Biocomputing: An Application for Micro-arrays?

Lila Kari
Department of Computer Science
University of Western Ontario
London, ON, Canada, N6A 5B7
lila@csd.uwo.ca, http://www.csd.uwo.ca/~ lila

Biocomputing, known also under the names of biomolecular computing, molecular computing and DNA computing, is a novel and fascinating development at the interface of computer science, mathematics, and molecular biology. It has emerged in recent years, not simply as an exciting technology for information processing, but also as a catalyst for knowledge transfer between information processing, nanotechnology, and biology. This area of research has the potential to change our understanding of the theory and practice of computing[1].

The main idea behind DNA computing is that DNA strands can be used to encode data and molecular biology techniques can be used to perform computations on that data.

Indeed, a (single-stranded) DNA strand can be viewed as a linear arrangement of four different building blocks, or bases: adenine, guanine, cytosine and thymine. In other words, a DNA strand can be thought of as a word over a four-letter alphabet, $\{A, C, G, T\}$ where two of the letters are complementary of the others. Indeed, A is the complement of T and C is the complement of G, and two complementary DNA single strands of opposite directionality will bind to each other to form a double-stranded DNA strand with is well-known helical shape. From the computational/informational point of view, this all amounts to the fact that we have at our disposal four symbols to encode information, which is more than sufficient considering that two bits, 0 and 1, suffice for the same purpose on an electronic computer. As synthesising a desired DNA strand is nowadays a routine procedure in molecular biology, we could think, for example, of a fictitious encoding of the letters of the English alphabet as A = ACA, B = $ACCA$, C = $ACCCA$, D = $ACCCCA$, the nth letter = AC^nA, and utilise this encoding to write any English text as a DNA strand. There are many reasons why this particular example would not work in practice, but it is an illustration of the fact that one could represent, with a suitable encoding, textual, numerical and symbolical information as DNA strands.

After encoding the information in DNA strands, one can use molecular biology lab techniques to perform operations. The so-called *bio-operations* that have so far been used for computations are:

- synthesis of a desired DNA strand;

- union: pour together the DNA (in solution) of two test tubes into a third one;

[1] See http://www.lcnc.nl/dna6/.

- separation of DNA strands by length from a given heterogeneous solution by using a technique called gel electrophoresis;

- "melting" of a double DNA strand into its constituent single strands and its opposite, "annealing" which amounts to binding together two complementary single strands with opposite orientation to form the corresponding double strand;

- separation, from a heterogeneous solution of DNA single strands, of those that contain a certain pattern as a subsequence, by using a technique called affinity separation;

- making copies of a given DNA strand by using PCR (Polymerase Chain Reaction);

- cutting a DNA double-strand at a specific location by using restriction enzymes;

- pasting together DNA strands with compatible "sticky-ends" by using DNA ligases;

- "reading out" or sequencing the letters of a DNA strand from a homogeneous solution, i.e. from a solution that contains mainly many copies of the same strand.

These bio-operations and combinations of them have been used to solve computational problems where the input to the problem is encoded as a collection of DNA strands, usually with many copies of each strand present in the solution, and the computation consists of a sequence of bio-operations. The output is a DNA solution which is ultimately sequenced to find out the answer.

The first attempt of solving a computational problem using DNA computing was Len Adleman's, [1], who reported the results of an experiment solving a 7- node instance of the Directed Hamiltonian Problem using only bio-operations. The Directed Hamiltonian Path Problem has as input a directed graph, and two designated nodes, "in" and "out". The question is whether this graph has a Hamiltonian Path, i.e. a path that starts at the "in" node, ends at the "out" node and enters every other node exactly once. Adleman's solution consisted in encoding each node as a 20-letter DNA single strand and then encoding the directed edges between nodes as follows: the edge ij was a strand consisting of the 2nd part of the strand encoding node i and the first half of the strand encoding node j. By this ingenious encoding scheme, when putting together in a test tube all strands that encoded for edges and all complement of strands encoding nodes, all possible paths through the graph were formed by the property of annealing of complementary strands. Indeed, the strand encoding for the edge ij would, by construction, bind to both the complement of strand i (in its first half) and to the strand representing the node j (in its second half). After all the possible candidates to the Hamiltonian Path were generated by self-assembly of DNA strands, by using successively some of the above mentioned bio-operations, the paths that were not Hamiltonian were eliminated.

Following Adleman's experiment, other experiments were proposed for solving various computational problems with DNA. For example, in [17] Lipton proposed a DNA algorithm for solving the Satisfiability Problem and other NP-complete problems. This started one of the directions in DNA computing research, that of building special purpose computers. A special purpose computer is a device that serves to solve efficiently a particular problem or class of problems. Much of the experimental research in DNA computing has been of this application-oriented type. Experiments using DNA molecules to solve computational problems that have actually been carried out in the laboratory include the Travelling Salesman Problem [1], [2], the Maximal Clique Problem [23], the Satisfiability Problem [18], [26], [22], the Knights' Problem [27], the Royal Road Problem [6], encryption and data security [3], etc. Each of these experiments is a step towards the design of a DNA-based device that would outperform its electronic counterpart for a spe-

cific application.

While special purpose DNA computers could provide a tailor-made solution for each particular application, a general purpose computer is a more ambitious project: a universal device capable of running any program and thus of solving any problem.

Theoretical studies, [8], [11], [10], [24], [25], have proved that the existing formal models of DNA computation are equivalent in computational power to Turing machines (the widely accepted formal model of electronic computers). This shows that, in principle, it is possible to design and build a DNA-based programmable computer, and that none of the existing practical obstacles is insurmountable.

Experimental research also has been directed towards investigating which tools from the molecular biologist's tool-chest are best suited for computational applications. The search for the optimal basic instructions of a future high-level molecular programming language include DNA-based addition of binary numbers [7] and computing DNA tiles that self-assemble [30], [20].

Lastly, another research direction is the study of DNA computing *in vivo*. The model developed by Landweber and Kari in [16], [14] for the guided homologous recombinations that take place during gene unscrambling proved to have the computational power of a Turing machine. This indicates that, in principle, these unicellular organisms may have the capacity to perform at least any computation carried out by an electronic computer. Moreover, this opens the possibility of envisaging a programmable cell which could be used for a variety of computational and medical purposes.

Weiss et al. [28] present another approach to *in vivo* computation by proposing a mapping from digital logic circuits to genetic regulatory networks with the following property: the chemical activity of such a genetic network *in vivo* implements the computation specified by the corresponding digital circuit.

In [29] the authors undertake a biological implementation of cell to cell communication. This work demonstrates the construction and testing of engineered genetic circuits which exhibit the ability to send a controlled signal from one cell, diffuse that signal through the intercellular medium, receive that signal within a second cell, and activate a remote transcriptional response.

In combination with other ongoing work in gene circuits [5], [4], [21], [12] the approach in [28], [29] provides components for a biological substrate for expressing pattern formation and for engineering with living organisms.

While most approaches deal with biocomputations that happen in a test-tube, the Madison team, [18], uses a surface-based approach based on DNA microarrays. The argument for using this approach is that all the solution-based methods share problems of scale-up for a number of reasons, including poor efficiencies in the purification and separation steps. In contrast, the surface-based computations manipulate strands that are immobilised on a surface using chemical linkers. This implies that at least one of the operations used in solution-based computing, that of selectively separating strands in different test-tubes, cannot be performed. As the surface-based approach is two-dimensional rather that three-dimensional, the number of DNA strands is limited to roughly 10^{12} per square centimeter, [19]. Nevertheless, this approach might gain in efficiency where it loses in data-compression, and a demonstration of solving an instance of a SAT problem has been reported in [19]. The surface-based computing uses three basic operations, MARK, UNMARK and DESTROY [19]. In the MARK operation, a combinatorial mixture of DNA corresponding to the query would be added to the surface and complementary strands

would bind: the marked strands would be duplexed while the unmarked ones would remain single-stranded. The DESTROY operation consists of adding an exonuclease specific for single-stranded DNA. Thus, every unmarked strand is destroyed, leaving on the surface only the MARKED DNA molecules. The UNMARK operation consists of subjecting the surface to conditions under which hybrids dissociate into single strands. Subsequent washing removes the free strands and regenerates the DNA modified surface.

After each cycle, fewer molecules remain on the surface. Repeated queries constitute the computation process, permitting subsets of the initial combinatorial solution space to be eliminated, and leaving the desired solution to the problem of interest. The READOUT operation consists of determining the sequence(s) of the surface-bound DNA molecules that remained. Both conventional gel-electrophoresis-based sequencing and hybridisation to word-specific addressed arrays have been studied [19].

The surface-based approach has recently been adopted also for the DNA implementation of successive state transitions, [13] as immobilising strands on a surface minimised the intermolecular reactions.

Besides the novelty of the approach, and in spite of the technical difficulties that arise from the error rates of bio-operations, there are several potential advantages to DNA computing over electronic computing. These include massive parallelism, memory capacity, and power requirements [10].

Indeed, due to its massive parallelism, a DNA computer could be between a thousand times and a million times faster than an electronic computer. Moreover, to encode the same information that can be stored in a micro-Mole of DNA (a dilute solution that fits in a 1 litre milk carton) using the current IBM technology, one would need a surface of 160 hectares. Concerning the power requirements, a DNA computer could be at least 1000 times more energy efficient than an electronic one. The comparisons above, while based on preliminary data, give a glimpse into why bio-molecules might be a preferred medium for computations in some applications. It is envisaged that in-vitro and in-vivo DNA computing research are preliminary steps that may ultimately lead to making DNA computing a viable complementary tool for computation and provide more insight into the computational capabilities of living organisms.

References

[1] Adleman, L. M. Molecular computation of solutions to combinatorial problems. *Science* 266(1994), 1021-1024.

[2] Adleman, L.M., Computing with DNA. *Scientific American*, 279(1998), 54-61.

[3] Clelland, C.T., Risca, V., Bancroft, C. Hiding messages in DNA microdots. *Nature*, 399(1999), 533-534.

[4] Elowitz, M., Leibler, S., A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(2000), 335-338.

[5] Gardner, T., Cantor R., Collins, J. Construction of a genetic togle switch in Esterichia coli. *Nature*, 403(2000), 339-342.

[6] Goode, E., Wood, D.H., Chen, J. DNA implementation of royal road fitness evaluation. Proceedings of the *DNA based computers 6*, Leiden, The Netherlands, 223-237.

[7] Guarnieri F., Fliss M., Bancroft C., Making DNA add. *Science* 273(1996), 220-223.

[8] Head, T. Formal language theory and DNA: an analysis of the generative capacity of recombinant behaviors. *Bul-*

letin of Mathematical Biology, 49(1987) 737–759.

[9] Kari, L., DNA computing in vitro and in vivo. In *Future generation computer systems*, Elsevier Science. In press.

[10] Kari, L. DNA computing: arrival of biological mathematics. *The Mathematical Intelligencer*, vol.19, nr.2, Spring 1997, 9–22.

[11] Kari, L., and Thierrin, G. Contextual insertions/deletions and computability. *Information and Computation*, vol.131, 1(1996), 47-61.

[12] Knight T., Jr., and Sussman, G.J. Cellular gate technology. *1st International Conference on Unconventional Models of Computation*, C.S. Calude, J.Casti, M.J. Dinneen, eds., Springer Verlag, 1998, 257-272.

[13] Komiya, K., Sakamoto, K., Gouzu, H., Yokoyama, S., Arita, M., Nishikawa, A., Hagiya, M. Successive state transitions with I/O interface by molecules. Proceedings of the *DNA based computers 6*, Leiden, The Netherlands, 21-30.

[14] Landweber, L.F. and Kari, L., The evolution of cellular computing: nature's solution to a computational problem. *Biosystems*, L.Kari, H.Rubin, D.Wood Eds, vol.52, Nos.1-3, 1999, Elsevier, Amsterdam, 3-13.

[15] Landweber, L.F., Kuo, T.C. and Curtis, E.A. Evolution and assembly of an extremely scrambled gene. *PNAS*, vol.97, no.7(2000), 3928-3303.

[16] Landweber, L.F. and Kari, L. Universal molecular computation in ciliates. In *Evolution as Computation*, L.Landweber, E,Winfree, Eds., Springer Verlag, 2000.

[17] Lipton, R.J. DNA solution of hard computational problems. *Science*, vol.268, April 1995, 542–545.

[18] Liu Q., Wang L., Frutos A.G., Condon A., Corn R.M., Smith L.M. DNA computing on surfaces. *Nature* 403(2000), 175-179.

[19] Liu, Q. et al. Progress toward demonstration of a surface based DNA computation: a one word approach to solve a model satisfiability problem. *Biosystems*, Elsevier, 52(1999), 25-33.

[20] Mao, C., Sun, W., Shen, Z., Seeman, N.C. A nanomehanical device based on the B-Z transition of DNA. *Nature*, 397(1999), 144-146.

[21] McAdams, H.H. and Arkin, A. Simulation of prokaryotic genetic circuits. *Ann. Rev. Biophys.Biomol.Struc.*, 27(1998), 199-224.

[22] Ogihara, M., Ray, A. Molecular computation: DNA computing on a chip. *Nature* 403(2000), 143-144.

[23] Ouyang Q., Kaplan P.D., Liu S., Libchaber A., 1997. DNA solution of the maximal clique problem. *Science* 278(1997), 446-449.

[24] Paun, G. On the splicing operation. *Discrete Applied Mathematics*, 70(1996), 57–79.

[25] Paun, G. On the power of the splicing operation. *International Journal of Computer Mathematics*, 59(1995), 27–35.

[26] Sakamoto K, Gouzu H, Komiya K, Kiga D, Yokoyama S, Yokomori T, Hagiya M., Molecular computation by DNA hairpin formation. *Science* 288(2000), 1223-1226.

[27] Seife C. Molecular computing. RNA works out knight moves. *Science* 287(2000), 1182-1183.

[28] Weiss, R., Homsy, G.E., Knight, T.F. Jr., Towards in vivo digital circuits. In *Evolution as Computation*, L.F.Landweber, E.Winfree, Eds, Springer Verlag, 2000.

[29] Weiss, R., Knight, T.F. Jr., Engineered communication for microbial robotics. Proceedings of the *6th International Meeting of DNA based computers*, Leiden, The Netherlands, June 13-17, 2000, 5-19.

[30] Winfree, E., Liu F., Wenzler L.A., Seeman N.C. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394 (1998), 539-544.