BIOLOGICALLY INSPIRED COMPUTING

By Nancy Forbes

HUMAN BEINGS HAVE BEEN CALCULATING SINCE BEFORE 2000 BC. DRIVEN BY NECESSITY, CHANCE, OR INVENTIVENESS, WE HAVE PROGRESSED FROM USING OUR FINGERS TO MARKING ON TABLETS, TO THE ABACUS, TO THE MECHANICAL ADDING machine, and eventually, to the electronic computer. The modern computer has had a staggering impact on our ability to compute, letting us create amazingly complex algorithms to solve problems—sometimes in a matter of milliseconds.

The computer's influence on people's lives—their work, play, and communications—is well documented. Perhaps less commonly known is its impact on the economy. A number of economists attribute the current trend of steady growth and accelerating productivity in large part to improvements in computer productivity.[1] In 1999, US companies spent over $200 billion on computers and related items, more than they invested in any other type of capital good.[2]

Nevertheless, today's computers have their limitations. With transistors on a silicon chip doubling in number roughly every 18 to 24 months, shrinking device size and increasing density will cause some physical problems in the next 10 to 20 years, despite bringing large speed and functionality increases with a substantial drop in costs. Given the small size of devices (about 180 nanometers (nm)) in traditional computer architectures, scientists predict eventual electron leakage across a small number of atoms, lack of uniformity in the distribution of dopants in semiconductors, low yields in the number of usable chips manufactured, and heat dissipation from high densities of devices on the chip. Furthermore, as chip manufacturing becomes more complex, the cost of fabrication will rise to where it's no longer economically feasible (today, an IC-manufacturing facility can cost from $1 billion to $2 billion). Tom Theis, Director of Physical Sciences at IBM Research, estimates that

> when silicon microelectronics reaches ultimate physical limits to further miniaturization, the smallest silicon transistor will still contain over a million atoms. Yet, there is no fundamental reason why switches that process information cannot be made far smaller than that.

If the pace of computer cost and performance improvements continues long term, researchers will have to fully explore these issues and possible alternative technologies.

An alternative

Computing systems inspired by biological systems (biocomputation) are one possible alternative currently being investigated. Whether it will impact information technology as defined today is still unclear. The field of biocomputation has a twofold definition: the use of biology or biological processes as metaphor, inspiration, or enabler in developing new computing technologies and new areas of computer science; and conversely, the use of information science concepts and tools to explore biology from a different theoretical perspective. (Biocomputation, as defined here, doesn't include the use of computers, analysis, or data management in biology—for example, bioinformatics or computational biology. Although artificial neural networks and genetic and evolutionary algorithms fit the definition given here, they're also not included for space reasons.) The field is highly multidisciplinary, attracting a host of extremely bright computer scientists, molecular biologists, geneticists, mathematicians, physicists, and others. In addition to its potential applications, such as DNA computation, nanofabrication, storage devices, sensing, and healthcare, biocomputation also has implications for basic scientific research. It can provide biologists, for example, with an IT-oriented paradigm for looking at how cells "compute" or process information, or help computer scientists construct algorithms based on natural systems, such as evolutionary and genetic algorithms.

Biocomputing has the potential to be a very powerful tool. For example, when compared to conventional computers, certain operations in DNA computing (for example, hybridization—the bonding of two DNA strands to form the double helix) are over a billion times more

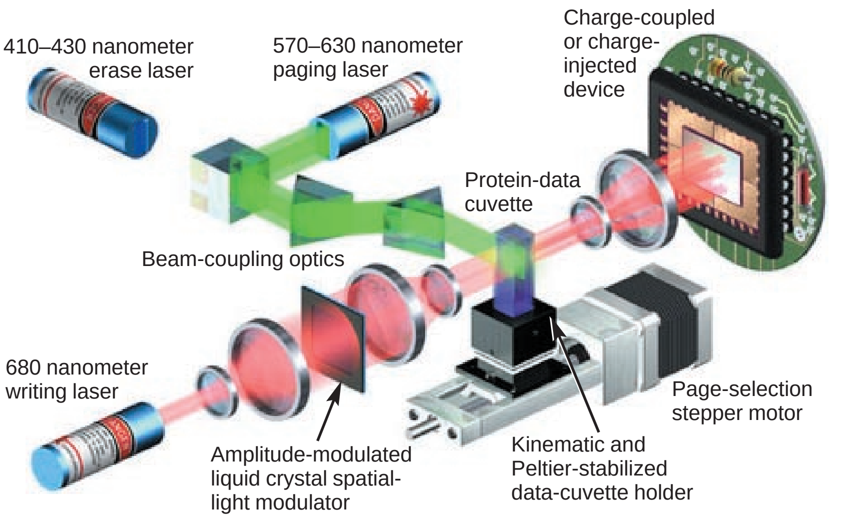

Figure 1. A 3D-optical-memory device based on bacteriorhodopsin. The protein is encapsulated in a hydrogel sealed in a plastic protein-data cuvette. The two lasers that are needed to store information using baceteriorhodopsin are fired with the appropriate temporal separation. The paging laser initiates photochemistry and primes the protein to receive the information. The orthogonally fired writing laser is spatially encoded with the information to be stored. This architecture's two main advantages are data throughput (due to parallel writing and reading operations) and memory-medium cost. The protein can be isolated and processed with minimal cost.

energy efficient. Also, DNA stores information at a density of about one bit per nm^3—about a trillion times as efficiently as videotape. DNA computing is also massively parallel and can reach approximately 10^{20} operations per second compared to today's teraflop supercomputers—although some in the field question these claims. Says George Church, Professor of Genetics at Harvard Medical School and biocomputing practitioner,

> Manufacturing and energy costs scale with size in DNA computing, so there are real limitations to DNA parallelism which are set by size—i.e., number of atoms used in the operation. None of the DNA experiments address this yet.

Another subdiscipline is bioelectronics, which uses biological molecules such as bacteriorhodopsin in electronic or photonic devices. This protein is being used for thin films, biosensors, and associative volumetric memories. Says Bob Birge, Director of the W.M. Keck Center for Molecular Electronics at Syracuse University,

> Evolution and natural selection have optimized many biological molecules to perform tasks required for certain device applications, while self-assembly and genetic engineering can provide the sophisticated control and manipulation needed for large molecules.[3]

Birge and his group have been developing bacteriorhodopsin-based holographic memory devices (see Figure 1). He chose this molecule because of its excellent holographic properties and efficiency in information storage, and because nature designed it to function in high temperature and intense light.

Interest in biological systems as a metaphor or inspiration for computing machines is not new—from the mid-1940s, John von Neumann spent considerable time studying how biological systems processed information, and used this to help formulate his theory of automata, both natural and artificial. Speaking at Caltech in 1948, he remarked,

> Natural organisms are, as a rule, much more complicated and subtle, and therefore much less well understood in detail, than are artificial automata. Nevertheless, some regularities which we observe in the organization of the former may be quite instructive in our thinking and planning of the latter.[4]

Perhaps today is the right time for these developments. Discoveries over the last 20 years have given us a better understanding of fundamental biological mechanisms at the molecular level—enabled by methods of investigation such as X-ray diffraction and electron microscopy—revealing phenomena unanticipated by classical theories, such as the structure and function of genes and the ways they can be modified by recombination and other mechanisms. Additionally, the engineering trend toward microelectromechanical systems and nanotechnology—making smaller devices at cheaper cost—is also conducive to the fusing of these two disciplines.

DNA computing

One much-talked-about application of biocomputing involves using DNA strands to compute. Although it's highly unlikely that people will send e-mail from their DNA computers in the next 50 years, it might one day have value for highly specialized computing applica-

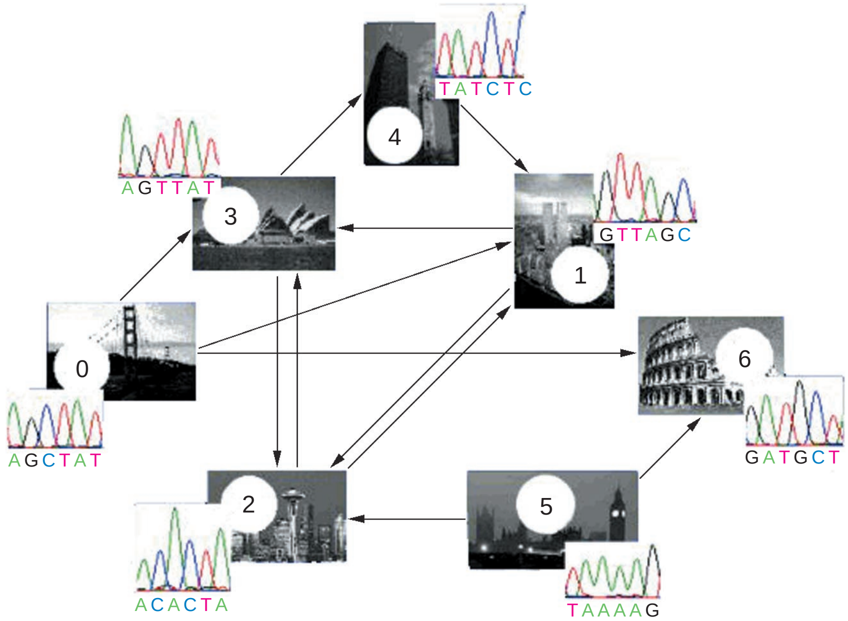

Figure 2. DNA computing solved the seven-city directed-path problem by reacting coded DNA strands, as represented by the letter sequences next to the pictures, to imprint a successful route on the product material.[6]

tions, while the insights it yields might pave the way for new biological or cellular technologies. The brainchild of computer scientist Leonard Adleman (one of the originators of the RSA public-key cryptosystem) of the University of Southern California in 1994, DNA computation relies on devising algorithms that solve problems using the encoded information in the sequence of nucleotides that make up DNA's double helix—the bases adenine, guanine, thymine, and cytosine (A, G, T, and C, respectively)—and then breaking and making new bonds between them to reach the answer.[5]

Adleman used DNA to solve a version of the well-known traveling salesman problem, which involves finding the shortest traveling distance between the seven cities (see Figure 2). Adleman created DNA strands to represent an airplane flight from each of seven cities, then combined them to produce every possible route. Given its vast parallelism, the DNA strands yielded 10^9 answers in less than one second—but the problem of separating the correct ones from the incorrect remained. Approximately seven days of painstaking lab work were required to separate out the right answers. The difficulty of identifying the correct answer, compounded by the errors inherent to the technique might put considerable obstacles in the path of future progress. Moreover, no killer application has emerged so far, which might spur progress in the field.

Other obstacles arise because lots of DNA (possibly even pounds) would be required for practical DNA computing. In addition, carrying out a typical algorithm could mean thousands of biological steps in a wet lab, compounded by errors. One observer believes this means that DNA computing is at a dead end, claiming,

> While the melding of biology and computer science may be good for idea generation, the workings of biological and silicon devices are so fundamentally different, that cross-hybridization of the field is largely untenable.

However, biocomputing practitioner Erik Winfree of Caltech still dreams of the "one-pot reaction." He comments,

> We'd like to be able to throw a bunch of reactants together and watch them self-assemble without any more purification, separation, or poking and prodding—other than something simple like heat cycling. The problem is that setting up the kinds of chemical reactions that are rich enough to support computation is hard to do without cross talk between them. Cells manage to decrease cross talk by compartmentalizing chemical reactions physically with membranes or "virtually" by segregating them in the nanoenvironments of enzymes.

Since Adleman's original experiment, many accomplished scientists have been drawn to the biocomputing challenge. For instance, Adleman and his group are currently exploring an *exquisite-detection* assay that detects as few as one molecule in a background of 10^{20} (for references to this project, other projects mentioned in this article, and other resources, see the "Additional resources" sidebar). Computer scientist Richard Lipton, at Georgia Tech, has shown how DNA computing could be used to solve NP-hard problems and the Boolean satisfiability problem, central to computing and complexity theory.[7] Additionally, both he and Adleman have proposed DNA computing algorithms to break the Data Encryption Standard, an encryption methodology the National Security Agency developed.[8,9] A group at the University of Wisconsin has been

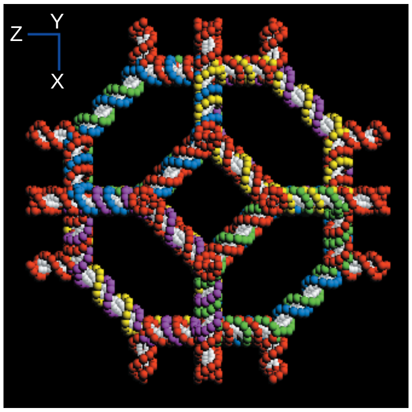

Figure 3. A truncated octahedron constructed from DNA in Ned Seeman's lab at NYU. Ultimate goals for this approach include the assembly of a biochip computer and the synthesis of periodic matter. Using this methodology to do DNA computations might also be possible.

performing DNA computations on surfaces.[10] John Reif of Duke is working on several biocomputing applications, including DNA self-assembly techniques, and using DNA as a data storage medium with an elaborate associative search methodology.[11] Carter Bancroft at Mt. Sinai School of Medicine focuses on DNA steganography, a method of using DNA to encode hidden messages.[12] Biologist Laura Landweber's lab at Princeton explores the question of how cells compute.[13] She notes,

> Cells read and "rewrite" DNA all the time, by processes that modify sequence at the DNA or RNA level. My laboratory combines three approaches—comparative sequence analysis, functional in vitro selection experiments, and computational biology—to study early molecular evolution and the origin of biological information processing.

Another subfield is DNA self-assembly, where the bonding properties of DNA molecules are used to build uniform DNA structures (DNA tiling), which could possibly serve as scaffolds for constructing nanostructures, as in Ned Seeman's work at NYU (see Figure 3). Additionally, scientists such as Winfree have used self-assembling DNA as a first step toward programming molecular reactions and molecular structures.

Amorphous computing is a biocomputing project undertaken by Hal Abelson, Tom Knight, Gerry Sussman, and their students at the MIT Artificial Intelligence Lab. Essentially, it's an enabling technology for building and organizing information processing systems comprising many molecular-scale components. From an engineering standpoint, it explores the principles that underlie a swarm of bees cooperating to form a coherent collective in a hive, for example, or the analogous behavior of cells. The group also seeks to address the problem of fault tolerance in computing by devising an amorphous system that can adapt to its environment, function with redundancy, and even self-repair. Says Knight,

> If you open up a computer and damage one small component, it stops working. Biological systems, on the other hand, can absorb faults and malfunctions, and don't have this degree of fragility.

One approach they've taken is to build *computational particles*—integrated microchips that combine logic circuits, microsensors, actuators, and advanced communications devices on the same chip. Using amorphous computing, they see this technology being used in applications such as *smart paint*, where many of these participles are mixed with bulk materials, such as paints, gels, and concrete, at a very low cost. By coating, for example, bridges or buildings with this paint, the chips could compute to sense data, act on the data, and externally communicate the data's actions.

Biological hardware

Biological hardware is the use of biological components or mechanisms that are engineered to function as computer components. This includes

Additional resources

Amorphous Computing: www-swiss.ai.mit.edu/projects/amorphous
A Bibliography of Molecular Computation and Splicing Systems: www.wi.leidenuniv.nl/~pier/dna.html
Church Lab: http://twod.med.harvard.edu
Erik Winfree: http://hope.caltech.edu/winfree
Erik Winfree's Molecular Computation Page (DNA computing): http://hope.caltech.edu/winfree/DNA.html
IBM Journal of Research & Development, Vol. 44, No. 3 ("Directions in Information Technology"): www.research.ibm.com/journal/rd/443/tocpdf.html
Laboratory for Molecular Science at USC: www-scf.usc.edu/~pwkr/index.html
Laura Landweber's Spring 1999 course at Princeton, DNA Computing: The Origin of Biological Information Processing: www.princeton.edu/~lfl/frs.html
Nadrian C. Seeman: http://seemanlab4.chem.nyu.edu
Tom Knight: www.ai.mit.edu/people/tk/tk.html

everything from biological logic gates to memory and storage devices, clocks, processors, and toggle switches. Some of those devices might end up taking the place of conventional computer hardware, operating in the same way within a hybrid computing system as the silicon elements they replace. Other types of biological hardware are being designed to remain in vivo, enabling cells to be programmed to do specific tasks. One of the real advantages of these biological (and molecular) devices is the ability to design and fabricate them from the bottom up, providing a level of control through genetic engineering or biological synthesis that methods such as traditional lithography could probably never have provided. Furthermore, biological engineering means potentially reducing the size of certain devices to levels practically unreachable by silicon.

With engineered in vivo biological hardware, scientists talk of one day being able to program cells to perform tasks such as injecting the proper amount of insulin into a diabetic's bloodstream or detecting the presence of a particular toxin within the body. In this area, Tom Knight, Gerry Sussman, and Ron Weiss at MIT are constructing an in vivo digital inverter in living cells. In Church's lab at Harvard Medical School, they are investigating generalizable in vivo protein codes that could be used as a molecular switching device, with the long-term potential of developing them into self-replicating computational systems. Tim Gardner at Cellicon Biotechnologies in Boston is continuing work on an on–off switch in E. coli that lasts through many cell divisions for up to 20 hours, which he terms a *genetic applet*. Michael Elowitz and his colleagues at Rockefeller University have engineered a genetic cycle that works by repressing certain reactions regularly, thus functioning like a clock.[14,15] These in vivo synthetic genetic circuits, says Gardner, can help monitor and control cell function.

Only time will tell where all this will lead. Says IBM's Theis,

> I don't feel that many of these research directions will impact information technology as we now define it; however, much of it will define entirely new directions for information technology.

Len Adleman concurs:

> Whether or not DNA computers will ever become stand alone competitors for electronics computers—which is unlikely—is not the point. Every living cell is filled with thousands of incredibly small, amazingly precise instruments in the form of molecules, which comprise "Nature's Tool Chest." I believe things like DNA computing, along with the other ways we are learning to use these wonderful tools, will eventually lead the way to a "molecular revolution," which ultimately will have a very dramatic effect on the world.

References

1. E. Brynjolfsson and L. Hitt, "Beyond Computation: Information Technology, Organization Transformation, and Business Performance," *J. Economic Perspectives*, Vol. 14, No. 3, Fall 2000.
2. Charles C. Mann, "The End of Moore's Law," *Technology Rev.*, Vol. 103, No. 3, May/June 2000, p. 48.
3. R.B. Birge et al., "Biomolecular Electronics: Protein-Based Associative Processors and Volumetric Memories," *J. Physical Chemistry B*, Vol. 103, No. 49, 1999, pp. 10746–10766.
4. L. Jeffress, ed., *Cerebral Mechanisms of Behavior*, Hafter, New York, 1967, p. 1.
5. L.M. Adleman, "Molecular Computation of Solutions to Combinatorial Problems," *Science*, Vol. 266, 1994, pp. 1021–1024.
6. L. Kari and L.F. Landweber, "Computing with DNA," *Bioinformatics Methods and Protocols*, S. Misener and S. Krawetz, eds., Humana, Totowa, N.J., 1999, p. 415.
7. R. Lipton, "DNA Solution to Hard Computational Problems," *Science*, Vol. 268, No. 5,210, Apr. 1995, pp. 542–545.
8. R. Lipton et al., *Breaking DES Using a Molecular Computer*, Tech. Report TR-489-95, Computer Science Dept., Princeton University, Princeton, N.J., 1997; http://ncstrl.cs.Princeton.edu/expand.php3?id=TR-489-95 (current Oct. 2000).
9. L. Adleman et al., "On Applying Molecular Computation to the Data Encryption Standard," *DNA-Based Computers II*, Vol. 44, L. Landweber and E. Baum, eds., Am. Math. Soc., Providence, R.I., 1999, pp. 31–44.
10. Q. Liu et al., "DNA Computing on Surfaces," Nature, Vol. 403, No. 6,766, Jan. 2000, pp. *175–179*; http://corninfo.chem.wisc.edu/writings/DNAcomputing.html#overview (current Oct. 2000).
11. T.H. LaBean et al., "The Construction, Analysis, Ligation, and Self-Assembly of DNA Triple Crossover Complexes," *J. Am. Chemistry Soc.*, Vol. 122, No. 9, 2000, pp. 1848–1860; www.cs.duke.edu/~reif (current Oct. 2000).
12. C. Bancroft et al., "Hiding Messages in a DNA Microdot," *Nature*, Vol. 399, No. 6,736, 1999, pp. 533–534.
13. D. Faulhammer et al., "Molecular Computation: RNA Solutions to Chess Problems," *Proc. Nat'l Academy of Science USA*, Vol. 97, No. 4, pp. 1385–1389.
14. T.S. Gardner et al., "Construction of a Genetic Toggle Switch in *Escherichia Coli*," *Nature*, Vol. 403, No. 6,767, 2000, pp. 339–342.
15. M.B. Elowitz and S.M. Leibler, "A Synthetic Oscillatory Network of Transconscriptional Regulators," *Nature*, Vol. 403, No. 6,767, 2000, pp. 335–338.

Nancy Forbes is a senior member of the Technical Staff at Litton TASC, where she supports the research efforts of the military and intelligence communities in assessing emerging technologies. She has an MA in physics from Columbia University. She is a contributing editor for AIP's *The Industrial Physicist* and is writing a book on biologically inspired computation for the MIT Press. Contact her at nforbes@erols.com.