

Engineering Natural Computation by Autonomous DNA-Based Biomolecular Devices

John H Reif¹ and Thomas H LaBean^{1,2,3}

Abstract

This Chapter overviews the past and current state of a selected part of the emerging research area of the field of DNA-based biomolecular devices. We particularly emphasize molecular devices that are:

- *Autonomous*: executing steps with no exterior mediation after starting, and
- *Programmable*: the tasks executed can be modified without entirely redesigning the nanostructure.

We discuss work in this area that makes use of synthetic DNA to self-assemble into DNA nanostructure devices. Recently, there have been a series of impressive experimental results - which have taken the technology from a state of intriguing possibilities into demonstrated capabilities of quickly increasing scale. We discuss various such programmable molecular-scale devices that achieve:

- computation,
- 2D patterning,
- amplified sensing, and
- molecular or nano-scale transport.

This article is written for a general audience, and particularly emphasizes the interdisciplinary aspects of this quickly evolving and exciting field.

¹ Department of Computer Science, Duke University, Durham, NC 27708 USA

² Department of Chemistry, Duke University, Durham, NC 27708 USA

² Department of Biomedical Engineering, Duke University, Durham, NC 27708 USA

1 Introduction

1.1 Autonomous Programmable Molecular Devices

This chapter introduces the reader to a rapidly evolving and very exciting topic of Nanoscience, namely self-assembled DNA nanostructures, and their use for molecular computation and molecular transport. To tighten the focus, and to emphasize techniques that seem most promising, the discussion will be limited to processes that are: *Autonomous* (executing steps with no exterior mediation after starting), and *Programmable* (the tasks executed can be modified without entirely redesigning the nanostructure).

1.2 Questions about Biomolecular Devices

We will see that DNA self-assembly processes can be made computational-based and programmable, and it seems likely that computer science techniques will be essential to the further development of this emerging field of biomolecular computation. This chapter particularly illustrates the way in which computer science techniques and methods have impact on this emerging field. Some of the key questions one might ask are given below.

- What is the theoretical basis for these devices?
- How will such devices be designed and implemented?
- How can we simulate them prior to manufacture?
- How can we optimize their performance?
- How will such devices be manufactured?
- How much do the devices cost?
- How scalable is the device design?
- How will efficient I/O be accomplished?
- How will they be programmed?
- What efficient algorithms can be programmed?
- What will be their applications? What can they control?
- How can we correct for errors or repair them?

Note that these questions are exactly the sort of questions that computer scientists routinely ask about conventional computing devices. The discipline of computer science has developed a wide variety of techniques to address such basic questions, and we will later point out some which have an important impact to molecular-scale devices.

1.3 Why Bottom-Up Self-Assembly ?

Construction of molecular scale structures and devices is one of the key challenges facing science and technology in the twenty-first century. This challenge is at the core of an emerging discipline of Nanoscience. A key deficiency is the need for robust, error-free

methods for self-assembly of complex devices out of large numbers of molecular components. This key challenge requires novel approaches.

For example, the macro-electronics industry is now reaching the limit of miniaturization possible by top-down lithographic fabrication techniques. New bottom-up methods are needed for self-assembling complex, aperiodic structures for nanofabrication of molecular electronic circuits that are significantly smaller than conventional electronics.

1.4 Why use DNA for Assembly of Biomolecular Devices ?

The particular molecular-scale devices that are the topic of this chapter 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. Conventional electronics fabrication is reaching the limit of miniaturization possible by top-down techniques. 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.

In attempting to understand the modern development of DNA self-assembly, it is interesting to recall 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.

1.5 The Dual Role of Theory and Experimental Practice

In many cases, 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).

However, the area of DNA self-assembled nanostructures and molecular robotics is by no means simply a theoretical topic - many dramatic experimental demonstrations have already been made and a number of these will be discussed. The complexity of these demonstrations has increased at an impressive rate (even in comparison to the rate of

improvement of silicon-based technologies).

1.6 Applications of Autonomous Programmable Molecular Devices

This chapter 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 (akin to artificial, programmable versions of cellular transport mechanisms) (iv) exquisitely sensitive molecular detection and amplification of single molecular events, and (v) transduction of molecular sensing to provide drug delivery. Error-correction techniques for correct assembly and repair of DNA self-assemblies have also been recently developed. Computer based design and simulation are also essential to the development of many complex DNA self-assembled nanostructures and systems.

1.7 The Operation of Programmable Molecular Devices

While we will describe a wide variety of methods for executing molecular computation, these methods can be partitioned into two basic classes:

- *Distributed parallel molecular computations* execute a computation for which they require multiple distinct molecules that interact to execute steps of each computation. An example are the tiling assembly computations described in Section 5.
- *Local molecular computations* execute computations within a single molecule, possibly in parallel with other molecular computing devices. An example is Whiplash PCR described in Section 9.1.

2 Introducing DNA, its Structure and its Manipulation

2.1 Introducing DNA

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. While this makes the topic quite intellectually exciting, it also makes it challenging for a typical computer science reader. Having no training in biochemistry, he or she must obtain a coherent understanding of the topic from a short article. For this reason, this chapter was written with the expectation that the reader is a computer scientist with little background knowledge of chemistry or biochemistry. On the other hand, a reader with a basic knowledge of DNA, its structure and its enzymes can skip this section and proceed to the next.

2.2 DNA and its Structure

Here we give a brief introduction to DNA. *Single stranded DNA* (denoted ssDNA) is a linear polymer consisting of a sequence of nucleotide bases spaced along a backbone with chemical directionality (i.e. the 5-prime and 3-prime ends of the backbone are chemically distinct). 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 monomer units (base + sugar + phosphate) of an ssDNA molecule are connected 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 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 physiologic-like buffer solution with appropriate temperature, pH, and salinity.

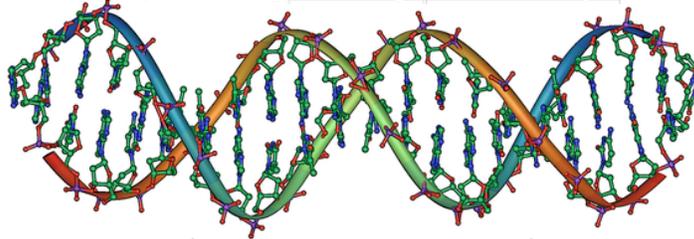


Figure 2.1: Structure of a DNA double helix (Image by Michael Ströck and released under the GNU Free Documentation License (GFDL).)

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 often occurs in a (random) zipper-like manner, similar to a biased one-dimensional random walk.

Whereas ssDNA is a relatively floppy molecule, dsDNA is quite stiff (over lengths of less than 150 or so bases) and has the well characterized double helix structure. The exact geometry (angles and positions) of each segment of a double helix depends slightly on the component bases of its strands and can be determined from known tables. There are about 10.5 bases per full rotation on this helical axis. A *DNA nanostructure* is a multi-molecular complex consisting of a number of ssDNA that have partially hybridized along their sub-segments.

2.3 Manipulation of DNA

Here we list some techniques and known enzymes used for manipulation of DNA nanostructures.

Strand displacement, which is the displacement of a DNA strand from a prior hybridization with another complementary strand is a key process in many of the DNA protocols for running DNA autonomous devices. Figure 2.2a illustrates DNA strand displacement via branch migration.

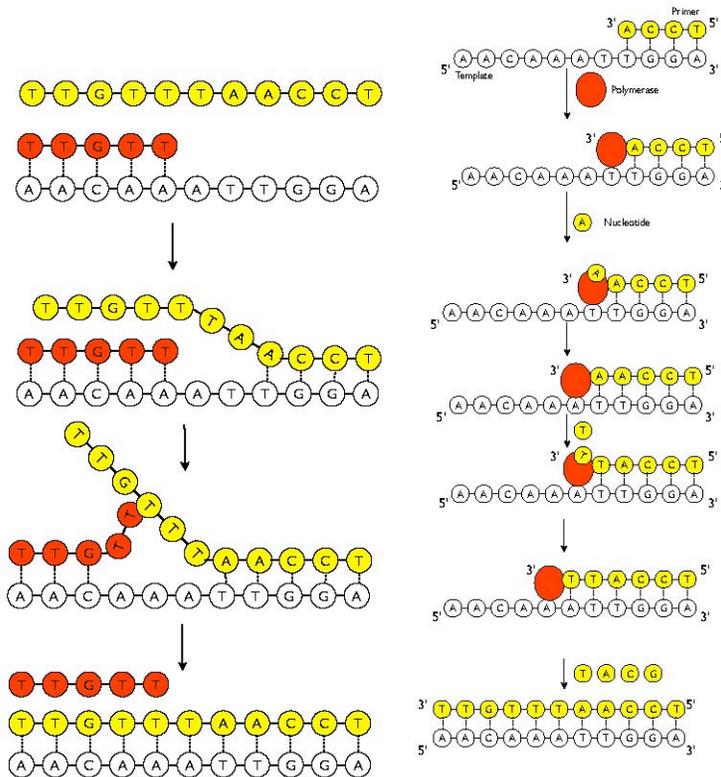


Figure 2.2a: (on left) Strand Displacement of dsDNA via a Branch Migration

Hybridization Reaction: Figure illustrates DNA strand displacement of a DNA strand (indicated in red) induced by the hybridization of a longer strand (indicated in yellow), allowing the structure to reach a lower energy state.

Figure 2.2b (on right) Extension of primer strand bound to the Template by DNA polymerase.

In addition to the hybridization reaction described above, 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*, some of which can cut (or nick, which is to cut only

one strand) strands of a DNA helix at locations determined by short specific DNA base sequences.

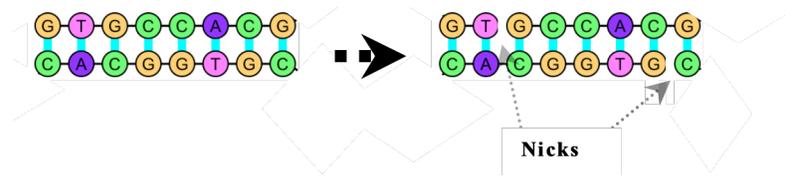


Figure 2.3: Example of Restriction enzyme cuts of a doubly stranded DNA subsequence.

- *Ligase enzymes* that can heal nicks in a DNA helix (i.e. form covalent bonds to join two backbones).
- *Polymerase Enzyme*, which given an initial “primer” DNA strand hybridized onto a segment of a template DNA strand, can extend the primer strand in the 5’ to 3’ direction by appending free DNA nucleotides complementary to the template’s nucleotides (see Figure 2.2b). Certain polymerase enzymes (e.g., phi-29) can, as a side affect of their polymerization reaction, efficiently displace previously hybridized strands.
- In addition, *Deoxyribozymes (DNAzymes)* are a class of class of nucleic acid molecules that possess enzymatic activity - they can, for example, cleave specific target nucleic acids. Typically, they are discovered by in-vivo evolution search. They have had some use in DNA computations (e.g., see [SS03]).

Besides their extensive use in other biotechnology, the above reactions, together with hybridization, are often used to execute and control DNA computations and DNA 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.

2.4 Why use DNA to Assemble Molecular-Scale Devices?

Below we list some reasons why DNA is uniquely suited for assembly of 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 structure of most complex DNA nanostructures can be reduced to

determining the structure of short segments of dsDNA. The basic geometric and thermodynamic properties of dsDNA are well understood and can be predicted by available software systems from key relevant parameters like sequence composition, temperature and buffer conditions.

- 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 (developed at NYU, Caltech, Arizona State, and Duke University) for design of the DNA sequences composing DNA tiles and for optimizing their stability, which employ heuristic optimization procedures for this combinatorial sequence design task (see Section 4.4 for more details).

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

- The solid-phase chemical synthesis of custom 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 melting temperature, 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 can provide information about the relative molecular mass of DNA molecules, as well as some information regarding their assembled structures. Other techniques like Atomic Force Microscopy (AFM) and Transmission Electron Microscopy (TEM) provide images of the actual assembled DNA nanostructures on 2D surfaces.

3 Adelman's Initial Demonstration of a DNA-based Computation

3.1 Adleman's Experiment

The field of DNA computing began in 1994 with a laboratory experiment [A94] [A94,A98]. The goal of the experiment was to find a Hamiltonian path in a graph, which is a path that visits each node exactly once. To solve this problem, a set of ssDNA were 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 (or a subset) 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.

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.

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. This autonomous property 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.

4 Self-Assembled DNA Tiles and Lattices

4.1 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 [R87,S04].

Particularly useful types of motifs often found in DNA nanostructures include:

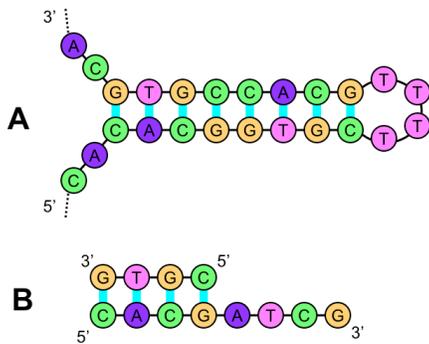


Figure 4.1: DNA Stem-Loop and a DNA Sticky End.

A *stem-loop* (A), 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 shown loop consists of the ssDNA region with sequence TTTT. Stem-loops are often used to form patterning on DNA nanostructures. A *sticky end* (B), where unhybridized ssDNA 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 4.1 shows the antiparallel nature of dsDNA with the 5' end of each strand pointing toward the 3' end of its partner strand.

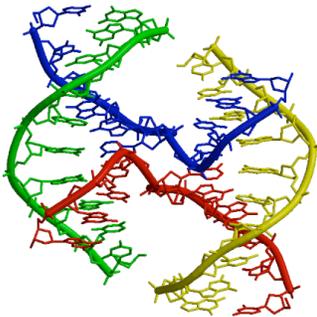


Figure 4.2: DNA Holliday Junction (Created by Miguel Ortiz-Lombardía, CNIO, Madrid, Spain.)

A *Holliday junction*, as illustrated in Figure 4.2, where two parallel 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.

4.2 Computation By Self-Assembly

The most basic way that 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 [W98] was the first to propose applying the concepts of computational tiling assemblies to DNA molecular constructs. His core idea was to use tiles composed of DNA to perform computations during their self-assembly process. To understand this idea, we will need an overview of DNA nanostructures, as presented in Section 4.3.

4.3 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.

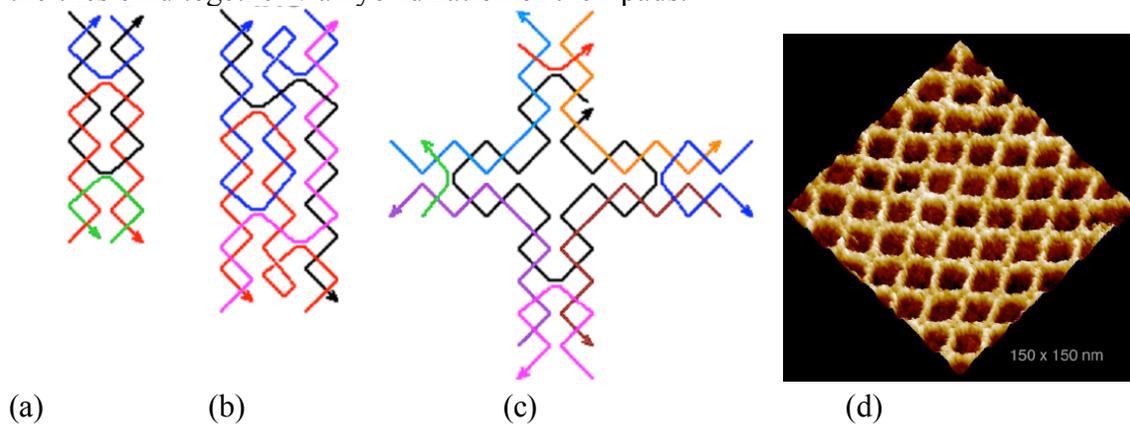


Figure 4.3: DNA Tiles (a) DX tile. (b) TX tile. (c) Cross tile. (d) AFM image of 2D DNA lattice of Cross tiles.

Figure 4.3 describes some principal DNA tiles (Also see [L06]).

Seeman and Winfree in 1998 developed a family of DNA tiles known collectively as DX tiles (see Figure 4.3a) that consisted of two parallel DNA helices linked by 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 Figure 4.3b) composed of three DNA helices. 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 of the component strands. 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).

Other DNA tiles known as *Cross tiles* (see Figure 4.3c) [Y03b] are shaped roughly square (or more accurately, square cruciform), 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 4.3(d) shows an AFM image of a 2D DNA lattice using Cross tiles.

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 (two pairs of sticky ends in the case of Cross tiles) of neighboring tiles are complementary, so only those pads hybridize together.

4.4 Software for Design of DNA Tiles

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. Figure 4.4 gives a screen shot of a software system known as TileSoft [Y04], developed jointly by Duke and Caltech, which provides a graphically-interfaced sequence optimization system for designing DNA secondary structures. A more recent commercial product, NanoEngineer, with enhanced capabilities for DNA design and a more sophisticated graphic interface, was developed by Nanorex, Inc.

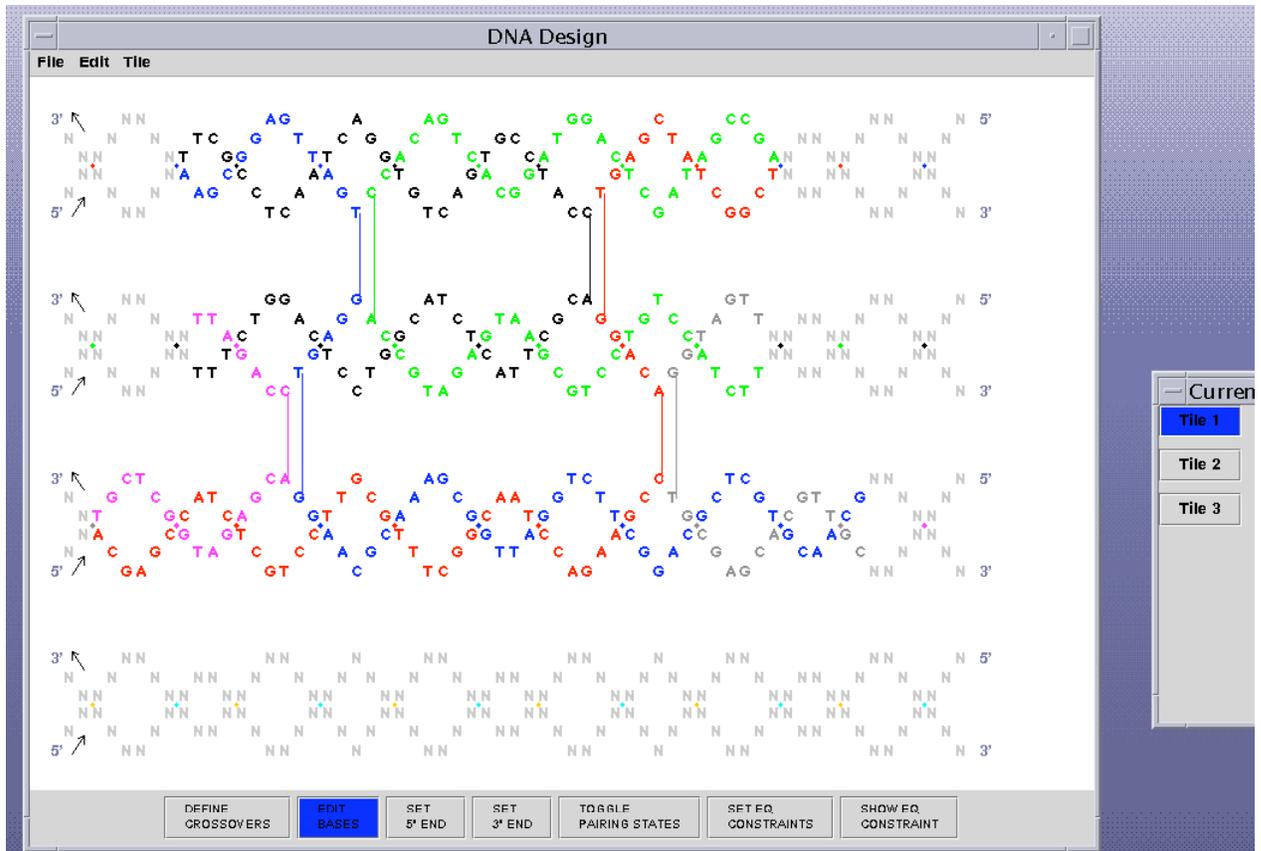


Figure 4.4. TileSoft: Sequence Optimization Software For Designing DNA Secondary Structures

5 Autonomous Finite State Computation Using Linear DNA Nanostructures

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 were done in 1999 [L99,L00,M00]. Among the experiments demonstrated [M00] was a 2-layer, linear assembly of TX tiles that executed a bit-wise cumulative XOR computation. In this computation, n bits are input and n bits are output, where the i^{th} output is the XOR of the first i input bits. This is the computation occurring when one determines the output bits of a full-carry binary adder circuit found on most computer processors. The experiment [M00] is illustrated in Figure 5.1.

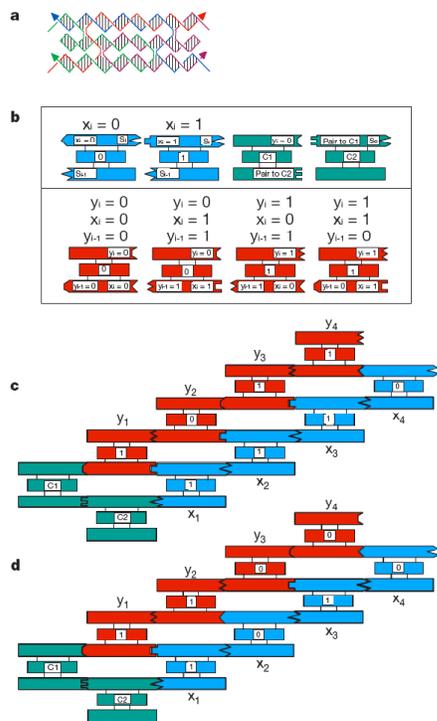


Figure 5.1: Sequential Boolean Computation via a Linear DNA Tiling Assembly (adapted with permission from [M00]) (a) TX tile used in assembly. (b) Set of TX tiles providing logical programming for computation. (c), (d) example resulting computational tilings.

These experiments [L99,L00,M00] provided initial answers to some of the most basic questions of how autonomous molecular computation might be done:

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

In this experiment the input sequence of n bits was defined using a specific series of “input” tiles with the input bits (1’s & 0’s) encoded by distinct short subsequences. Two different tile types (depending on whether the input bit was 0 or 1, these had specific sticky-ends and also specific subsequences at which restriction enzymes can cut the DNA backbone) were assembled according to specific sticky-end associations, forming the blue input layer illustrated in Figure 5.1.

Figure 5.1 shows a unit TX tile (a) and the sets of input and output tiles (b) with geometric shapes conveying sticky-end complementary matching. The tiles of (b) execute binary computations depending their pads, as indicated by the table in (b). The (blue) input layer and (green) corner condition tiles were designed to assemble first (see example computational assemblies (c) & (d)). The (red) output layer then assemble specifically starting from the bottom left using the inputs from the blue layer. (See [M00] for more details of this molecular computation.) 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 (see below). In principle, the output could be used for subsequent computations.

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 such 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 to be the XOR of these two bits.

A 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 reporter strand, then digesting separate aliquots with each endonuclease separately and the two together, and 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. The patterning was made by designing the tiles computing a bit 1 to have a stem loop protruding from the top of the tile. The sequence of this molecular patterning was clearly observable under appropriate AFM imaging conditions.

Although only very simple computations, the experiments of [L99,L00,M00,Y03b] did demonstrate for the first time 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 will be presented below in Subsection 5.2.

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 [S06]. Their technique essentially operated in the reverse of the assembly methods described above, and instead can be thought of as disassembly. They began with a linear double-stranded 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.

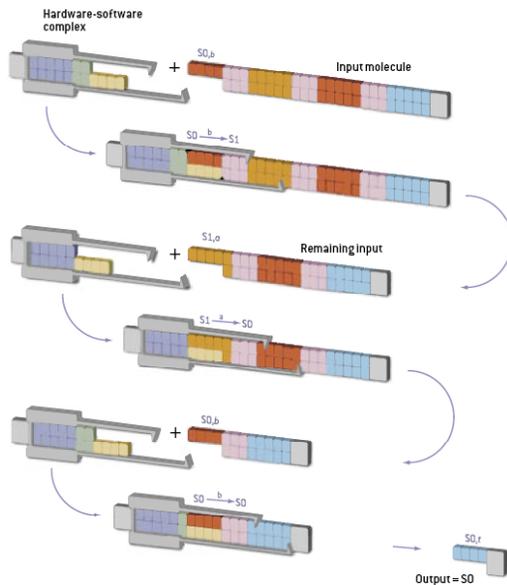


Figure 5.2: Autonomous Finite-State Computations via Disassembly of a double-stranded DNA Nanostructure (Figure adapted with permission from [S06]).

The hardware-software complex for this molecular device 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).

This ingenious design is an excellent demonstration that there is often more than one way to do any task at the molecular scale. [B04] (see the conclusion Section 11 for further discussion) demonstrated in the test tube a potential application of such a finite-state computing device to medical diagnosis and therapeutics.

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 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. Examples of addressable 2D DNA lattices will be given in Section 6.2.

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.

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 compounds, carbon nanotubes (via various attachment chemistries), etc. These attachment technologies provide for a molecular-scale method attaching heterogeneous materials to DNA nanostructures. For example, it can potentially be used for attaching molecular electronic devices to a 2D or 3D DNA nanostructure. [Y03c,P06] describes conductive wires fabricated from self-assembled DNA tubes plated with gold or silver, as illustrated in Figure 6.1.

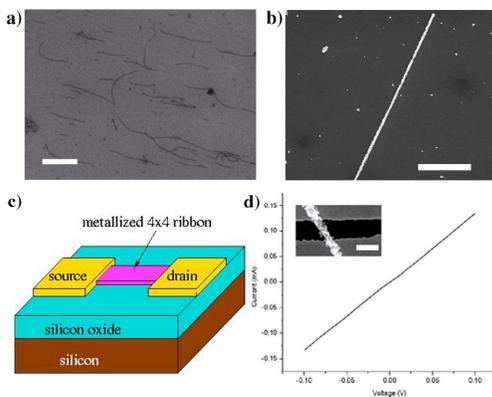


Figure 6.1. Conductive wires fabricated from self-assembled DNA tubes plated with silver. (a) DNA tubes prior to plating. (b) DNA tubes after silver plating. (c) Illustration of conductivity test on silicon oxide substrate. (d) TEM image of conductivity test on silicon oxide substrate.

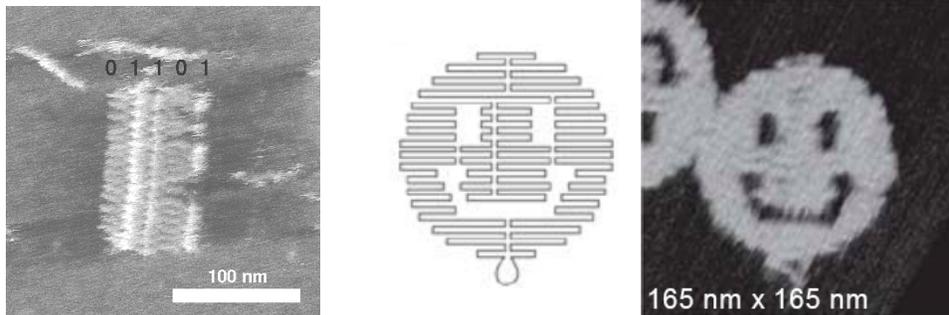
6.2 Methods for Programmable Assembly of Patterned 2D DNA Lattices

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 for assembling patterned 2D DNA lattices that now have been experimentally demonstrated, as described in the next few subsections.

6.2.1 Methods for Programmable Assembly of Patterned 2D DNA Lattices by use of

Scaffold Strands



(a) Barcode Patterning (b) DNA origami design (c) AFM imaging of DNA origami

Figure 6.2: Methods for programmable assembly of patterned 2D DNA Lattices by use of scaffold strands.

A *scaffold strand* is a long ssDNA around which shorter ssDNA assemble to form structures larger than individual tiles. Scaffold strands were used to demonstrate programmable patterning of 2D DNA lattices in [Y03a] 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 the desired distinct sequence of 2D barcode patterns (left panel of Figure 6.2). 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* [R06a]. 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 few 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 the middle panel of Figure 6.2, and an AFM image of the resulting assembled origami is shown in the right panel of Figure 6.2. This landmark work of Rothemund [R06a] 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 principle this “molecular origami” method with staple strands can be used to form arbitrary complex 2D patterned nanostructures as defined. DNA origami has now been extended to simple 3D filaments that were used to partially orient membrane proteins in structural studies employing solution NMR [D07].

6.2.2 Programmable Assembly of Patterned 2D DNA Lattices by Computational

Assembly

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. Winfree's group has experimentally demonstrated various 2D computational assemblies, and furthermore provided AFM images of the resulting nanostructures:

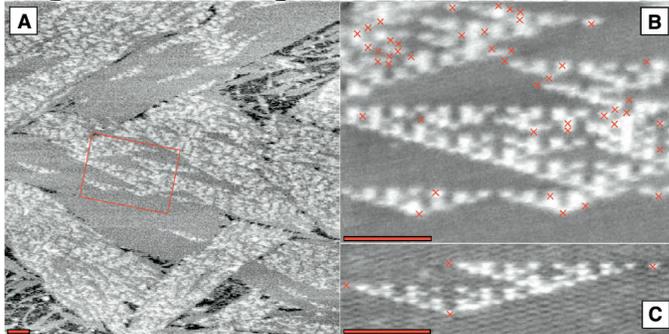
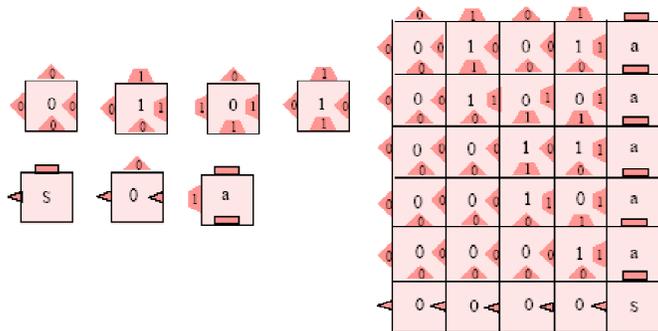


Figure 6.3: Programmable Assembly of Sierpinski Triangle by use of Computational Assembly (Figure adapted with permission from [R04].)

Figure 6.3 gives a modulo-2 version of Pascal's Triangle (known as the Sierpinski Triangle) [R04], where each tile determines and outputs to neighborhood pads the XOR of two of the tile pads. Example AFM images (scale bars = 100 nm) of the assembled structures are shown in the three panels of Figure 6.3.



(a) tiles used

(b) binary counter assembly

Figure 6.4: Rothmund's and Winfree's design for a self-assembled binary counter using Tilings.

Figure 6.4 gives Rothmund's and Winfree's design for a self-assembled binary counter, 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 to patterning molecular electronic circuits.

6.2.3 Programmable Assembly of Patterned 2D DNA Lattices by Hierarchical Assembly

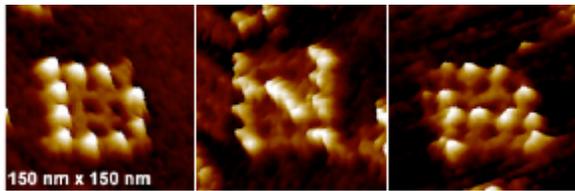


Figure 6.5: 2D Patterns By Hierarchical Assembly (Figure adapted with permission from [P06a].) AFM images of characters D, N, and A.

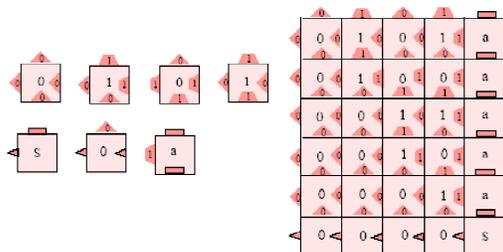
A further approach, known as hierarchical assembly [P06a], is to assemble DNA lattices in multiple stages. Figure 6.5 gives three examples of preprogrammed patterns displayed on addressable DNA tile lattices. Tiles are assembled prior to mixing with other preformed tiles. Unique ssDNA pads direct tiles to designed locations. White pixels are “turned on” by binding a protein (avidin) 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.

7. Error correction and Self-repair at the Molecular-Scale

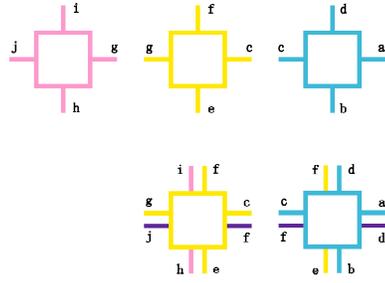
7.1 The Need for Error correction at the Molecular-Scale

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 comprising 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, as described below.

7.2 Winfree’s Proofreading Scheme for Error-Resilient Tilings



(a) Original tiles:

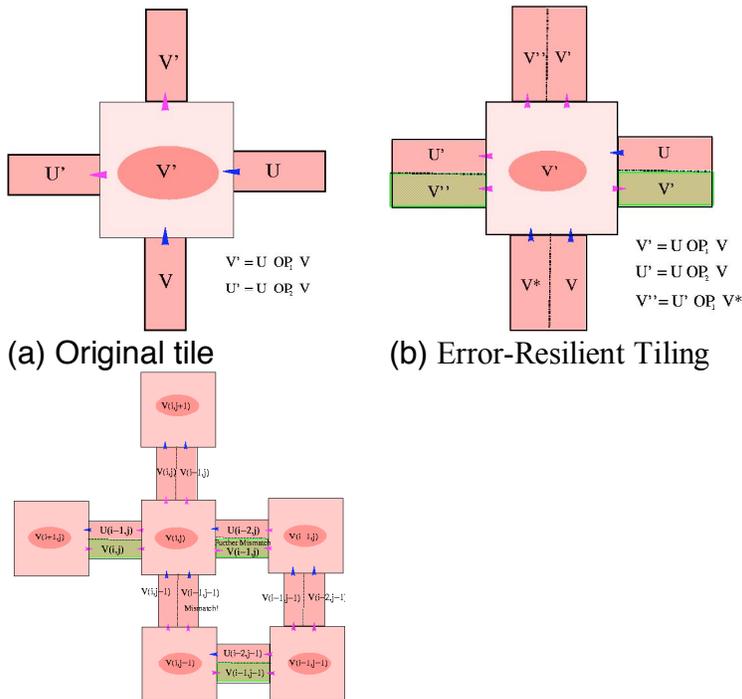


(b) Error resilient tiles:

Figure 7.1: Winfree's Proofreading Scheme for Error-Resilient Tilings

Winfree and Bekbolatov in 2004 [W04] developed a “proofreading” method of replacing each tile with a subarray of tiles that provide sufficient redundancy to quadratically reduce errors, as illustrated in Figure 7.1. This method however increased the area of the assembly by a factor of 4.

7.3 Reif's Compact Scheme for Error-Resilient Tilings



(c) A single pad mismatch causes another pad mismatch so destabilizing assembly

Figure 7.2: A Compact Scheme for Error-Resilient Tilings.

Reif et al [R06b] proposed a more compact method for decreasing assembly errors, as illustrated in Figure 7.2. This method modifies the pads of each tile, so that essentially each tile both executes the original computation required at that location, as well as the computation of a particular neighbor, providing 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 (such as error-correcting codes) developed in computer science may also be utilized.

7.4 Activatable Tiles for Reducing Errors

The uncontrolled assembly of tiling assemblies in reverse directions is potentially a major source of errors in computational tiling assemblies, and a roadblock in the development of applications of large patterned computational DNA lattices. Methods for controlled directional assembly of tiling assemblies would eliminate these errors. Majumder et al [U07] have recently developed novel designs for an enhanced class of error-resilient DNA tiles (known as *activatable tiles*) for controlled directional assembly of tiles. While conventional DNA tiles store no state, the activatable tiling systems makes use of a powerful DNA polymerase enzyme that allows the tiles to transition between active (allowing assembly) and inactive states. A *protection-deprotection* process strictly enforces the direction of tiling assembly growth so that the assembly process is robust against entire classes of growth errors. Initially, prior to binding with other tiles, the tile will be in an inactive state, where the tile is protected from binding with other tiles and thus preventing lattice grow in the (unwanted) reverse direction. After appropriate other tiles bind to this tile, the tile transitions to an active state, allowing further growth.

8 Three Dimensional DNA Lattices

8.1 Potential Applications of 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 very important applications to nanoelectronics and biology if this can be done, as described below.

8.2 Scaffolding of 3D nanoelectronic architectures

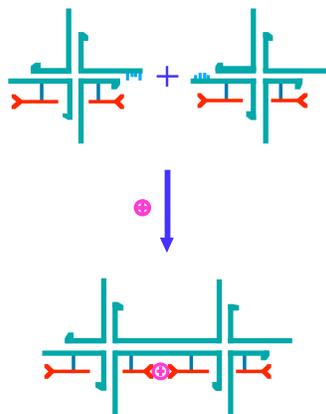


Figure 8.1: Scaffolding of 3D nanoelectronic architectures (adapted with permission from [R87].)

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 8.1 shows DNA (cyan) and protein (red) organizing functional electronic structures

8.2 Scaffolding of 3D nanoelectronic architectures

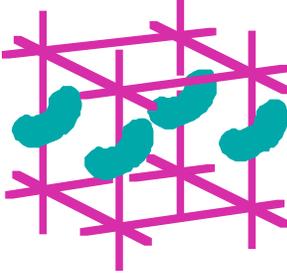


Figure 8.2: Scaffolding of proteins into regular 3D arrays (adapted with permission from [R87].)

It has been estimated that at least one half of all natural proteins can not 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 (see Figure 8.2). 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.

9 From Nucleic Detection Protocols to Autonomous Computation

9.1 The Detection Problem

A fundamental task of many biochemical protocols is to sense a particular molecule and then amplify the response. In particular, the detection of specific strands of RNA or DNA is an important problem for medicine. Typically, a protocol for nucleic detection is specialized to a subsequence of single stranded nucleic acid (DNA or RNA oligonucleotide) to be detected. Give a sample containing a very small number of the nucleic strand molecules to be detected, a detection protocol must amplify this to a much larger signal. Ideally, the detection protocol is exquisitely sensitive, providing a response from the presence of only a few of the target molecules.

There are a number of novel methods for doing DNA computation that can be viewed as

being derived from protocols for detection of DNA. Therefore, understanding the variety of detection protocols can provide insight into these methods used by for DNA computation

9.2 Methods for Autonomous Molecular Computation derived from PCR

9.2.1 The polymerase chain reaction (PCR)

The original and still the most frequently use method for DNA detection is the *polymerase chain reaction (PCR)* [S85], which makes use of DNA polymerase to amplify a strand of DNA by repeated replication, using rounds of thermal-cycling. (Recall that given an initial “primer” DNA strand hybridized onto a segment of a template DNA strand, polymerase enzyme can extend the primer strand by appending free DNA nucleotides complementary to the template’s nucleotides.) In addition to DNA polymerase, the protocol requires a pair of “primer” DNA strands, which are extended by the DNA polymerase, each followed by heating and cooling, to allow displacement of the product strands.

9.2.2 Whiplash PCR: A Method for Local Molecular Computation

A method for DNA computation, known as Whiplash PCR [S98] makes use of a strand of DNA that essentially encodes a “program” describing state transition rules of a finite state computing machine; the strand is comprised of a sequence of “rule” subsequences (each encoding a state transition rule), and each separated by stopper sequences (which can stop the action of DNA polymerase). On each step of the computation, the 3’ end of the DNA strand has a final sequence encoding a state of the computation. A computation step is executed when this 3’ end hybridizes to a portion of a “rule” subsequence, and the action of DNA polymerase extends the 3’ end to a further subsequence encoding a new state.

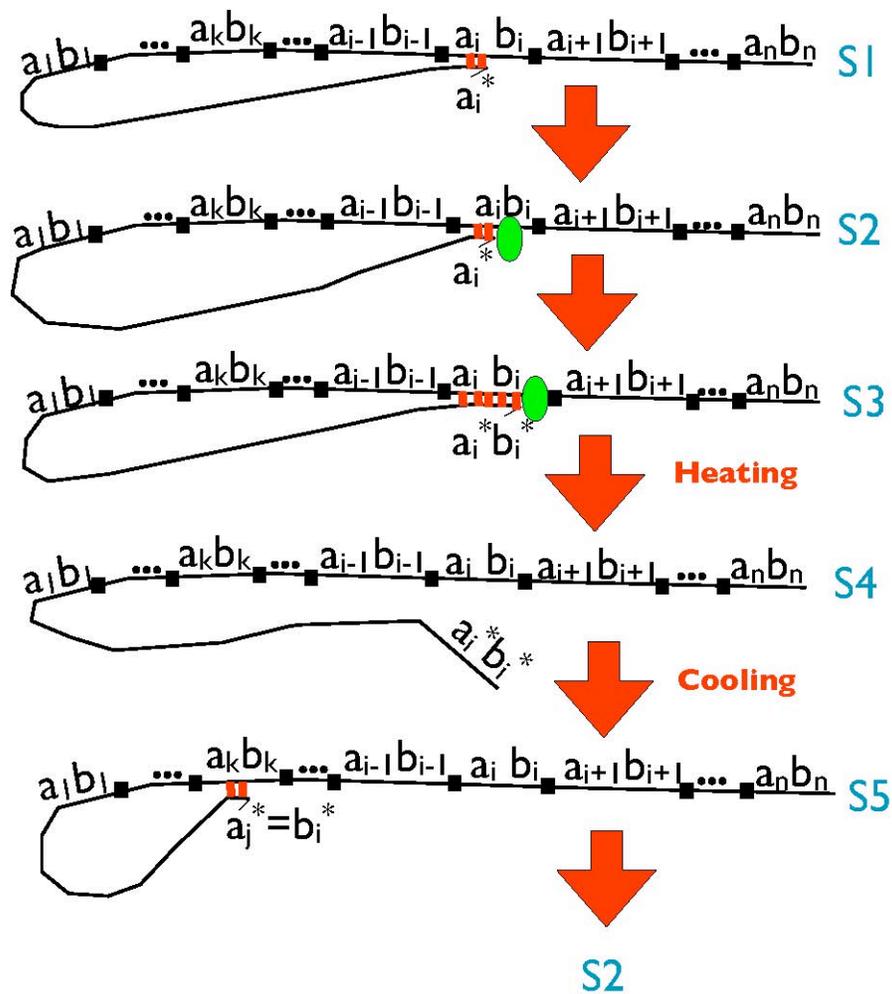


Figure 9.1: Whiplash PCR: Repeated rounds of heating and cooling allow for the execution of a finite state transition machine with state transitions $a_i \Rightarrow b_i$ encoded by the DNA strand sequence. Each of the state transitions is executed in the cool stages, where if the current state is a_i , the 3' end of the DNA strand has the complement at its 3' end, which hybridizes to a sequence a_i and the state transition via polymerization extension step at the 3' end, allowing a transition to a state b_i . The heating stage allows for displacement of the extended 3' end.

Whiplash PCR is interesting, since it executes a local molecular computation (recall that A molecular computation is local if the computation within a single molecule, possibly in parallel with other molecular computing devices). In contrast, most methods for autonomous molecular computation (such as those based on the self-assembly of tiles) provide only a capability for distributed parallel molecular computation since to execute a computation they require multiple distinct molecules that interact to execute steps of each computation.

9.3 Isothermal and Autonomous PCR Detection and Whiplash PCR Computation

Protocols

Neither the original PCR protocol nor the Whiplash PCR of [S98] executes autonomously – they require thermal cycling for each step of their protocols. Walker et al [W92a, W92b] developed isothermal (requiring no thermal cycling) methods for PCR known as Strand Displacement Amplification (SDA) in which strands displaced from DNA polymerase are used for the further stages of the amplification reaction. Reif and Majumder recently developed [R09] an autonomously executing version of Whiplash PCR (known as isothermal reactivating Whiplash PCR) that makes use of a strand-displacing polymerization enzyme (Recall however that certain polymerase enzymes such as phi-29 can, as a side effect of their polymerization reaction, displace previously hybridized strands) with techniques to allow the reaction to process isothermally. In summary, an isothermal variant (Strand-displacement PCR) of the basic PCR detection protocol provided insight on how to design an autonomous method for DNA computation. Like Whiplash PCR, this new isothermal reactivating Whiplash PCR provides for local molecular computation.

9.4 Autonomous Molecular Cascades and Related Hybridization Reactions for Autonomous DNA Computation

9.4.1 Autonomous Molecular Cascades for DNA Detection

Dirks and Pierce [DP04] demonstrated an isothermal, enzyme-free (most known detection protocols require the use of protein enzymes) method for highly sensitive detection of a particular DNA strand. This protocol makes a triggered amplification by hybridization chain reaction briefly illustrated in Figure 9.2.

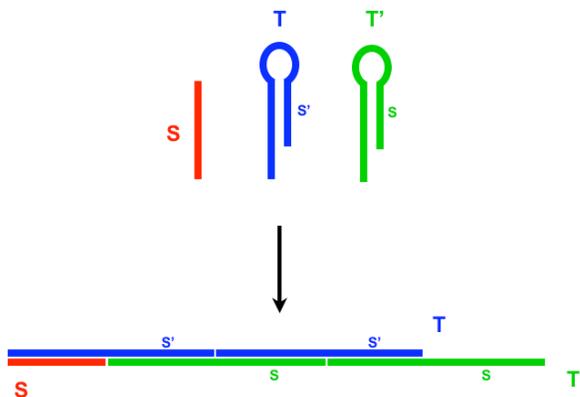


Figure 9.2 Autonomous Molecular Cascade for Signal Amplification

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

9.4.1 Hybridization Reactions for Autonomous DNA Computation

Winfrey [Z07] developed a general methodology for designing systems of DNA molecules by the use of catalytic reactions that are driven by entropy. In particular, [Z07] demonstrates a general, powerful scheme for executing any Boolean circuit computation via cascades of DNA hybridization reactions. The unique common property of the above detection protocol [DP04] and the molecular computations of [Z07] are their use only of hybridization, making no use of restriction enzyme or any other protein enzymes. Following on this work, Yin [Y08] developed an elegant and highly descriptive labeled diagram scheme (with nodes indicating inputs, products, etc.) for illustrating the programming of biomolecular self-assembly and reaction pathways.

9.5 Autonomous Detection Protocols and Molecular Computations Using DNAzyme

In addition, Mao [M04] demonstrated a novel method for DNA detection which made use of a dual set of DNAzyme (recall a DNAzyme is a DNA molecule that possess enzymatic activity, in particular cutting particular single stranded DNA) that provided for amplified DNA detection. This led to the DNAzyme based autonomous DNA walker [T05] described in section 10.4.2.

10 Autonomous Molecular Transport Devices Self-Assembled from DNA

10.1 Molecular Transport

Many molecular-scale tasks may require the transport of molecules and there are a number of other tasks that can be done at the molecular scale that would be considerably aided by an ability to transport within and/or along nanostructures. For example of the importance of molecular transport in nano-scale systems, consider the cell, which uses protein motors fueled by ATP to do this.

10.2 Non-autonomous DNA motor devices

In the early 2000's a number of researchers developed and demonstrated motors composed of DNA nanostructures; for example, Yurke [Y00] demonstrated a DNA actuator powered by DNA hybridization (complementary pairing between DNA

strands). However, all of these DNA motor devices required some sort of externally mediated changes (such as temperature-cycling, addition or elimination of a reagent, etc.) per work-cycle of the device, and so did not operate autonomously.

10.3 The Need for Autonomous Molecular Transport

Almost all of the conventionally-scaled motors used by mankind run without external mediation, and almost all natural systems for molecular motors are also autonomous (e.g., the cell's protein motors are all autonomous). The practical applications of molecular devices requiring externally mediated changes per work-cycle are quite limited. So it is essential to develop autonomous DNA devices that do not require external mediation while executing movements.

10.4 Autonomous DNA Walkers

Reif [R03] first described the challenge of autonomous molecular transport devices which he called “DNA walkers” that traversed DNA nanostructures, and proposed two designs that gave bidirectional movement. In 2004, Sherman and Seeman demonstrated a DNA walker [S04], but it was non-autonomous since it required external mediation for every step it made.

10.4.1 Restriction Enzyme Based Autonomous DNA Walkers

The first autonomous DNA walker was experimentally demonstrated in 2004 by Yin, Turberfield and Reif [Y04]. It employed restriction enzymes and ligase; see [Y05] for its detailed general design.

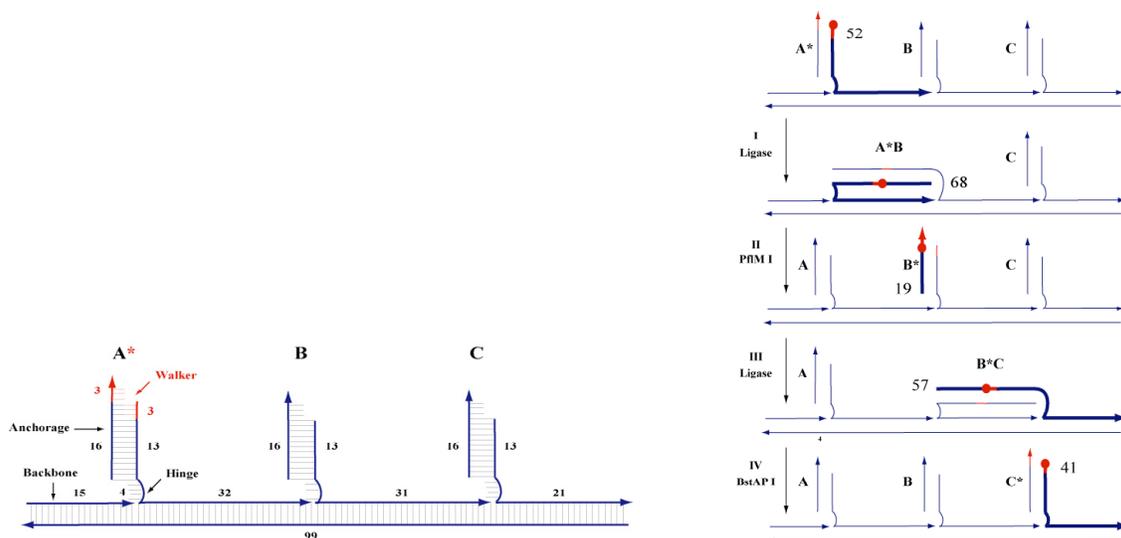


Figure 10.1: Autonomous Molecular Transport Devices Self-Assembled from DNA

The device is described in Figure 9.1.

- Initially a linear DNA nanostructure (the “road”) with a series of attached ssDNA strands (the “steps”) is self-assembled. Also, a fixed-length segment of DNA helix (the “walker”) with short sticky ends (it’s feet”) hybridized to the first two steps

of the road.

- Then the walker proceeds to make a sequential movement along the road, where at the start of each step, the feet of the walker are hybridized to two further consecutive two 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.

10.4.2 DNAzyme Based Autonomous DNA Walkers

Subsequently in 2005 Mao's group [T05] demonstrated an autonomous DNA walker that made use of a DNAzyme motor [C04] that used the cuts provided by the enzymatic activity of DNAzyme to progress along a DNA nanostructure.

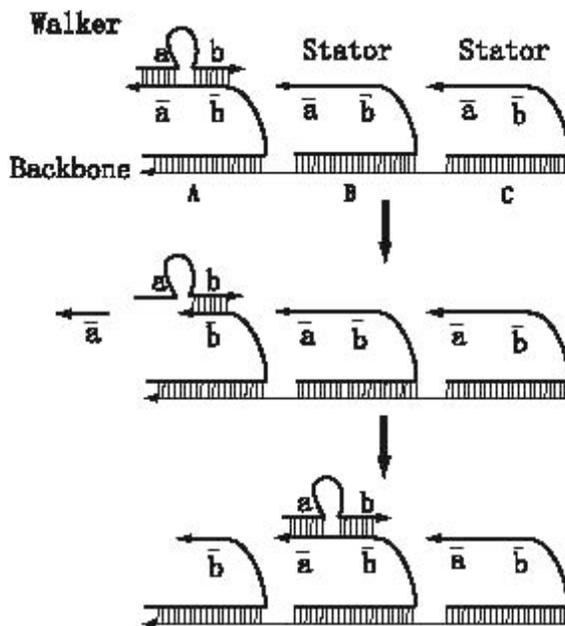


Figure 10.2: Mao's DNAzyme Walker

J. Bath and A. Turberfield [B07] also give an extensive survey of these and further recent DNA motor and walker devices.

10.5 Programmable Autonomous DNA Devices: Nanobots

There are some important applications of these autonomous DNA walkers including transport of molecules within large self-assembled DNA nanostructures. However, the potential applications may be vastly increased if they can be made to execute

computations while moving along a DNA nanostructure. This would allow them, for example to make programmable changes to their state and to make movements programmable. We will call such programmable autonomous DNA walker devices “programmable DNA nanobots”. Yin et al. [Y06] describe an extension of the design of the restriction-enzyme based autonomous DNA walker [Y04] described above in subsection 10.4.3, to allow programmed computation while moving along a DNA nanostructure.

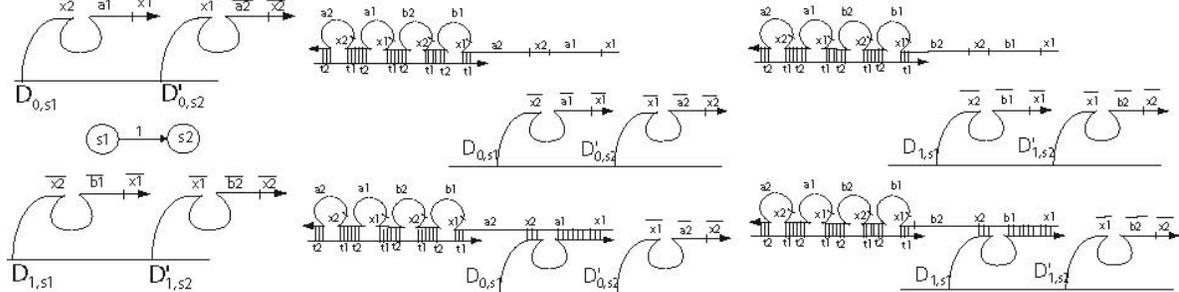


Figure 10.3: Reif and Sahu’s DNA nanobot: (a) Figure illustrates the implementation of a state transition through DNazymes. (b) $D_{0,s1}$ in the transition machinery for state transition at 0 combines with input nanostructure when active input symbol encoded by the sticky end is 0. When the active input symbol encoded by the sticky end is 1, $D_{1,s1}$ in the transition machinery for state transition at 1 combines with the input nanostructure.

Another DNA nanobot design (see Figure 10.3) for programmed computation while moving along a DNA nanostructure was developed by Reif and Sahu [R07] using in this case an extension of the design of the DNzyme based autonomous DNA walker [T05] also described above.

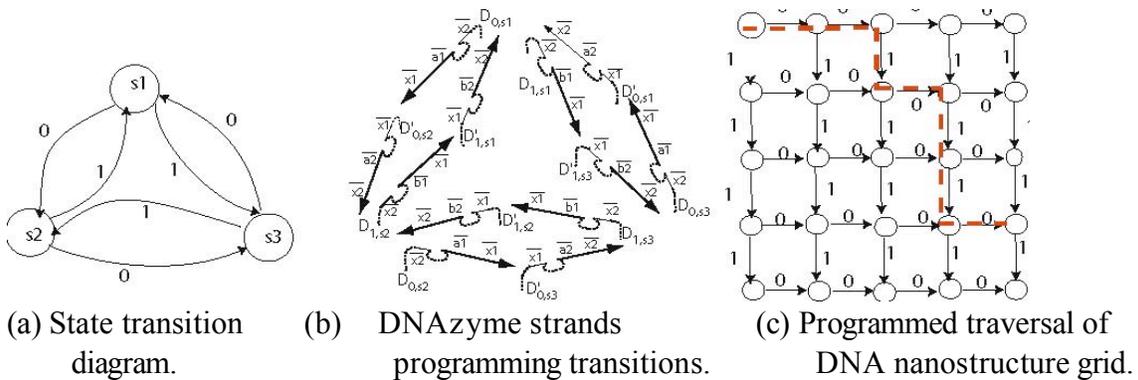


Figure 10.4: Use of Reif and Sahu’s DNA nanobot to execute a programmed traversal of a grid DNA nanostructure: (a) The DNzyme implementation of the finite state machine shown on left. (b) Illustration of programmable routing in two dimensions.

It remains a challenge to experimentally demonstrate these.

11 Conclusions and Challenges

11.1 What was Covered and What was Missed: Further Reading

Our chapter has covered most of the major known techniques and results for autonomous methods for DNA-based computation and transport.

However, there is a much larger literature of DNA-based computation that includes methods that are non-autonomous, but otherwise often ingenious and powerful. As just one notable example, Stojanovic demonstrated a deoxyribozyme-based molecular automaton [S03] and demonstrated its use to play the optimal strategy for a simple game.

Other excellent surveys of DNA nanostructures and devices have been given by Seeman [S04,S05], Mao [D06], de Castro [C06], LaBean and Li [L07], Yan and Liu [Y07], and Bath and Turberfield [B07].

11.2 Future Challenges for Self-Assembled DNA Nanostructures

There are a number of key challenges still confronting this emerging field:

Experimentally demonstrate:

(1) *Complex, error-free DNA patterning to the scale say at least 10,000 pixels – as required say for a functional molecular electronic circuit for a simple processor.*

Note: This would probably entail the use of a DNA tiling error correction method as well as a significant improvement over existing DNA patterning techniques.

(2) *A programmable DNA Nanobot autonomously executing a task critical to nano-assembly.*

Note: The first stage might be a DNA walker that can be programmed to execute various distinct, complex traversals of a 2D DNA nanostructure, and to load and unload molecules at chosen sites on the nanostructure.

(3) *An application of self-assembled DNA nanostructures to medical diagnosis.*

[B04] was the first to propose and to demonstrate in the test tube a finite-state computing DNA device for *medical diagnosis*: the device detect RNA levels (either over or under expression of particular RNA), compute a diagnosis based on a finite-state computation, and then provide an appropriate response (e.g, the controlled release of a single-stranded DNA that either promotes or interfere with expression). They demonstrated in the test tube a potential application of such a finite-state computing device to medical diagnosis

and therapeutics. Reif and Sahu [R07] described a DNAzyme based autonomous DNA nanobot (see Section 10.4) that also can function as a finite-state computing DNA device for medical diagnosis.

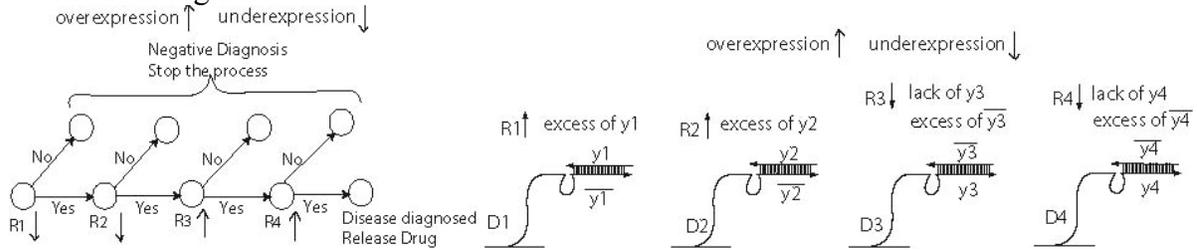


Figure 10.4: A finite-state computing DNA device for medical diagnosis based on Reif and Sahu’s DNAzyme based autonomous DNA nanobot. (a) A state diagram for DNAzyme doctor nanobot that controls the release of a “drug” RNA on the basis of the RNA expression tests for a disease. (b) The figure shows the consequences of overexpression and underexpression of different RNAs on the concentrations of the respective characteristic sequences. The overexpression of R1 and R2 results in excess of y_1 and y_2 respectively, and they block the path of input nanostructure by hybridizing with D1 and D2. Similarly underexpression of R3 and R4 results in excess of \bar{y}_3 and \bar{y}_4 respectively, to block the path of input nanostructure.

It remains a challenge to apply such a finite-state computing DNA device for medical diagnosis within the cell, rather than in the test tube.

11.3 Credits and Thanks

Thanks to Nikhil Gopalkrishnan, Urmi Majumder, and Sudheer Sahu for their very useful comments on this article. Supported by NSF Grants CCF-0829797 and CCF-0829798

12 References

[A94] L.M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, 1994.

[A98] Leonard Adleman, Computing with DNA, *Scientific American*, 279(2), p 34-41, (August 1998).

[R87] Robinson, B.H. & Seeman, N.C. (1987), *Protein Eng.* 1, 295-300.

[W98] 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.

- [B05] Robert D. Barish, Paul W. K. Rothemund, and Erik Winfree, Two Computational Primitives for Algorithmic Self-Assembly: Copying and Counting, *NanoLetters* Vol. 5, No. 12 2586-2592, (2005).
- [P06b] 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.
- [L99] 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).
- [L00] 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).
- [M00] 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).
- [L06] 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).
- [R04] Paul W.K. Rothemund, Nick Papadakis, Erik Winfree, Algorithmic Self-Assembly of DNA Sierpinski Triangles, *PLoS Biology* 2 (12), (Dec., 2004).
- [R06a] Paul W. K. Rothemund, Folding DNA to create nanoscale shapes and patterns, *Nature* 440, 297-302 (16 March 2006).
- [P06a] 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.
- [Y04] Peng Yin, Bo Guo, Christina Belmore, Will Palmeri, Erik Winfree, Thomas H. LaBean, John H. Reif, TileSoft: Sequence Optimization Software For Designing DNA Secondary Structures, Abstract, Preliminary Proceedings, Tenth International Meeting on DNA Based Computers (DNA10), Milano, Italy, (June 7-10, 2004).

[W04] Erik Winfree and Renat Bekbolatov, Proofreading Tile Sets: Error Correction for Algorithmic Self-Assembly, in DNA Computers 9, LNCS, (2943), pp. 126-144, (2004).

[R06b] 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 (eds. J. Chen; N. Jonoska & G. Rozenberg), pages 79-104, (2006).

[U07] Urmi Majumder, Thomas H LaBean, and John H. Reif, Activatable Tiles: Compact, Robust Programmable Assembly and Other Applications, in DNA Computing: DNA13 (edited by Max Garzon and Hao Yan), Springer-Verlag Lecture Notes for Computer Science (LNCS), Springer, Berlin Heidelberg, Volume 4848, pp. 15-25 (2007).

[Y03a] 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).

[Y03b] 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.

[B04] Adar R., Benenson Y., Linshiz G., Rozner A, Tishby N. and Shapiro E. (2004) Stochastic computing with biomolecular automata. Proc. Natl. Acad. Sci. USA, 101, 9960-65.

[Y03c] 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).

[S85] Saiki, RK; Scharf S, Faloona F, Mullis KB, Horn GT, Erlich HA, Arnheim N. Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia, Science 230 (4732): 1350-4.(December 20 1985)

[S98] Kensaku Sakamoto, Daisuke Kiga, Ken Momiya, Hidetaka Gouzu, Shigeyuki Yokoyama, Shuji Ikeda, Hiroshi Sugiyama, and Masami Hagiya. State transitions with molecules. In Proceedings of the 4 th DIMACS Meeting on DNA Based Computers, held at the University of Pennsylvania, June 16-19, 1998. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Providence, RI, American Mathematical Society, 1999.

[W92a] Walker, GT; Little, MC; Nadeau, JG; Shank, DD. Isothermal in vitro

amplification of DNA by a restriction enzyme/DNA polymerase system. *Proc Natl Acad Sci U S A*. 1992 Jan 1;89(1):392–396 (1992).

[W92b] G T Walker, M S Fraiser, J L Schram, M C Little, J G Nadeau, and D P Malinowski, Strand displacement amplification--an isothermal, in vitro DNA amplification technique, *Nucleic Acids Res*. April 11; 20(7): 1691–1696. (1992).

[DP04] Dirks, R. M. & Pierce, N. A.. Triggered amplification by hybridization chain reaction. *Proc. Natl Acad. Sci. USA* 101, 15275–15278 (2004).

[Z07] David Yu Zhang, Andrew J. Turberfield, Bernard Yurke, and Erik Winfree, Engineering Entropy-Driven Reactions and Networks Catalyzed by DNA, *Science*, 318:1121-1125, (2007).

[Y08] Programming Biomolecular Self-Assembly Pathways. P. Yin, H.M.T. Choi, C.R. Calvert, N.A. Pierce. *Nature*, 451:318-322, (2008).

[R08] John H. Reif and Urmi Majumder, Isothermal Reactivating Whiplash PCR for Locally Programmable Molecular Computation, Fourteenth International Meeting on DNA Based Computers (DNA14), Prague, Czech Republic (June, 2008). Lecture Notes for Computer Science (LNCS), NYC, NY, (edited by Ashish Goel and Friedrich C. Simmel), Springer-Verlag, New York, (2009). Invited Paper, Special issue in *Natural Computing*, (2009).

[M06] Ye Tian, Yu He, Chengde Mao, Cascade Signal Amplification for DNA Detection, *ChemBioChem*, Volume 7 Issue 12, Pages 1862 – 1864 (29 Sep 2006).

[Y00] B. Yurke, A.J. Turberfield, A.P. Mills, Jr., F.C. Simmel, and J.L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406:605–608, 2000.

[S04] WB Sherman and NC Seeman, A Precisely Controlled DNA Biped Walking Device. *Nano Letters*, American Chemical Society, No. 4, pp. 1203. (2004).

[R03] John H. Reif, The Design of Autonomous DNA Nanomechanical Devices: Walking and Rolling DNA. DNA Based Computers (DNA8), Sapporo, Japan, June 10-13, 2002, (Edited by Masami Hagiya and Azuma Ohuchi), Lecture Notes in Computer Science, No. 2568, Springer-Verlag, New York, (2003), pp. 22-37. Published in *Natural Computing*, DNA8 special issue, Vol. 2, p 439-461, (2003).

[Y04] 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).

[C04] Y. Chen, M. Wang, and C. Mao. An autonomous DNA nanomotor powered by a DNA enzyme. *Angew. Chem. Int. Ed.*, 43:3554–3557, (2004).

[T05] Y. Tian, Y. He, Y. Chen, P. Yin, and C. Mao. Molecular devices - a DNAzyme that walks processively and autonomously along a one-dimensional track. *Angew. Chem. Intl. Ed.*, 44:4355–4358, 2005.

[R09] John H. Reif and Urmi Majumder, Isothermal Reactivating Whiplash PCR for Locally Programmable Molecular Computation, Fourteenth International Meeting on DNA Based Computers (DNA14), Prague, Czech Republic (June, 2008). Lecture Notes for Computer Science (LNCS), NYC, NY, (edited by Ashish Goel and Friedrich C. Simmel), Springer-Verlag, New York, (2009). Invited Paper, Special issue in Natural Computing, (2009).

[SS03] Stojanovic, M. N. & Stefanovic, D. A deoxyribozyme-based molecular automaton. *Nature Biotechnol.* 21, 1069–1074 (2003).

[BT07] J. Bath and A. Turberfield. DNA nanomachines. *Nature Nanotechnology*, 2:275–284, May 2007.

[Y05] Peng Yin, Andrew J. Turberfield, Sudheer Sahu, John H. Reif, Designs for Autonomous Unidirectional Walking DNA Devices, Tenth International Meeting on DNA Based Computers (DNA10), Milano, Italy, June 7-10, 2004. Lecture Notes in Computer Science (Edited by C Ferretti, G. Mauri and C. Zandron), Vol. 3384, Springer-Verlag, New York, (2005), pp. 410-425.

[Y06] Peng Yin, Andrew J. Turberfield, John H. Reif, Design of Autonomous DNA Cellular Automata, Eleventh International Meeting on DNA Based Computers (DNA11), London, Ontario, Canada (June, 2005). Springer-Verlag Lecture Notes for Computer Science (LNCS), NYC, NY, (edited by Alessandra Carbone and Niles Pierce), Vol. 3892, Springer-Verlag, New York, (2006), pp. 399-416.

[S06] Ehud Shapiro and Yaakov Benenson, Bringing DNA Computers to Life. *Scientific American*, 45-51 (May 2006).

[S04] Nadrian C. Seeman, Nanotechnology and the Double Helix; *Scientific American*, 290 (6), 64-75 (June 2004).

[D06] Zhaoxiang Deng, Yi Chen, Ye Tian, Chengde Mao, A fresh look at DNA nanotechnology, in "Nanotechnology: Science and Computation", Springer Verlag series in Natural Computing (eds. J. Chen; N. Jonoska & G. Rozenberg), pp 23~34, (2006).

[C06] Leandro Nunes de Castro, *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*, Published by Chapman & Hall/CRC Computer and Information Sciences, (2006).

[S05] Ruojie Sha, Xiaoping Zhang, Shiping Liao, Pamela E. Constantinou, Baoquan Ding, Tong Wang, Alejandra V. Garibotti, Hong Zhong, Lisa B. Israel, Xing Wang, Gang Wu, Banani Chakraborty, Junghuei Chen, Yuwen Zhang, Hao Yan, Zhiyong Shen, Wanqiu Shen, Phiset Sa-Ardyen, Jens Kopatsch, Jiwen Zheng, Philip S. Lukeman, William B. Sherman, Chengde Mao¹, Natasha Jonoska², and Nadrian C. Seeman, *Structural DNA Nanotechnology: Molecular Construction and Computation*, C.S. Calude et al. (Eds.):UC 2005, LNCS 3699, Springer-Verlag Berlin Heidelberg, pp. 20–31, (2005).

[Y07] Hao Yan and Yan Liu, *DNA Nanotechnology: an Evolving Field*, Invited Chapter in *Nanotechnology: Science and Computation*, edited by Junghuei Chen, Nataša Jonoska, Grzegorz Rozenberg, Natural Computing Series, Published by Springer, Berlin Heidelberg, pp. 35-53, (2006).

[P06] S-H. Park, M.W. Prior, T.H. LaBean, and G. Finkelstein, *Optimized fabrication and electrical analysis of silver nanowires templated on DNA molecules*. *Applied Physics Letters* 89, 033901, (2006).

[D07] Douglas, S. M., Chou, J. J. & Shih, W. M. *DNA-nanotube-induced alignment of membrane proteins for NMR structure determination*. *PNAS* 104, 6644–6648, (2007).

[L07] Thom LaBean and Hanying Li, *Using DNA for Construction of Novel Materials*, *Nano Today*, 2, 26-35, (2007).