The Evolution of Molecular Computation

Leonard M. Adleman (1) and Richard J. Lipton (2) describe how molecular computing could be used to solve complex problems by physical selection of the correct sequence string from a single huge library of DNA sequences. However, others (3) point out that even a simple 23-node Hamiltonian path problem requires kilogram amounts of DNA. The size of the DNA library needed to solve a 100-node Hamiltonian would approximate the number of particles in the universe (10^{70}).

Does this mean that molecular computing is impractical for solving very complex problems? To answer this question, it is useful to compare molecular computing to natural biological evolution and its experimental derivative, in vitro evolution (4, 5): The process of physical selection of the best sequences from large pools of sequences is similar in both settings. In natural as well as in in vitro evolution, these pools of sequences are relatively small (5 $\times$ 10^9 people on Earth; 10^{15} for aptamers). Yet, selection from such modest populations has yielded the vast complexity of biological DNA sequences. The explanation is that multiple, recursive cycles of selection from small pools can "compute" complex answers that far exceed the capacity of any single library of sequences.

Whereas the published examples of molecular computing focus on selection of a complex answer by screening all of the individuals in a single huge pool of sequences, evolution uses many recursive cycles of selection from pools of modest size. Recursive means that the best (partial) solutions selected from one pool are amplified and mutated to form the next pool. The natural mutation process consists of sexual, homologous recombination, with a low frequency of point mutation. For in vitro evolution, the recombination can be obtained by sexual polymerase chain reaction (6). Recursive selection and recombination was

shown to be sufficient to "compute" a sharply improved enzyme (4).

Consider the difficulty of designing sequences such as the human genome. The human genome codes for approximately 100,000 proteins. Even a small protein of 300 amino acids has a sequence space of 20^{300} or 10^{390} (a "sequence space" being the number of possible sequences of a particular length). How complex would a random library need to be to contain a functional copy of, for example, β-lactamase? It appears that no library can be made large enough to allow one-step selection of an average, functional gene. For example, the size of a library of random sequence DNA needed to obtain a specific 8-bp restriction site, using the naïve algorithm given by Adleman (1), is 64,000. To obtain five different 8-bp restriction sites in specific locations would require a library of $(64,000)^5$ or 10^{24}, which is too large to be obtained from a single library. But the five sites could efficiently be obtained in five recursive selection cycles from a library of 10^5 to 10^6 bp, selecting for a different restriction site at each of five cycles.

The recursive approach allows one to move rapidly through a vast sequence space, sampling only a small fraction of all the sequences that need to be accessed with a single pool approach (five cycles of 10^6 versus 10^{24}). Only the most promising sequences are selected and used as the basis for further mutagenesis and selection.

Parallel access to a larger number of sequences was regarded as the main advantage of molecular computation over traditional computing (1–3). For those complex problems where smaller, recursive pools of sequences can be equally or more effective, molecular computation (with evolution) may be less promising than improved genetic algorithms, which simulate the same re-

Fig. 1. Applications of evolutionary engineering.

	Solution	
Tool	**Structure**	**Number**
DNA	In vitro evolution	Molecular computing
Chip	Artificial life	Genetic algorithms

cursive process of selection and recombination of pools of sequences, but run on existing computers (Fig. 1). In the end, the advantages of the different approaches depend on the ruggedness of the sequence space landscape of each particular problem (7).

Although DNA sequencing of the selected solutions poses a practical problem for molecular computation, this drawback does not exist for computation with genetic algorithms or for in vitro evolution or computer simulations of natural evolution, called artificial life (Fig. 1).

In summary, sequence evolution appears to be a useful general tool for solving many complex problems, whether the solution is a number, sequence, program, or structure.

Willem P. C. Stemmer
Affymax Research Institute,
4001 Mirando Avenue,
Palo Alto, CA 94304, USA
E-mail: pim_stemmer@qmgates.affymax.com

REFERENCES

1. L. M. Adleman *Science* **266**, 1021 (1994).
2. R. J. Lipton, *ibid.* **268**, 542 (1995).
3. M. Linial and N. Linial, *ibid*, p. 481; Y.-M. D. Lo, F. C. Yiu, S. L. Wong, *ibid*., p. 481; B. Bunow, *ibid*, p. 482.
4. W. P. C. Stemmer, *Nature* **370**, 389 (1994).
5. D. P. Bartel and J. W. Szostak, *Science* **261**, 1411 (1993).
6. W. P. C. Stemmer, *Proc. Natl. Acad. Sci. U.S.A.* **91**, 10747 (1994).
7. S. A. Kaufmann, *The Origins of Order* (Oxford Univ. Press, New York, 1993).

7 June 1995; accepted 14 September 1995

Science

The Evolution of Molecular Computation

W. P. C. Stemmer

ARTICLE TOOLS	http://science.sciencemag.org/content/270/5241/1510
REFERENCES	This article cites 8 articles, 7 of which you can access for free http://science.sciencemag.org/content/270/5241/1510#BIBL
PERMISSIONS	http://www.sciencemag.org/help/reprints-and-permissions

Use of this article is subject to the Terms of Service