

AUTONOMOUS PROGRAMMABLE BIOMOLECULAR DEVICES

USING SELF-ASSEMBLED DNA NANOSTRUCTURES

*Surveying recent developments in
bio-DNA computing.*

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

ILLUSTRATION BY JEAN-FRANÇOIS PODEVIN

This article illustrates the way in which computer science techniques and methods influence this emerging field. Some of the key questions one might ask about biomolecular devices include:

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

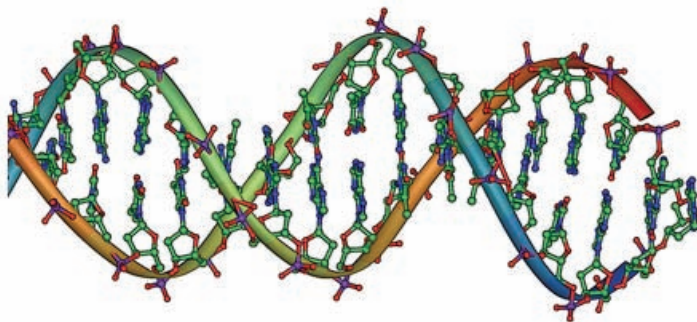
Note that these questions are exactly the sort of questions that computer scientists routinely ask about conventional computing devices. The discipline of computer science has developed a wide

variety of techniques to address such basic questions, and we will later point out some that have an important impact on molecular-scale devices.

DNA Nanotechnology and its Use to Assemble Molecular-Scale Devices. In general, nanoscience research is highly interdisciplinary. In particular, DNA self-assembly uses techniques from multiple disciplines, such as biochemistry, physics, chemistry, and material science, as well as computer science and mathematics. While this makes the topic quite intellectually exciting, it also makes it challenging for a typical computer science reader. Having no training in biochemistry, he or she must obtain a coherent understanding of the topic from a short article. For this reason, this article was written with the expectation that the reader is a computer scientist with limited background knowledge of chemistry or biochemistry (see the sidebar “A Brief Introduction to DNA”; in the sidebar “Why Use DNA to

A BRIEF INTRODUCTION TO DNA

Single-stranded DNA (denoted ssDNA) is a linear polymer consisting of a sequence of DNA bases oriented along a backbone with chemical directionality. By convention, the base sequence is listed starting from the 5-prime end of the polymer and ending at the 3-prime end (these names refer to particular carbon atoms in the deoxyribose sugar units of the sugar-



Structure of a DNA double helix (created by Michael Ströck and released under the GNU Free Documentation License).

DNA hybridization occurs in a buffer solution with appropriate temperature, pH, and salinity.

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

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

phosphate backbone, the details of which are not critical to the discussion in this article). The consecutive bases (monomer units) of an ssDNA molecule are joined via covalent bonds. There are four types of DNA bases: adenine, thymine, guanine, and cytosine, typically denoted by the symbols A, T, G, and C. These bases form the alphabet of DNA; the specific sequence comprises DNA's information content. The bases are grouped into complementary pairs (G, C) and (A, T).


The most basic DNA operation is hybridization, where two ssDNA oriented in opposite directions can bind to form a double-stranded DNA helix (dsDNA) by pairing between complementary bases.

WHY USE DNA TO ASSEMBLE MOLECULAR-SCALE DEVICES?

There are many advantages of DNA as a material for building things at the molecular scale. From the perspective of design, the advantages are:

- The structure of most complex DNA nanostructures can be reduced to determining the structure of short segments of dsDNA. The basic geometric and thermodynamic properties of dsDNA are well understood and can be predicted by available software systems from key relevant parameters like sequence composition, temperature, and buffer conditions.
- Design of DNA nanostructures can be assisted by software. To design a DNA nanostructure or device, one needs to design a library of ssDNA strands with specific segments that hybridize to (and only to) specific complementary segments on other ssDNA. There are a number of software systems (developed at NYU, Caltech, and Duke University) for design of the DNA sequences composing DNA tiles and for optimizing their stability, which employ heuristic optimization procedures for this combinatorial sequence design task.

From the perspective of experiments, the advantages are:

- The synthesis of ssDNA is now routine and inexpensive; a test tube of ssDNA consisting of any specified short sequence of bases (<150) can be obtained from commercial sources for modest cost (about half a U.S. dollar per base at this time); it will contain a very large number (typically at least 10^{13}) identical ssDNA molecules. The synthesized ssDNA can have errors (premature termination of the synthesis is the most frequent) but can be easily purified by well-known techniques, such as electrophoresis.
- The assembly of DNA nanostructures is a very simple experimental process: in many cases, one simply combines the various component ssDNA into a single test tube with an appropriate buffer solution at an initial temperature above the melting temperature and then slowly cools the test tube below the melting temperature.
- The assembled DNA nanostructures can be characterized by a variety of techniques. One such technique is electrophoresis. It provides information about the relative molecular mass of DNA molecules, as well as some information regarding their assembled structures, depending on what type (denaturing or native, respectively) of electrophoresis is used. Other techniques like atomic force microscopy (AFM) and transmission electron microscopy (TEM) provide images of the actual assembled DNA nanostructures on 2D surfaces. 

Assemble Molecular-Scale Devices?” we provide some reasons why DNA is uniquely suited for this application; the sidebar “Manipulation of DNA” lists some known enzymes used for manipulation of DNA nanostructures).

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

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

ADLEMAN’S INITIAL DEMONSTRATION OF A DNA-BASED COMPUTATION

Adleman’s Experiment. The field of DNA computing began in 1994 with a laboratory experiment described in [1]. The goal of the experiment was to find a Hamiltonian path in a graph, which is a path that visits each node exactly once. To solve this problem, a set of single-stranded DNA (ssDNA) were designed based on the set of edges of the graph. When combined in a test tube and cooled, they self-assembled into double-stranded DNA (dsDNA). Each of these DNA nanostructures was a linear DNA helix that corresponded to a path in the graph. If the graph had a Hamiltonian path, then one of these DNA nanostructures encoded the Hamiltonian path. By conventional biochemical extraction methods, Adleman was able to isolate only DNA nanostructures encoding Hamiltonian paths, and by determining their sequence, the explicit Hamiltonian path. It should be mentioned that this landmark experiment was designed and experimentally demonstrated by Adleman alone, a computer scientist with limited training in biochemistry.

MANIPULATION OF DNA

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

- Restriction enzymes, some of which can cut (or nick, which is to cut only one strand) strands of a DNA helix at locations determined by short specific DNA base sequences.
- Ligase enzymes that can heal nicks in a DNA helix.
- Polymerase, which given an initial “primer” DNA strand hybridized onto a segment of a template DNA strand, can extend the primer strand in the 5' to 3' direction by appending free nucleotides complementary to the template's nucleotides.

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

The Non-Scalability of Adleman's Experiment. While this experiment founded the field of DNA computing, it was not scalable in practice, since the number of different DNA strands needed increased exponentially with the number of nodes of the graph. Although there can be an enormous number of DNA strands in a test tube (10^{15} or more, depending on solution concentration), the size of the largest graph that could be solved by Adleman's method was limited to at most a few dozen nodes. This is not surprising, since finding the Hamiltonian path is an NP-complete problem, whose solution is likely to be intractable using conventional computers. Even though DNA computers operate at the molecular scale, they are still equivalent to conventional computers (for example, deterministic Turing machines) in computational power. This experiment provided an important lesson to the DNA computing community (which is now well recognized): to carefully examine scalability issues and to judge any proposed experimental methodology by its scalability.

Autonomous Biomolecular Computation. Shortly following Adleman's experiment, there was a burst of further experiments in DNA com-

puting, many of which were quite ingenious. However, almost none of these DNA computing methods were autonomous and instead required many tedious laboratory steps to execute. In retrospect, one of the most notable aspects of Adleman's experiment was that the self-assembly phase of the experiment was completely autonomous—it required no exterior mediation. This autonomous property makes an experimental laboratory demonstration much more feasible as the scale increases. The remainder of this article focuses on autonomous devices for biomolecular computation based on self-assembly.

SELF-ASSEMBLED DNA TILES AND LATTICES

Computation By Self-Assembly. The most basic way that computer science ideas have influenced DNA nanostructure design is via the pioneering work by theoretical computer scientists on a formal model of 2D tiling due to Wang in 1961, which culminated in a proof by Berger in 1966 that universal computation could be done via tiling assemblies. Winfree was the first to propose applying the concepts of computational tiling assemblies to DNA molecular constructs. His core idea was to use tiles composed of DNA to perform computations during their self-assembly process. To better understand this idea, see the sidebar “DNA Nanostructures.”

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

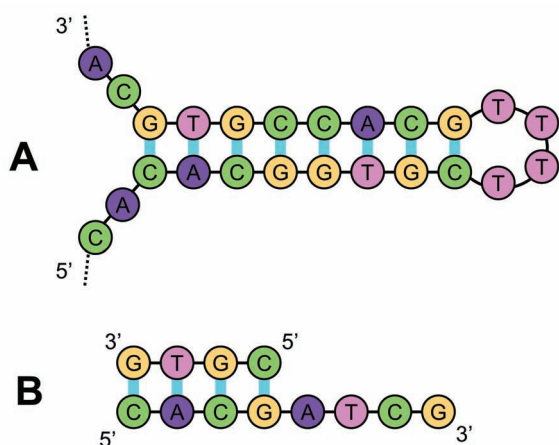
To program a tiling assembly, the pads of tiles are designed so the tiles assemble together as intended. Proper designs ensure that only the adjacent pads of neighboring tiles are complementary, so only those pads hybridize together (see the sidebar “DNA Tiles”). For a recent review of DNA technology, see [2].

AUTONOMOUS FINITE STATE COMPUTATION USING LINEAR DNA NANOSTRUCTURES

The first experimental demonstrations of computation using DNA tile assembly was [3]. It

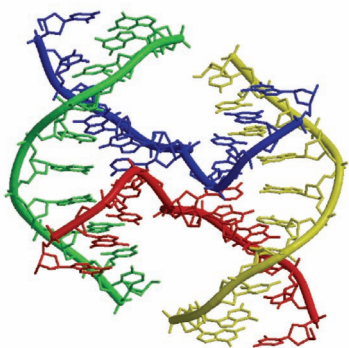
DNA NANOSTRUCTURES

A DNA nanostructure is a multi-molecular complex consisting of a number of ssDNA that have partially hybridized along their subsegments. The field of DNA nanostructures was pioneered by Seeman [7]. Particularly useful types of motifs often found in DNA nanostructures include:



A stem-loop and a sticky end.

- Stem-loop and sticky end. A stem-loop (A), where ssDNA loops back to hybridize on itself (that is, one segment of the ssDNA (near the 5' end) hybridizes with another segment further along (nearer the 3' end) on the same ssDNA strand). The shown stem consists of the dsDNA region with sequence CACGGTGC on the bottom strand. The shown loop consists of the ssDNA region with sequence TTTT. Stem-loops are often used to form patterning on DNA nanostructures. A sticky end (B) is where unhybridized ssDNA protrudes from the end of a double helix. The sticky end shown (ATCG) protrudes from dsDNA (CACG on the bottom strand). Sticky ends are often used to combine two DNA nanostructures together via hybridization of their complementary ssDNA. The figure shows the antiparallel nature of dsDNA with the 5' end of each strand pointing toward the 3' end of its partner strand.
- A Holliday junction, where two parallel DNA helices form a junction with one strand of each DNA helix (blue and red) crossing over to the other DNA helix. Holliday junctions are often used to tie together various parts of a DNA nanostructure. **C**



A Holliday junction
(created by Miguel Ortiz-Lombardía,
CNIO, Madrid, Spain).

demonstrated a two-layer, linear assembly of TX tiles that executed a bit-wise cumulative XOR computation. In this computation, n bits are input and n bits are output, where the i^{th} output is the XOR of the first i input bits. This is the computation occurring when one determines the output bits of a full-carry binary adder circuit. The experiment [3] is described further in the sidebar “Sequential Boolean Computation via a Linear DNA Tiling Assembly.” This experiment [3] provided answers to some of the most basic questions that are most likely to be of interest and concern to a computer scientist:

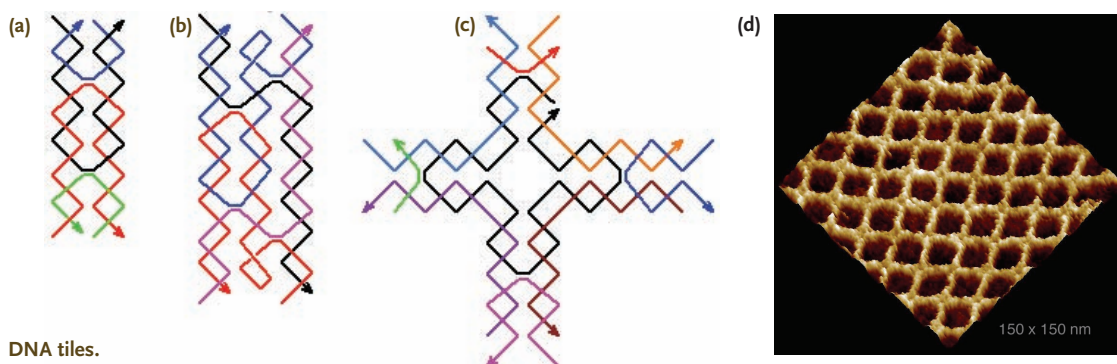
How can one provide data input to a molecular computation using DNA tiles? In this experiment the input sequence of n bits was defined as an “input” ssDNA strand with the input bits (1s and 0s) encoded by distinct short subsequences. Two different tile types (depending on whether the input bit was 0 or 1, these had specific stick-ends and specific subsequences at which restriction enzymes can cut the DNA backbone) were available to assemble around each of the short subsequences comprising the input strand, forming the blue input layer illustrated in the sidebar “Sequential Boolean Computation via a Linear DNA Tiling Assembly.”

How can one execute a step of computation using DNA tiles? To execute steps of computation, the TX tiles were designed to have pads at one end that encoded the cumulative XOR value. Also, since the reporter strand segments ran through each such tile, the appropriate input bit was also provided within its structure. These two values implied the opposing pad on the other side of the tile to be the XOR of these two bits.

How can one determine and/or display the output values of a DNA tiling computation? The output in this case was read by determining which of two possible cut sites (endonuclease cleavage sites) were present at each position in the tile assembly. This was executed by first isolating the reporter strand, then digesting separate aliquots with each endonuclease separately and the two together. Finally these samples were examined by gel electrophoresis, and the output values were displayed as banding patterns on the gel.

Another method for output is the use of atomic force microscopy (AFM) observable patterning. The patterning was made by designing the tiles computing a bit 1 to have a stem loop protruding from the top of the tile. The sequence of this molecular patterning was clearly viewable under appropriate AFM imaging conditions.

DNA TILES



Seeman and Winfree in 1998 developed a family of DNA tiles known collectively as DX tiles (see left tile (a)) that consisted of two parallel DNA helices linked by immobile Holliday junctions. They demonstrated that these tiles formed large 2D lattices, as viewed by AFM. (b) Subsequently, other DNA tiles were developed to provide for more complex strand topology and interconnections, including a family of DNA tiles known as TX tiles (see b) composed of three DNA helices. Both the DX tiles and the TX tiles are rectangular in shape, where two opposing edges of the tile have pads consisting of ssDNA sticky ends of the component strands. In addition, TX tiles have topological properties that allow for strands to propagate in useful ways through tile lattices (this property is often used for patterning DNA lattices). (c) Other DNA tiles known as Cross-Over tiles (see d) [10] are shaped roughly square, and have pads on all four sides, allowing for binding of the tile directly with neighbors in all four directions in the lattice plane. **C**

Although only very simple computations, the experiments of [3] and [10] demonstrated for the first time methods for autonomous execution of a sequence of finite-state operations via algorithmic self-assembly, as well as for providing inputs and for outputting the results.

Autonomous Finite-State Computations via Disassembly of DNA Nanostructures. An alternative method for autonomous execution of a sequence of finite-state transitions was subsequently developed by [8]. Their technique essentially operated in the reverse of the assembly methods described previously and instead was based on disassembly. They began with a linear DNA nanostructure whose sequence encoded the inputs and then executed a series of steps that digested the DNA nanostructure from one end. On each step, a sticky end at one end of the nanostructure encoded the current state, and the finite transition was determined by hybridization of the current sticky end with a small “rule” nanostructure encoding the finite-state transition rule. Then a restriction enzyme, which recognized the sequence encoding the current input, as well as the current state, cut the appended end of the linear DNA nanostructure to expose a new sticky end encoding the next state.

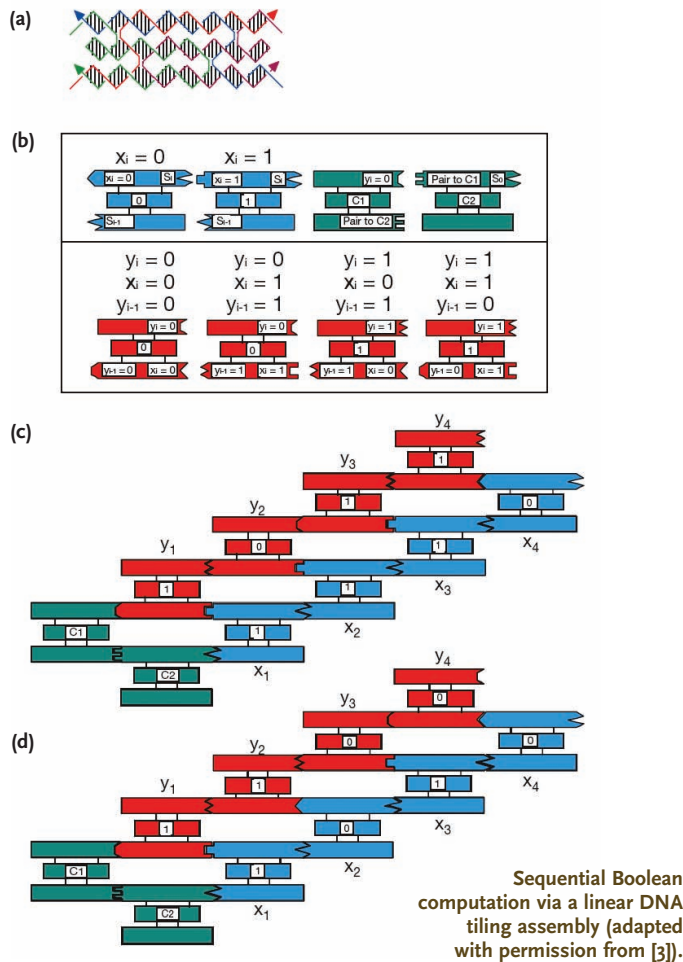
CONCLUSION

Other significant topics, not addressed in this article include:

- Assembling patterned and addressable 2D DNA lattices, such as attaching materials to DNA [11], and methods for programmable assembly of patterned 2D DNA lattices [5, 6, 9].
- Autonomous molecular transport devices self-assembled from DNA [3, 12].
- Future challenges for self-assembled DNA nanostructures, such as error correction and self-repair at the molecular scale [4], and 3D DNA lattices.

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

SEQUENTIAL BOOLEAN COMPUTATION VIA A LINEAR DNA TILING ASSEMBLY



The figure here shows a unit TX tile (a) and the sets of input and output tiles (b) with geometric shapes conveying sticky-end complementary matching. The tiles of (b) execute binary computations depending their pads, as indicated by the table in (b). The (blue) input layer and (green) corner condition tiles were designed to assemble first (see example computational assemblies (c) and (d)). The (red) output layer then assemble specifically starting from the bottom-left using the inputs from the blue layer. (See [3] for more details of this molecular computation.) The tiles were designed such that an output reporter strand ran through all the n tiles of the assembly by bridges across the adjoining pads in input, corner, and output tiles. This reporter strand was pasted together from the short ssDNA sequences within the tiles using ligation enzyme mentioned previously. When the solution was warmed, this output strand was isolated and identified. The output data was read by experimentally determining the sequence of cut sites. In principle, the output could be used for subsequent computations. **C**

assembly processes [6, 9]. We have observed that many of these self-assembly processes are computational-based and programmable, and it seems likely that computer science techniques will be essential to the further development of this emerging field of biomolecular computation. **C**

REFERENCES

- Adleman, L. Computing with DNA. *Scientific American* 279, 2 (Aug. 1998), 34–41.
- Deng, Z., Chen, Y., Tian, Y., and Mao, C. A fresh look at DNA nanotechnology. In J. Chen, N. Jonoska, and G. Rozenberg, Eds. *Nanotechnology: Science and Computation*. Springer Verlag series in Natural Computing, 2006, 23–34.
- Mao, C., LaBean, T.H., Reif, J.H., and Seeman, N. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature* 407 (Sept. 28, 2000), 493–495.
- Reif, J.H., Sahu, S., and Yin, P. Compact Error-Resilient Computational DNA Tiling Assemblies, In J. Chen, N. Jonoska, and G. Rozenberg, Eds. *Nanotechnology: Science and Computation*. Springer Verlag series in Natural Computing, 2006, 79–104.
- Rothemund, P.W.K. Folding DNA to create nanoscale shapes and patterns. *Nature* 440 (Mar. 16, 2006), 297–302.
- Rothemund, P.W.K., Papadakis, N., and Winfree, K. Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biology* 2, 12 (Dec. 2004).
- Seeman, N.C. Nanotechnology and the double helix. *Scientific American* 290, 6 (June 2004), 64–75.
- Shapiro, E. and Benenson, Y. Bringing DNA computers to life. *Scientific American* (May 2006), 45–51.
- Yan, H., LaBean, T.H., Feng, L., and Reif, J.H. Directed nucleation assembly of barcode patterned DNA lattices. *PNAS* 100, 14 (July 8, 2003), 8103–8108.
- Yan, H., Feng, L., LaBean, T.H., and Reif, J. DNA nanotubes, parallel molecular computations of pairwise exclusive-or (XOR) using DNA “string tile” self-assembly. *Journal of the American Chemistry Society (JACS)* 125, 47 (2003), 14246–14247.
- Yan, H. et al. DNA-templated self-assembly of protein arrays and highly conductive nanowires. *Science* 301 (Sept. 26, 2003), 1882–1884.
- Yin, P. et al. A unidirectional DNA walker moving autonomously along a linear track. *Angewandte Chemie (International Edition)* 43, 37 (Sept. 20, 2004), 4906–4911.

JOHN H. REIF (reif@cs.duke.edu) is Hollis Edens Distinguished Professor in Trinity College of Arts and Sciences at Duke University in Durham, NC.

THOMAS H. LABEAN (thomas.labean@duke.edu) is an associate research professor in the departments of computer science and chemistry at Duke University in Durham, NC.

Support for research appearing in this article has been provided by NSF grants CCF-0523555, CCF-0432038, CCF-0432047. An extended version of this article that includes the topics listed in the Conclusion section is available at www.cs.duke.edu/~reif/paper/AutonomousDNA/AutonomousDNA.pdf.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

© 2007 ACM 0001-0782/07/0900 \$5.00