Computing Handbook Set - Computer Science (Volume I)

Chapter: DNA Computing

Sudhanshu Garg, Reem Mokhtar, Tianqi Song, Hieu Bui, Nikhil Gopalkrishnan, John Reif
reif@cs.duke.edu
Department of Computer Science, Duke University
(Reif is also Adjunct, Technology (FCIT), King Abdulaziz University (KAU), Jeddah, Saudi Arabia)

Organization of Chapter

Molecular computing is computation done at the molecular scale. DNA computing is a class of molecular computing that does computation by the use of reactions involving DNA molecules. DNA computing has been by far the most successful (in scale and complexity of the computations and molecular assemblies done) of all known approaches to molecular computing, perhaps due in part to the very well established biotechnology and biochemistry on which its experimental demonstration relies, as well as the frequent teaming of scientists in the field with multiple essential disciplines including chemistry, biochemistry, physics, material science, and computer science.

This chapter surveys the field of DNA computing. It begins in Section 1 with a discussion of the underlying principles, including motivation for molecular and DNA computations (Section 1.1), brief overviews of DNA structures (Section 1.2), chemical reaction systems (Section 1.3), DNA reactions (Section 1.4), and classes of protocols and computations (Section 1.5). Then, the chapter discusses potential applications of DNA computing research (Section 2). The main section on research issues (Section 3.1) overviews how DNA computation is done, with a discussion of DNA hybridization circuits, including both solution-based as well as localized hybridization circuits. It also discusses design and simulation software for the same. We discuss DNA detectors in Section 3.2, DNA replicators in Section 3.3, DNA nanorobotic devices in Section 3.4, and DNA dynamical systems in Section 3.5. Section 3.6 overviews research on tiling assembly computations, including theoretical models and results in Section 3.6.1, experimental methods for assembly of tiling lattices in Section 3.6.2, as well as design and simulation software in Section 3.6.3, assembly in various dimensions in Sections 3.6.4-3.6.5, step-wise tiling assemblies in Section 3.6.6, activatable tiles in Section 3.6.7, and tiling error-correction methods in Section 3.6.8.

1 Underlying Principles

1.1 Motivation: Why do Molecular Computation and Why use DNA for Computation and Self-Assembly ?

In an era where electronic computers are powerful and inexpensive, why do we need molecular computation? One response to this question is that conventional electronic devices have been miniaturized to the point where traditional top-down methods for manufacturing these devices are approaching their inherent limits due to constraints in photolithographic techniques, and further miniaturization is not cost-effective. On the other hand bottom-up manufacturing methods such as molecular self-assembly have no such scale size limits. Another response is that molecular computation provides capabilities that traditional computers can not provide; there are computations that need to be done in environments and at scales where a traditional computer can not be positioned, for example within a cell or within a synthetic molecular structure or material.

Why use Nucleic Acids such as DNA for Computation and Self-Assembly?

DNA, and nucleic acids in general, are unique in multiple aspects:

1. First of all, they hold and can convey information encoded in their sequences of bases. Most of their key physical properties are well understood.

2. Their tertiary structure is much more predictable, compared to molecules such as proteins.

3. Their hybridization reactions, which allow for addressing of specific subsequences, are also well understood and productively controllable.

4. They allow for a large set of operations to be performed on them. Well-known enzymatic reactions for manipulation of DNA exist.

5. Finally, there is a well-developed set of methods such as gel-electrophloresis, FRET, plasmonics, AFM imaging, etc. for quantifying the success of experiments involving DNA and DNA nanostructures.

Before we delve into how molecular computation is done, we will discuss DNA structure and function, how information may be stored in it, and what environment it needs to efficiently do computation.

1.2 Brief Overview of DNA Structure

DNA is a polymer which can exist in either single or double stranded form. Each strand of DNA is made up of a repeating set of monomers called *nucleotides*. A nucleotide consists of three components, a 5 carbon sugar molecule, a nitrogenous base, and a phosphate group. The ...-phosphate-sugar-phosphate-... covalent bond forms the backbone of a DNA strand. The phosphate group is attached to carbon atom 5 (C5) on one end, and C3 on another end. This gives the DNA strand directionality, and the two ends of a DNA strand are commonly termed the 5' (prime) and the 3' ends. This can be seen in Figure 1.

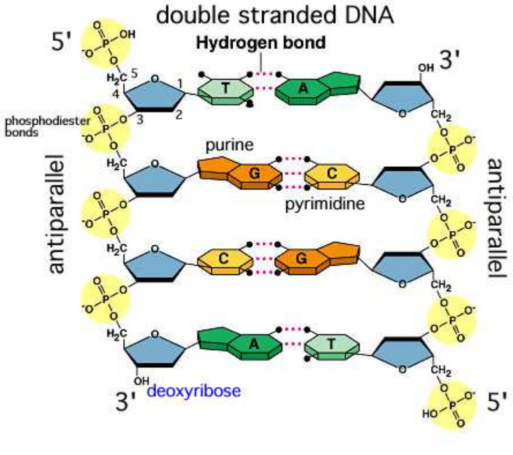


Figure 1: DNA Backbone (on left) and DNA Bases involved in hydrogen bonding (middle) [62]

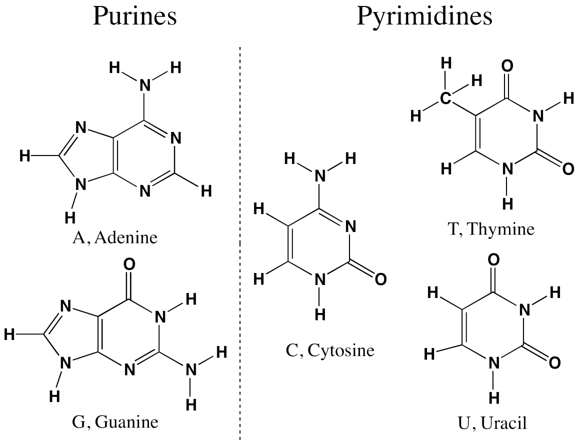


Figure 2: Structure of DNA/RNA bases

1.2.1 DNA Bases

Nitrogenous bases are the component of nucleotides not involved in forming the backbone of a strand. There are 5 types of these bases, named Adenine(A), Guanine(G), Cytosine(C), Thymine(T) and Uracil(U). Only 4 of these A,G,C,T are present in DNA, while T is replaced by U in RNA.

Bases A and G belong to a class called *Purines*, while C,T and U fall under the *Pyrimidines* class. Figure 2 shows the difference in structure of Purines and Pyrimidines. A purine has a pyrimidine ring fused to an imidazole ring, and contains 4 nitrogen atoms as opposed to 2 nitrogen atoms in a pyrimidine.

1.2.2 ssDNA & dsDNA Structure: The Double Helix

DNA can exist either in single stranded DNA (ssDNA) form, or as a result of two complementary ssDNA binding together via hydrogen bonds to form double-stranded DNA (dsDNA). The two ssDNA are always *antiparallel* when bound, i.e. one strand has 5' to 3' direction, while the other has a 3' to 5' direction. DNA exists as a double helix, as shown in Figure 3.

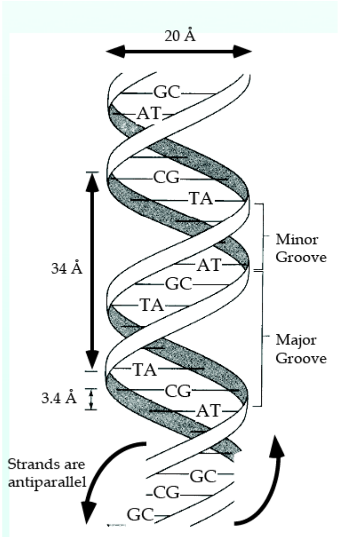


Figure 3: Double Helix form of dsDNA (B-form) [43]

The nitrogen bases in each ssDNA bind with a complementary base in the other strand, to give rise to this structure: A binds with T, and G binds to C. This pairing of bases, is called the *Watson-Crick bonding* in DNA, as shown in detail in Figure 4. An important note is that a *purine always binds to a pyrimidine* and this can be seen in Figure 1.

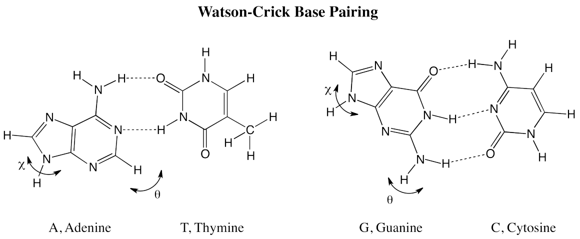


Figure 4: Watson Crick Hydrogen Bonding

ssDNA and dsDNA can have different helical conformations, namely the A, B, C, D, T and Z forms. The most common dsDNA forms are A, B and Z, and they can transform from one conformation to another based on the hydration conditions, the pH and the ionic concentration of the environment. The most common form of DNA is the B form, which it assumes when hydrated. A relative comparison of the these 3 conformations is shown in Table 1.

In its ssDNA form, DNA exists as a long single thread, or in many cases, it forms a *secondary structure*, where the strand loops around itself, and forms hydrogen bonds with other bases on itself (called *a random coil*).

Table 1: Comparison of A, B, Z forms of DNA.

Characteristic	A-DNA	B-DNA	Z-DNA
Helix Sense	Right-handed	Right-handed	Left-handed
Residues per turn (base pairs)	11	10.4	12
Axial Rise	2.55 Å	3.4 Å	3.8 Å
Helix Pitch	28 Å	35 Å	45 Å
Base Pair Tilt	20^o	-6^o	7^o
Helix Width (Diameter)	23 Å	20 Å	18 Å
Phosphate-Phosphate Distance	5.9 Å	7.0 Å	5.9 Å
Dimension	Broad	Normal	Narrow

*Data compiled from [8, 79, 93]

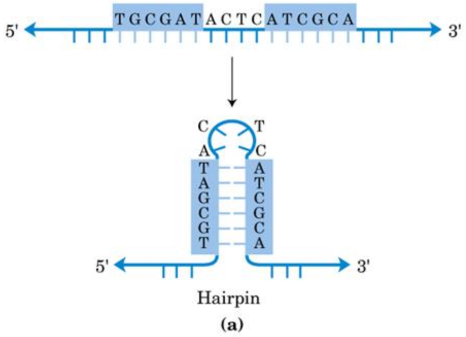


Figure 5: Hairpin open and closed forms [11]

1.2.3 DNA Hairpins

DNA hairpins are a special *secondary structure* formed by an ssDNA, and contain a neck/stem double stranded region, and an unhybridized loop region, as seen in Figure 5. Hairpins have been recognized as a useful tool in molecular computation because of 3 reasons: (1) Hairpins store energy in their unhybridized loop, and on hybridization, energy is released driving the reaction forward. (2) In their hairpin form, they are relatively unreactive with other DNA strands, and act as excellent monomers until an external entity (usually another DNA strand) causes the stem region to open and react with other DNA complexes. Hence, they can persist with low leaks for a long amount of time. (3) A common way to create DNA complexes is to anneal them. DNA complexes usually contain a large number of strands, and multiple different structures can be formed because of varied interactions between different strands. In low concentrations, DNA hairpins usually form without error, and are not involved in spurious structure formation. This is because their formation is not diffusion dependent, i.e. the two ends of a hairpin hybridize with each other before two ends of different hairpins hybridize. This property, is known as *locality*, and is a strong motivation for the use of hairpins.

Hairpins [32], and metastable DNA hairpin complexes [85,102] have been used as fuel in chain reactions to form large polymers [20], in programming pathways in self-assembly [119], and in logic circuits [84]. A common technique to help open a DNA hairpin is via a process known as *toehold mediated strand displacement*, which we shall discuss in more detail in Section 1.4.1.

1.3 Brief Overview of Chemical Reaction Networks in DNA Computing

Chemical reaction networks (CRNs) are becoming central tools in the study and practice of DNA computing and molecular programming. Their role is twofold - as a model for analyzing, quantifying and understanding the behavior of certain DNA computing systems and as a specification/programming language for prescribing information processing (computational) behavior. The first of these roles is traditional and is analogous to the role played in Biology by CRNs in describing biochemical processes and genetic reaction networks. The latter role - thinking of CRNs as a programming language - is unique to the field of DNA computing and is a consequence of the ability of DNA to act as an information processing medium and emulate (with certain restrictions) any CRN set down on paper. We will discuss both these roles briefly in the following paragraphs.

1.3.1 CRNs Model DNA Strand Displacement Reaction Networks

Enzyme-free DNA computing devices can execute i) Boolean circuits and linear threshold gate networks (the latter model neural networks) [68, 69, 83], (ii) Nucleic acid amplifiers [20, 119, 127] (iii) Finite state automata [30] and (iv) Molecular walkers [31, 121]. All of these devices are examples of strand displacement reaction networks (SDRNs). In a toehold-mediated strand displacement reaction an incoming DNA strand displaces a competing DNA strand hybridized to a DNA substrate strand. The incoming strand first binds to a toehold - a short single stranded portion of the substrate - and then competitively displaces the outgoing strand from the substrate by a one-dimensional random walk process. A cascade (network) of such strand displacement reactions are called *SDRNs*.

The modular design characteristics of SDRNs allow them to be modelled as CRNs. In particular, the types and rates of reactions are limited. We can infer them from prior experience and/or predict them from thermodynamic parameters [129]. This allows us to predict the CRN model and verify its predictions experimentally.

1.3.2 CRNs as a Programming Language

In theory [95], SDRNs closely approximate the dynamic behaviour of any CRN upto a time and concentration scaling. They illustrate how any CRN that we set down upon paper can be translated into a set of DNA molecules that when mixed together in the appropriate concentrations will emulate the behavior of the CRN. Certain CRNs seem hard to emulate in practice and no successful SDRN implementations currently exist for these but many others have been successfully implemented.

CRNs are more abstract than SDRNs and can be thought of as a higher level programming language. The process of translating a CRN into its corresponding SDRN is then analogous to compiling a higher level programming language down to a lower level programming language. Programming in the CRN language has the advantages inherent in programming in higher level languages versus programming in lower level languages.

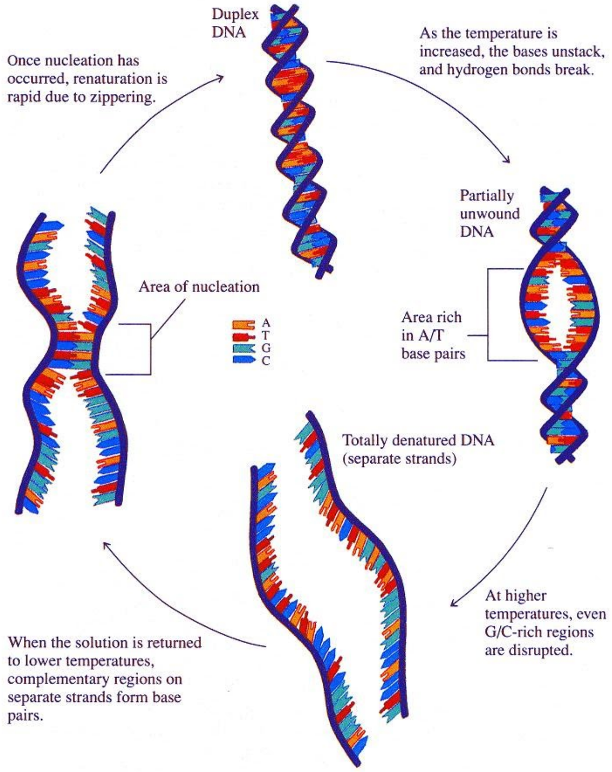
How powerful is the CRN language? Quite powerful, it turns out. It is proven in [94] that a finite CRN obeying stochastic dynamics can achieve efficient Turing-universal computation with arbitrarily small (non-zero) error. Error-free computation is impossible in this setting, only semilinear functions can be computed without errors [14].

1.4 DNA reactions

In order to be able to efficiently compute with DNA, we should be aware of its properties, and the types of reactions it can undergo. We classify this set of reactions into three types - DNA hybridization reactions, DNA enzyme reactions, and DNAzyme reactions. (1) DNA hybridization reactions are usually enzyme free and isothermal, and encapsulate strand displacement reactions. (2) DNA enzyme reactions are powerful reactions, which can help cut and join the backbone of DNA strands, as well as synthesize new strands, and are often employed due to this versatility. Enzymatic reactions are often extremely rapid, and extremely low error, hence making them attractive to use. (3) More recently, deoxyribozymes (DNAzymes) and aptamers have been discovered and used similar to enzymes to manipulate DNA reactions.

1.4.1 DNA Hybridization Reactions

A well-known example of DNA hybridization reactions is the Watson-Crick DNA hybridization between two complimentary ssDNA strands as discussed in Section 1.2.2. Two ssDNA strands can attach to each other. However, they can also detach from one another, if the temperature is greater than the melting temperature of the strands (Figure 6). The *melting temperature* of a dsDNA is defined as the temperature at which 50% of the dsDNA has converted to single stranded form.


Duplex DNA
As the temperature is increased, the bases unstack, and hydrogen bonds break.
Once nucleation has occurred, renaturation is rapid due to zippering.
Partially unwound DNA
Area of nucleation
A
T
G
C
Area rich in A/T base pairs
Totally denatured DNA (separate strands)
At higher temperatures, even G/C-rich regions are disrupted.
When the solution is returned to lower temperatures, complementary regions on separate strands form base pairs.

Moran, L.A. et al. (1994) "Biochemistry" Neil Patterson Publishers/Prentice Hall p. 24-16

Figure 6: DNA Denaturation Renaturation [60]

Toehold Mediated Strand Displacement Yurke *et al.* reported an interesting DNA hybridization reaction through their DNA tweezers system [124]. As illustrated in Figure 7a), two ssDNA strands (B and $\overline{CB}$ are bound to one another, with one strand, called the *incumbent strand* (strand B) completely bound, while the other $\overline{CB}$ has a few unbound bases. These bases ($\overline{C}$) can together be called a *sticky end*, *overhang*, or a *toehold*. A third ssDNA, called the *incoming strand* (strand BC), complementary to the ssDNA with the toehold ($\overline{CB}$), can hybridize to the toehold region ($\overline{C}$) and displace the incumbent strand (B). This process is termed *toehold-mediated strand displacement*. Typical toehold lengths used for toehold-mediated strand displacement hybridization reactions range from 3 to 7 nucleotides. The rate constant of the toehold-mediated strand displacement ranges from 1 $M^{-1}s^{-1}$ to 6×10^6 $M^{-1}s^{-1}$.

Toehold Exchange Toehold exchange is an extension to toehold mediated strand displacement, but it is extremely powerful. Zhang and Winfree [129] made the reaction in the previous section reversible, by introducing a small exit toehold. As seen in Figure 7b), toehold exchange proceeds in the same manner as toehold mediated strand displacement. Strand B has been logically divided into two parts B^m and B_m, where B^m is a small segment (3-7 nt) long. The incoming strand is now shorter (B_mC). On toehold mediated strand displacement, it does not completely dehybridize the incumbent strand ($B = B^mB_m$), and B^m of the incumbent bound. This segment floats away autonomously since its binding strength is too low to keep it in place at the current temperature (usually room temperature - 25C).

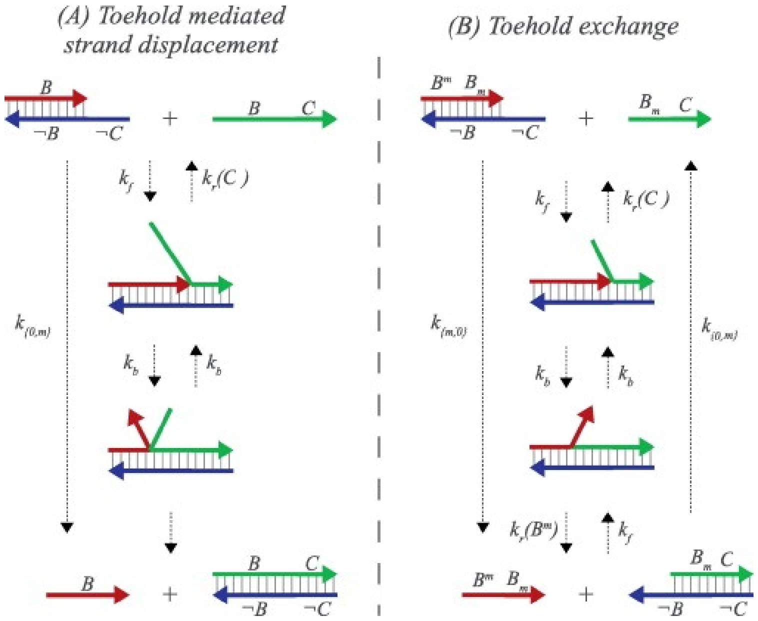


Figure 7: Toehold-mediated strand displacement and Toehold Exchange [37]

Note that now, the incumbent strand has a toehold ($\overline{B^m}$), which it can attach to, and in the process, displace the strand B_mC. By simply shortening the incumbent strand, the process has been made reversible. This set of reactions, as we shall see in Section 3.1, are extremely useful in designing hybridization circuits.

1.4.2 DNA Enzyme Reactions

Enzymes are proteins that facilitate biochemical reactions. Enzymes are highly specific and usually catalyze their desired reactions. Theoretically, enzymes can be reused in a biochemical reaction, without being consumed by the reactants. They also provide high efficiency to their reactions. For example, one enzyme named catalase in liver can break down roughly five-million molecules of hydrogen peroxide into oxygen and water in 5 minutes.

In the field of DNA-based computation, scientists are currently working with a small subset of enzymes such as restriction enzymes, nicking enzymes, ligase enzymes, and polymerase enzymes. The purpose of restriction enzymes and nicking enzymes is to cleave the phosphodiester bond within a chain of nucleotides whereas the purpose of ligase enzymes is to repair the phosphodiester bond as illustrated in Figure 8. In order for an enzyme to cleave or repair a specific location, the location has to have a recognition site that is designated for that particular enzyme. A recognition site (highlighted in red in Figure 8) is normally a few nucleotides along the DNA double helix. For example, a *SmaI* restriction enzyme cleaves two phosphodiester bonds on both sides of a given DNA double helix to give an end result of two blunted ends of DNA double helices (Figure 8). Similarly an *EcoRI* restriction enzyme cleaves two phosphodiester bonds on both sides of a given DNA double helix to give an end result of two separated overhang DNA helices (Figure 8). A *Nb.BsmI* nicking enzyme on the other hand, cleaves a single phosphodiester bond on one side of a given DNA double helix. Furthermore, a broken phosphodiester bond can be repaired using a T4 ligase enzyme as illustrated in Figure 8. To amplify a particular DNA sequence, researchers often employ the polymerase chain reaction (PCR) technique. This technique involves the use of polymerase enzymes to catalyze the polymerization of nucleoside trisphosphates (dNTPs) into a multiple copies of the target DNA sequence.

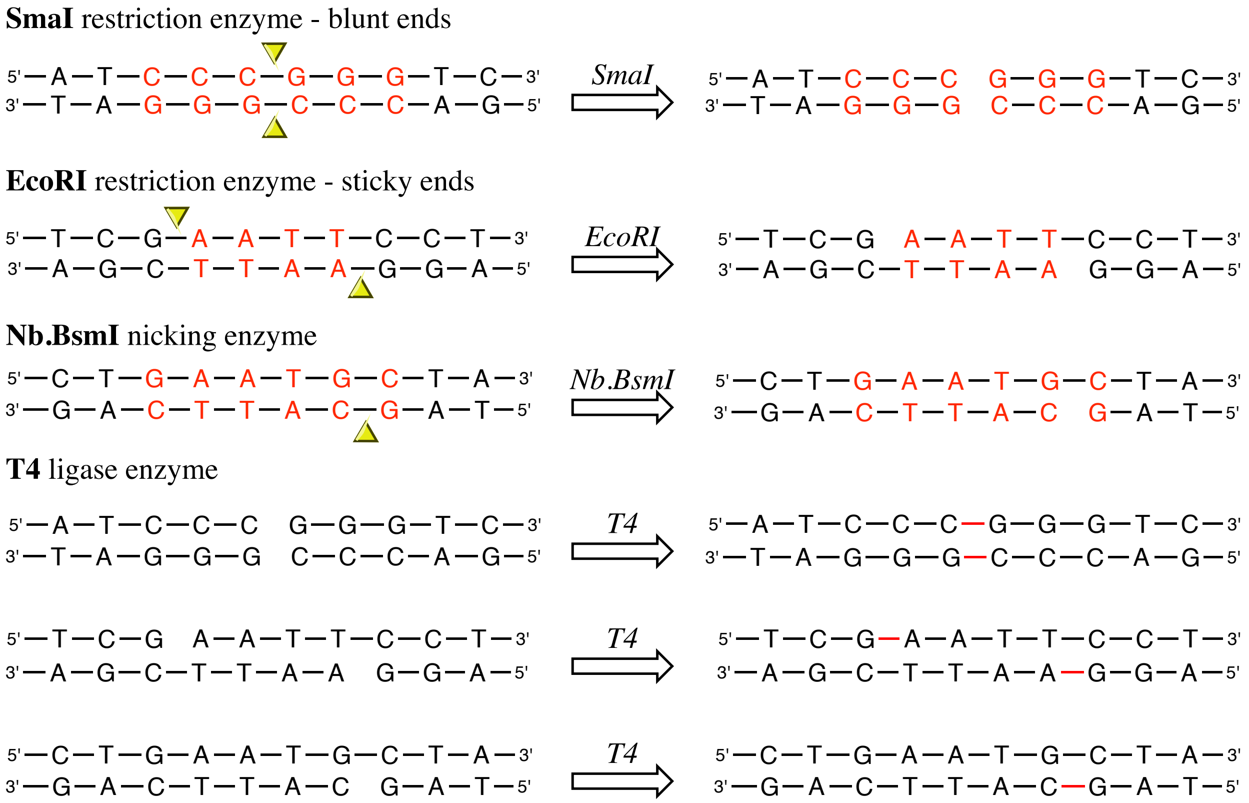

SmaI restriction enzyme - blunt ends
SmaI
EcoRI restriction enzyme - sticky ends
EcoRI
Nb.BsmI nicking enzyme
Nb.BsmI
T4 ligase enzyme
T4
T4
T4

Figure 8: DNA Enzymes

1.4.3 DNAzyme Reactions

Recently researchers have been using deoxyribozymes (DNAzymes) and aptamers to catalyze DNA hybridization reactions. DNAzymes are isolated by *in vitro* selection while aptamers are discovered by *in vivo* evolution search. Both DNAzymes and aptamers are DNA-based sequences that possess enzymatic activities which extend beyond Watson-Crick hybridization and they have highly specific binding to target molecules. An example of the use of DNAzyme for DNA computation was demonstrated by Stojanovic *et al.* by building a molecular automaton using a network of DNAzymes [96].

1.5 Classes of Protocols and Computations

Molecular computations, specifically DNA computations, are generally conducted by the use of well-defined protocols that are reproducible. In the field of biochemistry, the term *protocol* denotes a precise method for the synthesis of materials such as preparing reagents, adding solutions and reagents to a test tube, and changing physical parameters (*i.e.*, temperature), as well as separation of materials, and readout of data, etc. Most computations fall into one of the following classes of protocols.

1.5.1 Autonomous vs NonAutonomous Molecular Computations

The execution of a lengthy protocol may require a very considerable human effort. A particularly favorable class of protocols are those that are *autonomous*, requiring no further externally mediated operations after the solutions and reagents are added together. (Otherwise, the protocol is termed *nonautonomous*.)

1.5.2 Stepped Protocols

Stepped protocols [71] are a more general class of protocols that proceed by a sequential series of steps, where each step is autonomous, but between each consecutive step there may be an externally mediated operation involving a single test tube.
Staged protocols [19] are an even more general class of protocols that proceed by a sequential series of stages involving multiple test tubes at the same time, where each stage is autonomous, but between stages there may be a single externally mediated operation involving each test tube or pairs of the test tubes.

1.5.3 Local vs Solution-Based Protocols

A reaction is *solution-based* if it occurs within a solution, and materials have to diffuse to each other in order to interact. Likewise, it is *local* if it occurs between materials that are attached to a surface, thereby reducing/eliminating the diffusion time.

1.5.4 Activatable Molecular Devices

A molecular device is *activatable* if it has two classes of states: *inactive*, where it generally cannot undergo state transitions until an activating event occurs, and *active*, where it can undergo state transitions. An activatable device usually is initialized inactive, and transitions to an active state after an activation event.

2 Impact on Practice

Some of the practical applications of DNA computations and DNA nanoassemblies include the following:

1. **Detection of nucleic acid sequences:** For many diseases, detection of the disease is done via the determination of a characteristic sequence s of DNA or RNA, where s has base-length no more than 20 or so base pairs. This detection is conventionaly done via the well known PCR protocol, but this requires somewhat bulky and expensive device for repeated thermocycles and optical detection. The market for these detection protocols is over a billion dollars. DNA computation protocols may provide a much more cost-effective and portable means for detection. Section 3.2 describes various isothermal (requiring no thermal cycling) protocols for esquisit sensitive detection of DNA and RNA sequences.

2. **3D DNA Nanoassemblies of Proteins for x-Ray Crystalography Structure Determination:** Almost half of all proteins of interest to medicine can not be crystallized, and so their 3D structure can be determined from conventional X-ray crystallogaphy studies. An important application of DNA nanoassembly, proposed by Seeman, is the assembly of 3D DNA lattices that can hold a given protein at regular postions in the crystalline lattice, allowing for X-ray crystallography studies of an otherwise uncrystallizable protein. For further discussion of this application, see this Chapter's Section 3.6.5 discussion three-dimensional DNA nanoassemblies [130]

3. **3D DNA Nanoassemblies for Alignment of Proteins for NMR Structure Determination:** Douglas and Shi [22] have proposed and demonstrated a novel method for improved NMR studies of structure determination that makes use of long DNA nanoassemblies to partially align proteins.

4. **2D DNA Nanoassemblies for Molecular Patterning** DNA nanoassemblies can be used for programmed patterning of 2D molecular surfaces. This has applications to assembly of molecular electronic devices on molecular surfaces, and 2D readout methods. [47, 76, 118]

5. **DNA nanorobotic devices for molecular assembly lines:** Programmed control and sequencing of chemical reactions at the molecular scale has major applications in the chemical industry. DNA nanorobotic devices (see Section 3.4) have been demonstrated that execute a programmed sequence of chemical reactions at predetermined positions along a DNA nanostructure [33, 57].

3 Research Issues

3.1 DNA Hybridization Circuits

Construction of circuits made out of DNA, has been an area of interest since the advent of molecular computing. Having millions to billions of nanoscale DNA gates working in tandem, to accept an input, and produce an unambiguous output, is a challenging task. There have been various designs to create Logic Gates and boolean circuits using DNA motifs [1, 44, 64, 65, 67, 69, 80, 81, 83, 84, 97, 98, 113, 128]. We examine a few such attempts at creating these circuits and the advantages and pitfalls associated with each. We classify the circuits into two types: DNA Circuits contain a large number of DNA molecules, and each molecule reacts with a small subset of other molecules. Circuits in which molecules diffuse through solution, to find a molecule from this subset, are said to be *solution based*. On the other hand, DNA molecules can be attached to a surface, and these molecules only interact with other molecules within their reach. Such circuits are known as localized hybridization circuits.

3.1.1 Solution-Based DNA Hybridization Circuits

Qian and Winfree developed a scalable solution based DNA hybridization circuits. Their basic motif is a see-saw gate based on the toehold-exchange protocol as seen in Section 1.4 developed by [129]. The system consists of a set of see-saw gates. Each gate can accept an input (strand), and produce an output(strand). The input strands act catalytically, hence a single input strand can help release multiple output strands from multiple see-saw gates. The output strands act as inputs for the next see-saw gates downstream in the circuit. The Figure 9 shows the basic mode of operation of a see-saw gate motif.

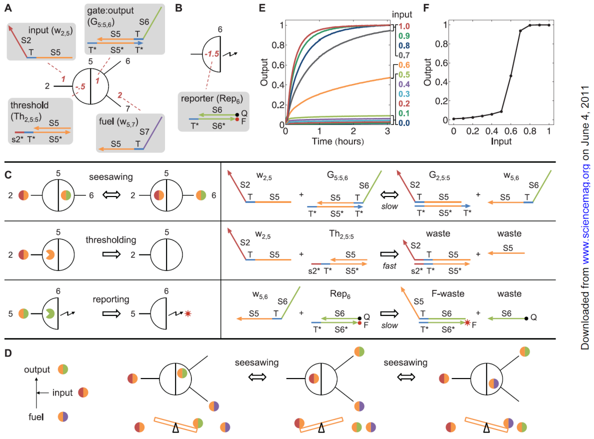


Figure 9: Components of a see-saw system [68]

The input strand consists of 3 domains, S2, T and S5, and the output strand consists of S5, T and S6. Note that the 5' end of the input and the 3' end of the output need to have the same set of domains, while the converse is not true (S2 and S6 can be different). These domains S2 and S6 can be used to help differentiate different input strands and different output strands that interact with the same gate. We now show how the authors use this difference to construct OR and AND gates.

Construction of AND - OR Gates Having constructed the basic motif, the authors then constructed AND and OR gates using the same motif. In order to construct an OR gate, either one of the input strands (x1, x2) should release the output strand. I.e. the presence of either x1 OR x2 gives the output. Take the following construction: x1 = S10 T S5, x2 = S11 T S5. We can trivially see that either one of x1 or x2 will release the output strand. Hence, this gate, is equivalent to an OR gate.

Construction of an AND gate is non-trivial. Only when both x1 and x2 are present should the output strand be released. The authors introduce a gate called the *'threshold gate'* as seen in Figure 9. The threshold gate can be thought of as a garbage collector, that sucks a strand in, and makes it unusable for further reactions. By varying the concentration of the threshold gate, both AND and OR gates can be constructed. This is seen in Figure 9.

A DNA Circuit that computes the Square Root of n $\leq$ 15 Using the AND and OR gate motifs, we can now construct any circuit, by connecting a set of such gates sequentially. The authors construct a circuit that computes the square root of all numbers upto 15. It takes in 8 inputs, a 0 or a 1 for each bit. The output set of strands, denote the set of bits that are set to 1, and hence encode the result.

The system described above is an example of how DNA can be used to do logical computation *in-vitro*, and *detect* any input strands that are present in the vicinity. There are several disadvantages of this approach, with the primary one being that this methodology is slow, and as the depth of the circuit increases, the amount of time taken to perform a reliable computation increases linearly. Also, circuits such as these are susceptible to errorneous reactions taking place between gates and strands that are not intended to happen, and this could lead to faulty outputs. In order to fix these problems, we discuss another set of circuits in the next section.

3.1.2 Localized DNA Hybridization Circuits

The solution based systems described above demonstrate the enormous potential of DNA nanosystems. But most of the systems relied on diffusion based hybridization to perform complex state changes/computation. At low concentrations and temperatures, diffusion can be quite slow and could impede the kinetics of these systems. At higher concentrations and temperature, unintended spurious interactions (often called *leaks*) could hijack the systems from its programmed trajectory. Localized hybridization networks are set of DNA strands attached to an addressable substrate. This localization increases the relative concentration of the reacting DNA strands thereby speeding up the kinetics. Diffusion based systems possess global states encoded via concentration of various species and hence exhibit limited parallel ability. In contrast, localized hybridization systems allow for each copy of the localized hybridization network to operate independently of each other. Localized hybridization networks also allow one to reuse the same DNA sequence to perform different actions at distinct location on the addressable substrate, increasing the scalability of such systems by exploiting the limited sequence space. An advantage of localized hybridization computational circuit is sharper switching behavior as information is encoded over state of a single molecule. This also eliminates the need for thresholding as computation is performed locally eliminating the need for a global consensus. These advantages are expounded in greater detail in (Chandran, et al., 2011). Quite recently, other articles such as (Genot, et al., 2011) and (Aldaye, et al., 2011), have also demonstrated that locality can be used successfully.

3.1.3 Design & Simulation Software

Computational tools are a growing necessity in order to design DNA strands for use in both simulations as well as experiments. Visual DSD from [42] is extremely useful in simulating DNA strand displacement reactions. NUPACK [125] and MFOLD [132] help in design and analysis of sequences at the base level, in order to weed out wrong designs.

3.2 Strand Displacement Reaction Network (SDRN) based DNA Detectors

The detection problem is the problem of designing a solution based chemical molecular circuit (henceforth, the *detector*) for detecting the presence of an input molecular species by producing an output molecule (or molecules) in sufficient quantities so that it can be detected by a measuring instrument. The detector must be sensitive to small amounts of input, specific to the correct type of input and robust to stochastic variability in the operation of the CRN. We might also ask that the detector function correctly over a wide range of concentrations, temperatures and buffer conditions.

In this section we will concentrate on the problem of detecting a short piece of DNA of known sequence. Various other molecules can often be transduced into a short DNA sequence by means of aptamer technology. This problem, interesting in its own right, will not be discussed in this chapter.

3.2.1 Advantages of SDRN Detectors

The canonical DNA detector is the polymerase chain reaction (PCR), an enzymatic detector which requires temperature cycling. Driven by exquisitely evolved enzymes and technological advances over the past fifty years, real time PCR protocols are sensitive to as little as 2000 DNA molecules in an mL of solution. However, PCR is not often specific to single base mutations in input, requires temperature cycling and works only in a narrow range of temperature and buffer conditions. Enzymatic detectors that seek to replace PCR have their own host of issues and rely on special enzymes.

In contrast, SDRN detectors are isothermal, easily modifiable to detect various sequences, can be made specific to single base mutations and may work correctly over a wide range of temperature, concentration and buffer conditions [126]. They can also function as a module (subroutine) within a larger SDRN performing a complex computation. For instance, an SDRN circuit that produces an output molecule only in the presence of two types of inputs (AND logic) or one detecting either one of two types of inputs, but not both together (XOR logic) or any arbitrary Boolean predicate. The key limitation with current SDRN detectors is their lack of sensitivity to small amounts of input, but the hope is that this issue can be surmounted [29]. A sensitive SDRN detector could have applications in a variety of settings, including *in vivo*.

3.2.2 Leaks Limit Sensitivity of Detectors

A sensitive detector must necessarily amplify small input signal into sufficiently large output signal. This amplification must be conditional, in the presence of input, the amplification must be be switched on and switched off in its absence. In terms of CRNs, this implies that detectors must have catalytic reaction pathways. Thermodynamics tells us that if a reaction proceeds in the presence of a catalyst it must also occur in the absence of that catalyst, typically at a slower rate. These unplanned reactions are termed as *leaks* and they limit the sensitivity of detectors [28]. Even in the absence of input, the detector produces output due to leak reactions. If the quantity of output produced (as a function of time) in the presence of input is sufficiently larger (detectable by measuring instrument) than the output produced (as a function of time) in the absence of input, the input is detectable. Otherwise the detector is not sensitive to the input.

Identifying and mitigating leak reactions is the key to implementing sensitive detectors. For simple CRNs the leak reactions may be easily guessed, since catalytic reactions are easily identified. What are the correct leak reactions when implementing more complex CRNs? Do the leak reactions depend on the particular technology we use to implement the CRN? Ref. [28] hypothesize that the correct leak model for a CRN is the *saturation* (defined in [27]) of that CRN. Informally, the *saturation* of a CRN introduces some basis set of leak reactions that gets rid of all purely catalytic pathways. There may be many choices for the basis. The *rates* of these reactions depend on the type of technology we use to implement them, but their nature (upto a choice of basis) is determined purely by the connectivity of the CRN.

In SDRN detectors, having identified the leak reactions, knowledge of strand displacement kinetics [129] allow us to guess the rates of these reactions, and hence we have an accurate model of the dynamic behavior of the SDRN detector before implementing them in the laboratory.

3.2.3 Overview of SDRN Detectors

Various successful implementations of SDRN detectors have been demonstrated. The hybridization chain reaction (HCR) of [20] triggers the formation of a linear nicked double stranded polymer in the presence of an input DNA strand. The polymer units are two DNA strands initially trapped in a metastable hairpin form. The input triggers the opening of one the hairpins, which in turn triggers the other and a linear cascade of triggers results. The driving force for the reaction is the decrease in enthalpy due to the formation of additional hybridization bonds in the nicked double stranded polymer as compared to the metastable hairpins. The system is described in Figure 10.

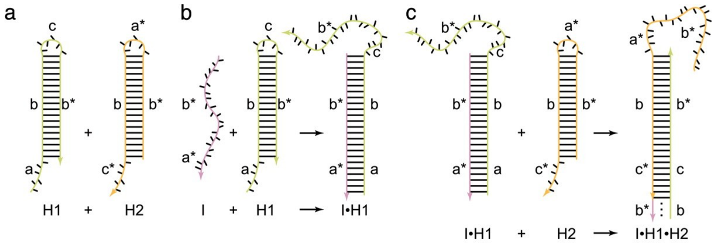


Figure 10: Hybridization Chain Reaction by [20] (a) Hairpins H1 and H2 are stable in the absence of initiator I. (b) I nucleates at the sticky end of H1 and undergoes an unbiased strand displacement interaction to open the hairpin. (c) The newly exposed sticky end of H1 nucleates at the sticky end of H2 and opens the hairpin to expose a sticky end on H2 that is identical in sequence to I. Hence, each copy of I can propagate a chain reaction of hybridization events between alternating H1 and H2 hairpins to form a nicked double-helix, amplifying the signal of initiator binding.

Ref. [127] demonstrated a SDRN detector driven by entropy increase. The input DNA strand displaces the output from the DNA substrate via a toehold mediated strand displacement reaction. The input is released back into the solution by a fuel strand that displaces it from the substrate. The net effect is the fuel replacing an output molecule on the substrate, with the input acting catalytically. The direct displacement of output by fuel, in the absence of the input, is kinetically infeasible due to the lack of a toehold. This simple SDRN detector, like HCR, exhibits linear kinetics. However when the output from one such SDRN detector is funnelled as input to a second such detector, the overall detector exhibits quadratic kinetics. Even an auto-catalytic system where the output feeds back into the detector and a cross catalytic system were demonstrated. These feed back systems exhibit exponential kinetics and hence detect input quickly. However, they suffer from higher leak rates, limiting sensitivity. The catalytic pathway is shown in Figure 11.

Ref. [119] adapted a hairpin based SDRN detector to allow it to exhibit exponential behaviour. Their cross catalytic system consists of two hairpins trapped in a metastable state. The hairpins are triggered by input and form a double stranded complex with single stranded overhangs and in the process release the input back into the solution. One of the single stranded overhangs of the complex also acts as a catalytic trigger, resulting in exponential kinetics: Figure 12. The sensitivity of this SDRN detector is comparable to that obtained in [127].

3.3 DNA Replicators

Scientists have long tried to discover the origin of life. What was the sequence of steps that nature took, in order to create life as we now know it. Given that initially there was just matter, nature used energy from different sources, and made autonomous self-replicating cellular machinery. With the discovery of the hereditary material DNA and RNA, it has been speculated that nucleic acids contain all the information required in order to completely define an organism, and that the replication of these constituents is sufficient to ensure the replication of the entire organism.

Scientists at the Craig Venter Institute, have created synthetic life, by artificially synthesizing a bacterial chromosome, with all the genes necessary for self replication, and using it to boot up a cell, and creating many copies of the same [26]. They ruled out the cause of self-replication being some external contaminant, by placing watermarks within the artificial DNA, and re-sequencing the DNA from the cell copies, to prove

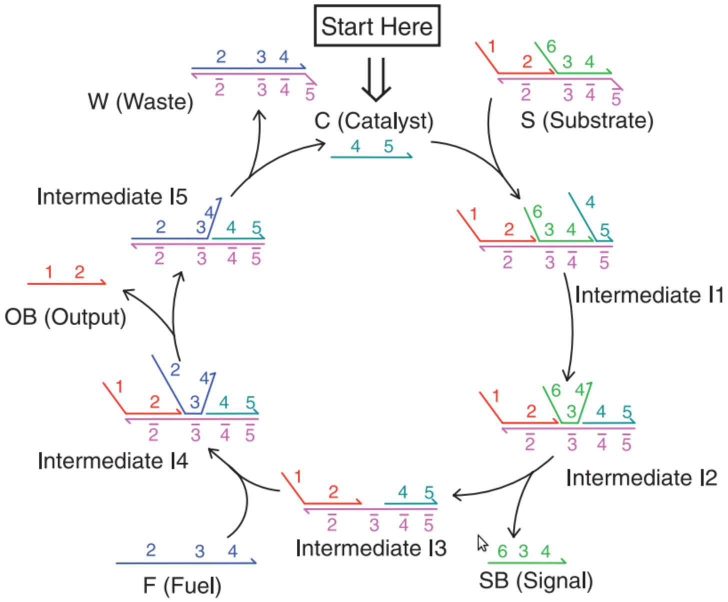


Figure 11: Entropy Driven Catalyst by [127] C (catalyst) first binds to the single-stranded toehold domain 5 on S to form the four-stranded intermediate I1, which then rearranges (by branch migration) to form I2. The binding between toehold domains 3 and 3 is too weak to keep SB attached, so I2 spontaneously dissociates into SB and I3. Newly exposed 3 then facilitates the binding of F, resulting in I4, which then quickly rearranges to release OB and I5. Finally, I5 rearranges so that C is attached only by the binding of 5 and 5 ,which spontaneously dissociates to leave W (waste) and regenerate C.

their claim. However, this cell makes use of various biological molecules (enzymes) already engineered by nature to aid in the replication of an organism. How did organisms first gain the ability to self-replicate, in the absence of these enzymes?

Attempts to answer this question have been made by the RNA world, who believe the RNA was the first self-sustaining molecule, before DNA was chosen to be the information bearing consituent of a cell. RNA, unlike DNA, is highly versatile, and can perform ligation [24] and polymerization [39], which argue in favor of it being enough to self-replicate an organism. However, the hypothesis that nature accidentally created RNA is highly debated. Many believe that in a pre-RNA world, another molecule(s) was the hereditary material, which could transfer information to and from RNA. Recent research in synthetic molecules such as XNA and TNA is an attempt to resolve questions in this direction.

What about in the absence of enzymes? An example of a non-enzymatic self-replicating system, is by Zhang and Yurke. As in other template based self-replication proposals, their initial system consists of a templated structure. In order to replicate, they exploit the physical properties of DNA (persistence length), in order to cause fission in this structure, leading to two partial copies. A sequential addition of constituents leads to two complete structures being formed. This scheme is shown in Figure [17]. Another theoretical model is by Chandran, Gopalkrishnan and Reif, which they term Meta-DNA [13]. This system consists of a template based on DNA nanotechnology, where a basic unit is a T-junction tile. Coupling multiple such tiles together forms a template, and they emulate a huge set of enzymatic operations, including polymerization, that can be done in this model.

There have been a few experimental models of self-replication in DNA nanotechnology as well. Scientists at NYU [107] use the BTX tile motif, and have created an organism P (parent) made up of 7 such BTXtiles. Organism P acts as a template on which self-replication can be done. In order to self-replicate, a set of BTX tiles that represent the complement of P, are added to the mix. This aids the formation of a daughter

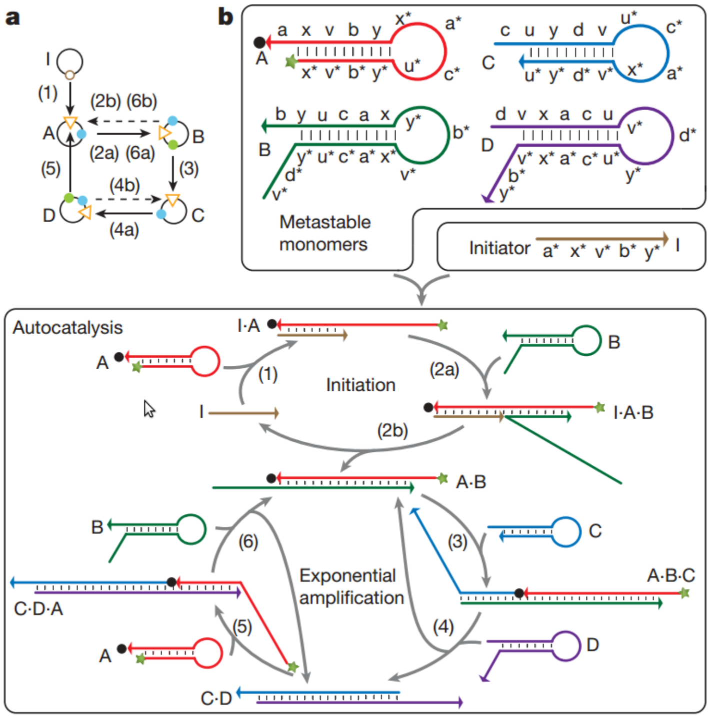


Figure 12: Hairpin based cross-catalytic detector by [119] Four hairpin species, A, B, C and D, coexist metastably in the absence of initiator I (In b)). The initiator catalyses the assembly of hairpins A and B to form duplex ANB (steps 1-2, bottom), bringing the system to an exponential amplification stage powered by a cross-catalytic circuit: the duplex ANB has a single-stranded region that catalyses the assembly of C and D to form CND (steps 3-4); duplex CND in turn has a singlestranded region that is identical to I and can thus catalyse A and B to form ANB (steps 5-6). Hence, ANB and CND form an autocatalytic set capable of catalysing its own production.

organism D, which in turn acts as a template again, and aids in the formation of a grandchild G, which is a duplicate of P, in terms of the set of tile motifs that both are made up of. This mechanism is depicted in Figure 13. In each replication step, a certain amount of error is introduced into the system, not unlike natural self-replication, where mutations are common. In this system, after two replications, about 31% of the original sequences are retained.

Another experimental demonstration of self-replication is by Schulman et. al [82], who attempt the problem of self-replication from a computer science perspective. They encode information in a set of DNA tiles, and induce growth via self-assembly. They have a proof-reading mechanism built in, in order to reduce the amount of errors in bit-copying. Next, they induce scission in the tiles, that is they break it apart by mechanical stress. These pieces of assemblies, now act as separate growth fronts for self-replication, and hereby starting a new cycle of replication. As compared to Seeman's model, about 76% of the original bits are propagated to the next set of organisms.

3.3.1 Conclusions and Open Problems

There have been several attempts to design enzyme-free DNA/RNA based replicators, but none are convincing enough, to reliably say that life could have originated via a particular route. Indeed, there could be different pathways via which life came to exist as we know of it today, and building different systems will only bring us closer to discovering this. There are several characteristics of these systems however, that one must appreciate. First, there is a distinction between self-replicating systems, and self-reproducing systems, and life we believe, might have originated from the former. Also, we must account for the environmental conditions in which these systems might have evolved in nature. DNA replicators are an attempt to designing such robust template based self-replicating systems.

3.4 DNA Nanorobotic Devices

3.4.1 DNA Walkers

Introduction

Molecular motors exist in nature, and many motor proteins of the dynein, kinesin and myosin families - that have been identified are known to support the tasks and functions that are essential to life. One of the goals of nanoscience is the ability to synthesize molecular motors that are programmable and capable of operating autonomously. Drawing inspiration from nature, many examples of such synthetic motors have been designed. DNA poses a relatively easy material to use as the building block for these motors, due to the fact that its components are well understood compared to other materials at the nanoscale [72].

Walkers are synthesized DNA systems that are able to move on a substrate. A major ambition for those designing and experimenting with DNA walkers is to be able to design autonomous systems that can be easily controlled and programmed with specific functions. Autonomy in this case can be seen as creating a system that can function in a constant environment, without external intervention, such that when the source of energy that drives it is depleted, the system halts [123].

Investigators have attempted to alter the method by which the walker moves, the substrate that the walker is moving on, or the fuel source that powers the walker in order to fully realize these two goals. In addition, investigators have also shown that walkers can perform specific tasks as a proof of their viability and potential, such as transporting cargo [121], moving along different programmable pathways [52, 72], assembling other structures [33], or performing controlled multistep chemical reactions [35]. One can divide the different classes of walkers in terms of 1) their degree of autonomy, or 2) the type of fuel that powers these walkers, or 3) both. We will start by looking at walkers with minimal autonomy, which respond to changes to their environment. Next, we will cover walkers powered by enzymes, followed by those that are driven by strand displacement reactions, and finally we will address walkers that are considered autonomous as of this writing. In this section, the terms stators, anchorage sites, or footholds on a track are synonymous for most of the walkers being mentioned, even if they differ in their specific designs; a place where the walker can attach to on a track or substrate.

Environmentally-driven Walkers

Nanomachines that function through the introduction of environmental alterations, such as variations to the pH of the solution, salt concentration, or the introduction of agents that cause synthetic DNA to change its conformation or shape, are considered non-autonomous, due to the fact these nanomachines are heavily dependent on external intervention. Such environmental characteristics have been observed since the early 70s (see [66], [48], [15] for some early examples), and later demonstrations of conformational changes based on the properties of the i-motif, (see [48], [49], [51], [50], [46], [58]). However, walkers that operate by means of environmental changes have only recently arisen.

Wang et.al. [105] demonstrate the operation of a walker that switches back and forth between two sites on a track (Figure 14). Each site is made of single strands of DNA that is partially hybridized to the track, with a loose, dangling region (15 bp) that acts as the binding site for the walker. The sites are separated by 15 nt, and the first site is made up of an i-motif DNA strand, which collapses onto itself at high acidic levels (pH 5) and switches back to a loose conformation in an alkaline solution (pH 8, For more details, see [45]).

Initially, when the walker is introduced, it hybridizes with the first site if the solution is alkaline. Once the solution becomes acidic, the loose strand of the first site forms a compact i-motif structure, which causes the walker to disassemble from the first site and hybridize with the second site. The limitation of this design is that the walker strand, moves back and forth between only two sites, without progressing any further.

Another example is the bipedal walker and stepper by Wang et al. [109] (Figure 15). This walker operates using 4 footholds(I,II,III,IV), each having a partially complementary strand attached to the track, such that the non-complementary site (1',2',3',4') acts as a sticky end for the walker to attach to. The sticky end of the second foothold is an i-motif. The walker is made up of 2 strands, each partially hybridized to a scaffold strand (8). Activation occurs by adding Hg^{2+}, which causes the formation of a thymine-Hg^{2+}-thymine bridge on the third site, and moves the walker to the second and third footholds. If the acidity of the solution is increased (increasing H^+), the i-motif on the second foothold creates a compact structure, leading the walker to disassociate from it and attach to the fourth site, such that it is now bound to the third and fourth sticky ends. This walker can cycle back to its original state. Increasing alkalinity, which causes the i-motif to deprotonate and take on the conformation of a random coil, causes the walker to dissociate from the last site and hybridize with the second site. Adding cysteine causes the thymine-Hg^{2+}-thymine bonds to break from the third site and return to the original, preferred state of hybridization. The authors also describe an extension of this design, to create a circular bipedal walker such that the track becomes circular, with the footholds attaching outwardly on the scaffold of the track.

Enzymatically-driven Walkers

Walkers that operate via enzymatic reactions were designed next. The first such walker was introduced by Reif et al. [121], where a unidirectional walker was able to move from one site to another by the introduction of cleaving and ligating enzymes that targeted specific sites between the walker and the footholds at each state of movement. The walker is composed of 6 nt that walk over 3 anchorage sites (A,B,C) on a track (See Figure 16 The asterix denotes the current location of the walker). The anchorage sites are attached via a 'hinge', or 4 nt single strand of DNA, which adds flexibility to the sites. The walker starts off at the first anchorage site (A), and when Ligase (Step I) is added, it causes the walker to ligate with the second anchorage (B), which creates a recognition site.

When the restriction enzyme (Pf1M I) is added, it cleaves the two anchorages such that the walker ends up on the second site. This cycle gets repeated in steps III (Ligase) and IV(Restriction enzyme BstAPI). Note that the restriction sites of Pf1M I and BstAPI are different, causing the walker to move in a unique direction. Refs. [7] and [100] each also had notable unidirectional walkers. The walker given in [7] is an example of a 'burnt-bridge' mechanism [54], where random walks are biased in one direction, preventing the walker from moving backwards with a certain probability on a track with weak 'bridges' or stators. The walker in this case cleaves the stator that it is hybridized to through a restriction enzyme, which moves the walker forward to hybridize with the next stator. Since the previous stator(s) are 'burnt', the walker is much less likely to hybridize with them then it is to stay with the current stator, until another restriction enzyme cut catalyzes its next move.

Another example of a walker is by Sekiguchi et al. [89], where they design a unidirectional walker that moves along a predetermined path or pattern of stators or footholds. The walker is composed of three legs, with additional parts that anneal and cleave the different stators on the track to move forward.

The last example is a walker by Wickham et.al. [111] (based on their earlier work in [7]) where a walker of length 16bp walks over 16 stators, each separated by approximately 6 nm in a track of approximately 100 nm on a rectangular DNA origami (Figure 17a)). The walker was powered by a nicking enzyme, which can only act once the walker strand is hybridized to the stator strand (hence the enzyme operates in a stepwise fashion). The movement of the walker was directly observed using real-time atomic force microscopy as it walked over the origami stators. The walker is initially hybridized to the starting stator. Addition of the nicking enzyme (which targets a specific site on the backbone of the stator once a duplex forms), causes a cut in the DNA backbone 6 nt from the 5' end (Figure 17b). The 6nt strand dehybridizes and floats away, resulting in a toehold being created. The adjacent stator hybridizes to the walker at the toehold, and subsequent strand displacement helps the walker move to this adjacent stator. This process is repeated with each stator until the last stator is reached, which does not contain the nicking recognition site. This stops the walker from further movement. Hairpins were added as an AFM imaging reference parallel to the track. In addition, FRET experiments were run by having the walker carry a quencher with it that reacts with different fluorophores on the track to determine when the walkers reaches specific points on or at the end of their tracks.

Walkers driven by strand-displacement reactions

Strand displacement reactions involve the use of a short, unhybridized portion (called a *toehold*) of a strand in a duplex to mediate the displacement of its complementary hybridized strand. These reactions can power walkers, as seen in the first such example by Seeman and Sherman [91], where a biped walker used set and unset strands to move along a three foothold track. Figure 18 shows a cartoon depicting individual steps of the walker. Figure 18a) shows two set strands 'Set 1A' and 'Set 2B', keep the walker attached to the footholds. In Figure 18b), an unset strand 'Unset 2B' uses toehold mediated strand displacement to unbind the 'Set 2B' strand from both the biped's foot and the foothold strands. This creates a duplex waste product Figure 18c), and causes the biped's foot to dissociate from the foothold. In Figure 18d), Strand 'Set 2C' is added, which causes one foot of the walker to hybridize to foothold 2C. In Figure 18e), 'Unset 1A' detaches the left foot, and in Figure 18f), 'Set 1B' attaches the left foot to foothold B. In this manner, the walker is propelled forward.

The walker by Shin and Pierce [92] is similar to Seeman's walker, except that there the trailing foot starts the motion forward along the track, while the leading foot keeps the walker attached to the track. Other similar examples include [101]'s circular biped walker [[101]], [31]'s Brownian ratchet walker [[31]], and [103]'s autonomous cargo transporter [[103]].

Muscat et al. [61] demonstrated a more recent example, where the walker is composed of a cargo strand, powered forward by the introduction of a fuel strand (Figure 19). What distinguishes this walker is its usage of a combination of toehold hybridization and junction migration. In addition, the fuel strands are able to provide the walker with direction instructions.

Anchorage sites attached to a double-stranded track are made up of the domains $\overline{X}\overline{b}\overline{c}$, where $\overline{X}$ is unique to each site, while $\overline{b}\overline{c}$ are common to all. Note that $X = X_1\ X_2$. A cargo strand abc can bind to anchorage site at $\overline{b}\overline{c}$. The domain a is left to split (hence the term split toehold) from the anchorage site and expose ($\overline{X}$ (the identifying address domain). This exposure is key for the walker to function. Another split toehold (between toehold y and domain $\overline{c}$) is made at the second anchorage site, where the strand R_y (composed of Y_1, a, b and y) is partially complementary to the second anchorage site.

The fuel strand is made up of 2 main parts: one to hybridize with the first anchorage site, and the other part to hybridize with the complementary strand of the second anchorage site (Figure 19a)). The fuel strand starts by partially hybridizing to this identifying region first, which is called *split toehold hybridization* (Figure 19b)). Afterwards, through junction migration the fuel strand moves the cargo from the anchorage site onto its common binding domain (Figure 19c)). Once that occurs, another split toehold hybridization reaction occurs, this time between y and $\overline{c}$ and the now unsequestered complementary domains $\overline{y}$ and the split toehold on the cargo strand c (Figure 19d)). Through junction migration(Figure 19e)), the cargo strand is finally moved onto the second anchorage site, and the fuel strand anchors R_y to the first anchorage site (Figure 19f)).

Another example includes a transporter by Wang et al. [110], which is composed of two parts. The main nanostructure is composed of a 3-arm structure that has 4 footholds. Three on the circumference, and one at the center (Figure 20(a)). To each foothold, there is attached one strand each (A,B,C,A_x). This structure is named Module I. The second part of the system (Module II) consists of an arm, that can transport a cargo strand (P_x) around the central axis in a circular fashion. Similar to the set and unset strands seen in [91] earlier, the cargo strand is detached from the site it is initially hybridized to via the introduction of a fuel strand (Panel I, (Figure 20(b))) complementary to that site. The arm is free to attach to the next site, B, (Panel II) until another fuel strand is introduced that essentially displaces the cargo strand from B to attach to C (Panel III). Anti-fuel strands can be introduced to reverse the sequence of movement. In order to ensure that the walker moves in the order specified, the duplexes that the cargo forms at each site were made energetically less stable starting from the first site, A to the last site, C.

Autonomous, Programmable Walkers

The class of walkers described here are those that are considered programmable and autonomous. [72] introduced the theoretical design of an autonomous programmable RNA walker that acts as a finite state machine. The machine is given instructions as input, and based on the previous input, it transitions to the next state, on a 2D addressable lattice. The input instructions are fed as a sequence of hairpins, and DNAzymes proceed to consume each instruction that the walker has already transitioned from, exposing the next instruction.

Another example is that of walkers that operate through multiple legs, or spiders in [64]. These spiders are comprised of a streptavidin core, and DNAzymes for legs, which bind and cleave RNA sites on a substrate. The walker is biased and moves towards unvisited substrates. Ref. [52] expanded on this work by demonstrating a spider exhibiting Brownian random motion by preprogramming the pathway on which their spider walks, in addition to making it respond to different actions such as stopping and turning.

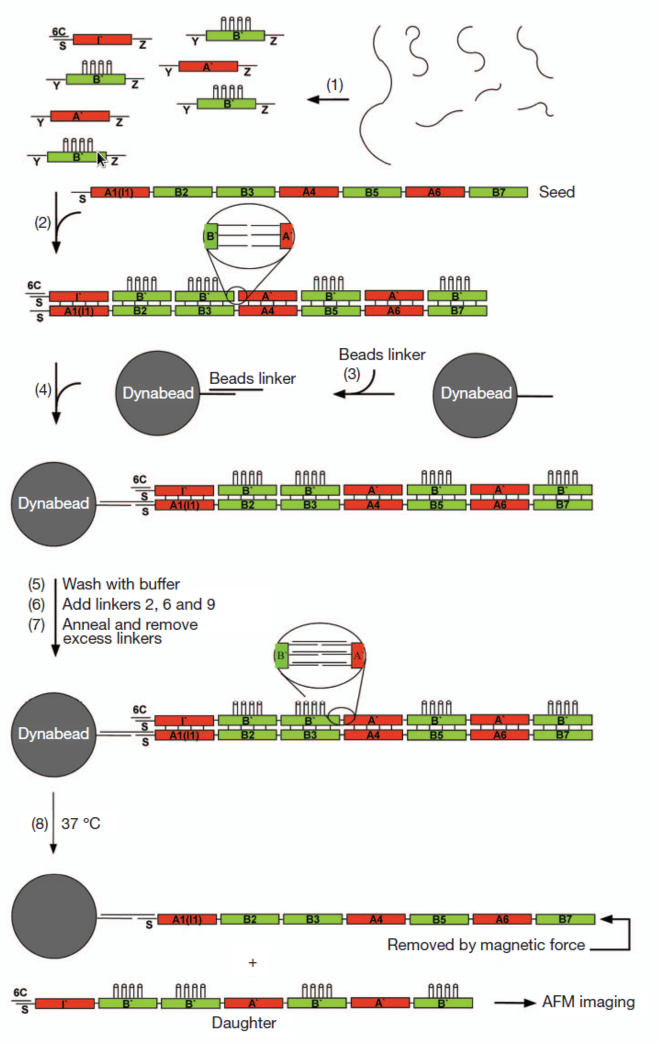


Figure 13: A seven tiled DNA replicator [107] Replication of the seven-tile seed pattern in the first generation. In step 1, strands are annealed into tiles, all flanked by the same connectors, designated Y and Z. The initiator tile (I9) contains a protected S-strand, paired with a cover strand, 6C. The B9 tiles contain the 4-hairpin markers for AFM imaging. In the presence of the seed (step 2), the strands assemble into a pattern mimicking the seed pattern. The magnetic dynabead (large grey circle) is prepared in step 3, and attached to the seed (step 4). This is followed by a wash step, the addition of linkers, and their annealing (steps 5-7). Heating the system to 37C results in the separation of the daughter 7-tile complex and the seed (removed magnetically, using the dynabead).

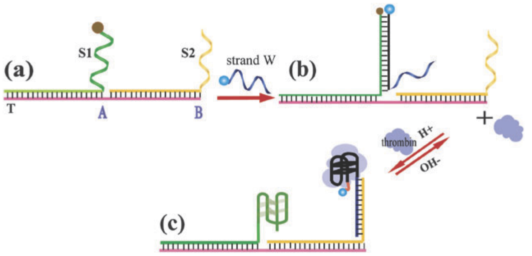


Figure 14: A walker powered by the pH levels of its environment [105]

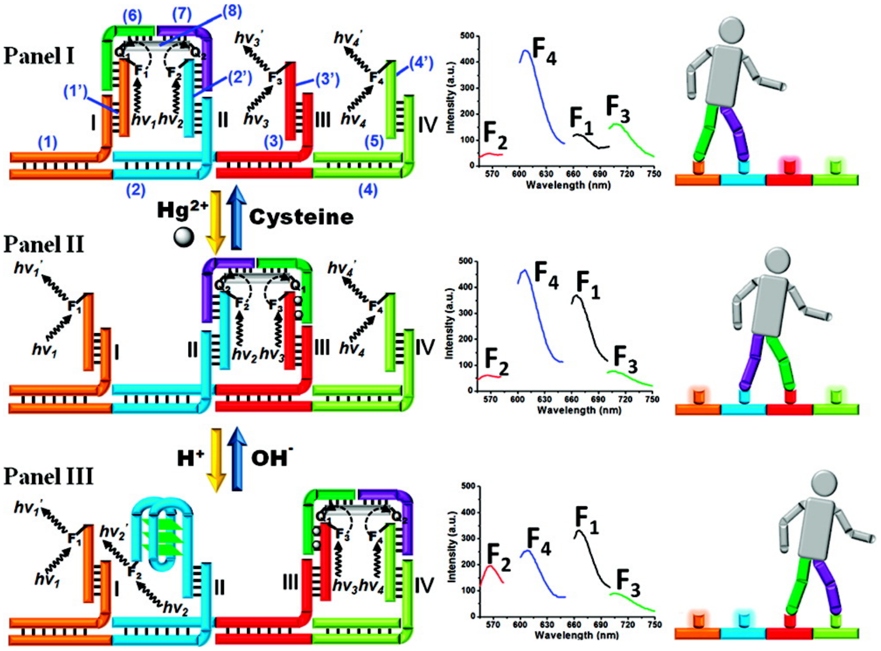


Figure 15: A walker that moves using both the i-motif, and Hg^{+}2 and cysteine [109]

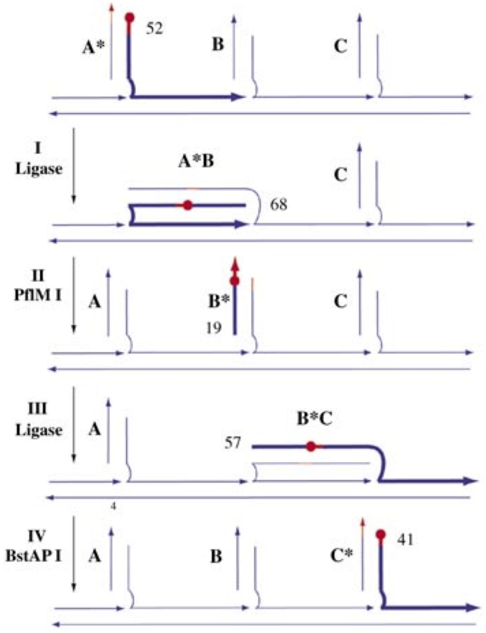


Figure 16: A unidirectional walker powered by enzymatic reactions [121]

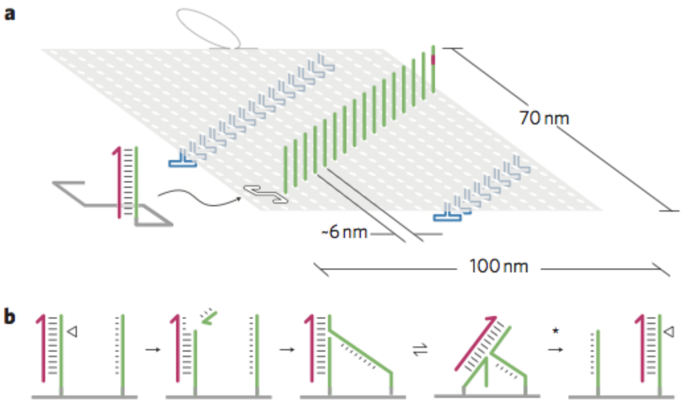


Figure 17: Stepwise observation of a walker moving along a DNA origami substrate [111]

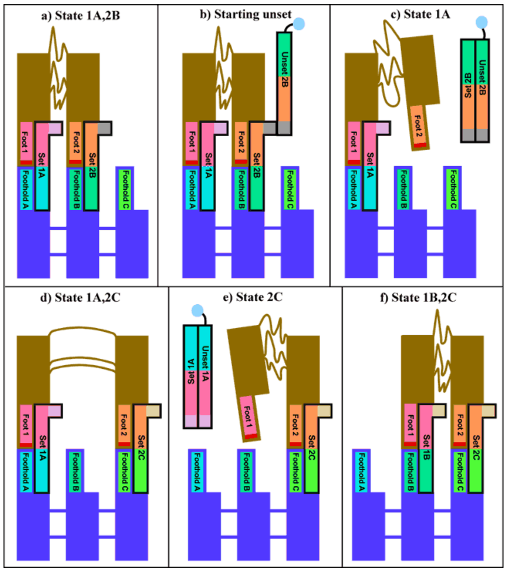


Figure 18: A biped DNA walker using set & unset strands to move at each step [91]

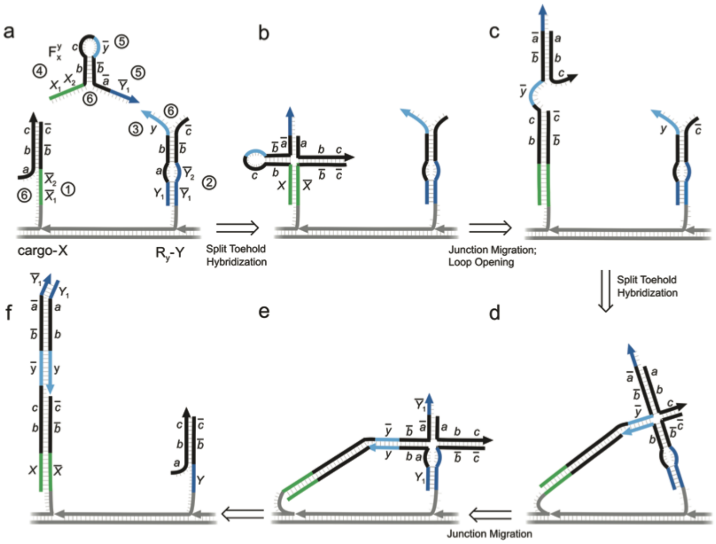

a
b
c
d
e
f
cargo-X
R_y-Y
Split Toehold Hybridization
Junction Migration; Loop Opening
Split Toehold Hybridization
Junction Migration

Figure 19: A walker that is powered by split toehold hybridization and junction migration [61]

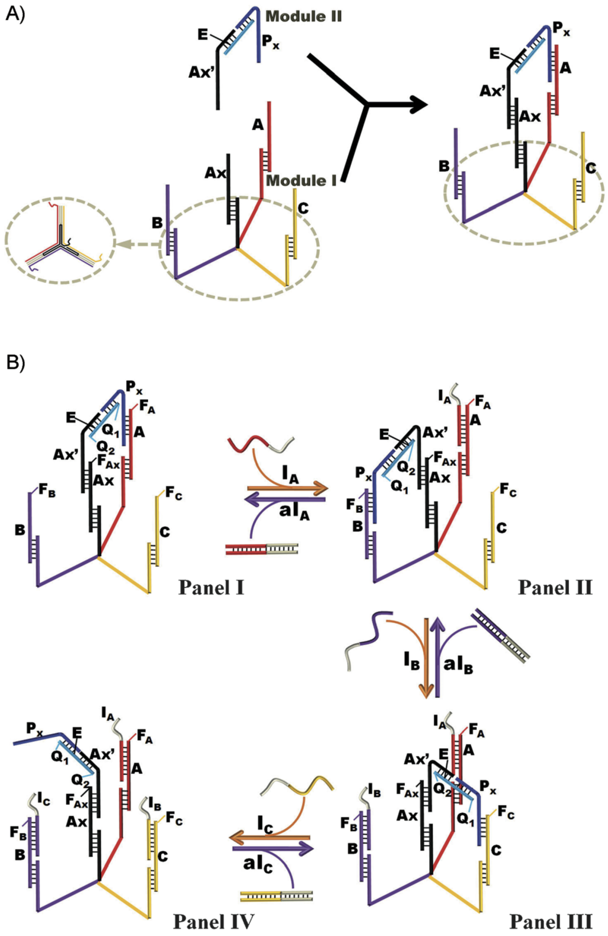


Figure 20: A transporter that can move cargo along a circular track [110]

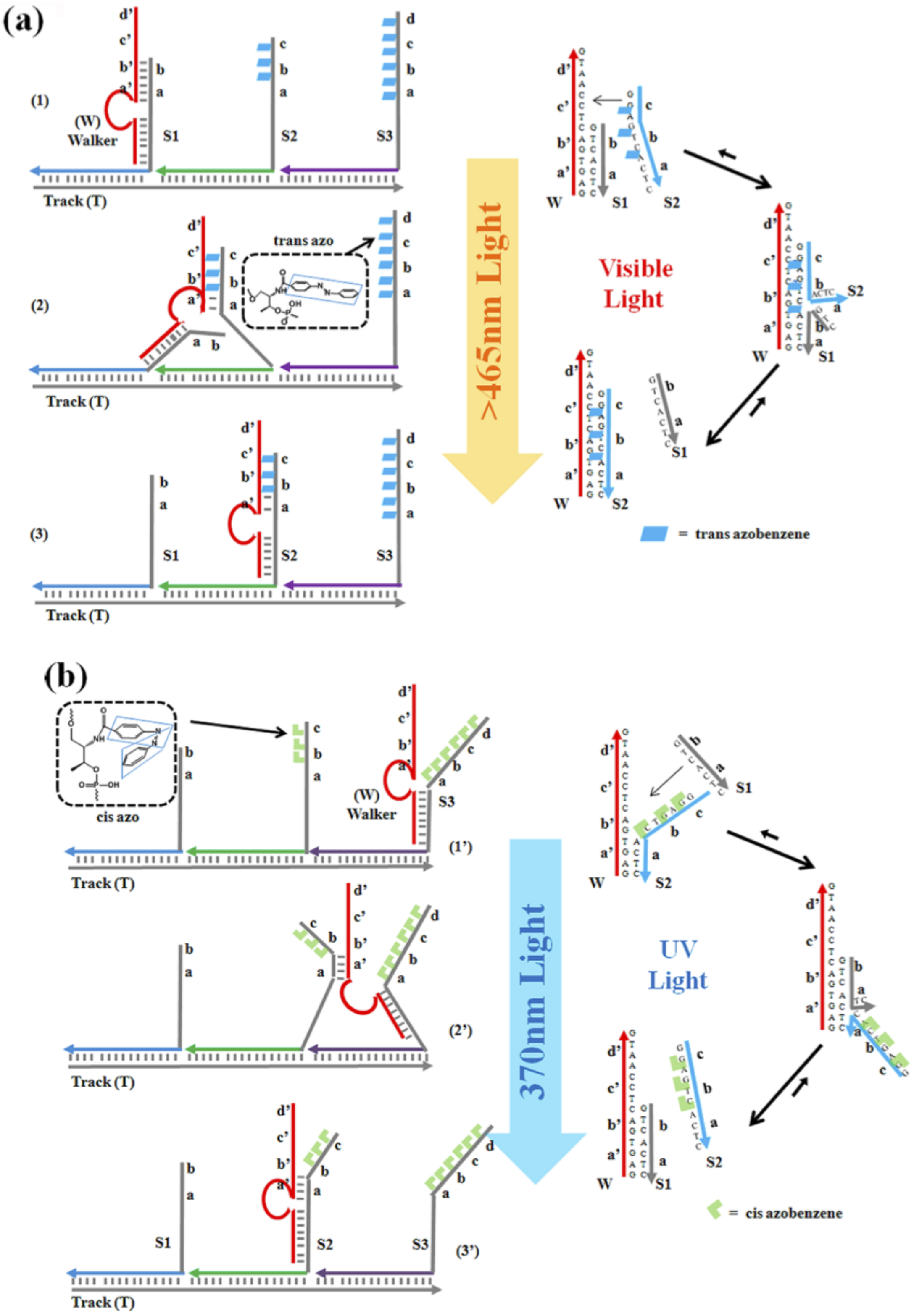


Figure 21: A reversible walker powered by visible [122]a) and UV b) light irradiation

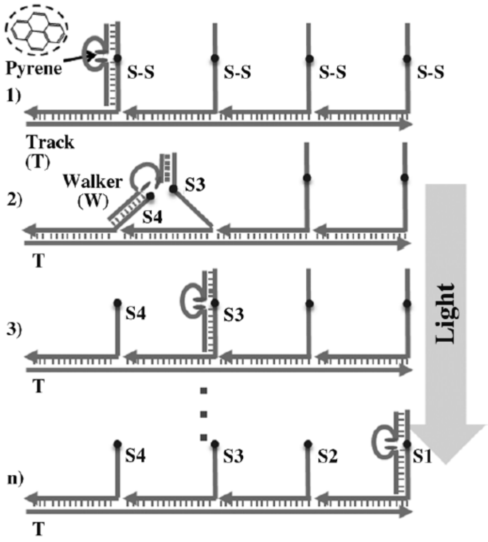


Figure 22: A Light driven Pyrene Assisted Autonomous Walker [122]

One of the more recent examples given in [122] is an autonomous and controllable light-driven DNA walking device through the use of pyrene-assisted photolysis. The walker, a single strand, is thought of as having two legs, one on each end. The 3' end is the shorter leg (7nt), while the 5' end has the longer leg (16nt). Both the legs are linked together by pyrene (Figure 22-Step 1)). The anchorage segments are linked together by a weak disulfide bond (black dot). Using a fluorometer, a light source at 350 nm was used to initiate photolysis. The pyrene molecule cleaves the disulphide bond in the anchorage site, and as a result, the 7nt segment on the anchorage site floats away. The short leg then hybridizes with the next anchorage site and through toehold mediated strand displacement, the long leg hybridizes to its complementary segment on the next anchorage site. This process is repeated upto step n, if there are n anchorage sites. To control the direction of movement, a 'burnt bridge' mechanism (discussed earlier) was used.

However, the limitations of this walker involve the fact that it is hard to control its actions due to its autonomy. You et al. [123] altered this walker to make another light-powered DNA walker with reversible movement, depending on the light's wavelength. In visible light, the walker moves forward, whereas UV light irradiation moves the walker backwards on a track. The anchorage site toeholds, also known as anchorage extender segments, have a different number of azobenzene molecules (one incorporated in every two bases of the toeholds). Azobenzene is a chiral molecule, and it switches its conformation from cis to trans if light having wavelength greater than 465 nm falls on it. The reverse it also true, namely it switches back to cis in the presence of light with wavelength less than 465nm.

The trans conformation of Azobenezene stabilizes the double helix, while the cis conformation destabilizes it. Hence, in visible light, the walker moves from left to right, via toehold mediated strand displacement; Figure 21a). In UV light however, azobenzene switches from trans-to-cis, destabilizing the helix, and the walker prefers to return to a more stable state, by reversing its direction, again moving via toehold mediated strand displacement; Figure 21b).

Discussions

DNA Walkers are stepping stones to constructing DNA systems that have dynamic behaviour, and are able to perform mechanical tasks at the nanoscale. By programming the directed movement of DNA molecules along a track, we can target the walker to deliver a cargo(s) reliably, send or receive information, and achieve nanoscale synthesis.

Multistep biosynthesis has been demonstrated by Liu et.al. [35]. They designed a DNA walker that performs the synthesis of a linear product, in the order of the anchorage sites on the DNA track. At each step, the walker builds on the product, by attaching the cargo at the current site to that of the prior sites. At the final step, a synthesized final chain of cargo's is obtained, not unlike mRNA translation.

3.5 DNA Dynamical Systems

The behavior of DNA complexes can be controlled via the action of DNA enzymes that act upon specific sequences of DNA strands [74], competitive DNA hybridization [131] or environmental changes such as pH or temperature [108]. One of the first applications of strand displacement to a DNA nanomachine was the molecular tweezer, by [124]. Dynamic behaviour of the tweezer is controllable, using strand displacement. Other types of devices include walkers, [70], [121], Molecular detectors [20], both in-vitro and in-vivo [59,104]. [18] also made nano structures within a cell. Ref. [33] made a molecular assembly line. These and many other devices have been constructed and studied.

3.6 DNA Tiling Assembly

3.6.1 Models of Tile Assembly

In this section, we give a brief review of the theoretical models of self-assembly. In DNA self-assembly, a tile is the basic unit of assembly, with the initial state consisting of a set of unordered tiles, and the final state consists of an assembly formed by some or all of these tiles sticking to one another. Theoretical models of tile assembly have been developed to study the computational properties of self-assembly. Tile Assembly dates back to 1961, when Wang [106] proposed the Wang domino problem. In his tiling model, a tile is a square with a glue on each side. For two tiles to attach, their abutting glues should be the same. Retroflexion and rotation of tiles are not permitted. The problem statement is: "Given a finite set of tiles, can they tile the entire plane". Infinite number of copies of each tile type are provided. A more general problem is: "Does there exist an algorithm that can decide if a finite set of tiles can tile the whole plane". It has been proved that such an algorithm does not exist, and that the tiling problem is undecidable. [9, 75]. The proof shows that any instance of Turing's halting problem can be reduced to an instance of the Wang domino problem. Since the halting problem is not decidable, hence the Wang domino problem is not decidable either.

Rothemund and Winfree proposed the abstract Tile Assembly Model(aTAM) in 2000 [78]. In aTAM, a tiling system is modeled as a quadruple $< T, s_0, \tau, g >$. (1) T here is the tile set. As in Wang's model, a tile in aTAM is a square with a glue on each side. All glues are from a glue set Σ. Retroflexion and rotation of tiles are forbidden. (2) s_0 is a special tile called the *seed tile*. At the beginning of the assembly process, only the seed tile is fixed at a particular position and the other tiles are floating in the plane. Assembly begins from the seed tile. (3) τ is the temperature of the system. When a tile is at a position, it will be fixed at that position if and only if the accumulative glue strength between this tile and tiles around it is not less than τ. (4) g is the glue strength function. It defines the glue strength between two glues. An assembly, that grows from the seed tile, can grow infinitely large, or it may terminate. A configuration is *terminal* if it cannot grow any more. This happens when there is no additional site for a tile to attach to at temperature τ. In Winfree's original definition, [78], the aTAM is formally defined in 2D, but it can be extended to multi-dimensions. The 2D-aTAM has been proved to be Turing Universal [117].

We begin by introducing some important definitions in aTAM. (1) The tile complexity of shapes [78]. The tile complexity of a shape is the minimum number of tile types that *uniquely* assemble that shape. (2) The time complexity of assembling a shape [3]. This models how fast a shape can be assembled. The assembly process can be modelled by a continuous Markov process. The states of the Markov process, are the different tile assembly configurations. To proceed from one state to another, a tile(s) attaches or detaches from the growing assembly. The rate of transition from state S_1 to state S_2 is dependent on the concentration of tile x if S_2 is formed by attaching x to S_1. The time taken to reach the terminal state, from the initial state is a random variable and time complexity is defined as the expected value of this random variable.

One important problem in aTAM is the tile complexity and time complexity of assembling a fixed shaped assembly. The fixed shape used by Winfree and Rothemund is an $N \times N$ square, where N is any positive integer. In other words, 'What is the minimum number of tile types needed to construct an $N \times N$ square at temperature τ?' and 'What is the minimum amount of time needed to construct an $N \times N$ square at temperature τ?'

Rothemund and Winfree give a construction to show that the upper bound of tile complexity of assembling $N \times N$ square in a τ=2 system is $O(logN)$ [78]. The basic idea of their construction is to use $O(logN)$ tile types to build the frame of an $N \times N$ square. Then, use a constant number of tile types to fill the frame. The frame has two components: a counter of size $N \times logN$ that grows towards the north and a diagonal of the $N \times N$ square that grows towards the northeast. They also proved a tile complexity lower bound $\Omega(\frac{logN}{loglogN})$ for the the same, using Kolmogorov Complexity. This lower bound is true for infinitely many N. The question is whether there exists a construction that can reach this lower bound to prove that it is tight.

In 2001, Adleman et al. [3] proved that the lower bound is indeed tight. Adleman et al. proposed a new construction of tile complexity $\Theta(\frac{logN}{loglogN})$ that reaches the lower bound proved by Rothemund and Winfree. What we should keep in mind is that their construction is a τ=3 construction. The trick to improve the tile complexity was the technique by which they encoded a number. Rothemund and Winfree encoded N in base 2 (binary), and this technique led to the $O(logN)$ tile complexity. In the new design by Adleman et al., they encoded the number in base $\Theta(\frac{logN}{loglogN})$. This strategy reduced the tile complexity to $\Theta(\frac{logN}{loglogN})$.

Adleman et al. also generalized Winfree's aTAM by incorporating time complexity to the original model. The construction of tile complexity $O(logN)$ from Rothemund and Winfree was proved to have $\Theta(NlogN)$ time complexity. Adleman et al. gave a time complexity of $\Theta(N)$ which is optimal for assembling an $N \times N$ square.

Many variants of the aTAM have been developed. Winfree developed the kinetic Tile Assembly Model(kTAM) [114] to model the kinetic properties of crystal growth. This model is based on four assumptions that are essential to it. Aggarwal et al. [4] developed generalized models including the multiple temperature model, flexible glue model and q-tile model. Sahu et al. [99] proposed the time-dependent glue strength model. In their model, the glue strength between two glues increases with time, until it reaches a maximum. The advantage of this model is that it can model catalysis and self-replication. Reif [71] proposed the Step-wise Assembly Model in 1999. In this model, the assembly process is done by multiple steps. At each step, a separate assembly with a different tile sets and different temperature is created. Only the terminal product of one step is given as input to the next step. The terminal product of final step is the terminal product of the whole system. We give more details of Reif's model in Section 3.6.6. In 2008, Demaine et al. proposed a generalized Step-wise Assembly Model called the *Staged Assembly Model* [19]. They also gave strategies to assemble shapes with $O(1)$ glues under their model. It has been proved that both the Step-wise Assembly Model and the Staged Assembly Model are Turing universal at temperature 1 [6].

Theoretical Results of Self-assembled Shapes All constructions above try to store the information of the target shape in the tile set. This implies the need for a larger tile set assemble a target shape of a larger size. However, it is not feasible to synthesize a large number of tile types in practice. There are many constraints to implementing large tile sets such as the length of the pad, spurious reactions among pads of same tile, sequence design, etc.

An alternate strategy is encoding the information of target shapes in the concentrations of tile types. By this strategy, we can reduce the tile complexity of some self-assembled shapes to $O(1)$ [21, 40]. Kao and Robert proposed a random assembly construction for approximate squares [40] by encoding the shape information in the tile concentration. By their construction, an $N \times N$ square can be assembled from a constant number of tile types with high approximation factor and high probability where N should be large enough. The question is whether there exists a construction of $O(1)$ tile complexity that assembles an exact $N \times N$ square with high probability. Doty published such a design in 2009 [21].

The problem of storing shape information in the tile concentration is that it is difficult to control the concentration precisely during the reaction process. Other methods to reduce the tile complexity, include attempts by Reif in 1999, the Step-wise Assembly Model [71]. Demaine proposed a generalized version of Step-wise Assembly Model called the *Staged Assembly Model* [19]. The basic idea of these two models is assembling the target shape by multiple steps. This strategy can reduce the tile complexity efficiently [19,38]. The cost is stage complexity and bin(or tube) complexity [19, 38].

3.6.2 Tile Assembly Experimental Methods

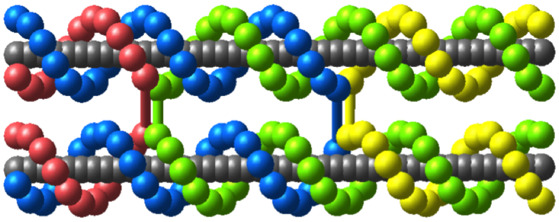


Figure 23: DX Tile

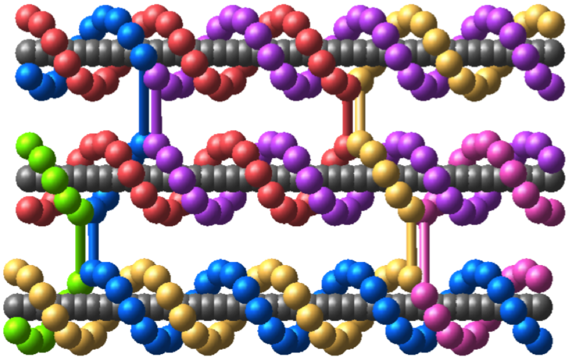


Figure 24: TX Tile

There are several DNA motifs that are used to design DNA tiles. These include DX and TX motifs [41,116] Figures 23, 24, DNA Holliday Junctions [25], and DNA origami [76]. These motifs are made stable by the incorporation of *crossover* junctions, where a DNA strand initially bound to a strand, crosses over and starts binding to another strand.

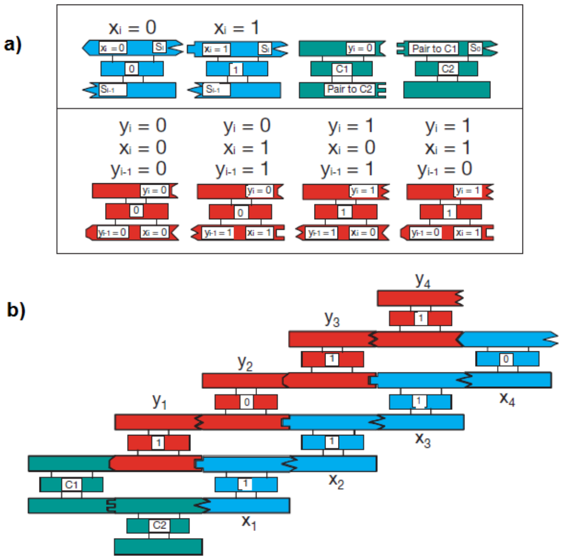


Figure 25: Calculation of cumulative XOR by self-assembly of DNA tiles. a)Component tiles. The three helical domains are drawn as rectangles, anked by sticky ends shown as geometrical shapes. The value of each tile is in the central rectangle. The meaning of each sticky end is also indicated. Shown are the two x tiles (blue), the four y tiles (red), and the initialization corner tiles, C1 and C2 (green). The y_i tiles are upside down from the x_i tiles. b) The values of each tile are the same as in a), and the sticky ends are the same. Note the complementarity of sticky ended association at each molecular interface. The operations are designed to proceed from lower left to upper right, because the x_i and C1 and C2 tiles have longer sticky ends than the y_i tiles. [56]

A motif attaches to a growing assembly via intermolecular contacts called *sticky ends*. These are complementary single stranded DNA, which bind to each other once in close proximity. Arbitrarily complex tile assemblies can be made using a small number of component tiles where each tile specifies an individual step of the computation. One of the first 1D experimental demonstrations after Adleman's Travelling Salesman [2] was the computation of a cumulative XOR function. The inputs were x_1, x_2, x_3, x_4, and the outputs were y_1, y_2, y_3, y_4 where $y_i = y_{i-1}$ XOR x_i and $y_1 = x_1$. A sample computation with input $x_1x_2x_3x_4 = 1110$ and output $y_1y_2y_3y_4 = 1011$ can be seen in Figure 25 b) in four steps, by the self assembly of DNA TX tiles [56]. Binary inputs are encoded in a DNA tile each, and the presence of a unique combination of input tiles results in the formation of a unique set of output tiles, seen in Figure 25 a).

A more complex construction was the set of 2D DNA Sierpinski Triangles made by Rothemund et. al [77]. They designed a 1D cellular automata, whose update rule computes a binary XOR, and the assembly results in a fractal pattern. A detailed design is in Figure 26. Atomic Force Microscope (AFM) images can be seen in Figure 27. These constructions give evidence that DNA can perform Turing Universal computation, and has the ability to implement any algorithmically computable system.

3.6.3 Tile Assembly Design & Simulation Software

CAD softwares help Molecular Architects in the design of DNA based tiles. To design the DNA sequences of these tiles, sequence symmetry minimization [87], is a common technique used to avoid spurious DNA strand interactions. Tools used in the design of such tiles are NanoEngineer [36], Tiamat [112], TileSoft [120] GIDEON [10]. Another tool, caDNAno, developed by Douglas et. al [23], primarily designed for the use of DNA origami, can also be used for the same.

Simulating the behaviour of DNA based tiles yields useful theoretical results, which can help predict the kind of errors we should expect when using DNA tiles for computation or construction. Winfree wrote a

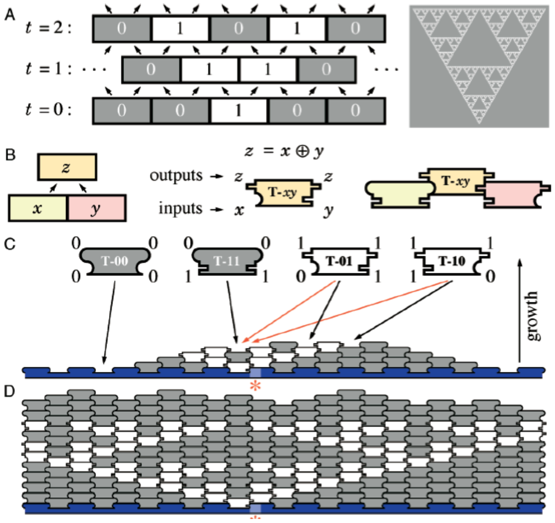


Figure 26: The XOR Cellular Automaton and Its Implementation by Tile-Based Self- Assembly. (A) Left: Three time steps of its execution drawn as a spacetime history. Cells update synchronously according to XOR by the equation shown. Cells at even time steps are interleaved with those at odd time steps; arrows show propagation of information. Right: the Sierpinski triangle. (B) Translating the spacetime history into a tiling. For each possible input pair, a tile T-xy that bears the inputs represented as shapes on the lower half of each side and the output as shapes duplicated on the top half of each side. (C) The four Sierpinski rule tiles, T-00, T-11, T-01, and T-10, represent the four entries of the truth table for XOR: 0 XOR 0 → 0, 1 XOR 1 → 0, 0 XOR 1 → 1, and 1 XOR 0 → 1. Lower binding domains on the sides of tiles match input from the layer below; upper binding domains provide output to both neighbors on the layer above. (D) Error-free growth results in the Sierpinski pattern. [77]

simulator Xgrow [114] as part of simulating the kTAM (kinetic TAM) model, with extensions to it from [5]. Another simulator is ISU TAS, developed by Patitz at Iowa State University [63].

3.6.4 One Dimensional DNA Tiling Assemblies

In this section, we will give a brief review of Probabilistic Linear Tile Assemble Model(PLTAM) and results under this model [12]. Assembling linear structure of given length is an important problem in self-assembly both theoretically and experimentally. Many complex structures can be made from linear structures. One question is how big the smallest tile set that assembles linear structure of length N is. If the linear assembly process is deterministic, it can be proved that the smallest tile set to assemble linear structure of length N is of size N.

Chandran et al. proposed a probabilistic linear tile assembly model [12], where “probabilistic” means that given a tile type, there may be multiple tile types that can attach to its western or eastern pad, and “linear” means that tiles in this model only have two pads: western pad and eastern pad. They also developed three schemes under PLTAM to assemble linear structure of expected length N with tile complexity $\Theta(logN)$, $O(log^2N)$ and $O(log^3N)$. All schemes are for equimolar standard linear assembly system. A linear assembly system is standard if and only if it is haltable, east-growing, diagonal and uni-seeded where “haltable” means that the assembly process can stop, “east-growing” means that the assembly process is from west to east, “diagonal” means that the glue strength between two glues is 0 if they are different and otherwise the strength is 1 if the glue is not null glue, and “uni-seeded” means that there is only one seed tile. It can be proved that standard systems can simulate a larger set of systems. The reason why they focus on equimolar system is that equimolar system is more practical.

The basic trick of designing these three schemes is simulating some well-designed stochastic processes by linear tile sets. One issue is that all three schemes do not have a sharp tail bound of assembly length for a general N. What has been proved is that the $O(log^3N)$ scheme has sharp tail bound for infinite many

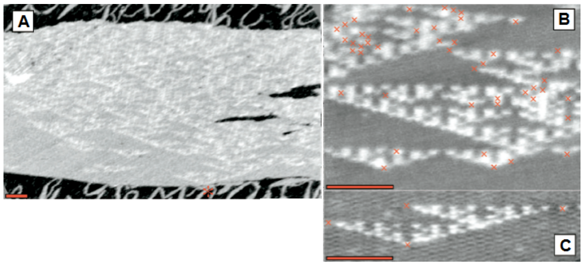


Figure 27: AFM Images of DNA Sierpinski Triangles. (A) A large templated crystal in a 5-tile reaction. A single 1in the input row (asterisk) initiates a Sierpinski triangle. (B) A zoomed in region of A containing roughly 1,000 tiles and 45 errors. (C) A close up of a Sierpinski Triangle. Red crosses in (B and (C) indicate tiles that have been identified (by eye) to be incorrect with respect to the two tiles from which they receive their input. Scale bars are 100 nm. [77]

N, and for large enough N, we can get sharp tail bound by dividing N into numerous identical chunks. Therefore, the open problem is whether there exists a scheme of $o(N)$ tile complexity with sharp tail bound of assembly length for general N.

3.6.5 Three Dimensional DNA Tiling Assemblies

3D assemblies are a natural extension to the construction of 2D assemblies. Cook et. al [16] ask the question: *What is the number of tile types needed, to create an $N \times N$ 2D square, using a 3D tile system?* In a 3D tile system, a Wang tile is replaced by a Wang cube, each cube having 6 faces, and hence 6 glues. They give a construction showing that using $O(logn)$ tile types (cubes in 3D) can be used to construct an nxn square, at temperature 1, as opposed to the temperature 2 result by Rothemund and Winfree [78].

Constructions have been done using 3D Wang assembly models, that are used to solve NP-complete problems such as Maximal Clique [53]. The computational power of this system is Turing Universal as suggested by Winfree in [117], since it is a superset of the 2D tiling problem, but the additional computing power of this system at different temperatures has still to be ascertained.

3.6.6 Step-Wise Tiling Assemblies

In this section, we will give a brief review of Reif's Step-wise Assembly Model [71]. Reifs model reduces the tile complexity by dividing the assembly process into steps. It has been proved that this strategy can efficiently reduce tile complexity [19,38]. A tiling system under the Step-Wise Assembly Model is a quadruple $< T, s_0, g, \Phi >$, where $T = T_1 \cup T_2 ... \cup T_m$, and $\Phi = \{\tau_1, \tau_2, ..., \tau_m\}$. The assembly process has m steps. In the first step, tiles in T_1 including s_0 assemble in tube 1 at temperature τ_1 for a long enough time. The terminal product of step 1 is delivered to tube 2. Add T_2 to tube 2 and let tiles (or supertiles) in tube 2 assemble at temperature τ_2 for a long enough time. This process continues until the step m is complete. The terminal product of step m is the terminal product of the whole system. Maňuch et al. proposed schemes of $O(1)$ tile complexity to assemble an $N \times N$ square [38]. Behsaz et al. proved that Step-wise Assembly Model is Turing universal at temperature 1 [6]. Demaine et al. proposed Staged Assembly Model in 2008 [19]. It is a generalized version of Step-wise Assembly Model.

3.6.7 Activatable Tiles: Transition of State Across Tiles

Experimental demonstration of the Turing universal tile based algorithmic DNA self-assembly has been limited by significant assembly errors. An important class of errors, called *co-ordinated growth errors*, occur when an incorrect tile binds to a growing assembly even when some of its pads are mismatched with its neighbors. Activatable DNA tiles, introduced originally by [55], employ a protection/deprotection strategy to strictly enforce the direction of tiling assembly growth that prevents these errors, ensuring the robustness of the assembly process. Tiles are initially inactive, meaning that each tile's pads are protected and cannot bind with other tiles. After an activation event, the tile transitions to an active state and its pads are exposed, allowing further growth.

3.6.8 Error-Correction in Tiling Assemblies

In this section, we give a brief review of error correction techniques in self-assembly. Tile Assembly has been beset with different types of errors. These are broadly classified into Growth errors and Nucleation Errors. Growth errors happen when a tile incorrectly binds to a growing assembly, and propagates this error by causing other incorrect tiles to attach. Nucleation errors happen when an assembly does not grow from the seed tile, but it randomly starts growing using other tiles from the tile set.

Errors in assembly are usually reduced by two approaches. (1) Reducing the errors by optimizing the reaction environment. [114]. By this approach, the inherent error rate is reduced. (2) Reducing the errors by optimizing the tile set design [73, 115]. However, this approach can only reduce the amount of errors in the assembled pattern and cannot improve the inherent error rate. Winfree proposed an error resilient scheme by the second approach called *Proofreading Tile Sets* [115]. The basic idea of his scheme is using a $k \times k$ block of tiles to replace each tile in the original tile set. Although it does bring down the error from ϵ to ϵ^k, this scheme has two problems. (1) It amplifies the assembly size. (2) It can only reduce the errors in the output, while the errors inside the assembly pattern are dismissed.

Reif et al. proposed a compact error resilient scheme [73]. The basic idea of their scheme is adding proofreading portions to pads. Their scheme does not amplify the assembly size and can reduce the errors inside the assembly pattern. The problem of this scheme is that it is not suitable to general case.

4 Summary

DNA computing has matured since Adleman used it to solve a 7-node instance of the Hamiltonian path problem. Adlemans original aim was to leverage the inherent parallelism of DNA for computation, and its high information density for storage. DNA Computation started with solving a set of "hard" problems interesting to computer scientists, but has since evolved to have other applications. Scientists have used DNA devices for a variety of uses, such as in biosensors for molecular detection, diagnostic applications, in nanofabrication, for structure determination, and as potential drug delivery devices.

It is important to analyzing the power of any computational device. This is in order to be able to judget it against other devices with similar functionality, and to get an estimate of the theoretical limits of any device. This is why CRNs (Section 1.3) are important, to model the DNA strand displacement techniques that are inherent to quite a few DNA devices.

DNA can be used to perform complex nanofabrication tasks. Both periodic structures such as lattices [34,86], and non-periodic structures such as a smiley face, world map [76], have been constructed using DNA. These structures can be used as scaffolds to place gates, transistors, routing elements at precise locations, and thus create circuits from the bottom-up, as opposed to traditional top-down lithographic techniques. However, as we discuss in Section 3.6.8, errors are inherent to scaffold construction, and even without errors, the size of the scaffolds that can be created, pales in comparison to that achievable by silicon-based lithography. A parallel line of research, is not in using DNA as a scaffold, but in using DNA itself to construct the circuit elements. These designs are quite promising as discussed in Section 3.1. Using such devices, it would be possible to have medical diagnostic biochemical circuits, with applications similar to Benenson's DNA Doctor [90].

DNA devices that are involved in detection have been a steady focus area among scientists. There exists a large set of techniques that has led to the construction of these devices. DNA machines typically respond to external stimuli such as pH, salt concentration, temperature, and other signalling molecules (DNA, proteins), that leads to a change in their state. This mechanical state change is detectable by techniques such as FRET, and has led to the development of sensors. In addition, logical operations can be performed on DNA motifs, and coupled with sensors, complex biological systems can be designed drug delivery systems.

The programmable self-assembly of DNA has also enabled us to construct bio-chemical circuits that can perform synthesis of complex molecules. For e.g., DNA-directed synthesis has been shown to be achievable using a variant of DNA walker systems [33,35]. Being programmable, the synthesis of any oligomer can be controlled at every step, not unlike the central dogma that dictates protein synthesis.

One of Seemans original goals of constructing lattices of DNA, was to use it to determine the structure of other foreign (protein) molecules via x-ray crystallography [88]. Most lattices created with DNA have a lot of defects, hence not making DNA very suitable for this use. However, with the latest advances in structural DNA nanotechnology, that goal, and others seem to be finally in sight.

4.1 Defining Terms

Antiparallel: Each DNA strand has a 5' and 3' end. When two strands bind to each other, the 5' end of one strand is adjacent to the 3' end of the other strand.
Base Pairing: The binding of A (Adenine) to T (Thymine), or G (Guanine) to C (Cytosine).
Secondary Structure: The entire set of base pairs formed in a given set of DNA strands, where each pair may lie on either the same strand, or on different strands.
Locality: The property of different reactants being in close proximity, due to structural or environmental constraints, causing them to react faster.
Random Walk: A random walk in 1D, is the path taken by an object, in either one of two directions: left or right. At each step, the next direction is determined by an independent fair toin coss.
DNAzyme: A molecule made of DNA, that possesses enzymatic activity, as also shown by protein enzymes.
Aptamer: DNA molecule that binds to a select target molecule such as other nucleic acids, proteins etc.
Nucleotide: A molecule made up of three components, a 5 carbon (pentose) sugar molecule, a nitrogenous

base (one of A,T,G,C), and a phosphate group.
Sensitivity: Probability of an actual target (strand) being correctly identified.

References

[1] Rivka Adar, Yaakov Benenson, Gregory Linshiz, Amit Rosner, Naftali Tishby, and Ehud Shapiro. Stochastic computing with biomolecular automata. *Proceedings of the National Academy of Sciences of the United States of America*, 101(27):9960–9965, 2004.

[2] Leonard Adleman. Molecular Computation of Solutions to Combinatorial Problems. *Science*, 266(5178):1021–1024, 1994.

[3] Leonard Adleman, Qi Cheng, Ashish Goel, and Ming-Deh Huang. Running Time and Program Size for Self-Assembled Squares. *Symposium on Theory of Computing*, pages 740–748, 2001.

[4] Gagan Aggarwal, Michael Goldwasser, Ming-Yang Kao, and Robert Schweller. Complexities for Generalized Models of Self-Assembly. *Symposium on Discrete Algorithms*, pages 880–889, 2004.

[5] Anshul Chaurasia and Sudhanshu Dwivedi and Prateek Jain and Manish K. Gupta. XTile: An Error Correction Package for DNA Self-Assembly. *Foundations of Nanoscience*, 6, 2009.

[6] Bahar Behsaz, Ján Maňuch, Ladislav Stacho . Turing universality of step-wise and stage assembly at temperature 1. *LNCS*, 7433:1–11, 2012.

[7] Jonathan Bath, Simon Green, and Andrew Turberfield. A Free-Running DNA Motor Powered by a Nicking Enzyme. *Angewandte Chemie International Edition*, 44(28):4358–4361, 2005.

[8] Jeremy M. Berg, John L. Tymoczko, and Lubert Stryer. *Biochemistry, 7th edition.* W.H. Freeman Publishers, Dec 2010.

[9] Robert Berger. The Undecidability of the Domino Problem. *Memoirs of American Mathematical Society*, 66:1–72, 1966.

[10] Jeffrey J. Birac, William B. Sherman, Jens Kopatsch, Pamela E. Constantinou, and Nadrian C. Seeman. Architecture with GIDEON, a program for design in structural DNA nanotechnology. *Journal of molecular graphics & modelling*, 25(4):470–480, 2006.

[11] David Blicq. DNA Hairpin. `http://xnet.rrc.mb.ca/davidb/dna.htm`.

[12] Harish Chandran, Nikhil Gopalkrishnan, and John Reif. The Tile Complexity of Linear Assemblies. *International Colloquium on Automata, Languages and Programming*, pages 235–253, 2009.

[13] Harish Chandran, Nikhil Gopalkrishnan, Bernard Yurke, and John Reif. Meta-DNA: Synthetic Biology via DNA Nanostructures and Hybridization Reactions. *Journal of the Royal Society Interface*, 2012.

[14] Ho-Lin Chen, David Doty, and David Soloveichik. Deterministic Function Computation with Chemical Reaction Networks. *DNA Computing*, 2012.

[15] Yi Chen, Seung-Hyun Lee, and Chengde Mao. A DNA Nanomachine Based on a Duplex-Triplex Transition. *Angewandte Chemie International Edition*, 43(40):5335–5338, 2004.

[16] Matthew Cook, Yunhui Fu, and Robert T. Schweller. Temperature 1 self-assembly: Deterministic assembly in 3d and probabilistic assembly in 2d. In *SODA*, pages 570–589, 2011.

[17] David Y. Zhang and Bernard Yurke. A DNA superstructure-based replicator without product inhibition. *Natural Computing*, 5(2):183–202, 2006.

[18] Delebecque, Camille J. and Lindner, Ariel B. and Silver, Pamela A. and Aldaye, Faisal A. Organization of Intracellular Reactions with Rationally Designed RNA Assemblies. *Science*, 333(6041):470–474, 2011.

[19] Erik Demaine, Martin Demaine, Sándor Fekete, Mashhood Ishaque, Eynat Rafalin, Robert Schweller, and Diane Souvaine. Staged Self-assembly: Nanomanufacture of Arbitrary Shapes with $O(1)$ Glues. *DNA Computing*, pages 1–14, 2007.

[20] Robert Dirks and Niles Pierce. Triggered Amplification by Hybridization Chain Reaction. *Proceedings of the National Academy of Sciences of the United States of America*, 101(43):15275–15278, 2004.

[21] David Doty. Randomized Self-Assembly for Exact Shapes. *Foundations of Computer Science*, pages 85–94, 2009.

[22] Shawn Douglas, James Chou, and William Shih. 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, 2007.

[23] Shawn M. Douglas, Adam H. Marblestone, Surat Teerapittayanon, Alejandro Vazquez, George M. Church, and William M. Shih. Rapid prototyping of 3D DNA-origami shapes with caDNAno. *Nucleic Acids Research*, 37(15):5001–5006, 2009.

[24] EH Ekland, JW Szostak, and DP Bartel. Structurally complex and highly active RNA ligases derived from random RNA sequences. *Science*, 269(5222):364–370, 1995.

[25] Tsu-Ju Fu, Yuk-Ching Tse-Dinh, and Nadrian C. Seeman. Holliday junction crossover topology. *Journal of Molecular Biology*, 236(1):91 – 105, 1994.

[26] Daniel G. Gibson, John I. Glass, Carole Lartigue, Vladimir N. Noskov, Ray-Yuan Chuang, Mikkel A. Algire, Gwynedd A. Benders, Michael G. Montague, Li Ma, Monzia M. Moodie, Chuck Merryman, Sanjay Vashee, Radha Krishnakumar, Nacyra Assad-Garcia, Cynthia Andrews-Pfannkoch, Evgeniya A. Denisova, Lei Young, Zhi-Qing Qi, Thomas H. Segall-Shapiro, Christopher H. Calvey, Prashanth P. Parmar, Clyde A. Hutchison, Hamilton O. Smith, and J. Craig Venter. Creation of a bacterial cell controlled by a chemically synthesized genome. *Science*, 329(5987):52–56, 2010.

[27] Manoj Gopalkrishnan. Catalysis in Reaction Networks. *Bulletin of Mathematical Biology*, 73(12):2962–2982, 2011.

[28] Manoj Gopalkrishnan and Nikhil Gopalkrishnan. Exquisite Detection with Chemical Circuits. *In preparation*, 2012.

[29] Nikhil Gopalkrishnan. *Engineering Exquisite Nanoscale Behavior with DNA*. PhD thesis, Duke University, 2012.

[30] Nikhil Gopalkrishnan, Harish Chandran, and John Reif. High-Fidelity DNA Hybridization using Programmable Molecular DNA Devices. *DNA Computing*, 2010.

[31] Simon Green, Jonathan Bath, and Andrew Turberfield. Coordinated Chemomechanical Cycles: A Mechanism for Autonomous Molecular Motion. *Physical Review Letters*, 101(23), 2008.

[32] Simon Green, Daniel Lubrich, and Andrew Turberfield. DNA Hairpins: Fuel for Autonomous DNA Devices. *Biophysical Journal*, 91(8):2966–2975, 2006.

[33] Hongzhou Gu, Jie Chao, Shou-Jun Xiao, and Nadrian Seeman. A Proximity-based Programmable DNA Nanoscale Assembly Line. *Nature*, 465(7295):202–205, 2010.

[34] Yu He, Yi Chen, Haipeng Liu, Alexander Ribbe, and Chengde Mao. Self-Assembly of Hexagonal DNA Two-Dimensional (2D) Arrays. *Journal of the American Chemical Society*, 127(35):12202–12203, 2005.

[35] Yu He and David Liu. Autonomous Multistep Organic Synthesis in a Single Isothermal Solution Mediated by a DNA Walker. *Nature Nanotechnology*, 5(11):778782, 2010.

[36] Nanorex Inc. NanoEngineer-1. http://www.nanoengineer-1.com/content/, 2006.

[37] Iaki Sainz de Murieta and Alfonso Rodrguez-Patn. DNA biosensors that reason. *Biosystems*, 109(2):91–104, 2012.

[38] Ján Maňuch, Ladislav Stacho and Christine Stoll. Step-wise tile assembly with a constant number of tile types. *Natural Computing*, 11(3):535–550, 2012.

[39] Wendy K. Johnston, Peter J. Unrau, Michael S. Lawrence, Margaret E. Glasner, and David P. Bartel. RNA-Catalyzed RNA Polymerization: Accurate and General RNA-Templated Primer Extension. *Science*, 292(5520):1319–1325, 2001.

[40] Ming-Yang Kao and Robert Schweller. Randomized Self-Assembly for Approximate Shapes. *International Colloquium on Automata, Languages and Programming*, pages 370–384, 2008.

[41] Thomas LaBean, Hao Yan, Jens Kopatsch, Furong Liu, Erik Winfree, John Reif, and Nadrian Seeman. Construction, Analysis, Ligation, and Self-Assembly of DNA Triple Crossover Complexes. *Journal of the American Chemical Society*, 122(9):1848–1860, 2000.

[42] Matthew Lakin, Simon Youssef, Filippo Polo, Stephen Emmott, and Andrew Phillips. Visual DSD: a design and analysis tool for DNA strand displacement systems. *Bioinformatics*, 27(22):3211=3213, 2012.

[43] Richard Lavery. DNA Structure. www.phys.ens.fr/ monasson/Houches/Lavery/L1.ppt.

[44] Harvey Lederman, Joanne Macdonald, Darko Stefanovic, and Milan N. Stojanovic. Deoxyribozyme-Based Three-Input Logic Gates and Construction of a Molecular Full Adder. *Biochemistry*, 45(4):1194–1199, 2006. PMID: 16430215.

[45] Jean Louis Leroy, Kalle Gehring, Abdelali Kettani, and Maurice Gueron. Acid multimers of oligodeoxycytidine strands: Stoichiometry, base-pair characterization, and proton exchange properties. *Biochemistry*, 32(23):6019–6031, 1993. PMID: 8389586.

[46] Tim Liedl, Michael Olapinski, and Friedrich C. Simmel. A Surface-Bound DNA Switch Driven by a Chemical Oscillator. *Angewandte Chemie International Edition*, 45(30):5007–5010, 2006.

[47] Chenxiang Lin, Ralf Jungmann, Andrew M. Leifer, Chao Li, Daniel Levner, George M. Church, William M. Shih, and Peng Yin. Submicrometre geometrically encoded fluorescent barcodes self-assembled from DNA. *Nature Chemistry*, page 832839, 2012.

[48] Dongsheng Liu and Shankar Balasubramanian. A Proton-Fuelled DNA Nanomachine. *Angewandte Chemie International Edition*, 42(46):5734–5736, 2003.

[49] Dongsheng Liu, Andreas Bruckbauer, Chris Abell, Shankar Balasubramanian, Dae-Joon Kang, David Klenerman, and Dejian Zhou. A Reversible pH-Driven DNA Nanoswitch Array. *Journal of the American Chemical Society*, 128(6):2067–2071, 2006.

[50] Huajie Liu, Yun Xu, Fengyu Li, Yang Yang, Wenxing Wang, Yanlin Song, and Dongsheng Liu. Light-Driven Conformational Switch of i-Motif DNA. *Angewandte Chemie International Edition*, 46(14):2515–2517, 2007.

[51] Yi Lu and Juewen Liu. Functional DNA Nanotechnology: Emerging Applications of DNAzymes and Aptamers. *Current Opinion in Biotechnology*, 17(6):580–588, 2006.

[52] Kyle Lund, Anthony J. Manzo, Nadine Dabby, Nicole Michelotti, Alexander Johnson-Buck, Jeanette Nangreave, Steven Taylor, Renjun Pei, Milan N. Stojanovic, Nils G. Walter, Erik Winfree, and Hao Yan. Molecular robots guided by prescriptive landscapes. *Nature*, 465(7295):206–210, May 2010.

[53] Jingjing Ma, Li Jia, and Yafei Dong. DNA 3D Self-assembly Algorithmic Model to Solve Maximum Clique Problem. *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, 3(3):41–48, 2011.

[54] J. Mai, I. M. Sokolov, and A. Blumen. Directed particle diffusion under burnt bridges conditions. *Physical Review E*, 64(1):011102, June 2001.

[55] Urmi Majumder, Thomas LaBean, and John Reif. Activatable Tiles: Compact, Robust Programmable Assembly and Other Applications. *DNA Computing*, 2007.

[56] Chengde Mao, Thomas Labean, John Reif, and Nadrian Seeman. Logical Computation Using Algorithmic Self-assembly of DNA Triple-crossover Molecules. *Nature*, 407:493–496, 2000.

[57] Mireya L. McKee, Phillip J. Milnes, Jonathan Bath, Eugen Stulz, Rachel K. O'Reilly, and Andrew J. Turberfield. Programmable One-Pot Multistep Organic Synthesis Using DNA Junctions. *Journal of the American Chemical Society*, 134(3):1446–1449, 2012.

[58] Souvik Modi, Swetha M. G, Debanjan Goswami, Gagan D. Gupta, Satyajit Mayor, and Yamuna Krishnan. A DNA nanomachine that maps spatial and temporal pH changes inside living cells. *Nature Nanotechnology*, 4(5):325–330, 2009.

[59] Souvik Modi and Yamuna Krishnan. A Method to Map Spatiotemporal pH Changes Inside Living Cells Using a pH-Triggered DNA Nanoswitch. *Methods in molecular biology (Clifton, N.J.)*, 749:61–77, May 2011.

[60] L A Moran, K G Scrimageour, H R Horton, R S Ochs, and J D Rawn. *Biochemistry, second edition.* Neil Patterson Publishers, distributed by Prentice Hall Inc., 1994.

[61] Richard A. Muscat, Jonathan Bath, and Andrew J. Turberfield. A programmable molecular robot. *Nano Letters*, 11(3):982–987, March 2011.

[62] Neil Campbell and Jane Reece and Lawrence Mitchell. *Biology, 5th edition.* Benjamin Cummings, 1999.

[63] Matthew J. Patitz. Simulation of self-assembly in the abstract tile assembly model with isu tas. *CoRR*, abs/1101.5151, 2011.

[64] R Pei and S K Taylor. Deoxyribozyme-based autonomous molecular spiders controlled by computing logic gates. *IPCBEE Proceedings*, 2009.

[65] Renjun Pei, Elizabeth Matamoros, Manhong Liu, Darko Stefanovic, and Milan N. Stojanovic. Training a molecular automaton to play a game. *Nature Nanotechnology*, 5(11):773–777, 2010.

[66] Fritz Pohl and Thomas Jovin. Salt-induced co-operative Conformational Change of a Synthetic DNA: Equilibrium and Kinetic Studies with poly(dG-dC). *Angewandte Chemie International Edition*, 67(3):375–396, 1972.

[67] Lulu Qian and Erik Winfree. A Simple DNA Gate Motif for Synthesizing Large-scale Circuits. *DNA Computing*, pages 70–89, 2009.

[68] Lulu Qian and Erik Winfree. Scaling up Digital Circuit Computation with DNA Strand Displacement Cascades. *Science*, 332(6034):1196–1201, 2011.

[69] Lulu Qian, Erik Winfree, and Jehoshua Bruck. Neural Network Computation with DNA Strand Displacement Cascades. *Nature*, 475(7356):368–372, 2011.

[70] John Reif. The Design of Autonomous DNA Nano-mechanical Devices: Walking and Rolling DNA. *DNA Computing*, pages 439–461, 2003.

[71] John Reif. Local Parallel Biomolecular Computation. *International Journal of Unconventional Computing*, 2013. Special Issue: Biomolecular Computing - From Theory to Practical Applications. Invited Paper.

[72] John Reif and Sudheer Sahu. Autonomous Programmable Nanorobotic Devices Using DNAzymes. *DNA Computing*, pages 66–78, 2007.

[73] John Reif, Sudheer Sahu, and Peng Yin. Compact Error-Resilient Computational DNA Tiling Assemblies. *DNA Computing*, pages 293–307, 2004.

[74] RJ Roberts and Kenneth Murray. Restriction Endonucleases. *CRC Crit. Rev. Biochem*, 4(2):124–164, 1976.

[75] Raphael Robinson. Undecidability and Nonperiodicity for Tilings of the Plane. *Inventiones Mathematicae*, 12:177–209, 1971.

[76] Paul Rothemund. Folding DNA to Create Nanoscale Shapes and Patterns. *Nature*, 440:297–302, 2006.

[77] Paul Rothemund, Nick Papadakis, and Erik Winfree. Algorithmic Self-Assembly of DNA Sierpinski Triangles. *PLoS Biology*, 2:424–436, 2004.

[78] Paul Rothemund and Erik Winfree. The Program-Size Complexity of Self-Assembled Squares. *Symposium on Theory of Computing*, pages 459–468, 2000.

[79] Wolfram Saenger. *Principles of nucleic acid structure.* Springer-Verlag, New York. 556 pp, 1984.

[80] Kensaku Sakamoto, Hidetaka Gouzu, Ken Komiya, Daisuke Kiga, Shigeyuki Yokoyama, Takashi Yokomori, and Masami Hagiya. Molecular Computation by DNA Hairpin Formation. *Science*, 288:1223–1226, 2000.

[81] Kensaku Sakamoto, Daisuke Kiga, Ken Momiya, Hidetaka Gouzu, Shigeyuki Yokoyama, Shuji Ikeda, Hiroshi Sugiyama, and Masami Hagiya. State Transitions by Molecules. *Biosystems*, pages 81–91, 1999.

[82] Rebecca Schulman, Bernard Yurke, and Erik Winfree. Robust self-replication of combinatorial information via crystal growth and scission. *Proceedings of the National Academy of Sciences*, 2012.

[83] Georg Seelig, David Soloveichik, David Yu Zhang, and Erik Winfree. Enzyme-Free Nucleic Acid Logic Circuits. *Science*, 314(5805):1585–1588, 2006.

[84] Georg Seelig, Bernard Yurke, and Erik Winfree. DNA Hybridization Catalysts and Catalyst Circuits. In *DNA*, volume 3384, pages 329–343. Springer, 2004.

[85] Georg Seelig, Bernard Yurke, and Erik Winfree. Catalyzed Relaxation of a Metastable DNA Fuel. *Journal of the American Chemical Society*, 128(37):12211–12220, 2006.

[86] Nadrian C. Seeman. Nucleic Acid Junctions and Lattices. *Journal of Theoretical Biology*, 99:237–247, 1982.

[87] Nadrian C. Seeman. De novo design of sequences for nucleic acid structural engineering. *J Biomol Struct Dyn.*, 8(3):3211=3213, 1990.

[88] Nadrian C. Seeman. Construction of Three-Dimensional Stick Figures from Branched DNA . *DNA and Cell Biology.*, 10(7):475–486, SEPTEMBER 1991.

[89] Hiroyuki Sekiguchi, Ken Komiya, Daisuke Kiga, and Masayuki Yamamura. A realization of DNA molecular machine that walks autonomously by using a restriction enzyme. In Max Garzon and Hao Yan, editors, *DNA Computing*, volume 4848, pages 54–65. Springer Berlin / Heidelberg, 2008.

[90] Ehud Shapiro and Yaakov Benenson. Bringing DNA Computers to Life. *Scientific American*, 17(3):40–47, 2006.

[91] William Sherman and Nadrian Seeman. A Precisely Controlled DNA Biped Walking Device. *Nano Letters*, 4:1203–1207, 2004.

[92] Jong-Shik Shin and Niles Pierce. A Synthetic DNA Walker for Molecular Transport. *Journal of American Chemical Society*, 126(35):1083410835, 2004.

[93] Richard R. Sinden. *DNA Structure And Function.* Academic Press Inc., California, 1994, 398 pp, 1994.

[94] David Soloveichik, Matthew Cook, Erik Winfree, and Jehoshua Bruck. Computation with Finite Stochastic Chemical Reaction Networks. *Natural Computing*, 7:615–633, 2008.

[95] David Soloveichik, Georg Seelig, and Erik Winfree. DNA as a Universal Substrate for Chemical Kinetics. *Proceedings of the National Academy of Sciences*, 107:5293–5398, 2010.

[96] Milan Stojanovic and Darko Stefanovic. A Deoxyribozyme-based Molecular Automaton. *Nature Biotechnology*, 21(9):1069–1074, August 2003.

[97] Milan N. Stojanovic, Tiffany E. Mitchell, and Darko Stefanovic. Deoxyribozyme-Based Logic Gates. *Journal of the American Chemical Society*, 124(14):3555–3561, 2002.

[98] Milan N. Stojanovic, Stanka Semova, Dmitry Kolpashchikov, Joanne Macdonald, Clint Morgan, and Darko Stefanovic. Deoxyribozyme-based ligase logic gates and their initial circuits. *Journal of the American Chemical Society*, 127(19):6914–6915, 2005.

[99] Sudheer Sahu , Peng Yin , John H. Reif. A self-assembly model of time-dependent glue strength. *DNA11*, 2005.

[100] Ye Tian, Yu He, , Yi Chen, Peng Yin, and Chengde Mao. A DNAzyme That Walks Processively and Autonomously along a One-Dimensional Track. *Angewandte Chemie International Edition*, 44(28):4355–4358, 2005.

[101] Ye Tian and Chengde Mao. Molecular Gears: A Pair of DNA Circles Continuously Rolls against Each Other. *Journal of American Chemical Society*, 126(37):1141011411, 2004.

[102] Andrew Turberfield, James Mitchell, Bernard Yurke, Allen Mills, Myrtle Blakey, and Friedrich Simmel. DNA Fuel for Free-Running Nanomachines. *Physical Review Letters*, 90(11), 2003.

[103] Suvir Venkataraman, Robert Dirks, Paul Rothemund, Erik Winfree, and Niles Pierce. An Autonomous Polymerization Motor Powered by DNA Hybridization. *Nature Nanotechnology*, 2:490–494, 2007.

[104] Suvir Venkataramana, Robert Dirks, Christine Uedab, and Niles Pierce. Selective Cell Death Mediated by Small Conditional RNAs. *Proceedings of the National Academy of Sciences*, 107(39):16777–16782, 2010.

[105] Chunyan Wang, Jingsong Ren, and Xiaogang Qu. A stimuli responsive DNA walking device. *Chemical Communications*, 47(5):1428, 2011.

[106] Hao Wang. Proving Theorems by Pattern Recognition II. *Bell Systems Technical Journal*, 1961.

[107] Tong Wang, Ruojie Sha, Remi Dreyfus, Mirjam E. Leunissen, Corinna Maass, David J. Pine, Paul M. Chaikin, and Nadrian C. Seeman. Self-replication of information-bearing nanoscale patterns. *Nature*, 478(7368):225–228, 2011.

[108] Wenxing Wang, Yang Yang, Enjun Cheng, Manchun Zhao, Haifeng Meng, Dongsheng Liu, and Dejian Zhou. A pH-driven, reconfigurable DNA nanotriangle. *Chem. Commun.*, pages 824–826, 2009.

[109] Zhen-Gang Wang, Johann Elbaz, and Itamar Willner. DNA machines: Bipedal walker and stepper. *Nano Letters*, 11(1):304–309, January 2011.

[110] Zhen-Gang Wang, Johann Elbaz, and Itamar Willner. A dynamically programmed DNA transporter. *Angewandte Chemie International Edition*, 51(18):43224326, 2012.

[111] Shelley F J Wickham, Masayuki Endo, Yousuke Katsuda, Kumi Hidaka, Jonathan Bath, Hiroshi Sugiyama, and Andrew J Turberfield. Direct observation of stepwise movement of a synthetic molecular transporter. *Nature Nanotechnology*, 6:166–169, February 2011.

[112] Sean Williams, Kyle Lund, Chenxiang Lin, Peter Wonka, Stuart Lindsay, and Hao Yan. *Tiamat: A Three-Dimensional Editing Tool for Complex DNA Structures.* 2008.

[113] Itamar Willner, Bella Shlyahovsky, Maya Zayats, and Bilha Willner. DNAzymes for Sensing, Nanobiotechnology and Logic Gate Applications. *ChemInform*, 39(33), 2008.

[114] Erik Winfree. Simulations of Computing by Self-Assembly. Technical report, California Institute of Technology, 1998.

[115] Erik Winfree and Renat Bekbolatov. Proofreading Tile Sets: Error Correction for Algorithmic Self-Assembly. *DNA Computing*, pages 126–144, 2003.

[116] Erik Winfree, Furong Liu, Lisa Wenzler, and Nadrian Seeman. Design and Self-assembly of Two-dimensional DNA Crystals. *Nature*, 394:539–544, 1998.

[117] Erik Winfree, Xiaoping Yang, and Nadrian Seeman. Universal Computation via Self-assembly of DNA: Some Theory and Experiments. *DNA Based Computers II, DIMACS*, 44:191–213, 1996.

[118] Hao Yan, Thomas LaBean, Liping Feng, and John Reif. 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, 2003.

[119] Peng Yin, Harry Choi, Colby Calvert, and Niles Pierce. Programming Biomolecular Self-assembly Pathways. *Nature*, 451(7176):318–322, 2008.

[120] Peng Yin, Bo Guo, Christina Belmore, Will Palmeri, Erik Winfree, Thomas LaBean, and John Reif. TileSoft: Sequence Optimization Software for Designing DNA Secondary Structures. Technical report, Duke and California Institute of Technology, 2004.

[121] Peng Yin, Hao Yan, Xiaoju Daniell, Andrew Turberfield, and John Reif. A Unidirectional DNA Walker Moving Autonomously Along a Linear Track. *Angewandte Chemie International Edition*, 116(37):5014–5019, 2004.

[122] Mingxu You, Yan Chen, Xiaobing Zhang, Haipeng Liu, Ruowen Wang, Kelong Wang, Kathryn R. Williams, and Weihong Tan. An autonomous and controllable light-driven DNA walking device. *Angewandte Chemie International Edition*, 51(10):24572460, 2012.

[123] Mingxu You, Fujian Huang, Zhuo Chen, Ruo-Wen Wang, and Weihong Tan. Building a nanostructure with reversible motions using photonic energy. *ACS Nano*, 6(9):7935–7941, September 2012.

[124] Bernard Yurke, Andrew Turberfield, Allen Mills, Friedrich Simmel, and Jennifer Neumann. A DNA-fuelled Molecular Machine Made of DNA. *Nature*, 406(6796):605–608, 2000.

[125] Joseph Zadeh, Conrad Steenberg, Justin Bois, Brian Wolfe, Marshall Pierce, Asif Khan, Robert Dirks, and Niles Pierce. NUPACK: Analysis and Design of Nucleic Acid Systems. *Journal of Computational Chemistry*, 32(1):170–173, 2010.

[126] David Zhang, Sherry Xi Chen, and Peng Yin. Optimizing the specificity of nucleic acid hybridization. *Nature Chemistry*, 4:208–214, 2012.

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

[128] David Yu Zhang and Georg Seelig. DNA-based fixed gain amplifiers and linear classifier circuits. In *Proceedings of the 16th international conference on DNA computing and molecular programming*, pages 176–186, Berlin, Heidelberg, 2011. Springer-Verlag.

[129] David Yu Zhang and Erik Winfree. Control of DNA Strand Displacement Kinetics Using Toehold Exchange. *Journal of the American Chemical Society*, 131(48):17303–17314, 2009.

[130] Jianping Zheng, Jens Birktoft, Yi Chen, Tong Wang, Ruojie Sha, Pamela Constantinou, Stephan Ginell, Chengde Mao, and Nadrian. From Molecular to Macroscopic via the Rational Design of a Self-assembled 3D DNA Crystal. *Nature*, 461(7260):74–78, 2009.

[131] Zipursky, S Lawrence, and James Darnell. *Molecular Cell Biology (4th ed.)*. Freeman & Co, New York, NY 2000, 1084 pp, 1993.

[132] Michael Zuker. MFOLD Web Server for Nucleic Acid Folding and Hybridization Prediction. *Nucleic Acids Research*, 31(13):3406–3415, 2003.

Additional Readings

[133] Jonathan Bath and Andrew Turberfield. DNA Nanomachines. *Nature Nanotechnology*, 2:275–284, 2007.

[134] Harish Chandran, Nikhil Gopalkrishnan, Sudhanshu Garg, and John Reif. *Biomolecular Computing Systems - From Logic Systems to Smart Sensors and Actuators*. Wiley-VCH, 2012. Invited Chapter.

[135] Anne Condon. Designed DNA Molecules: Principles and Applications of Molecular Nanotechnology. *Nature Reviews Genetics*, 7:565–575, 2006.

[136] Zhaoxiang Deng, Yi Chen, Ye Tian, and Chengde Mao. A Fresh Look at DNA Nanotechnology. *Nanotechnology: Science and Computation*, pages 23–34, June 2006.

[137] Thomas LaBean, Kurt Gothelf, and John Reif. Self-Assembling DNA Nanostructures for Patterned Molecular Assembly. *Nanobiotechnology II*, pages 79–97, 2007.

[138] Adam J. Ruben and Laura F. Landweber. The past, present and future of molecular computing. *Nature Reviews Molecular Cell Biology*, 1:69–72, 2000.

[139] Ruojie Sha, Xiaoping Zhang, Shiping Liao, Pamela Constantinou, Baoquan Ding, Tong Wang, Alejandra Garibotti, Hong Zhong, Lisa 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 Lukeman, William Sherman, Natasha Jonoska Chengde Mao, and Nadrian Seeman. Structural DNA Nanotechnology : Molecular Construction and Computation. *Unconventional Computing*, pages 20–31, 2005.

[140] Erik Winfree. DNA Computing by Self-Assembly. *NAE's The Bridge*, 33:31–38, 2003.

[141] Hao Yan, Peng Yin, Sung Ha Park, Hanying Li, Liping Feng, Xiaoju Guan, Dage Liu, John Reif, and Thomas LaBean. Self-assembled DNA Structures for Nanoconstruction. *American Institute of Physics Conference Series*, 725:43–52, 2004.

4.2 Acknowledgements

This work was supported by NSF Grants CCF- 1217457 and CCF-1141847.