On the Path to Computation with DNA

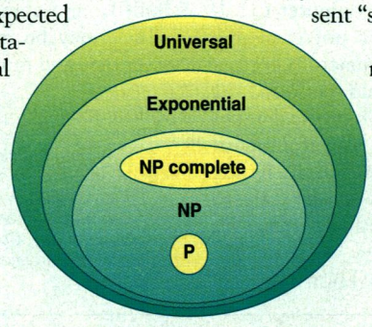
David K. Gifford

On page 1021 of this issue, Adleman demonstrates how simple DNA manipulations can be used to find the solution to a directed Hamiltonian path problem (*1*). For computer scientists it shows that so-called "NP-complete" computations can be performed with simple protocols on DNA substrates. For biologists, the results indicate that simple biological systems have the ability to compute in unexpected ways. Such new computational models for biological systems could have implications for the mechanisms that underlie such important biological systems as evolution and the immune system.

Hundreds of difficult practical computational problems (*2*), including optimal shop scheduling, the longest path in a graph, and Boolean logic satisfaction, can be translated into a classic form known as the directed Hamiltonian path problem. Simply stated, a path through a graph is Hamiltonian if it visits each vertex exactly once. Finding such paths for complex graphs is hard.

Adleman has approached the Hamiltonian path problem in a biological context where each vertex and edge of the graph can be represented by a short oligonucleotide sequence. An oligonucleotide is a synthetic single stranded DNA molecule that is made with a chosen sequence. Adleman shows that the binding together of chosen oligonucleotides representing vertices results in DNA molecules that encode the solution to a Hamiltonian path problem (*2*). A complete solution is encoded in a single DNA molecule. Adleman further describes how to choose the right oligonucleotides to "input" a problem into his method, and how to isolate and interpret the DNA molecules that represent solutions which are the "output" of his method. In essence, Adleman has used the enormous parallelism of solution-phase chemistry to solve a hard computational problem.

The take home lesson for molecular biologists is that DNA ligation can effectively search a large space of potential solutions to a given problem. It is conceivable that similar protocols could be developed for protein or epitope design given an efficient way to isolate molecules that represent "solutions."



Subsets of difficulty. The diagram shows how the complexity of computational problems is classified. Adleman used DNA to solve the equivalence class of computational problems called "NP complete."

The limitations of Adleman's technique are yet to be discovered. In its present form, it is not practical enough to replace traditional computers even for the particular problem that Adleman has solved. It is safe to assume that future research will explore these limitations and create related methods that reduce the number of manipulations and the elapsed time required. Once these limitations are removed, biological methods could become a practical method for solving real-world computational problems.

The simplicity of Adleman's method is surprising given that it solves a hard computational problem. The figure shows how computer science problems are ranked in difficulty, starting with easy polynomial time problems at the inside, ranging up to difficult universal machines at the outside. This ranking is based on how long the best algorithm to solve a problem will take to execute on a single computer bounded by a constant factor times the size of the problem. Algorithms whose running time is bounded by a polynomial function are in the complexity class "polynomial time" (P for short), and algorithms whose running time is bounded by an exponential function are in the complexity class "exponential time."

A problem is referred to as intractable if there is no polynomial time algorithm that can solve it. It is easy to see why. For example, if an $O(n^2)$ algorithm takes 1 μs to solve a problem of size 10, then it will take 100 μs to solve a problem of size 100. If an $O(2^n)$ algorithm takes 1 μs to solve a problem of size 10, then it will take 3.9 times 10^{11} centuries to solve a problem of size 100.

The directed Hamiltonian path problem is in the complexity class called "nondeterministic polynomial time," or NP for short. An NP problem is a problem that can have its answer checked in polynomial time. The checked part is the important clause. If an oracle can tell you the answer to your problem and you can verify the answer in polynomial time, then your problem can be solved in principle with Adleman's method. The "oracle" in Adleman's method is the immense computational capacity of a ligation reaction that produces billions of products and by brute force tries all possible solutions.

The directed Hamiltonian path is a special kind of problem in NP known as "NP-complete." Any problem in NP can be translated into an NP-complete problem in polynomial time. Thus, NP-complete problems are the granddaddies of the NP family and can be used to solve any problem in the class. One of the largest open questions in computer science is the question of whether NP hard problems can be solved in polynomial time. This mathematical question is written "P = NP?". At present, no polynomial-time solution is known for NP-complete problems, and thus, they are considered to be intractable.

Given the discussion to this point, one would believe that computers by and large run polynomial-time problems. This in fact is not the case. Many important problems are NP complete. To make these NP-complete problems practical, their sizes are limited, and acceptable approximations are made in order to trade precision for execution time.

What are the implications of Adleman's findings? From a computations point of view, the power of natural systems to solve a problem in parallel by brute force is compelling. In the case of Adleman's approach, billions of ligation reactions proceed in parallel, and only a small number of them need to result in molecules that encode a solution to the problem at hand. The power of DNA amplification (by the polymerase chain reaction technique) is used to selectively enrich these solution molecules, and is followed by a purification procedure that isolates solution molecules.

Logical future work following Adleman's will seek to provide direct translations for other computational problems into biological systems and will seek to find new ways to identify and output solutions. To be practical, Adleman's method needs its "output" technique to be less labor-intensive. In addition, negative controls need to be run to ensure that the method does not find solutions when there are no solutions. The method also needs to be tried on larger problems to test its limitations well beyond a small graph with 8 nodes

The author is in the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA.

and 15 edges, which Adleman used for his experiments.

Despite these limitations, the compelling aspect of this type of computational system is that an entire result that describes a complete solution is encoded in a single molecule. This coding enables an information representation density that is unheard of in conventional computers and also permits extremely energy-efficient computation; both of these issues are discussed by Adleman (*2*).

Evolution and NP-complete problems do not seem to have much in common with one another. Or do they? In the case of Adleman's method, he observed that adapter oligonucleotides could constrain a random process toward a solution, even though in a strict sense, constraints are not necessary with adequate screening of solution candidates. Might similar processes be at work in biological systems that evolve? In such a scenario, genetic plasticity would still be created by random events, but constraints might direct mutations to make desired outcomes more probable. If such constraints exist, understanding them would certainly be a major accomplishment.

A second lesson is that computation can take on forms that are not immediately recognizable to us as computation. We have seen how a process as ordinary as ligation can yield solutions to a hard computational problem. Possible computational applications of other common enzymatic reactions have yet to be fully explored, and thus, it is worthwhile keeping an open mind about the nature of computation in a cell. Transcriptional control and other gene regulation mechanisms certainly play a paramount role in the programming of cell behavior, but there may be other computational mechanisms lurking behind seemingly simple biological processes.

As shown in the figure, NP complete problems are not the most powerful computational systems known. This honor is held by so-called universal systems, which can simulate any computation that can be performed on a deterministic computer (*3*). If we were able to construct a universal machine out of biological macromolecular components, then we could perform any computation by means of biological techniques. There are certainly powerful practical motivations for this approach, including the information-encoding density offered by macromolecules and the high energy efficiency of enzyme systems.

At present, there is no known way of creating a synthetic universal system based on macromolecules. Universal systems require the ability to store and retrieve information, and DNA is certainly up to the task if one could design appropriate molecular mechanisms to interpret and update the information in DNA. This ultimate goal remains elusive, but once solved, it will revolutionize the way we think about both computer science and molecular biology.

A great hope is that as we begin to understand how biological systems compute, we will identify a naturally occurring universal computational system. Understanding such a system would give us unprecedented insight into complex biological processes. Perhaps we will ultimately discover that developmental programs and other intricate biological behaviors are built from a common vocabulary of idioms which may be of value to both computer scientists and molecular biologists.

References

1. L. Adleman, *Science* **266**, 1021 (1994).
2. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, New York, 1979).
3. M. L. Minsky, *Computation: Finite and Infinite Machines* (Prentice-Hall, Engelwood Cliffs, NJ, 1967).

Neuroscience on the Net

Peter T. Fox and Jack L. Lancaster

Sure, the Internet provides low-cost entertainment: transcontinental trivia browsing by information junkies; late-night, on-line chat (Fig. 1); electronic junk mail; and even pornography. But a large and growing community of "wired" neuroscientists have found loftier ways to use the Net.

No one will deny that conversation is an important aspect of Net traffic. Discussions with colleagues further away than the next lab are usually by e-mail, instantaneous but buffered. Like traditional mail, you reply in your own time. Person-to-person data transmissions, once cumbersome, are now commonplace. Manuscripts, graphics, and massive data sets hurtle around the world guided by point-and-click interfaces, such as Mosaic, Gopher, and Fetch. This ease of access is complemented by a similar ease of creation. Thousands of laboratories have crafted WWW (World Wide Web) "Home Pages," which provide paths to research program information, preprints, public databases, software, and the like (*1*).

The authors are at the Research Imaging Center, University of Texas Health Science Center at San Antonio, San Antonio, TX 78784–6240, USA.

Publication, too, is being revolutionized by the Net. Submission by diskette, yesterday's leading edge, is being rendered obsolete by e-mail submission. Still more avant garde are Internet journals. There are now over 70 fully electronic, peer-reviewed, scholarly journals (*2*). *Psycholoquy*, the most established electronic journal of neuroscience, uses the Net for every aspect of publication: submission, peer review, revision, and distribution. Although revolutionary, electronic publishing is probably not the Internet's most far-reaching restructuring of scientific communication.

Community databases open to all members of a scientific discipline offer the greatest potential for scientific exploitation of the Net. It takes but a moment to understand why. Envision this: On-line access to all relevant results produced by any laboratory in the world, before designing your next experiment. Alternatively, imagine similar access to aid in interpreting an unexpected result. Such is the goal. How do we get there?

The genome community has databased via the Net for roughly a decade. Before proceeding too far, prospective developers of neuroscience databases should absorb the collective wisdom of these network pioneers. There are dozens of public genetics databases. Most begin as in-house compilations and, when successful, evolve into "collaborative" databases (that is, allowing off-site access by formal agreement). The

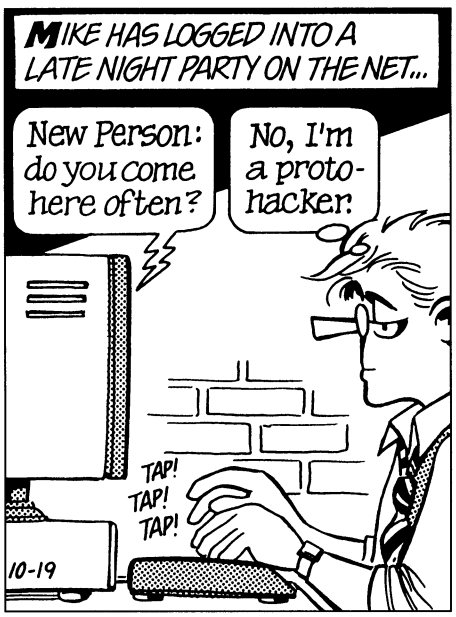


Fig. 1. The Internet as entertainment. [Doonesbury © 1993 G. B. Trudeau. Reprinted with permission of Universal Press Syndicate. All rights reserved]