Jing Qi, Yuwen Zhang, Tsu-Ju Fu, Junghuei Chen and Erik Winfree. "Two Dimensions and Two States in DNA Nanotechnology." Journal of Biomolecular Structure and Dynamics, Conversation 11, No. 2, May 2000. pp. 253-262. SQLYL+96

87) Seeman, N. C., Y. Zhang, and J. Chen, *DNA nanoconstructions*, J. Vac. Sci. Technol., **12:4**, 1895–1905, (1994).

88) A.J. Turberfield, B. Yurke and A.P. Mills, Jr , DNA Hybridization Catalysts and Molecular Tweezers in DNA Based Computers V, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 54 ed. E. Winfree and D. K Gifford (American Mathematical Society, Providence, RI, U.S.A. 2000) p.171

89) Wang, L., Q. Liu, A. Frutos, S. Gillmor, A. Thiel, T. Strother, A. Condon, R. Corn, M. Lagally, L. Smith, *Surface-based DNA computing operations: DESTROY and READ-OUT"*, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

90) Winfree, E. *Complexity of Restricted and Unrestricted Models of Molecular Computation*, In DNA Based Computers: Proceedings of a DIMACS Workshop, April 4, 1995, Princeton University (Volume 27 in DIMACS). Richard J. Lipton and Eric B. Baum, editors. American Mathematical Society, 1996. pp. 199-221.

91) Winfree, E. *On the computational power of DNA annealing and ligation*, DNA based computers, Lipton, R.J. and Baum, E.B. eds., Am. Math. Soc.,(Volume 27 in DIMACS), Providence, RI, (1996).

92) Winfree, E., *Simulations of computing by self-assembly*, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

93) Winfree, E., *Whiplash PCR for O(1) computing*, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

94) Erik Winfree, Furong Liu, Lisa A. Wenzler, Nadrian C. Seeman, *Design and Self-Assembly of Two Dimensional DNA Crystals*, Nature 394: 539–544, (1998).

95) Winfree, E., X. Yang, N.C. Seeman, *Universal Computation via Self-assembly of DNA: Some Theory and Experiments*, 2nd Annual DIMACS Meeting on DNA Based Computers, June 10-12, 1996 (Volume 44 in DIMACS). Laura F. Landweber and Eric B. Baum, editors. American Mathematical Society, 1998. pp. 191-213.

96) E. Winfree, T. Eng, and G. Rozenberg, String Tile Models for DNA Computing by Self-Assembly, In Proceedings of the Sixth Annual Meeting on DNA Based Computers, held at Leiden University, June 13-17, 2000. LNCS 2054, 63-88, Springer Verlag, 2001.

97) Erik Winfree. "Algorithmic Self-Assembly of DNA: Theoretical Motivations and 2D Assembly Experiments." Journal of Biomolecular Structure and Dynamics, Conversation 11, No. 2, May 2000. pp. 263-270.

98) H. Yan, X. Zhang, Z. Shen and N.C. Seeman, A robust DNA mechanical device controlled by hybridization topology, Nature, 415, 62-65, 2002.

99) Yurke, B., Turberfield, A. J., Mills, A. P.Jr, Simmel, F. C. and Neumann, J. L. A DNA-fuelled molecular machine made of DNA. Nature 406, 605-608 (2000).

100) Yurke, B., A.P. Mills, A.J. Turberfield , A Molecular Machine Made Of and Powdered By DNA, Biophys. J. 78, 2629 (2000).