# Natural Computing

Lecture 15

Michael Herrmann
mherrman@inf.ed.ac.uk
phone: 0131 6 517177
Informatics Forum 1.42

8/11/2011

# DNA Computing

# Natural computing

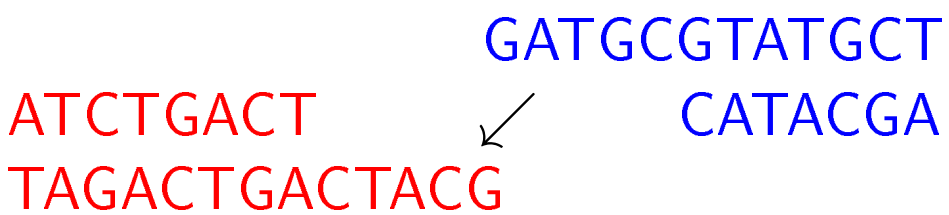
- Beyond bioinspired algorithms
- Use the complex dynamics of a biophysical system
  - Parallel by nature
  - Non-deterministic
  - capable of information processing
- Encoding problems must be solved in applications to practical problems
- Computation works well in many problems but may be ineffective or inefficient on some problems
- Potential for engineering

*We turn to biology not just as a metaphor, but as an actual implementation technology ...*

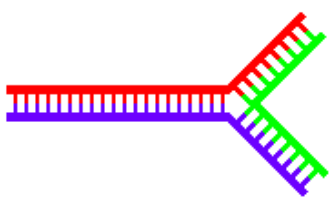
Harold Abelson et al., MIT, 2000

# DNA

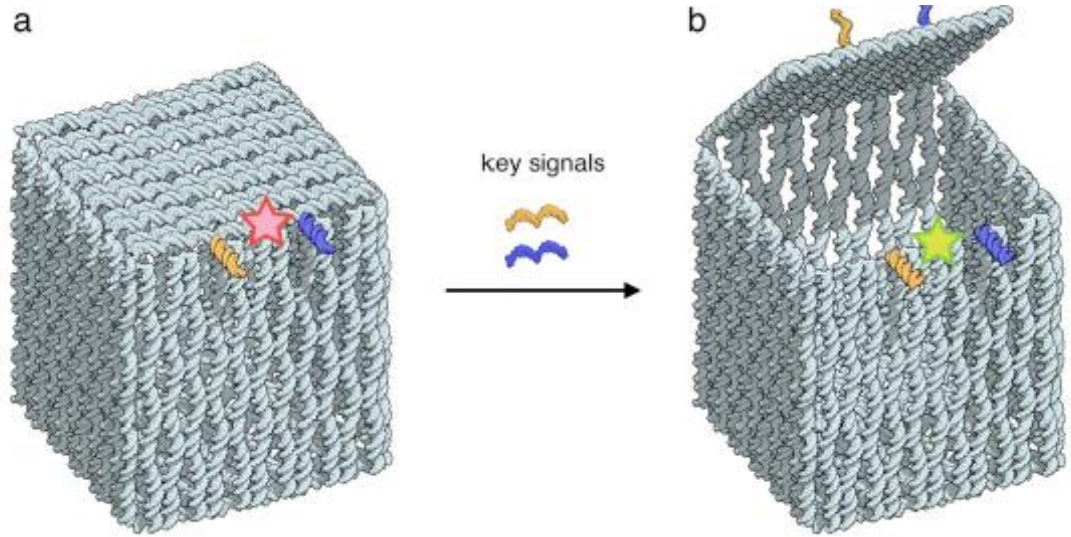
- Two chains of polymers, the nucleotides: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T) or Uracil (U)
- Backbones made of sugars and phosphate groups joined by ester bonds
- **Exploit DNA as programmable matter to do computations as determined by material properties (i.e. by nucleotide sequence)**
- Overhangs can be used to attach a strand to a specific end of another string

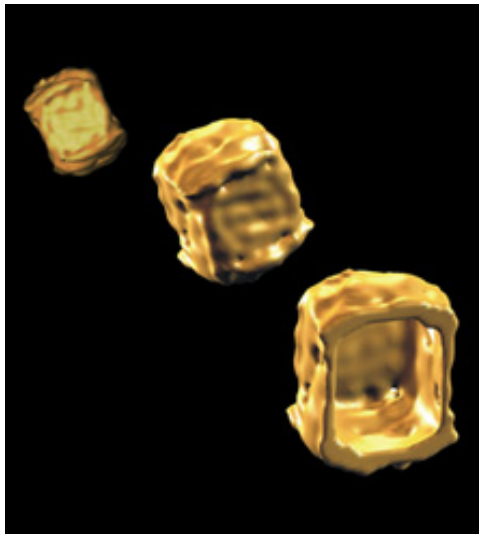



- Branches: if the strings do not match or if branching is triggered by a third molecule



# Self-Assembly of a Nano-Size DNA Box



Purposes for the box:

- calculator or logic gate
- controlled release, for example of drugs, in response to external stimuli
- sensor - where the thing you are sensing causes the box to open or close and give a readout

E S Andersen ... and J Kjems, Nature 2009 DOI:10.1038/nature07971

# DNA: Computing by Molecules

## DNA computing in numbers

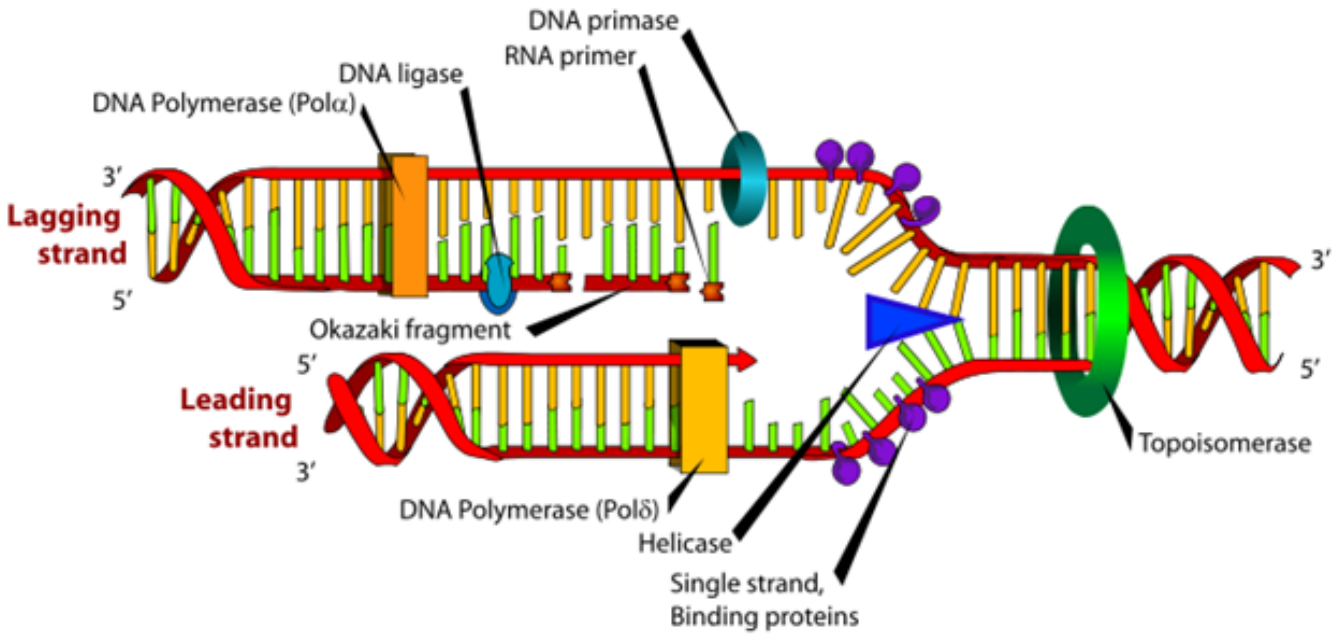
- Energy consumption $2 * 10^{19}$ operations/Joule (a billion times better than classical computers)
- 5 grams of DNA contain $10^{21}$ bases (Zetta Bytes) [the size of the Internet is still measured in Exa Bytes ($10^{18}$)]
- Each DNA strand represents a processor (300 trillion)
- Relatively slow speed: 500-5000 base pairs a second
- Still the fastest and most economical among the existing computing mechanisms

# DNA Computing: History

- 7-point Hamiltonian path problem (L. Adleman, 1994)
- Programmable molecular computing machine (E. Shapiro, 2002)
- DNA computer (E. Shapiro, 2004) capable of diagnosing cancerous activity within a cell and releasing an anti-cancer drug upon diagnosis.
- "DNA origami" use a raster to impose precision and use "genetic operators" (here in a different sense: cut, insert, splice etc.) to fix the structure
- In theory, DNA computers can emulate Turing machines

# DNA Computing

- Parallelism by (typically) trillions of "processors"
- Complementarity makes DNA unique ⇒ error correction (improving upon 1 transcription error per every 100,000 bases)
- Basic suite of operations: AND, OR, NOT in a CPU must be represented by cutting, linking, pasting, amplifying or repairing DNA



# Basic operations

Basic operations can be carried out on DNA sequences by commercial available enzymes:

- Cutting. An enzyme restriction endonuclease permits to recognize a small portion of the DNA. Any double helix that contains it can be cut in that exact place.
- Linking. An enzyme DNA ligase allows to join the end of a DNA sequence with the beginning of another one.
- Replication. An enzyme DNA polymerase makes possible the DNA replication.
- Destruction. With the enzyme exonuclease, it is possible to eliminate certain DNA subsequences.

C.A. Alonso Sanches, N.Y. Soma / Applied Mathematics and Computation 215 (2009) 2055–2062

# Seminal Example: Adleman's solution of the Hamiltonian Directed Path Problem (HDPP)

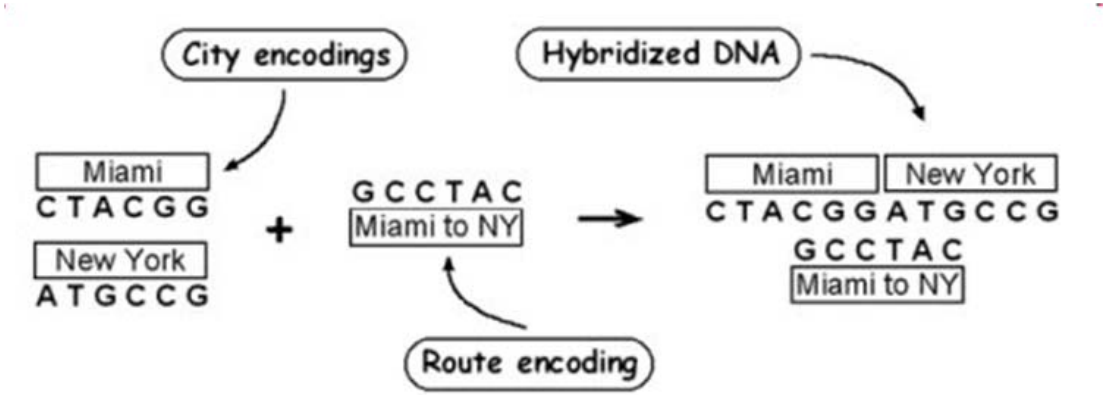
Algorithm to be implemented in the DNA computer:

1. Generate random paths
2. From all paths created in step 1, keep only those that start at $s$ and end at $t$.
3. From all remaining paths, keep only those that visit exactly $n$ vertexes.
4. From all remaining paths, keep only those that visit each vertex at least once.
5. If any path remains, return "yes" otherwise, return "no"

# Example with 5 cities: 1. Encoding and path construction

Encoding of nodes and edges using artificial gene synthesis

| city | code |
|---|---|
| Los Angeles | GCTACG |
| Chicago | CTAGTA |
| Dallas | TCGTAC |
| Miami | CTACGG |
| New York | ATGCCG |



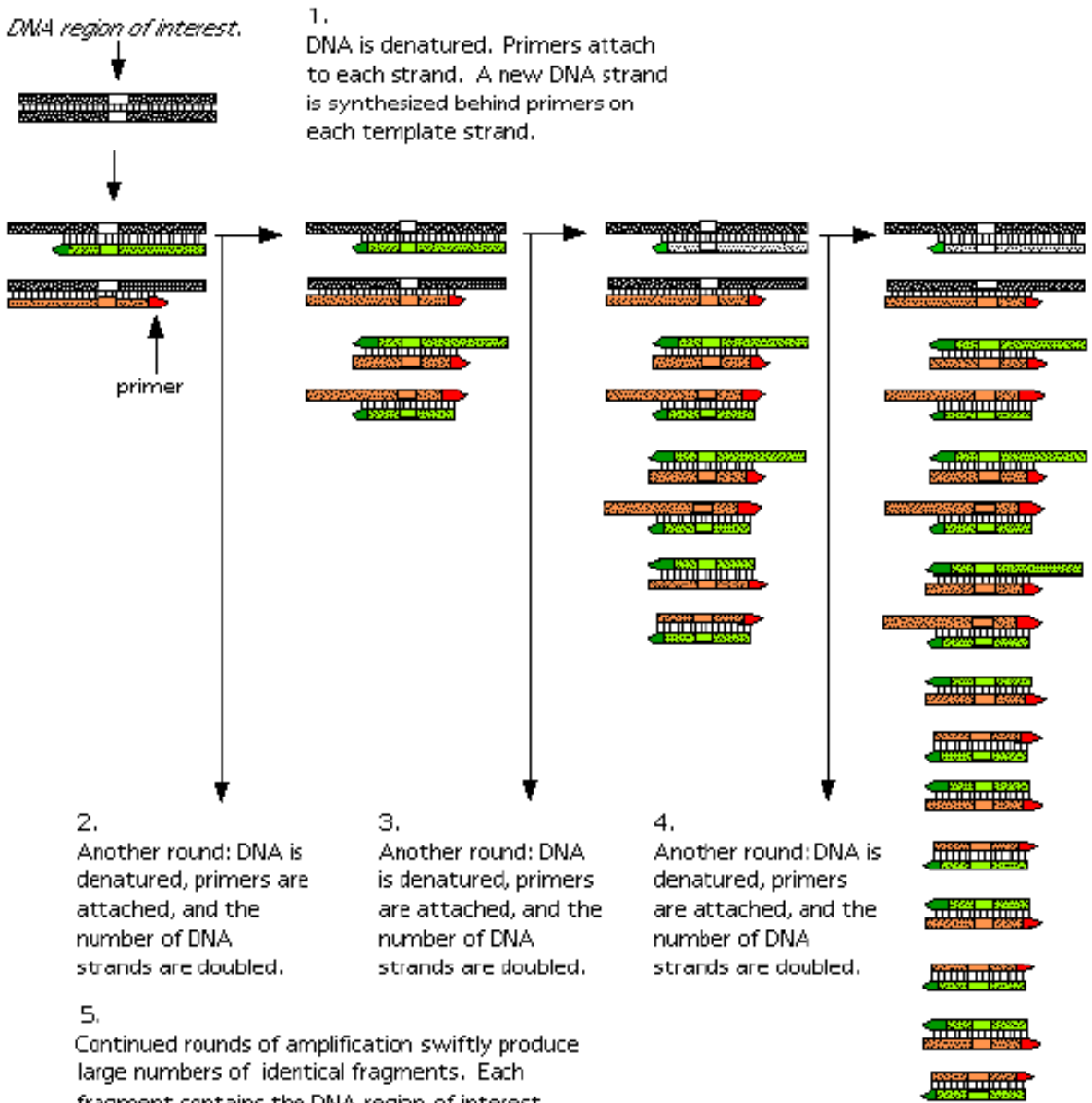

L.A -> Chicago -> Dallas -> Miami -> New York would simply be GCTACGCTAGTATCGTACCTACGGATGCCG

# Polymerase Chain Reaction (PCR)

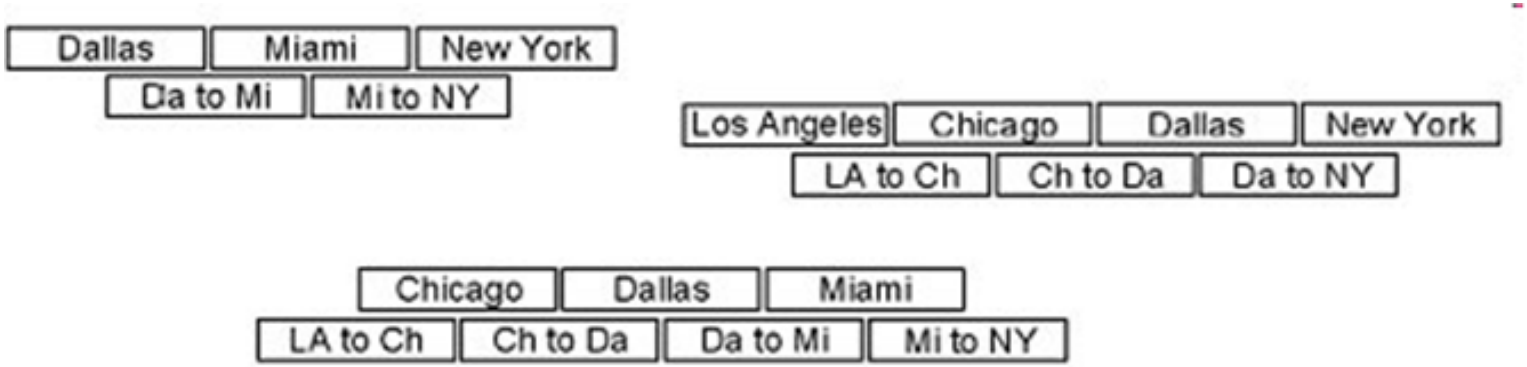
PCR: One way to amplify DNA. (Kary Mullis, 1985)
PCR alternates between two phases:

- separate DNA into single strands using heat
- convert into double strands using primer and polymerase reaction

PCR rapidly amplifies a single DNA molecule into billions of molecules



# Step 2: PCR selecting paths with correct start and end

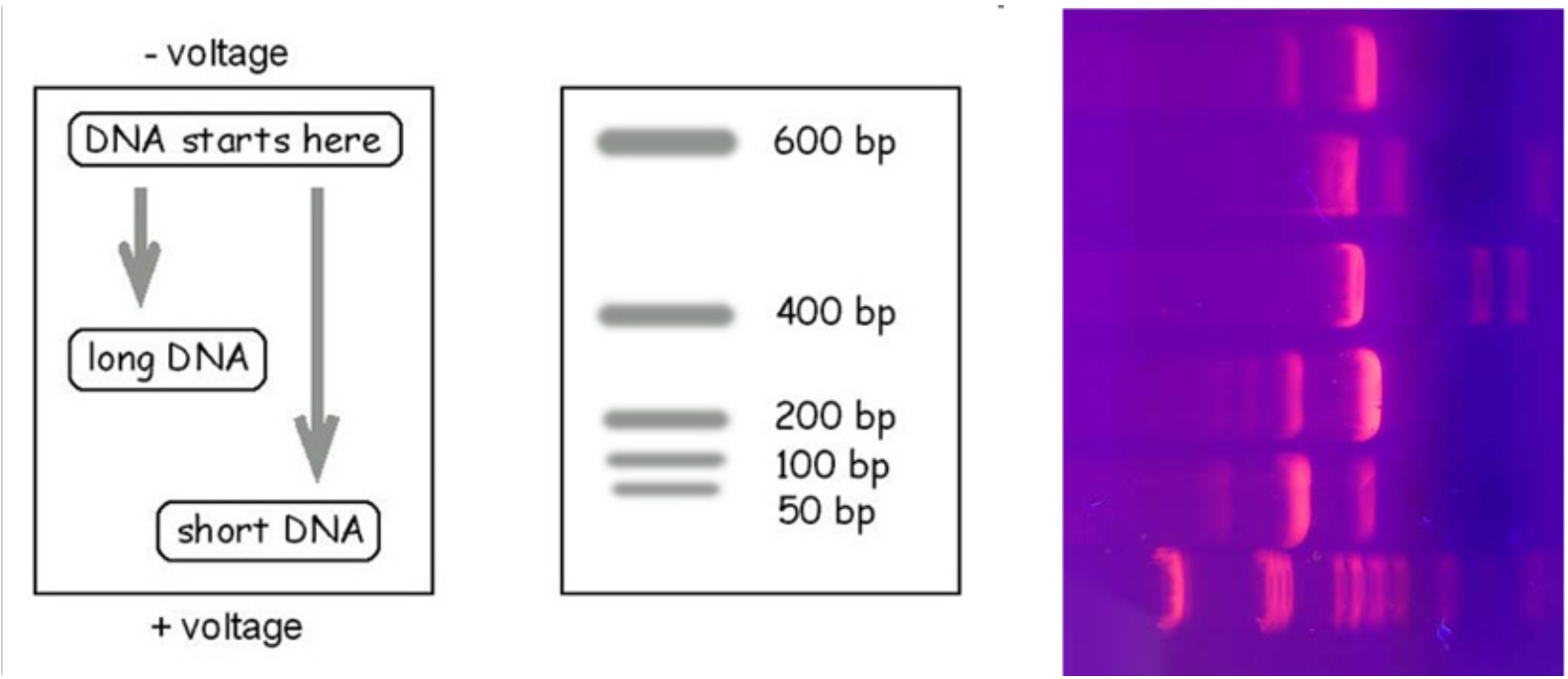



(a billion copies of each)

- Starting a PCR with the starting city as a primer (i.e. the primer is the complement of the city code)
- and a primer for termination corresponding to the goal city.
- produces lots of strands with the correct start and end (but with variable lengths and with possible repetitions)

# Step 3: Gel electrophoresis for measuring lengths

- Used to measure the length of a DNA molecule.
- Smaller DNA molecules are travel faster in an electric field (for same charge)

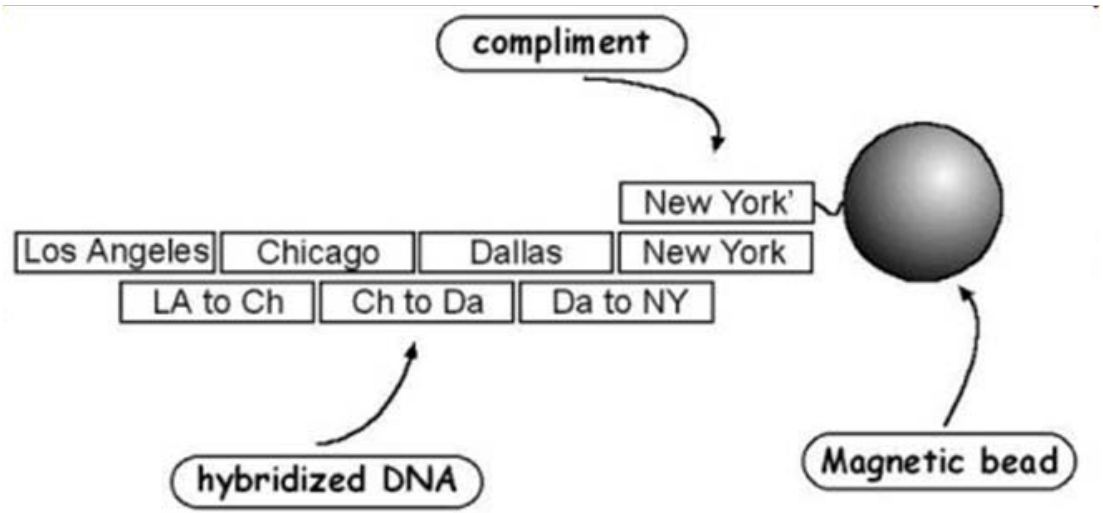



- Keep only those paths that visit exactly $n$ vertexes.
- Isolate the DNA if 30 base pairs long (5 cities $\times$ 6 base pairs).

http://arstechnica.com/reviews/2q00/dnahttp://de.wikipedia.org/wiki/Gelelektrophorese

# Step 4: Affinity purification for selecting admissible paths

- Select itineraries that have a complete set of cities
- Sequentially affinity-purify fives times, using a different city complement for each run.




- We are left with itineraries that start in LA, visit each city once, and end in NY.

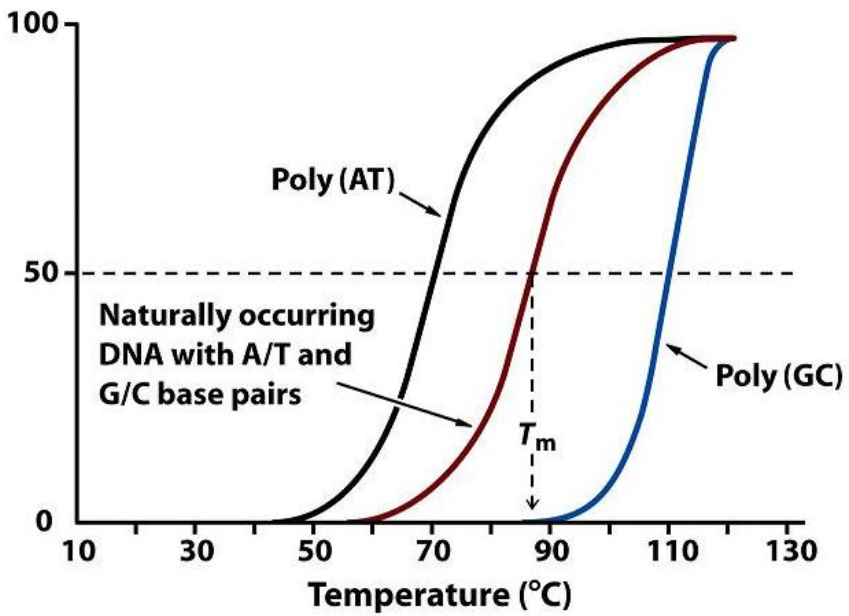
http://arstechnica.com/reviews/2q00/dna

# Step 5: Reading out the answer

- Result can be obtained by sequencing the DNA strands
- More effectively by using **graduated PCR**:
  - A series of PCR amplifications using the different primer for each city in succession.
  - Measuring the various lengths of DNA for each PCR product reveals the final sequence of cities.
  - For example, starting with LA gives 30 base pairs. If LA and Dallas primers give 24 base pairs, Dallas is the fourth city in the itinerary.

# Extension to TSP

Use differences in melting temperature (depends on the fraction of AT vs. GC) to encode real numbers. Note that the representation is not neutral to the selection anymore.




**Step 1**: Generation of answer pool [Hybridization & Ligation]

**Step 2**: Selection of paths satisfying the conditions of TSP [PCR with primer $v_{out}$ and $v_{in}$, and affinity-separation]

**Step 3**: More amplification of the more economical paths [DTG-PCR]

**Step 4**: Separation of the most economical path among the candidate paths [TGGE]

**Step 5**: Readout of the final path [Cloning and sequencing]

Ji Youn Lee (2004) Solving traveling salesman problems with DNA molecules encoding numerical values. BioSystems 78 (2004) 39–47

http://sandwalk.blogspot.com/2007/12/dna-denaturation-and-renaturation-and.html

# Caveats

- Adleman solved a seven city problem. What about more cities?
- The complexity of the problem still increases exponentially.
- For Adleman's method the amount of DNA scales exponentially: to solve a 200 city HP problem would take an amount of DNA that weighed more than the earth
- Each step contains statistical errors which grow if the strands become longer, more operations are being performed and less DNA is used per potential solution
- Cost: For a computation with n = 100, one would need to purchase at least 300 strands at a cost of $3,000 (costs are dropping)

Shrenik Shahy: DNA Computation and Algorithm Design. Harvard University 2009 Cambridge, MA 02138 (student article)

# Solved/solvable problems

- Ravinderjit S. Braich, Nickolas Chelyapov, Cliff Johnson, Paul W. K. Rothemund, Leonard Adleman: Solution of a 20-Variable 3-SAT Problem on a DNA Computer. Science 296, 19 April 2002
- S. Paul. G. Sahoo: Procedure for Multiplication based on DNA Computing2009 Int. Conf. on Adv. in Computing, Control, and Telecommunication Technologies
- C. A. A. Sanches , N. Y. Soma: A polynomial-time DNA computing solution for the Bin-Packing Problem. Applied Mathematics and Computation 215 (2009) 2055–2062
- t.b.c.

# Adleman–Lipton model

## From chemistry to language

A "test tube" language describing DNA computation on a multi-set of finite strings over the alphabet $\{A; C; G; T\}$.

- Amplify. Given a test tube $T$; *amplify*$(T; T_1; T_2)$ produces two new test tubes $T_1$ and $T_2$ that are identical copies of $T$, and this latter one becomes empty.
- Merge. Given two test tubes $T_1$ and $T_2$, this operation generates a new tube with the content of both, that is, it is equivalent to decant the contents of the $T_1$ and $T_2$ into a third one without modifying no molecule.
- Append. Given a test tube $T$ and a sequence $S$; append$(T; S)$ affixes $S$ at the end of each sequence in $T$.
- Extract. Given a test tube $T$ and a sequence $S$, generates two tubes: $+(T; S)$ with all the sequences in $T$ that had $S$ as a subsequence, and $-(T; S)$ with the remaining sequences of $T$.
- Detect. Given a tube $T$, this operation returns the logic value *yes* if there is at least a DNA molecule in it, and *no* otherwise. Discard. Given a test tube $T$, this operation simply discards it.

C. A. A. Sanches , N. Y. Soma: A polynomial-time DNA computing solution for the Bin-Packing Problem. Applied Mathematics and Computation 215 (2009) 2055–2062

# Seminal Example: Adleman's solution of the Hamiltonian Directed Path Problem (HDPP)

Algorithm to be implemented in the DNA computer:

1. Generate random paths (Append)
2. From all paths created in step 1, keep only those that start at $s$ and end at $t$. (Extract)
3. From all remaining paths, keep only those that visit exactly $n$ vertexes. (Extract)
4. From all remaining paths, keep only those that visit each vertex at least once. (Extract)
5. If any path remains, return "yes" otherwise, return "no" (Detect)

# Sticker model enhances the Adleman–Lipton model

An enhanced language on a multi-set of finite strings over$\{A; C; G; T\}$.

- Combine: Given two test tubes $T_1$ and $T_2$ filled with DNA's sequences, this operation gives a third test tube $T_3$ with a union of the first two. The operation corresponds to a merge to the original model.
  Notation : $T_3 \leftarrow T_1 \cup T_2$
- Separate: Given a test tube $T$ and a given bit in position $i$, this operation separates DNA's sequences into two groups: in test tube $T_1$, those with a value 1 in that position, and in test tube $T_0$ the remaining ones. Moreover, the entire content of $T$ is discarded.
  Notation : $(T_0; T_1) \leftarrow (T; i)$
- Set: Given a test tube $T$ and a particular bit at position $i$, all the DNA's sequences receive a sticker that corresponds to that bit.
  Notation : $set(T; i)$
- Clear: Given a test tube $T$ and a particular bit at position $i$, all the DNA's sequences with eventual stickers are removed in such a way that there is no match at that position.
  Notation : $clear(T; i)$

C. A. A. Sanches , N. Y. Soma: A polynomial-time DNA computing solution for the Bin-Packing Problem, Applied Mathematics and Computation 215 (2009) 2055–2062

# DNA algorithms

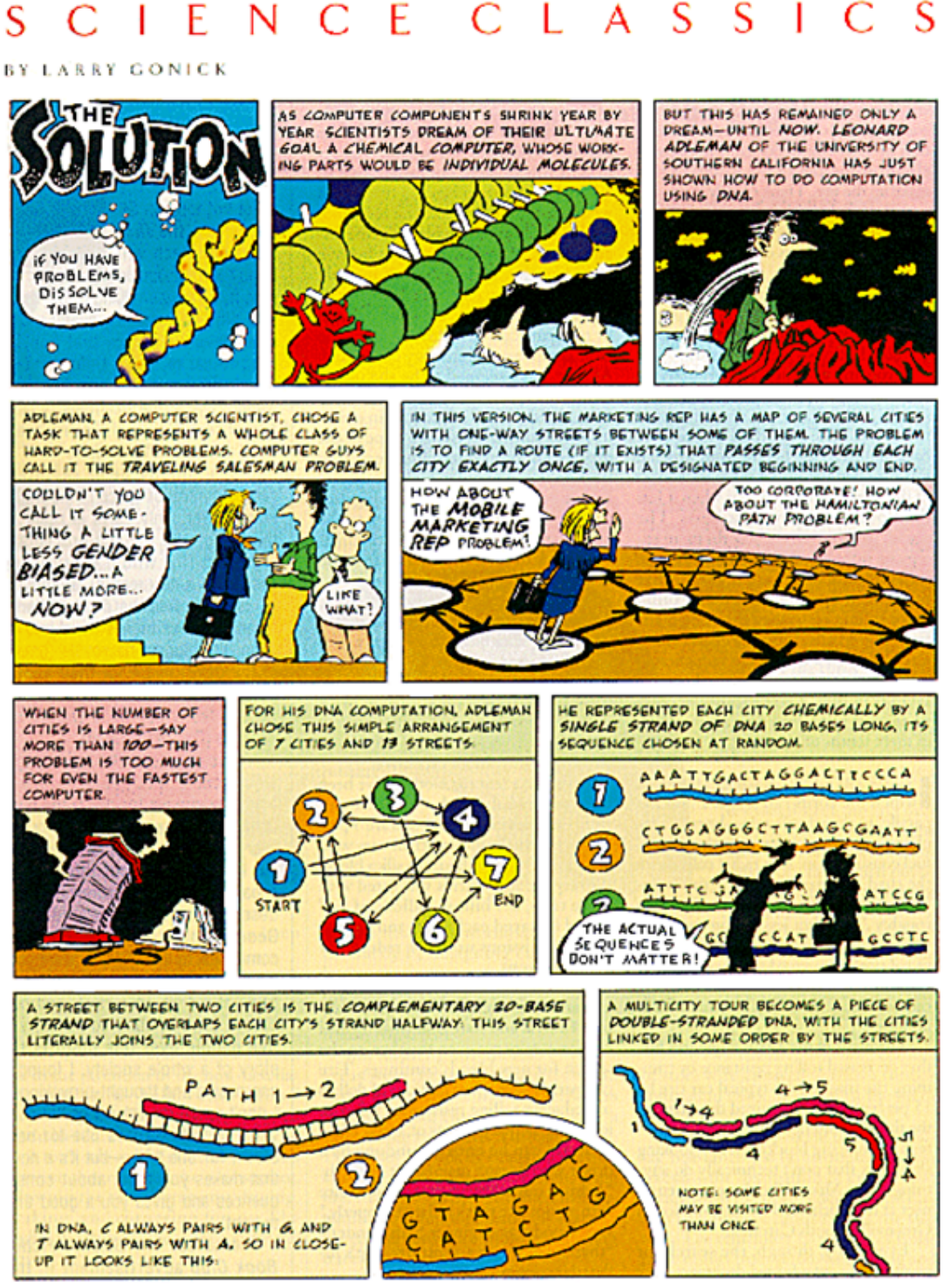
- The language is amended by initialisers and iterators
- Combinations of the models (e.g. Stickers plus amplify, append and detect) can be shown to solve NP-hard problems in polynomial time (assuming an exponential amount of DNA)
- A more formal approach is taken in $H$-systems (splicing systems) which have (for a finite set of slicing rules) the computing power as finite automata.
- Universal computation can be achieved by extended $H$-systems with permitting contexts which compute at the level of Turing machines
- Algorithms are obviously quite complex and unrealistic from a practical point of view.
- Note: algorithms run on multi-sets (i.e. set which may have more than one element of a kind)

http://www.scholarpedia.org/article/Splicing_systems#Formal_Models_of_DNA_recombination

# Conclusion on DNA computing

- Rather unsophisticated w.r.t. the computational problem, possibly to become more efficient (other molecules?)
- Can be performed automatically
- Most genetic operations are relatively cheap today, they become faster and smaller, improving in price/performance exponentially
- DNA chips (microarrays) are in operation (for diagnosis and research)
- Micro-electromechanical systems (MEMS) are there to operate them

L.M. Adleman: Molecular computation of solutions to combinatorial problems. Science 226 (1994), 1021-1024.G. Paun: Computing with Bio-Molecules, Springer-Verlag, 1998.




*Discover magazine published an article in comic strip format about Leonard Adleman's discovery of DNA computation. Not only entertaining, but also the most understandable explanation of molecular computation I have ever seen.*

Deepthi Bollu