Computing with DNA

Getting DNA-based computers off the drawing board and into the wet lab

By IVARS PETERSON

"We use natural materials to make unnatural objects." This is how chemist Nadrian C. Seeman of New York (N.Y.) University describes the research that goes on in his laboratory. He and his collaborators have spent the last 15 years working with DNA molecules, assembling short strands into various branched structures, several types of knots, and a number of DNA-edged geometric shapes, including cubes and octahedrons with cutoff corners.

Along the way, the researchers have learned how to construct different kinds of branched junctions and how to attach sticky ends to these molecular stalks, enabling them to fashion pieces of DNA into unusual geometries. One of their main goals has been to develop methods of handling molecules to fabricate molecular-scale machinery and electronics (SN: 12/10/94, p. 396).

Now, Seeman has eased into a new role as adviser to computer scientists venturing into the biology lab to try out their ideas about computing with DNA molecules.

"Our experience with these systems has uncovered a large number of experimental pitfalls that may confront individuals working with DNA computing," Seeman says. He provides the kind of handy, practical advice—the tricks of the DNA trade—rarely mentioned in scientific papers, textbooks, or lab manuals.

"We're trying to take advantage of the rapidly evolving technology for manipulating DNA in the laboratory," says computer scientist Richard J. Lipton of Princeton University. Lipton and others envision computation taking place in test tubes rather than on silicon chips; they see information storage occurring in DNA-laced drops of water instead of on magnetic disks.

DNA has a number of qualities that computer scientists believe could make it an effective vehicle for delivering high-performance computing. DNA-based computers, Lipton maintains, would offer advantages in speed, memory capacity, and energy efficiency over conventional electronics for solving certain types of problems.

The hope in this new field is that the pattern-matching and polymerization processes of DNA chemistry, multiplied by the enormous number of molecules that fit into a small volume, can handle computations too difficult for conventional silicon-chip-based computers.

The idea of using molecules as the working elements of a computer goes back more than a decade (SN: 6/11/83, p. 378). It wasn't until 1994, however, that anyone actually stepped into the laboratory and succeeded in solving a computational problem in a test tube (SN: 11/12/94, p. 308).

That was when computer scientist Leonard M. Adleman of the University of Southern California in Los Angeles, using techniques from molecular biology, manipulated strands of DNA to answer a mathematical question: Given seven points linked by one-way paths, what route from a specified starting point visits each point once on the way to a given end point?

Adleman relied on the basic properties of DNA. A single strand of DNA consists of a chain of simpler molecules called bases, which come in four types: adenine (A), thymine (T), guanine (G), and cytosine (C). Any strand of DNA will adhere tightly to its complementary strand, in which T substitutes for A, G for C, and vice versa. For example, a single-stranded DNA segment consisting of the base sequence TAGCC will stick to a section of another strand made up of the complementary sequence ATCGG.

Adleman assigned each of the seven points in his array a unique code name made up of a single-stranded DNA sequence of 20 bases. Each one-way link between every pair of points was represented by another short strand consisting of the complements of the last 10 bases of the starting point and the complements of the first 10 bases of the destination point.

When these DNA strands, representing the points and the links between them, were mixed together, they joined to create longer, double-stranded molecules of different lengths, corresponding to all possible paths from point to point in Adleman's array. Using biochemical techniques, he then filtered out and sequenced the one type of molecule that gave the correct answer to his route-finding problem.

Adleman's pioneering demonstration was quickly followed by a flurry of proposals suggesting ways to exploit various types of DNA operations to solve a range of problems that typically stymie conventional computers. Many of these ideas have turned out to be wildly impractical; a few have shown promise.

Testing these possibilities in the laboratory, however, has proved daunting. Both the painstaking, time-consuming lab work necessary to complete the required operations and the complexity of the chemistry have presented significant obstacles.

Factors such as concentration and reaction rate can have a very strong effect on yields, says Stuart A. Kurtz of the University of Chicago. Slow steps in a long sequence of operations, for example, can lead to incomplete or misleading results.

It's even possible that the success of Adleman's beautifully executed experiment can be attributed in part to the fortuitous selection of appropriate reaction rates and DNA concentrations. Applying his technique to larger, more complicated, problems turns out to be no simple matter.

Modern biotechnology offers computer scientists a wide range of tools for manipulating DNA. They can synthesize custom DNA strands made up of any desired string of bases. They can extract out of a mixture all of the strands that have a chosen length or incorporate a specified short sequence of bases. They can clip strands into smaller segments. They can create double-stranded DNA by allowing segments with complementary sequences to stick to each other. They can use the polymerase chain reaction to generate copies of a given DNA sequence.

These operations can be combined in different ways to solve computational problems or to store and retrieve information. "The embarrassment is that there are so many biotechnological operations available to us that we still don't know the right ones to use or the best way to do anything," Lipton says.

"It's very exciting research," says Dana M. Latch, director of the computing theory program at the National Science Foundation. "There are lots of theoretical possibilities. The hard part is getting the biological operations to perform the way you want them to."

One troubling aspect is that biochemical processes are generally error-prone. DNA-duplicating operations involving enzymes don't guarantee perfect reliability and faithful copying. Separation or extraction processes sometimes remove the wrong strands or fail to remove enough extraneous material. Moreover, bases can pair up in more complicated ways than those specified by the usual rules.

Organisms have various repair and error-correction processes to keep everything in order. Such processes are absent from test-tube operations. In a lengthy series of reactions, errors may accumulate and result in an incorrect answer. Furthermore, the particular strand encoding the result of a computation may get lost, perhaps sticking to the side of a test tube or being filtered out by mistake.

Mathematics offers a potential solution to some of these DNA separation problems. Computer scientist Richard M. Karp of the University of Washington in Seattle and his colleagues have worked out an ingenious scheme for taking advantage of a series of relatively crude separations to achieve a highly reliable one.

For example, in a separation in which only 90 percent of the molecules go into the correct test tube and 10 percent into the wrong one, repeating the operation many times in just the right way can reduce the overall error considerably. This improvement, however, comes at the cost of many more steps and test tubes.

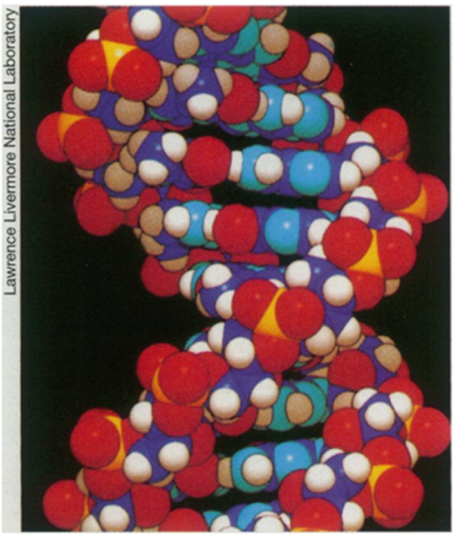
Coping with errors was a major theme at a meeting on DNA-based computers held last month at Princeton University and at an earlier workshop on biomolecular computing sponsored by NSF at Johns Hopkins University in Baltimore. Researchers proposed a variety of strategies for making DNA computations more efficient, robust, and error-resistant.

One particularly promising scheme is known as the sticker model of DNA computation. Developed by Adleman, Sam T. Roweis of the California Institute of Technology in Pasadena, and their coworkers, this approach involves the use of short pieces of DNA (stickers) that adhere to complementary segments of much longer, or memory, strands of DNA (SN: 6/22/96, p. 391). With its sticker attached, a memory strand corresponds to a 1; without the sticker, it corresponds to a 0.

In putting together their sticker model, the researchers addressed a number of concerns, including errors caused when stickers spontaneously pop off a memory strand. They noted that the recent development of a DNA analog known as peptide nucleic acid (PNA) offers a possible solution to this problem. Stickers made from short PNA strands are likely to act with greater specificity and to adhere more firmly than their DNA counterparts.

In general, says Caltech's Erik Winfree, every time researchers encounter a major obstacle, they manage to find ways around it.

A group at the University of Wisconsin-Madison has adopted an alternative approach to reduce the errors in DNA computing. Instead of dealing with DNA molecules floating in a liquid, these researchers work with strands anchored to a surface. This strategy cuts losses during purification steps, reduces interference between different DNA strands, and eases the handling of samples, but it also decreases the potential scale of the computations and slows down some DNA manipulations.



Short segment of a DNA molecule.

The group brings together specialists in computer science, materials science, surface chemistry, and DNA sequencing and interactions. "It's a very interesting team," says Max G. Lagally, a member of the Wisconsin group. "We don't know if we can build a DNA computer, though we think we can. We do know that we can learn a lot of fascinating science along the way."

Another member of the team, computer scientist Anne E. Condon, has developed procedures using a special set of DNA manipulations applicable to anchored strands that she predicts will be valuable for solving a particular class of mathematical problems. Chemist Qinghua Liu is already experimenting with some of these operations on DNA molecules attached to a glass slide.

Still lacking, however, is a significant body of lab work demonstrating the feasibility of DNA operations on a scale anywhere close to that contemplated by computer scientists. Only a few groups have even started an experimental program.

"It's a tricky area to get into," Seeman says. "It may take a few years before there is enough experimental work to able to say what works and what doesn't."

Researchers are generally optimistic that something useful will emerge out of research on DNA-based computing. At present, however, no one has come up with an application that would justify a major effort to develop such a computer.

"The trouble with current examples of DNA computations is that none of the applications is compelling yet," Lipton admits. "None is clearly so important that it by itself justifies the construction of DNA computers."

Lipton and his colleagues have put a lot of effort into identifying what such a "killer application" might be. For example, they have shown that, in principle, sequences of DNA operations could be used to crack a widely used, powerful cryptographic scheme known as the Data Encryption Standard.

The trouble in practice is that by the time DNA separations can be done with sufficient precision to break the Data Encryption Standard, advances in electronic computing could easily wipe out any possible advantages of the DNA approach.

Nonetheless, even without a killer application, research on DNA computing may still have valuable spin-offs. For example, the complexity and precision of DNA processing required for computation could serve as an incentive for improving DNA manipulation techniques commonly used in the biology lab.

The real benefit so far has come out of the need for scientists from different fields to work together. "The most exciting thing for me has been the interaction between computer scientists and biologists," Roweis says.

"Biology, particularly at the molecular level, can be viewed for many purposes as an information science," Karp notes. "To understand the cell, the brain, or the immune system, you sometimes have to view it as a very complex information-processing system."

This interaction of computer science and biology has benefits for the burgeoning field of computational molecular biology. Extracting meaning out of the rapidly accumulating quantities of data about genetic structure, molecular processes in living cells, and other biological systems requires massive computation and the development of new computational procedures for recognizing and interpreting patterns.

"There are tons of problems out there where computer science can play a role," Karp says. The ideas that are coming out of DNA-based computing could contribute to the Human Genome Project and other efforts to understand the genetic database of life. □