Beyond Biology: Designing a New Mechanism for Self-Replication and Evolution at the Nanoscale

Rebecca Schulman
Department of Physics
University of California Berkeley
Berkeley, CA 94720
rschulman@berkeley.edu

ABSTRACT

As biology demonstrates, evolutionary algorithms are an extraordinarily powerful way to design complex nanoscale systems. While we can harness the biological apparatus for replicating and selecting DNA sequences to evolve enzymes and to some extent, organisms, we would like to build replication machinery that would allow us to evolve designs for a much wider variety of materials and systems. Here we describe work that uses techniques from the new field of structural DNA nanotechnology to modularly design nanoscale components that together can be assembled into a system for self-replicating a new form of chemical information or genome, and thus for evolving a new type of chemical sequence.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and genetics

General Terms

Design

Keywords

Nanoscale Systems, Self-Replicating Molecular Machines

1. INTRODUCTION

A major current scientific challenge is to learn how to design materials with nanoscale features and to exploit the unique properties of materials available at this scale. Some of the benefits of nanoscale engineering are widely familiar: the increasing density with which we can organize transistors on a chip is largely responsible for the increasing speed of our computers. But there are many other cases where nanoscale features change the properties of materials in ways that we can exploit: for example, the optical and electronic properties of nanometer-scale crystals and wires can be dependent on their dimensions [27, 23].

GECCO'11, July 12–16, 2011, Dublin, Ireland.

Further, we expect that much of the engineering possibility at the nanoscale remain to be discovered. Perhaps the most dramatic demonstration of the benefits that could be gain by having molecular-scale control over matter is biology. Inside cells, the production, transformation, and functions of individual molecules are precisely controlled. These features are essential to the capacity of biology for self-replication, self-healing and metamorphosis. By having similar control over molecular synthesis and nanoscale geometry in synthetic systems, it should be possible to achieve these features as well as many others in synthetic materials.

Biology's sophisticated architecture is the product of the Darwinian evolution of a genomic sequence, an organism's program for growth and function. Evolution is therefore an extraordinarily powerful design strategy for nanoscale materials and devices. And evolutionary algorithms for molecular design such as SELEX for evolving RNA molecules with catalytic function [22, 16] and directed evolution for evolving functional proteins [2] have been more successful than comparable rational design strategies.

But there is currently an important limitation on our ability to solve molecular design problems using Darwinian evolution: we can only replicate, and thus evolve, DNA or RNA sequences. This replication can take place in cells or in the test tube, but in either case the form of the information replicated, a sequence of nucleic acids, is the same. While changing the representation of the information being evolved in an *in silico* process is straightforward, translating the representation of chemical information is extremely challenging.

Biology has figured out some mechanisms for accomplishing this representation change: the "central dogma" of molecular biology is that DNA can be transcribed into an RNA sequence and then translated into an amino acid sequence, which folds into a protein; a set of proteins can then together synthesize other molecules. But there is no obvious way to translate DNA sequence information into instructions for autonomously constructing many structures we might be interested in, such as silicon-based circuitry.

The chemical translation problem is not theoretically difficult, but difficult in practice: even trying to augment the genetic code to include one new kind of amino acid has been a major technical challenge [41]. In the past decade, there have been initial attempts to build a more general *in vitro* apparatus for translating DNA sequences into synthesis recipes [19, 20] that might allow us to evolve a much wider array of products.

And at the same time, new technologies for designing libraries of possible sequences (*i.e.* controlling the mutation operation) have improved the process of evolutionary design of proteins [21]. But because of the challenges inherent to chemical translation, we might ask more generally whether evolving a sequence of 4 bases is the most efficient way to solve all molecular design problems. In software, both the representation of the information being evolved (as well as how this information is used to produce the function being evolved) and the mechanism of mutation are important for efficiently solving design problems using genetic and evolutionary algorithms [25, 31]. If instead of evolving DNA sequences that are replicated in cells or by enzymes extracted from cells, we could design systems for molecular replication and mutation the way we can design evolutionary algorithms, we might be able to solve a much wider variety of chemical design problems and build new nanoscale materials with evolution.

We are still far from being able to design arbitrary molecular machinery capable of processes as complex as self-replication *de novo*, and we know only a little about which aspects of replication and evolution in molecular systems are the major determinants of their efficiency [15, 7]. But important progress is being made: we are learning how to design modular molecular components and how to combine these components into functional molecular machines. And from these modular parts we can begin to build devices for chemical self-replication.

Here we give an account of the development of components for a new system for molecular information replication and of how evolution could proceed in such a system. We first describe how we can design molecular components made from synthetic DNA, (short DNA sequences made chemically in the laboratory rather than by enzymes within cells). The component DNA sequences of these structures, arbitrary sequences of A's, T's, G's and C's, can be designed and optimized on the computer. We then describe how we can use synthetic DNA components, called DNA tiles, in a self-assembly process. This self-assembly process is analogous in some sense to solving a jigsaw puzzle and performs computation during assembly.

That is, for any given computation, we can design a set of DNA tiles that executes that computation via self-assembly. We describe how to design a set of DNA tiles that copies a sequence of information during assembly. The assembly process propagates the sequence; and when mechanical forces fracture an assembly, new sites on the fragmented assembly become available where the sequence can be propagated, increasing the rate of sequence propagation. Cycles of sequence propagation (assembly) and fragmentation exponentially replicate the sequences. We describe how to implement this process experimentally and how evolution would occur in this system.

The processes we can design using synthetic DNA continue to increase in both complex and variety. There are now several proposals for building systems for sequence replication (and thus Darwinian evolution) from synthetic DNA components [47, 24]. As the set of systems available for molecular sequence replication and evolution grows, we will have new opportunities to both learn about evolution of physical systems and to design efficient algorithms for evolution and selection in these new systems.

2. DNA TILES AND ALGORITHMIC SELF-ASSEMBLY

DNA is most familiar as the material in which our genome is stored. What underlies DNA's capacity for storing and replicating information is its propensity for Watson-Crick complementary DNA bases to hybridize and form double-helical DNA. Recently, DNA's sequence specific binding capacity has become an engineering tool: it is possible to design a sequence and its complement and to know that these two sequences will bind but that they will not interact with other DNA molecules in the environment.

In 1982, Nadrian Seeman described how synthetic DNA might be used for nanoscale-construction. Seeman imagined using DNA molecules as programmable molecular tinkertoys that would self-assemble into designed structures because the complementary regions of the designed sequences would hybridize while other sequences would not react. He described how we might make branched DNA structures, and thus program the formation of 2- and 3-dimensional assemblies [37]. As Seeman described it, nanotechnology could happen the way it does in biology: autonomously – we would simply design the sequences, synthesize them, and put them together in a test tube and wait.

Designing a system of DNA molecules has turned out to be more tractable than the design of other types of complex molecular systems: the rate of DNA hybridization and the stability of base-paired DNA are generally predictable in polynomial time [48], and the double-helical structure of hybridized DNA is well-characterized and largely independent of the particular based-paired sequence [8]. These properties have enabled the design of extended 2- and 3-dimensional structures [45, 29, 4], programmed molecular machines [46, 5] and active structures [43, 46, 14] via the design of a set of DNA molecules and their relative abundances.

A DNA "tile" (Figure 1a) is a primitive for nanoscale construction [17, 45]. A DNA tile consists of a double-stranded "core" and 4 single-stranded "sticky ends." Tiles attach to each other via sticky end hybridization and can form extended two-dimensional lattices [45]. In principle, the arrangement of tile types within the lattices that form can be designed by designing appropriate DNA tile sticky end logic, a process akin conceptually to designing the pieces of a jigsaw puzzle and their interlocking nubs (Figure 1b). Given a desired sticky end logic, we can design and synthesize a set of DNA sequences that assemble into tiles that implement this logic (e.g. [45, 30, 3]).

Complex patterns can be constructed from DNA tiles efficiently by a technique known as algorithmic self-assembly [42]. The basic premise of algorithmic self-assembly is that an object is constructed *algorithmically*, that is by executing a program.

Algorithmic self-assembly has its roots in the tiling problem, the question of whether a given set of shapes can tile the plane, which is undecidable [39, 40, 6]. Using observations derived from the hardness of plane tiling, Winfree described a set of tiles and a constructive method for their assembly that executes a computer program [42, 43].

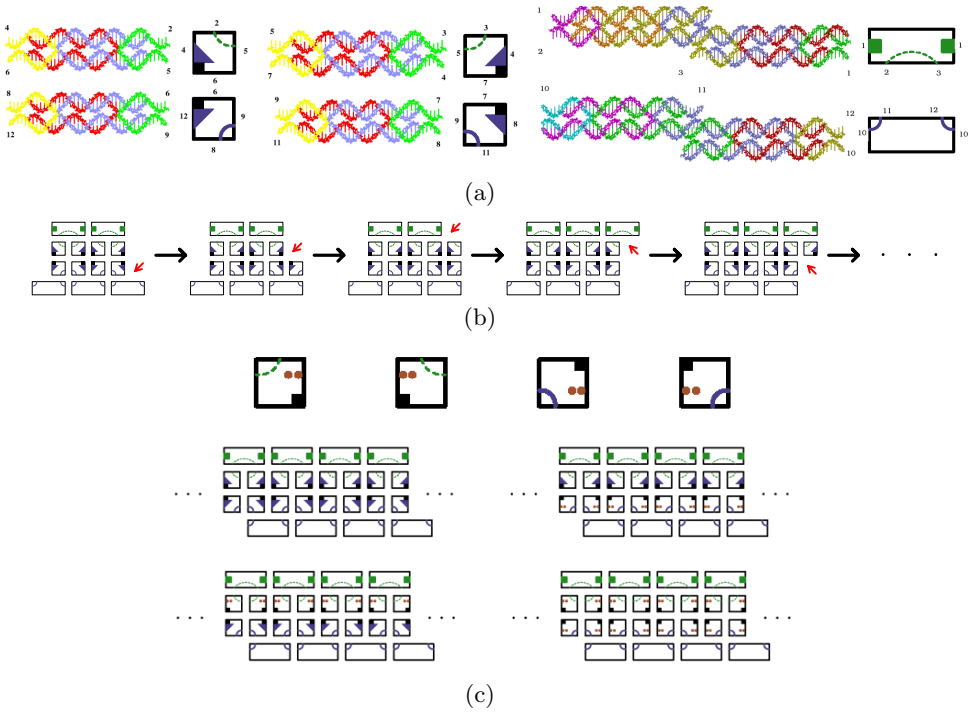


Figure 2: Zig-zag tiles. (a) The basic zig-zag tile set. Each square and rectangle shown is a logical representation of the molecule shown to its left. (b) Zig-zag growth. At each growth step, a new tile may be added at the location designated by the small arrow. Two alternating tile types in each row enforce the placement of the double tiles on the top and bottom, ensuring that growth occurs in a zig-zag pattern. Although only growth on the right end of the molecule is shown here, growth occurs simultaneously on both ends of the assembly. (c) The tile set shown in Figure 2b forms only one type of assembly. A tile set consisting of the tiles in (b) and the four tiles shown here allows four types of assemblies to be formed. The vertical column of each type contains a crystal's 2-bit binary sequence.

In Winfree's construction, growth of a tile crystal begins from a seed tile or structure whose sticky ends encode the initial state of a computation. Under physical conditions where tiles can attach to the seed only by two sticky ends simultaneously (*i.e.* just cooler than the melting temperature of the crystal), the growth of a DNA tile crystal, or lattice, can in principle simulate the execution a 1-dimensional blocked cellular automaton, and therefore perform universal computation. Intuitively, the two sticky ends a tile must match in order to attach to a growing crystal are "input" states to a cellular automaton and the remaining two sticky ends are the "output" of a single computing step. Since growth can continue indefinitely, arbitrarily long computations can be performed. Notably, the entire history of a computation is stored in the arrangement of tile types within the assembled crystal. In many cases this arrangement may form a useful structure that is difficult to assemble by other means [13].

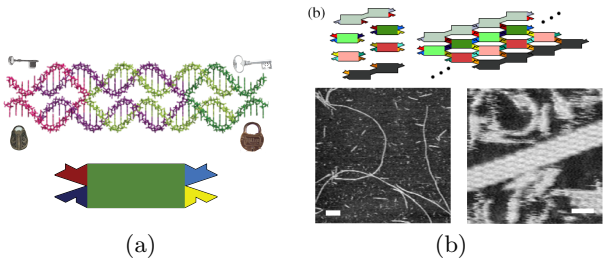
The assembly of the designed structure requires that at each step of assembly a valid tile, *i.e.* a tile that matches two sticky end binding sites simultaneously, be added to the crystal. However, in initial experiments [30] as many as 1%-10% of attachments were *errors*, or not valid—only one of the "input" edges of the tile matched the available inputs on the growing crystal. The wrong logical operation was being performed at those sites.

As would be expected of a computation in which 1–10% of the primitive operations were computed incorrectly, the patterns that formed were generally not the designed patterns.

The error rate can be reduced by logically redesigning the tiles to perform the same computation during assembly, but more robustly. "Proofreading" tile sets [44, 12, 28, 38] transform a tile set by replacing each individual tile with a $k \times k$ block of tiles, exponentially reducing seeded growth errors with respect to the size of the block. Along with the improvement of the structure where computation begins, the "seed" [4] and new techniques to prevent growth that does not begin from a seed, proofreading techniques allowed assembly to proceed much more accurately, *i.e.* with error rates as low as 1 in 1000 tiles. Structures such as Sierpinski gaskets [30, 18] and "binary counters" [3, 4] have been assembled using these techniques.

3. SELF-REPLICATING DNA CRYSTALS

In 1966, Graham Cairns-Smith proposed a simple mechanism by which polytypic clay crystals (clays that can take on one of many crystal structures) could replicate information in the absence of biological enzymes [9, 10]. Some polytypic clay crystals contain discrete layers, each of which contain molecules of a particular identity or orientation.



(a) (b)

Figure 1: DNA tiles and tile nanostructures. (a) A DNA tile is a nanoscale construction primitive. Top, a molecular model of a tile that contains short DNA molecules. Each strand is depicted in a different color. Bottom, a schematic shows the effective shape of a tile along with the logic of its sticky ends. Tile "cores" (e.g. the green portion of the schematic tile shown here) are double-stranded; the assembled core maximizes the number of Watson-Crick complementary base pairs between the component strands and is therefore a favorable configuration. Single-stranded "sticky ends" (the colored claws in the schematic) function as locks and keys: they specifically hybridize (*i.e.* bind) to complementary sticky end sequences on other tiles. (b) Tiles designed to form a 4-tile-wide ribbon, and atomic force micrographs of the ribbons, which assembled as designed. Scale bars are 500 nm (left) and 25 nm (right) (image from [33], copyright Proceedings of the National Academy of Sciences, USA).

A cross-section of such a crystal can contain an information-bearing sequence. Cairns-Smith proposed that crystal growth could extend the layers, copying the sequence (the crystal's genotype). Occasionally, physical forces could break a crystal apart. Because crystals replicate their genotype many times during growth, splitting of a crystal can yield multiple pieces, each containing at least one copy of the information-bearing sequence. Cycles of growth and fragmentation could therefore allow a sequence to be exponentially amplified.

We have adapted Cairns-Smith's ideas about spontaneous information replication in crystals to *design* a system for self-replication using DNA tiles as crystal monomers [32]. A simple set of DNA tiles can form *zig-zag crystals* that can propagate information during growth [33, 4]. The tiles shown in Figure 2a form the zig-zag crystal shown in Figure 2b. Matching rules determine which tile fits where. Under conditions where each tile addition must form two or more sticky end bonds (Figure 2a), growth is constrained to occur in a zig-zag pattern. It is easy to confirm that under such conditions, there is always a unique tile that may be added on each end of the ribbon.

Zig-zag crystals are designed so that under conditions where a tile must attach to a crystal by at least two bonds, growth produces one new row at a time (*i.e.* one copy of a sequence) and continued growth repeatedly copies a sequence. The requirement that a tile must attach by two bonds means that a tile being added must match both its vertical neighbor (another tile that is part of the new column being assembled), and its horizontal neighbor (in a previously assembled row).

Several tiles might match the label on the vertical neighbor, but because tiles must make two correct bonds in order to join the assembly, only a tile that also matches the label on the horizontal neighbor can be added. The tile being added in the new column must therefore correspond to the one in the previous column. As a result, information is inherited through templated growth. The set of tiles formed by adding the tiles in Figure 2c to those shown in Figure 2b can propagate one of four strings. Additional tiles may be added to the set of tiles in Figures 2b and 2c to create a tile set that can propagate arbitrary binary sequences.

The growth of a zig-zag DNA crystal increases the number of copies of the original information present in the ribbon but does not change the rate at which new copies of the sequence are produced. The rate of copying can be sped up by breaking the crystals. With each new crystal that is created by breakage, two new "growth fronts" become available where tiles can attach and information can be copied. Repeated cycles of growth and breakage exponentially amplify an initial piece of information. Occasionally, a tile matching only one bond rather than two will join the assembly, resulting in occasional copying errors, which are also inherited. If errors happen during copying, which they will under almost any achievable condition [43], and crystals with particular sequences grow faster than others, then evolution can occur.

4. SELECTION IN PHYSICAL SYSTEMS

In general, in an evolutionary or genetic algorithm a population is generated and afterwards some portion of the individuals is selected on the basis of their fitness. This subpopulation is used to create a population for the next generation via mutation and/or recombination. In a physical system the process of filtering and creation of a population for the next generation must be physically realizable, which is currently a strong limitation. Many types of fitness that we would like to select for, such as determining whether a molecule has a particular catalytic function, are difficult to measure in practice, and the partitioning of molecules or species based on their fitness is also challenging experimentally. While molecular "tricks" can sometimes permit autonomous selection of fit individuals [16], there are no general methods for evolution and selection based on function.

If we want to build novel systems for the evolution and selection of molecules or other physical entities, therefore, we will also need to develop ways to make this selection process easier. In biology, the desired function is the capacity to reproduce quickly with respect to other individuals in a population. Could we tie function to this capacity in artificial systems? To answer this question we must first understand why some species might replicate more quickly than others in a given self-replication process. Below we examine why some DNA tile sequences might be replicated more quickly than others, and consider as a result what selection processes for "fit" DNA tile sequences might be feasible.

5. EVOLUTION OF DNA CRYSTALS FOR FAST GROWTH: THE ROYAL ROAD

A selection process in a physical self-replicating system involves both an environment (a set of resources for growth, their chemistry and the ambient physical conditions) and an initial population of organisms (sequences).

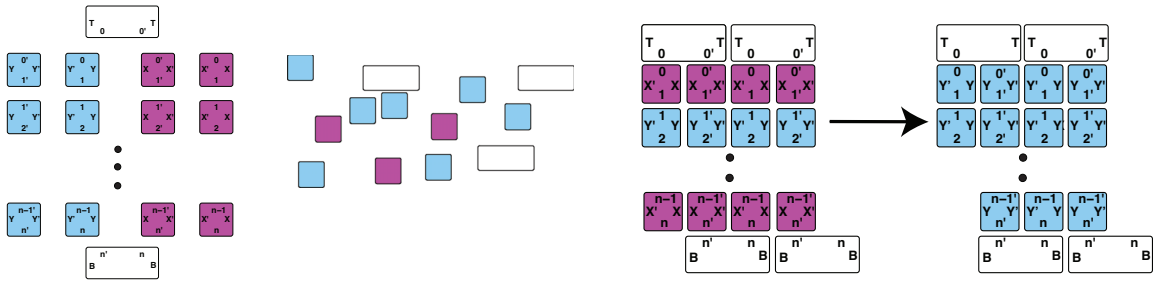


Figure 3: Royal Road Selection. (a) For a DNA tile ribbon containing sequences of width n, the Royal Road tile set contains $4n + 2$ tile types. Matching sticky ends have identical labels. Each position of the sequence contains either a cyan time (from the left group of tile types) or magenta tile (from the right group of tile types). (b) An environment where cyan tile types are present in higher concentrations than magenta tile types. (c) Selection in the environment in (b) favors sequences containing cyan tiles, since cyan tiles will be added to crystals faster than magenta tiles.

In a DNA tile replication process, the environment includes a set of DNA tiles. The set of DNA tiles determines the set of sequences which may be copied and the "chemistry" of the system, *i.e.*, the rules by which tiles bind to each other. A particular arrangement of DNA tiles is the information that is propagated in these experiments, the genotype; it is the organism being evolved. The phenotype of a sequence is its replication rate in the environment. In this section we first describe a tile set that allows many kinds of sequences to grow and then how selection pressure results from physical conditions in which the concentration of tile types differ.

A DNA crystal grows by adding tiles. Tiles come in contact with the crystal as the crystals and tiles diffuse randomly in the aqueous solution where growth occurs. Generally this growth takes place in a well-mixed reaction vessel, *i.e.* the density of crystals and monomers is on average uniform across the reaction container. In this case, the higher the concentration (*i.e.* density in solution) of a tile type that the vessel contains, the more quickly a tile of that type will contact a crystal where it can be legally added. Therefore, one simple selection pressure results from a difference in concentration between tile types used to copy sequence information: assemblies with sequences containing tile types present at high concentrations will grow faster than assemblies with sequences containing tile types present at very low concentrations.

A tile set in which one of two bits can be propagated at each of n sequence positions is shown in Figure 3a. Let X_i and Y_i be the two tile types that can be propagated at sequence position i. If Y_i's concentration is higher than X_i's concentration in solution, as suggested by the illustration in Figure 3b, the resulting fitness landscape resembles the simplest case of a well-studied problem in genetic algorithms, the "royal road" [26]. The growth rate of a crystal is proportional to the number of Y's in the sequence being propagated. For each position i, as long as the concentration of Y_i is higher than the concentration of X_i, sequences containing only Y_i tiles will be fitter and quickly dominate the population during a selection process (Figure 3c).

6. EVOLUTION OF DNA CRYSTAL ALGORITHMS

The previous section demonstrates how the scarcity of tile resources can lead to selection. But it does not address the question of how this selection could be used to evolve or improve a useful function of a molecular system: in the Royal Road process as we described it, the evolution process is a straightforward optimization problem with a known solution; no function or algorithm is being discovered.

If in contrast the sequence being evolved were a template or directive for an algorithm or device, the evolution process could select for functional behavior. To achieve such functional evolution it is necessary to define the language, or representation, of the information being evolved and the process of translating this information into a particular function.

How could we make the information being replicated functional? DNA crystals, as described in Section 2, can compute during growth as well as copy information. We can use this capacity to build sequences that function as programs. In fact, any program, no matter how complex, can be selected for [34, 35]. Thus, DNA crystals can in principle evolve powerful and complex functions. We review the mechanisms by which such selections can occur here.

As we described in Section 2, DNA crystals can perform a computation via the attachment of tiles to a growing crystal. A tile that can favorably attach at a growth site must match two labels at the growth site, the "input" labels. This simultaneous matching of two input labels is an elementary computing step. The other two labels on the attaching tile, the output labels, determine which tiles can fit in subsequent growth sites, so that information about the state of the computation is transmitted during growth.

Collectively, these tile attachments can simulate a Turing machine [42] where the initial state of the computation is determined by the structure of the seed where tile assembly begins. It is also possible to build a set of tiles that function as a universal Turing machine – the structure of the initial inputs on the seed determine which computation occurs during growth [34, 11].

In principle, such a tile set can be expanded to make a tile set that builds ribbons that have two parts – a segment that runs a program on the universal Turing machine, and a segment that makes copies of this program [34]. Such a zig-zag ribbon tile set would be a sort of "universal alphabet," with which we could build crystals that simultaneously store a program (its genome), and run it. During replication, the program's source code would be inherited, and in an evolution process that used this tile set, crystals containing particularly fit programs would be selected for.

How could a program make a crystal fit? First, the execution of crystal programs can build algorithmic patterns with potentially interesting features [13] that we could test via an artificial selection process. If we attached small devices to individual tiles, a program that built a binary counter might produce a pattern suitable for templating a demultiplexer circuit, for example [13]; other patterns might arrange molecules or nanoparticles into a combinatorial ensemble of interesting geometries. These assembled patterns could have optical, electronic or chemical functionality that could be selected for (given an available selection protocol), just as chemical functionality is currently selected for in SELEX or directed evolution experiments.

A tile program could also be a control system for adaptively sensing and responding to the environment. As we described in Section 5, the most basic reason for fitness is rapid growth, and crystals which use tile types that are abundant in the environment grow rapidly: a tile t is added at an average rate $\frac{k_f}{[t]}$ where k_f is a tile-independent rate constant, and $[t]$ is the concentration (density in solution) of tile t. More generally, if we disregard the frequency of fragmentation, the fitness of a crystal is proportional to the time it takes to grow a crystal layer [35], which is the sum of the times it takes to add each new tile in the layer. Thus, each tile addition makes a contribution to a crystal's fitness.

A fit crystal control program would be a program that could learn what tile types are abundant and then adopt the growth process to use as many of the most abundant tile types as possible. One way for a tile program to continually use abundant (as opposed to rare) tile types would be for the growing crystal executing the program to read information about whether tiles are abundant or rare at specified growth sites where multiple tile types could attach. The program could then use this input to determine which other tile types are abundant and thus should be used for computation. Such a program could be viewed as a sort of "metabolism" for crystals that figures out what nutrients are available and uses the available nutrients for energy and growth, in a process akin to metabolic sensing and response by biological cells. This kind of "crystal" control system sounds primitive, but in principle it could be arbitrarily complex: because crystals can simulate a Turing machine, they can assemble a program that senses and responds to any computable correlation between the abundances of tile types over time. If the correlations between tile type concentrations were very complex, then a very complex tile program to compute and take advantage of these correlations would evolve.

This tile set and evolutionary process (the changing concentrations of tile types over tile) could be a model system for studying evolution in non-biological molecular systems: we have a quantitative model of crystal behavior and the system as a whole and we have control over the concentrations of each tile type. In contrast, in biological systems we do not have control over many variables that are important to fitness, and the system dynamics are largely not understood: even the best-understood organisms produce hundreds of proteins whose functions are not known [1].

And while tile concentrations are not generally quantities that have immediate real-world interest, we could include modules in the growth environment that translate signals of other types into tile concentrations [36]. These translation systems would function as separate components, *i.e.* as molecular sensors that as output either produced or used up tiles, thus changing their concentrations. In a more sophisticated tile-based replication system, arrangements of tiles could themselves function as sensors and thus have function.

7. CONCLUSIONS

DNA tile crystal growth and scission is a novel synthetic mechanism for molecular sequence self-replication. In principle, evolution in tile crystal systems is as computationally rich as evolution in any system: if the mutation rate during crystal growth could be made arbitrarily low, then eventually any program, no matter how complex, can evolve if it is the most fit program for the environment.

It may thus be that for physical systems, the capacity to perform universal computation and tie this computation in some way to the environment may be sufficient for open-ended evolution in a self-replicating system. In practice the speed of evolution and selection is also vital: if an evolutionary optimization process took more time than the age of the universe to complete, it would be of no practical interest. Thus what is needed is a study of how to quickly and robustly evolve solutions to problems of interest.

The challenge of evolving these structures in the laboratory will teach us new things about how to encode evolutionary processes in physical, as opposed to purely computational systems. The DNA crystals described here replicate molecular information in one way. In the future we will broaden our library of mechanisms for self-replicating systems which will allow to grow closer to engineering evolutionary algorithms for a variety of molecular design problems. It will also allow us to examine the trade-offs in not only the implementation of an alphabet within a single self-replicating mechanism, but also the trade-offs inherent in the design of the mechanism itself.

8. REFERENCES

[1] M. Arifuzzaman, M. Maeda, A. Itoh, K. Nishikata, C. Takita, R. Saito, T. Ara, K. Nakahigashi, H.-C. Huang, A. Hirai, K. Tsuzuki, S. Nakamura, M. Altaf-Ul-Amin, T. Oshima, T. Baba, N. Yamamoto, T. Kawamura, T. Ioka-Nakamichi, M. Kitagawa, M. Tomita, S. Kanaya, C. Wada, and H. Mori. Large-scale identification of protein-protein interaction of Escherichia coli K-12. *Genome Res.*, 16:686–691, 2006.

[2] F. H. Arnold. Design by directed evolution. *Accounts of Chemical Research*, 31:125–131, 1998.

[3] R. D. Barish, P. W. K. Rothemund, and E. Winfree. Two computational primitives for algorithmic self-assembly: Copying and counting. *Nano Lett.*, 5:2586–2592, 2005.

[4] R. D. Barish, R. Schulman, P. W. K. Rothemund, and E. Winfree. An information-bearing seed for nucleating algorithmic self-assembly. *P. Natl. Acad. Sci.*, 106(15):6054–6059, 2009.

[5] J. Bath and A. J. Turberfield. DNA nanomachines. *Nat. Nanotechnol.*, 2:275–284, 2007.

[6] R. Berger. The undecidability of the domino problem. *Memoirs of the AMS*, 66:1–72, 1966.

[7] C. K. Biebricher, M. Eigen, and R. Luce. Kinetic analysis of template-instructed and de novo RNA synthesis by Qβ replicase. *J. Mol. Biol.*, 148:391–410, 1981.

[8] V. A. Bloomfield, D. M. Crothers, and I. Tinoco. *Nucleic acids: structures, properties, and functions*. University Science Books, Mill Valley, Cal., 2000.

[9] A. G. Cairns-Smith. The origin of life and the nature of the primitive gene. *J. Theor. Biol.*, 10:53–88, 1966.

[10] A. G. Cairns-Smith. The chemistry of materials for artificial Darwinian systems. *Int. Rev. Phys. Chem.*, 7:209–250, 1988.

[11] H.-L. Chen, Q. Cheng, A. Goel, M.-D. Huang, and P. M. de Espanés. Invadable self-assembly: Combining robustness with efficiency. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 883–892, 2005.

[12] H.-L. Chen and A. Goel. Error free self-assembly using error prone tiles. In C. Ferretti, G. Mauri, and C. Zandron, editors, *DNA Computing 10*, volume LNCS 3384, pages 62–75, Berlin Heidelberg, 2005. Springer-Verlag.

[13] M. Cook, P. W. K. Rothemund, and E. Winfree. Self-assembled circuit patterns. In J. Chen and J. Reif, editors, *DNA Computing 9*, volume LNCS 2943, pages 91–107, Berlin Heidelberg, 2004. Springer-Verlag.

[14] R. M. Dirks and N. A. Pierce. Triggered amplification by hybridization chain reaction. *P. Natl. Acad. Sci.*, 101(43):15275–15278, 2004.

[15] M. Eigen, J. McCaskill, and P. Schuster. Molecular quasi-species. *J. Phys. Chem.*, 92:6881–6891, 1988.

[16] A. D. Ellington and J. W. Szostak. *In vitro* selection of RNA molecules that bind specific ligands. *Nature*, 346:817–821, 1990.

[17] T.-J. Fu and N. C. Seeman. DNA double-crossover molecules. *Biochemistry*, 32:3211–3220, 1993.

[18] K. Fujibayashi, R. Hariadi, S. H. Park, E. Winfree, and S. Murata. Toward reliable algorithmic self-assembly of DNA tiles: a fixed-width cellular automaton pattern. *Nano Letters*, 8:3554–3560, 2008.

[19] D. R. Halpin and P. B. Harbury. DNA display II. genetic manipulation of combinatorial chemistry libraries for small-molecule evolution. *PLOS Biol.*, 2:e174, 2004.

[20] Y. He and D. R. Liu. Autonomous multistep organic synthesis in a single isothermal solution mediated by a DNA walker. *Nat. Nanotechnol.*, 5:778–782, 2010.

[21] C. J ackel, P. Kast, and D. Hilvert. Protein design by directed evolution. *Annu. Rev. Biophys.*, 37:153–173, 2008.

[22] G. F. Joyce. Amplification, mutation and selection of catalytic RNA. *Gene*, 82:83–87, 1989.

[23] K. L. Kelly, E. Coronado, L. L. Zhao, and G. C. Schatz. The optical properties of metal nanoparticles: The influence of size, shape, and dielectric environment. *J. Phys. Chem. B*, 107:668–677, 2003.

[24] M. E. Leunissen, R. Dreyfus, R. Sha, T. Wang, N. C. Seeman, D. J. Pine, and P. M. Chaikin. Towards self-replicating materials of DNA-functionalized colloids. *Soft Matter*, 5:2422–2430, 2009.

[25] G. E. Liepinsa and M. D. Vose. Representational issues in genetic optimization. *J. Exp. Theor. Artif. In.*, 2:101–115, 1990.

[26] M. Mitchell, S. Forrest, and J. H. Holland. The royal road for genetic algorithms: Fitness landscapes and GA performance. In *Proceedings of the First European Conference on Artificial Life*, 1992.

[27] T. W. Odom, J.-L. Huang, P. Kim, and C. M. Lieber. Atomic structure and electronic properties of single-walled carbon nanotubes. *Nature*, 391:62–64, 1998.

[28] J. H. Reif, S. Sahu, and P. Yin. Compact error-resilient computational DNA tiling assemblies. In C. Ferretti, G. Mauri, and C. Zandron, editors, *DNA Computing 10*, volume LNCS 3384, pages 293–307, Berlin Heidelberg, 2005. Springer-Verlag.

[29] P. W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440:297–302, 2006.

[30] P. W. K. Rothemund, N. Papadakis, and E. Winfree. Algorithmic self-assembly of DNA Sierpinski triangles. *PLOS Biology*, 2:424–436, 2004.

[31] F. Rothlauf. *Representations for genetic and evolutionary algorithms*. Physica-Verlag, Heidelberg New York, 2002.

[32] R. Schulman and E. Winfree. Self-replication and evolution of DNA crystals. In *Advances in Artificial Life, 8th European Conference*, volume 3630, Berlin Heidelberg, 2005. Springer-Verlag.

[33] R. Schulman and E. Winfree. Synthesis of crystals with a programmable kinetic barrier to nucleation. *P. Natl. Acad. Sci.*, 104(39):15236–15241, 2007.

[34] R. Schulman and E. Winfree. How crystals that sense and respond to their environments could evolve. *Natur. Comp.*, 7:219–237, 2008.

[35] R. Schulman and E. Winfree. Simple evolution of complex crystal species. In *DNA Computing 16*, volume 6518, Berlin Heidelberg, 2010. Springer-Verlag.

[36] G. Seelig, D. Soloveichik, D. Y. Zhang, and E. Winfree. Enzyme-free nucleic acid logic circuits. *Science*, 314:1585–1588, 2006.

[37] N. C. Seeman. Nucleic-acid junctions and lattices. *J. Theor. Biol.*, 99(2):237–247, 1982.

[38] D. Soloveichik and E. Winfree. Complexity of compact proofreading for self-assembled patterns. In *DNA Computing 11*, Berlin Heidelberg, 2005. Springer-Verlag.

[39] H. Wang. Proving theorems by pattern recognition. II. *Bell Syst. Tech J.*, 40:1–42, 1961.

[40] H. Wang. An unsolvable problem on dominoes. Technical Report BL-30 (II-15), Harvard Computation Laboratory, 1962.

[41] L. Wang, J. Xie, and P. G. Schultz. Expanding the genetic code. *Annu. Rev. Biophys. Bio.*, 35:225–249, 2006.

[42] E. Winfree. On the computational power of DNA annealing and ligation. In *DNA Based Computers*, pages 199–221, 1995.

[43] E. Winfree. Simulations of computing by self-assembly. Technical Report CS-TR:1998.22, Caltech, 1998.

[44] E. Winfree and R. Bekbolatov. Proofreading tile sets: Error-correction for algorithmic self-assembly. In J. Chen and J. Reif, editors, *DNA Computing 9*, volume LNCS 2943, pages 126–144, Berlin Heidelberg, 2004. Springer-Verlag.

[45] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–544, 1998.

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

[47] D. Y. Zhang and B. Yurke. A DNA superstructure-based replicator without product inhibition. *Natur. Comp.*, 5:183–202, 2006.

[48] M. Zuker. Calculating nucleic acid secondary structure. *Curr. Opin. Chem. Biol.*, 10:303–310, 2000.