DNA COMPUTING

By David I. Lewin

CAN BIOMOLECULES COMPUTE? AS FAR BACK AS 1945, PHYSICIST ERWIN SCHROEDINGER DESCRIBED THE MOLECULAR NATURE OF THE GENE IN INFORMATION-THEORETIC TERMS IN HIS BOOK *WHAT IS LIFE?* (*WHAT IS LIFE? THE PHYSICAL ASPECT OF THE Living Cell*, Cambridge University Press, 1945). Now, nearly 60 years later, computer scientists are joining forces with molecular biologists and chemists to explore the potential for computation using information-carrying biological polymers such as nucleic acids (DNA and RNA). "DNA computing is a subset of molecular computing," says Nadrian C. Seeman, a chemist at New York University. "The key feature of DNA for computing is its information content."

A number of labs are taking different approaches, but all take advantage of the complementarity of the two strands of double-stranded DNA, and the ability of strings of DNA subunits (called bases or nucleotides) to bind together using Watson-Crick pairing (A [adenosine] with T [thymidine] and C [cytosine] with G [guanidine]), says Seeman. In essence, says DNA computing researcher Erik Winfree of the California Institute of Technology in Pasadena, there is a code of "stick" or "don't stick," with DNA chains either binding to form regions of double-stranded molecules or remaining free as regions of single-stranded DNA. "Why does anyone care about molecular computing?" Seeman asks. "Because you get to use molecularity's enormous parallelism. In a small volume, you can have 10^{12} to 10^{13} different molecules." These molecules can each function in parallel, and users can select the right answer.

From Turing machines to DNA computing

When Alan Turing proposed the finite automata concept—the theory underlying digital computers—in 1936, he left the device's physical nature unspecified. In most descriptions of automata, they are represented by a tape on which symbols are written and erased, systems to read and write the symbols, and a set of rules that act on the initial set of input symbols. Within decades, electronic devices were used to construct such computing devices—first using vacuum tubes, then transistors. Leonard Adleman demonstrated the possibility of using DNA for computation in 1994 by solving a seven-node graph problem (finding a path that visits each node exactly once) using recombinant DNA techniques. In his experiment, a sequence of short, single-stranded DNA polymers (oligonucleotides) encoded the names of nodes, a self-assembly technique then generated a library of possible solutions, recombinant DNA techniques selected those sequences that met the problem's requirements, and gel eletrophoresis—a standard technique for separating macromolecules—served as the means of output. Since his experiment, several other laboratories have produced demonstration systems that use DNA's properties to solve more complicated problems, but to operate they need full-scale laboratories and human intervention.

Ehud Shapiro of the Weizmann Institute of Science in Rehovot, Israel, and his colleagues at Technion-Israel Institute of Technology in Haifa demonstrated the possibility of constructing an autonomous computing device from macromolecules. In the 21 November 2001 issue of *Nature*, Shapiro and his colleagues report the development of a macromolecular version of a two-state finite automaton with an alphabet of two input signals ("a" and "b" are each represented by six-nucleotide sequences) and a six-nucleotide termination sequence. Eight distinct, short, double-stranded DNA molecules represented the eight possible transition rules operating on "a" and "b."

Transition rules are expressed through specific molecular biological events. Enzymes that recognize and act on specific DNA base sequences—either by cleaving a recognized sequence to form a four-base-long sticky end (*FokI*) or by creating a chemical bond (ligation) between fragments of single-stranded DNA held adjacent by an overlapping complementary sequence (*ligase*)—cause transitions according to the rules. A double-stranded DNA molecule also represents the au-

Computing in SCIENCE & ENGINEERING

EDITORIAL OFFICE
COMPUTING in SCIENCE & ENGINEERING
10662 Los Vaqueros Circle, PO Box 3014
Los Alamitos, CA 90720-1314
phone +1 714 821 8380; fax +1 714 821 4010; computer.org/CiSE

tomaton's initial state, and DNA molecules with unpaired sticky ends act as output detectors. Once set up, the liquid-phase "computer" functions without additional intervention—though human intervention is needed to observe the output molecules that are formed. Shapiro likens this additional step to a computer program calling the print command of an operating system to type out its results.

Shapiro, who was in computer science before he became interested in biomolecular computing, says that his goal is to construct nanoscale DNA computers that operate in liquid phase. His reaction mixture, a volume of 120 microliters, contains approximately 10^{12} automata sharing the same software and running in parallel. In some of his team's experiments reported in *Nature*, they used equal amounts of different inputs, and they discovered all of the expected outputs. "We're not aiming for high performance computing at all. These are not complicated problems we're trying to solve," Shapiro says.

DNA computing's limitations

A current limitation to this branch of DNA computing is the use of natural enzymes, which only recognize and act on certain nucleotide sequences. The development of designer enzymes that can identify additional sequences might take decades, Shapiro notes. "In the medium term, we can envision biotechnology applications for such automata, such as to analyze DNA without first sequencing it," Shapiro says. In the longer term, he foresees construction of artificial "cells" with synthetic-DNA programming. "A lot of processes in the living cell resemble computing in fundamental ways."

Other researchers have reported liquid-phase systems for DNA computing. Laura Landweber and her Princeton University colleagues' article in *Proceedings of the National Academy of Sciences*' 15 February 2000 issue used a hybrid DNA-RNA computing system to compute solutions to a variant of a well-known chess problem. The "Knight problem" asks what configurations of knights can be placed on a chess board with *n* squares on a side such that none of the knights is attacking another. RNA strands with 10 bits (each bit represented by 15 nucleotides separated by 5-nucleotide spacers) were randomly synthesized. In this approach, an enzyme, RNase H, breaks down those RNA chains that did not meet the problem's constraints by attacking the RNA strand of DNA-RNA hybrid molecules.

The Princeton researchers write in their PNAS article, "A value of 0 or 1 is assigned to a specific bit position by destroying all strands in the RNA library which do not have the value at this position." Researchers repeat this process by applying the other constraints, and DNA molecules complementary to the remaining RNA strands were generated using polymerase chain reaction. The researchers sequenced complementary DNA to the resulting 43 distinct RNA molecules, and read them by sequencing the DNA. Of the 94 possible solutions to a 3x3 Knight problem, the researchers randomly selected 42 proper solutions; one solution, involving an illegal placement of a knight, was incorrect, giving a 97.7 percent success rate in finding correct solution strands.

Originally, the goal of DNA computing, as envisioned by Adleman and others, was to solve numerical problems. "You pose some hard optimization problems, convert it into DNA and use DNA's massive parallelism to search a large number of possibilities, with a chemical reaction to filter out the ones you don't want," Winfree says. In contrast, much of the research underway in DNA computation is about self-assembly of structures, "gaining a different kind of control on molecular processes." Researchers in this field are "essentially using the idea of an algorithm in information processing—a specific set of rules that can be iterated to accomplish some task" and applying it to chemical tasks. "The task is now one in which you have chemical input—the components for the structure you want to build—and chemical output—the final structure," Winfree says. Because an algorithm is involved, the assembly can be viewed as a form of computation. "It's another case where the tools and concepts of computer science are brought to bear on how to solve the problem."

In particular, Seeman and collaborator John Reif of Duke University in Durham, North Carolina, have used DNA structures to form self-assembling tilings, analogous to Wang tiles in which the multicolored tiles self-assemble to form a mosaic with the same color flanking every edge of the mosaic. Instead of colors, the "tiles" are multi-armed hybrid DNA molecules with four arms, each with a single-stranded sticky end, says Seeman. The sticky ends are very important, giving predictable affinity and structure to the molecules. "About a year and a half ago, we reported a cumulative XOR [exclusive OR] computation" by self-assembly of these DNA tiles, Seeman says. "The nice thing about such systems is that it appears that they will scale nicely."

DNA technology—building computers or robots?

The self-assembly properties of DNA suggest an indirect application to computing, says Winfree. "For example, one

DEPARTMENT EDITORS

Book & Web Reviews: Susan McKay, Univ. of Maine, susan_mckay@umit.maine.edu, and George Cybenko, Dartmouth College, gvc@dartmouth.edu
Computing Prescriptions: Isabel Beichl, Nat'l Inst. of Standards and Tech., isabel.beichl@nist.gov, and Julian Noble, Univ. of Virginia, jvn@virginia.edu
CSE in Industry: Thomas L. Tysinger, Fluent Inc., tlt@fluent.com
Education: Denis Donnelly, Siena College, donnelly@siena.edu
Grid Computing: Geoffrey Fox, Florida State Univ., gcf@grids.ucs.indiana.edu
Interfaces: Norris Parker Smith, nsmith03@snet.net
Scientific Programming: Paul Dubois, Lawrence Livermore Nat'l Labs, dubois1@llnl.gov, and David Beazley, Univ. of Chicago, beazley@uchicago.edu
Technology News & Reviews: Donald Shirer, Yale Univ., donshirer@earthlink.net, and Douglas Tougaw, Valparaiso Univ., doug.tougaw@valpo.edu
Visualization Corner: Jim X. Chen, George Mason Univ., jchen@cs.gmu.edu, and R. Bowen Loftin, Old Dominion Univ., bloftin@odu.edu

STAFF

Associate Lead Editor: Jenny Ferrero, jferrero@computer.org
Group Managing Editor: Crystal Chweh
Senior Editor: Dale Strok
Associate Editors: Dennis Taylor and Shani Murray
Staff Editors: Scott L. Andresen and Kathy Clark-Fisher
Editorial Assistants: Ty Manuel and Rebecca Deuel
Magazine Assistants: Dawn Craig and Pauline Hosillos
Contributing Editors: Kirk Kroeker, David I. Lewin, Keri Schreiner, Joan Taylor, and Margaret Weatherford
Design Director: Toni Van Buskirk
Production Artists: Carmen Flores-Garvey and Larry Bauer
Production Assistant: Monette Velasco
Technical Illustration: Alex Torres

Publisher: Angela Burgess
Assistant Publisher: Dick Price
Assistant Advertising Coordinator: Debbie Sims
Marketing Manager: Georgann Carter
Business Development Manager: Sandra Brown

IEEE ANTENNAS AND PROPAGATION SOCIETY LIAISON

Don Wilton, Univ. of Houston, wilton@uh.edu

IEEE SIGNAL PROCESSING SOCIETY LIAISON

Elias S. Manolakos, Northeastern Univ., elias@cdsp.neu.edu

AMERICAN INSTITUTE OF PHYSICS ADVISORY COMMITTEE

Martin S. Weinhous (chair), Cleveland Clinic, weinhous@radonc.ccf.org
Norman Chonacky, Columbia Univ., chonacky@chem.columbia.edu
Michael J. Creutz, Brookhaven Nat'l Lab., creutz@bnl.gov
Efthimios Kaxiras, Harvard Univ., kaxiras@cmt.harvard.edu
David P. Landau, Univ. of Georgia, dlandau@hal.physast.uga.edu
John Rundle, Univ. of Colorado, rundle@cires.colorado.edu
Henrik Schmidt, MIT, henrik@keel.mit.edu

CS MAGAZINE OPERATIONS COMMITTEE

George Cybenko (chair), James H. Aylor, Thomas J. Bergin, Frank Ferrante, Forouzan Golshani, Rajesh Gupta, Steve McConnell, Ken Sakamura, M. Satyanarayanan, Nigel Shadbolt, Munindar P. Singh, Francis Sullivan, James J. Thomas

CS PUBLICATIONS BOARD

Rangachar Kasturi (chair), Jean Bacon, Mark Christensen, George Cybenko, Gabriella Sanniti di Baja, Lee Giles, Thomas Keefe, Dick Kemmerer, Anand Tripathi

IEEE Antennas & Propagation Society

AMERICAN INSTITUTE OF PHYSICS

IEEE Signal Processing Society

thing that people think seriously about is how to create extremely small electrical circuits where the components are now on the molecular scale." At that scale, lithography, if feasible at all, would be horribly expensive. "Could you chemically synthesize the components of an electrical circuit, like [the pieces of] a jigsaw puzzle, toss them into a chemical reaction, shake it for awhile, and have the correct circuit pattern create itself?" he asks. On paper, it looks like some simple electrical circuits could be assembled in this manner. "The question becomes, can we use DNA as the mechanism for directing the self-assembly of these circuits?"

A limitation for such applications is the error rate in self-assembly—"sometimes pieces that are supposed to stick don't stick in place and sometimes pieces that are not supposed to stick, stick in place," Winfree notes. "What we have to understand is what controls the specificity in the DNA structures that we design, and how we control the error rate through physical factors, such as DNA sequences' temperature and design." Controlling the starting point from which the self-assembly begins is important to controlling the process, he says.

The concept of self-assembly, which biological systems have evolved to form such structures as viruses, flagella, and microtubules (which serve as structural and motile components of cells), can lead the way to using DNA as the basis of nanorobotics. DNA-tilings researchers see the opportunity to use self-assembly to form designer materials. Because DNA is found in two physically different forms (B-DNA and Z-DNA, depending on how the nucleotide chains coil), this suggests a mechanism for using DNA for nanorobotic actuators. Actuators that produce either rotational or translational motion have been demonstrated, but development of DNA actuators that do both has not, says Reif.

Thus, it appears that macromolecular computing is blurring the lines between computing, materials science, and robotics. At least at the molecular scale, it might be possible to use the same class of molecules to build, program, and operate molecular-scale devices whose applications have only been foreshadowed by science fiction.

David I. Lewin is contributing editor to *Computing in Science & Engineering, IEEE Intelligent Systems,* and *Distributed Systems Online.* Trained in biology, he writes on science, technology, and medicine from Silver Spring, Md.