Computer in a TestTube

DNA computing

Hendrik Jan Hoogeboom
Computer Science Leiden
13 december 2010
natural computation

- genetic algorithms
- neural networks

DNA computing

bio-informatics

“the mad scientist at work”

http://www.usc.edu/dept/molecular-science/fm-adleman.htm
“In other words, one could program a Turing machine to produce Watson-Crick complementary strings, factor numbers, play chess and so on. This realization caused me to sit up in bed and remark to my wife, Lori, ‘Jeez, these things could compute.’ I did not sleep the rest of the night, trying to figure out a way to get DNA to solve problems.”

Leonard M. Adleman - Computing with DNA
Scientific American August 1998
Physicists plunder life's tool chest

If we look inside the cell, we see extraordinary machines that we couldn't make ourselves, says Len Adleman. “It's a great tool chest - and we want to see what can we build with it.”

Adleman created the first computer to use DNA to solve a problem. He was struck by the parallels between DNA, with its long ribbon of information, and the theoretical computer known as the Turing Machine.

Nature News Service April 2003
Adleman tackled the famous \textit{‘travelling salesman’} problem - finding the shortest route between cities. Such problems rapidly become mind-boggling. The only way is to examine every possible option. With many cities, this number is astronomical.

DNA excels at getting an astronomical amount of data into a tiny space. “One gram of DNA can store as much information as a trillion compact discs,” says Adleman. Myriad DNA molecules can examine every possible route at once, rather than one at a time, as in a conventional computer.
DNA ... the tool chest

problem complexity ... P & NP

Hamilton Path Problem

Adleman's algorithm

comments

theory ... Turing machine

recent work + future
DNA ... the tool chest
problem complexity ... P & NP
Hamilton Path Problem
Adleman’s algorithm
comments
theory ... Turing machine
recent work + future
Base pairs
Watson & Crick
[ & Rosalind Franklin ]

A=T
adenine - thymine
C=G
guanine - cytosine
single – double strand

5' to 3' double strand

low temp
denaturing

annealing

high temp
single strands
complementarity
restriction enzymes

BamHI

G G A T T C C
C C T A G G

sticky ends
subsequence selection

magnetic beads
DNA gel electrophoresis
multiplication / amplification

primer

polymerase

PCR – polymerase chain reaction
DNA ... the tool chest

problem complexity ... P & NP

Hamilton Path Problem

Adleman’s algorithm

comments

theory ... Turing machine

recent work + future
## Complexity

<table>
<thead>
<tr>
<th>n=10</th>
<th>30</th>
<th>50</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>$10^{-5}$ s</td>
<td>$3 \times 10^{-5}$ s</td>
<td>$5 \times 10^{-5}$ s</td>
</tr>
<tr>
<td>n$^2$</td>
<td>$10^{-4}$ s</td>
<td>$9 \times 10^{-4}$ s</td>
<td>$2 \times 10^{-3}$ s</td>
</tr>
<tr>
<td>n$^5$</td>
<td>$10^{-1}$ s</td>
<td>24 s</td>
<td>1.7 m</td>
</tr>
<tr>
<td>$2^n$</td>
<td>$10^{-3}$ s</td>
<td>18 m</td>
<td>13 d</td>
</tr>
<tr>
<td>$3^n$</td>
<td>$6 \times 10^{-2}$ s</td>
<td>6.5 y</td>
<td>3855 c</td>
</tr>
</tbody>
</table>

### Polynomial vs. Exponential

<table>
<thead>
<tr>
<th>now</th>
<th>100x</th>
<th>1000x</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>N</td>
<td>100N</td>
</tr>
<tr>
<td>n$^2$</td>
<td>N</td>
<td>10N</td>
</tr>
<tr>
<td>n$^5$</td>
<td>N</td>
<td>2.5N</td>
</tr>
<tr>
<td>$2^n$</td>
<td>N</td>
<td>N+6.6</td>
</tr>
<tr>
<td>$3^n$</td>
<td>N</td>
<td>N+4.2</td>
</tr>
</tbody>
</table>
The general idea

custom made single strands of DNA (many copies)

is there a double strand with my desired properties?

properties:
• length,
• subsequence.

if we can do this, then we can solve certain problems (efficiently)!
given: directed graph (points & connections)
question: is there a path that visits each point exactly once?
HPP: Hamilton Path Problem

solution

‗travelling salesman‘

given: directed graph (points & connections)
question: is there a path that visits each point exactly once?
HPP: Hamilton Path Problem

no solution?

exponential time:
  try all possibilities

representative class ‘NP complete’

heuristics
P
polynomial algorithm to **find** a solution

NP
polynomial algorithm to **verify** a solution

**NP-complete**

millenium prize problem  P=NP

www.claymath.org/Millennium_Prize_Problems/
- DNA ... the tool chest
- problem complexity ... P & NP
- Hamilton Path Problem
- Adleman’s algorithm
- comments
- theory ... Turing machine
- recent work + future
Adleman’s algorithm

1. generate ‘all’ paths keep only paths
2. ... from $v_{in}$ to $v_{out}$
3. ... that enter $n$ vertices
4. ... that enter all vertices
5. if any path remains OK

‗massive parallelism‘
building blocks
Adleman’s algorithm

0. coding the graph
1. generate ‘all’ paths
   keep only paths
2. ... from \(v_{in}\) to \(v_{out}\)
3. ... that enter \(n\) vertices
4. ... that enter all vertices
   any path remains OK
Adleman’s algorithm

0. coding the graph
1. generate ‘all’ paths
   keep only paths
2. ... from \( v_{in} \) to \( v_{out} \)
3. ... that enter \( n \) vertices
   that enter all vertices

ACGG GTGG ATCC TAGT TTGC AACT ATCC TAGT

CACC TAGG ATCA AACG TTGA TAGG
0. coding the graph
1. generate ‘all’ paths
   keep only paths
2. … from $v_{in}$ to $v_{out}$
3. … that enter $n$ vertices
4. … that enter all vertices
5. if any path remains OK

- PCR with $v_{in}$ and $v_{out}$ primers
- gel: separate on length, amplify & purify
- magnetic beads: select strands
- PCR amplification & gel
DNA ... the tool chest

problem complexity ... P & NP

Hamilton Path Problem

Adleman’s algorithm

comments

theory ... Turing machine

recent work + future
“clear that the methods could be scaled up to ... larger graphs”

- bath tub of DNA?
- suitable algorithms

approximately 7 days of lab work

- automation
- alternative molecular algorithms

possibility of errors

- pseudopaths: accidental ligation
- PCR, separation procedures
- hairpin loops
- stability when scaled
comments ...

- "power of this method of computation"
  - $10^{14}$ operations $10^{20}$ plausible
  - exceed supercomputers by thousandfold
  
  :)

- "not clear whether ... used to solve real computational problems"
  
  . multiplying 100 digit numbers

- potential: massively parallel searches
- DNA ... the tool chest
- problem complexity ... P & NP
  Hamilton Path Problem
- Adleman’s algorithm
- comments
- theory ... Turing machine
- recent work + future
1. mark a
2. move to b’s
   mark b
3. move to c’s
   mark c
4. if another c
5. then back to a’s
goto 1.
else back to a’s
6. check marks
   stop
‘universal’ Turing machine

- cut states with restriction enzyme
- mix ‘instructions’ with ‘tape’
- ‘activate’ instructions (cut protected end)
- ligate to form circles
- cut old symbol
- recircularize

Rothemund
FokI
circular DNA
DNA ... the tool chest
problem complexity ... P & NP
Hamilton Path Problem
Adleman's algorithm
comments
theory ... Turing machine
recent work + future
first “medium-scale integrated molecular circuit”, integrating 128 deoxyribozyme-based logic gates, 32 input DNA molecules, and 8 two-channel fluorescent outputs across 8 wells.
\( o_2 = (i_6 \land i_7 \land \neg i_2) \lor (i_7 \land i_9 \land \neg i_1) \lor (i_8 \land i_9 \land \neg i_1) \)
b

\[
\begin{align*}
o_1 &= \frac{i_4}{\text{edge (24)}} \\
o_2 &= \frac{(i_6 \land i_7 \land \neg i_2)}{\text{edge (23)}} \\
o_3 &= \frac{(i_1 \land i_6) \lor (i_4 \land i_9)}{\text{edge (22)}} \\
o_4 &= \frac{i_1}{\text{edge (21)}} \\
o_5 &= \frac{i_1}{\text{edge (20)}} \\
o_6 &= \frac{(i_1 \land i_2 \land \neg i_6) \lor (i_1 \land i_3 \land \neg i_6) \lor (i_1 \land i_7 \land \neg i_6) \lor (i_1 \land i_8 \land \neg i_6) \lor (i_1 \land i_9 \land \neg i_6)}{\text{edge (1)}} \\
o_7 &= \frac{(i_2 \land i_6 \land \neg i_7) \lor (i_6 \land i_8 \land \neg i_7) \lor (i_6 \land i_9 \land \neg i_7) \lor (i_6 \land i_2 \land \neg i_1)}{\text{edge (6)}} \\
o_8 &= \frac{i_9 \land i_7 \land \neg i_4}{\text{edge (9)}} \\
o_9 &= \frac{(i_7 \land i_8 \land \neg i_4) \lor (i_4 \land i_2 \land \neg i_9) \lor (i_4 \land i_3 \land \neg i_9) \lor (i_4 \land i_6 \land \neg i_9) \lor (i_4 \land i_7 \land \neg i_9) \lor (i_4 \land i_8 \land \neg i_9)}{\text{edge (10)}} \\
o_{10} &= \frac{(i_7 \land i_8 \land \neg i_4) \lor (i_4 \land i_2 \land \neg i_9) \lor (i_4 \land i_3 \land \neg i_9) \lor (i_4 \land i_6 \land \neg i_9) \lor (i_4 \land i_7 \land \neg i_9) \lor (i_4 \land i_8 \land \neg i_9)}{\text{edge (11)}} \\
o_{11} &= \frac{(i_7 \land i_8 \land \neg i_4) \lor (i_4 \land i_2 \land \neg i_9) \lor (i_4 \land i_3 \land \neg i_9) \lor (i_4 \land i_6 \land \neg i_9) \lor (i_4 \land i_7 \land \neg i_9) \lor (i_4 \land i_8 \land \neg i_9)}{\text{edge (12)}} \\
o_{12} &= \frac{(i_7 \land i_8 \land \neg i_4) \lor (i_4 \land i_2 \land \neg i_9) \lor (i_4 \land i_3 \land \neg i_9) \lor (i_4 \land i_6 \land \neg i_9) \lor (i_4 \land i_7 \land \neg i_9) \lor (i_4 \land i_8 \land \neg i_9)}{\text{edge (13)}} \\
o_{13} &= \frac{(i_7 \land i_8 \land \neg i_4) \lor (i_4 \land i_2 \land \neg i_9) \lor (i_4 \land i_3 \land \neg i_9) \lor (i_4 \land i_6 \land \neg i_9) \lor (i_4 \land i_7 \land \neg i_9) \lor (i_4 \land i_8 \land \neg i_9)}{\text{edge (14)}} \\
o_{14} &= \frac{(i_7 \land i_8 \land \neg i_4) \lor (i_4 \land i_2 \land \neg i_9) \lor (i_4 \land i_3 \land \neg i_9) \lor (i_4 \land i_6 \land \neg i_9) \lor (i_4 \land i_7 \land \neg i_9) \lor (i_4 \land i_8 \land \neg i_9)}{\text{edge (15)}}
\end{align*}
\]
“There are many practical hurdles. Even with the best techniques of today, DNA still lags behind silicon computers,” says Ehud Shapiro. Instead, he advocates creating DNA devices that can do things, and go to places, that silicon can't - such as inside our cells, to make and control drugs.

... 

Ultimately, Seeman hopes to build DNA scaffolding for electrical circuits, or for other molecular machines.

... 

Yurke is focusing on DNA machines with moving parts. In 2000, he and his colleagues devised a set of DNA tweezers

....
Cross-fertilization between evolutionary computation and DNA-based computing  T. Back; J.N. Kok; G. Rozenberg  Proceedings 1999 Evolutionary Computation.
Researchers make significant advances in molecular computing, University of Kent, 10-Dec-2009

Dr Chu explained: 'Our research demonstrates that the speed of bio-molecular computers is fundamentally limited by their metabolic rate or their ability to process energy. One of our main findings is that a molecular computer has to balance a trade-off between the speed with which a computation is performed and the accuracy of the result. However, a molecular computer can increase both the speed and reliability of a computation by increasing the energy it invests in the computation. With molecular computers this energy may be derived from food sources.'
... they tried the system with simple "if... then..." propositions. One of these went as follows: "All men are mortal. Socrates is a man. Therefore, Socrates is mortal."

The answer was encoded in a flash of green light. Some of the DNA strands were equipped with a naturally glowing fluorescent molecule bound to a second molecule which keeps the light covered.

The system can take in facts and rules as a computer file of simple text. The robotic "compiler" can then turn those facts and rules into the DNA starting products of a logical query.

In other words, computers that go to work inside a cell.
"This soup of DNA and enzymes implements a well known mathematical model of computation known as finite automaton," he explained. "This finite automaton knows how to do very simple computation such as recognising whether a list of zeros and ones has an even number of ones."

In the case of his 2004 computer this method of computation was used to analyze ratios of specific molecules related to prostate cancer and a specific type of lung cancer.

The "computer" consisted of a chain of three segments of DNA and an enzyme which could cut the strands.
"The biocomputer would sense biomarkers and immediately react by releasing counter-agents for the disease," says Itamar Willner, who led the work.

The new logic gates are formed from short strands of DNA and their complementary strands, .... Two strands act as the input: each represents a 1 when present or a 0 when absent. ... Take the "exclusive OR" or XOR logic gate. It produces an output when either of the two inputs is present but not when both are present or both are absent.

Willner and his team added molecules to both the complementary strands that caused them to fluoresce when each was present in isolation, representing a logical 1 as the output. But when both were present, the complementary strands combined and quenched the fluorescence, representing a 0 output.
self assembly

ACGG GTGG ATCC TAGT TTGC AACT ATCC TAGT

CACC TAGG ATCA AACG TTGA TAGG
Sierpinski triangle

Pascal's triangle

Sierpinski triangle

⊕ XOR

even / odd

⊕

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

|   | 1 | 1 | 1 |
|---|---|---|
| 1 | 1 | 2 | 1 |
| 1 | 3 | 3 | 1 |
| 1 | 4 | 6 | 4 | 1 |
| 1 | 5 | 10 | 10 | 5 | 1 |
| 1 | 6 | 15 | 20 | 15 | 6 | 1 |
| 1 | 7 | 21 | 35 | 35 | 21 | 7 | 1 |
| 1 | 8 | 28 | 56 | 70 | 56 | 28 | 8 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
Sierpinski triangle

⊕ XOR
even / odd

‘fractal’
Algorithmic Self-Assembly of DNA Sierpinski Triangles


http://dx.doi.org/10.1371/journal.pbio.0020424
self assembly: DNA origami

Folding DNA to create nanoscale shapes and patterns
Self Assembly: DNA origami

Self-assembly of DNA into nanoscale three-dimensional shapes
S.M. Douglas, H. Dietz, T. Liedl, B. Hogberg, F. Graf, W.M. Shih,
Nature 459, 414-418 (21 May 2009)
DNA can be used for applications it was not "intended" for computing. A very interesting proof of concept find niche.