

## The Birth of DNA Computing

- Leonid Adleman is a computer scientist and mathematician.
- A pretty good one—he helped invent RSA (It stands for Rivest-Shamir-Adleman), the standard algorithm used for encryption.
- In the 1990s he started to get interested in the mathematics of AIDS, so started to learn about molecular biology...

---

---

---

---

---

---

---

---

## The Moment of Insight



- One night in 1993 he was reading the *Molecular Biology of the Gene*. He sat up in bed and said to his wife:
  - “Geez, these things could compute”
- He then showed how to solve the Hamiltonian Path problem and solved a small instance of it in the laboratory.
- His subsequent paper was published in *Science* in 1994 and started the field of DNA computing...
- Now huge with hundreds of papers, annual conferences, etc

---

---

---

---

---

---

---

---

## DNA Computing--Overview

- Basics of DNA
- Adleman's experiment
- Programming a DNA computer
- Prospects for DNA computing

---

---

---

---

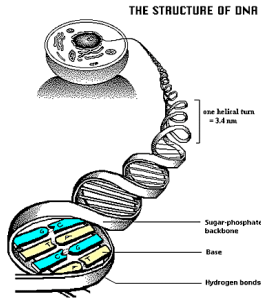
---

---

---

---

## DeoxyriboNucleic Acid (DNA)



- DNA is a right handed double helix, with about 10 nucleotide pairs per helical turn.
- Each spiral strand, composed of a sugar phosphate backbone and attached bases, is connected to a complementary strand by hydrogen bonding (non-covalent) between paired bases,
  - adenine (A) with thymine (T) and
  - guanine (G) with cytosine (C).
- First described by James Watson and Francis Crick in 1953. And don't forget Maurice Wilkes or Rosalind Franklin!

---

---

---

---

---

---

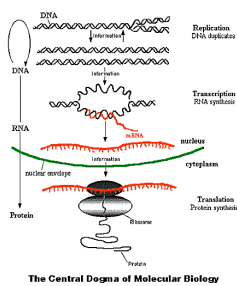
---

---

---

---

## The Role of DNA



The cell performs two important operations on DNA.

- The DNA **replicates** itself when the cell splits.
- The DNA instructs the cell how to make proteins.

Proteins are fundamental to life. They are involved in almost all biological activities, structural or enzymatic.

---

---

---

---

---

---

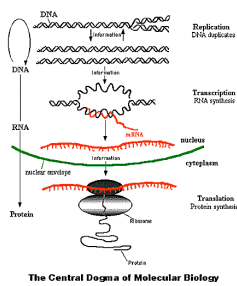
---

---

---

---

## The Role of DNA



Construction of proteins has three stages

1. The information in the DNA is copied to messenger RNA (mRNA) This is called **transcription**.
2. The mRNA migrates from the cell nucleus to the cell body.
3. Ribosomes "read" the information on the mRNA and use it for protein synthesis. This process is called **translation**.

---

---

---

---

---

---

---

---

---

---

## Ribonucleic Acid (RNA)

RNA is a chemical similar to a single strand of DNA.

In RNA, the letter U, which stands for **uracil**, is substituted for T in the genetic code.

RNA delivers DNA's genetic message to the cytoplasm of a cell where proteins are made.

---



---



---



---



---



---



---

## Translation from mRNA to Protein

mRNA is read as a sequence of **codons** each of which is a triplet of bases.

Each codon has a specific meaning:

- Start
- Stop
- Create a specific amino acid

---



---



---



---



---



---



---

## Translation from mRNA to Protein

Construction of the protein proceeds as follows

- The ribosome binds to the mRNA at the **start codon (AUG)** that is recognized only by the initiator tRNA.
- The ribosome moves from codon to codon along the mRNA. Amino acids are added one by one to the sequence of amino acids
- At the end, a release factor binds to the stop codon, terminating translation and releasing the complete protein from the ribosome.

---



---



---



---



---

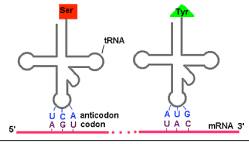


---



---

## Genetic Code



		2nd base in codon					
		U	C	A	G		
1st base in codon	U	Phe	Ser	Tyr	Cys	U	
	Phe	Ser	TYP	Cys	U		
	Leu	Ser	STOP	Trp	C		
	Leu	Phe	His	Arg	U		
C	Leu	Phe	His	Arg	U		
	Leu	Phe	Gln	Arg	A		
	Leu	Phe	Gln	Arg	G		
	Ile	Thr	Asn	Ser	U		
A	Ile	Thr	Asn	Ser	U		
	Ile	Thr	Lys	Arg	A		
	Met	Thr	Lys	Arg	G		
	Val	Ala	Asp	Gly	U		
G	Val	Ala	Asp	Gly	U		
	Val	Ala	Glu	Gly	A		
	Val	Ala	Glu	Gly	G		

The Genetic Code

- Ala: Alanine
- Cys: Cysteine
- Asp: Aspartic acid
- Glu: Glutamic acid
- Phe: Phenylalanine
- Gly: Glycine
- His: Histidine
- Ile: Isoleucine
- Lys: Lysine
- Leu: Leucine
- Met: Methionine
- Asn: Asparagine
- Pro: Proline
- Gln: Glutamine
- Arg: Arginine
- Ser: Serine
- Thr: Threonine
- Val: Valine
- Trp: Tryptophane
- Tyr: Tyrosine

---

---

---

---

---

---

---

---

---

---

---

---

## Operations on DNA Molecules

**Separating DNA strands.** By gently heating DNA the two strands come apart (called *denaturation*)

**Binding DNA strands.** By cooling DNA the two strands come apart (called *annealing*)

When two strands do not match completely we obtain DNA with *sticky ends*

5'-ACCTAGCGC-3'  
3'-TCGCGTTA-5'

□

ACCTAGCGC  
TCGCGTTA

---

---

---

---

---

---

---

---

---

---

---

---

## Operations on DNA Molecules

**Filling in incomplete DNA strands.** A DNA molecule with sticky ends can be *completed*

ACCTAGCGC  
TCGCGTTA

□

ACCTAGCGCAAT  
TGGATCGCGTTA

---

---

---

---

---

---

---

---

---

---

---

---

## Operations on DNA Molecules

### Synthesizing DNA.

We can use "synthesizing robots" to create an arbitrary DNA strand (3' to 5' end)  
Then if desired we can create the complementary strand.

---

---

---

---

---

---

---

---

## Operations on DNA Molecules

### Cutting DNA.

Enzymes called *endonucleases* can be used to split a DNA strand or double strand. Some cut at any point while others are site restricted, binding to a particular pattern and cutting at a specified point.  
Cuts can be blunt (straight through both strands) or staggered leaving sticky ends.

5'-ACCGAATTC AAT-3'  
3'-TGGCTTAAGTTA-5'

□ cut with *EcoRI* which matches 5'-GAATTC (but not 3'-GAATTC!)

5'-ACCG-3'      5'-AATTC AAT-3'  
3'-TGGCTTAA-5'      3'-GTTA-5'

The enzymes are catalysts which means that with enough time they cut at all sites they match

---

---

---

---

---

---

---

---

## Operations on DNA Molecules

### Linking DNA.

Enzymes called *ligases* can be used to join two DNA molecules with complementary sticky ends. Also blunt ends but this is not useful. This is called *ligation*.

5'-ACCG-3'      5'-AATTC AAT-3'  
3'-TGGCTTAA-5'      3'-GTTA-5'

□

5'-ACCGAATTC AAT-3'  
3'-TGGCTTAAGTTA-5'

---

---

---

---

---

---

---

---

## Operations on DNA Molecules

### Multiplying DNA.

A technique called *amplification* can be used to produce  $2^n$  copies of a particular molecule in the solution in  $n$  steps.

Basically we separate the chains, add primers to identify the single strands to be multiplied, and then use these as templates.

### Filtering DNA.

We can separate out those single strands in a solution that contain a particular sub-pattern

Basically we attach to a solid support strands which are complementary to the desired pattern and pour the solution over them. The desired strands will adhere to these.

---

---

---

---

---

---

---

---

## Operations on DNA Molecules

### Separation by length

It is possible to separate the molecules in a solution by length (actually weight) using *gel electrophoresis*.

### Reading DNA

It is also possible to read the nucleotides in a strand of DNA, however this process is quite slow: it's easier to "write" DNA than "read" DNA!

---

---

---

---

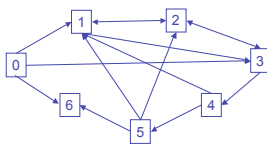
---

---

---

---

## Adleman's Experiment



- The Hamiltonian path problem is to find if there is a path in a directed graph that passes exactly once through each vertex.
- This is an NP-hard problem and on conventional computers the best algorithms have exponential time complexity.
- Adleman used DNA to solve the problem for the example graph.
- The number of laboratory steps was **linear** in the size of the graph

---

---

---

---

---

---

---

---

### Adleman's Algorithm

- Input:* A directed graph  $G$  with  $n$  vertices  
 Two designated vertices  $v_{in}$  and  $v_{out}$
- Step 1:* Generate paths in  $G$  randomly in large quantities.  
*Step 2:* Remove all paths that do not start with  $v_{in}$  and end with  $v_{out}$   
*Step 3:* Remove all paths that are not of length  $n$   
*Step 4:* For each of the  $n$  vertices  $v$ , remove all paths that do not contain  $v$
- Output:* Yes if any path remains, no otherwise
- Uses massive parallelism to perform search
  - Linear in the number of vertices
  - Not guaranteed to be correct if it says no!!!

---

---

---

---

---

---

---

---

---

---

### DNA Encoding of Problem

- Each vertex  $v_i$  was encoded by a unique sequence of 20 nucleotides
  - $v_2 = 5'-TATCGGATCGGTATATCCGA-3'$
  - $v_3 = 5'-GCTATTCGAGCTTAAAGCTA-3'$
  - $v_4 = 5'-GGCTAGGTACCAGCATGCTT-3'$
- Each edge  $v_i \rightarrow v_j$  was encoded by  $e_i s_j$  where edge  $v_i$  is encoded by  $e_i$ ,  $s_j$  and  $e_j$  is the complement to  $e_i$ 
  - $v_2 \rightarrow v_3 = 3'-CATATAGGCTCGATAAGCTC-5'$
  - $v_3 \rightarrow v_2 = 3'-GAATTCGATATAGCCTAGC-5'$
  - $v_3 \rightarrow v_4 = 3'-GAATTCGATCCGATCCATG-5'$

---

---

---

---

---

---

---

---

---

---

### DNA Encoding of Problem

```

      v2                v3
TATCGGATCGGTATATCCGA GCTATTCGAGCTTAAAGCTA
      CATATAGGCTCGATAAGCTCGAATTCGATCCGATCCATG
          v2->v3                v3->v4
  
```

- By annealing the codes of the vertices act as splints with the code for the edges, allowing longer and longer molecules to be formed representing paths

---

---

---

---

---

---

---

---

---

---

## DNA Programming

- Based on the filtering approach to DNA computing several "test tube" programming languages have been suggested starting with Lipton in 1995.
- A (test) tube is a set of sequences over the alphabet A, C, G, T.
- They can be understood as belonging to a new paradigm for computing languages: The DNA Paradigm.

---

---

---

---

---

---

---

---

## Basic Operations

- **Merge:** given tubes  $T_1$  and  $T_2$  form their union  $T_1 \sqcup T_2$
- **Amplify:** given tube  $T$  produce two copies  $T_1$  and  $T_2$
- **Select:** given tube  $T$  select an element of  $T$  at random or else if  $T$  is empty return *empty*.
- **Separate:** given tube  $T$  and a sequence  $w$ , produce two tubes
  - $+(T,w)$  which contains all strands in  $T$  which contain  $w$
  - $-(T,w)$  which contains all strands in  $T$  which do not contain  $w$
- **Length-separate:** given tube  $T$  and a length  $n$ , produce two tubes
  - $le(T,n)$  which contains all strands in  $T$  with length  $\leq n$
  - $gt(T,w)$  which contains all strands in  $T$  with length  $> n$
- **Position-separate:** given tube  $T$  and a sequence  $w$ , produce two tubes
  - $B(T,w)$  which contains all strands in  $T$  which start with  $w$
  - $E(T,w)$  which contains all strands in  $T$  which end with  $w$

---

---

---

---

---

---

---

---

## DNA Programming

- This language (or close variants) has been used to program solutions to
  - Hamiltonian path (Adleman, 1993)
  - SAT (Lipton 1995)
  - Subgraph isomorphism (Amos 1997)
  - three vertex colourability (Amos 1997)
  - finding maximum clique (Amos 1997)
- And it has been claimed that all NP-hard problems can be solved in it (although no proof)

---

---

---

---

---

---

---

---

## DNA Programming

- However this language is **not** computationally complete
  - not all computable functions can be programmed
  - some operations are more expensive to program in the test tube language than with a Turing machine, e.g. it is possible to invert a function defined by a circuit in linear time but not with this language
- The problem is that it doesn't utilise ligation and annealing.

---

---

---

---

---

---

---

---

## Practicality of Adleman's Algorithm

- Although linear, Adleman's algorithm does not scale up.
- It requires  $n!$  copies of vertices, edges to work where  $n$  is the number of vertices.
- To solve a problem with 200 vertices (a size of practical importance and easily handled using traditional OR techniques) require  $3 \times 10^{28}$  Kg of DNA which is more than the weight of the earth!
- There is also a problem with long sequences, they tend to be very fragile.
- Other algorithms which are designed to overcome these limitations have been suggested but not yet implemented.

---

---

---

---

---

---

---

---

## Conclusion

- DNA Computing has enormous potential
    - one gram of dried DNA stores as much as a trillion CD-ROM disks
    - potential for massive parallelism, allowing NP hard problems to be solved in polynomial time
  - However,
    - There is no real programming paradigm,
      - filtering is not powerful enough
    - Also handling errors, fragmenting DNA etc is tricky
- But it has only been 9 years....

---

---

---

---

---

---

---

---

## References

- *The Bit and the Pendulum*. T. Siegfried. John Wiley & Sons. 2000.
- *Computing with Cells and Atoms*. C. S. Calude, G. Paun
- Some images are taken from the The National Health Museum and the Access Excellence Fellows' Graphics Gallery at <http://www.accessexcellence.org/AB/GG/>

---

---

---

---

---

---

---