

# DNA Nanotechnology and its Biological Applications<sup>1</sup>

Chapter **13** of Book:

*Bio-inspired and Nano-scale Integrated Computing* (Edited by Mary Eshaghian)

Publisher: Wiley, USA, (2007).

**John H Reif<sup>2</sup> and Thomas H LaBean<sup>2,3</sup>**

## 13.0 Summary

This Chapter overviews the emerging research area of DNA nanostructures and biomolecular devices. We discuss work involving the use of synthetic DNA to self-assemble DNA nanostructure devices. Recently, there have been a series of quite astonishing experimental results - which have taken the technology from a state of intriguing possibilities into demonstrated capabilities of quickly increasing scale. We particularly emphasize molecular devices that are programmable and autonomous. By programmable, we mean the tasks executed can be modified

---

<sup>1</sup> Supported by NSF grants CCF-0523555, CCF-0432038, CCF-0432047.

<sup>2</sup> Department of Computer Science, Duke University, Durham, NC 27708 USA

<sup>3</sup> Department of Chemistry, Duke University, Durham, NC 27708 USA

without entirely redesigning the nanostructure. By autonomous, we mean that the steps are executed with no exterior mediation after starting. We discuss various such programmable molecular-scale devices that achieve various capabilities, including: computation, 2D patterning, amplified sensing, and molecular transport.

## **13.1 Introduction**

### **13.1.1 DNA Nanotechnology and its use to Assemble Molecular-Scale Devices**

The particular molecular-scale devices that are the topic of this article are known as DNA nanostructures. As will be explained, DNA nanostructures have some unique advantages among nanostructures: they are relatively easy to design, fairly predictable in their geometric structures, and have been experimentally implemented in a growing number of labs around the world. They are constructed primarily of synthetic DNA. A key principle in the study of DNA nanostructures is the use of self-assembly processes to actuate the molecular assembly. Since self-assembly operates naturally at the molecular scale, it does not suffer from the limitation in scale reduction that so restricts lithography or other more conventional top-down manufacturing techniques. Other surveys of DNA nanotechnology and devices have been given by LaBean [1], Mao [2], Reif [3], and Seeman [4].

In attempting to understand these modern developments, it is worth recalling that mechanical methods for computation date back to the very onset of computer science, for example to the cog-based mechanical computing machine of Babbage. Lovelace stated in 1843 that Babbage's "Analytical Engine weaves algebraic patterns just as the Jacquard-loom weaves flowers and leaves". In some of the recently demonstrated methods for biomolecular computation described here, computational patterns were essentially woven into molecular fabric (DNA lattices) via carefully controlled and designed self-assembly processes.

In general, nanoscience research is highly interdisciplinary. In particular, DNA self-assembly uses techniques from multiple disciplines such as biochemistry, physics, chemistry, and material science, as well as computer science and mathematics. We will observe that many of these self-assembly processes are computational-based and programmable, and it seems likely that a variety of interdisciplinary techniques will be essential to the further development of this emerging field of biomolecular computation.

### **13.1.2 The Topics Discussed in this Article**

While a high degree of interdisciplinarity makes the topic quite intellectually exciting, it also makes it challenging for a typical reader. For this reason, this article was written with the expectation that the reader has little background knowledge of chemistry or biochemistry. We define a few relevant technical

terms in subsection 13.2.1. In subsection 13.2.2 we list some known enzymes used for manipulation of DNA nanostructures. In subsection 13.2.3 we list some reasons why DNA is uniquely suited for assembly of molecular-scale devices.

In many cases, the self-assembly processes are programmable in ways analogous to more conventional computational processes. We will overview theoretical principles and techniques (such as tiling assemblies and molecular transducers) developed for a number of DNA self-assembly processes that have their roots in computer science theory (e.g., abstract tiling models and finite state transducers). Computer-based design and simulation are also essential to the development of many complex DNA self-assembled nanostructures and systems. Error-correction techniques for correct assembly and repair of DNA self-assemblies are also discussed.

The area of DNA self-assembled nanostructures and robotics is by no means simply a theoretical topic - many dramatic experimental results have already been demonstrated, and a number of these will be discussed. The complexity of these demonstrations has been increasing at an impressive rate (even in comparison to the rate of improvement of silicon-based technologies). This article discusses the accelerating scale of complexity of DNA nanostructures (such as the number of addressable pixels of 2D patterned DNA nanostructures) and provides some predictions for the future.

Molecular-scale devices using DNA nanostructures have been engineered to have various capabilities, ranging from (i) execution of molecular-scale computation, (ii) use as scaffolds or templates for the further assembly of other materials (such as scaffolds for various hybrid molecular electronic architectures or perhaps high-efficiency solar-cells), (iii) robotic movement and molecular transport, and (iv) exquisitely sensitive molecular detection and amplification of single molecular events, and (v) transduction of molecular sensing to provide drug delivery.

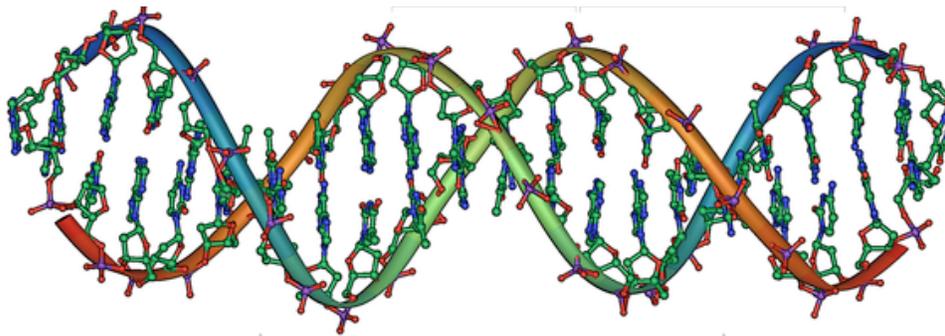
## **13.2 Introductory Definitions**

### **13.2.1 A Brief Introduction to DNA**

*Single stranded DNA* (denoted ssDNA) is a linear polymer consisting of a sequence of DNA bases oriented along a backbone with chemical directionality. By convention, the base sequence is listed starting from the 5-prime end of the polymer and ending at the 3-prime end (these names refer to particular carbon atoms in the deoxyribose sugar units of the sugar-phosphate backbone, the details of which are not critical to the present discussion). The consecutive nucleotide bases (monomer units) of an ssDNA molecule are joined through the backbone via covalent bonds. There are 4 types of *DNA* bases adenine, thymine, guanine and cytosine typically denoted by the symbols A, T, G, and C,

respectively. These bases form the alphabet of DNA; the specific base sequence comprises DNA's information content. The bases are grouped into *complementary pairs* (G, C) and (A, T).

The most basic DNA operation is *hybridization* where two ssDNA oriented in opposite directions can bind to form a double stranded *DNA helix* (dsDNA) by pairing between complementary bases. DNA hybridization occurs in a buffer solution with appropriate temperature, pH, and salinity. A dsDNA helix is illustrated in Figure 13.1.



**Figure 13.1: Structure of a *DNA double helix*** (Created by Michael Ströck and released under the GNU Free Documentation License(GFDL).)

Since the binding energy of the pair (G, C) is approximately half-again the binding energy of the pair (A, T), the association strength of hybridization depends on the sequence of complementary bases, and can be approximated by known software packages. The *melting temperature* of a DNA helix is the temperature at which half of all the molecules are fully hybridized as double helix, while the other half are single stranded. The kinetics of the DNA hybridization process is quite well understood; it occurs in a (random) zipper-like manner, similar to a biased one-dimensional random walk.

Whereas ssDNA is a relatively flexible molecule, dsDNA is quite stiff (over lengths of less than 150 or so bases) and has the well-characterized double helix structure. There are about 10.5 bases per full rotation on this helical axis. The exact geometry of the double helix depends slightly on the base sequence in a way readily computed by existing software. A *DNA nanostructure* is a multi-molecular (supramolecular) complex consisting of a number of ssDNA that have partially hybridized, as designed, along their sub-segments.

## 2.2 Manipulation of DNA

In addition to the hybridization reaction, there are a wide variety of known enzymes and other proteins used for manipulation of DNA nanostructures that have predictable effects. (Interestingly, these proteins were discovered in natural bacterial cells and tailored for laboratory use.) These include:

- *Restriction enzymes* can cut (double-strand break) or nick (single-strand break) a DNA backbone at specific locations determined by short base sequences.
- *Ligase enzymes* can heal or repair DNA nicks by forming covalent bonds in the sugar-phosphate backbone.
- *Polymerase* can extend an ssDNA by covalently coupling further

complementary bases as dictated by a template ssDNA, thus forming a longer sequence of dsDNA.

Besides their extensive use in other biotechnology procedures, the above reactions, together with hybridization, are often used to execute and control DNA computations and DNA molecular robotic operations. The restriction enzyme reactions are programmable in the sense that they are site specific, only executed as determined by the appropriate DNA base sequence. The latter two reactions, using ligase and polymerase, require the expenditure of energy via consumption of ATP molecules, and thus can be controlled by ATP concentration.

### **13.2.3 Why use DNA to Assemble Molecular-Scale Devices?**

There are many advantages of DNA as a material for building things at the molecular scale.

(a) From the perspective of design, the advantages are:

- The basic geometric and thermodynamic properties of dsDNA are well understood and can be modeled by available software systems. The structure of a large number of more complex DNA nanostructures can be predicted by a number of prototype software systems from details like the

sequence composition, temperature and buffer conditions (which are the key relevant parameters).

- Design of DNA nanostructures can be assisted by software. To design a DNA nanostructure or device, one needs to design a library of ssDNA strands with specific segments that hybridize to (and only to) specific complementary segments on other ssDNA. There are a number of software systems for this combinatorial sequence design task and for design of DNA nanostructures with desired structures.

(b) From the perspective of experiments, the advantages are:

- The chemical synthesis of ssDNA is now routine and inexpensive; a test tube of ssDNA consisting of any specified short sequence of bases (<150) can be obtained from commercial sources for modest cost (about half a US dollar per base at this time); it will contain a very large number (typically at least  $10^{12}$ ) identical ssDNA molecules. The synthesized ssDNA can have errors (premature termination of the synthesis is the most frequent error), but can be easily purified by well-known techniques (e.g., electrophoresis as mentioned below).
- The assembly of DNA nanostructures is a very simple experimental

process: in many cases, one simply combines the various component ssDNA into a single test tube with an appropriate buffer solution at an initial temperature above the expected melting temperature of the most stable base-pairing structure, and then slowly cools the test tube below the melting temperature.

- The assembled DNA nanostructures can be characterized by a variety of techniques. One such technique is electrophoresis. It provides information about the relative molecular mass of DNA molecules, as well as some information regarding their assembled structures, depending on what type of electrophoresis (denaturing or native, respectively) is used. Other techniques like Atomic Force Microscopy (AFM) and Transmission Electron Microscopy (TEM) provide images of the actual assembled DNA nanostructures on 2D surfaces.

### **13.3 Adelman's Initial Demonstration of a DNA-based Computation**

#### **13.3.1 Adleman's Experiment**

The field of DNA computing began in 1994 with a laboratory experiment described in [5 & 6]. The goal of the experiment was to find, within a given

directed graph, a Hamiltonian path, which is a path that visits each node exactly once. To solve this problem, a set of ssDNA was designed based on the set of edges of the graph. When combined in a test tube and cooled, they self-assembled into dsDNA. Each of these DNA nanostructures was a linear DNA helix that corresponded to a path in the graph. If the graph had a Hamiltonian path, then one of these DNA nanostructures encoded the Hamiltonian path. By conventional biochemical extraction methods, Adelman was able to isolate only DNA nanostructures encoding Hamiltonian paths, and by determining their sequence, the explicit Hamiltonian path. It should be mentioned that this landmark experiment was designed and experimentally demonstrated by Adleman alone, a computer scientist with limited training in biochemistry.

### **13.3.2 The Non-Scalability of Adelman's Experiment**

While this experiment founded the field of DNA computing, it was not scalable in practice, since the number of different DNA strands needed increased exponentially with the number of nodes of the graph. Although there can be an enormous number of DNA strands in a test tube ( $10^{15}$  or more, depending on solution concentration), the size of the largest graph that could be solved by his method was limited to at most a few dozen nodes. This is not surprising, since finding the Hamiltonian path is an NP complete problem, whose solution is likely to be intractable using conventional computers. Even though DNA computers operate at the molecular-scale, they are still equivalent to conventional

computers (e.g., deterministic Turing machines) in computational power. This experiment taught a healthy lesson to the DNA computing community (which is now well-recognized): to carefully examine scalability issues and to judge any proposed experimental methodology by its scalability.

### **13.3.3 Autonomous Biomolecular Computation**

Shortly following Adleman's experiment, there was a burst of further experiments in DNA computing, many of which were quite ingenious. However, almost none of these DNA computing methods were autonomous, and instead required many tedious laboratory steps to execute. In retrospect, one of the most notable aspects of Adleman's experiment was that the self-assembly phase of the experiment was completely autonomous - it required no exterior mediation (the bulk of the labor was in the non-autonomous molecular sorting steps). The strategy can be termed generate-and-sort, since all possible answers are created and incorrect solutions are subsequently discarded. Maximizing molecular autonomy makes an experimental laboratory demonstration much more feasible as the scale increases. The remaining article mostly discusses autonomous devices for bio-molecular computation based on self-assembly.

### **13.4 Self-Assembled DNA Tiles and Lattices**

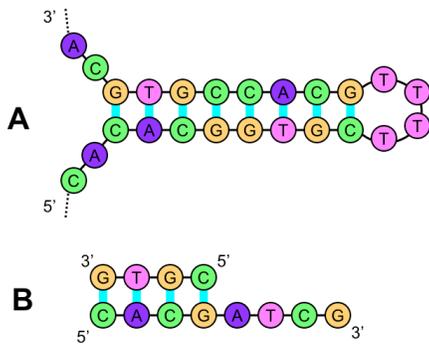
### **13.4.1 Computation By Self-Assembly**

The most fundamental way computer science ideas have impacted DNA nanostructure design is via the pioneering work by theoretical computer scientists on a formal model of 2D tiling due to Wang (in 1961), which culminated in a proof by Berger in 1966 that universal computation could be done via tiling assemblies. Winfree [7] was the first to apply the concepts of computational tiling assemblies to DNA molecular constructs. His core idea was to use tiles composed of DNA to perform computations during the process of self-assembly, where only valid solutions to the computation are allowed to assemble. To understand this idea, we will need an overview of DNA nanostructures, as presented in the next subsection 4.2.

### **13.4.2 DNA Nanostructures**

Recall that a DNA nanostructure is a multi-molecular complex consisting of a number of ssDNA that have partially hybridized along their sub-segments. The field of DNA nanostructures was pioneered by Seeman [4].

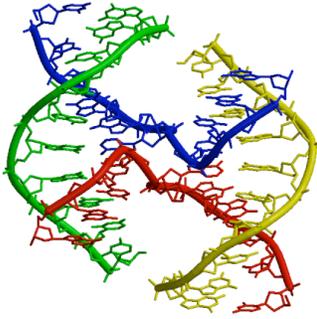
Particularly useful types of motifs often found in DNA nanostructures include Stem-Loops and Sticky Ends, as illustrated below.



**Figure 13.2: A Stem-Loop and a Sticky End.**

Figure 13.2A illustrates a *stem-loop*, where ssDNA loops back to hybridize on itself (that is, one segment of the ssDNA (near the 5' end) hybridizes with another segment further along (nearer the 3' end) on the same ssDNA strand). The shown stem consists of the dsDNA region with sequence CACGGTGC on the bottom strand. The loop in this case consists of the ssDNA region with sequence TTTT. Stem-loops are often used as markers for visualizing programmed patterning on DNA nanostructures.

Figure 13.2B illustrates a *sticky end*, where unhybridized sDNA protrudes from the end of a double helix. The sticky end shown (ATCG) protrudes from dsDNA (CACG on the bottom strand). Sticky ends are often used to combine two DNA nanostructures together via hybridization of their complementary ssDNA. The Figure 13.2B shows the antiparallel nature of dsDNA with the 5' end of each strand pointing toward the 3' end of its partner strand.



**Figure 13.3: A Holliday Junction** (Image created by Miguel Ortiz-Lombardía, CNIO, Madrid, Spain and used with permission.)

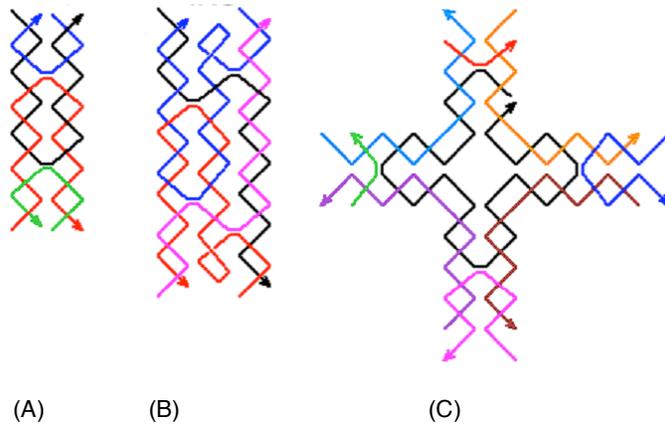
Figure 13.3 illustrates a *Holliday junction*, where two adjacent DNA helices form a junction with one strand of each DNA helix (blue and red) crossing over to the other DNA helix. Holliday junctions are often used to tie together various parts of a DNA nanostructure.

#### 13.4.2 DNA Tiles and Lattices

A *DNA tile* is a DNA nanostructure that has a number of sticky ends on its sides, which are termed *pads*. A DNA lattice is a DNA nanostructure composed of a group of DNA tiles that are assembled together via hybridization of their pads. Generally the strands composing the DNA tiles are designed to have a melting temperature above those of the pads, ensuring that when the component DNA molecules are combined together in solution, first the DNA tiles assemble, and only then, as the solution is further cooled, do the tiles bind together via hybridization of their pads. A number of prototype computer software systems

have been developed for the design of the DNA sequences composing DNA tiles, and for optimizing their stability.

To program a tiling assembly, the pads of tiles are designed so that tiles assemble together as intended. Proper designs ensure that only the adjacent pads of neighboring tiles are complementary, so only those pads hybridize together. Figure 13.4 illustrates some principal DNA tiles.



**Figure 13.4: DNA Tiles**

Winfrey and Seeman [7] developed a family of DNA tiles known collectively as DX tiles (see left tile illustrated in Figure 13.4A) that consisted of two parallel DNA helices linked by two immobile Holliday junctions. They demonstrated that these tiles formed large 2D lattices, as viewed by AFM. Subsequently, other DNA tiles were developed to provide for more complex strand topology and interconnections, including a family of DNA tiles known as *TX tiles* (see tile

illustrated in Figure 13.4B) [8] composed of three DNA helices linked by four crossover junctions. Both the DX tiles and the TX tiles are rectangular in shape, where two opposing edges of the tile have pads consisting of ssDNA sticky ends. In addition, TX tiles have topological properties that allow for strands to propagate in useful ways through tile lattices (this property is often used for aid in patterning DNA lattices as described below). (c) Other DNA tiles known as *Cross tiles* (see Figure 13.4C) [8] are shaped roughly square, and have pads on all four sides, allowing for binding of the tile directly with neighbors in all four directions in the lattice plane. Figure 13.5 gives an AFM image of 2D Lattice using Cross tiles.

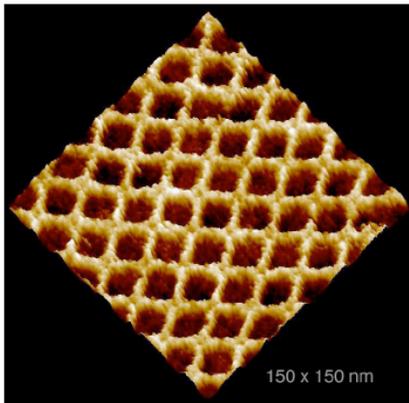
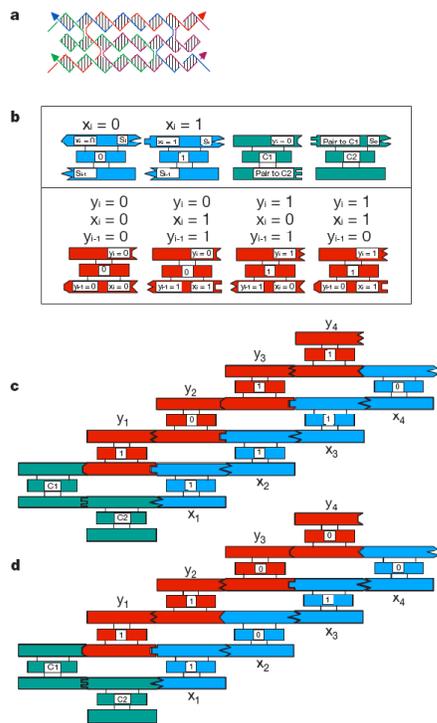


Figure 13.5: AFM image of 2D Lattice of Cross tiles

## 13.5 Autonomous Finite State Computation Using Linear DNA Nanostructures

### 13.5.1 The First Experimental Demonstration of Autonomous Computations using Self-Assembly of DNA Nanostructures

The first experimental demonstrations of computation using DNA tile assembly was [9], which demonstrated 2-layer, linear assemblies of TX tiles that executed a bit-wise cumulative XOR computation. (Given  $n$  bits as input, each  $i^{\text{th}}$  output is the XOR of the first  $i$  input bits, which is the computation occurring when one determines the output bits of a full-carry binary adder circuit.) The experiment [9] is described further in Figure 13.6.



**Figure 13.6: Sequential Boolean Computation via a Linear DNA Tiling Assembly** (adapted with permission from [9].)

Figure 13.6 shows a unit TX tile (a) and the sets of input and output tiles (b) with geometric shapes conveying sticky-end complementary matching. The (blue) input layer and (green) corner condition tiles were designed to assemble first (see example computational assemblies in Figure 13.6 (c) & (d)). The (red)

output layer would then assemble specifically starting from the bottom left using the inputs from the blue layer. The tiles were designed such that an output reporter strand ran through all the  $n$  tiles of the assembly by bridges across the adjoining pads in input, corner, and output tiles. This reporter strand was pasted together from the short ssDNA sequences within the tiles using ligation enzyme mentioned previously. When the solution was warmed, this output strand was isolated and identified. The output data was read by experimentally determining the sequence of cut sites (as described below). In principle, the purified output strands could be used for subsequent computations.

This experiment [9] provided answers to a basic question:

- *How can one provide data input to a molecular computation using DNA tiles?*

In this experiment the input bits (1's & 0's) were encoded on two different tile types with specific sticky-ends and specific endonuclease cleavage sites (subsequences at which protein enzymes can cut the DNA backbone). The input sequence was defined by specific sticky-ends that assembled a specific input layer (blue layers in Figure 13.6).

The next question of concern is:

- *How can one execute a step of computation using DNA tiles?*

To execute steps of computation, the TX tiles were designed to have pads at one end that encoded the cumulative XOR value. Also, since the reporter strand

segments ran through each tile, the appropriate input bit was also provided within its structure. These two values implied the opposing pad on the other side of the tile be the XOR of these two bits.

The final question of concern is:

- *How can one determine and/or display the output values of a DNA tiling computation?*

The output in this case was read by determining which of two possible cut sites (endonuclease cleavage sites) were present at each position in the tile assembly. This was executed by first isolating the ligated reporter strand, then digesting separate aliquots with each endonuclease separately and the two together, finally these samples were examined by gel electrophoresis and the output values were displayed as banding patterns on the gel.

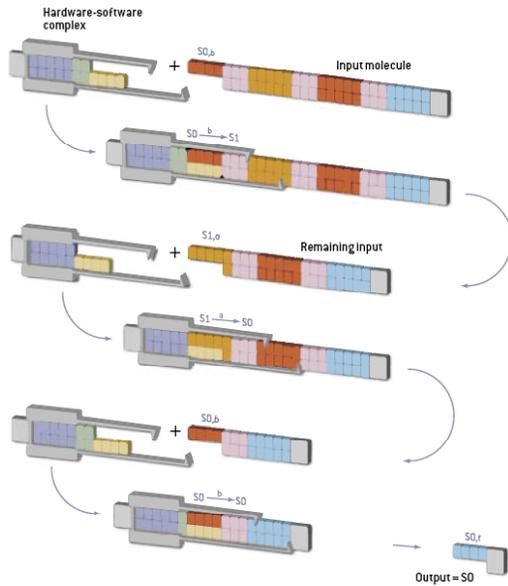
Another method for output (presented below) is the use of AFM observable patterning. Such patterning can be made by designing the tiles computing a bit 1 to have a stem loop protruding from the top of the tile or by providing a site for binding of a marker protein. Sequences of such molecular patterning is clearly viewable under appropriate AFM imaging conditions.

Although they are quite simple computations, the experiments of [9] and [10] did demonstrate pioneering methods for autonomous execution of a sequence of finite-state operations via algorithmic self-assembly, as well as for providing inputs and for outputting the results. Further DNA tile assembly computations [11

& 12] will be presented below in Figure 13.11.

### **13.5.2 Autonomous Finite-State Computations via Disassembly of DNA Nanostructures**

An alternative method for autonomous execution of a sequence of finite-state transitions was subsequently developed by [13]. Their technique essentially operated in the reverse of the assembly methods described above, and instead was based on disassembly. They began with a linear DNA nanostructure whose sequence encoded the inputs, and then they executed series of steps that digested the DNA nanostructure from one end. On each step, a sticky end at one end of the nanostructure encoded the current state, and the finite transition was determined by hybridization of the current sticky end with a small “rule” nanostructure encoding the finite-state transition rule. Then a restriction enzyme, which recognized the sequence encoding the current input as well as the current state, cut the appended end of the linear DNA nanostructure, to expose a new a sticky end encoding the next state. The hardware-software complex is composed of dsDNA with an ssDNA overhang (shown at top left ready to bind with the input molecule) and a protein restriction enzyme (shown as gray pinchers). See Figure 13.7 for further details. This ingenious design is an excellent demonstration that there is often more than one way to do any task at the molecular scale.



**Figure 13.7: Autonomous Finite-State Computations via Disassembly of DNA Nanostructures.** (Figure adapted with permission from [13]).

### 13.5.3 Applications of Autonomous Finite-State Computations at the Molecular Scale

Even very simple operations, such as the above Boolean or finite-state transitions, operating at the molecular-scale, could have important potential applications, for example, for drug mediation [13]. The idea is for the DNA nanostructures to take as input a set of RNA sequences, whose level of expression (or lack of expression) within a cell indicates a particular disease state. Then the execution of simple Boolean operations executable by finite-state transitions can determine that a disease exists, and execute a response (e.g., the release of RNA sequences which provide a remediation of the disease by altering the expression of proteins expressed by the cell). While such a scheme was demonstrated by [13] in the test tube, it remains to be demonstrated in the much

more challenging environment of a cell. Another class of applications is for control of molecular robotic devices, such as described in Section 13.7. **13.6**

### **Assembling Patterned and Addressable 2D DNA Lattices**

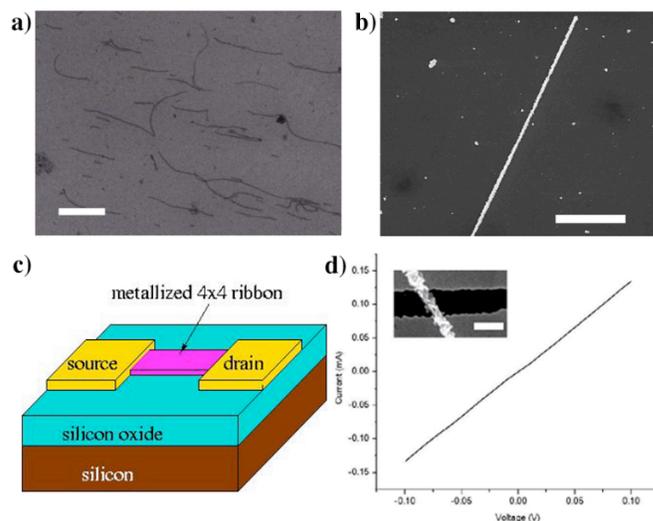
One of the most appealing applications of tiling computations is their use to form patterned nanostructures to which other, perhaps functional, materials can be selectively bound.

An *addressable* 2D DNA lattice is one that has a number of sites with distinct ssDNA. This provides a superstructure for selectively attaching other molecules at addressable locations. The input layer for the computational assembly described in Figure 13.6 is an example of an addressable system, since unique ssDNA pads defined the tile locations. Other examples will be presented below.

As discussed below, there are many types of molecules for which we can attach DNA. Known attachment chemistry allows them to be tagged with a given sequence of ssDNA. Each of these DNA-tagged molecules can then be assembled by hybridization of their DNA tags to a complementary sequence of ssDNA located within an addressable 2D DNA lattice. In this way, we can program the assembly of each DNA-tagged molecule onto a particular site of the addressable 2D DNA lattice.

### 13.6.1 Attaching Materials to DNA

There are many materials that can be made to directly or indirectly bind to specific segments of DNA using a variety of known attachment chemistries. Materials that can directly bind to specific segments of DNA include other (complementary) DNA, RNA, proteins, peptides, and various other materials. Materials that can be made to indirectly bind to DNA include a variety of metals (e.g., gold) that bind to sulfur-labeled compounds, carbon nanotubes (via various attachment chemistries), etc. These attachment strategies provide molecular-scale "Velcro" for attaching heterogeneous materials to DNA nanostructures. For example, they can potentially be used for attaching molecular electronic devices to 2D or 3D DNA nanostructures. See Figure 13.8 [14] for an example of conductive wires composed of self-assembled DNA tubes covered with gold or silver.



**Figure 13.8: DNA Templated Fabrication of Metal Nanowires**

Figure 13.8a is a SEM image of bare self-assembled DNA nanotube on silicon oxide surface (scale bar equals 500 nm). Figure 13.8b is a SEM image of gold coated DNA nanotube on silicon oxide surface (scale bar equals 500 nm). Figure 13.8c is a schematic representation of the measured device (inset of Figure 13.8d) showing the templated nanowire and source and drain electrodes fabricated by electron beam lithography. Figure 13.8d gives the current/voltage curve of a gold nanowire. Smaller, smoother silver nanowires are presented in [14 & 15].

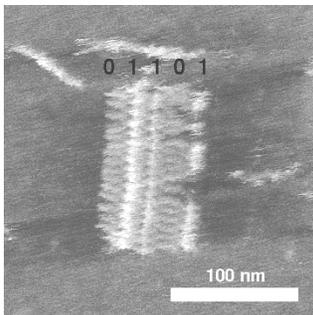
### **13.6.2 Methods for Programmable Assembly of Patterned 2D DNA Lattices**

The first experimental demonstration of 2D DNA lattices by Winfree and Seeman provided very simple patterning by repeated stripes determined by a stem loop projecting from every DNA tile on an odd column. This limited sort of patterning needed to be extended to large classes of patterns.

In particular, the key capability needed is a programmable method for forming distinct patterns on 2D DNA lattices, without having to completely redesign the lattice to achieve any given pattern. There are at least three methods [16] for assembling patterned 2D DNA lattices that now have been experimentally demonstrated, as described in the next subsections.

### 13.6.3 Use of Scaffold Strands for Programmable Assembly of Patterned 2D DNA Lattices

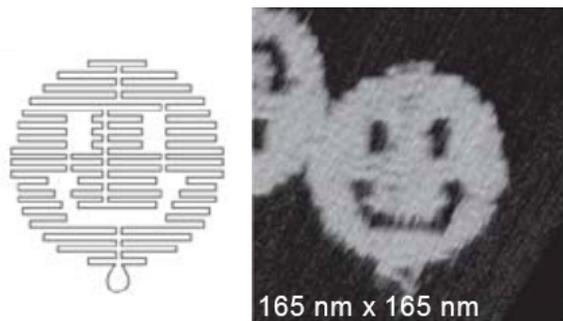
The first published use of a scaffold strand – a long ssDNA around which shorter ssDNA assemble to form structures larger than individual tiles is given in [1]. Scaffold strands were used to demonstrate programmable patterning of 2D DNA lattices in [17] by propagating 1D information from the scaffold into a second dimension to create AFM observable patterns. The scaffold strand weaves through the resulting DNA lattice to form a desired and distinct sequence of 2D barcode patterns (Figure 13.9).



**Figure 13.9: Use of Scaffold Strands for Programmable Assembly of Barcode Patterned 2D DNA Lattices**

In this demonstration, identical scaffold strands ran through each row of the 2D lattices, using short stem loops extending above the lattice to form pixels. This determined a bar code sequence of stripes over the 2D lattice that was viewed by AFM. In principle, this method may be extended to allow for each row's patterning to be determined by a distinct scaffold strand, defining an arbitrary 2D pixel image. A spectacular experimental demonstration of patterning via scaffold

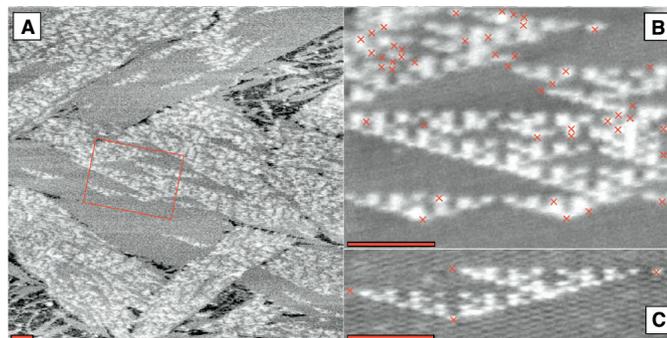
strand is also known as DNA origami [18]. This approach makes use of a long strand of scaffold ssDNA (such as from the sequence of a viral phage) that has only weak secondary structure and no long repeated or complementary subsequences. To this is added a large number of relatively short “staple” ssDNA sequences, with subsequences complementary to certain subsequences of the scaffold ssDNA. These staple sequences are chosen so that they bind to the scaffold ssDNA by hybridization, and induce the scaffold ssDNA to fold together into a DNA nanostructure. A schematic trace of the scaffold strand is shown in Figure 13.10 (left panel) and an AFM image of the resulting assembled origami is shown in Figure 13.10 (right panel). This landmark work of Rothemund [18] very substantially increases the scale of 2D patterned assemblies to hundreds of molecular pixels (that is, stem loops viewable via AFM) within square area less than 100 nanometers on a side. In principal this “molecular origami” method with staple strands can be used to form arbitrary complex 2D patterned nanostructures as defined.



**Figure 13.10: Use of DNA origami for Programmable Assembly of Patterned 2D DNA Lattices**

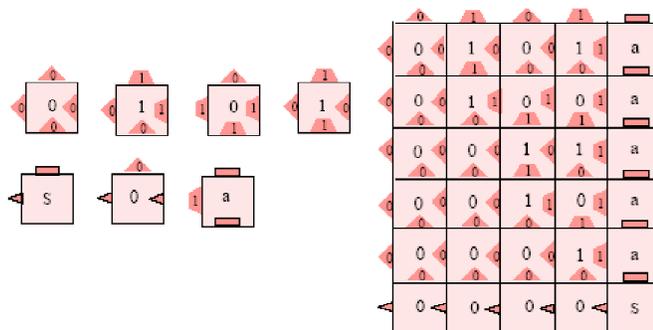
### 13.6.4 Use of Computational Assembly for Programmable Assembly of Patterned 2D DNA Lattices

Another very promising method is to use the DNA tile's pads to program a 2D computational assembly. Recall that computer scientists have in the 1970's shown that any computable 2D pattern can be so assembled. [11] and [12] have experimentally demonstrated two distinct and quite interesting 2D computational assemblies, and furthermore provided AFM images of the resulting nanostructures as illustrated in Figure 13.11.



**Figure 13.11: A modulo-2 version of Pascal's Triangle (known as the Sierpinski Triangle) [11].** (Figure adapted with permission from [11].)

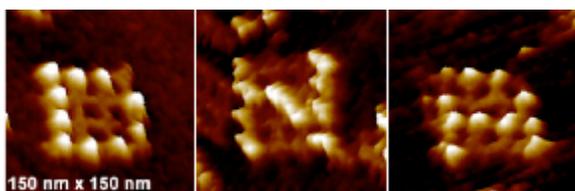
In Figure 13.11, each tile determines and outputs to neighborhood pads the XOR of two of the tile pads. Example AFM images of the assembled structures are shown in the three panels (scale bars = 100 nm).



**Figure 13.12: A modulo-2 version of Pascal's Triangle (known as the Sierpinski Triangle) [11].**

Figure 13.12 gives Winfree's design for a self-assembled binary counter [12], starting with 0 at the first row, and on each further row being the increment by 1 of the row below. The pads of the tiles of each row of this computational lattice were designed in a similar way to that of the linear XOR lattice assemblies described in the prior section. The resulting 2D counting lattice is found in MUX designs for address memory, and so this patterning may have major applications for patterning molecular electronic circuits.

### 13.6.5 Use of Hierarchical Assembly for Programming of Patterned 2D DNA Lattices



**Figure 13.13: 2D Patterns By Hierarchical Assembly.** (Figure adapted with permission

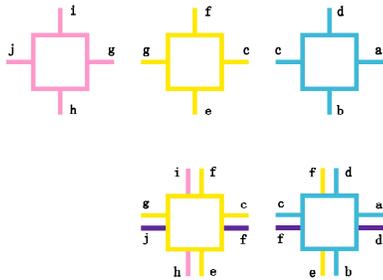
from [19].)

A further approach is to assemble DNA lattices in a hierarchical fashion [19]. Figure 13.13 gives three examples of preprogrammed patterns displayed on addressable DNA tile lattices. Tiles are assembled prior to mixing with other preformed tiles. Unique ssDNA pads direct tiles to designed locations. White pixels are “turned on” by binding a protein (streptavidin) at programmed sites as determined in the tile assembly step by the presence or absence of a small molecule (biotin) appended to a DNA strand within the tile. Addressable, hierarchical assembly has been demonstrated for only modest size lattices to date, but has considerable potential particularly in conjunction with the above methods for patterned assembly.

### **13.7 Error correction and Self-repair at the Molecular-Scale**

In many of the self-assembled devices described here, there can be significant levels of error. These errors occur both in the synthesis of the component DNA, and in the basic molecular processes that are used to assemble and modify the DNA nanostructures, such as hybridization and the application of enzymes. There are various purification and optimization procedures developed in biochemistry for minimization of many of these types of errors. However, there remains a need for development of methods for decreasing the errors of assembly and for self-repair of DNA tiling lattices comprised of a large number of

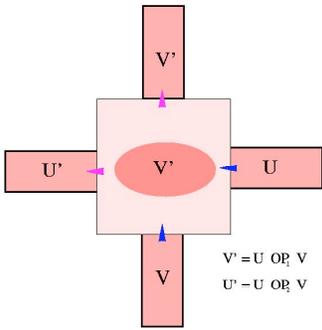
tiles. A number of techniques have been proposed for decreasing the errors of a DNA tiling assembly, by providing increased redundancy.



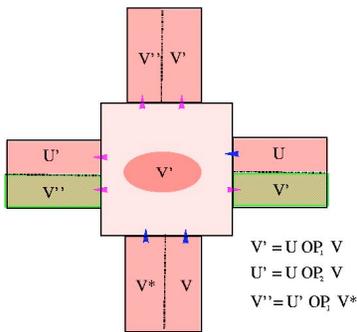
**Figure 13.14: Winfree’s Error resilient tiles**

Winfree [20] developed a “proofreading” method of replaced each tile with a subarray of tiles that provide sufficient redundancy to quadratically reduce errors, but increase the size of the assembly. This scheme is given in Figure 13.14 (top), with the original tiles in Figure 13.15(top) and the modified tiles in Figure 13.14 (bottom).

Reif et al [21] provides a more compact method for decreasing assembly errors. This compact method, takes each original tile (as illustrated in Figure13.15) and modifies the pads of each tile, to form a modified error-resultant tile (as illustrated in Figure13.15B).

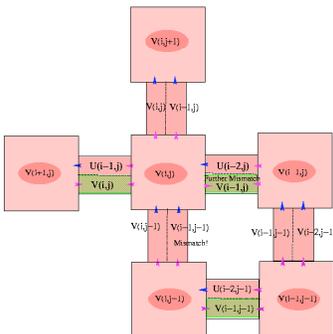


**Figure 13.15: Reif's Compact Error-Correction Scheme: An original tile**



**Figure 13.16: Reif's Compact Error-Correction Scheme: The modified tile**

The result is that essentially each tile both executes the original computation required at that location, as well as the computation of a particular neighbor. An illustration of the error propagation process is illustrated in Figure 13.17.



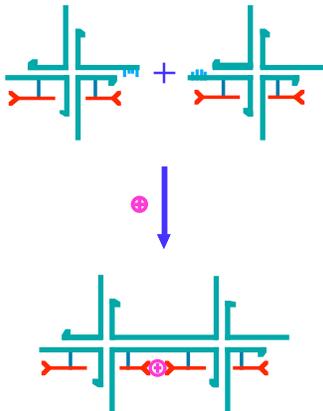
**Figure 13.17: Reif's Compact Error-Correction Scheme: The error propagation process**

This provides a quadratic reduction of errors without increasing the assembly

size. The experimental testing of these and related error-reduction methods is ongoing. It seems possible that other error-correction techniques developed in computer science may also be utilized.

### 13.8 Three-Dimensional DNA Lattices

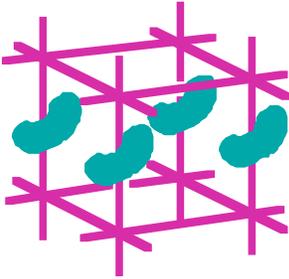
Most of the DNA lattices described in this article have been limited to 2D sheets. It appears to be much more challenging to assemble 3D DNA lattices of high regularity. There are some important applications if this can be done, as described in Figure 13.18 and 13.19.



**Figure 13.18: Scaffolding of 3D nanoelectronic architectures.** (The figure showing DNA (cyan) and protein (red) organizing functional electronic structures was adapted with permission from [22].)

Figure 13.18 illustrates the application of three-dimensional DNA lattices to scaffolding of 3D nanoelectronic architectures. The density of conventional

nanoelectronics is limited by lithographic techniques to only a small number of layers. The assembly of even quite simple 3D nanoelectronic devices such as memory would provide much improvement in density.



**Figure 13.19: Application of Three Dimensional DNA Lattices to Scaffolding of proteins into regular 3D arrays.**

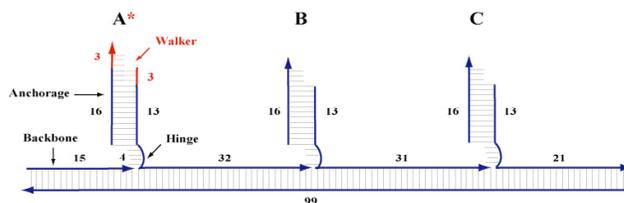
Figure 13.19 illustrates the application of three-dimensional DNA lattices to scaffolding of proteins into regular 3D arrays. It has been estimated that at least one half of all natural proteins can't be readily crystallized, and have unknown structure, and determining these structures would have a major impact in the biological sciences. Suppose a 3D DNA lattice can be assembled with sufficient regularity and with regular interstices (say within each DNA tile comprising the lattice). Then a given protein might be captured within each of the lattice's interstices, allowing it to be in a fixed orientation at each of its regularly spaced locations in 3D. This would allow the protein to be arranged in 3D in a regular way to allow for X-ray crystallography studies of its structure. This visionary idea

is due to Seeman. So far there has been only limited success in assembling 3D DNA lattices, and they do not yet have the degree of regularity (down to 2 or 3 Angstroms) required for the envisioned X-ray crystallography studies. However, given the successes up to now for 2D DNA lattices, this seems eventually achievable.

### 13.9 Autonomous Molecular Transport Devices Self-Assembled from DNA

There are a number of other tasks that can be done at the molecular scale that would be considerably aided by this technology. For example, many molecular-scale tasks may require the transport of molecules. The cell uses protein motors fueled by ATP to do this. While a number of motors composed of DNA nanostructures have been demonstrated, they did not operate autonomously, and instead require some sort of externally mediated changes (such as temperature-cycling) on each work-cycle of the motor.

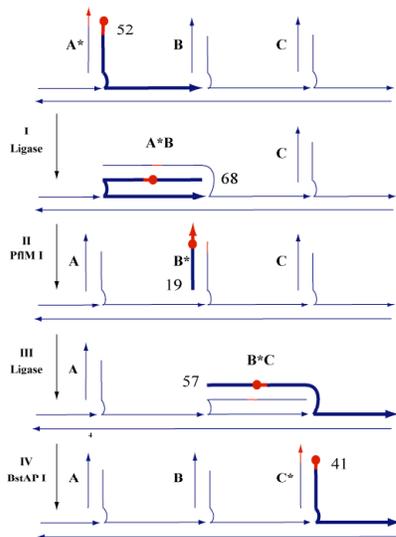
Peng et al [23] experimentally demonstrated the first autonomously operating device composed of DNA providing transport as described in Figures 13.20 and 13.21.



**Figure 13.20: Design of Autonomous Molecular Transport Devices Self-Assembled from**

## DNA

First a linear DNA nanostructure (the “road”) with a series of attached ssDNA strands (the “steps”) was self-assembled, as illustrated in Figure 13.20. A fixed-length segment of DNA helix (the “walker”) with short sticky ends (it’s feet”) assembled on one end of the road, with the feet of the walker hybridized to the first two steps of the road.



**Figure 13.21: Operation of Autonomous Molecular Transport Devices Self-Assembled from DNA**

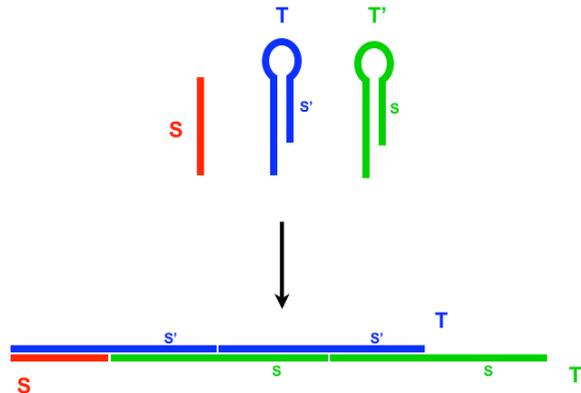
Then, as illustrated in Figure 13.21, the walker proceeded to make sequential movement along the road, where at the start of each step, the feet of the walker are hybridized to two consecutive steps of the road. Then a restriction enzyme cuts the DNA helix where the backward foot is attached, exposing a new sticky end forming a new replacement foot that can hybridize to the next step that is free, which can be the step just after the step where the other foot is currently

attached. A somewhat complex combinatorial design for the sequences composing the steps and the walker ensures that there is unidirectional motion forward along the road.

### **13.10 Autonomous Molecular Cascade Devices for Molecular Sensing**

Another type of task that can be done at the molecular scale that would be considerably aided by this technology is to sense a particular molecule and amplify a response signal to achieve detection with extremely few starting target molecules. There are a number of protocols such as PCR used to detect and amplify a given sequence of DNA, but most of these require a repeated temperature cycling and so are not autonomous. [24] demonstrated an autonomous system using DNA nanostructures that initiated a hybridization cascade reaction in response to detection of a given ssDNA sequence S. It is described in Figure 13.22.

The experiment made use of multiple copies of two distinct DNA nanostructures T and T' that are initially added to a test tube.



**Figure 13.22: Autonomous Molecular Cascade for Signal Amplification**

When ssDNA sequence S is added to the test tube, S initially has a hybridization reaction with a part of T, thus exposing a second ssDNA S' that had been previously hidden within the nanostructure of T.

Next, S' has a hybridization reaction with a part of T', thus exposing a second copy of S that had been previously hidden within the nanostructure of T'. That other copy of S then repeats the process of other similar (but so far unaltered) copies of T and T', allowing a cascade effect to occur completely autonomously. Such autonomous molecular cascade devices have applications to a variety of medical applications, where a larger response (e.g., a cascade response) is required in response to one of multiple molecular detection events.

### 13.11 Conclusions

We have overviewed a number of methods for assembling computational

patterns within the molecular fabric of DNA lattices. We have surveyed the varied interdisciplinary techniques for carefully designing and controlling these self-assembly processes. Many of these self-assembly processes are computational-based and programmable and it seems likely that interdisciplinary techniques will be essential to other emerging subfields of nanoscience and biomolecular computation. We have also discussed a number of key challenges still confronting this emerging field on DNA nanostructures, including the need for error-correction and the challenge and applications of constructing three-dimensional DNA lattices.

**Thanks and Credits.** Thanks to N. Gopalkrishnan, U. Majumder, and S. Sahu for their very useful comments on this article.

## References

[1] LaBean, T.H., Winfree, E., and Reif, J.H. Experimental Progress in Computation by Self-Assembly of DNA Tilings. in *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Volume 54, pp 123-140, Editors: E. Winfree and D.K. Gifford, Proceedings of the 5th DIMACS Workshop on DNA Based Computers, MIT, Cambridge, (1999).

[2] Zhaoxiang Deng, Yi Chen, Ye Tian, Chengde Mao, A fresh look at DNA

nanotechnology, chapter in *Nanotechnology: Science and Computation* (eds. J. Chen; N. Jonoska & G. Rozenberg), Springer, pp 23-34, (2006).

[3] John H. Reif and Thomas H. LaBean, Autonomous Programmable Biomolecular Devices Using Self-Assembled DNA Nanostructures, invited paper, Published in Communications of the ACM (CACM), Special Section entitled “New Computing Paradigms (edited by Toshinori Munakata), to appear 2007.

[4] Nadrian C. Seeman, Nanotechnology and the Double Helix; *Scientific American*, 290 (6), 64-75 (June 2004).

[5] Adleman L. Molecular computation of solutions to computational problem. *Science* **266**,1021–1024 (1994).

[6] Leonard Adleman, Computing with DNA, *Scientific American*, 279(2), p 34-41, (August 1998).

[7] Erik Winfree, Xiaoping Yang, Nadrian C. Seeman, Universal Computation via Self-assembly of DNA: Some Theory and Experiments, in *DNA Based Computers II*, pgs 191-213, 1998.

[8] T. H. LaBean, 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 Chemistry Society (JACS)* 122, 1848-1860 (2000).

[9] C. Mao, LaBean, T.H. Reif, J.H., Seeman, Logical Computation Using Algorithmic Self-Assembly of DNA Triple-Crossover Molecules, *Nature*, vol. 407, pp. 493–495. (Sept. 28 2000).

[10] Hao Yan, Liping Feng, Thomas H. LaBean, and John Reif, DNA Nanotubes, Parallel Molecular Computations of Pairwise Exclusive-Or (XOR) Using DNA "String Tile" Self-Assembly in *Journal of American Chemistry Society(JACS)*, Vol. 125, No. 47, pp. 14246-14247, 2003.

[11] Paul W.K. Rothmund, Nick Papadakis, Erik Winfree, Algorithmic Self-Assembly of DNA Sierpinski Triangles, *PLoS Biology* 2 (12), (Dec., 2004).

[12] Robert D. Barish, Paul W. K. Rothmund, and Erik Winfree, Two Computational Primitives for Algorithmic Self-Assembly: Copying and Counting, *NanoLetters* Vol. 5, No. 12 2586-2592, (2005).

[13] Ehud Shapiro and Yaakov Benenson, Bringing DNA Computers to Life. Scientific American, 45-51 (May 2006).

[14] Hao Yan, Sung Ha Park, Gleb Finkelstein, John H. Reif, and Thomas H. LaBean, DNA-Templated Self-Assembly of Protein Arrays and Highly Conductive Nanowires, Science, Vol. 301, pp. 1882-1884, (Sep 26 2003).

[15] S-H. Park, M.W. Prior, T.H. LaBean, and G. Finkelstein (2006) Optimized fabrication and electrical analysis of silver nanowires templated on DNA molecules. Applied Physics Letters, vol. 89, 033901.

[16] Thomas H. LaBean, Kurt V. Gothelf, and John H Reif, Self-assembling DNA Nanostructures for Patterned Molecular Assembly, invited chapter in textbook Nanobiotechnology, (edited by Chad A. Mirkin and Christof M. Niemeyer), John Wiley & Sons Pub., (2006).

[17] Hao Yan, Thomas H. LaBean, Liping Feng, and John H. Reif, Directed Nucleation Assembly of Barcode Patterned DNA Lattices, PNAS, Volume 100, No. 14, pp. 8103-8108, July 8, (2003).

[18] Paul W. K. Rothmund, Folding DNA to create nanoscale shapes and patterns, Nature 440, 297-302 (16 March 2006).

[19] Sung Ha Park, Constantin Pistol, Sang Jung Ahn, John H. Reif, Alvin R. Lebeck, Chris Dwyer, and Thomas H. LaBean, Finite-Size, Fully Addressable DNA Tile Lattices Formed by Hierarchical Assembly Procedures, *Angewandte Chemie [International Edition]*, pp. 735-739, Volume 45, Issue 5, Jan. 23, 2006.

[21] John H. Reif, Sudheer Sahu, Peng Yin, Compact Error-Resilient Computational DNA Tiling Assemblies, in "Nanotechnology: Science and Computation", Springer Verlag series in Natural Computing (edited by Natasha Jonoska), Springer-Verlag Berlin, Germany, pages 79-104, (2006).

[20] Erik Winfree and Renat Bekbolatov, Proofreading Tile Sets: Error Correction for Algorithmic Self-Assembly, in *DNA Computers 9, LNCS, (2943)*, pp. 126-144, (2004).

[22] Robinson, B.H. & Seeman, N.C. (1987), *Protein Eng.* 1, 295-300.

[23] Peng Yin, Hao Yan, Xiaojun G. Daniel, Andrew J. Turberfield, John H. Reif, A Unidirectional DNA Walker Moving Autonomously Along a Linear Track, *Angewandte Chemie [International Edition]*, Volume 43, Number 37, pp 4906-4911, (Sept. 20, 2004).

[24] Robert M. Dirks, and Pierce, Niles A. Triggered amplification by hybridization chain reaction. PNAS, 101 (43). pp. 15275-15278, (2004).