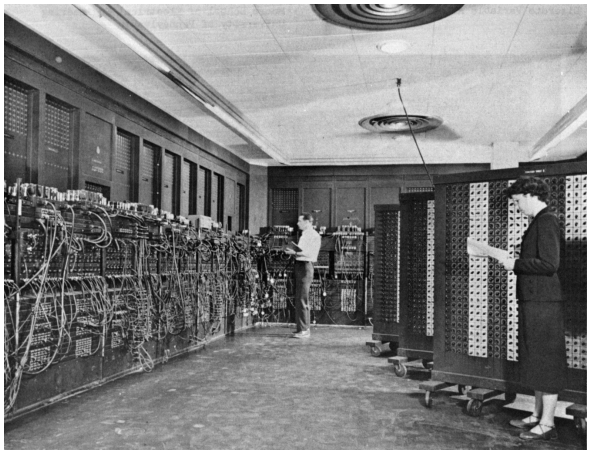
Molecular Computing

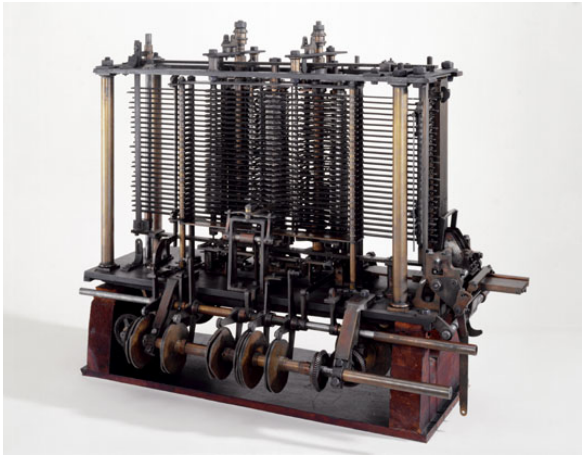
david.wishart@ualberta.ca
3-41 Athabasca Hall
Sept. 30, 2013

What Was The World's First Computer?

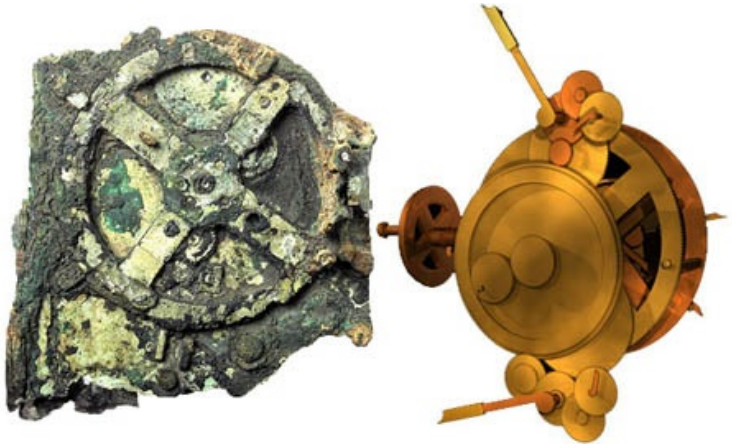
The World's First Computer?



ENIAC - 1946



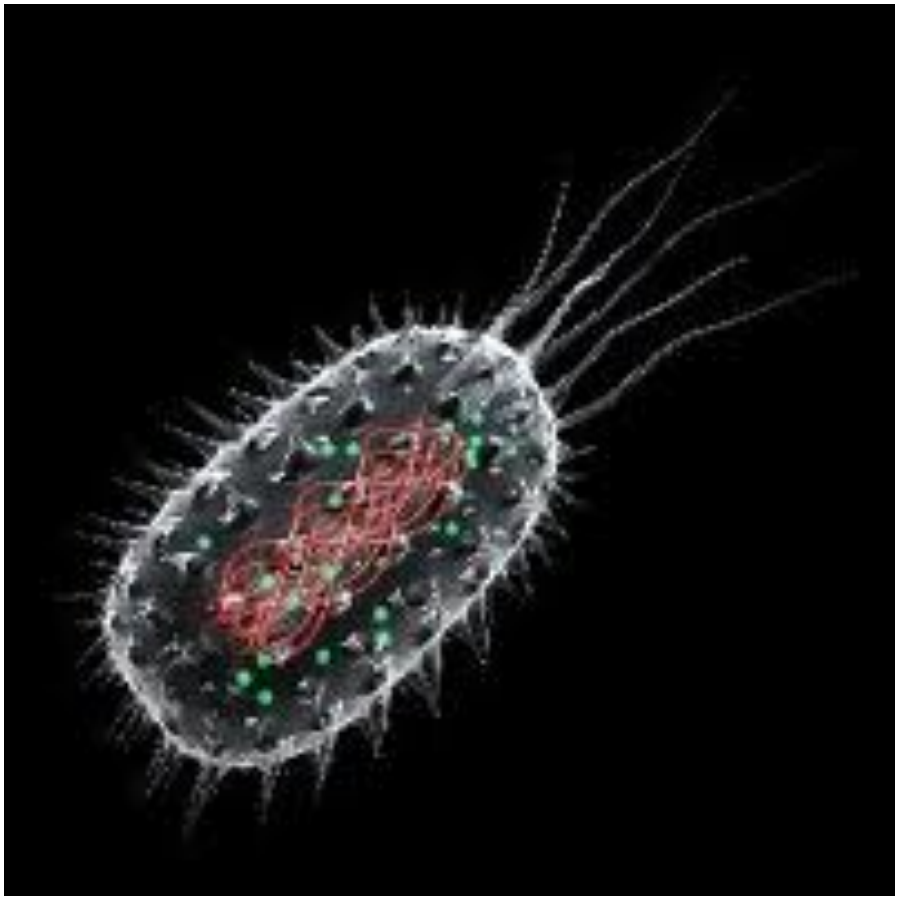
Babbage Analytical Engine - 1837



Antikythera Mechanism - 80 BP

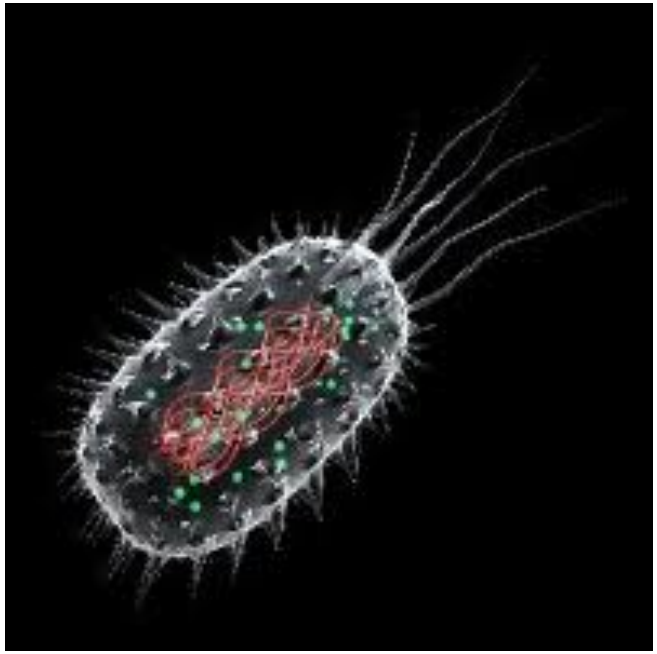
Human Brain – 1,000,000 BP

The World's First Computer?



The Cell – 2,000,000,000 BP

Cells versus Computers



Cell

Computer

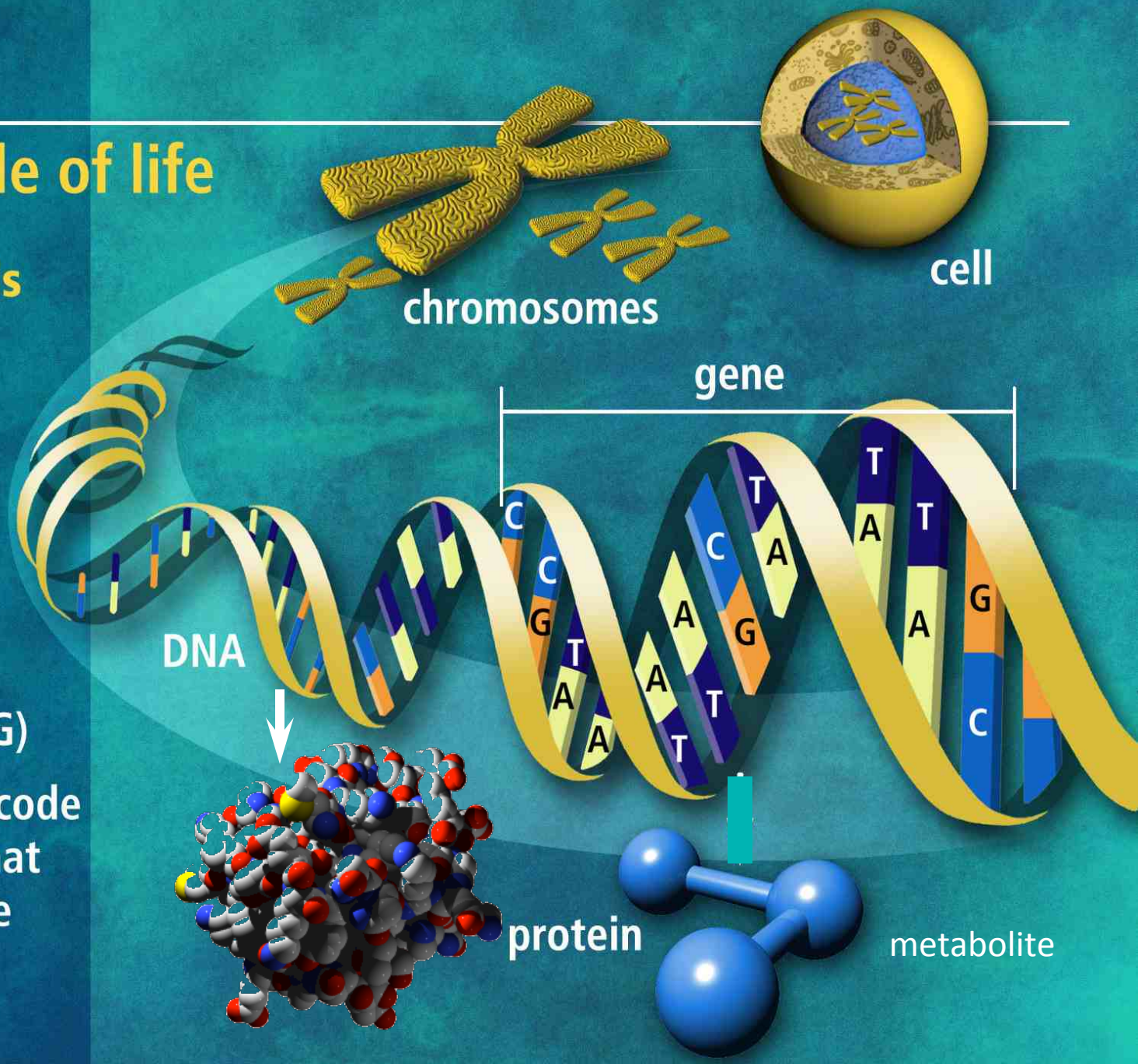
DNA

the molecule of life

Trillions of cells

Each cell:

- 46 human chromosomes
- 2 m of DNA
- 3 billion DNA subunits (the bases: A, T, C, G)
- 21,000 genes code for proteins that perform all life functions



Y-GA 98-090R

Cells versus Computers

- Base-4 (ACGT)
- DNA
- Bases
- Codons
- Genetic Code
- Gene/Protein
- Chromosome
- Genome Size

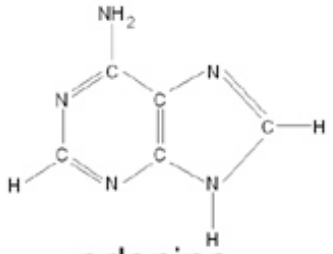
- Base-2 (101010)
- Magnetic tape/Disk
- Bits/Transistors
- Bytes
- Instruction Set
- Program
- Hard Disk
- Disk Capacity

Data Storage

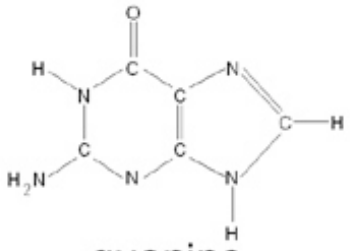
Cell **Computer**

On/Off Signals

Purines

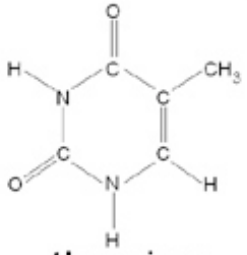


adenine

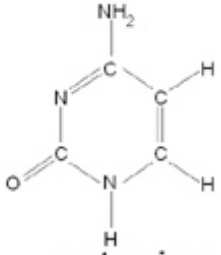


guanine

Pyrimidines



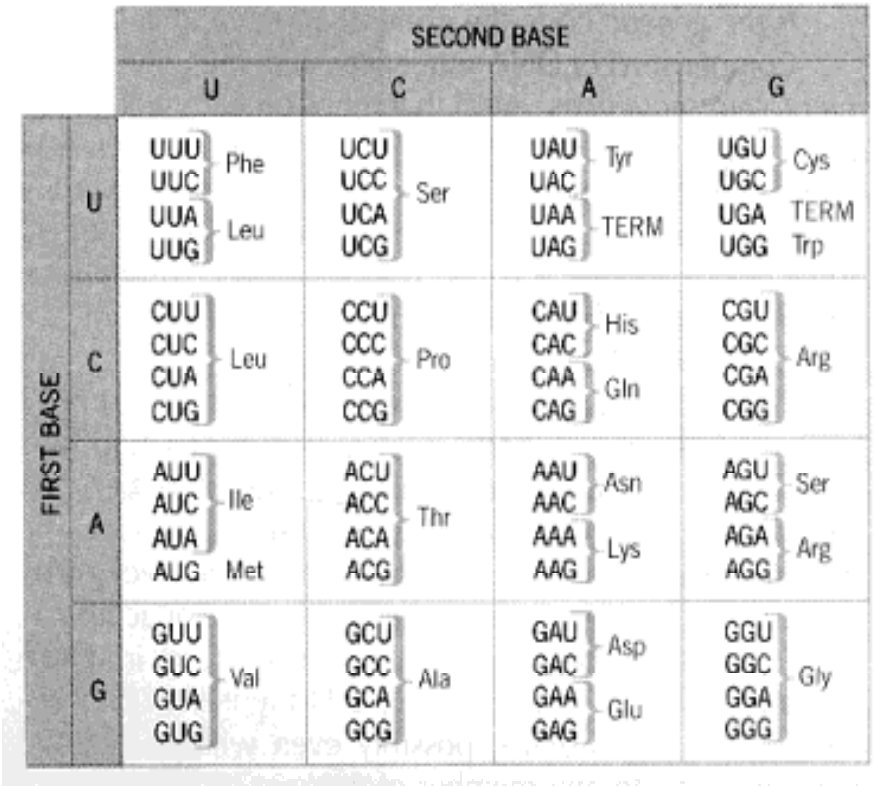
thymine



cytosine

Cell **Computer**

The Instruction Set



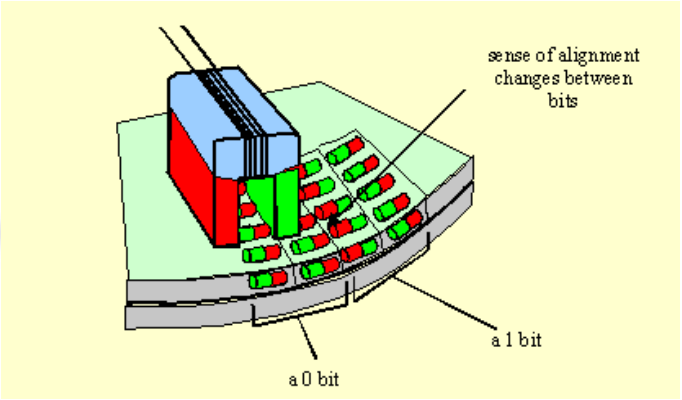
FIRST BASE	SECOND BASE: U	C	A	G
U	UUU Phe, UUC Phe, UUA Leu, UUG Leu	UCU Ser, UCC Ser, UCA Ser, UCG Ser	UAU Tyr, UAC Tyr, UAA TERM, UAG TERM	UGU Cys, UGC Cys, UGA TERM, UGG Trp
C	CUU Leu, CUC Leu, CUA Leu, CUG Leu	CCU Pro, CCC Pro, CCA Pro, CCG Pro	CAU His, CAC His, CAA Gln, CAG Gln	CGU Arg, CGC Arg, CGA Arg, CGG Arg
A	AUU Ile, AUC Ile, AUA Ile, AUG Met	ACU Thr, ACC Thr, ACA Thr, ACG Thr	AAU Asn, AAC Asn, AAA Lys, AAG Lys	AGU Ser, AGC Ser, AGA Arg, AGG Arg
G	GUU Val, GUC Val, GUA Val, GUG Val	GCU Ala, GCC Ala, GCA Ala, GCG Ala	GAU Asp, GAC Asp, GAA Glu, GAG Glu	GGU Gly, GGC Gly, GGA Gly, GGG Gly

00100110
00000001
10101010
11101101
00100110

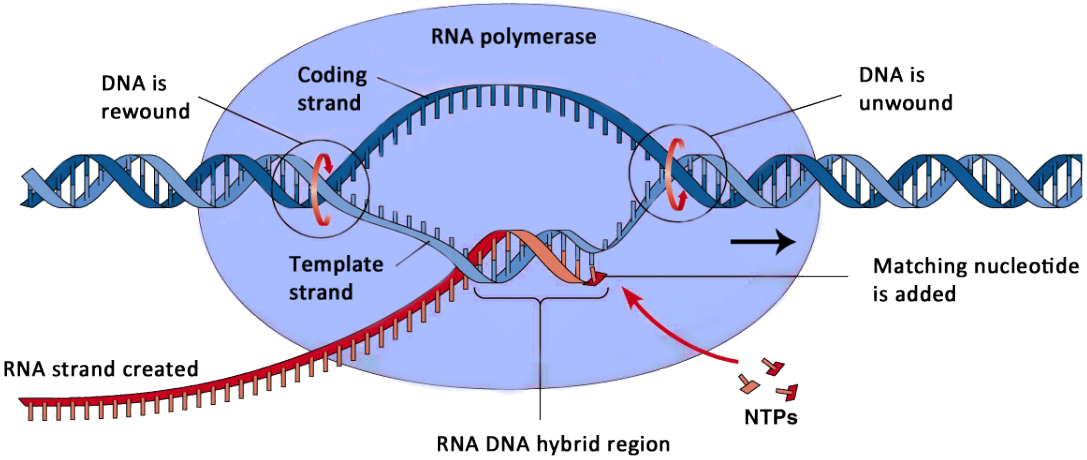
Cell **Computer**

Data Transcription

Computer



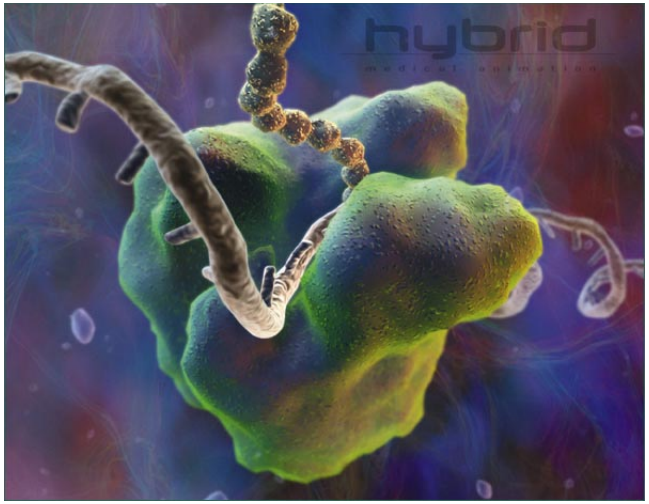
Cell



Data Processing

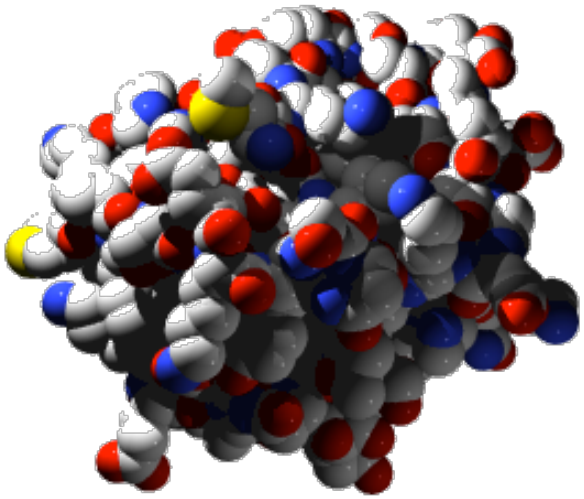
Computer

Cell



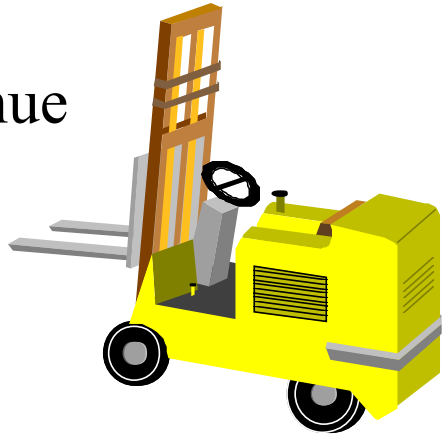
Data Interpretation

Cell



Computer

If light=red then
    turn north
else
    continue



World's Most Sophisticated Computer Language

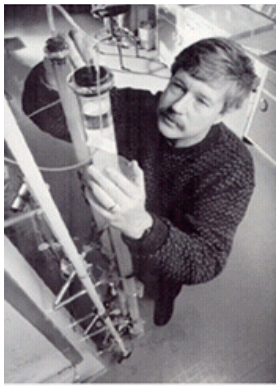
AUGGUCACU (UAG)
UCUGAAGUCA
GCUA
CUAG
GGA
CUUAUGC
AUUCGUAU

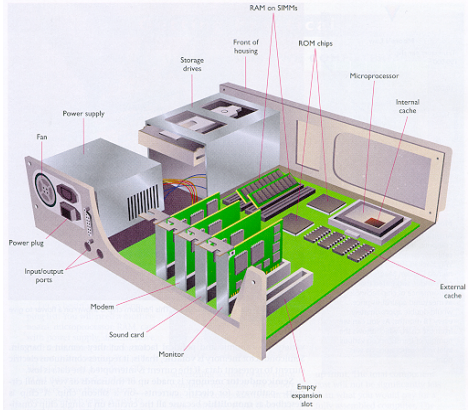
A program that codes for a self-assembling molecular machine

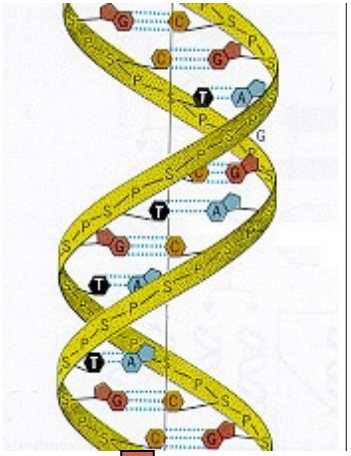
Key Differences...

- Cells use chemicals for information storage and transfer while computers use magnetic or electronic means
- In cells, proteins act as both programs and machines. In computers, programs and machines are separate with programs generally running the machines
- Proteins contain instructions for self-assembly, computers don't

DNA Computing



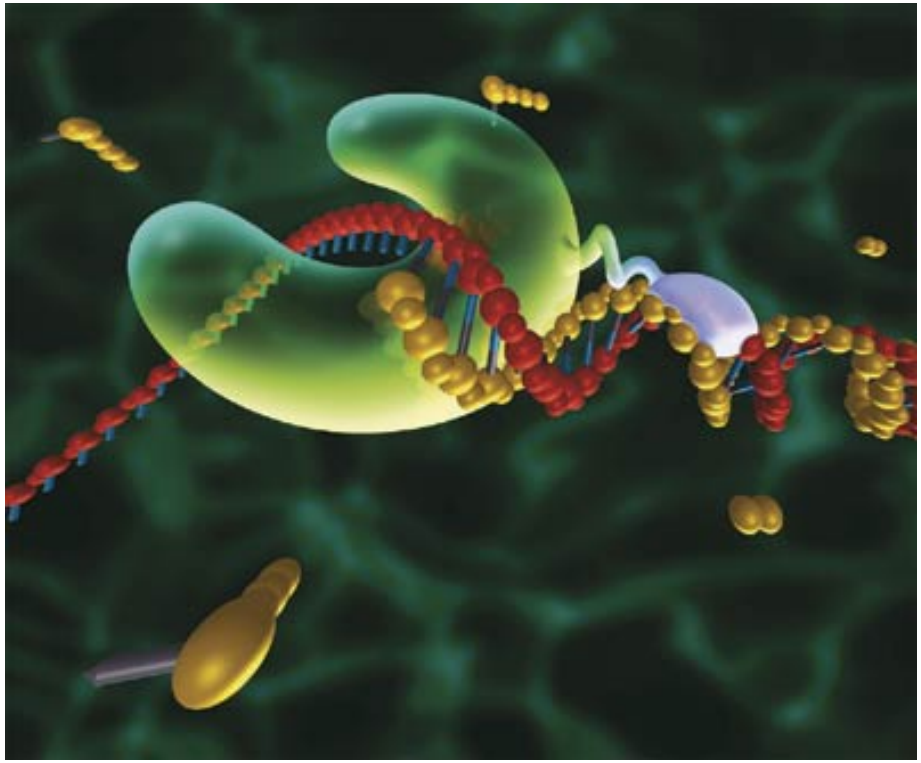




01100110101000 1

ATGCTCGAAGCT

The First Turing Machine**



DNA Polymerase

A Turing machine is a hypothetical device that manipulates symbols on a strip of tape according to a table of rules.

Computing With DNA

Display Settings: Abstract Send to:

Science. 1994 Nov 11;266(5187):1021-4.

Molecular computation of solutions to combinatorial problems.

Adleman LM.

Department of Computer Science, University of Southern California, Los Angeles 90089.

Abstract

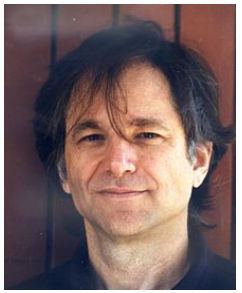
The tools of molecular biology were used to solve an instance of the directed Hamiltonian path problem. A small graph was encoded in molecules of DNA, and the "operations" of the computation were performed with standard protocols and enzymes. This experiment demonstrates the feasibility of carrying out computations at the molecular level.

Comment in

On the potential of molecular computing. [Science. 1995]
On the potential of molecular computing. [Science. 1995]
On the potential of molecular computing. [Science. 1995]
Building an associative memory vastly larger than the brain. [Science. 1995]
On the path to computation with DNA. [Science. 1994]

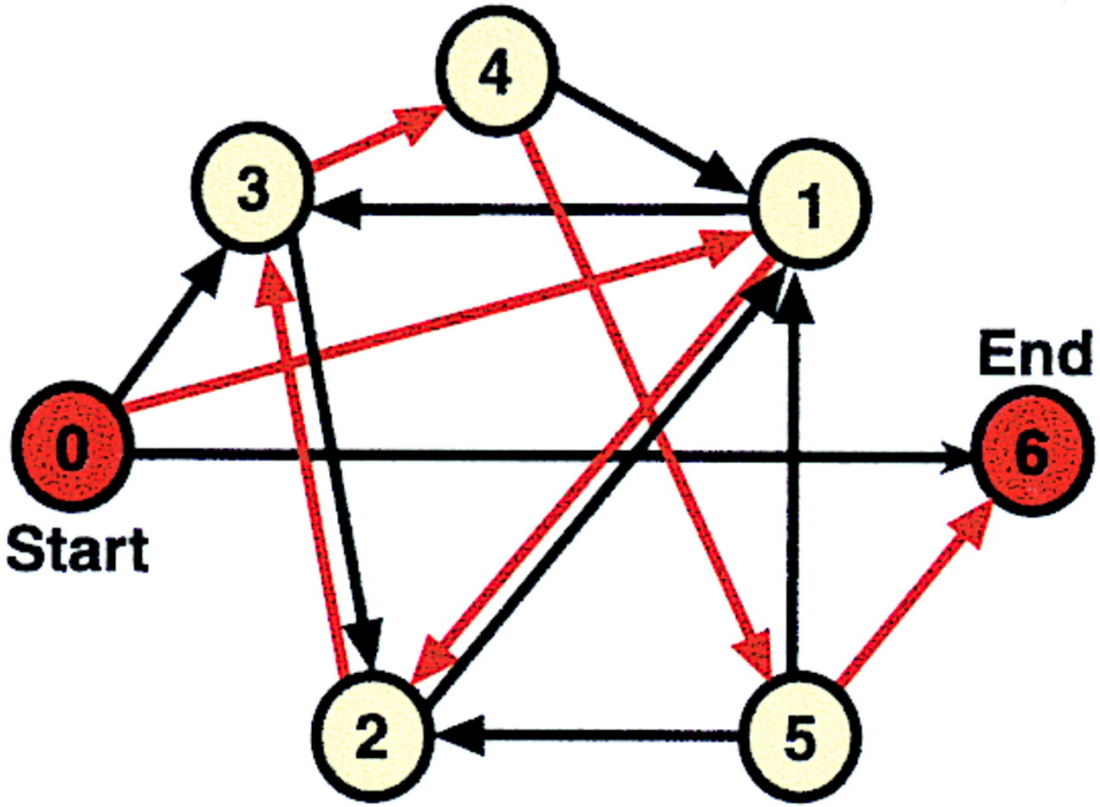
PMID: 7973651 [PubMed - indexed for MEDLINE]

Leonard Adleman**



- Mathematician, computer scientist, boxer
- Specialized in cryptography (RSA, 1983)
- Invented the term computer virus (1984)
- Became intrigued by “real” viruses (HIV)
- Published a paper on HIV in 1993 and decided to learn molecular biology
- Came up with DNA computing (1994) while studying “Molecular Biology of the Gene”
- Member of NAS, won Turing prize in 2002

Traveling Salesman Problem (7 cities, 14 air routes)



A salesman must find his way from city 0 to city 6, passing through each of the remaining cities only once
7 nodes, 14 edges – Correct path is marked in red

The Algorithm

- Generate Random paths through graph G
- From all paths created in step 1, keep only those that start at 0 and end at 6
- From all remaining paths, keep only those that visit exactly 7 vertices
- From all remaining paths, keep only those that visit each vertex at least once
- If any path remains, return“yes”;otherwise, return “no”

Represent Each City By A DNA Strand of 20 Bases

City 1 ATGCTCAGCTACTATAGCGA

City 2 TGCGATGTACTAGCATATAT

City 3 GCATATGGTACACTGTACAA

City 4 TTATTAGCGTGCGGCCTATG

City 5 CCGCGATAGTCTAGATTTCC

Etc.

Represent Each Air Route By Mixed Complementary Strands

City 1→2 TGATATCGCTACGCTACATG

City 2→3 ATCGTATATACGTATACCAT

City 3→4 GTGACATGTTAATAATCGCA

City 4→5 CGCCGGATACGGCGCTATCA

City 5→6 GATCTAAAGGTATGCATACG

Etc.

Connector DNA Links City DNA

ATGCTCAGCTACTATAGCGATGCGATGTACTAGCATATAT
TGATATCGCTACGCTACATG

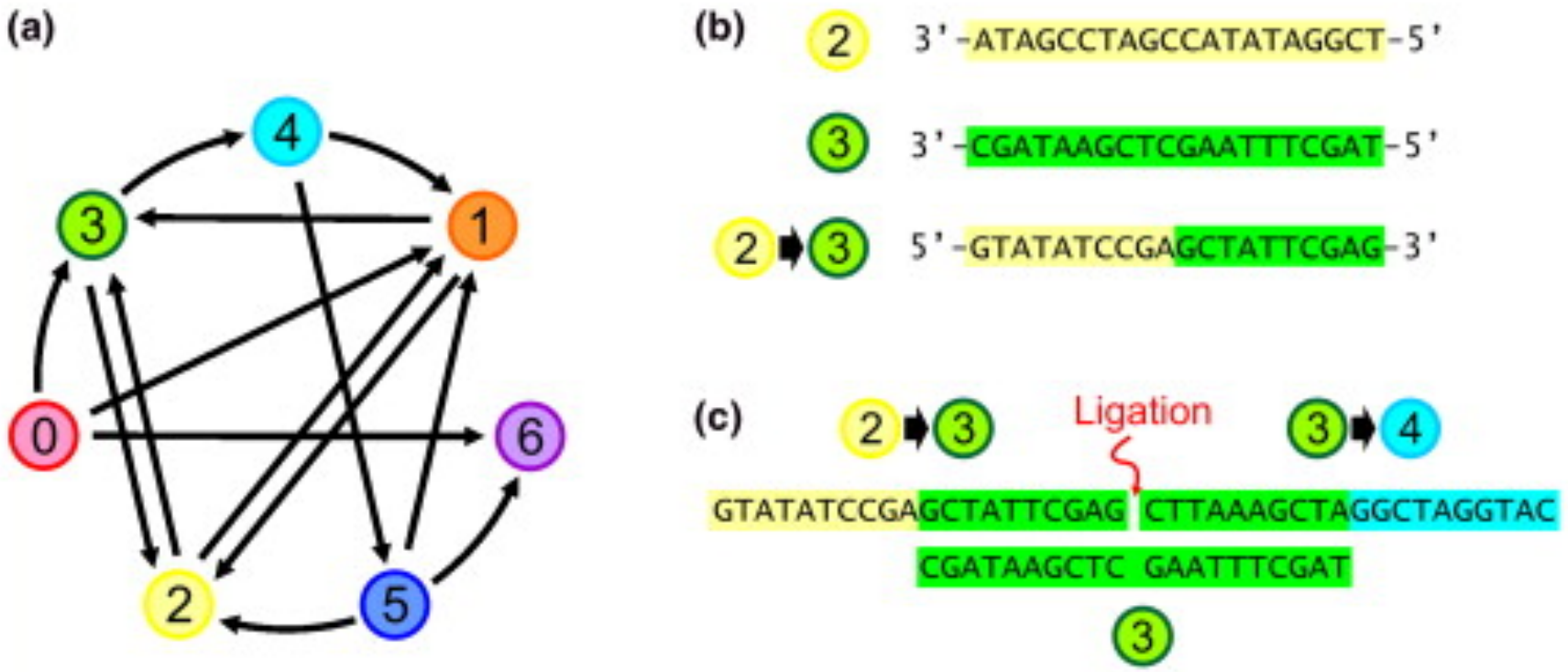
TGCGATGTACTAGCATATATGCATATGGTACACTGTACAA
ATCGTATATACGTATACCAT

GCATATGGTACACTGTACAATTATTAGCGTGCGGCCTATG
GTGACATGTTAATAATCGCA

TTATTAGCGTGCGGCCTATGCCGCGATAGTCTAGATTTCC
CGCCGGATACGGCGCTATCA

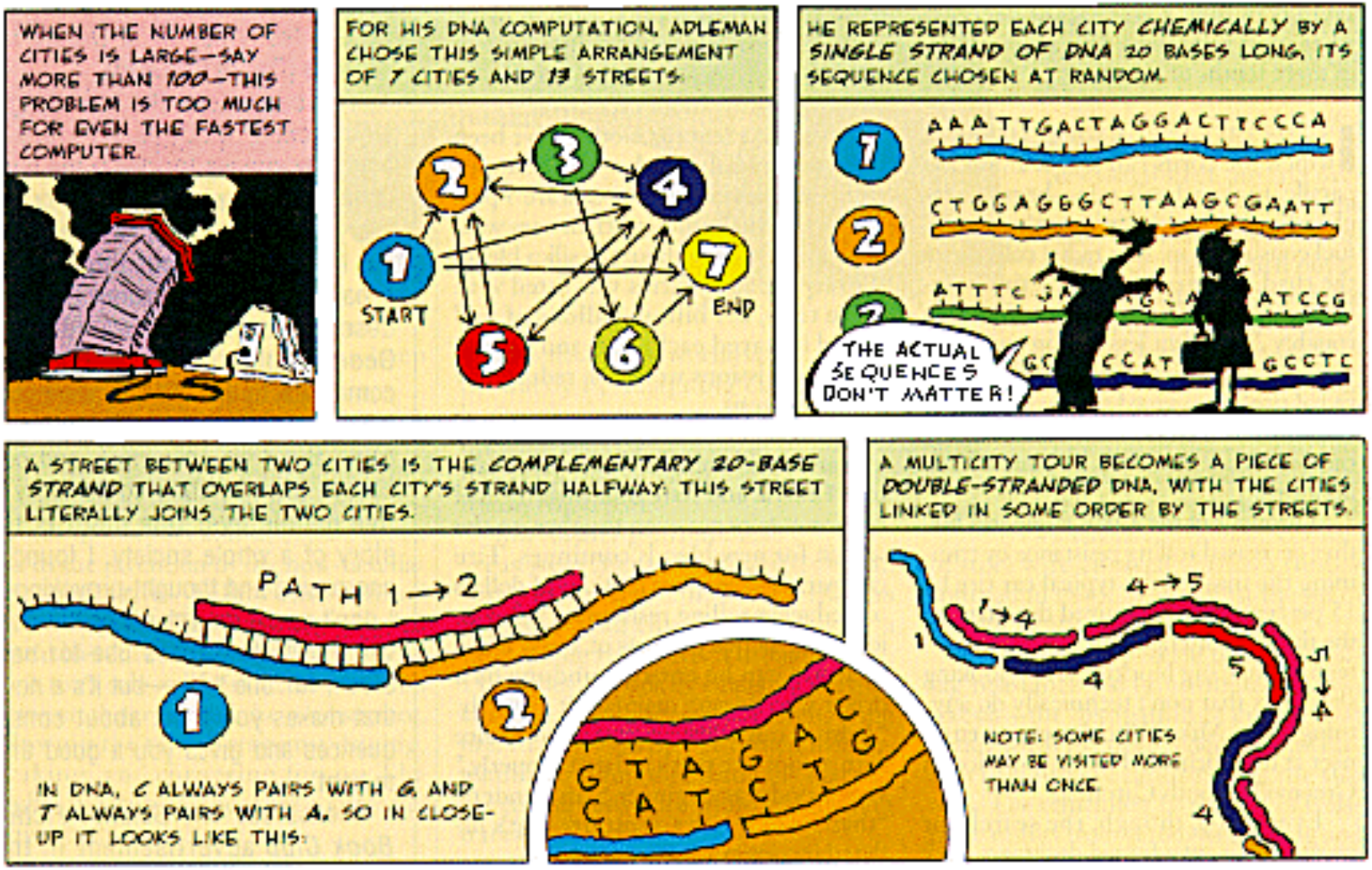
CCGCGATAGTCTAGATTTCCATACGTATGCTAGGCTATCG
GATCTAAAGGTATGCATACG

Anneal and Ligate DNA



Mix the City DNA with the Path DNA and let them randomly anneal (ligate with enzyme)
After annealing/ligation they will form (7-2)! different long (150 bp) DNA molecules
Select DNA molecules with the right start and ends (select by PCR) and length (gel)
Sequence the DNA to determine the best pathway (defined by the DNA sequence)

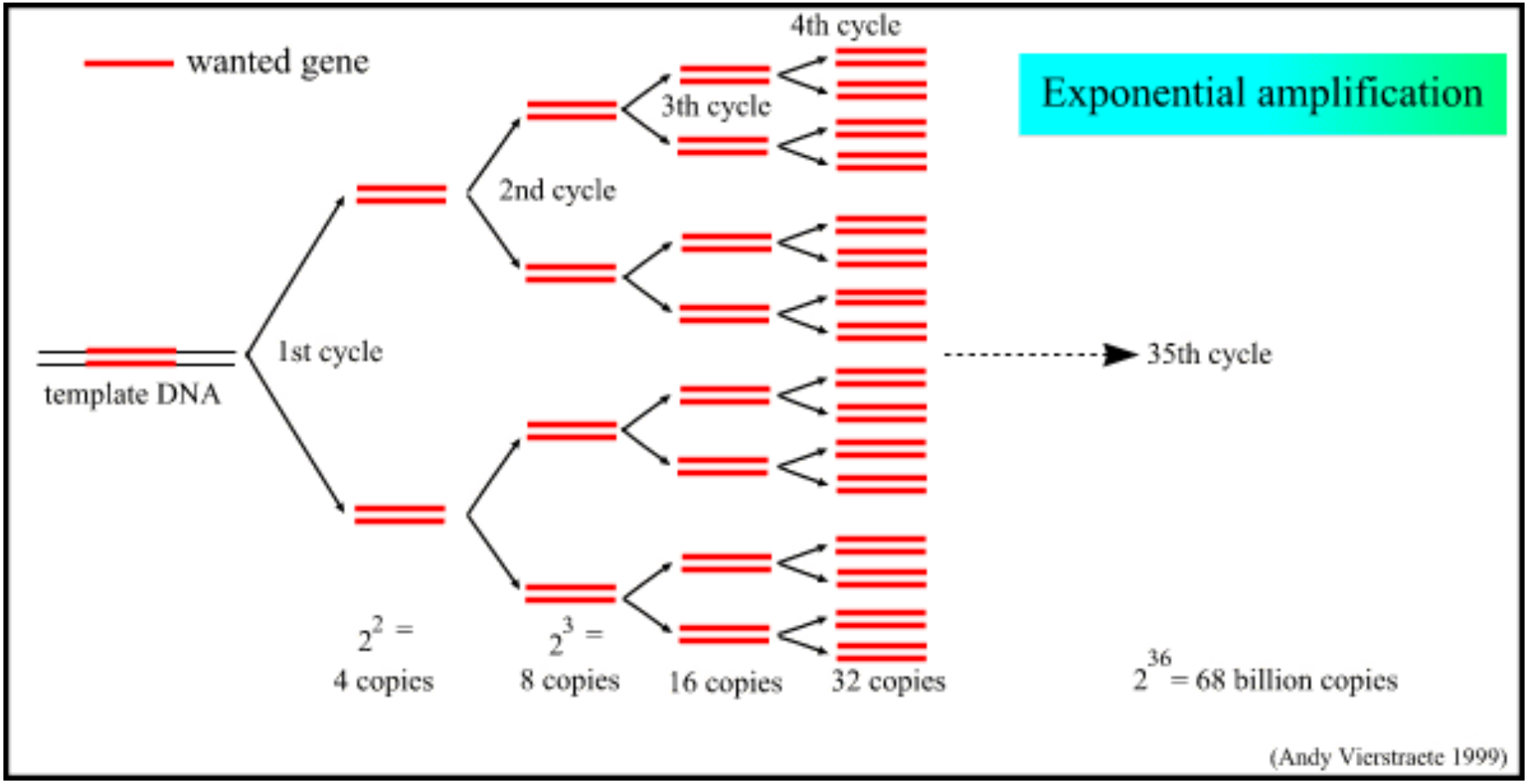
Cartoon Summary



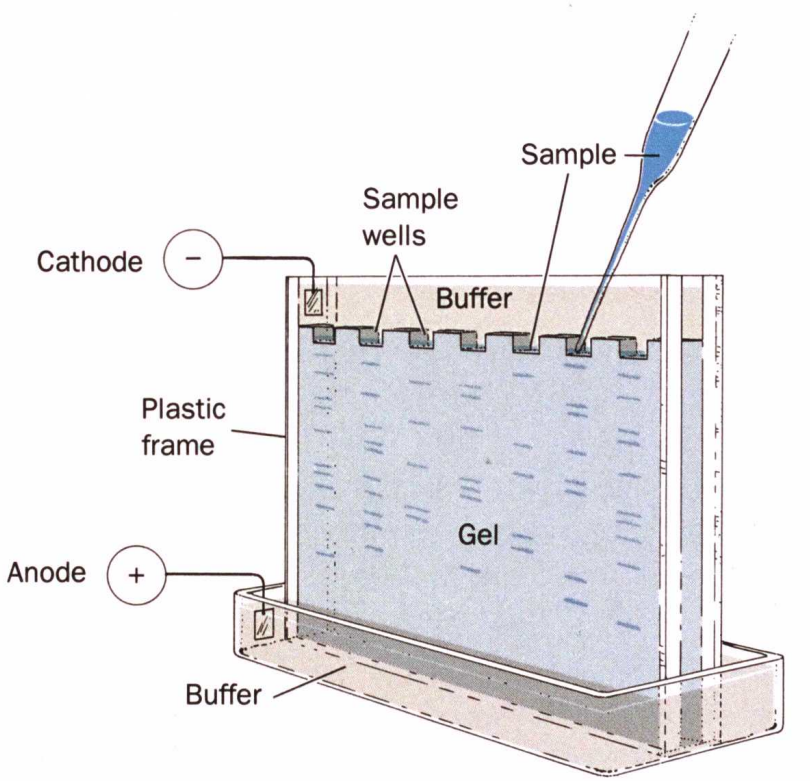
The DNA Algorithm

- Generate Random paths through graph G **(Annealing and Ligation)**
- From all paths created in step 1, keep only those that start at 0 and end at 6 **(PCR with selected primers)**
- From all remaining paths, keep only those that visit exactly 7 vertices **(Gel purification)**
- From all remaining paths, keep only those that visit each vertex at least once **(Magnetic bead purification)**
- If any path remains, return“yes”;otherwise, return “no” **(PCR)**

Principles of PCR



Gels & PCR



Not Exactly Routine “Calculations”

Advantages of DNA Computing

- With bases spaced at 0.35 nm along DNA, data density is 400,000 Gbits/cm compared to 3 Gbits/cm in typical high performance hard drive
- 1 gram of DNA can hold about $1x10^{14}$ MB of data
- A test tube of DNA can contain trillions of strands. Each operation on a test tube of DNA is carried out on all strands in the tube in parallel
- Adleman figured his computer was running 2×10^{19} operations per joule

Disadvantages of Adleman's Computer

- NOT competitive with the state-of-the-art algorithms on electronic computers
- Time consuming laboratory procedures (2 weeks)
- Good computer programs that can solve traveling salesman problem for 100 vertices in a matter of minutes
- Adleman's process to solve the traveling salesman problem for 200 cities would require an amount of DNA that weighed more than the Earth

More Advanced DNA Computers

Proc Natl Acad Sci U S A. 2003 Mar 4;100(5):2191-6. Epub 2003 Feb 24.

DNA molecule provides a computing machine with both data and fuel.

Benenson Y, Adar R, Paz-Elizur T, Livneh Z, Shapiro E.

Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel.

Abstract

The unique properties of DNA make it a fundamental building block in the fields of supramolecular chemistry, nanotechnology, nano-circuits, molecular switches, molecular devices, and molecular computing. In our recently introduced autonomous molecular automaton, DNA molecules serve as input, output, and software, and the hardware consists of DNA restriction and ligation enzymes using ATP as fuel. In addition to information, DNA stores energy, available on hybridization of complementary strands or hydrolysis of its phosphodiester backbone. Here we show that a single DNA molecule can provide both the input data and all of the necessary fuel for a molecular automaton. Each computational step of the automaton consists of a reversible software molecule input molecule hybridization followed by an irreversible software-directed cleavage of the input molecule, which drives the computation forward by increasing entropy and releasing heat. The cleavage uses a hitherto unknown capability of the restriction enzyme FokI, which serves as the hardware, to operate on a noncovalent software input hybrid. In the previous automaton, software input ligation consumed one software molecule and two ATP molecules per step. As ligation is not performed in this automaton, a fixed amount of software and hardware molecules can, in principle, process any input molecule of any length without external energy supply. Our experiments demonstrate 3 x 10(12) automata per microl performing 6.6 x 10(10) transitions per second per microl with transition fidelity of 99.9%, dissipating about 5 x 10(-9) W microl as heat at ambient temperature.

PMID: 12601148 [PubMed - indexed for MEDLINE] PMCID: PMC151317 **Free PMC Article**

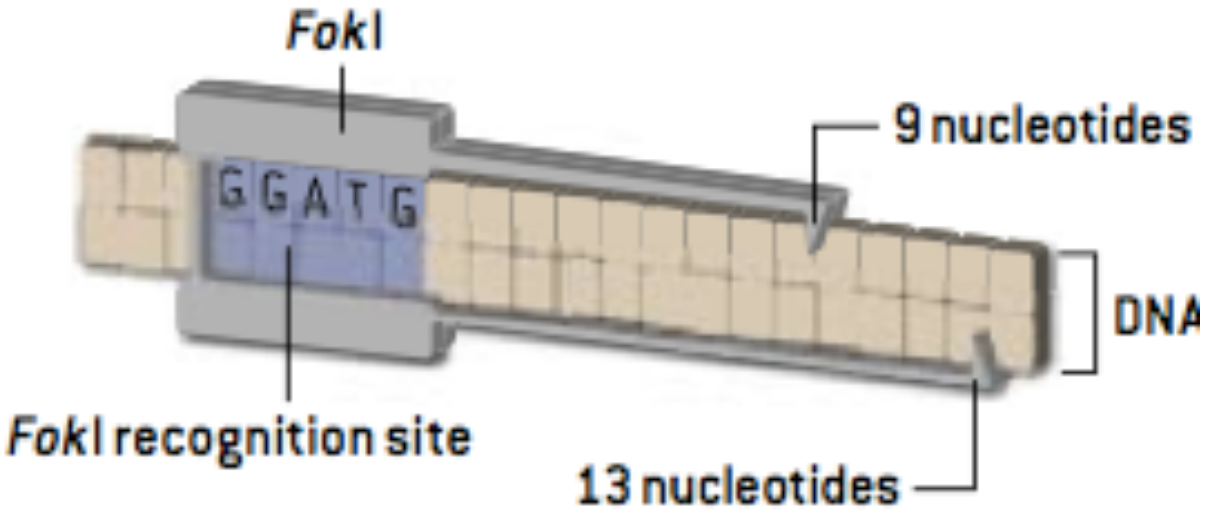
Computing with DNA

- Performs 330 trillion operations per second
- A spoonful contains 15,000 trillion “computers”/automatons
- Energy-efficiency is more than a million times that of a PC
- Guinness World Records recognized the computer as "the smallest biological computing device" ever constructed
- DNA acts as software, enzymes act as hardware
- Once the input, software, and hardware molecules are mixed in a solution it operates to completion without intervention
- The device can check whether a list of zeros and ones has an even number of ones
- It can only answer yes or no to a question

Finite Automaton

- Turing machine that moves in one direction
- Reads a series of symbols
- Changes its internal state according to transition rules
- A 2-state automaton can answer a yes-no question by alternating between states designated as 1 and 0
- Its state at the end of the calculation represents the result

Details



HARDWARE

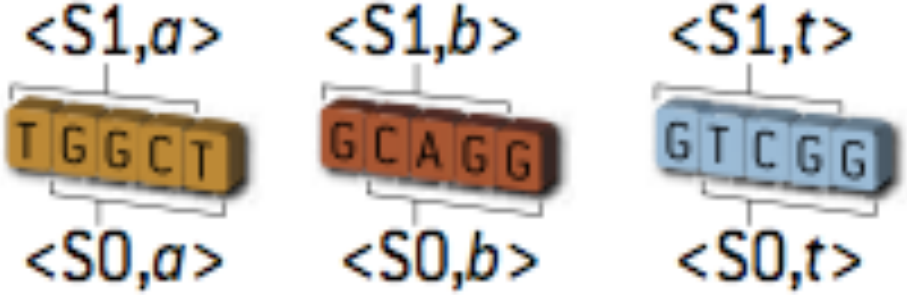
The *FokI* enzyme (*gray*) always recognizes the nucleotide sequence GGATG (*blue*) and snips a double DNA strand at positions 9 and 13 nucleotides downstream of that recognition site.

Details

SOFTWARE

Transition rules are encoded in eight short double-stranded DNA molecules containing the *FokI* recognition site (*blue*), followed by spacer nucleotides (*green*) and a single-stranded sticky end (*yellow*) that will join to its complementary sequence on an input molecule.

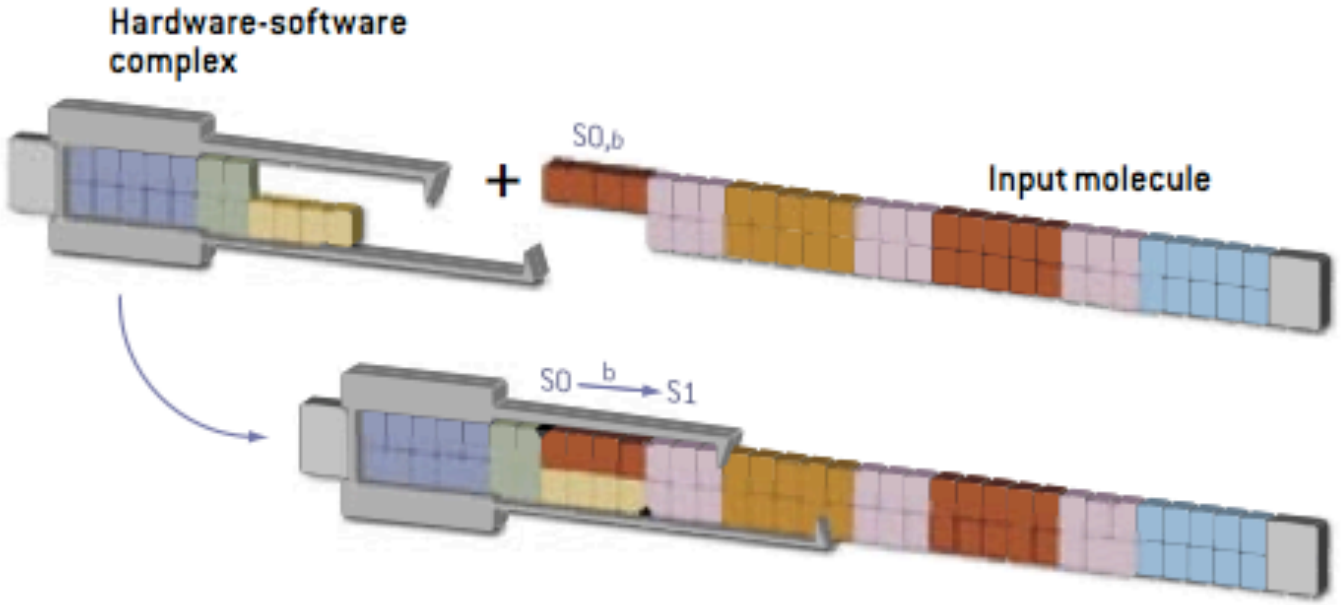
Details



SYMBOL AND STATE

Combinations of symbols *a*, *b* or *terminator* (*t*) and machine states 1 or 0 are represented by four-nucleotide sequences. Depending on how the five-nucleotide sequence TGGCT is cleaved into four nucleotides, for example, it will denote symbol *a* and a state of either 1 or 0.

Details



AUTONOMOUS COMPUTATION

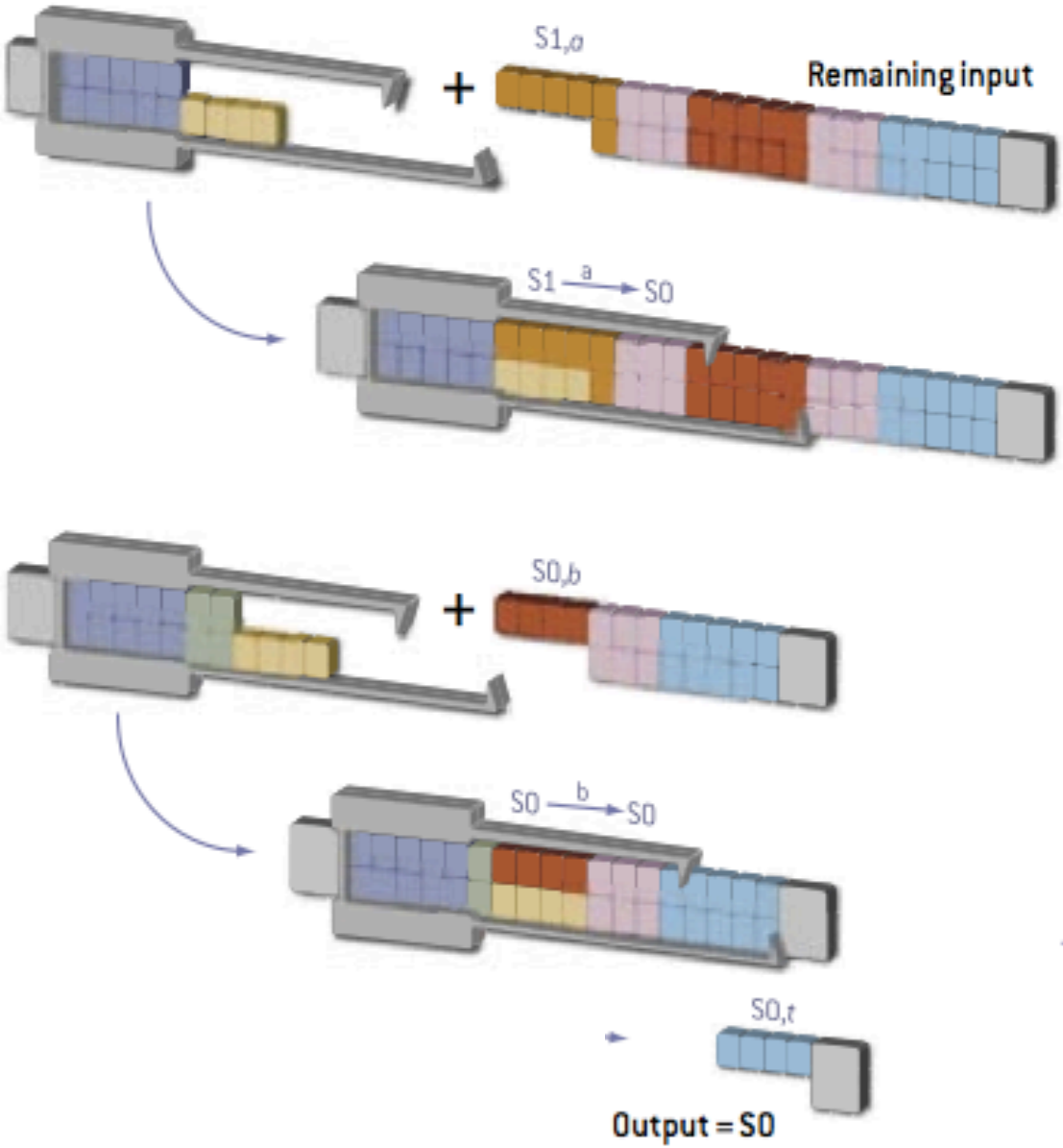
A hardware-software complex recognizes its complementary state/symbol combination on the input molecule. The molecules join to form a hardware-software-input complex, then *Fok*I cleaves the input molecule to expose the next symbol.

Details

A new hardware-software complex recognizes the next state and symbol on what remains of the input molecule.

Reactions continue until no rule applies or the terminator symbol is revealed.

In this example, computational cleavages leading to the final output (*far right*) have produced a four-nucleotide terminator symbol indicating a machine state of 0, the calculation's result.



More DNA Computing

- Concept has been extended to a “DNA doctor” computer
- mRNA serves as the disease indicator (mRNA of disease genes)
- Automaton starts the computation in a “yes” state and if all disease indicators are present it produces a “yes” state, if any mRNA components are missing it transitions to a “no” state
- Takes about 1 hour to complete calculation

DNA Doctor (Computer)

Nature. 2004 May 27;429(6990):423-9. Epub 2004 Apr 28.

An autonomous molecular computer for logical control of gene expression.

Benenson Y, Gil B, Ben-Dor U, Adar R, Shapiro E.

Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel.

Abstract

Early biomolecular computer research focused on laboratory-scale, human-operated computers for complex computational problems. Recently, simple molecular-scale autonomous programmable computers were demonstrated allowing both input and output information to be in molecular form. Such computers, using biological molecules as input data and biologically active molecules as outputs, could produce a system for 'logical' control of biological processes. Here we describe an autonomous biomolecular computer that, at least in vitro, logically analyses the levels of messenger RNA species, and in response produces a molecule capable of affecting levels of gene expression. The computer operates at a concentration of close to a trillion computers per microlitre and consists of three programmable modules: a computation module, that is, a stochastic molecular automaton; an input module, by which specific mRNA levels or point mutations regulate software molecule concentrations, and hence automaton transition probabilities; and an output module, capable of controlled release of a short single-stranded DNA molecule. This approach might be applied in vivo to biochemical sensing, genetic engineering and even medical diagnosis and treatment. As a proof of principle we programmed the computer to identify and analyse mRNA of disease-related genes associated with models of small-cell lung cancer and prostate cancer, and to produce a single-stranded DNA molecule modelled after an anticancer drug.

Comment in

Automata make antisense. [Nature. 2004]

PMID: 15116117 [PubMed - indexed for MEDLINE]

Are There Other Kinds of Molecular Computers?

Digital Information Encoding

Science. 2012 Sep 28;337(6102):1628. Epub 2012 Aug 16.

Next-generation digital information storage in DNA.

Church GM, Gao Y, Kosuri S.

Department of Genetics, Harvard Medical School, Boston, MA 02115, USA.

Abstract

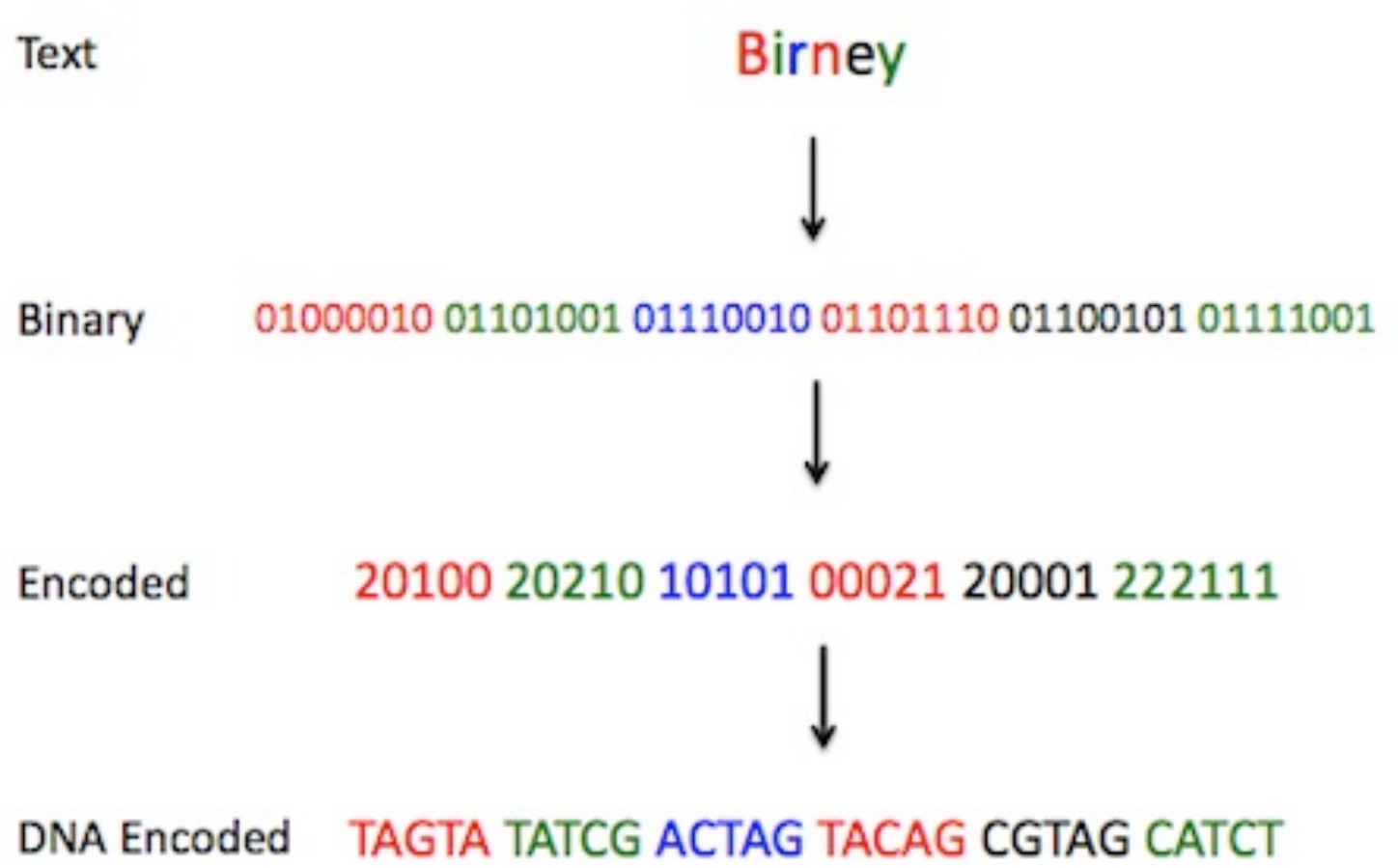
Digital information is accumulating at an astounding rate, straining our ability to store and archive it. DNA is among the most dense and stable information media known. The development of new technologies in both DNA synthesis and sequencing make DNA an increasingly feasible digital storage medium. We developed a strategy to encode arbitrary digital information in DNA, wrote a 5.27-megabit book using DNA microchips, and read the book by using next-generation DNA sequencing.

Comment in

To DNA, all information is equal. [Artif DNA PNA XNA. 2012]

PMID: 22903519 [PubMed - indexed for MEDLINE] **Free full text**

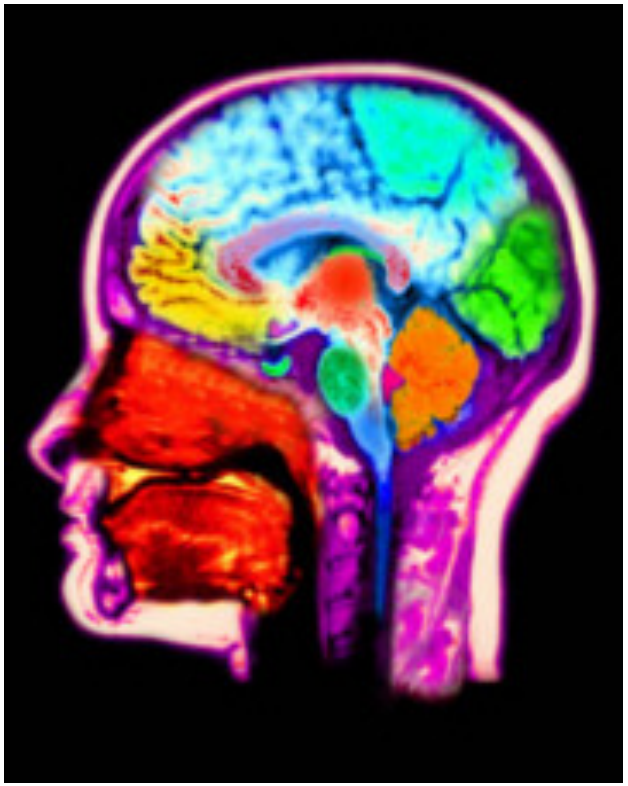
Digital Information Encoding



DNA Data Storage

- HTML draft of a 53,400 word book written by the lead researcher
- 11JPG images
- 1 JavaScript program
- All 154 of Shakespeare's sonnets
- 26 second audio clip of the "I Have a Dream" speech by Martin Luther King
- 5.5 petabits can be stored in each cubic millimeter of DNA
- $12,400 to encode data and $220 for retrieval (per Megabyte)

Other Molecular Computers: The Human Brain



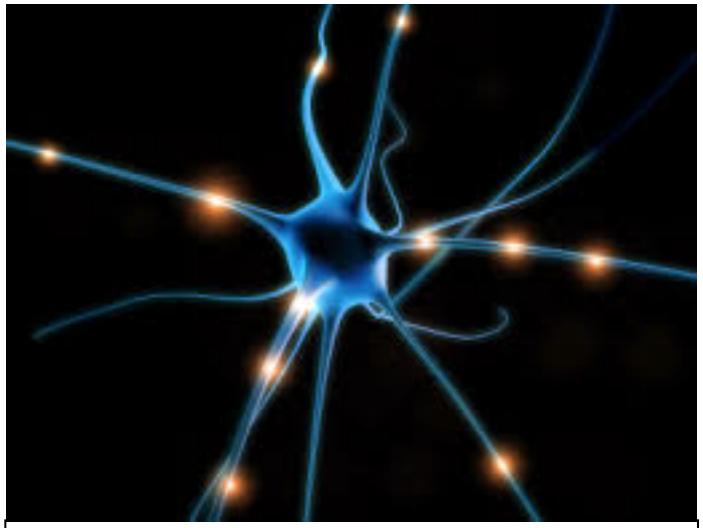
The world's most advanced molecular computer

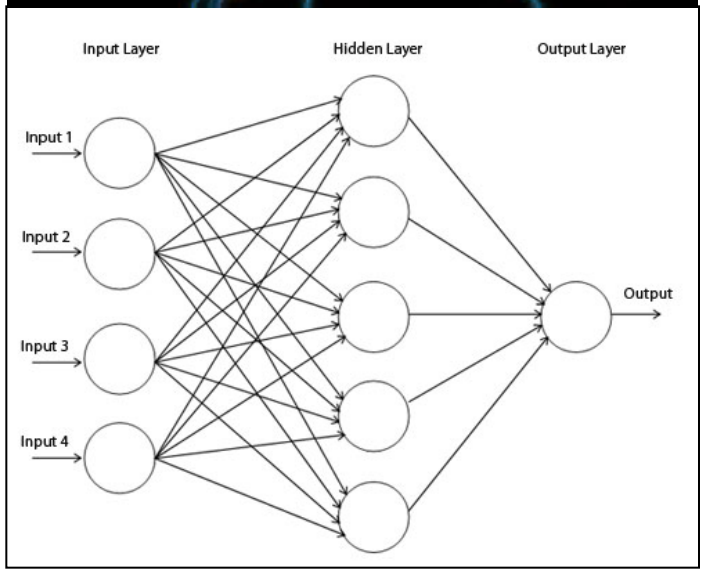
The Human Brain

- There are 10^{10} neurons in our brains
- There are roughly 10^{15} synapses operating at about 10 impulses/second (Biggest CPUs have 10^{9} transistors)
- Approximately 10^{16} synapse operations per second (Fastest super computers [TITAN] perform at 10^{16} FLOPS)
- Total energy consumption of the brain is about 25 watts (Blue Gene requires 1.5 Megawatts)

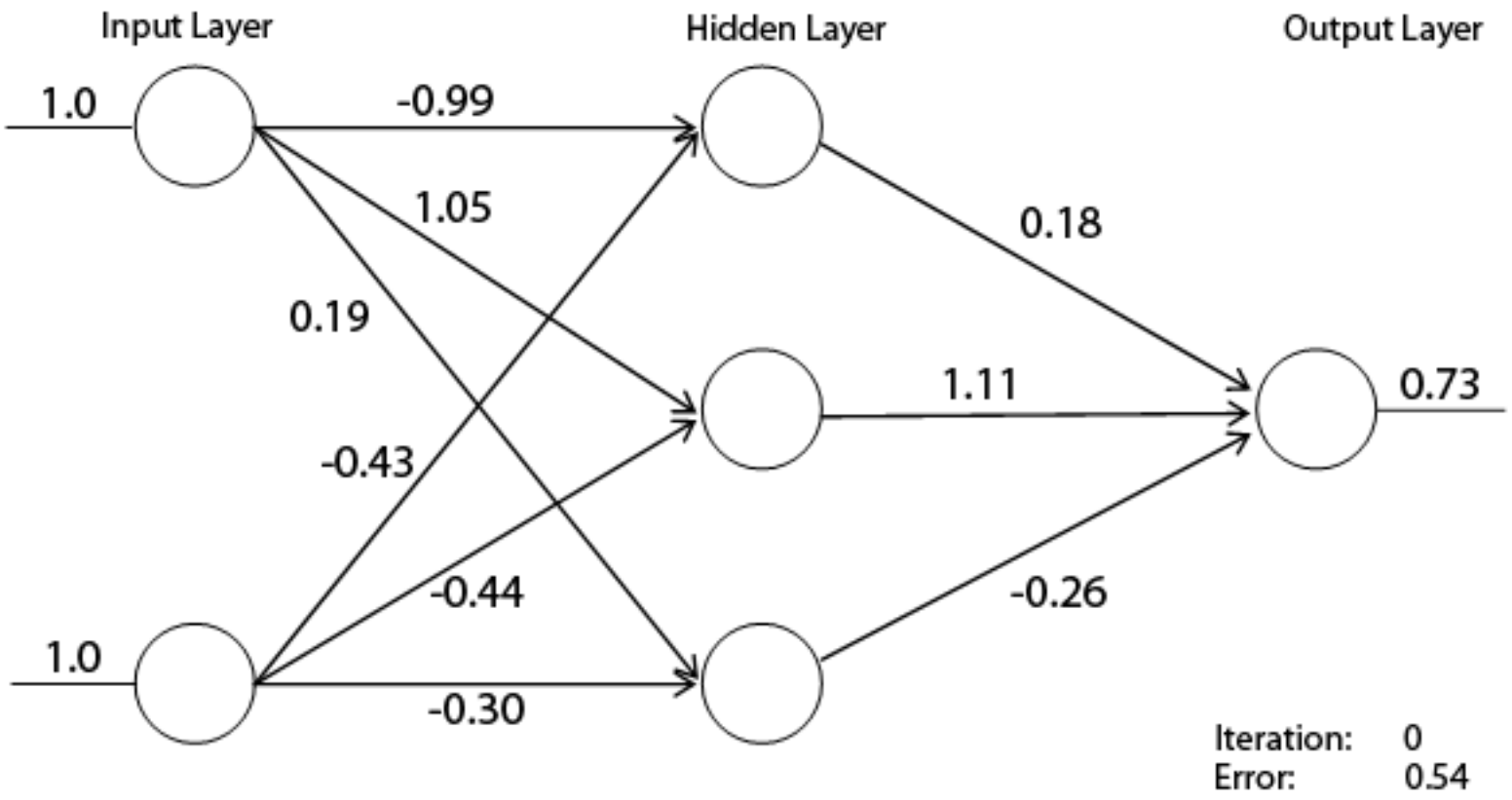
http://www.merkle.com/brainLimits.html

Neural Networks

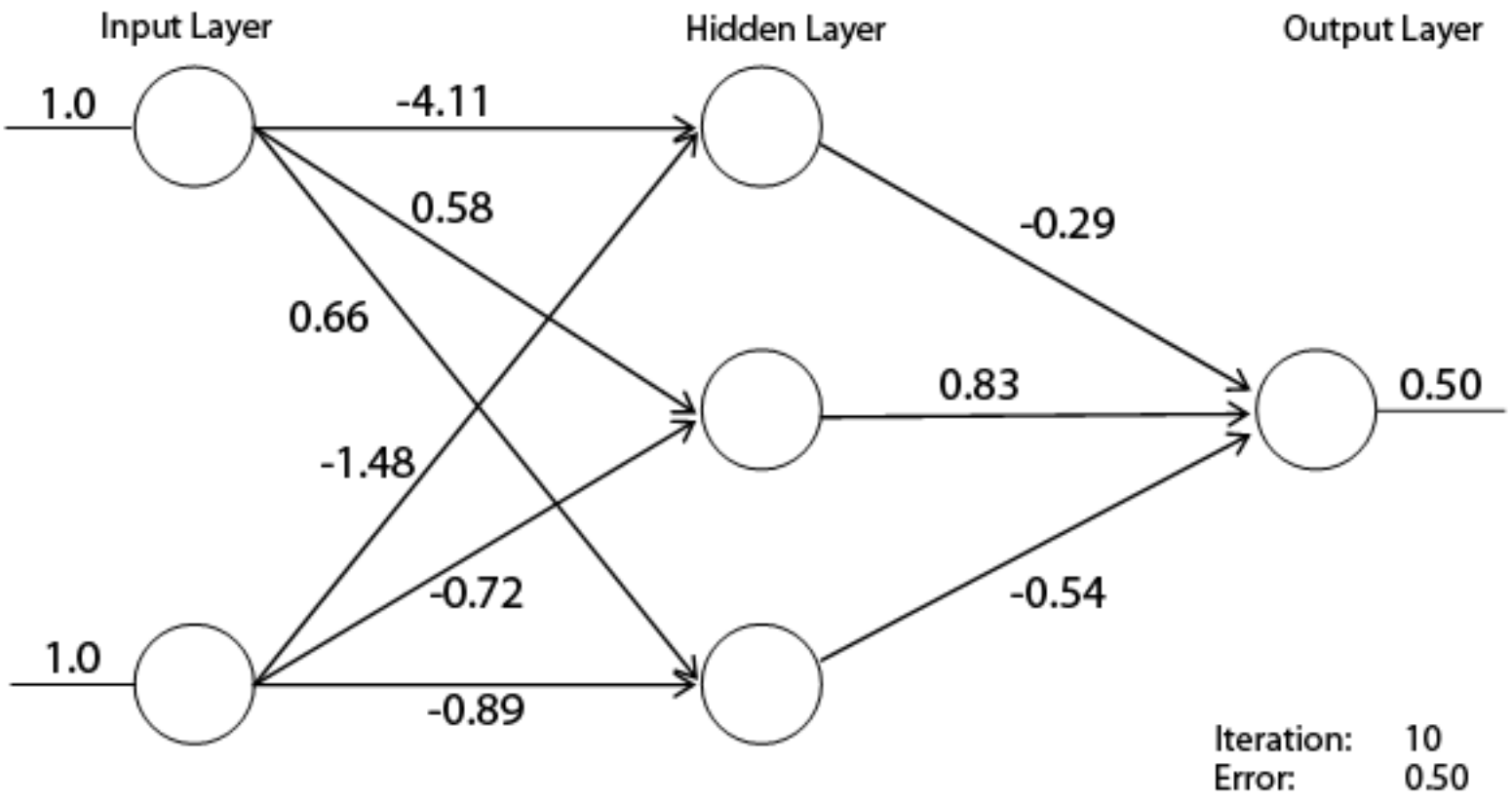




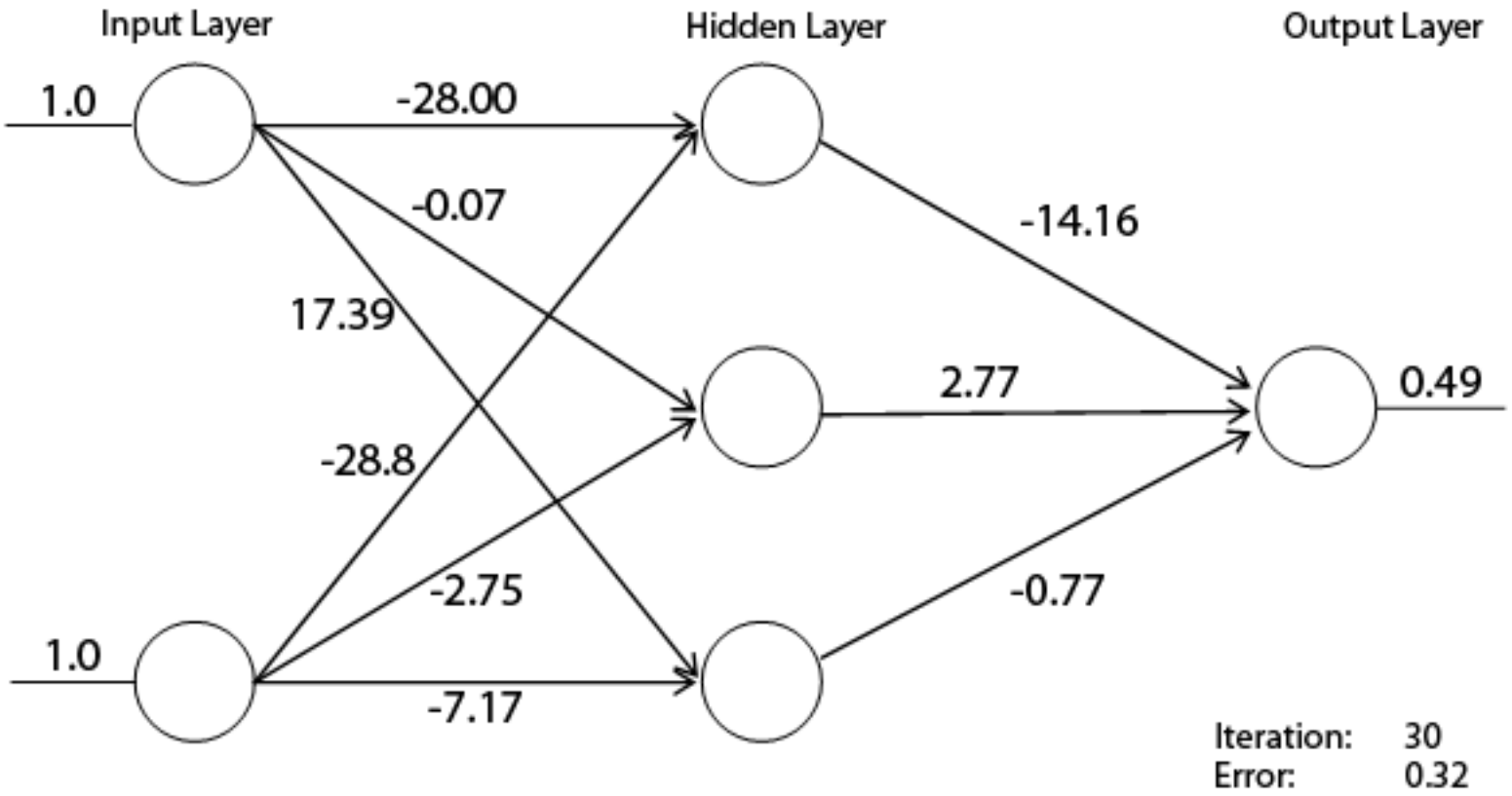
How We Learn


Input Layer
Hidden Layer
Output Layer
1.0
1.0
-0.99
1.05
0.19
-0.43
-0.44
-0.30
0.18
1.11
-0.26
0.73
Iteration: 0
Error: 0.54

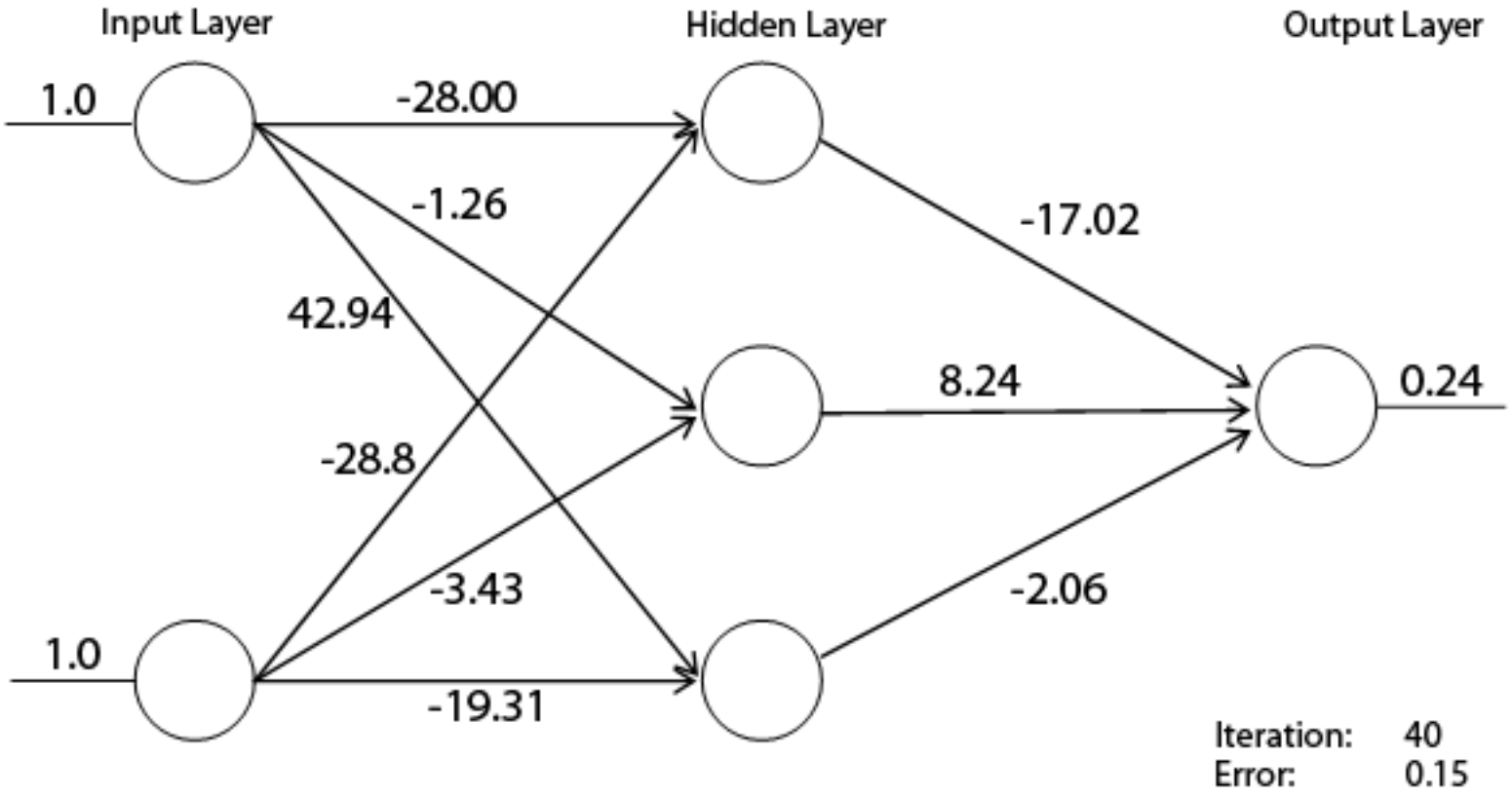
How We Learn


Input Layer
Hidden Layer
Output Layer
1.0
1.0
-4.11
0.58
0.66
-1.48
-0.72
-0.89
-0.29
0.83
-0.54
0.50
Iteration: 10
Error: 0.50

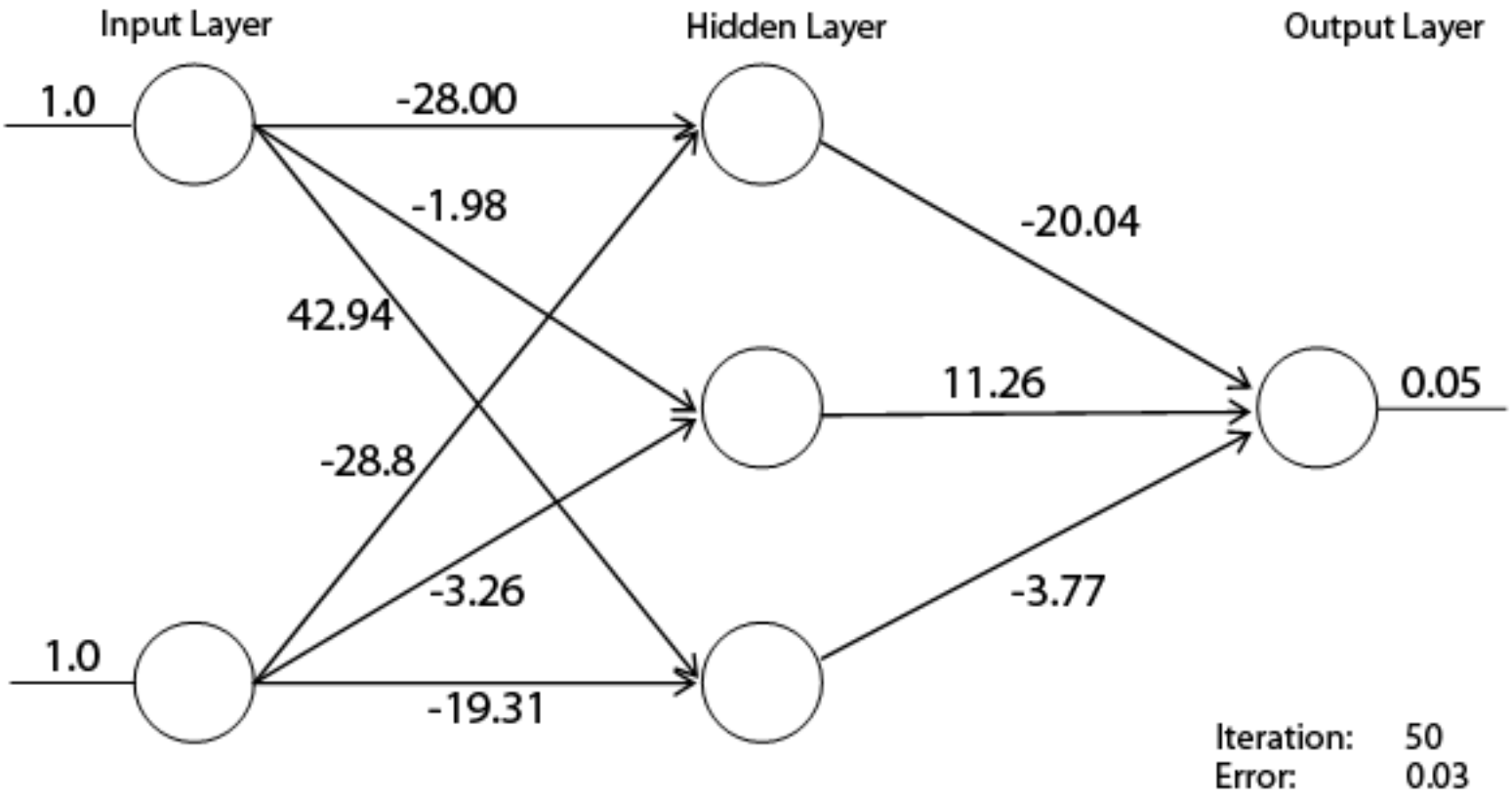
How We Learn


Input Layer
Hidden Layer
Output Layer
1.0
1.0
-28.00
-0.07
17.39
-28.8
-2.75
-7.17
-14.16
2.77
-0.77
0.49
Iteration: 30
Error: 0.32

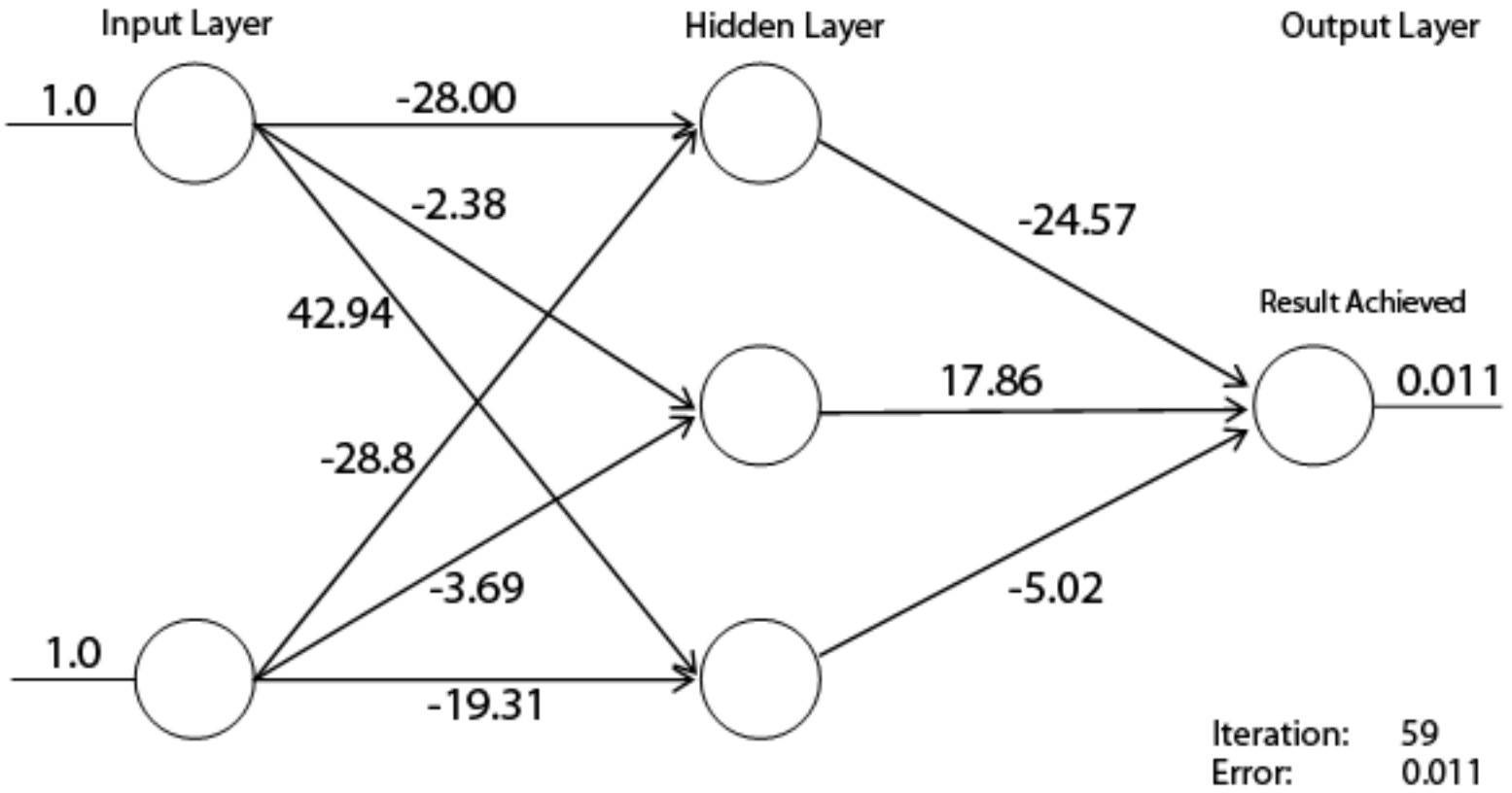
How We Learn


Input Layer
Hidden Layer
Output Layer
1.0
1.0
-28.00
-1.26
42.94
-28.8
-3.43
-19.31
-17.02
8.24
-2.06
0.24
Iteration: 40
Error: 0.15

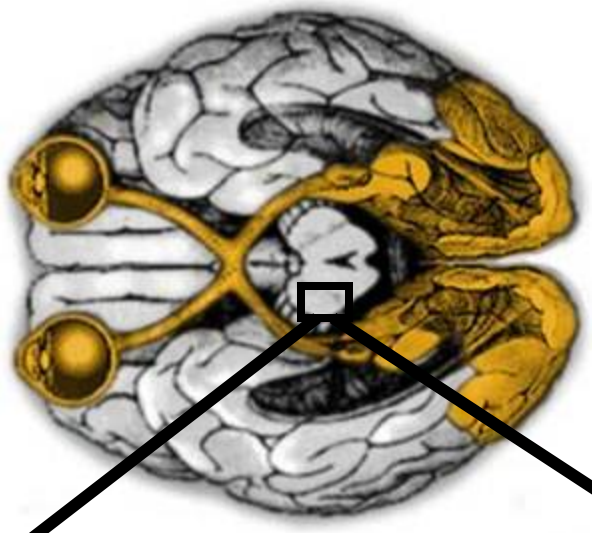
How We Learn

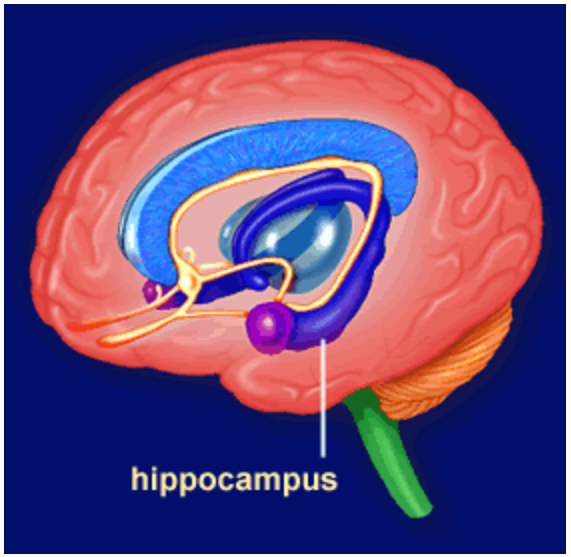

Input Layer
Hidden Layer
Output Layer
1.0
1.0
-28.00
-1.98
42.94
-28.8
-3.26
-19.31
-20.04
11.26
-3.77
0.05
Iteration: 50
Error: 0.03

How We Learn

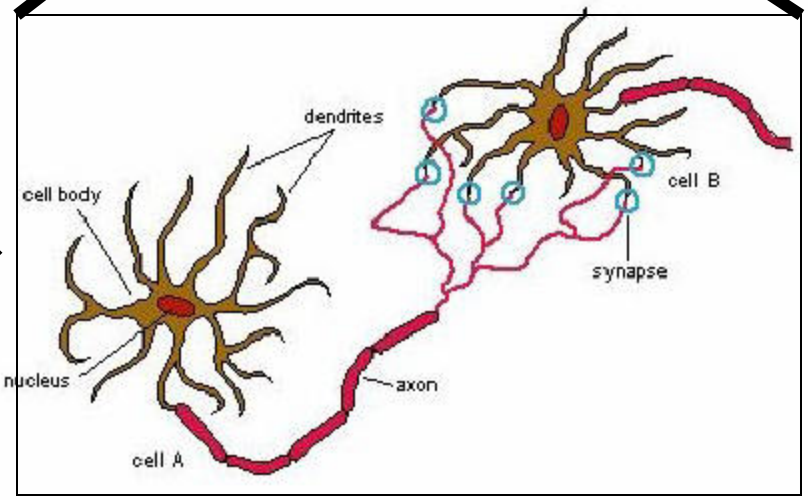

Input Layer
Hidden Layer
Output Layer
1.0
1.0
-28.00
-2.38
42.94
-28.8
-3.69
-19.31
-24.57
17.86
-5.02
Result Achieved
0.011
Iteration: 59
Error: 0.011

How Memories Are Formed





Emotional



Reinforcement

Why Should I Care?

- Considerable interest in molecular computing from theoretical computer scientists (graduate research opps.)
- Source for new algorithms (fast matrix multiplication techniques, solving bounded post correspondence problem, etc.) and new concepts in computing (data storage)
- Source for bio-inspired algorithms (genetic algorithms, neural nets, agent-based computing, machine learning, etc.)