Computation with DNA

Alain Hénaut et Didier Contamine
Université Versailles-Saint-Quentin

March 25, 1996

[summary by Eithne Murray]

Abstract

In 1994 Leonard Adleman published a paper giving an algorithm to solve the Hamiltonian path problem using DNA manipulations and presented the results of an actual experiment applying this algorithm to a particular graph. The basic operations and the algorithm are described, and the potential of these methods as a means of computation is discussed briefly.

1. Introduction

Using basic techniques of DNA manipulation and standard lab equipment, Adleman finds a Hamiltonian path in a directed graph consisting of 7 nodes and 14 edges (figure 1). Finding such a path, that starts and ends at specified vertices while passing through every other vertex exactly once, is a problem which has no known polynomial time solution. In fact, this problem is NP-complete, and so it is considered unlikely that such a solution will exist. This is the first time biological methods have been used to solve hard computer problems, and it is still unknown to what extent the available DNA operations may be used to solve other problems.

2. Basic Operations

DNA manipulations form the basic operations operations of a DNA computer. It should be emphasized that the biological techniques presented here are routine laboratory procedures, and require no special equipment or expertise. Strands of DNA are made up of sequences of bases represented by the letters $\{A, C, G, T\}$. Each sequence has a (Watson-Crick) complementary sequence, that is, the sequence that binds with the original to form a double strand. In the complementary sequence, each base in the original is replaced by its complement ($A \leftrightarrow T$, $C \leftrightarrow G$).

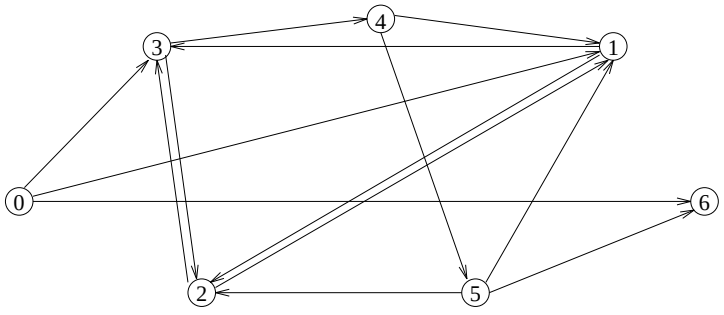

FIGURE 1. The directed graph used in the experiment.

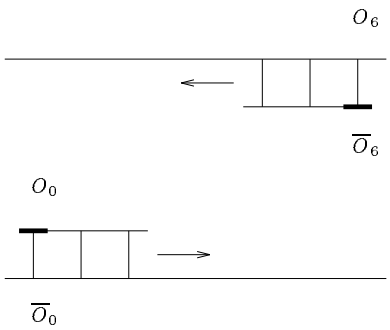

FIGURE 2. PCR - Polymerase Chain Reaction to replicate a specified segment of DNA

The following operations are available. Each one will be discussed briefly, without entering into too many technical details.

creation: A strand of DNA made up of a given sequence of bases can be created. These days, creating a specific short sequence is a matter of filling out the mail-order coupon, writing the check, and sending them off to the laboratory in the catalogue. Here "short" often means of length 20.
joining: Complementary strands will spontaneously join together to form a double strand. Strands can also be concatenated. If two strands are brought into juxtaposition because they have both joined to part of a third, complementary strand, then under the action of a ligase enzyme a bond forms between the first two strands so that they become a single longer strand. An example is found in figure 3. This bond persists even if the strand then separates from its complement.
copying: Many copies of a given strand of DNA can be created by polymerase chain reaction (PCR). The strand to be amplified is defined by two primers, which are segments of DNA. The primers are the start and the complement of the end of the sequence of interest. For example, say O_0 and O_6 are segments of DNA, and the problem is to create copies of every sequence of DNA in the test tube that contains O_0 followed by an unknown sequence of bases followed by O_6. Then the primers for this PCR are O_0 and $\overline{O}_6$, where the bar indicates the Watson-Crick complement. The amplification works roughly in the following way. Many copies of O_0 and $\overline{O}_6$ are added into the test tube. The mixture is heated, which causes the strands of DNA to separate. As it cools, the primers attach themselves where they can, that is, one to the beginning of the edges beginning with 0, the other to the end of the edges ending in 6. The primer then forms the start of a new chain that grows out from it, forwards from O_0 and backwards from $\overline{O}_6$, as shown in figure 2. This process is repeated, and the number of strands consisting of the segments of interest doubles each time. A few hours will suffice to have ample quantities of these strands, though in practise, the duration used is often "one night".
sorting: DNA strands can be sorted by length. This is achieved by gel electrophoresis, a process which involves separating the strands by their electrophoretic mobility, which is a function of the number of base pairs.
extraction: Strands containing a specific segment of DNA can be extracted from the test tube. Extraction is performed by separating the strands, and then using magnetic beads with a complement of the segment to be extracted attached to each bead. Only the DNA containing that segment will attach itself to the bead and be retained.
detection: The existence of DNA in a test tube is determined using PCR.

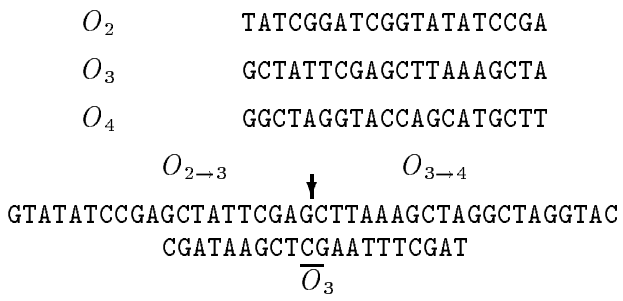

FIGURE 3. Encoding a graph in DNA. A path along the edges $2 \to 3$ and $3 \to 4$ is formed when each edge becomes attached to half of the complementary vertex $\overline{O}_3$ and a ligation reaction occurs.

3. The Algorithm

Adleman uses a naive brute-force algorithm. Given a directed graph on n vertices, where the path is to start at vertex v_{in} and finish at vertex v_{out}, the following steps will result in a solution if one exists.

(1) Input the graph (creation).
(2) Generate many many random paths through the graph (joining and copying).
(3) Keep only the paths that start at v_{in} and end at v_{out} (copying).
(4) Keep only those paths that enter exactly n vertices (sorting).
(5) Keep only those paths that enter all of the vertices at least once (extraction).
(6) If no paths remain, say "no", otherwise say "yes", and the remaining paths are solutions (detection, copying and sorting).

An ordinary computer would not normally attempt such an algorithm, due to the enormous numbers of cases to consider. Using DNA, these cases can be treated in parallel.

The algorithm is performed on the graph in figure 1, and the goal is to construct a path from 0 to 6 while passing through all the vertices exactly once. For convenience, the labels were chosen so that the solution is $0 \to 1 \to 2 \to 3 \to 4 \to 5 \to 6$, but of course this does not affect the difficulty of the problem. Obviously, in the case of this graph, the answer can be found by inspection. However, this experiment demonstrates the feasibility of the technique.

Each vertex of the graph is represented by a random 20 base sequence O_i. Using 20 bases means the chances of that sequence appearing elsewhere in the DNA is miniscule. The Watson-Crick complementary sequence is denoted $\overline{O}_i$. Each edge $i \to j$ in the graph is created by creating the 20-letter molecule that starts with the last ten bases of O_i and ends with the first 10 bases of O_j. This sequence is denoted $O_{i\to j}$.

Mixing together all the the edges $O_{i\to j}$ with $\overline{O}_i$ for $i = 1, \ldots, 5$ allows concatenations to occur that forms random paths through the graphs, as required by step 2. For instance, $O_{2\to 3}$ and $O_{3\to 4}$ are edges in the graph. These edges can be concatenated together by using $\overline{O}_3$ as a splint. This new molecule represents a path from $2 \to 3 \to 4$. See figure 3. Given the number of reactions and the number of molecules formed, it is statistically extremely likely that the Hamiltonian path will be created if it exists.

Step 3 is to keep only those random paths that start at 0 and end at 6. By "keep", it is meant that these strands are copied so many times that the presence of other strands becomes statistically insignificant in comparison.

Step 4 is achieved by sorting the strands by length, and keeping those that are 140-base pairs long, and thus enter exactly 7 vertices.

In order to keep only the strands that enter each vertex at least once (step 5), first the strands containing O_1 are extracted. Next, those strands containing O_2 are extracted, then O_3 etc.

Then, for step 6, the presence or absence of DNA in the test tube is detected. If absent, there is no Hamiltonian path for this graph. If present, amplification by PCR is performed, first using primers O_0 and O_1 to create copies of the path between 0 and 1, then using O_0 and O_2 to create copies of the path between 0 and 2, etc. Then the lengths are determined. In this case, the length of the molecule starting at O_0 and ending at O_1 is 40, indicating that the vertex 1 comes directly after vertex 0 in the solution. Multiple solutions would show up as multiple lengths for each segment, and by determining the various second vertices from the lengths, these solutions could be separated. A picture in the article [1] shows the result of this step. The solution found is indeed $0 \to 1 \to 2 \to 3 \to 4 \to 5 \to 6$.

4. Extensions

Richard Lipton has proposed an algorithm consisting of DNA experiments to solve the satisfaction problem (SAT) [2]. Given a boolean formula involving n variables the problem is to assign values to the variables such that the expression evaluates to true. A graph representation of the problem is used, where each path through the graph gives an assignment to the variables. The paths are generated using the same techniques as before. Briefly, the first step is to extract the DNA that makes the first clause true, then extract the DNA that makes the second clause true, etc. The paths through the graph can also be interpreted as n-bit binary numbers, where x_i is true means the ith bit is a 1, false means 0. Thus any binary number can be stored as a DNA molecule.

5. Advantages of DNA Methods

Both these problems are NP-complete, and so there is no polynomial time algorithm to solve them on traditional computers, and little hope of finding one. The incredible parallelism of the DNA-techniques means that exhaustive searches through all the possibilities can be done relatively rapidly, and may be able to provide a solution to problems that traditional computers cannot solve. For instance, it is estimated that DNA methods may be able to solve the Hamiltonian path problem on graphs of up to 70 edges.

There are also less obvious advantages. DNA techniques are energy efficient. Approximately 2×10^{19} ligation operations per 1 joule of energy can be performed, versus 10^9 operations per joule for existing supercomputers. It is estimated that the energy cost of the other operations is similarly tiny in comparison. Finally, as a storage medium, nothing else comes close. Information can be stored in approximately 1 bit per cubic nanometer. In contrast, videotapes store information at 1 bit per 10^{12} cubic nanometers.

More investigation is needed to determine which kinds of problems can be handled by these methods. The probability and effect of errors during the operations needs to be studied, as well as the possibility of creating new basic operations. It is possible but not yet known if a DNA molecule could encode a Turing machine, where the actions of certain enzymes would perform the operations of the machine.

Bibliography

[1] Adelman (Leonard M.). – Molecular computation of solutions to combinatorial problems. *Science*, vol. 266, 1994, pp. 1021–1024.

[2] Lipton (Richard J.). – DNA solution of hard computational problems. *Science*, vol. 268, 1995, pp. 542–545.