Self-assembled DNA Nanostructures and DNA Devices

John Reif

Department of Computer Science
Duke University
Durham, NC 27707
reif@cs.duke.edu

Harish Chandran

Department of Computer Science
Duke University
Durham, NC 27707
harish@cs.duke.edu

Nikhil Gopalkrishnan

Department of Computer Science
Duke University
Durham, NC 27707
nikhil@cs.duke.edu

Thomas LaBean

Department of Computer Science
and Department of Chemistry
Duke University
Durham, NC 27707
thl@cs.duke.edu

Abstract

This chapter overviews the past and current state of the emerging research area in the field of nanoscience that make use of synthetic DNA to self-assemble into DNA nanostructures and to make operational molecular-scale 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 and complexity. We discuss the design and demonstration of molecular-scale devices that make use of DNA nanostructures to achieve: molecular patterning, molecular computation, amplified sensing and nanoscale transport. We particularly emphasize molecular devices that make use of techniques that seem most promising, namely ones that are *programmable* (the tasks executed can be modified without entirely redesigning the nanostructure) and *autonomous* (executing steps with no external mediation after starting).

1. Introduction

1.1 Some unique advantages of DNA nanostructures

The particular molecular-scale constructs 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.

1.2 Use of 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 challenge is the need for robust, error-free methods for self-assembly of complex devices out of large number of molecular components. This requires novel approaches. For example, the micro-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.

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 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.

1.3 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.

1.4 The interdisciplinary nature of the field

DNA self-assembly is highly interdisciplinary and uses techniques from multiple disciplines such as biochemistry, physics, chemistry, material science, computer science and mathematics. While this makes the topic quite intellectually exciting, it also makes it challenging for a typical reader.

1.5 The rapid progress of complexity of DNA nanostructures

The complexity of experimental demonstrations of DNA nanostructures has increased 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. Other surveys are given by Seeman (2004), Deng et al. (2006) and Amin et al. (2009).

1.6 Programmable DNA nanostructures and devices

We particularly emphasize molecular assemblies that are: a*utonomous*: executing steps with no exterior mediation after starting, and *programmable*: the tasks executed can be modified without entirely redesigning the nanostructure. In many cases, the self-assembly processes are programmable in ways analogous to more conventional computational processes. 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.

1.7 Applications of DNA nanostructures

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 (v) transduction of molecular sensing to provide drug delivery (vi) vehicles for drug delivery inside cells, and (vii) protein structure determination. 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.

1.8 Organization: Topics discussed in this article

Section 2 gives a brief introduction to DNA, some known enzymes used for manipulation of DNA nanostructures and some reasons why DNA is uniquely suited for assembly of molecular-scale devices. Section 3 narrates the first experimental demonstration of autonomous biomolecular computation and its shortcomings. Section 4 describes common DNA motifs, DNA tiles, DNA lattices composed of assemblies of these tiles and software for tile design. Section 5 describes autonomous finite state computations using linear DNA nanostructures. Section 6 discusses various methods for assembling patterned and addressable 2D DNA nanostructures and algorithmic self-assembly. Section 7 overviews methods for error correction and self-repair of DNA tiling assemblies. Section 8 covers 3D DNA nanostructures, including wireframe polyhedra, 3D DNA lattices and 3D DNA origami. Section 9 reviews protocols for detection of molecular targets (DNA, RNA) and its application to autonomous molecular computation. Section 10 describes autonomous molecular transport devices self-assembled from DNA. Section 11 makes concluding remarks and sets out future challenges for the field.

2 Introducing DNA, it's structure and its manipulation

2.1 Introducing DNA

DNA self-assembly research is highly interdisciplinary and uses techniques from biochemistry, physics, chemistry, material science, computer science and mathematics. A reader having no training in biochemistry must obtain a coherent understanding of the topic from a short chapter. This section is written with the expectation that the reader has little background knowledge of chemistry or biochemistry. One 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

Single stranded DNA (ssDNA) is a long polymer made from repeating units called *nucleotides*. The nucleotide repeats contain both the segment of the *backbone* of the molecule, which holds the chain together, and a *base*. A base linked to a sugar is called a *nucleoside* and a base linked to a sugar and one or more phosphate groups is called a *nucleotide*. The backbone of the DNA strand is made from alternating phosphate and sugar residues. The sugar in DNA is 2-deoxyribose, which is a pentose (five-carbon) sugar. The sugars are joined together by phosphate groups that form phosphodiester bonds between the third and fifth carbon atoms of adjacent sugar rings. These asymmetric bonds mean a strand of DNA has a direction. The asymmetric ends of DNA strands are called the *5-prime* and *3-prime* ends, with the 5-prime end having a terminal phosphate group and the 3-prime end a terminal hydroxyl group. The four bases found in DNA are adenine (abbreviated A), cytosine (C), guanine (G) and thymine (T). These bases form the alphabet of DNA; the specific sequence comprises DNA's information content. Each base is attached to a sugar/phosphate to form a complete nucleotide. These bases are classified into two types; adenine and guanine are fused five-membered and six-membered heterocyclic compounds called *purines*, while cytosine and thymine are six-membered rings called *pyrimidines*. Each type of base on one strand overwhelmingly prefers a bond with just one type of base on the other strand. This is called *complementary base pairing*. Here, purines form hydrogen bonds to pyrimidines, with A bonding preferentially to T, and C bonding preferentially to G. This arrangement of two nucleotides binding together across the double helix is called a *base pair*. In living organisms, DNA does not usually exist as a single molecule, but instead as a pair of molecules called *double stranded DNA* (dsDNA) that are held tightly together via a reaction known as *DNA hybridization*. These two long strands entwine like vines, in the shape of a double helix. DNA hybridization occurs in a physiologic-like buffer solution with appropriate temperature, pH, and salinity.

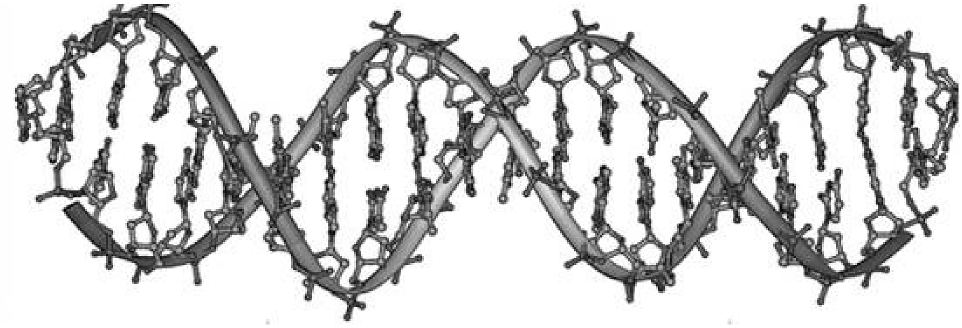


Figure 1: Structure of a DNA double helix. *Image by Michael Ströck and released under the GNU Free Documentation License*

In a double helix the direction of the nucleotides in one strand is opposite to their direction in the other strand: the strands are *antiparallel*. The DNA double helix is stabilized by hydrogen bonds between the bases attached to the two strands and stacking between contiguous base pairs. As hydrogen bonds are not covalent, they can be broken and rejoined relatively easily. The two strands of DNA in a double helix can therefore be pulled apart like a zipper, either by a mechanical force or high temperature. The two types of base pairs form different number of hydrogen bonds, AT forming two hydrogen bonds, and GC forming three hydrogen bonds. The association strength of hybridization depends on the sequence of complementary bases, stability increasing with length of DNA sequence and GC content. This association strength can be approximated by 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. Single stranded DNA is flexible and has a small persistence length when compared to double stranded DNA of comparable length. Single stranded DNA is sometimes thought of as a freely-jointed chain while double stranded DNA is more like a worm-like chain. 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 the helical axis. The width of the DNA double helix is 2.2 to 2.6 nanometers and the helical pitch is about 3.4 nanometers. 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*, is the displacement of a single strand of DNA from a double helix by an incoming strand with a longer complementary region to the template strand. The incoming strand has a *toehold*, an empty single stranded region on the template strand complementary to a subsequence of the incoming strand, to which it binds initially. It eventually displaces the outgoing strand via a kinetic process modeled as a one dimensional random walk. Strand displacement is a key process in many of the DNA protocols for running DNA autonomous devices. Figure 2 illustrates DNA strand displacement via branch migration.

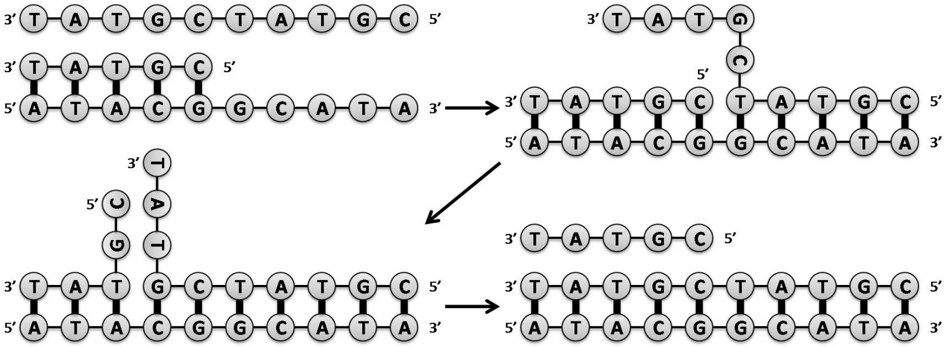
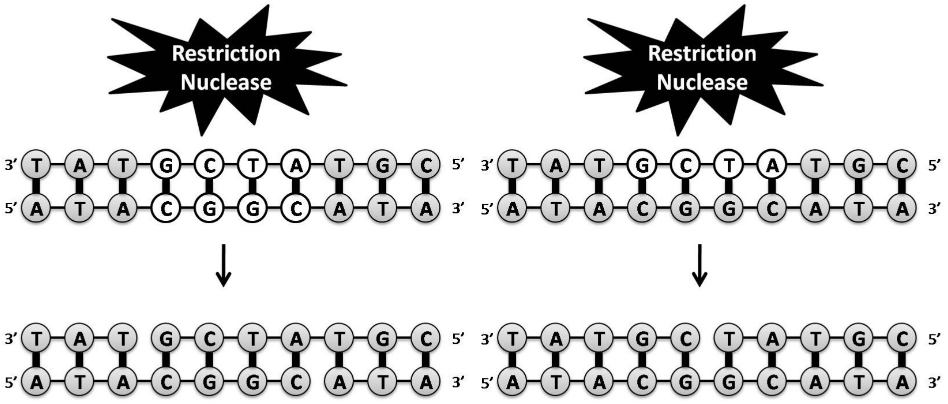

3′ T A T G C T A T G C 5′
3′ T A T G C 5′
5′ A T A C G G C A T A 3′
3′ T A T G
C
5′
3′ T A T G C T A T G C 5′
5′ A T A C G G C A T A 3′
3′ T A T
5′ C G
3′ T A T G C T A T G C 5′
5′ A T A C G G C A T A 3′
3′ T A T G C 5′
3′ T A T G C T A T G C 5′
5′ A T A C G G C A T A 3′

Figure 2: Strand displacement of dsDNA via a branch migration hybridization reaction: *Figure illustrates DNA strand displacement of a DNA strand induced by the hybridization of a longer strand, allowing the structure to reach a lower energy state.*

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.


Restriction Nuclease
Restriction Nuclease
3' T A T G C T A T G C 5'
5' A T A C G G C A T A 3'
3' T A T G C T A T G C 5'
5' A T A C G G C A T A 3'

Cutting

Nicking

Figure 3: Example of restriction enzyme cuts of a single stranded DNA sequence. *The subsequence recognized by the nuclease is unshaded*

DNA restriction (see figure 3) is the cleaving of phosphodiester bonds between the nucleotide subunits at specific locations determined by short (4-8 base) sequences by a class of enzymes called *nucleases*. *Endonucleases* cleave the phosphodiester bond within a polynucleotide chain while *exonucleases* cleave the phosphodiester bond at the end of a polynucleotide chain. Some nucleases have both these abilities. Some restriction enzymes cut both the strands of a DNA double helix while others cut only one of the strands (called *nicking*). *DNA ligation* (see figure 4) is the rejoining of nicked double stranded DNA by repairing the phosphodiester bond between nucleotides by the class of enzymes known as *ligases*.

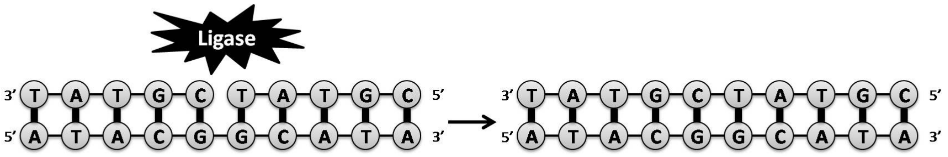

Ligase
3' T A T G C T A T G C 5'
5' A T A C G G C A T A 3'
3' T A T G C T A T G C 5'
5' A T A C G G C A T A 3'

Figure 4: Ligase healing a single stranded nick. *Note that the two parts are bound to the same template*

DNA polymerases (see figure 5) are a class of enzymes that catalyze the polymerization of nucleoside triphosphates into a DNA strand. The polymerase “reads” an intact DNA strand as a template and uses it to synthesize the new strand. The newly polymerized molecule is complementary to the template strand. DNA polymerases can only add a nucleotide onto a pre-existing 3-prime hydroxyl group. Therefore it needs a *primer*, a DNA strand attached to the template strand, to which it can add the first nucleotide. Certain polymerase enzymes (e.g., phi-29) can, as a side effect of their polymerization reaction, efficiently displace previously hybridized strands. Isothermal denaturization (breaking of base pairings) can also be achieved by *helicases* which are motor proteins that move directionally along a DNA backbone, denaturing the double helix. In addition, *Deoxyribozymes (DNAzymes)* are a 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 see Stojanovic and Stefanovic (2003) for an example.

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. Ligation and polymerization require the expenditure of energy via consumption of ATP molecules, and thus can be controlled by ATP concentration.

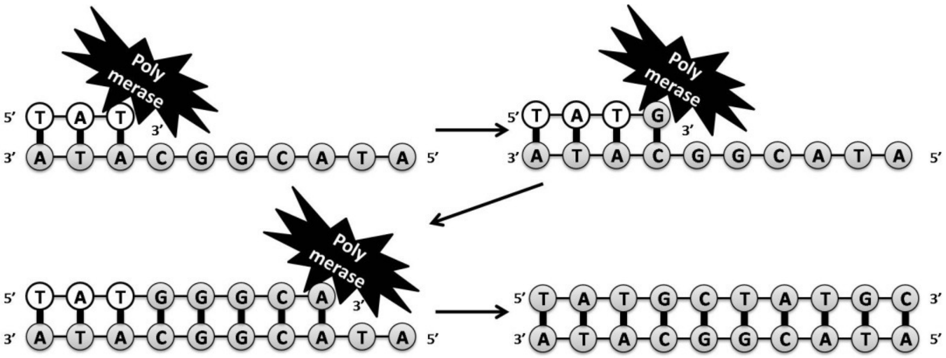

Poly merase
5′ T A T 3′
3′ A T A C G G C A T A 5′
Poly merase
5′ T A T G 3′
3′ A T A C G G C A T A 5′
Poly merase
5′ T A T G G G C A 3′
3′ A T A C G G C A T A 5′
5′ T A T G C T A T G C 3′
3′ A T A C G G C A T A 5′

Figure 5: Extension of primer strand (unshaded) bound to the Template by DNA polymerase.

2.4 Why use DNA to assemble molecular-scale devices?

There are many advantages of DNA as a material for building things at the molecular-scale. Below we list some reasons why DNA is uniquely suited for assembly of molecular-scale devices.

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

- A variety of geometries can be achieved by carefully programming DNA sequences to interact among themselves in a predictable manner. The shape of the DNA nanostructure is controlled by its component DNA strands and this gives us an ability to program a myriad of nanostructures.
- 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}) of 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), Transmission Electron Microscopy (TEM) and Cryo-Electron Microscopy (cyroEM) provide images of the actual assembled DNA nanostructures on 2D surfaces and in 3D.

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 (Adleman 1994; Adleman 1998). 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 annealed, they self-assembled into dsDNA. Each of these DNA nanostructures was a linear DNA double 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 Adleman'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 external 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 Nadrian Seeman (Robinson & Seeman 1987). Particularly useful types of motifs often found in DNA nanostructures include:

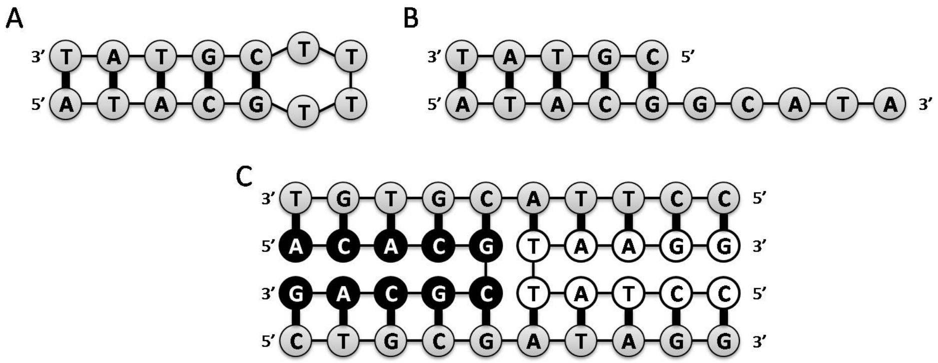

A
3′ T A T G C T T
5′ A T A C G T T
B
3′ T A T G C 5′
5′ A T A C G G C A T A 3′
C
3′ T G T G C A T T C C 5′
5′ A C A C G T A A G G 3′
3′ G A C G C T A T C C 5′
5′ C T G C G A T A G G 3′

Figure 6: Common DNA motifs.

- *Stem-loop* (also called hairpins) (see figure 6A): a ssDNA that loops back to hybridize on itself, i.e., one segment of the ssDNA (near the 5-prime end) hybridizes with another segment further along (nearer the 3-prime end) on the same ssDNA strand. The stem loop in fig. 4.1A has an unpaired region (with sequence TTTT) which is typical for this motif. Stem-loops are often used to form patterning on DNA nanostructures.
- S*ticky end* (see figure 6B): an unhybridized ssDNA that protrudes from the end of a double helix. The sticky end shown (GCATA) protrudes from dsDNA (ATACG on the bottom strand). Sticky ends are often used to combine two DNA nanostructures together via hybridization of their complementary ssDNA.
- *Holliday junction* (see figure 6C): two parallel DNA helices form a junction with one strand of each DNA helix crossing over to the other DNA helix. Holliday junctions are often used to hold 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 (1961), which culminated in a proof by Berger (1966), and later Robinson (1971), that universal computation could be done via tiling assemblies. Winfree (1995) 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, the DNA tiles assemble first, and only then, as the solution is further cooled, do the tiles bind together via hybridization of their pads.

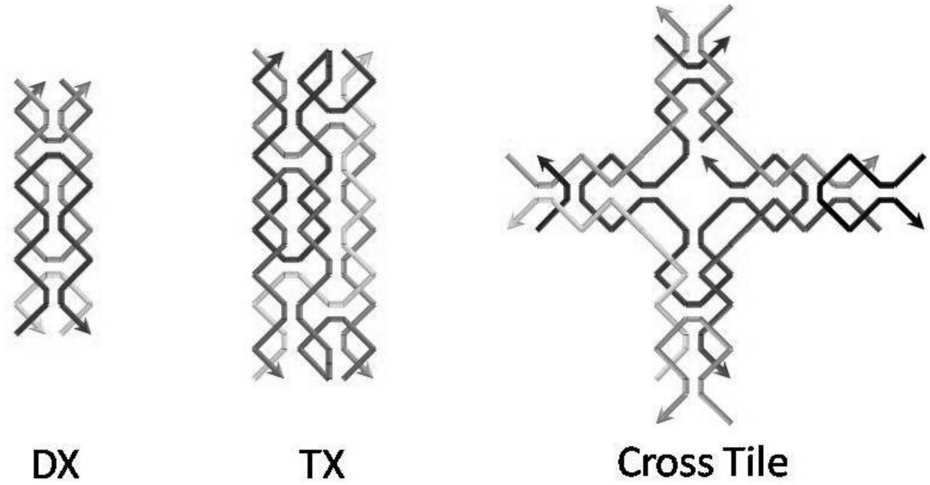


Figure 7: DNA Tiles: DX, TX and the Cross tile

Figure 7 illustrates some principal DNA tiles. Also see LaBean, Gothelf and Reif (2007). Winfree, Yang and Seeman (1996) developed a family of DNA tiles known collectively as DX tiles (see figure 7) 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 (see figure 8a).

Subsequently, other DNA tiles were developed by LaBean et al. (2000) to provide for more complex strand topology and interconnections, including a family of DNA tiles known as *TX tiles* (see figure 7) 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 c*ross tiles* developed by Yan et al. (2003*c*) (see figure 7) 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.

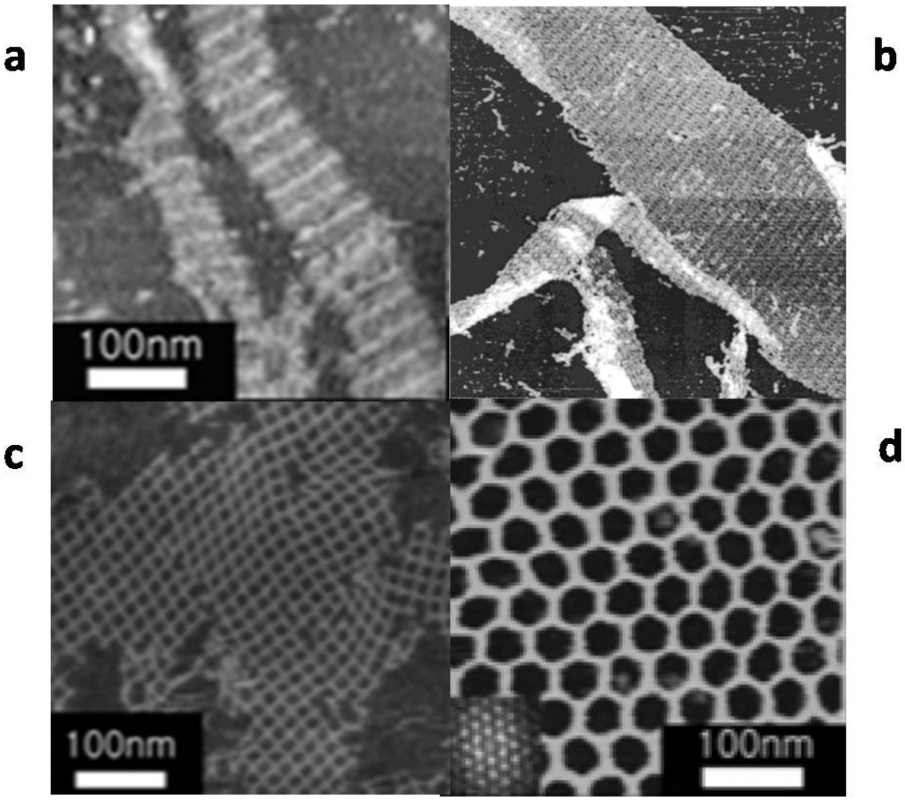
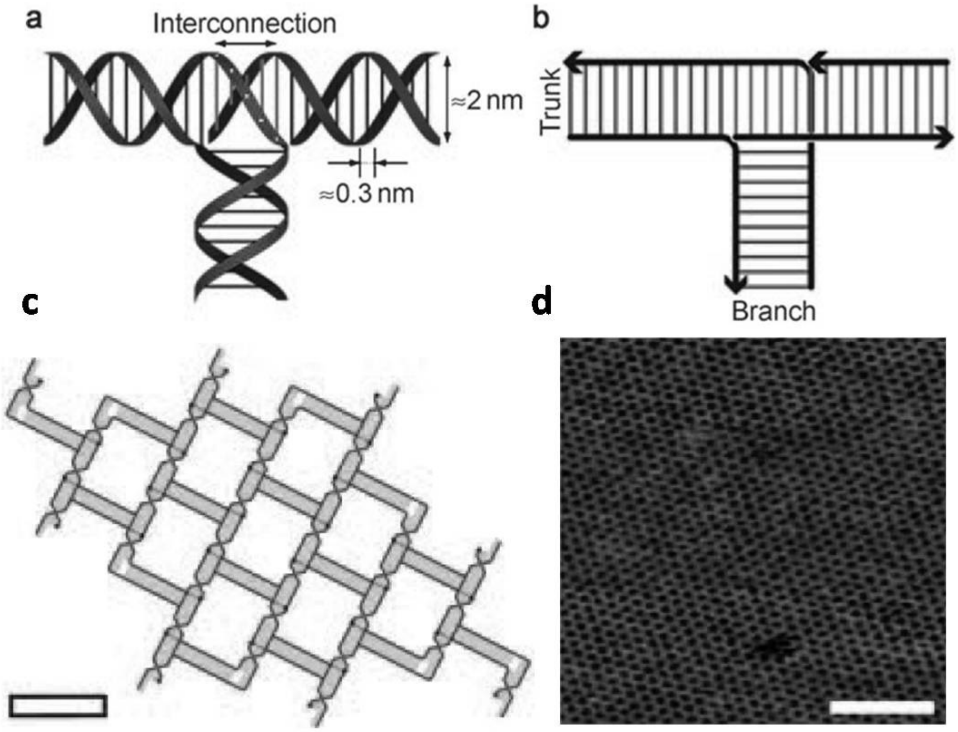


Figure 8: DNA lattices: a) DX lattice, b) TX ribbons c) Cross tile lattice and d) 3 point star hexagonal lattice

Subsequently, large hexagonal 2D DNA lattices were achieved by He et al. (2005*a*) using a *three point star* motif where each tile was connected to three neighbors in its plane. Figure 8 shows an AFM images of a 2D DNA lattices using the motifs described earlier. Recently, a tile in the shape of a T-junction (shown in figure 9) was used to assemble 2D lattices, 1D ladders and rings (Hamada & Murata 2009). These tiles are different from the tiles described earlier as they do not use Holliday junction.

The tiles described above are designed to be planar. But in reality they possess a small curvature, thus preventing large planar lattices. To counter this, a strategy called *corrugation* developed by Yan et al. (2003*c*) was introduced in which neighboring tiles are flipped with respect to each other, thus cancelling out their curvature. Another technique to minimize defects due to curvature and obtain large assemblies was *sequence symmetry* introduced by He et al. (2005*b*) in which geometrically symmetric parts of the tile are given the same sequence thus ensuring symmetric curvature.



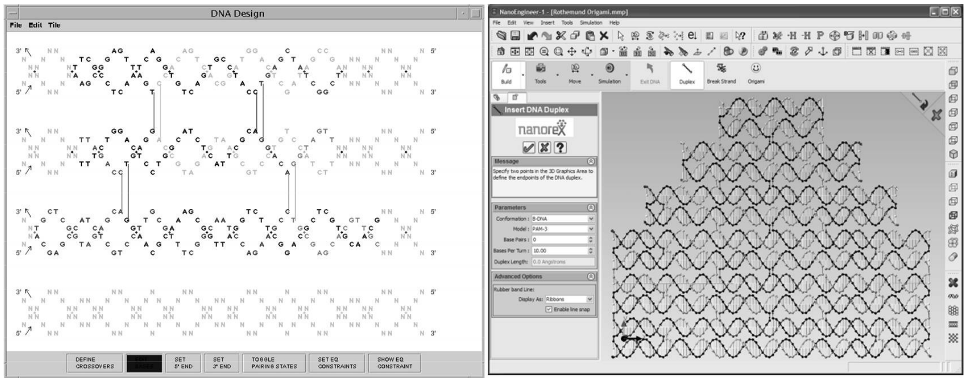
Scale bars: bottom left: 10nm, bottom right: 100nm

Figure 9: T-junction tiling a) and b) Design of the T-junction c) Lattice schematics D) AFM image of the lattice

To program a tiling assembly, the DNA sequence of the pads 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, developed jointly by Duke and Caltech, which provides a graphically-interfaced sequence optimization system for designing DNA secondary structures (Yin et al. 2004*a*). A more recent commercial product, NanoEngineer, with enhanced capabilities for DNA design and a more sophisticated graphic interface, was developed by Nanorex, Inc.



Specify two points in the 3D Graphics Area to define the endpoints of the DNA duplex.

Conformation : B-DNA
Model : PAM-3
Base Pairs : 0
Bases Per Turn : 10.00
Duplex Length: 0.0 Angstroms

Rubber band Line:
Display As: Ribbons
☑ Enable line snap

Tile Soft

Nanoengineer

Figure 10: TileSoft: sequence optimization software for designing DNA secondary structures and nanoengineer

5 Autonomous finite state computation using linear DNA nanostructures

5.1 Demonstration of autonomous computations using self-assembly of DNA nanostructures

The first experimental demonstrations of computation using DNA tile assembly were done in 1999 (LaBean, Winfree & Reif 1999; LaBean et al. 2000; Mao et al. 2000; Yan et al. 2003*a*). Among the experiments Mao et al. (2000) demonstrated 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 ith 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. This experiment is illustrated in figure 11.

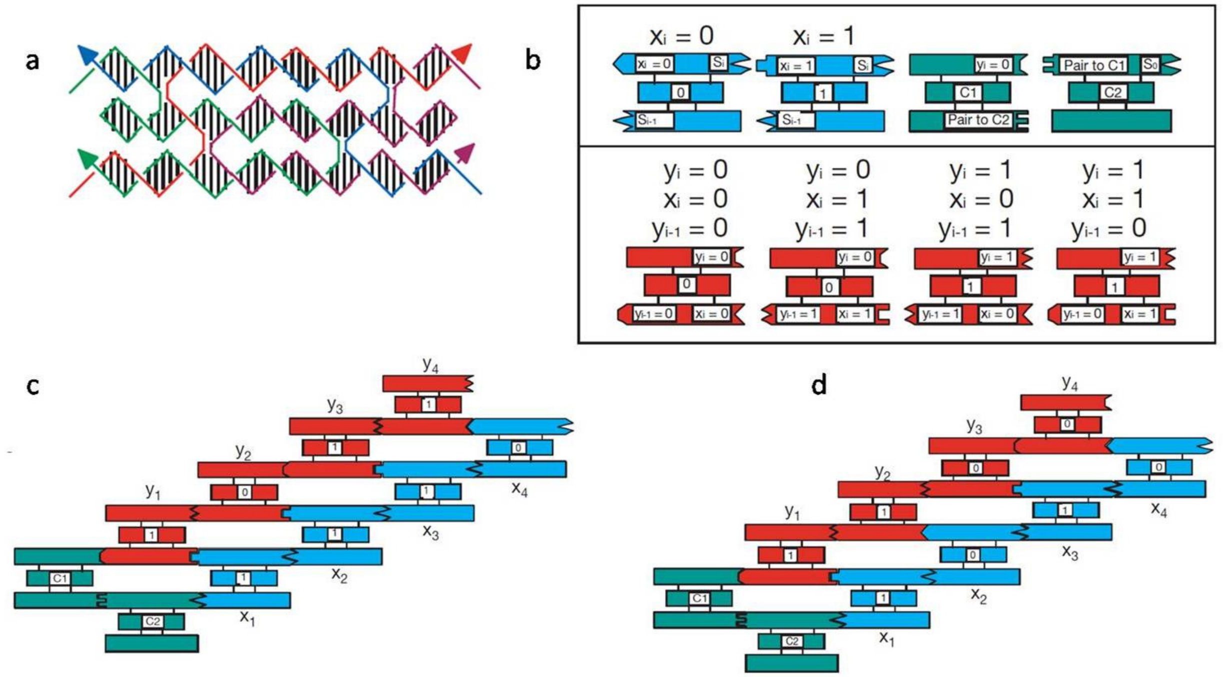


Figure 11: Sequential Boolean computation via a linear DNA tiling assembly a) TX tile used in assembly. b) Set of TX tiles providing logical programming for computation. c), d) example resulting computational tilings.

These experiments 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 11.

Figure 11 shows (a) a unit TX tile and the sets of input and (b) output tiles with geometric shapes conveying sticky-end complementary matching. The tiles of (b) execute binary computations depending on 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 assembles specifically starting from the bottom left using the inputs from the blue layer. See Mao et al. (2000) 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 though 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 would 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. This molecular patterning was clearly observable under appropriate AFM imaging conditions.

Although only very simple computations, these experiments 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

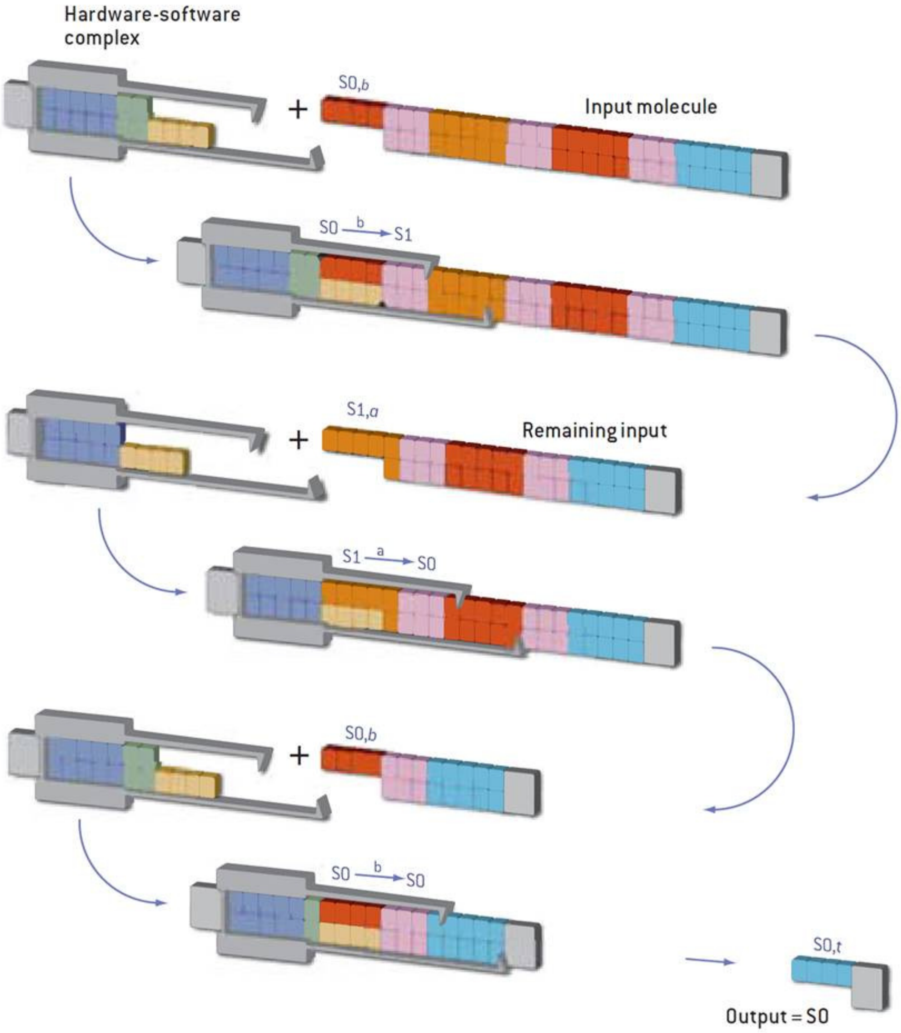


Figure 12: Autonomous finite-state computations via disassembly of a double-stranded DNA nanostructure

An alternative method for autonomous execution of a sequence of finite-state transitions was subsequently developed by Shapiro and Benenson (2006). 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 (see figure 12). 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 sticky end encoding the next state.

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. Adar et al. (2004) demonstrated in the test tube a potential application of such a finite-state computing device to medical diagnosis and therapeutics. See the conclusion section 11 for further discussion.

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 which we can attach to 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 organic materials like other (complementary) DNA, RNA, proteins, peptides etc. 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.

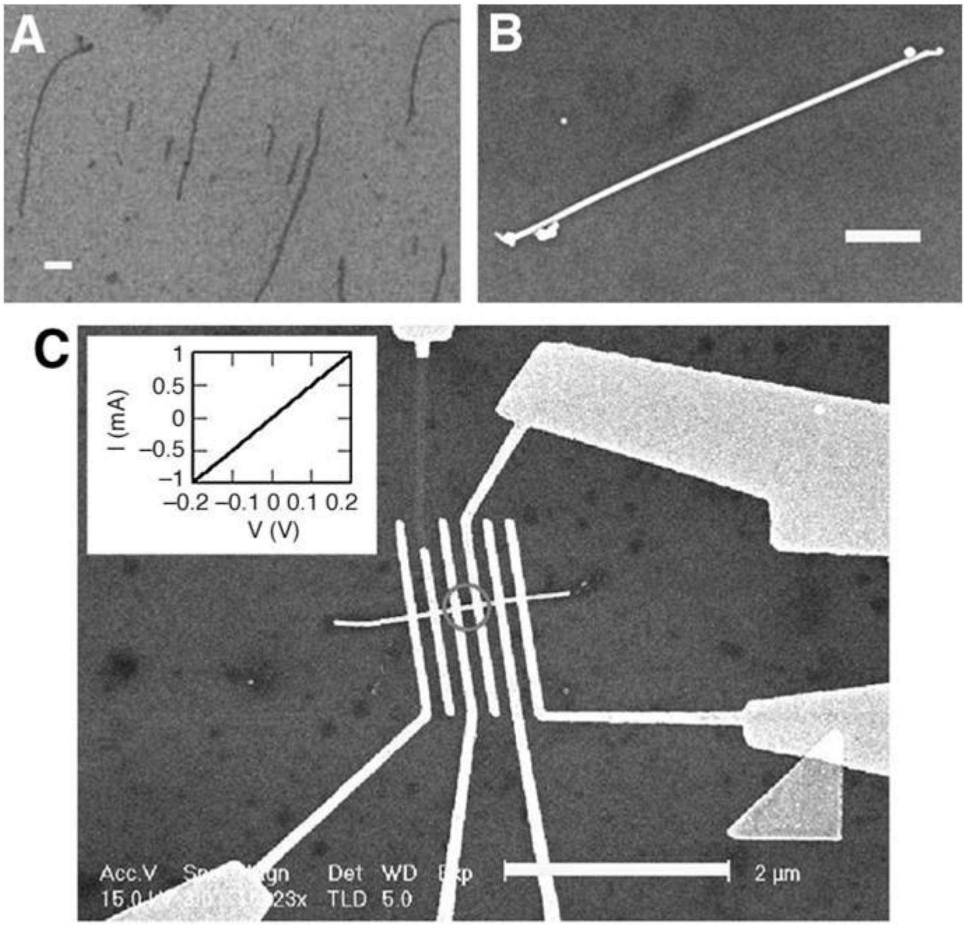


Figure 13: Conductive wires fabricated from self-assembled DNA tubes plated with silver. a) DNA tubes prior to plating. b) DNA tubes after silver plating c) SEM image of conductivity test on silicon oxide substrate.

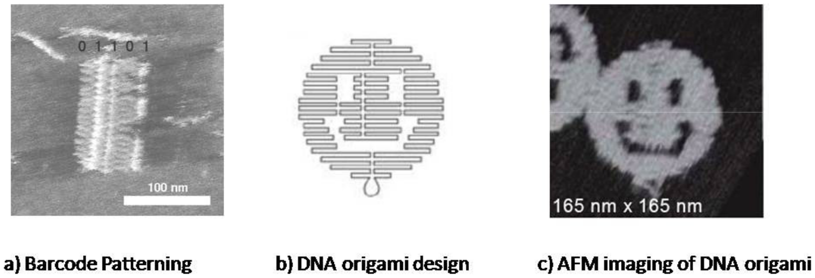
These technologies provide a molecular-scale method for attaching heterogeneous materials to DNA nanostructures. They can potentially be used for attaching molecular electronic devices to 2D or 3D DNA nanostructures. Yan et al. (2003*c*) and Park et al. (2006*b*) describes conductive wires fabricated from self-assembled DNA tubes plated with silver, as illustrated in figure 13.

6.2 Methods for programmable assembly of patterned 2D DNA lattices

The first experimental demonstration of 2D DNA lattices by Winfree et al. (1998) 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 have been experimentally demonstrated, as described in the next few subsections.

6.2.1 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 14: 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 by propagating 1D information from the scaffold into a second dimension to create AFM observable patterns (Yan et. al 2003*b*). The scaffold strand weaves through the resulting DNA lattice to form the desired distinct sequence of 2D barcode patterns (figure 14a). 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* (Rothemund 2006). This approach makes use of a long strand of "scaffold" ssDNA (such as from the genome of a viral phage) that has only weak secondary structure and few long repeated or self-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 fully addressable 2D DNA nanostructure. A schematic trace of the scaffold strand is shown in figure 14b, and an AFM image of the resulting assembled origami is shown in figure 14c. This method can be slightly modified to get patterning by extending staple strands at the end into a stem-loop structure. These stem-loops will stick out of the plane of the nanostructure and will appear as a bright dot on an AFM image (see figure 15). This landmark work of Rothemund (2006) 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.

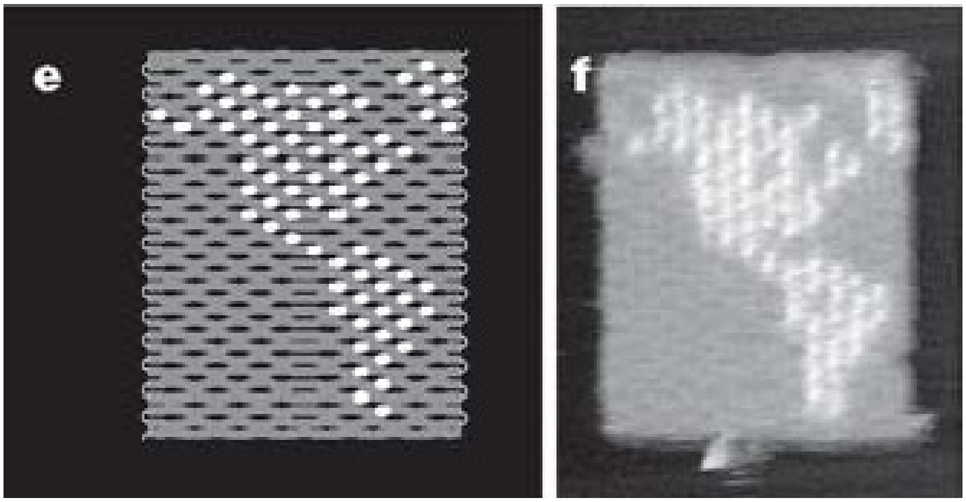


Figure 15: Patterned origami: *Bright dots are staples extended into a stem-loop structure, causing them to stick out of the plane*

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 (Barish, Rothemund & Winfree 2005; Fujibayashi et al. 2008). Figure 16 gives a modulo-2 version of Pascal's Triangle (known as the Sierpinski Triangle), where each tile determines and outputs to neighborhood pads the XOR of two of the tile pads (Rothemund, Papadakis & Winfree 2004). Example AFM images (scale bars = 100 nm) of the assembled structures are shown in the three panels of figure 16. Figure 17 gives Rothemund's and Winfree (2000) 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.

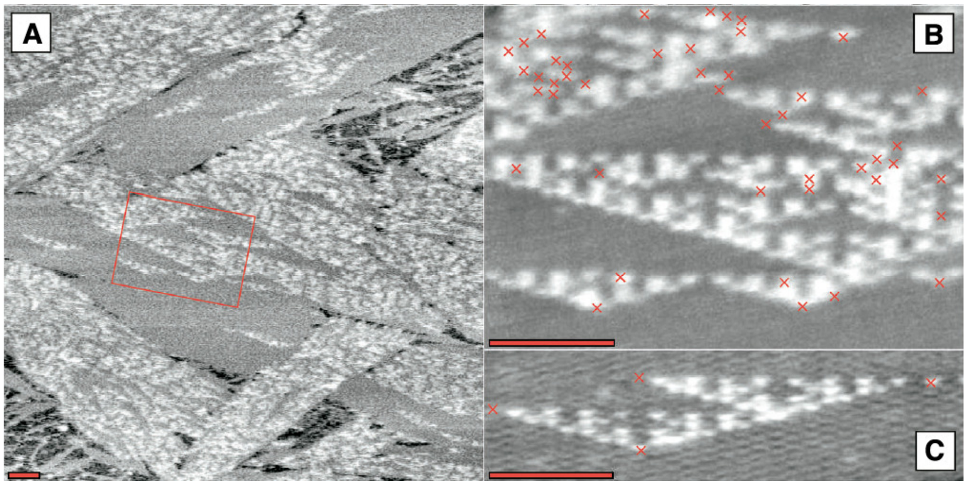


Figure 16: Programmable assembly of Sierpinski triangle by use of computational assembly *Scale bar = 100nm*

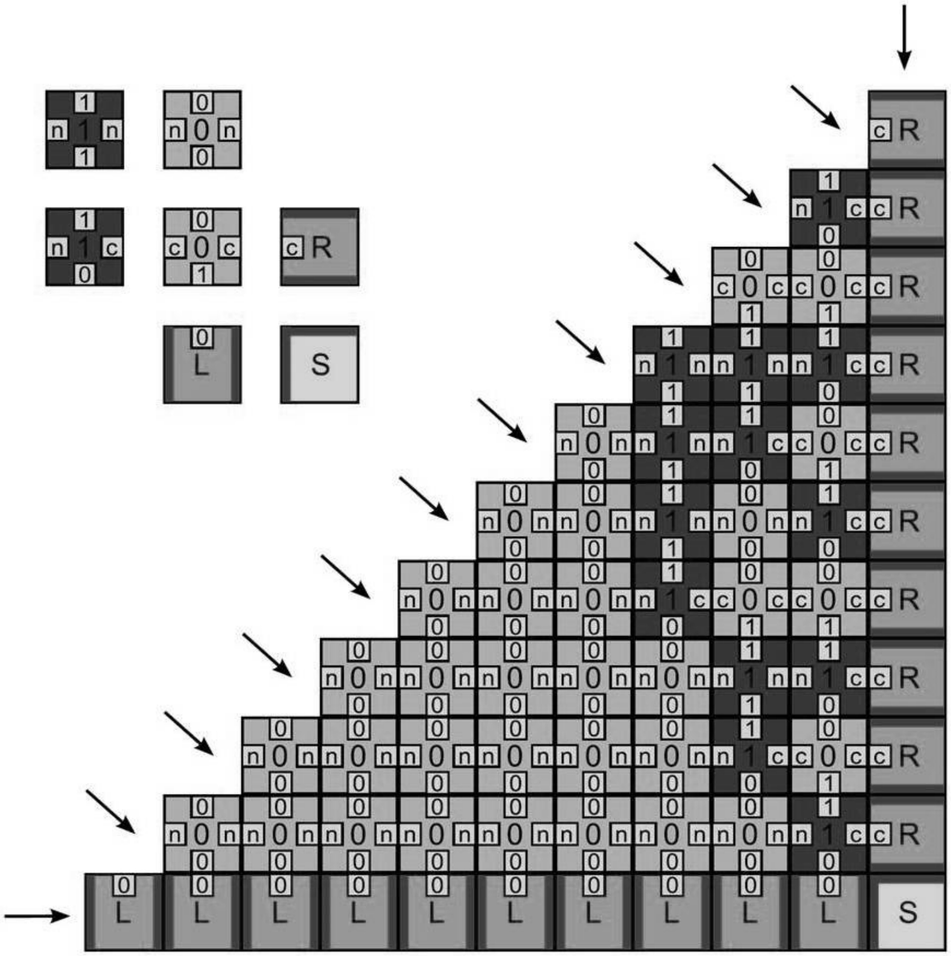


Figure 17: Rothemund's and Winfree's design for a self-assembled binary counter using tilings.

6.2.3 Programmable assembly of patterned 2D DNA lattices by hierarchical assembly

A further approach, known as hierarchical assembly, is to assemble DNA lattices in multiple stages (Park et al. 2006*a*). Figure 18 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.

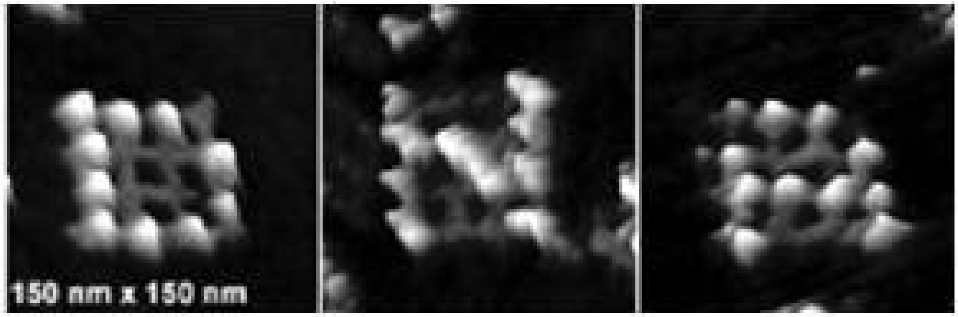


Figure 18: 2D Patterns by hierarchical assembly *AFM images of characters D, N, and A.*

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. In tile based self-assembly, there are three main kinds of errors:

- Nucleation error: Tile based nanostructures are grown from a special tile known as the *seed tile*. All nanostructures that grow out of non-seed tiles are erroneous assemblies.
- Growth error: Attachment of an incorrect tile instead of a better matched tile.
- Facet (roughening) error: Attachment of tiles along a facet (boundary) where no growth was intended to occur.

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 algorithmic 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 Proofreading schemes for error-resilient tilings

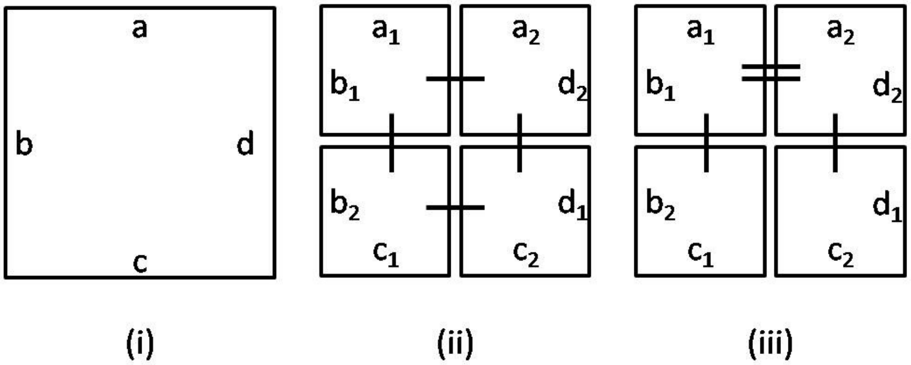


Figure 19: Proofreading schemes for error-resilient tilings. (i) Original tile (ii) Winfree et al. general 2x2 proofreading scheme (iii) Chen et al. general 2x2 snaked proofreading scheme. The lines represent pad strengths.

Winfree and Bekbolatov (2003) developed a “proofreading” method of replacing each tile with four tiles that provide sufficient redundancy to quadratically reduce errors, as illustrated in figure 19. Each tile is replaced by an array of 2x2 tiles that logically correspond to the original tile. The internal sides of the new block are given unique glues that are not present of any other tiles. Thus assembly proceeds like for the original tile set but, scaled up by a factor of 4 in area. When a mismatched tile is incorporated in this new tiling at some position, further assembly cannot proceed at that position without making an additional error. This gives mismatched tiles time to dissociate and thus the tiling is resilient to growth errors. Reif, Sahu and Yin (2004) proposed a more compact method for decreasing assembly errors, as illustrated in figure 20. 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. Chen and Goel (2004) proposed *snaked proofreading* (see figure 19) to correct facet errors in addition to growth errors. Both these techniques were experimentally tested by Chen et al. (2007). Nucleation errors were handled in Schulmann and Winfree (2009) by constructing tile sets that introduce arbitrarily large barriers to incorrect nucleation.

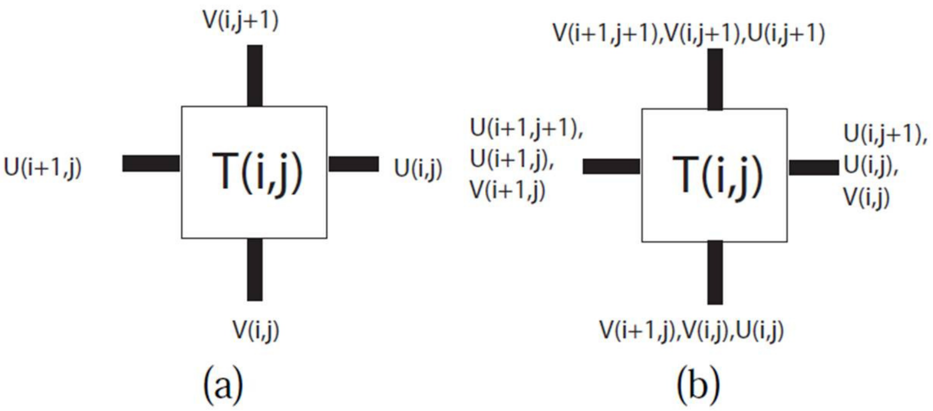


Figure 20: A compact scheme for error-resilient tilings. a) Original tile b) Error resilient tile.

By combining all the aforementioned techniques, it might be possible to design robust tile sets to perform tiling based computations. 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.

7.3 Activatable tiles for reducing errors

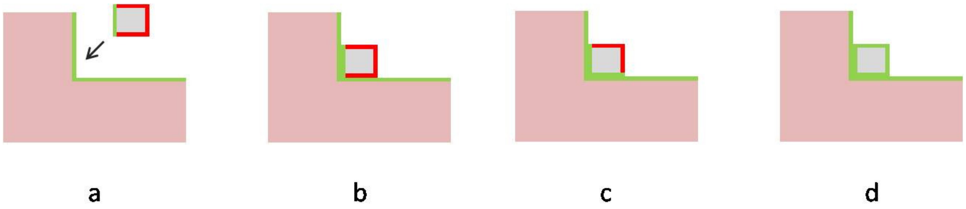


Figure 21: Activatable tiles. a) Partially formed assembly with 2 activated boundaries and a protected tile b) Protected tile binds to the boundary c) The other input pad is activated d) When both the inputs pads bind, the output pads are activated.

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, LaBean and Reif (2007) 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, some pads of the tile will be in an inactive state, where the tile is protected from unwanted binding with other tiles and thus preventing lattice grow in the (unwanted) reverse direction. After appropriate bindings and subsequent deprotections, the tile transitions to an active state, allowing further growth.

8 Three dimensional DNA nanostructures

8.1 Three dimensional DNA wireframe polyhedra

The first 3D wireframe object was obtained by Shih, Quispe and Joyce (2004) by folding a 1.7 kilobase single stranded DNA into nanoscale wireframe octahedron with the help of five 40 base synthetic DNA. The structure was imaged using cryo electron microscopy (see figure 22). This was followed soon after by Goodman, Berry and Turberfield (2004) who constructed a wireframe DNA regular tetrahedron from four 55 base ssDNA (see figure 22) in a single synthesis step. The structure was experimentally demonstrated to be structurally robust and the fabrication process was quick and simple. Another approach towards wireframe structures was demonstrated by He et al. (2008) when they used a three-point-star motif to hierarchically assembly tetrahedrons (4 three-point motifs), dodecahedra (20 three-point motifs) and buckyballs (60 three-point motifs) (see figure 23). Instead of many ssDNA of unique sequences, many copies of the same motif (three-point-star) assemble into different polyhedral structures depending on the flexibility of the arms and concentration of the motif.

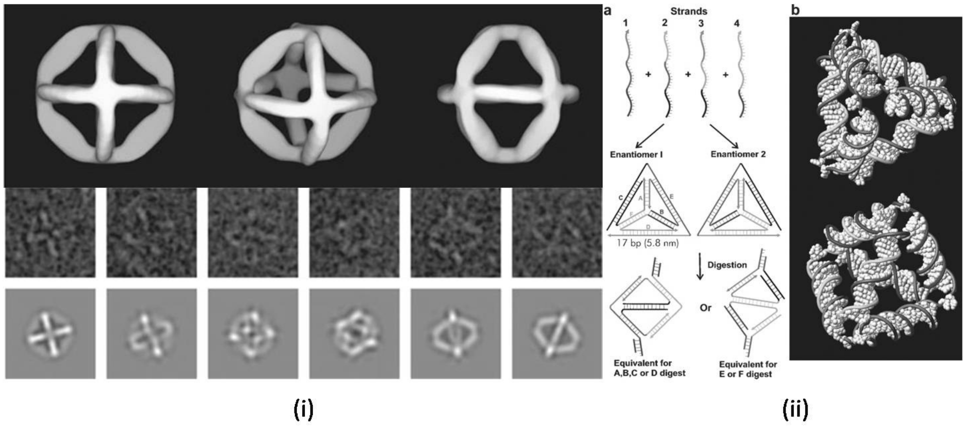


Figure 22: Wireframe polyhedra. i) Truncated octahedron ii) Tetrahedron

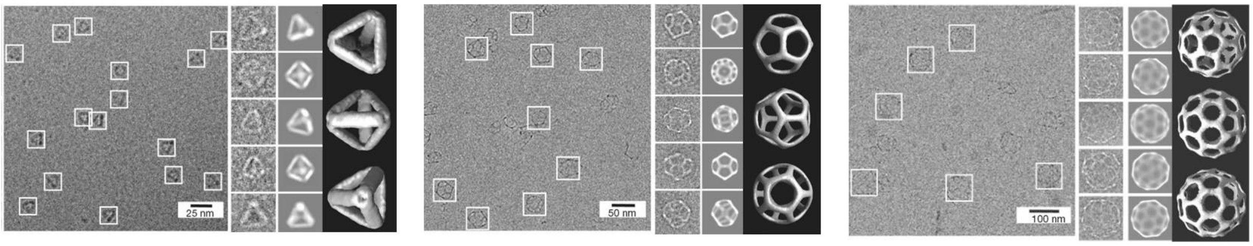


Figure 23: Creating various polyhedra using the 3 point motifs

8.2 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.

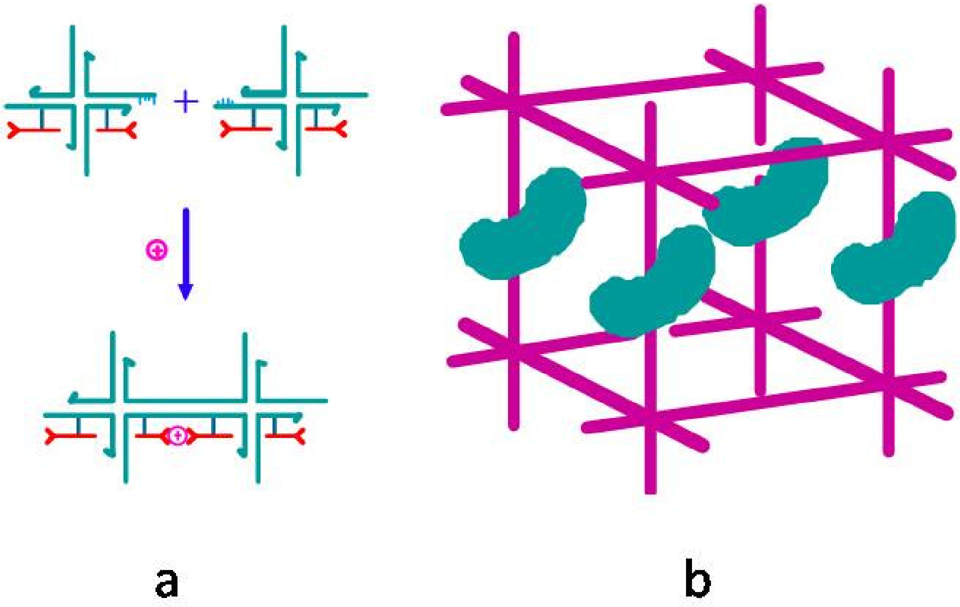


Figure 24: Scaffolding of a) 3D nanoelectronic architectures b) proteins into regular 3D arrays

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 24a shows DNA (cyan) and protein (red) organizing functional electronic structures.

It has been estimated that at least one half of all natural proteins cannot 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 24b). 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. The best effort thus far has been has been achieved by Zheng et al. (2009) through the tensegrity triangle which is a rigid DNA motif with three helical arms oriented along three linearly independent axes (see figure 25). Rhombohedral crystals of 4 Angstrom resolution were obtained.

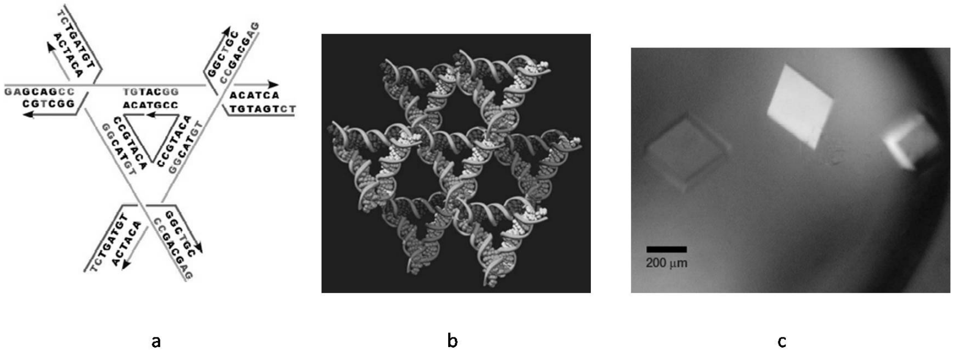


Figure 25: a) Schematics of the tensegrity tile b) Lattice structure c) Optical image of the 3D lattice

8.3 Three dimensional DNA origami

Rothemund's origami demonstrated arbitrary flat 2D nanostructures. Andersen et al. (2009) extended this technique to construct hollow containers (box) with walls of flat 2D origami. A cube like hollow box with a hinged lid that can be open and closed by a DNA strand as a key was constructed and imaged (see figure 26).

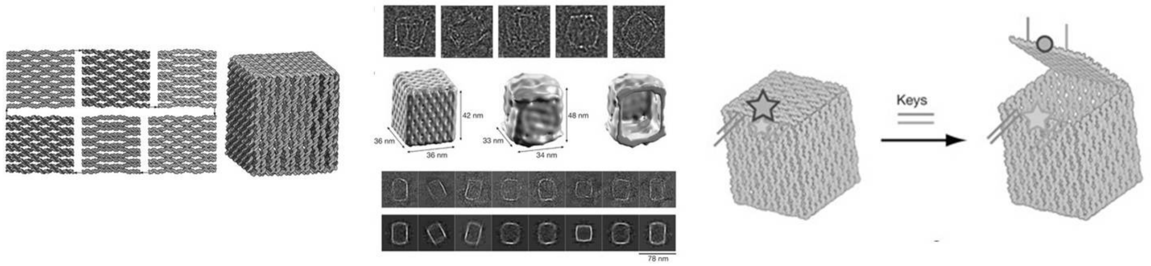


Figure 26: DNA box made by folding up planar origami.

DNA origami was extended to simple 3D cylindrical filaments that were used to partially orient membrane proteins in solution for structural studies employing NMR (Douglas, Chou & Shih 2007). In a new approach, Douglas et al. (2009) created stunning 3D origami by carving out 3D shapes from a honeycomb-like solid 3D structure (see figure 27). In addition, they provided design automation software, caDNAno (www.cadnano.org) that enables rapid prototyping of arbitrary 3D nanostructure with about 6nm resolutions. Dietz, Douglas and Shih. (2009) demonstrated the ability to bend and twist the honeycomb lattice by underwinding or overwinding the DNA double-helix (see figure 27). Honeycomb lattice based nanostructures have higher charge density and hence require longer annealing times than 2D DNA origami, carefully controlled salt concentrations and usually had lower yields than flat 2D DNA origami.

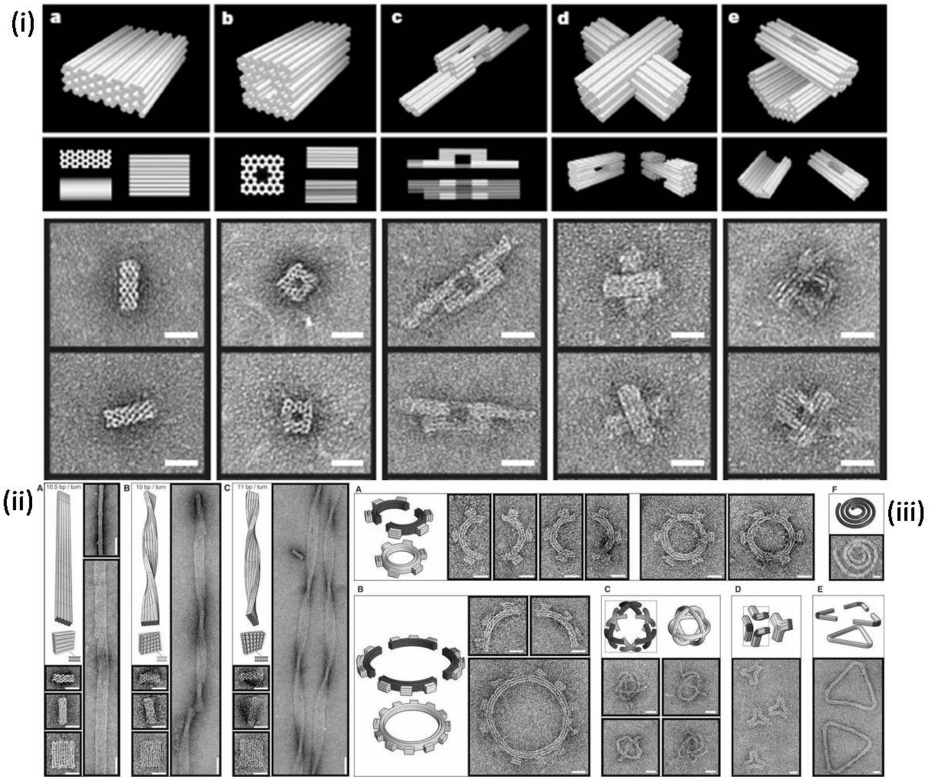


Figure 27: 3D DNA origami based on the honeycomb lattice i) Various 3D shapes ii) Twisting iii) Bending

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)*, which makes use of DNA polymerase to amplify a strand of DNA by repeated replication, using rounds of thermal-cycling (Saiki et al. 1985). (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

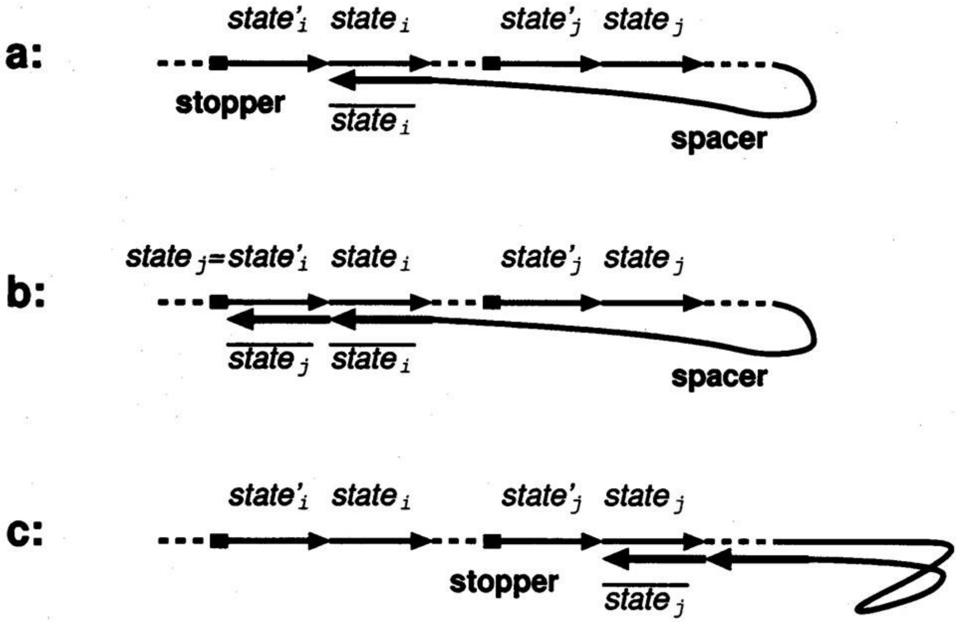


Figure 28: Whiplash PCR State transitions. The current state is annealed onto the transition table by forming a hairpin

structure (a). The current state is then extended by polymerase and the next state is copied from the transition table (b). After denaturation, the new current state is annealed to another part of the transition table to enable the next transition (c).

A method for DNA computation, known as *whiplash PCR*, introduced by Sakamoto et al. (1999), 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-prime end of the DNA strand has a final sequence encoding a state of the computation. A computation step is executed when this 3-prime end hybridizes to a portion of a "rule" subsequence, and the action of DNA polymerase extends the 3-prime end to a further subsequence encoding a new state.

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 executes autonomously – they require thermal cycling for each step of their protocols. Walker et al. (1992*a*; 1992*b*) 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 (2008) recently developed 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 proceed 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 for DNA detection

Dirks and Pierce (2004) 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 29.

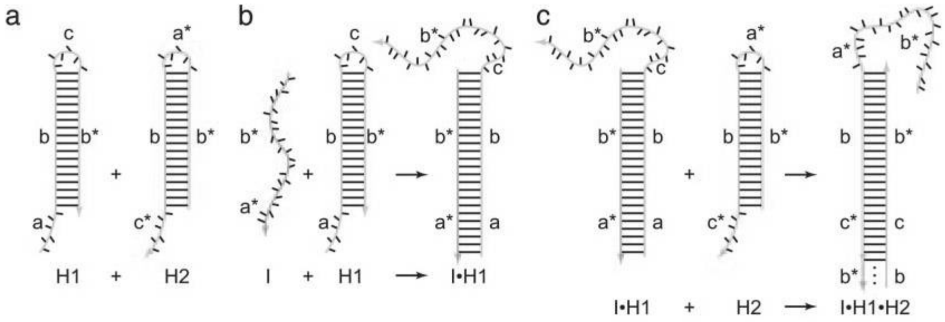


Figure 29: Autonomous Molecular Cascade for Signal Amplification

The protocol made use of multiple copies of two distinct DNA hairpins H1 and H2 that are initially added to a test tube. When ssDNA sequence I is added to the test tube, I initially has a hybridization reaction with subsequence ab of H1 via strand displacement, thus exposing c that had been previously hidden within the stemloop of H1. Next, cb* has a hybridization reaction with the subsequence c*b of H2, thus exposing a second copy of a* that had been previously hidden within the stemloop of H2. That other copy of a*b* then repeats the process with other similar (but so far unaltered) copies of H1 and H2, 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. Note that the response is linear in the concentration of strand I.

9.5 Hybridization reactions for autonomous DNA computation

Zhang et al. (2007) developed a general methodology for designing systems of DNA molecules by the use of catalytic reactions that are driven by entropy. In particular, it 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 of Dirks and Pierce (2004) and the molecular computations of Zhang et al. (2007) are their use only of hybridization, making no use of restriction enzyme or any other protein enzymes.

Following on this work, Yin et al. (2008) 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.

9.6 Autonomous detection protocols and molecular computations using DNAzyme

In addition, Tian, He and Mao (2006) demonstrated a novel method for DNA detection which involves amplification of the target strand via rolling circle amplification followed by the 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 colorimetric DNA detection at a limit of 1 picomolar. This led to the DNAzyme based autonomous DNA walker Tian et al. (2005) 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 et al. (2000) 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 (2003) 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. Sherman and Seeman (2004) demonstrated a DNA walker, 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 by Yin et al. (2004*c*). It employed restriction enzymes and ligase; see Yin et al. (2004*b*) for its detailed general design.

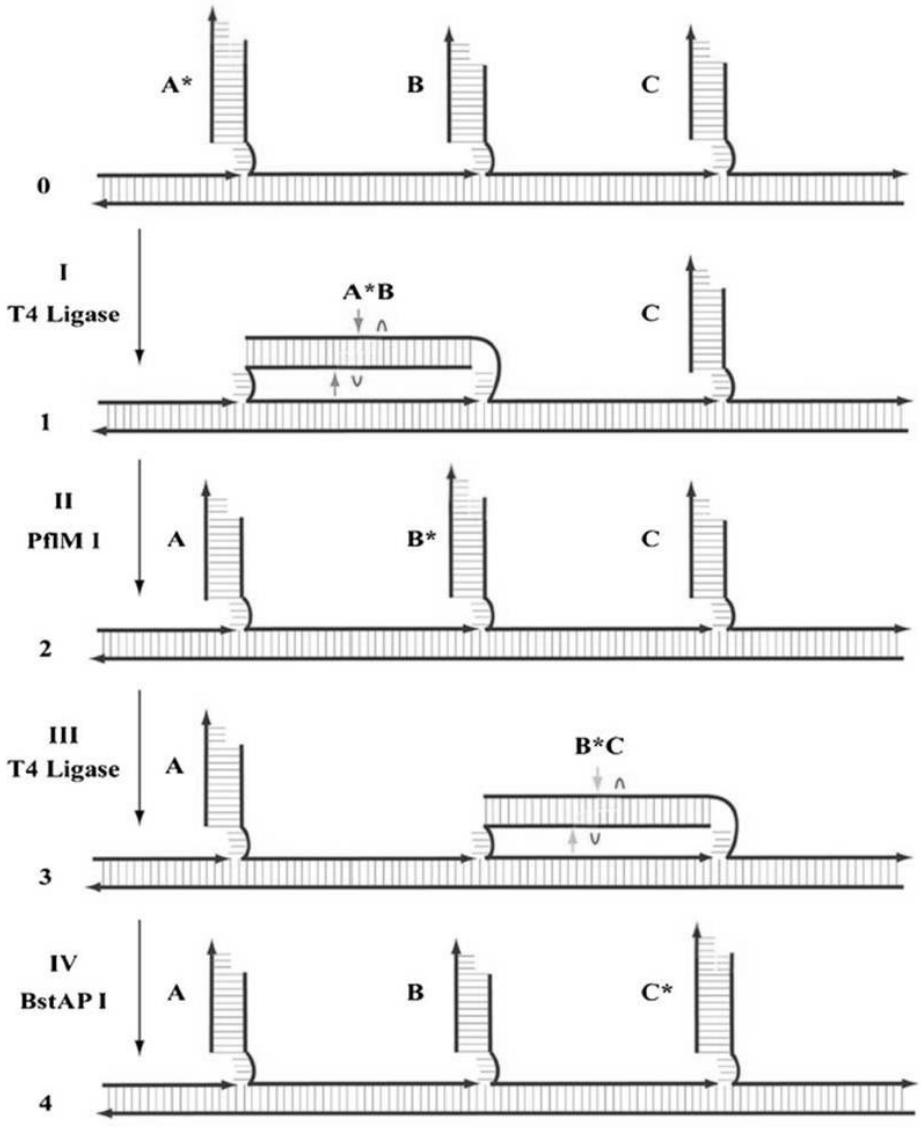


Figure 30: Autonomous Molecular Transport Devices Self-Assembled from DNA

The device is described in figure 30.

- 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 Tian et al. (2005) demonstrated an autonomous DNA walker that made use of a DNAzyme motor, designed by Chen, Wang and Mao (2004), which used the cuts provided by the enzymatic activity of DNAzyme to progress along a DNA nanostructure.

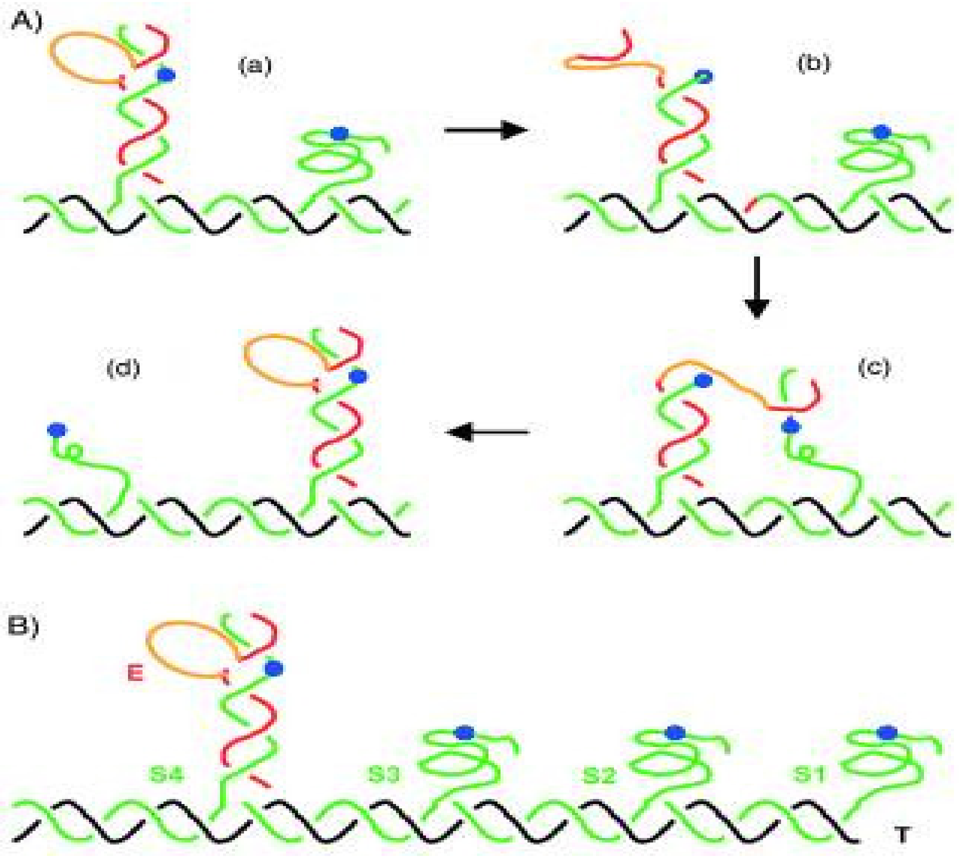
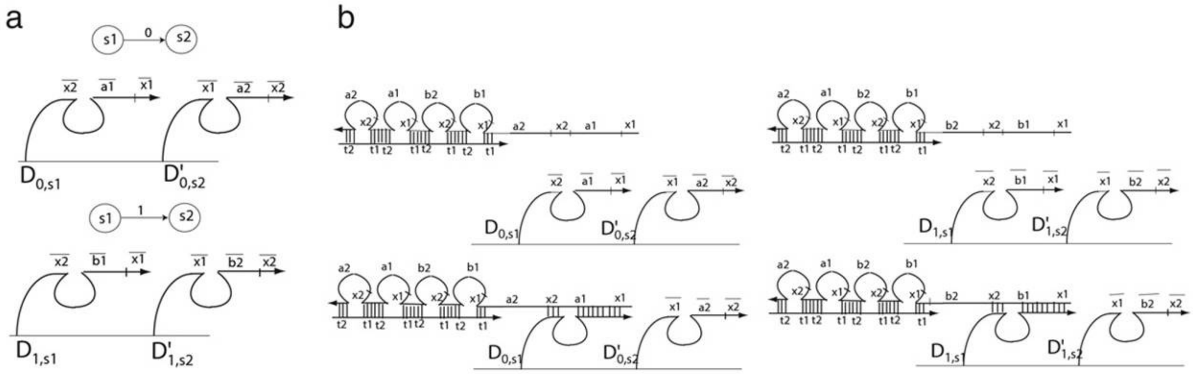


Figure 31: Mao's DNAzyme Walker. A) Walker moving from one spot to the next on the track. B) Walker and 4 foot holds for it. Orange subsequence is the DNAzyme, blue dots are places where the DNAzyme can cleave.

Bath and Turberfield (2007) 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 at al. (2005) describe an extension of the design of the restriction-enzyme based autonomous DNA walker of Yin et al. (2004*b*) described above in subsection 10.4.3, to allow programmed computation while moving along a DNA nanostructure.



a
s1
0
s2
$\overline{x2}$ $\overline{a1}$ $\overline{x1}$
$\overline{x1}$ $\overline{a2}$ $\overline{x2}$
$D_{0,s1}$
$D'_{0,s2}$
s1
1
s2
$\overline{x2}$ $\overline{b1}$ $\overline{x1}$
$\overline{x1}$ $\overline{b2}$ $\overline{x2}$
$D_{1,s1}$
$D'_{1,s2}$
b
a2 a1 b2 b1
x2 x1 x2 x1
t2 t1 t2 t1 t2 t1 t2 t1
a2 x2 a1 x1
b2 x2 b1 x1
$D_{0,s1}$
$D'_{0,s2}$
$D_{1,s1}$
$D'_{1,s2}$

Figure 32: 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 33) for programmed computation while moving along a DNA nanostructure was developed by Reif and Sahu (2007) using in this case an extension of the design of the DNAzyme based autonomous DNA walker of Tian et al. (2005) also described above. It remains a challenge to experimentally demonstrate these.

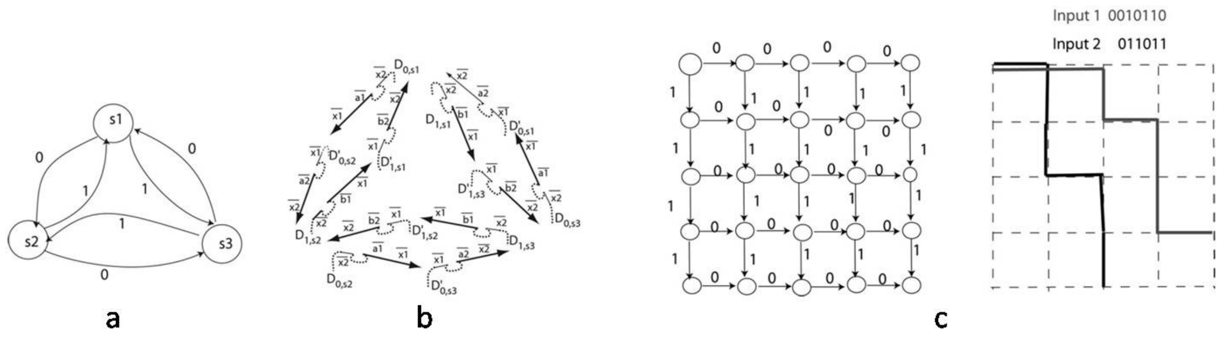


Figure 33: Programmed traversal of a grid DNA nanostructure: a) Transistion diagram of a finte state maci\hine b) The DNAzyme implementation of the finite state machine shown in a. c) Illustration of programmable routing in 2D

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-autonoumous, but otherwise often ingenious and powerful. As just one notable example, Stojanovic and Stefanovic (2003) demonstrated a deoxyribozyme-based molecular automaton and demonstrated it’s use to play the optimal strategy for a simple game.

Other excellent surveys of DNA nanostructures and devices have been given by Seeman (2004), Sha et al. (2005), Deng et al. (2006), de Castro (2006), LaBean and Li (2007), Lund et al. (2006) and Bath and Turberfield (2007).

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.*

Benenson et al. (2004) 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 (2007) 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.

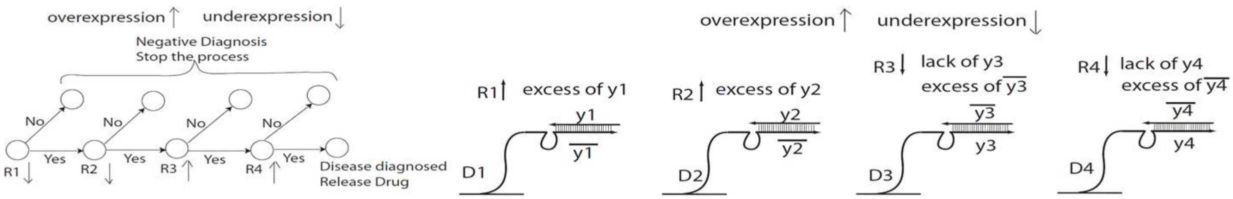


Figure 34: A finite-state computing DNA device for medical diagnosis based on Reif and Sahu's DNAzyme based autonomous DNA nanobot. (a) A state diagram for DNAzyme doctor nanobot that controls the release of a "drug" RNA on the basis of the RNA expression tests for a disease. (b) The figure shows the 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 y1 and y2 respectively, and they block the path of input nanostructure by hybridizing with D1and D2. Similarly underexpression of R3 and R4 results in excess of y3 and y4 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

We would like to thank #####

References

Adar, R., Benenson, Y., Linshiz, G., Rosner, A., Tishby, N. & Shapiro, E. (2004), Stochastic Computing with Biomolecular Automata', *Proceedings of the National Academy of Sciences of the United States of America 101*(27), 9960–9965.

Adleman, L. (1994), 'Molecular Computation of Solutions to Combinatorial Problems', *Science 266*(5178), 1021–1024.

Adleman, L. (1998), 'Computing with DNA', *Scientific American 279*(52), 54–61.

Amin, R., Kim, S., Park, S. H. & LaBean, T. (2009), 'Artificially Designed DNA Nanostructures', *NANO: Brief Reports and Reviews 4*(3), 119–139.

Andersen, E., Dong, M., Nielsen, M., Jahn, K., Subramani, R., Mamdouh, W., Golas, M., Sander, B., Stark, H., Oliveira, C., Pedersen, J. S., Birkedal, V., Besenbacher, F., Gothelf, K. & Kjems, J. (2009), 'Self-assembly of a Nanoscale DNA Box with a Controllable Lid', *Nature 459*(7243), 73–76.

Barish, R., Rothemund, P. & Winfree, E. (2005), 'Two Computational Primitives for Algorithmic Self-assembly: Copying and Counting', *NanoLetters 5*, 2586–2592.

Bath, J. & Turberfield, A. (2007), 'DNA Nanomachines', *Nature Nanotechnology* 2, 275–284.

Benenson, Y., Gil, B., Ben-Dor, U., Adar, R. & Shapiro, E. (2004), 'An Autonomous Molecular Computer for Logical Control of Gene Expression', *Nature 429*(6990), 423–429.

Berger, R. (1966), 'The Undecidability of the Domino Problem', *Memoirs of American Mathematical Society 66*, 1–72.

Chen, H.-L. & Goel, A. (2004), 'Error Free Self-assembly Using Error Prone Tiles', *DNA Computing* pp. 62–75.

Chen, H.-L., Schulman, R., Goel, A. & Winfree, E. (2007), 'Reducing Facet Nucleation during Algorithmic Self-assembly', *Nano Letters 7*, 2913–2919.

Chen, Y., Wang, M. & Mao, C. (2004), 'An Autonomous DNA Nanomotor Powered by a DNA Enzyme', *Angewandte Chemie International Edition 43*(27), 3554–3557.

de Castro, L. (2006), Fundamentals *of Natural Computing: Basic Concepts, Algorithms, and Applications*, Chapman and Hall.

Deng, Z., Chen, Y., Tian, Y. & Mao, C. (2006), 'A Fresh Look at DNA Nanotechnology', *Nanotechnology: Science and Computation* pp. 23–34.

Dietz, H., Douglas, S. & Shih, W. (2009), 'Folding DNA into Twisted and Curved Nanoscale Shapes', *Science 325*(5941), 725–730.

Dirks, R. & Pierce, N. (2004), ‘Triggered Amplification by Hybridization Chain Reaction’, Proceedings *of the National Academy of Sciences of the United States of America 101*(43), 15275–15278.

Douglas, S., Chou, J. & Shih, W. (2007), ‘DNA-nanotube-induced Alignment of Membrane Proteins for NMR Structure Determination’, *Proceedings of the National Academy of Sciences of the United States of America 104*(16), 6644–6648.

Douglas, S., Dietz, H., Liedl, T., Hogberg, B., Graf, F. & Shih, W. (2009), ‘Self-assembly of DNA into Nanoscale Three-dimensional Shapes’, *Nature 459*(7245), 414–418.

Fujibayashi, K., Hariadi, R., Park, S. H., Winfree, E. & Murata, S. (2008), ‘Toward Reliable Algorithmic Self-Assembly of DNA Tiles: A Fixed-Width Cellular Automaton Pattern’, *Nano Letters 8*(7), 1791–1797.

Goodman, R., Berry, R. & Turberfield, A. (2004), ‘The Single-step Synthesis of a DNA Tetrahedron’, *Chemical Communications* pp. 1372–1373.

Hamada, S. & Murata, S. (2009), ‘Substrate-assisted Assembly of Interconnected Single-Duplex DNA Nanostructures’, *Angewandte Chemie International Edition 48*(37), 6820–6823.

He, Y., Chen, Y., Liu, H., Ribbe, A. & Mao, C. (2005*a*), ‘Self-assembly of Hexagonal DNA Two-Dimensional (2D) Arrays’, *Journal of the American Chemical Society 127*(35), 12202–12203.

He, Y., Tian, Y., Chen, Y., Deng, Z., Ribbe, A. & Mao, C. (2005*b*), ‘Sequence Symmetry as a Tool for Designing DNA Nanostructures’, *Angewandte Chemie International Edition 44*(41), 6694–6696.

He, Y., Ye, T., Su, M., Zhang, C., Ribbe, A., Jiang, W. & Mao, C. (2008), ‘Hierarchical Self-assembly of DNA into Symmetric Supramolecular Polyhedra’, *Nature 452*(7184), 198–201.

LaBean, T., Gothelf, K. & Reif, J. (2007), ‘Self-Assembling DNA Nanostructures for Patterned Molecular Assembly’, *Nanobiotechnology II* pp. 79–97.

LaBean, T. & Li, H. (2007), ‘Constructing Novel Materials with DNA’, *NanoToday 2*(2), 26–35.

LaBean, T., Winfree, E. & Reif, J. (1999), ‘Experimental Progress in Computation by Self-assembly of DNA Tilings’, *DNA Based Computers V, DIMACS*.

LaBean, T., Yan, H., Kopatsch, J., Liu, F., Winfree, E., Reif, J. & Seeman, N. (2000), ‘Construction, Analysis, Ligation, and Self-assembly of DNA Triple Crossover Complexes’, *Journal of the*

American Chemical Society 122(9), 1848–1860.

Lund, K., Williams, B., Ke, Y., Liu, Y. & Yan, H. (2006), ‘DNA Nanotechnology: A Rapidly Evolving Field’, *Current Nanoscience 2*, 113–122.

Majumder, U., LaBean, T. & Reif, J. (2007), ‘Activatable Tiles: Compact, Ro- bust Programmable Assembly and Other Applications’, *DNA Computing*

Mao, C., Labean, T., Reif, J. & Seeman, N. (2000), ‘Logical Computation Using Algorithmic Self-assembly of DNA Triple-crossover Molecules’, *Nature 407*, 493–496.

Park, S. H., Pistol, C., Ahn, S. J., Reif, J., Lebeck, A. & LaBean, C. D. T. (2006*a*), ‘Finite-Size, Fully Addressable DNA Tile Lattices Formed by Hierarchical Assembly Procedures’, *Angewandte Chemie International Edition* 45(5), 735–739.

Park, S. H., Prior, M., LaBean, T. & Finkelstein, G. (2006*b*), ‘Optimized Fabrication and Electrical Analysis of Silver Nanowires Templated on DNA Molecules’, *Applied Physics Letters* 89(3).

Reif, J. (2003), ‘The Design of Autonomous DNA Nano-mechanical Devices: Walking and Rolling DNA’, *DNA 8* pp. 439–461.

Reif, J. & Majumder, U. (2008), ‘Isothermal Reactivating Whiplash PCR for Locally Programmable Molecular Computation’, *DNA Computing pp*. 41–56.

Reif, J. & Sahu, S. (2007), ‘Autonomous Programmable Nanorobotic Devices Using DNAzymes’, *DNA Computing pp*. 66–78.

Reif, J., Sahu, S. & Yin, P. (2004), ‘Compact Error-Resilient Computational DNA Tiling Assemblies’, *DNA Computing pp*. 293–307.

Robinson, B. & Seeman, N. (1987), ‘The Design of a Biochip: A Self-assembling Molecular-scale Memory Device’, *Protein Engineering 1*(4), 295–300.

Robinson, R. (1971), ‘Undecidability and Nonperiodicity for Tilings of the Plane’, Inventiones *Mathematicae 12*, 177–209.

Rothemund, P. (2006), ‘Folding DNA to Create Nanoscale Shapes and Patterns’, *Nature* 440, 297–302.

Rothemund, P., Papadakos, N. & Winfree, E. (2004), ‘Algorithmic Self-assembly of DNA Sierpinski Triangles’, PLoS *Biology 2*.

Rothemund, P. & Winfree, E. (2000), ‘The Program-Size Complexity of Self-assembled Squares’, *Symposium on Theory of Computation* pp. 459–468.

Saiki, R., Scharf, S., Faloona, F., Mullis, K., Horn, G., Erlich, H. & Arnheim, N. (1985), ‘Enzymatic Amplification of Beta-globin Genomic Sequences and Restriction Site Analysis for Diagnosis of Sickle Cell Anemia’, *Science 230*(4732), 1350–1354.

Sakamoto, K., Kiga, D., Momiya, K., Gouzu, H., Yokoyama, S., Ikeda, S., Sugiyama, H. & Hagiya, M. (1999), ‘State Transitions by Molecules’, *Biosystems* pp. 81–91.

Schulman, R. & Winfree, E. (2009), ‘Programmable Control of Nucleation for Algorithmic Self-assembly’, SIAM *Journal on Computing 39*(4), 1581–1616.

Seeman, N. (2004), ‘Nanotechnology and the Double Helix’, *Scientific American 290*(6), 64–75.

Sha, R., Zhang, X., Liao, S., Constantinou, P., Ding, B., Wang, T., Garibotti, A., Zhong, H., Israel, L., Wang, X., Wu, G., Chakraborty, B., Chen, J., Zhang, Y., Yan, H., Shen, Z., Shen, W., Sa-Ardyen, P., Kopatsch, J., Zheng, J., Lukeman, P., Sherman, W., Chengde Mao, N. J. & Seeman, N. (2005), ‘Structural DNA Nanotechnology: Molecular Construction and Computation’, *Unconventional Computing pp*. 20–31.

Shapiro, E. & Benenson, Y. (2006), ‘Bringing DNA Computers to Life’, *Scientific American 17*(3), 4047.

Sherman, W. & Seeman, N. (2004), ‘A Precisely Controlled DNA Biped Walking Device’, *Nano Letters 4*, 1203–1207.

Shih, W., Quispe, J. & Joyce, G. (2004), ‘A 1.7-kilobase Single-stranded DNA that Folds into a Nanoscale Octahedron’, *Nature 427*(6975), 618–621.

Stojanovic, M. & Stefanovic, D. (2003), ‘A Deoxyribozyme-based Molecular Automaton’, *Nature Biotechnology 21*(9), 1069–1074.

Tian, Y., He, Y.,, Chen, Y., Yin, P. & Mao, C. (2005), ‘A DNAzyme That Walks Processively and Autonomously along a One-Dimensional Track’, *Angewandte Chemie International Edition 44*(28), 4355–4358.

Tian, Y., He, Y. & Mao, C. (2006), ‘Cascade Signal Amplification for DNA Detection’, *ChemBioChem 7*(12), 1882–1864.

Walker, T., Fraiser, M., Schram, J., Little, M., Nadeau, J. & Malinowski, D. (1992*a*), ‘Strand Displacement Amplification - An Isothermal, in Vitro DNA Amplification Technique’, Nucleic *Acid Research 20*(7), 1691–1696.

Walker, T., Little, M., Nadeau, J. & Shank, D. (1992*b*), 'Isothermal in Vitro Amplification of DNA by a Restriction Enzyme/DNA Polymerase System', *Proceedings of the National Academy of Sciences of the United States of America 89*(1), 392–396.

Wang, H. (1961), 'Proving Theorems by Pattern Recognition II', *Bell Systems Technical Journal.*

Winfree, E. (1995), 'On the Computational Power of DNA Annealing and Ligation', DNA *Based Computers, DIMACS.*

Winfree, E. & Bekbolatov, R. (2003), 'Proofreading Tile Sets: Error Correction for Algorithmic Self-assembly', *DNA Computing pp*. 126–144.

Winfree, E., Liu, F., Wenzler, L. & Seeman, N. (1998), 'Design and Self-assembly of Two-dimensional DNA Crystals', *Nature 394*, 539–544.

Winfree, E., Yang, X. & Seeman, N. (1996), 'Universal Computation via Self- assembly of DNA: Some Theory and Experiments', *DNA Based Computers II, DIMACS* 44, 191–213.

Yan, H., Feng, L., Labean, T. & Reif, J. (2003*a*), 'Parallel Molecular Computations of Pairwise Exclusive-Or (XOR) Using DNA String Tile Self-assembly', *Nature* 125(47), 1424614247.

Yan, H., LaBean, T., Feng, L. & Reif, J. (2003*b*), 'Directed Nucleation Assembly of DNA Tile Complexes for Barcode-patterned Lattices', *Proceedings of the National Academy of Sciences of the United States of America 100*(14), 8103–8108.

Yan, H., Park, S. H., Finkelstein, G., Reif, J. & LaBean, T. (2003*c*), 'DNA-Templated Self-assembly of Protein Arrays and Highly Conductive Nanowires', *Science 301*(5641), 1882–1884.

Yin, P., Choi, H., Calvert, C. & Pierce, N. (2008), 'Programming Biomolecular Self-assembly Pathways', *Nature 451*(7176), 318–322.

Yin, P., Guo, B., Belmore, C., Palmeri, W., Winfree, E., LaBean, T. & Reif, J. (2004*a*), TileSoft: Sequence Optimization Software for Designing DNA Secondary Structures, Technical report, Duke and Caltech.

Yin, P., Sahu, S., Turberfield, A. & Reif, J. (2005), 'Design of Autonomous DNA Cellular Automata', *DNA Computing pp*. 376–387.

Yin, P., Turberfield, A., Sahu, S. & Reif, J. (2004*b*), 'Designs for Autonomous Unidirectional Walking

DNA Devices', *DNA Computing pp*. 410–425.

Yin, P., Yan, H., Daniell, X., Turberfield, A. & Reif, J. (2004*c*), 'A Unidirectional DNA Walker That Moves Autonomously Along a Linear Track', *Angewandte Chemie International Edition 116*(37), 4906–4911.

Yurke, B., Turberfield, A., Mills, A., Simmel, F. & Neumann, J. (2000), 'A DNA-fuelled Molecular Machine Made of DNA', *Nature 406*(6796), 605–608.

Zhang, D., Turberfield, A., Yurke, B. & Winfree, E. (2007), 'Engineering Entropy-Driven Reactions and Networks Catalyzed by DNA', *Science 318*, 1121–1125.

Zheng, J., Birktoft, J., Chen, Y., Wang, T., Sha, R., Constantinou, P., Ginell, S., Mao, C. & Nadrian (2009), 'From Molecular to Macroscopic via the Rational Design of a Self-assembled 3D DNA Crystal', *Nature 461*(7260), 74–78.