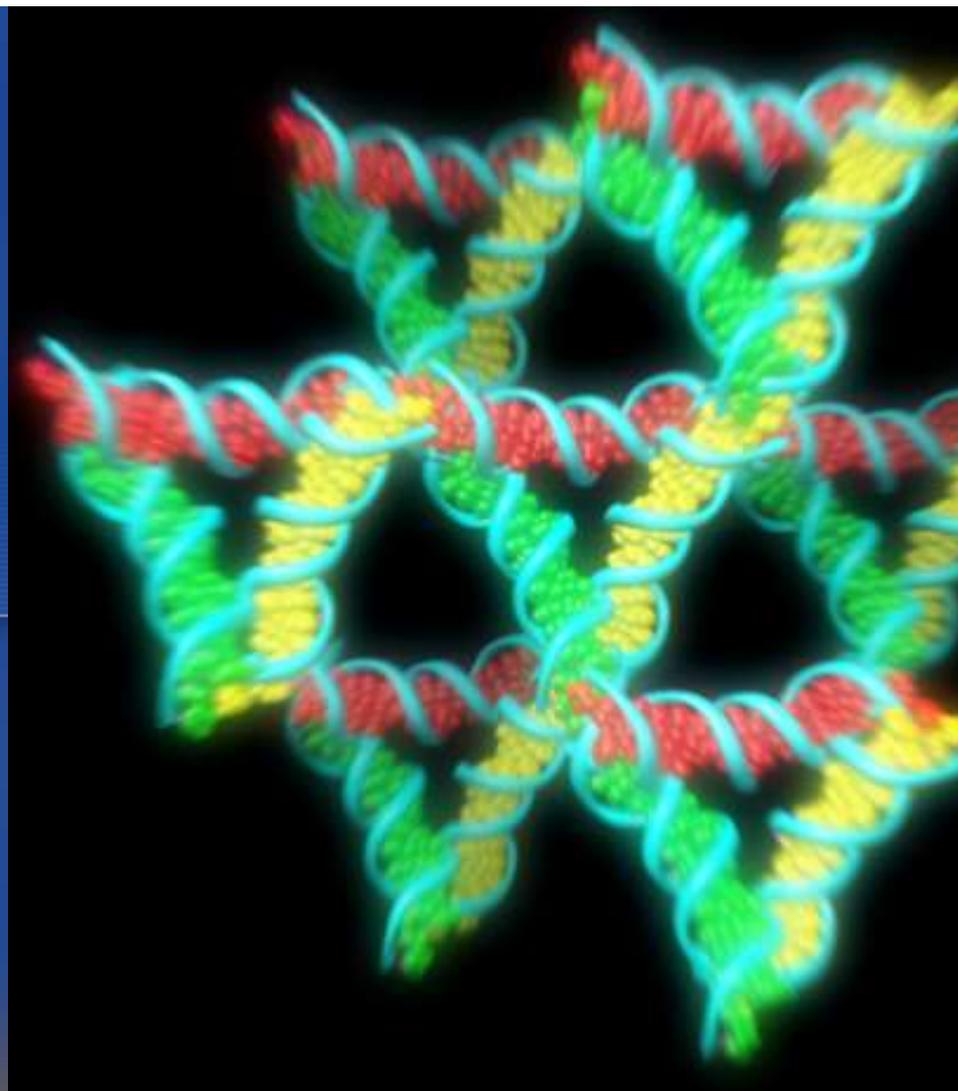


# Telling Molecules What to Do

Luca Cardelli

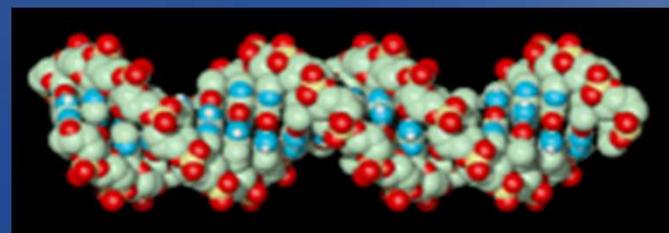
University of Oxford

2021-06-17, IMT Lucca



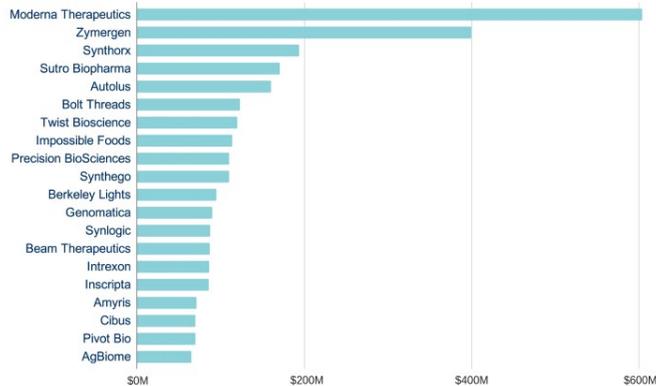
# Objectives

- The promises of Molecular Programming:
  - In Science & Medicine
  - In Engineering
  - In Computing
- The current practice of Molecular Programming
  - DNA technology
  - Molecular languages and tools
  - Molecular algorithms



# Synthetic Biology Market

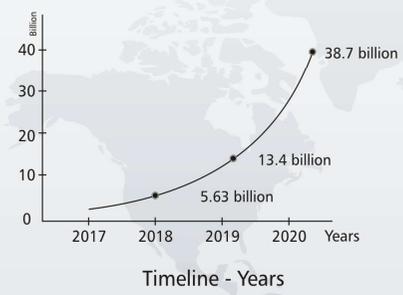
Top synthetic biology fundraisers of 2018 



## SYNTHETIC BIOLOGY TECHNOLOGY INNOVATIONS LANDSCAPE



### Synthetic Biology Market Size & Growth

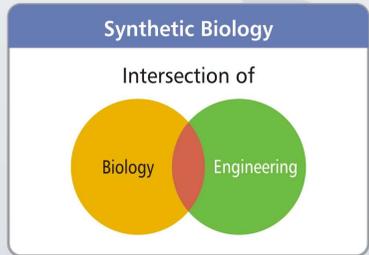


### Key Enabling Technologies

- ✓ DNA Synthesis
- ✓ Sequencing Technologies
- ✓ Genome Engineering
- ✓ Microfluidics Technologies
- ✓ Bioinformatics technologies
- ✓ Biological Components & integrated Systems
- ✓ Pathway engineering
- ✓ Biofuels Technologies

### Synthetic Biology : Core Market Segments

- DNA Synthesis
- Oligonucleotide Synthesis
- Pharmaceuticals
- Chemicals
- Biofuels
- Agriculture
- Synthetic DNA
- Synthetic Genes
- Synthetic Cells
- XNA
- Chassis organisms
- Enabling Products



For more information, Visit - <https://www.pintels.com> for email : [contactus@vajrasoftinc.com](mailto:contactus@vajrasoftinc.com)

# Some (ongoing) successes stories



- (\$4Bn) Reprogram a patient's own blood cells to recognise and destroy specific cancers.
- 90% remission in terminally ill leukemia patients



- (\$300M) Reprogram yeast to produce fine chemicals
- Antimalarial (in partnership with Sanofi)
- Jet fuel (in partnership with Total)

**Molecular "hacking"**



- Supply custom organisms for bio fabrication



- Grow meat, leather (\$100Bn market) in the lab
- Proofs of concept already in production

# Hacking Yoghurt

Tuur van Balen - Hacking Yoghurt  
- genetically modify your yoghurt in your own kitchen



<https://www.youtube.com/watch?v=Co8NOnErrPU>

# Live Clothing

Scientists Sew Genetically Modified E. Coli into Living Clothing



Harnessing the hygroscopic and biofluorescent behaviors of genetically tractable microbial cells to design biohybrid wearables

Wen Wang<sup>1,2</sup>, Lining Yao<sup>2</sup>, Chin-Yi Cheng<sup>2,3</sup>, Teng Zhang<sup>4</sup>, Hiroshi Atsumi<sup>5</sup>, Luda Wang<sup>4</sup>, Guanyun Wang<sup>2</sup>, Oksana Anilionyte...

+ See all authors and affiliations

# Molecular Programming

*A technology* (and theory of computation)  
based on information-bearing molecules  
of historically biological origin (DNA/RNA)  
non necessarily involving living matter

# Molecular Programming: The Hardware Aspect

Smaller and smaller things can be built

# Smaller and Smaller Very few Moore's cycles left!

First working transistor

John Bardeen and Walter Brattain, Dec. 23, 1947

First integrated circuit

Jack Kilby, Sep. 1958.

**50+ years later**

Jan 2010 25nm NAND flash

Intel&Micron. ~50atoms

Jun 2018 7nm (54nm pitch)

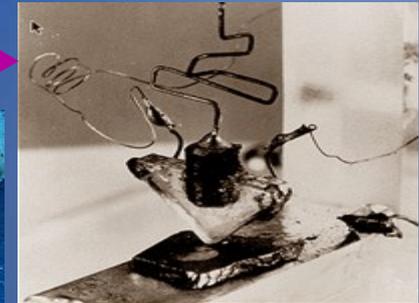
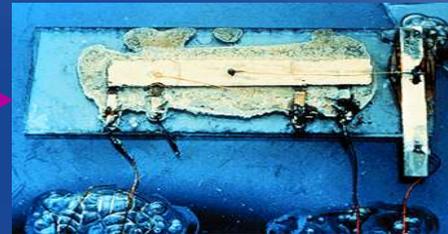
TSMC, Intel, Samsung, GlobalFoundries - mass production

Single molecule transistor

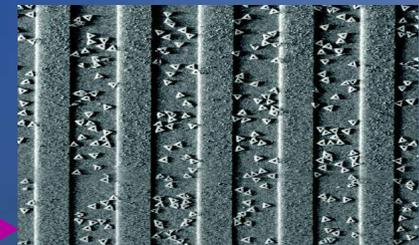
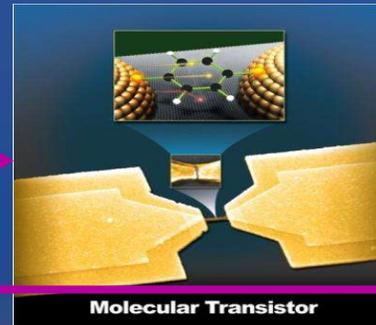
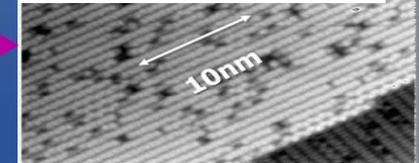
Observation of molecular orbital gating

*Nature*, 2009; 462 (7276): 1039

Molecules on a chip



Scanning tunneling microscope image of a silicon surface: 10nm is ~20 atoms (in cubic lattice)



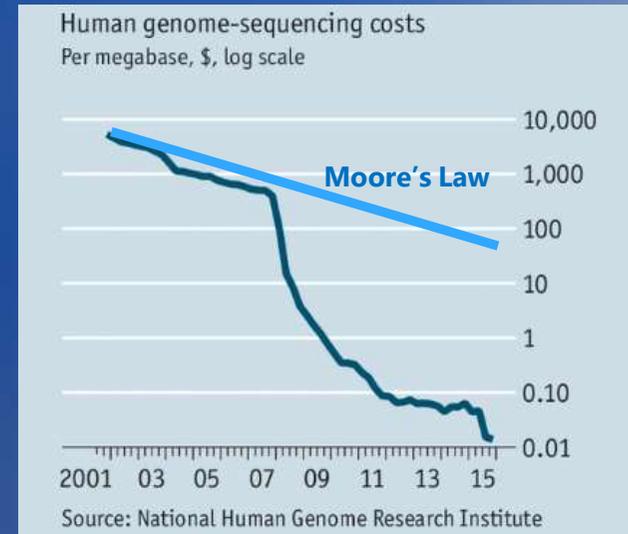
Placement and orientation of individual DNA shapes on lithographically patterned surfaces. *Nature Nanotechnology* 4, 557 - 561 (2009).

# Race to the Bottom

Moore's Law is approaching the single-molecule limit

Carlson's Curve is the new exponential growth curve in technology

In both cases, we are now down to *molecules*



**The Pace and Proliferation of Biological Technologies**  
March 4, 2004 by Rob Carlson

Waiter! There is fly DNA in my soup!

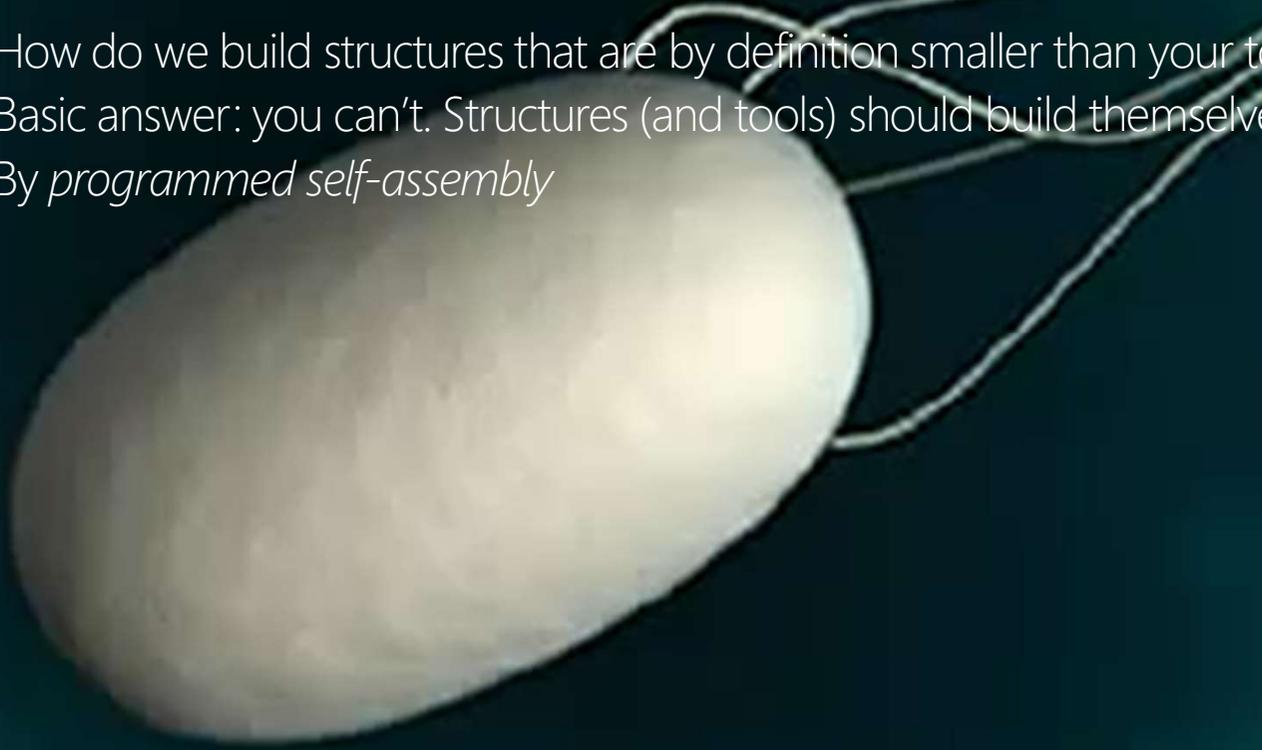
**The SmidgION: A portable DNA sequencer that runs on an iPhone**

Oxford Nanopore



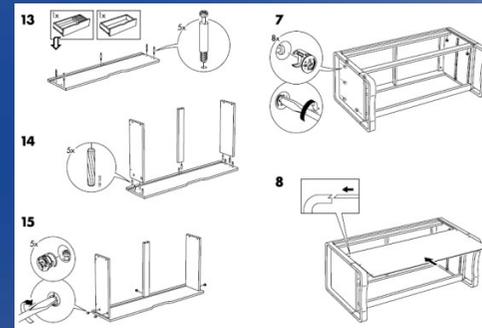
# Building the *Smallest Things*

- How do we build structures that are by definition smaller than your tools?
- Basic answer: you can't. Structures (and tools) should build themselves!
- By *programmed self-assembly*

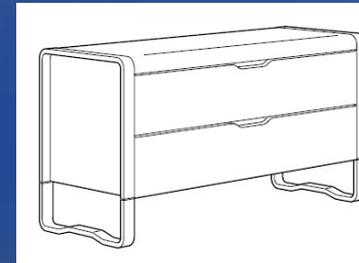


# Molecular IKEA

- Nature can self-assemble.  
Can we?
- "Dear IKEA, please send me a chest of drawers that assembles itself."
- We need a magical material where the pieces are pre-programmed to fit into to each other.
- At the molecular scale many such materials exist...



↓ Add water



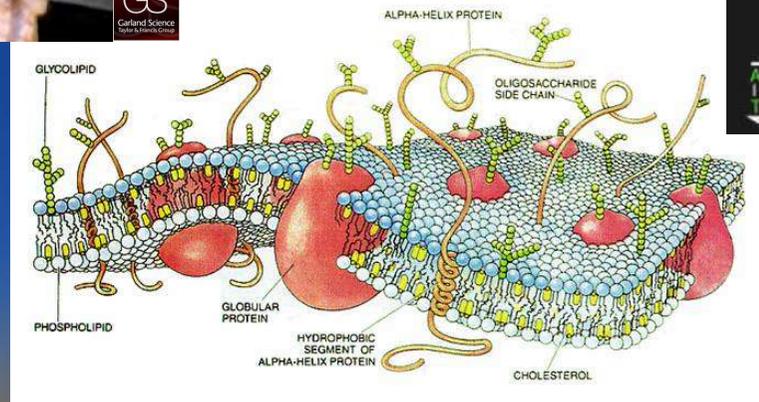
[http://www.ikea.com/ms/en\\_US/customer\\_service/assembly\\_instructions.html](http://www.ikea.com/ms/en_US/customer_service/assembly_instructions.html)

# Programmed Self-Assembly

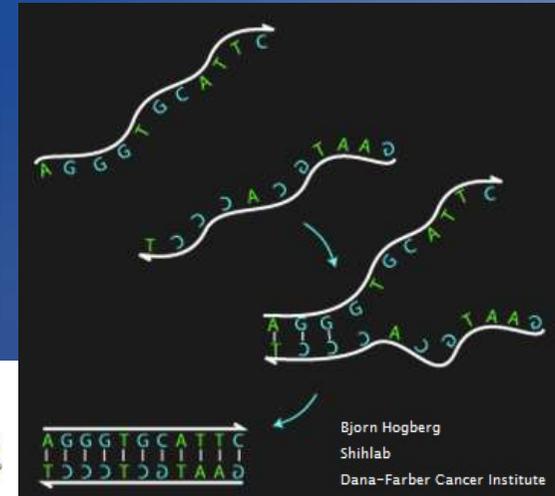
## Proteins



## Membranes



## DNA/RNA



# Molecular Programming: The Software Aspect

Smaller and smaller things can be programmed

# We can program...

- Information
  - Completely!



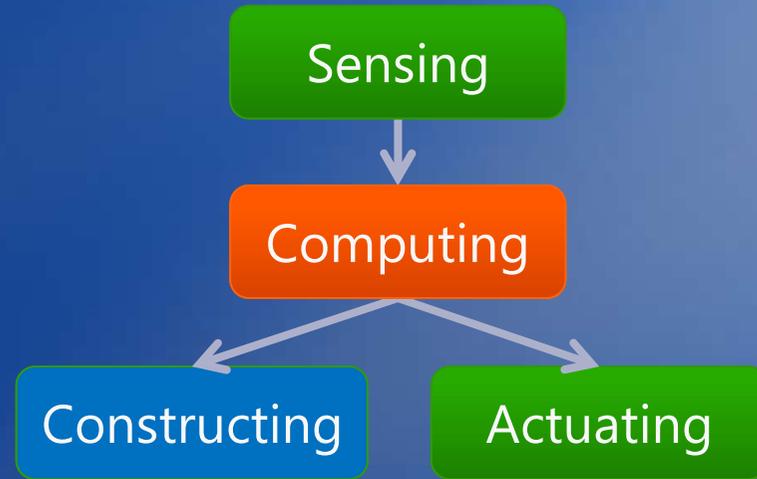
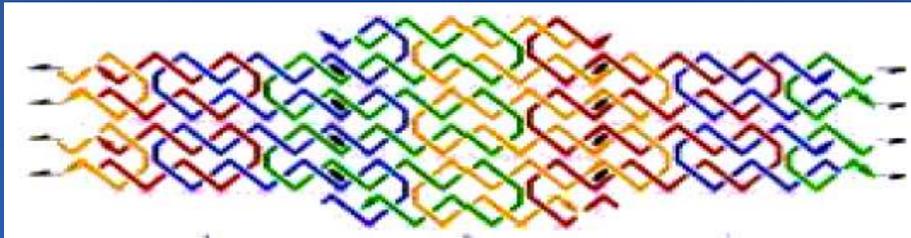
# We can program...

- Forces
  - Completely!  
(Modulo sensors/actuators)



# We can program...

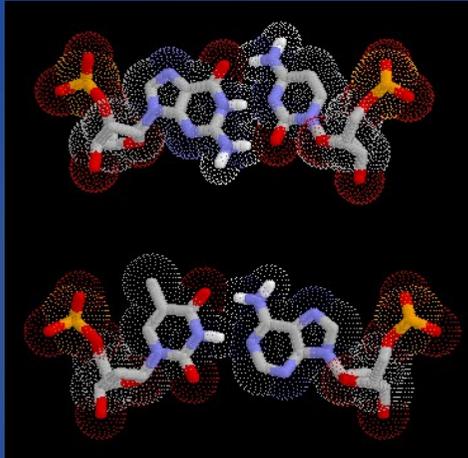
- Matter
  - Completely and directly! By self-assembly.
  - Currently: only DNA/RNA.



*It's like a 3D printer without the printer!*  
[Andrew Hellington]

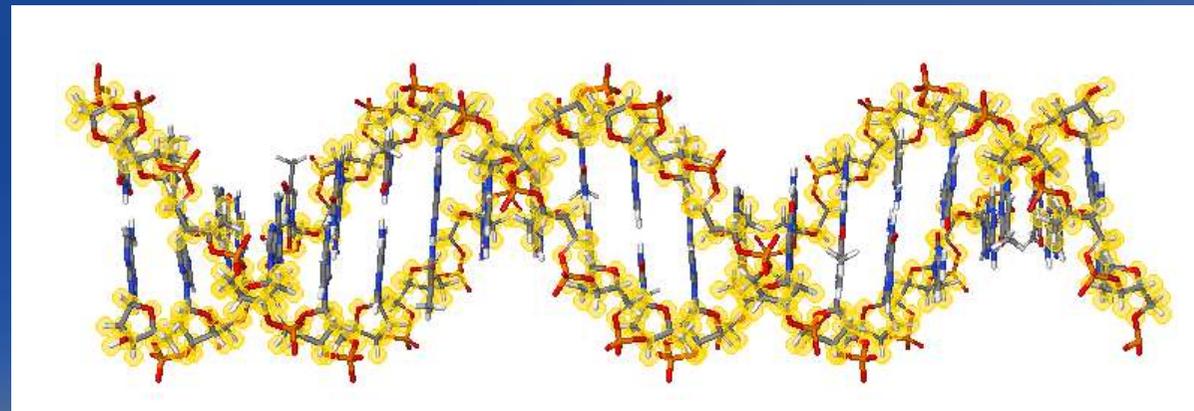
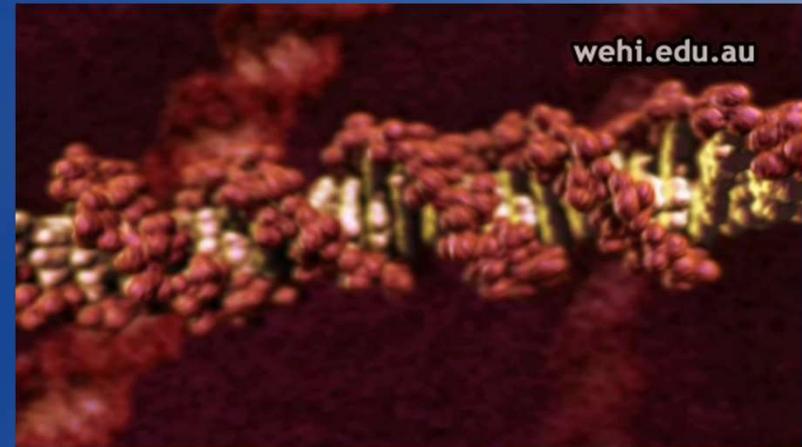
- But DNA is an amazing *material*

# DNA



G-C Base Pair  
Guanine-Cytosine

T-A Base Pair  
Thymine-Adenine



Sequence of Base Pairs (GACT alphabet)

[Interactive DNA Tutorial](http://www.biosciences.bham.ac.uk/labs/minchin/tutorials/dna.html)

(<http://www.biosciences.bham.ac.uk/labs/minchin/tutorials/dna.html>)

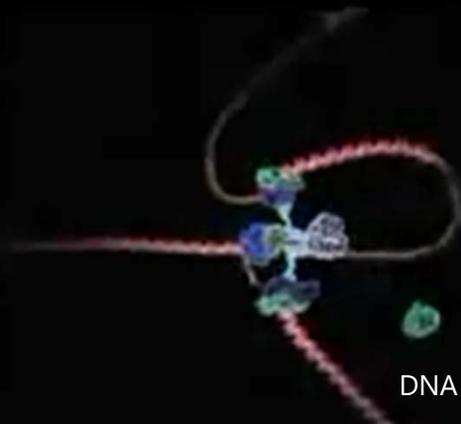
# DNA Specs

- DNA in each human cell
  - 3 billion base pairs
  - 2nm thick = 4 silicon atoms (in silicon lattice)!
  - 0.34nm per basepair = 2 bits in 2/3 silicon atom!
  - 2 meters long
    - copied in parallel at each cell division!
  - 750 megabytes
    - 80% functional, but only 1.5% protein coding
  - folded into a 6 $\mu$ m spherical nucleus
    - = 140 exabytes (million terabytes)/mm<sup>3</sup>
    - => all the data on the internet fits in a shoebox!
- DNA in each human body
  - 10 trillion cells
  - 133 Astronomical Units long
  - 7.5 octabytes (replicated)
- DNA in human population
  - 20 million light years long



Andromeda Galaxy  
2.5 million light years away

# DNA Benchmarks



DNA replication in *real time*

In Humans: 50 nucleotides/second  
Whole genome in a few hours (with parallel processing)

In Bacteria: 1000 nucleotides/second  
(higher error rate)



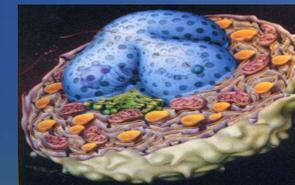
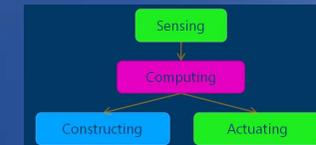
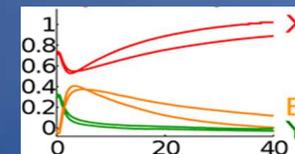
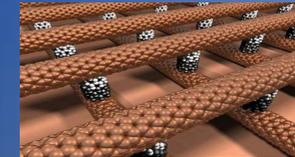
DNA transcription in *real time*

RNA polymerase II: 15-30 base/second

Drew Berry  
<http://www.wehi.edu.au/wehi-tv>

# One molecule to rule them all

- There are many, many nanofabrication techniques and materials
- But only DNA (and RNA) can:
  - Organize ANY other matter [caveats apply]
  - Execute ANY kinetics [caveats: up to time scaling]
  - Assemble Nano-Control Devices
  - Interface to Biology



H.Lodish & al. Molecular Cell Biology 4<sup>th</sup> ed.

# Building Nano-Control Devices

All the components of nanocontrollers can already be built entirely and solely with DNA, and interfaced to the environment



DNA Aptamers

Sensing

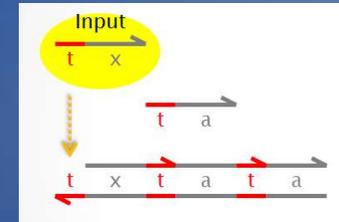
Computing

Constructing

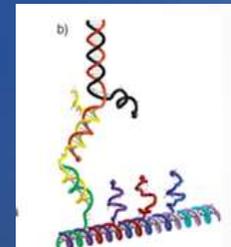
Actuating



Self-assembling DNA Tiles

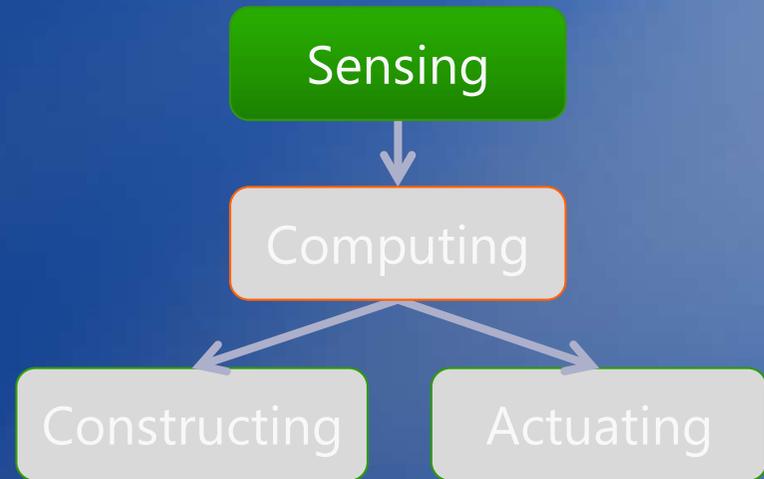


DNA Logical Gates



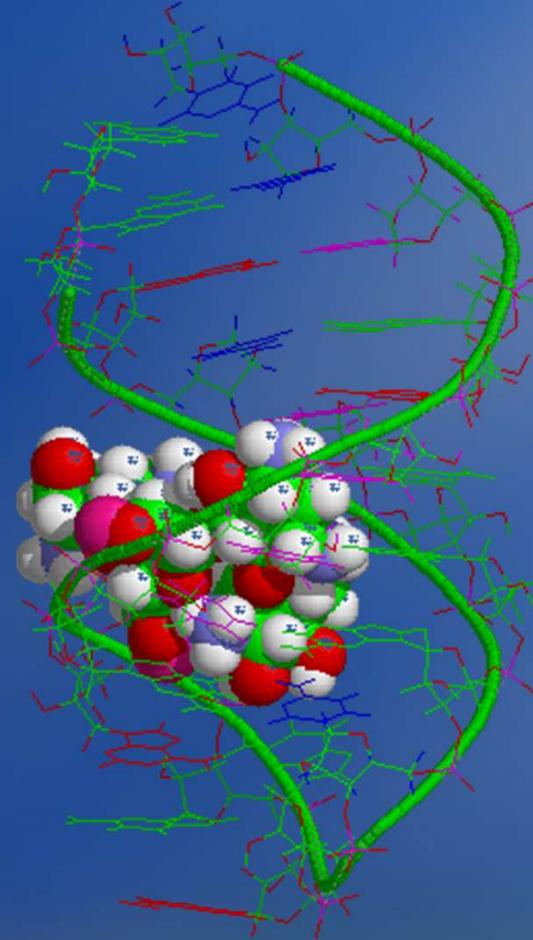
DNA Walkers & Cages

# Sensing



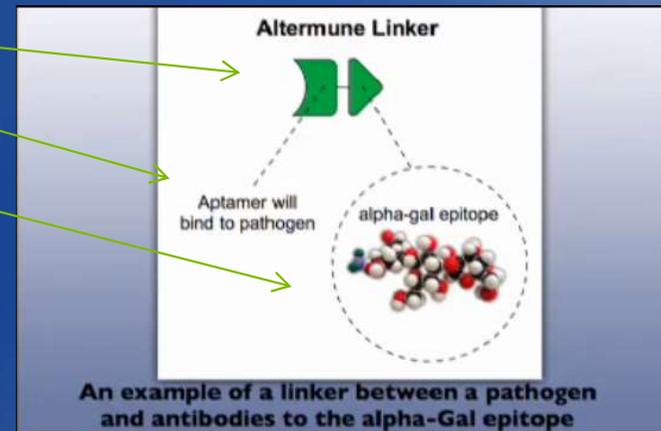
# Aptamers

Artificially evolved DNA molecules that stick to (almost) anything you like highly selectively

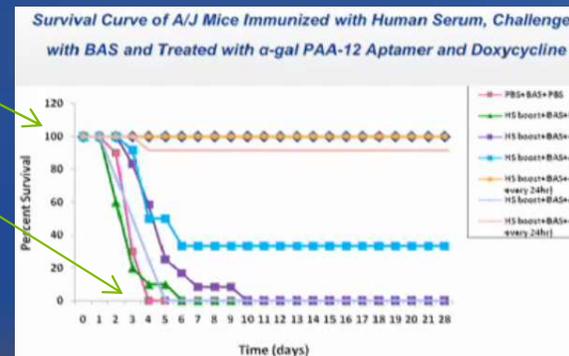


# Pathogen Spotlights

- DNA aptamer binds to:
  - A) a pathogen
  - B) a molecule our immune system (when allergic) hates and immediately removes (eats) along with anything attached to it!

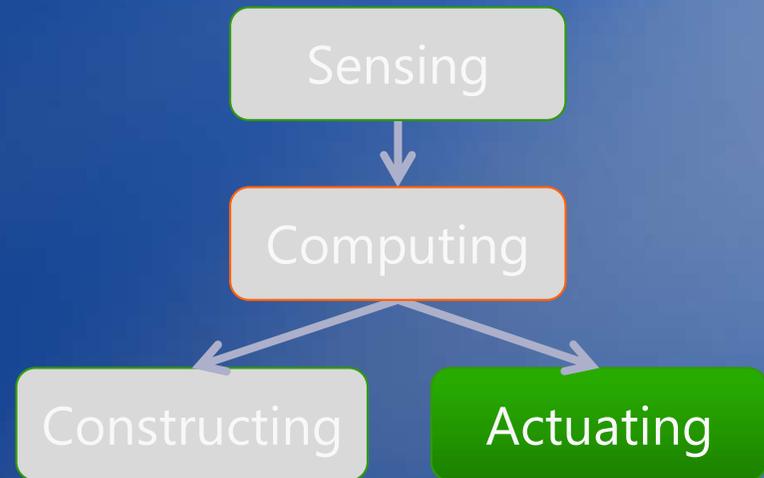


- Result: instant immunity
  - Mice poisoned with Anthrax plus aptamer (100% survival)
  - Mice poisoned with Anthrax (not so good)

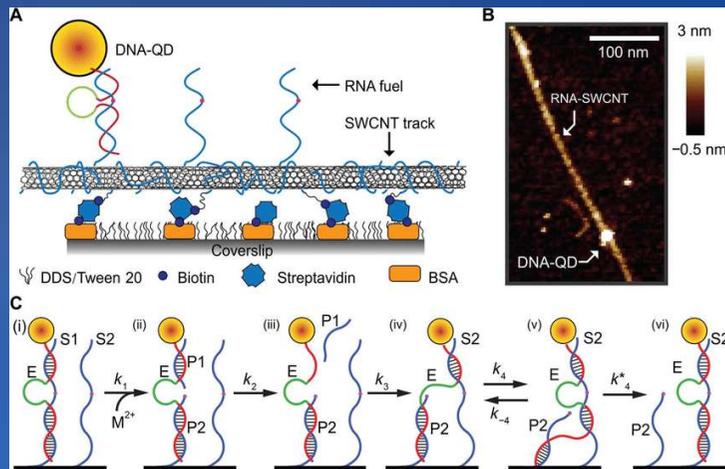
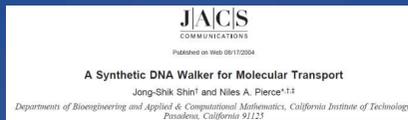


Kary Mullis (incidentally, also Nobel prize for inventing the Polymerase Chain Reaction)

# Actuating



# DNA Walkers



Visible/near-infrared subdiffraction imaging reveals the stochastic nature of DNA walkers

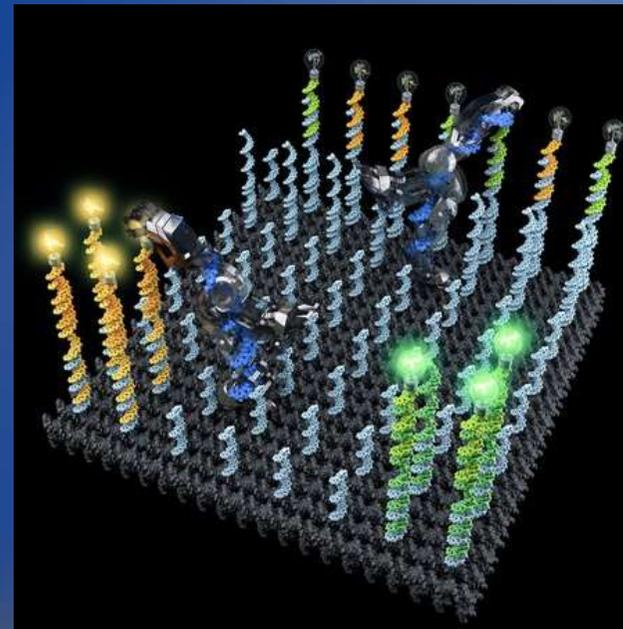
January 2017 · Science Advances 3(1):e1601600  
DOI: 10.1126/sciadv.1601600  
License: CC BY-NC 4.0

Jing Pan · Tae-Gon Cha · Feiran Li · Show all 6 authors · Jong Hyun Choi

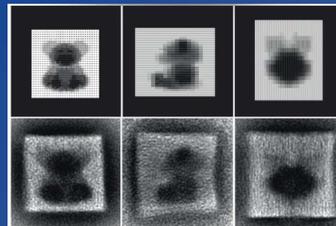
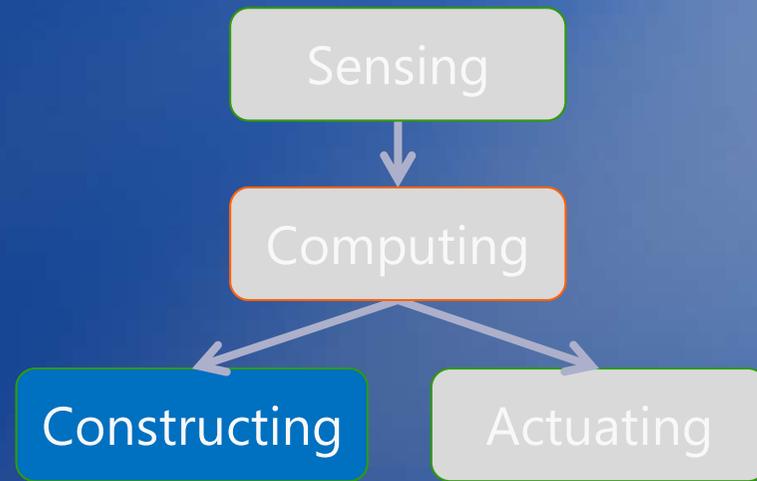
# DNA Robotics

## A cargo-sorting DNA robot

Thubagere, Anupama J. and Li, Wei and Johnson, Robert F. and Chen, Zibo and Doroudi, Shayan and Lee, Yae Lim and Izatt, Gregory and Wittman, Sarah and Srinivas, Niranjan and Woods, Damien and Winfree, Erik and Qian, Lulu (2017) A cargo-sorting DNA robot. *Science*, 357 (6356). Art. No. eaan6558. ISSN 0036-8075.

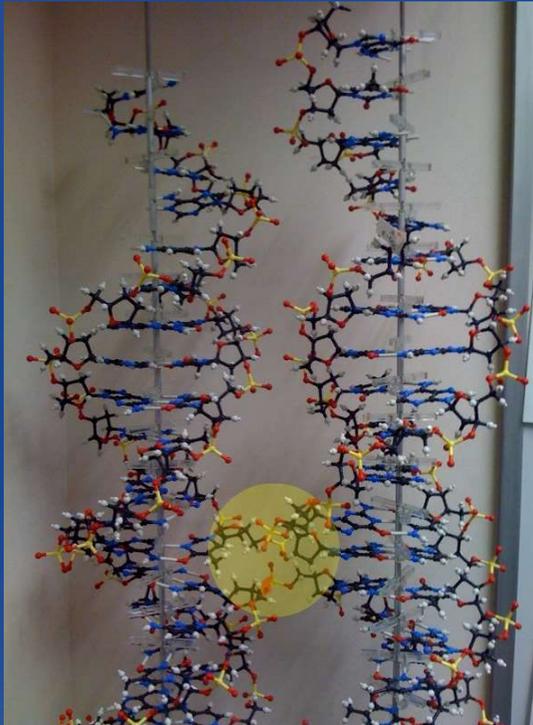
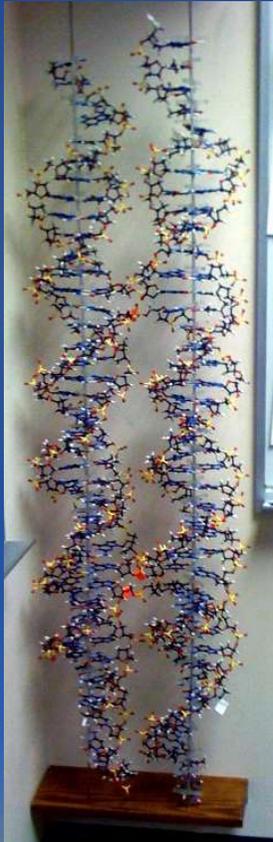


# Constructing



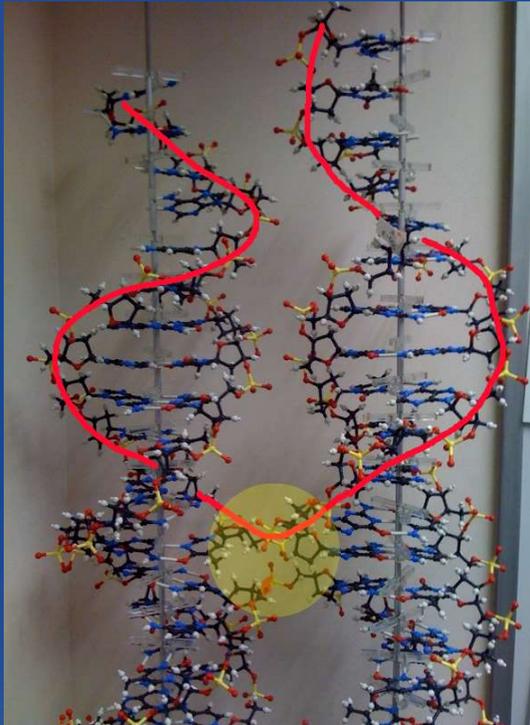
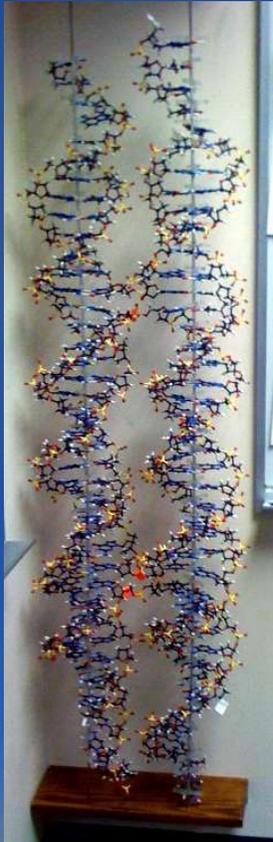
The 3D model of the computer-designed bear shape shown on top was fabricated into the nanostructures visualized with transmission electron microscopy (below). Credit: Wyss Institute at Harvard University

# Crosslinking



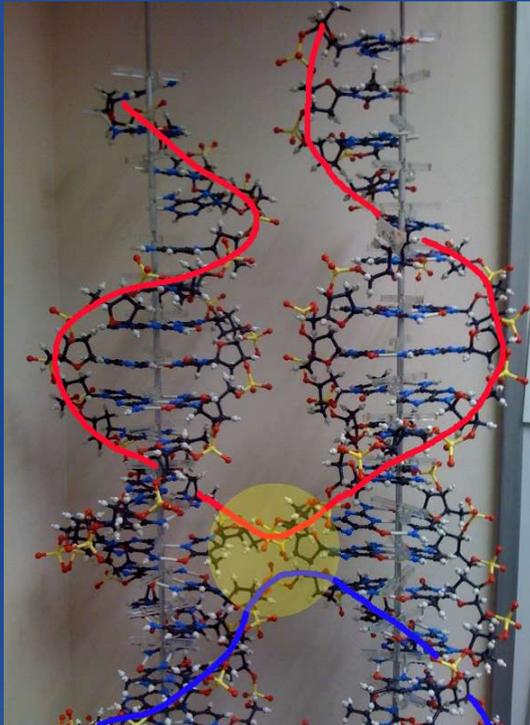
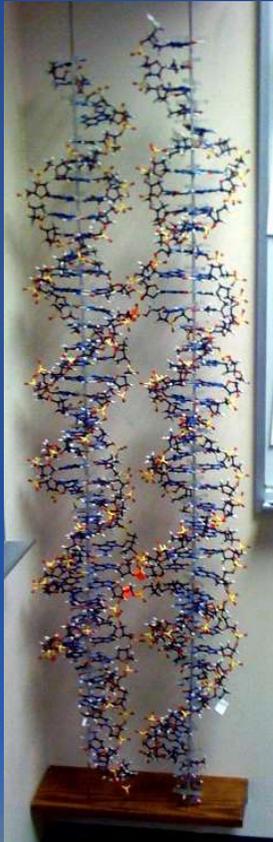
the dawn of  
structural DNA  
nanotechnology

# Crosslinking



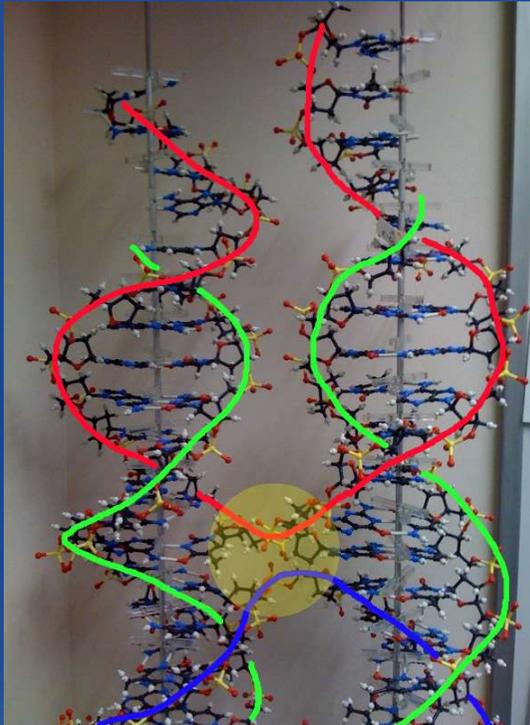
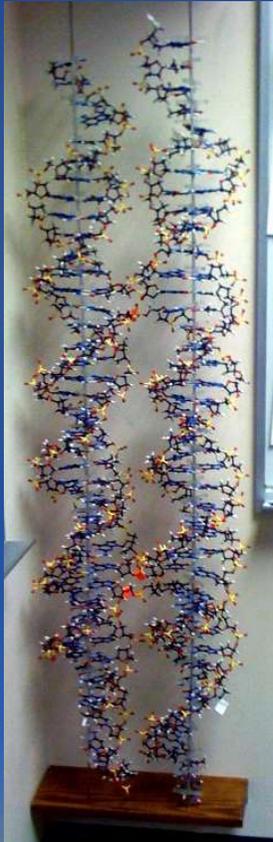
the dawn of  
structural DNA  
nanotechnology

# Crosslinking



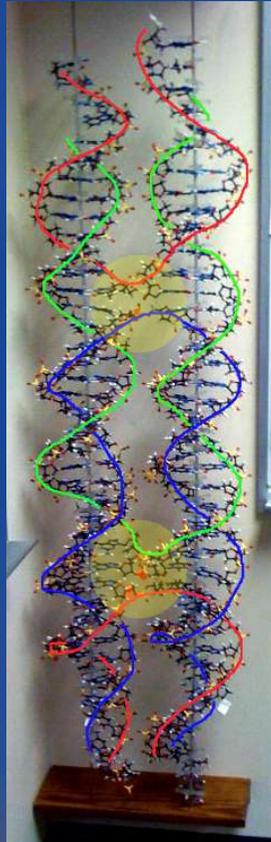
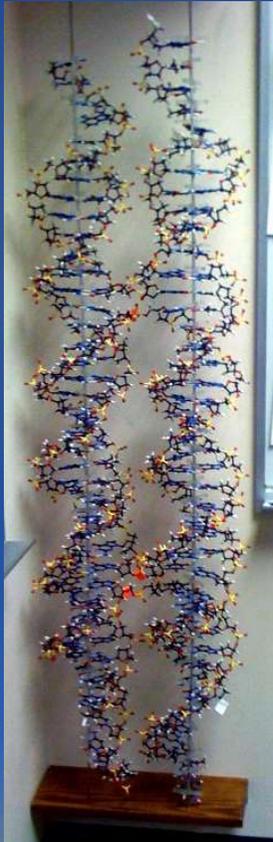
the dawn of  
structural DNA  
nanotechnology

# Crosslinking

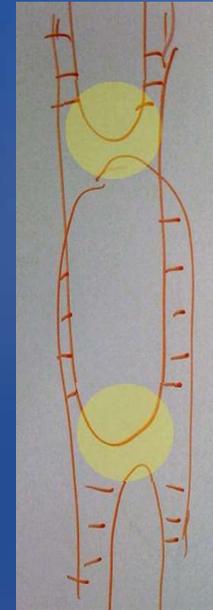


the dawn of  
structural DNA  
nanotechnology

# Crosslinking

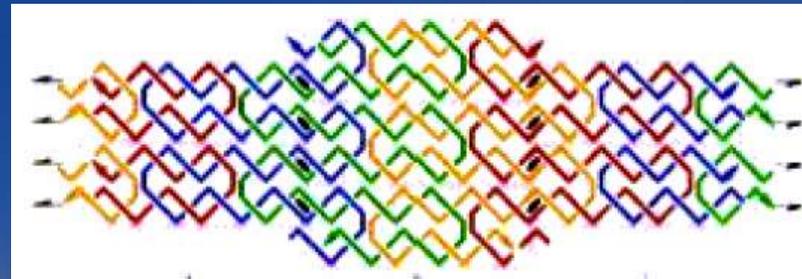
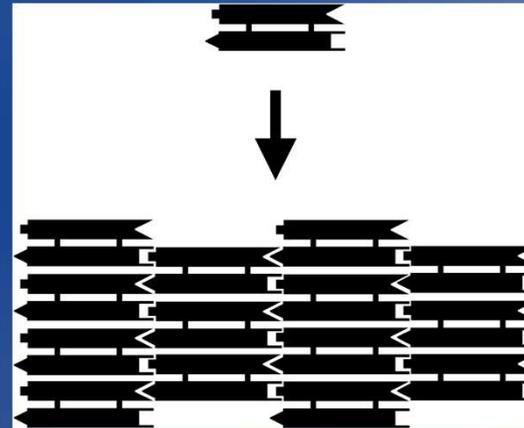
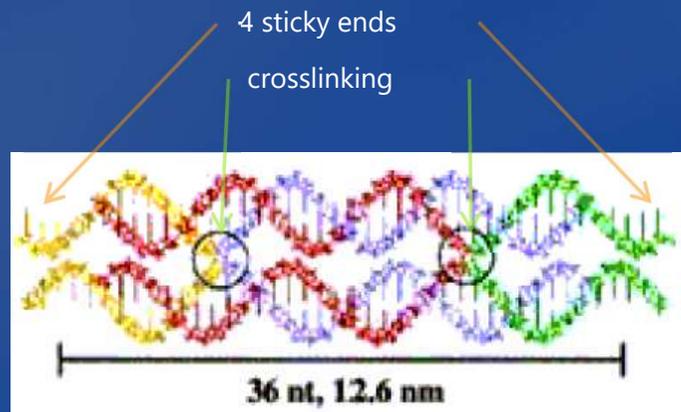


In nature, crosslinking is deadly (blocks DNA replication).



In engineering, crosslinking is the key to using DNA as a construction material.

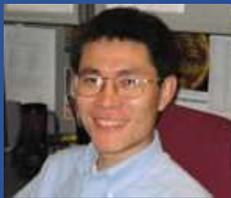
# DNA Tiling



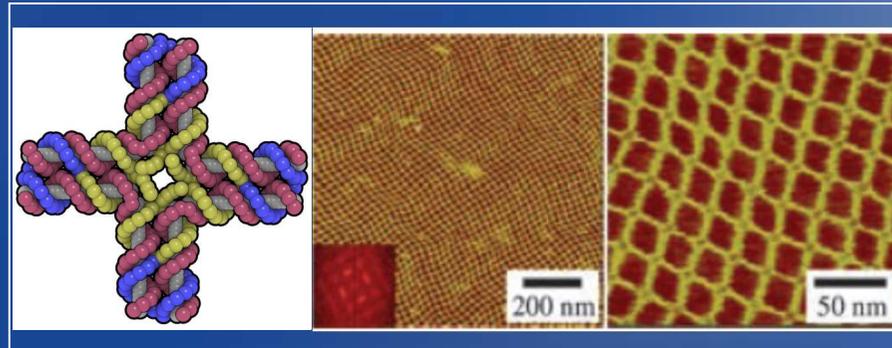
Construction and manipulation of DNA tiles in free space

Pankhudi

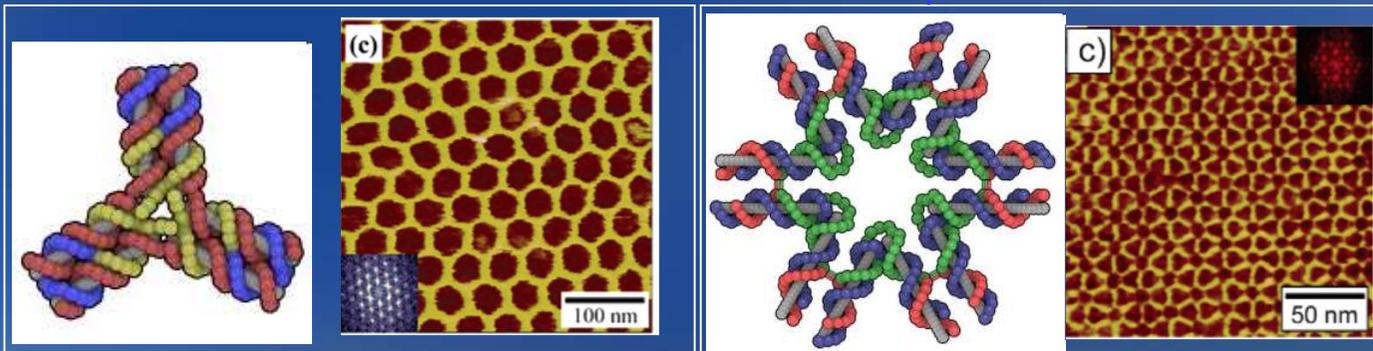
# 2D DNA Lattices



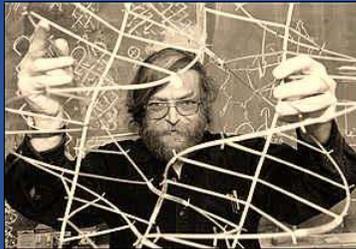
Chengde Mao  
Purdue University, USA



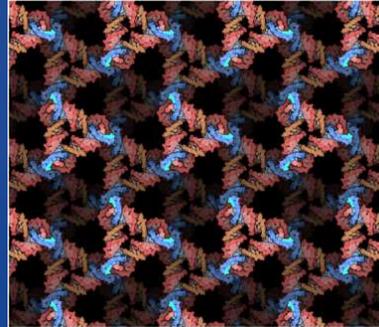
N-point Stars



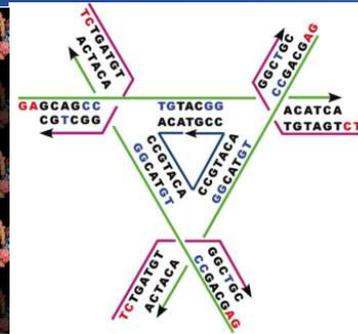
# 3D DNA Structures



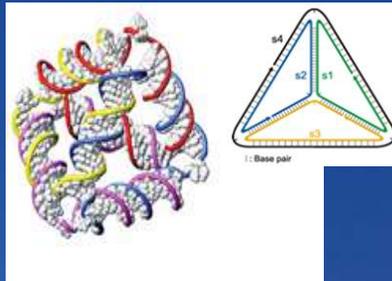
Ned Seeman  
NYU



3D Crystal



Andrew Tuberfield  
Oxford



Tetrahedron

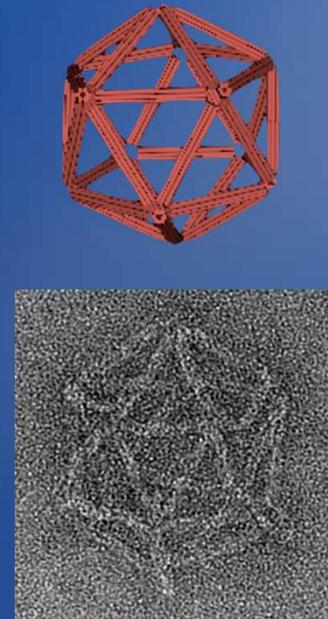
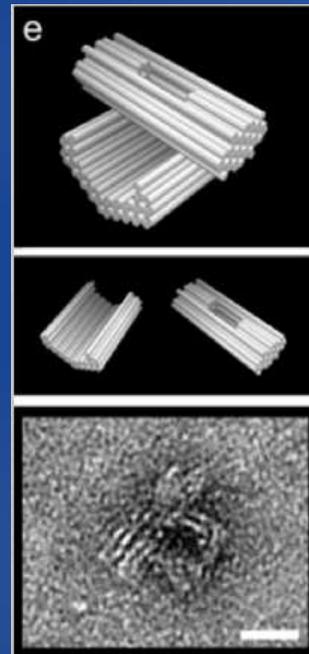
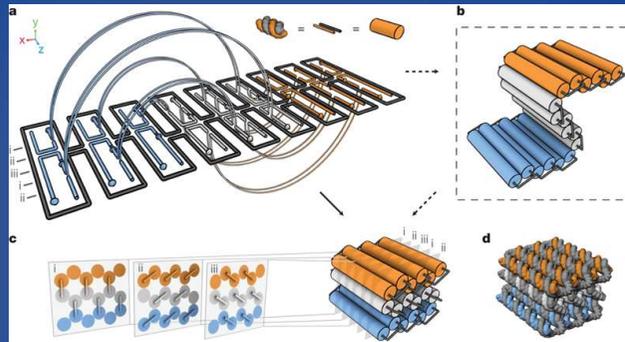


Friedrich Simmel  
Munich

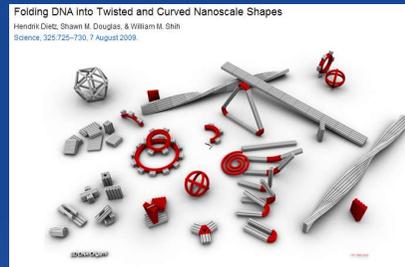


Robotic Arm

# CADnano



William Shih  
Harvard



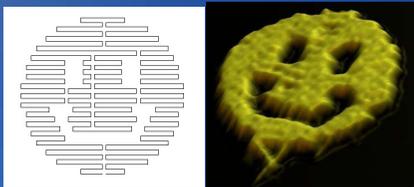
Folding DNA into Twisted and Curved Nanoscale Shapes  
Henrik Dietz, Shawn M. Douglas & William M. Shih  
Science, 329:725-730, 7 August 2009

<https://www.youtube.com/watch?v=Ek-FDPymygg>

S.M. Douglas, H. Dietz, T. Liedl, B. Högberg, F. Graf and W. M. Shih  
Self-assembly of DNA into nanoscale three-dimensional shapes, Nature (2009)

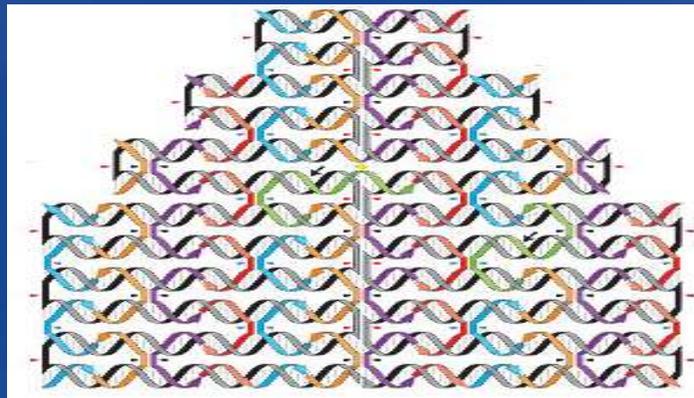
# DNA Origami

Folding a long (6407bp) naturally occurring circular ssDNA (from bacteriophage M13) via lots of short 'staple' strands that constrain its shape

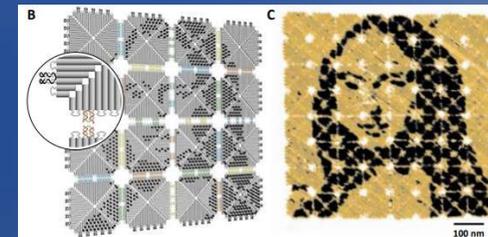


AFM image

Paul Rothemund's  
"Disc with three holes" (2006)  
*Nature* 440, 297, 2006



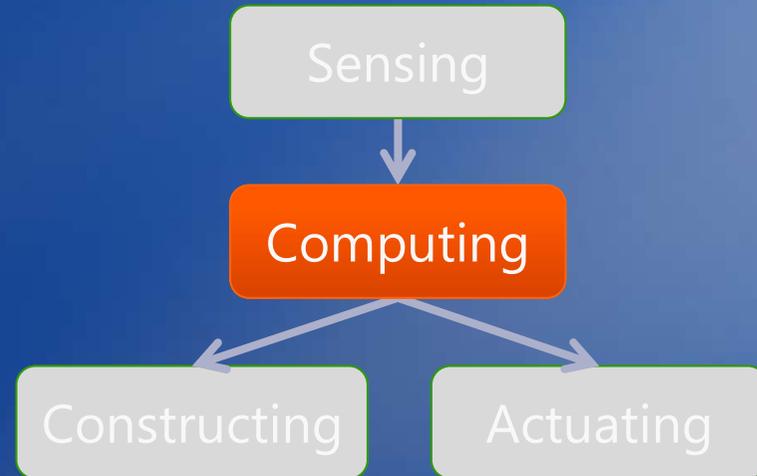
Black/gray: 1 long viral strand (natural DNA)  
Color: many short staple strands (synthetic DNA)



AFM image

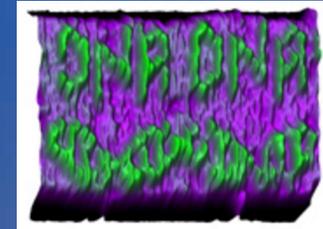
Lulu Qian's  
Hierarchical assembly (2017)  
*Nature*, 552(7683):67–71, 2017

# Computing



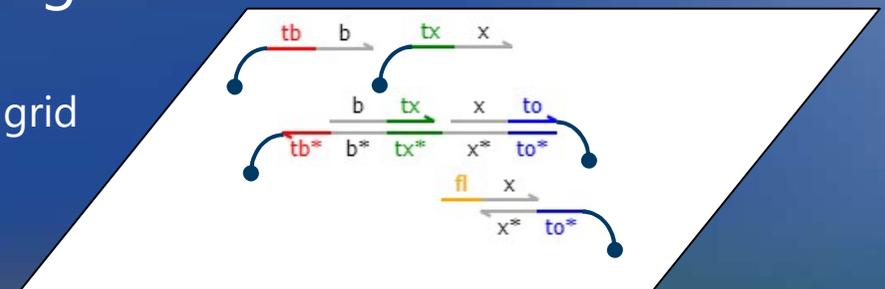
# DNA Circuit Boards

- DNA origami are arrays of uniquely-addressable locations
  - Each staple is different and binds to a unique location on the origami
  - It can be extended with a unique sequence so that something else will attach uniquely to it.



Some staples are attached to "green blobs" (as part of their synthesis) Other staples aren't

- More generally, we can bind "DNA gates" to specific locations
  - And so connect them into "DNA circuits" on a grid
  - Only neighboring gates will interact

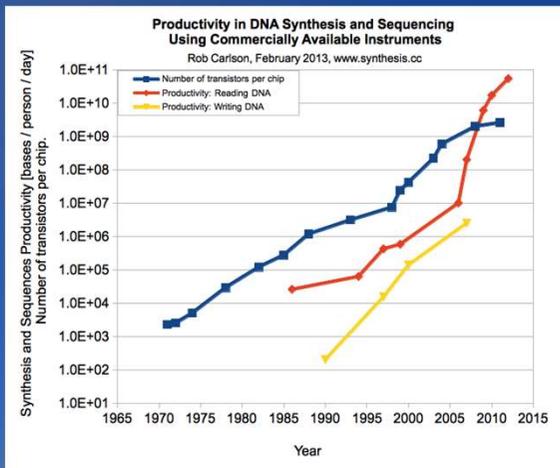


# DNA Storage (Read/Write)

Information-rich physical structures can be used for storage.

DNA has a data density of **140 exabytes** ( $1.4 \times 10^{20}$  bytes) per  $mm^3$  compared to state-of-the-art storage media that reaches ~500 megabytes ( $5 \times 10^8$  bytes) per  $mm^3$

DNA has been shown to be stable for millions of years

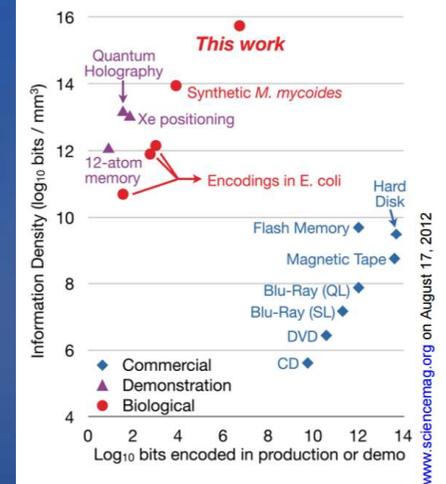


**The Pace and Proliferation of Biological Technologies**

March 4, 2004 by Rob Carlson

## Next-Generation Digital Information Storage in DNA

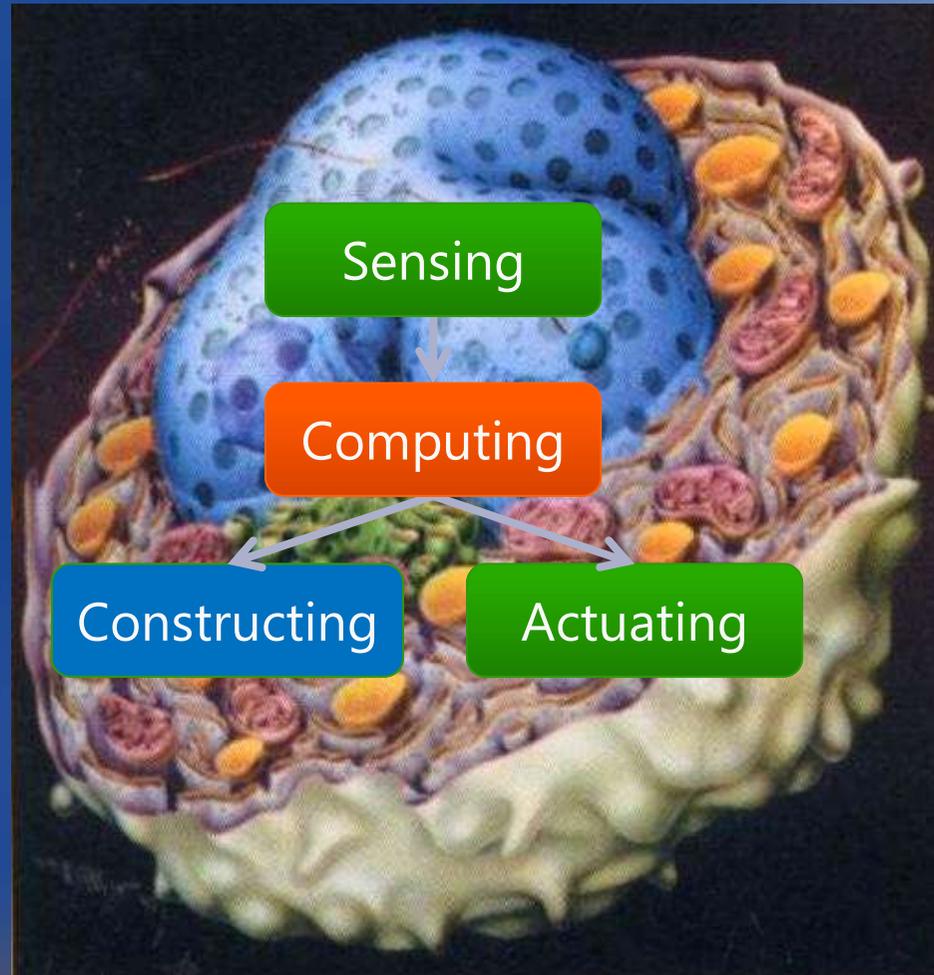
George M. Church,<sup>1,2</sup> Yuan Gao,<sup>3</sup> Sriram Kosuri<sup>1,2\*</sup>



We have machines that can read (sequence) and write (synthesize) DNA. The **Carlson Curve** of "productivity" is growing **much faster than Moore's Law**.

Cost of sequencing is decreasing rapidly (\$1000 whole human genome), while cost of synthesis is decreasing very slowly. [Rob Carlson, [www.synthesis.cc](http://www.synthesis.cc)]

# Curing



# Interfacing to Biology

- A doctor in each cell

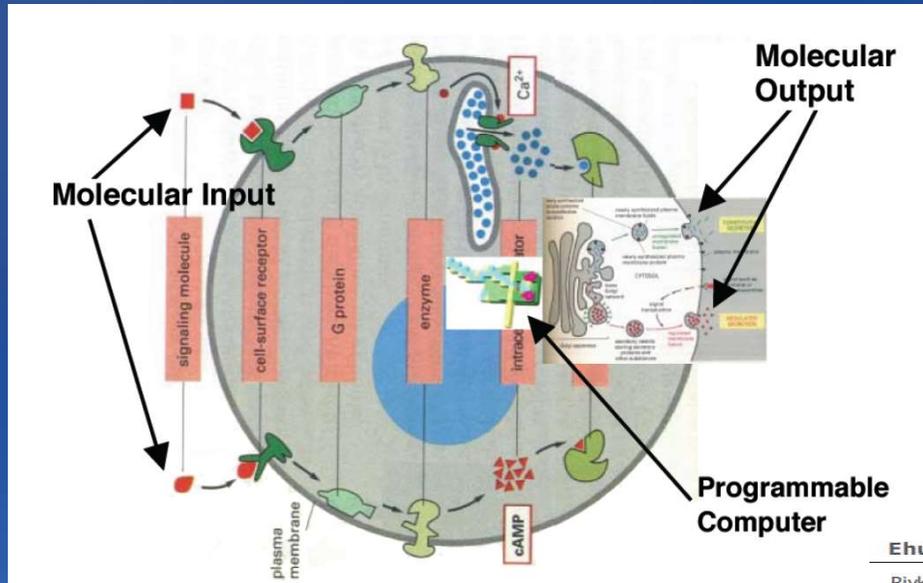


Fig. 1 Medicine in 2050: "Doctor in a Cell"

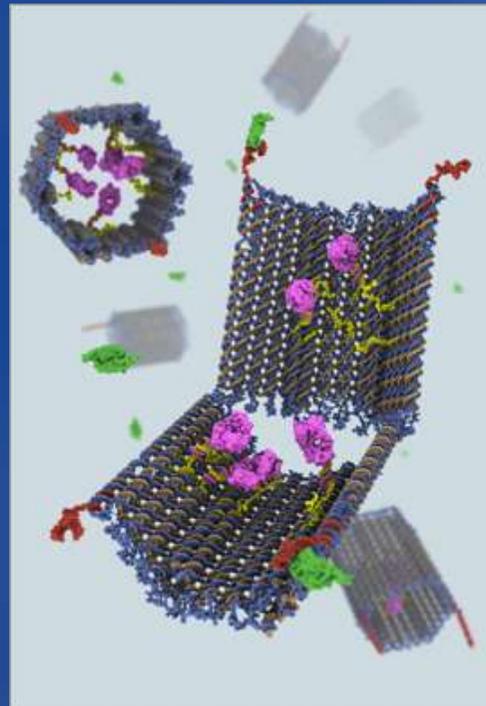
Ehud Shapiro

Rivka Adar  
Kobi Benenson  
Gregory Linshitz  
Aviv Regev  
William Silverman

**Molecules and  
computation**

~2002

# Programmed Drug Delivery



## A Logic-Gated Nanorobot for Targeted Transport of Molecular Payloads

Shawn M. Douglas<sup>†</sup>, Ido Bachelet<sup>†</sup>, George M. Church<sup>†</sup>

<sup>†</sup> See all authors and affiliations

*Science* 17 Feb 2012:  
Vol. 335, Issue 6070, pp. 831-834  
DOI: 10.1126/science.1214081

# Molecular Programming: The Execution Aspect

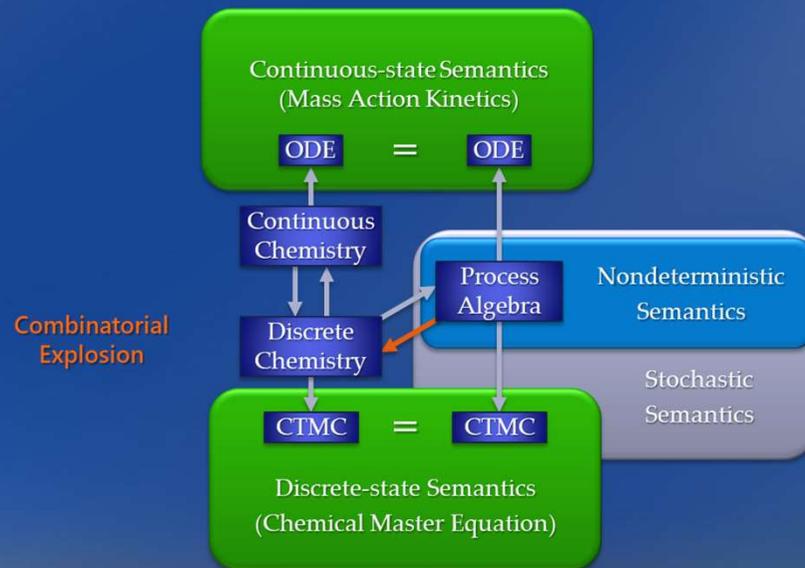
How do we "run" a molecular program?

# Programming Language: Chemistry

- A Lingua Franca between Biology, Dynamical Systems, and Concurrent Languages
- Chemical Reaction Networks
  - $A + B \xrightarrow{r} C + D$  (the program)
- Ordinary Differential Equations
  - $d[A]/dt = -r[A][B] \dots$  (the behavior)
- Rich analytical techniques based on Calculus and more recently on stochastic models

# Chemistry as a Concurrent Language

- A connection with the theory of concurrency
  - Via Process Algebra and Petri Nets



# Chemical Programming Examples

*specification*

$Y := \min(X1, X2)$

$Y := \max(X1, X2)$

*program*

$X1 + X2 \rightarrow Y$

$X1 \rightarrow L1 + Y$

$X2 \rightarrow L2 + Y$

$L1 + L2 \rightarrow K$

$Y + K \rightarrow 0$

$\max(X1, X2) =$   
 $(X1 + X2) - \min(X1, X2)$

(but is not computed  
"sequentially": it is a form  
of concurrent computation)

*chemical reaction network*

# Chemical Reaction Networks

- Finite list of chemical reactions over a finite set of species
  - N.B.: "abstract" species, not specific atoms/molecules that physically exist
- Computationally Powerful
  - Turing-complete up to an arbitrarily small error
- Full Turing Completeness
  - When including complexation (polymerization), which DNA enables (complexation encodes an actual infinity of chemical reactions by finite means)

# How do we “run” Chemistry?

- Chemistry is not easily executable
  - “Please Mr Chemist, execute me this bunch of reactions that I just made up”
- Most molecular languages are not executable
  - They are **descriptive** (modeling) languages
- How can we **execute** molecular languages?
  - With real molecules?
  - That we can design ourselves?
  - And that we can buy on the web?

# DNA Strand Displacement

An "unnatural" use of DNA for emulating  
*any* system of chemical reactions  
with real molecules

# Domains

- Subsequences on a DNA strand are called **domains**
  - *provided* they are "independent" of each other
- Differently named domains must not **hybridize**
  - With each other, with each other's complement, with subsequences of each other, with concatenations of other domains (or their complements), etc.

CTTGAGAATCGGATATTCGGATCGCGATTAAATCAAATG



x                      y                      z

oriented DNA  
single strand

# Short Domains



DNA double  
strand

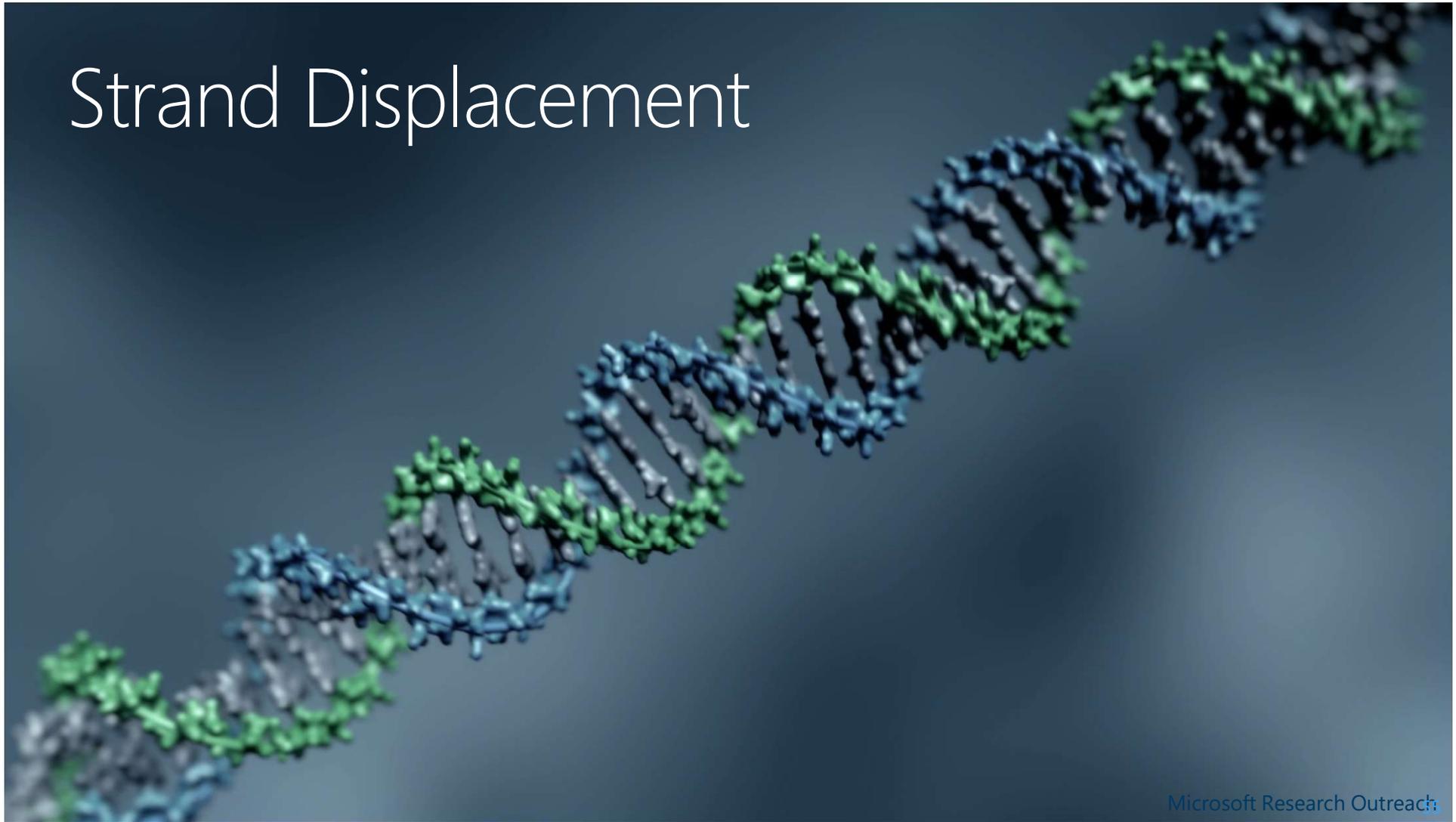
Reversible Hybridization

# Long Domains

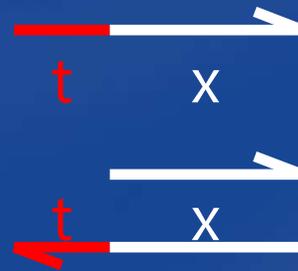


Irreversible Hybridization

# Strand Displacement



# Strand Displacement



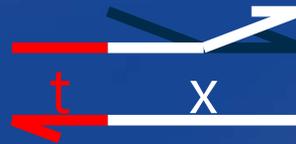
“Toehold Mediated”

# Strand Displacement



Toehold Binding

# Strand Displacement



Branch Migration

# Strand Displacement



Displacement

# Strand Displacement



Irreversible release

# Bad Match



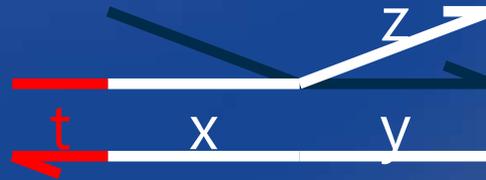
# Bad Match



# Bad Match



# Bad Match



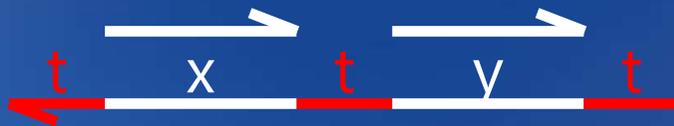
Cannot proceed  
Hence will undo

# Two-Domain Architecture

- Signals: 1 toehold + 1 recognition region



- Gates: "top-nicked double strands" with open toeholds



Garbage collection  
"built into" the gate  
operation

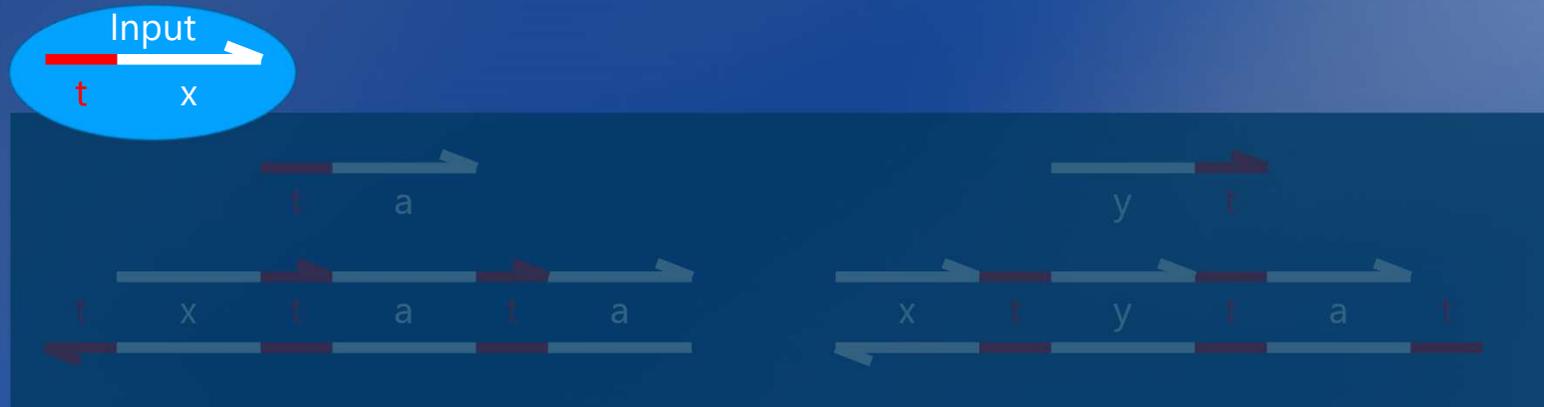
## Two-Domain DNA Strand Displacement

*Luca Cardelli*

In S. B. Cooper, E. Kashefi, P. Panangaden (Eds.):  
Developments in Computational Models (DCM 2010).  
EPTCS 25, 2010, pp. 33-47. May 2010.

# Transducer

# Transducer $x \rightarrow y$



# Transducer $x \rightarrow y$



Built by self-assembly!

$ta$  is a *private* signal (a different 'a' for each  $xy$  pair)

# Transducer $x \rightarrow y$



# Transducer $x \rightarrow y$



# Transducer $x \rightarrow y$



# Transducer $x \rightarrow y$



So far, a **tx** signal has produced an **at** cosignal.  
But we want signals as output, not cosignals.

# Transducer $x \rightarrow y$



# Transducer $x \rightarrow y$



# Transducer $x \rightarrow y$



# Transducer $x \rightarrow y$



Here is our output **ty** signal.

But we are not done yet:

- 1) We need to make the output irreversible.
- 2) We need to remove the garbage.

We can use (2) to achieve (1).

# Transducer $x \rightarrow y$



# Transducer $x \rightarrow y$



# Transducer $x \rightarrow y$



# Transducer $x \rightarrow y$



# Transducer $x \rightarrow y$



# Transducer $x \rightarrow y$

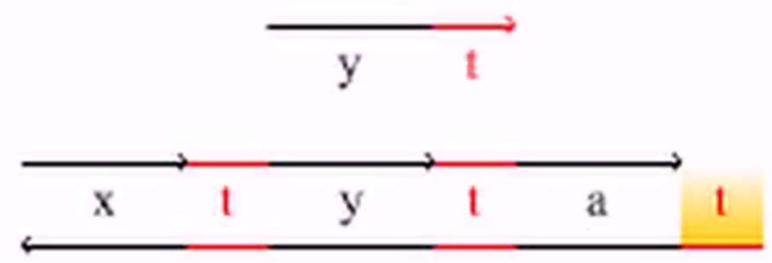
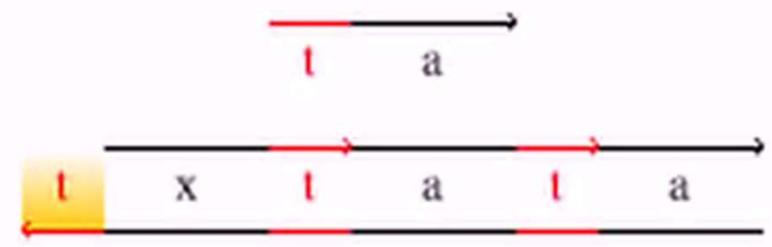


Done.

N.B. the gate is consumed: it is the energy source

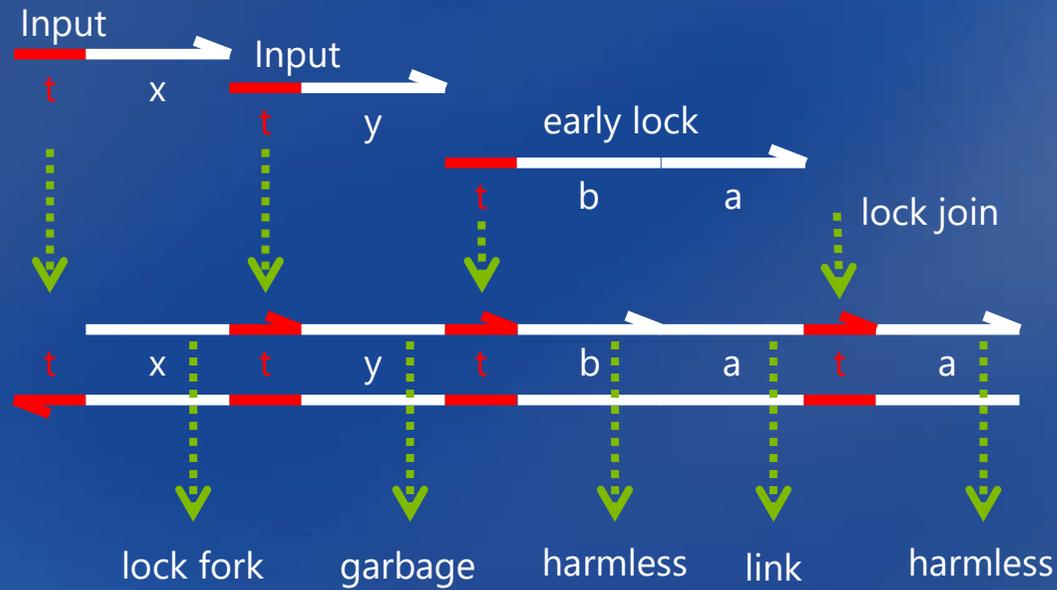
(no proteins, no enzymes, no heat-cycling, etc.; just DNA in salty water)

# Transducer $x \rightarrow y$



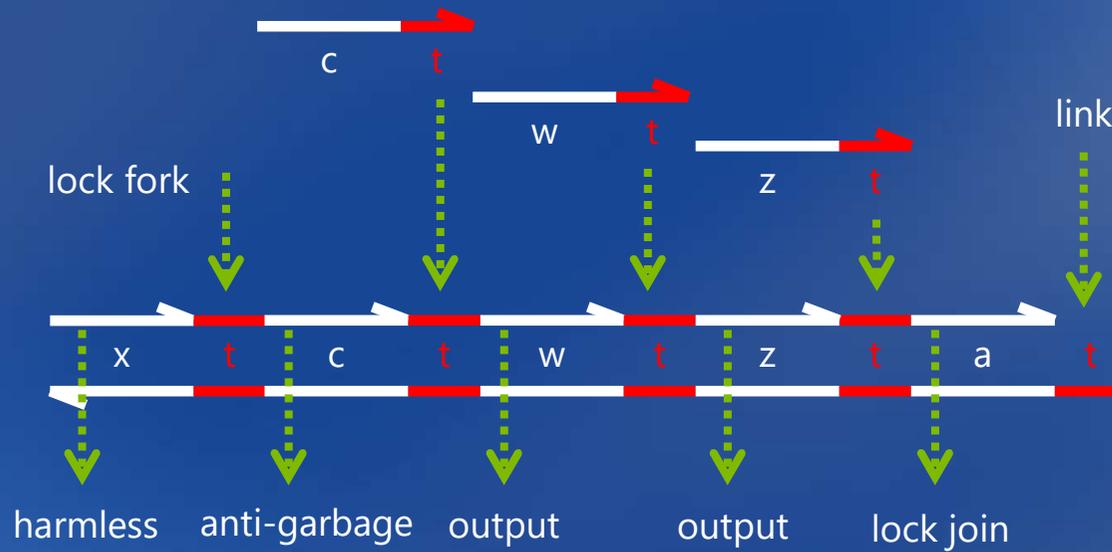
# Reaction $x + y \rightarrow z + w$

join  
half



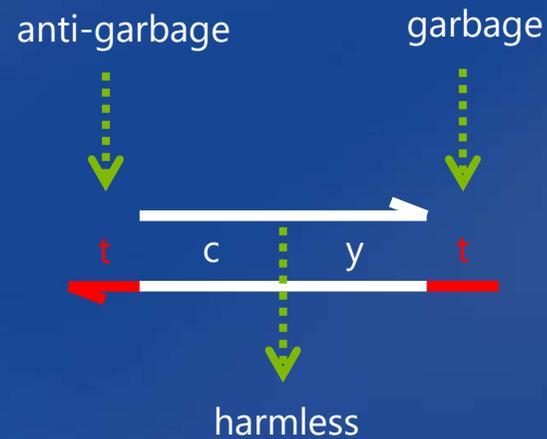
# Reaction $x + y \rightarrow z + w$

fork  
half



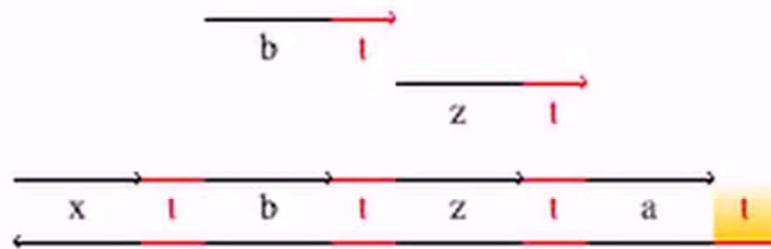
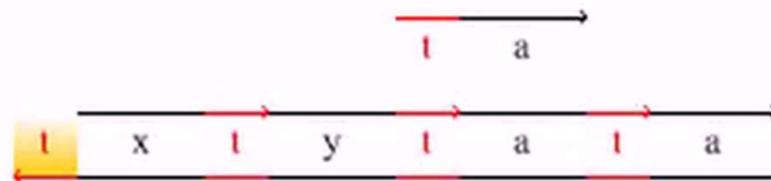


garbage  
collection



Powered by Sothink

Join  $x+y \rightarrow z$



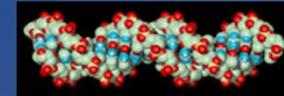
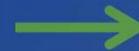
# Approximate Majority Algorithm

- Given two populations of agents (or molecules)
  - Randomly communicating by radio (or by collisions)
  - Reach an agreement about which population is in majority
  - By converting all the minority to the majority[Angluin et al., Distributed Computing, 2007]

- 3 rules of agent (or molecule) interaction



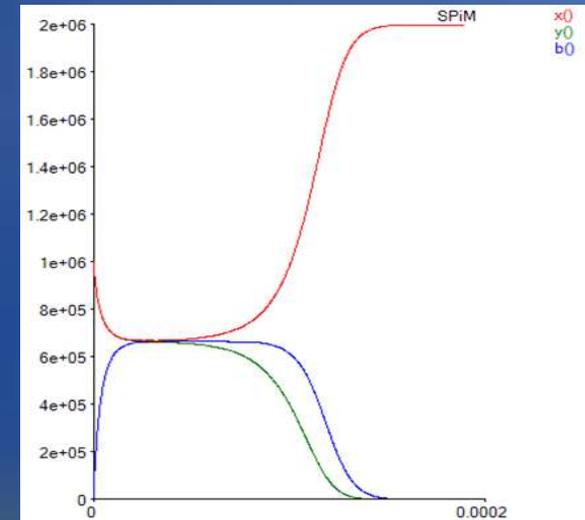
“our program”



# Optimal Consensus Algorithm

- Fast: reaches agreement in  $O(\log n)$  time w.h.p.
  - $O(n \log n)$  communications/collisions
  - Even when initially  $\#X = \#Y!$  (stochastic symmetry breaking)
- Robust: true majority wins w.h.p.
  - If initial majority exceeds minority by  $\omega(\sqrt{n \log n})$
  - Hence the agreement state is stable

Stochastic simulation of worst-case scenario with initially  $\#X = \#Y$

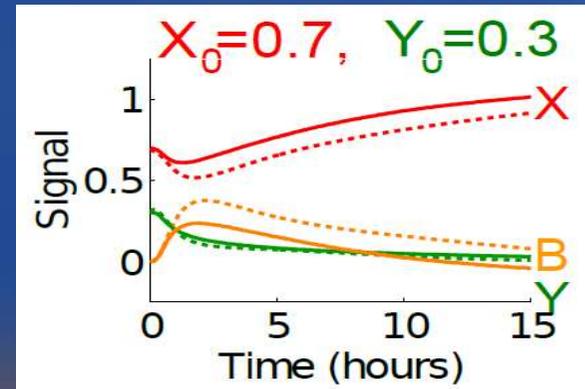
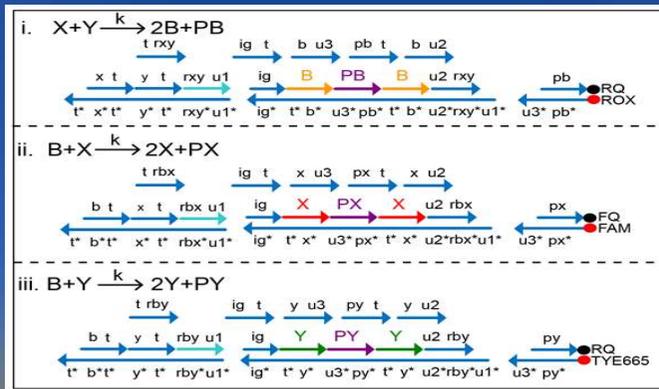


# DNA Implementation of the Approximate Majority algorithm

nature  
nanotechnology

Programmable chemical controllers made from DNA

Yuan-Jyue Chen, Neil Dalchau, Niranjan Srinivas, Andrew Phillips, Luca Cardelli, David Soloveichik & Georg Seelig

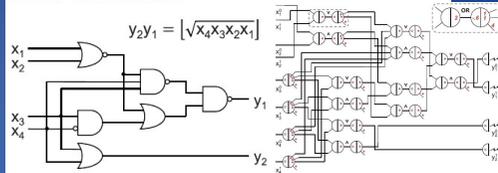


# Some Large-scale Circuits (so far...)

3 JUNE 2011 VOL 332 SCIENCE

## Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades

Lulu Qian<sup>1</sup> and Erik Winfree<sup>1,2,3\*</sup>

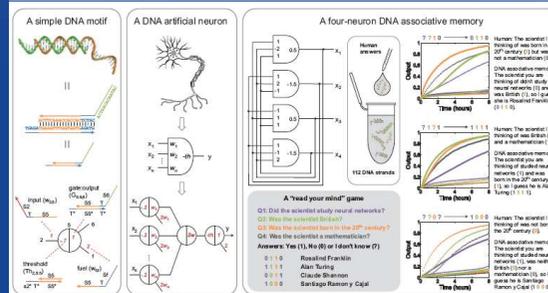


Computing the square root of a 4-bit number

368 | NATURE | VOL 475 | 21 JULY 2011

## Neural network computation with DNA strand displacement cascades

Lulu Qian<sup>1</sup>, Erik Winfree<sup>1,2,3</sup> & Joshua Bruck<sup>3,4</sup>

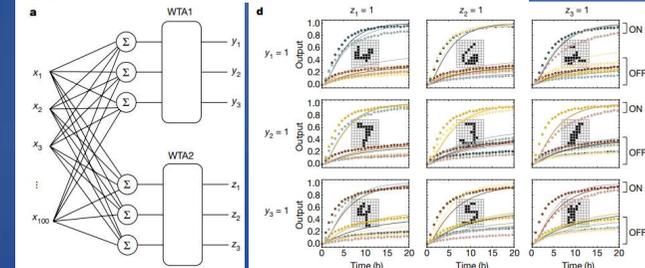


Classifying 4 distinct 4-bit patterns via 4 neurons

370 | NATURE | VOL 559 | 19 JULY 2018

## Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks

Kevin M. Cherry<sup>1</sup> & Lulu Qian<sup>1,2\*</sup>



Classifying 9 distinct 100-bit patterns via WTA networks

# Scaling up: DNA Circuit Boards

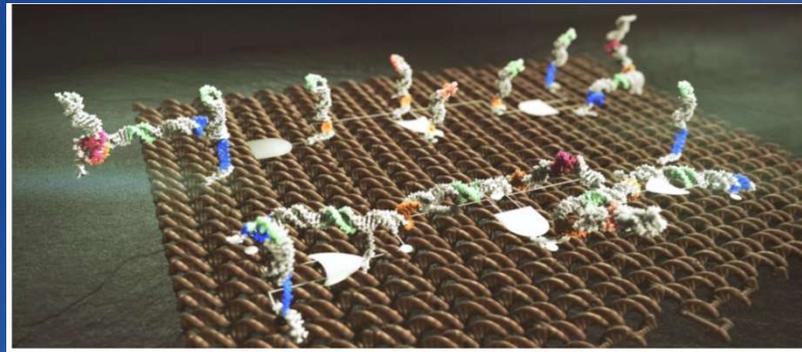
ARTICLES

PUBLISHED ONLINE: 24 JULY 2017 | DOI: 10.1038/NNANO.2017.127

nature  
nanotechnology

## A spatially localized architecture for fast and modular DNA computing

Gourab Chatterjee<sup>1</sup>, Neil Dalchau<sup>2</sup>, Richard A. Muscat<sup>3</sup>, Andrew Phillips<sup>2\*</sup> and Georg Seelig<sup>3,4\*</sup>

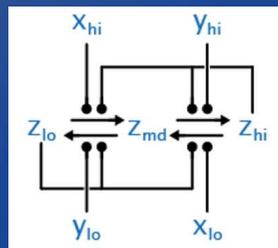
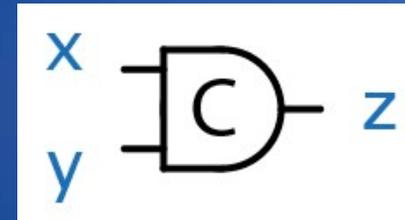


The first computational circuit boards made of DNA

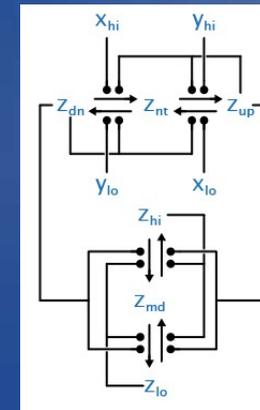
<https://www.microsoft.com/en-us/research/blog/researchers-build-nanoscale-computational-circuit-boards-dna>

# Avoiding Clocks

- Muller C-Element
  - A Boolean gate
  - When  $x = y$  then  $z = x = y$ , otherwise  $z$  remembers its *last* state.



Core C-Element  
(AM with external inputs)



Full C-Element with output  
rectified by another AM

**Chemical Reaction Network Designs for Asynchronous Logic Circuits.**

Luca Cardelli, Marta Kwiatkowska, Max Whitby.  
Natural Computing Journal.

# Algorithm Design

A software pipeline for Molecular Programming

# Development Tools

MSRC Biological Computation Group

## Visual DSD

A Development Environment for DNA Strand Displacement

The screenshot shows the Interface software interface. On the left, a code editor displays the following code:

```
def bind = kt*1.0e-9 (* /nM/s *)
def unbind = kt*exp_DeltaG_over_RT (* /s *)
new t@bind,unbind
new u@bind,unbind
new f1@0.0,0.0

def onex = 50.0

(* x + y -> y + z *)
def Cat(N,x,y,z) =
new a
( (1.5*N) * t^x:[x t^y]:[y u^z]:[a]
  (1.5*N) * [x]:[t^x z]:[t^y y]:u^z
  (2.0*N) * <u^ a>
  (2.0*N) * <z t^>
)

def Rep(N,x,f1) =
((3.0*N) * t^x:[x]<f1^>)

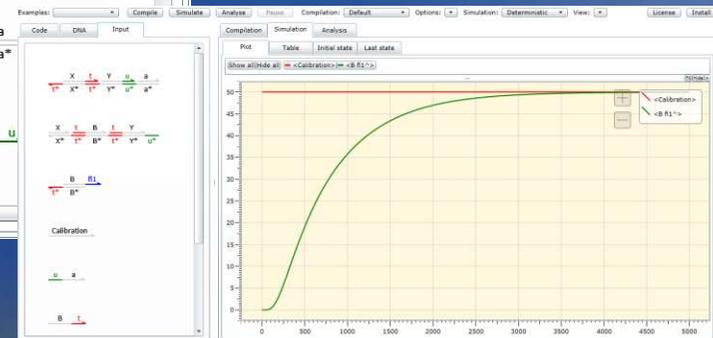
(onex = <Calibration>
| Cat(onex,X,Y,B)
| Rep(onex,B,f1)
| onex = <t^ X>
| onex = <t^ Y>
)
```

The central panel shows visual representations of DNA strand displacement reactions. It includes diagrams for species (X, Y, U, A), reactions (X + Y ↔ X:Y), and domains (t, u, a). The diagrams use colored lines to represent DNA strands and their interactions.

JOURNAL OF THE ROYAL SOCIETY **Interface**

**A programming language for composable DNA circuits**

Andrew Phillips and Luca Cardelli

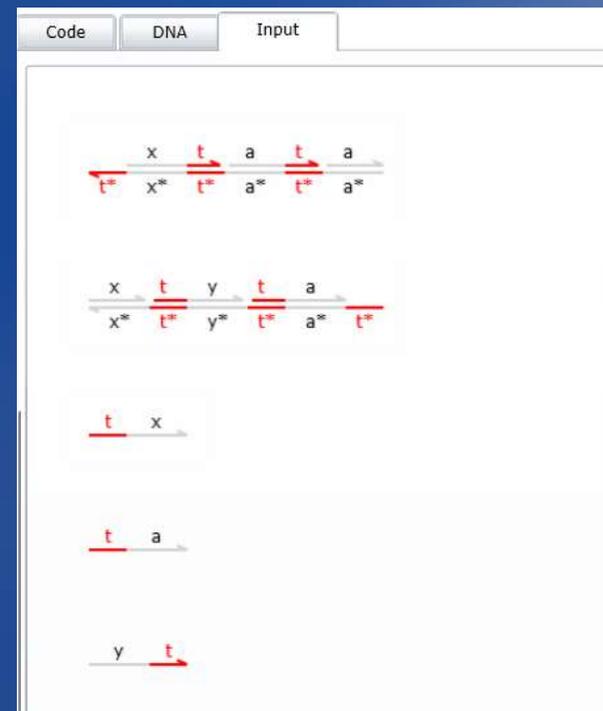


# A Language for DNA Structures

- Describe the initial *structures* (not behavior)

```
Code  DNA  Input
directive duration 10000.0 points 1000
directive plot <t^ x>; <t^ y>; <t^ z>
new t
def T(N,x,y) =
  new a
  ( N * <t^ a>
  | N * <y t^>
  | N * t^*: [x t^]: [a t^]: [a] (* Input gate *)
  | N * [x]: [t^ y]: [t^ a]: t^* (* Output gate *)
  )
  ( <t^ x> | T(1,x,y) )
```

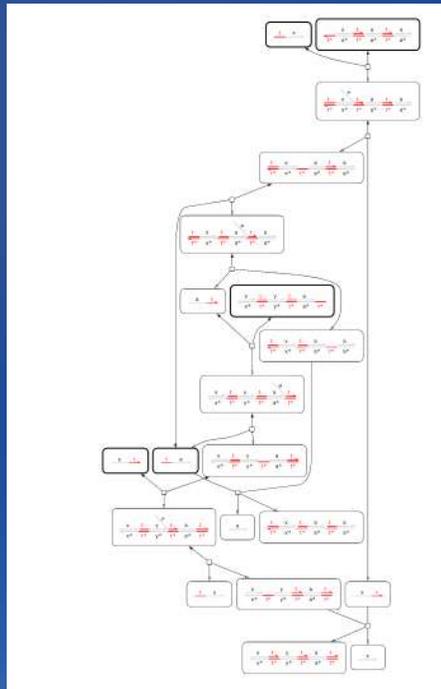
=



# Compute Species and Reactions

- Recursively computed from the initial structures

# Reaction Graph and Export



```
Compilation Simulation Analysis
Species Reactions Graph Text Domains SBML
Save as XML
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level2/version1" level="2" version="1">
<model>
<listOfCompartments>
<compartment id="c" size="1"/>
</listOfCompartments>
<listOfSpecies>
<species id="s_id0" name="&t^ x^" compartment="c" initialAmount="1" constant="false"/>
<species id="s_id1" name="&t^ a^" compartment="c" initialAmount="1" constant="false"/>
<species id="s_id2" name="&t^ y^" compartment="c" initialAmount="1" constant="false"/>
<species id="s_id3" name="[t^ x^][t^ a^]" compartment="c" initialAmount="1" constant="false"/>
<species id="s_id4" name="[t^ x^]&t^ x^" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id5" name="[t^ x^][t^ a^][t^ a^]" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id6" name="[t^ x^][t^ a^]&t^ a^" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id7" name="[t^ x^][t^ a^][t^ a^]" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id8" name="[t^ x^][t^ a^][t^ a^]" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id9" name="&t^ a^" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id10" name="&t^ a^ t^" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id11" name="&t^ x^ t^" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id12" name="[x^][t^ y^]:[t^ a^]" compartment="c" initialAmount="1" constant="false"/>
<species id="s_id13" name="[x^][t^ y^]:[t^ a^]&t^ a^" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id14" name="[x^][t^ y^]:[t^ a^]" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id15" name="[x^][t^ y^]&t^ y^" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id16" name="[x^][t^ y^]:[y^ t^]" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id17" name="[x^ t^]:[y^ t^]" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id18" name="&t^ x^" compartment="c" initialAmount="0" constant="false"/>
<species id="s_id19" name="&t^ y^" compartment="c" initialAmount="0" constant="false"/>
</listOfSpecies>
<listOfReactions>
<reaction id="r_id20" reversible="false">
<listOfReactants>
<speciesReference species="s_id3"/>
<speciesReference species="s_id0"/>

```

# Simulation

- Deterministic
- Stochastic (Gillespie)
- Probabilistic (CME)
- Linear Noise Approximation
- "JIT"



# State Space Analysis

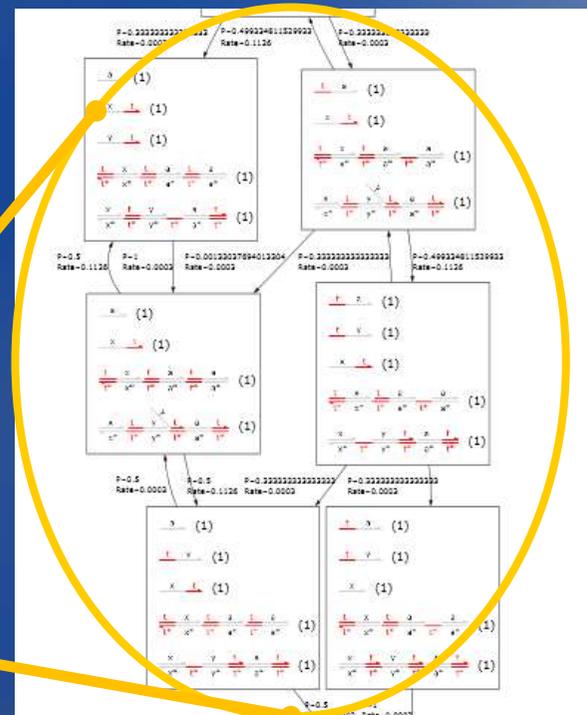
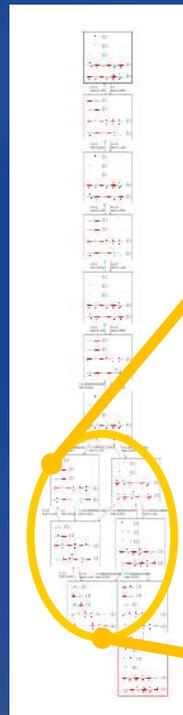
Compilation Simulation Analysis  
Graph Text PRISM Visualise

**INITIAL STATE:**

- $\frac{t}{x^m} a (1)$
- $\frac{t}{x^m} x (1)$
- $y \frac{t}{x^m} (1)$
- $\frac{x}{x^m} \frac{t}{t^m} \frac{y}{y^m} \frac{t}{t^m} \frac{a}{a^m} \frac{t}{t^m} (1)$
- $\frac{x}{t^m} \frac{t}{x^m} \frac{t}{t^m} \frac{a}{a^m} \frac{t}{t^m} \frac{a}{a^m} (1)$

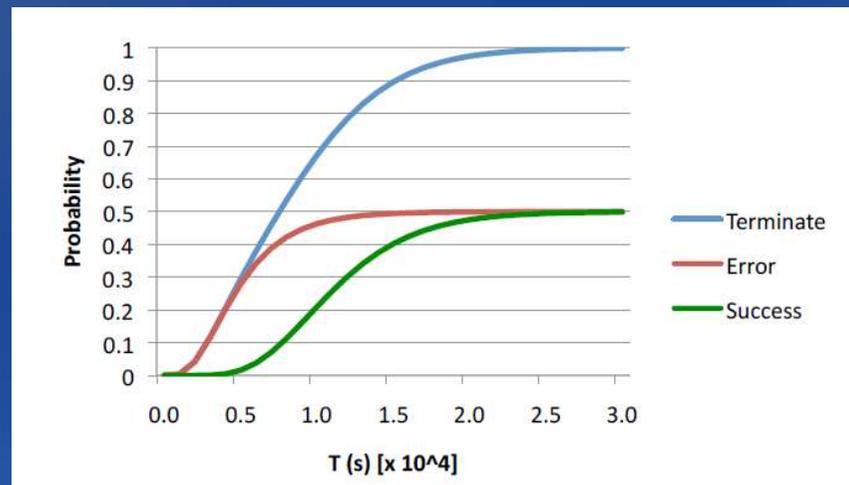
**TERMINAL STATE:**

- $a (1)$
- $\frac{t}{x^m} y (1)$
- $x (1)$
- $\frac{t}{t^m} \frac{x}{x^m} \frac{t}{t^m} \frac{a}{a^m} \frac{t}{t^m} \frac{a}{a^m} (1)$
- $\frac{x}{x^m} \frac{t}{t^m} \frac{y}{y^m} \frac{t}{t^m} \frac{a}{a^m} \frac{t}{t^m} (1)$



# Modelchecking

- Export to PRISM probabilistic modelchecker



JOURNAL  
OF  
THE ROYAL  
SOCIETY

Interface

Design and analysis of DNA strand displacement devices using probabilistic model checking

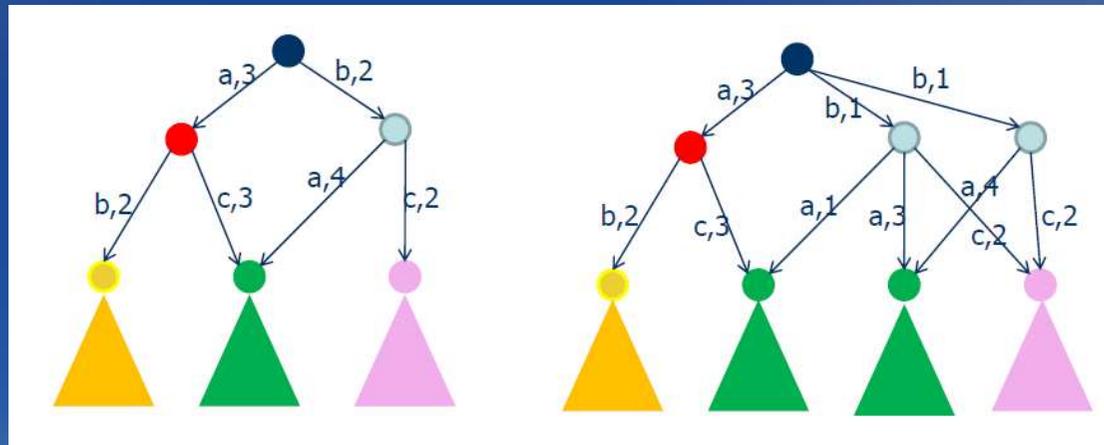
Matthew R. Lakin<sup>1,3,†</sup>, David Parker<sup>2,†</sup>, Luca Cardelli<sup>1</sup>,  
Marta Kwiatkowska<sup>2</sup> and Andrew Phillips<sup>1,\*</sup>

# Verification

- Quantitative theories of system equivalence and approximation.

CONTINUOUS MARKOVIAN LOGICS  
AXIOMATIZATION AND QUANTIFIED METATHEORY

RADU MARDARE, LUCA CARDELLI, AND KIM G. LARSEN



# Physical Execution

A wetlab pipeline for Molecular Programming

# Computer Aided Design

## MSRC Biological Computation Group

### Visual DSD

A Development Environment for DNA Strand Displacement

The screenshot shows the Interface software interface. On the left, the 'Code' window displays the following code:

```
def bind = kt*1.0e-9 (* /nM/s *)
def unbind = kt*exp_DeltaG_over_RT (* /s *)
new t@bind,unbind
new u@bind,unbind
new f1@0.0,0.0

def onex = 50.0

(* x + y -> y + z *)
def Cat(N, x, y, z) =
new a
( (1.5*N) * t^x:[x t^y]:[y u^z]:[a]
  (1.5*N) * [x]:[t^x z]:[t^y y]:u^z
  (2.0*N) * <u^ a>
  (2.0*N) * <z t^>
)

def Rep(N,x,F1) =
((3.0*N) * t^x:[x]<f1^>)

(onex = <Calibration>
| Cat(onex,X,Y,B)
| Rep(onex,B,F1)
| onex = <t^ X>
| onex = <t^ Y>
)
```

The central 'Reactions' window displays several DNA strand displacement reactions. Each reaction is represented by a set of colored lines (red, green, blue) representing DNA strands, with arrows indicating the direction of the reaction. The reactions are arranged in a grid, showing the interaction between different strands and the resulting products.

JOURNAL OF THE ROYAL SOCIETY

Interface

A programming language for composable DNA circuits

Andrew Phillips and Luca Cardelli

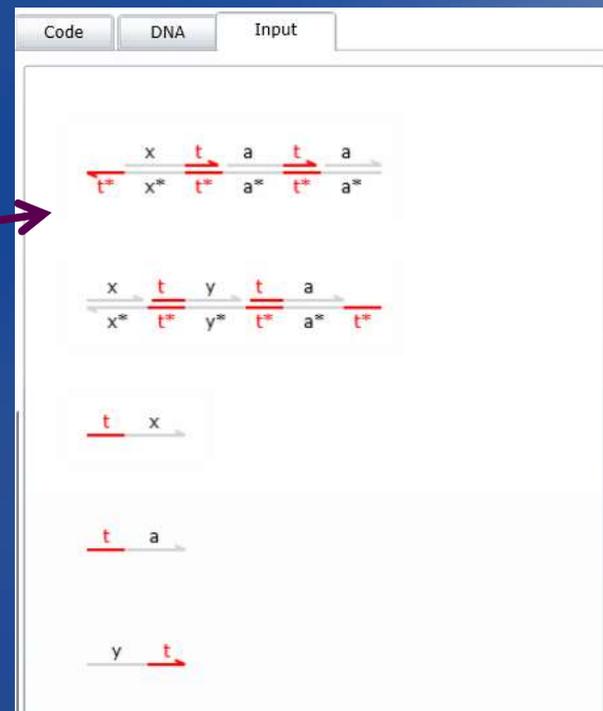


# Output of Design Process

- Domain structures
  - (DNA sequences to be determined)

“Ok, how do I run this for real?”

“What are the actual DNA sequences?”

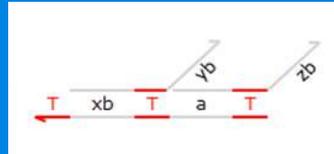


# From Structures to Sequences



www.nupack.org

DSD Structure



"Dot-Paren" representation

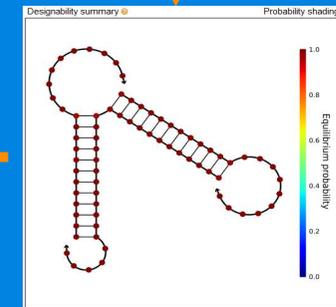
Nucleic acid type:  RNA  DNA  Temperature:  °C Number of designs:

Target structure:

Output Sequences

Ensemble defect (nt)	Normalized ensemble defect (%)	GC content (%)	Sequence	
0.2	0.3	57.5	GCUCGGAUACCCAAAGAAC AA+GCGAUCAAGCCCUUU UUUCC+GGGCUUGAUCGGG GUAUCGACGUCGC	<input type="button" value="To Utilities"/> <input type="button" value="To Analysis"/>

Thermodynamic Synthesis



"Ok, where do I buy these?"



# "DNA Synthesis"

dna synthesis × Search

About 8,610,000 results (0.24 seconds) Advanced search

► **Custom DNA Synthesis** Ads  
[www.Biomatik.com](http://www.Biomatik.com) High Quality Custom Gene **Synthesis**, Best Price Guaranteed! Get A Quote.

**Order Gene at GenScript**  
[www.GenScript.com](http://www.GenScript.com) \$0.29/bp. Any Gene in ANY Vector Proven increase protein expression

**Gene Synthesis \$0.35/bp**  
[www.epochlifescience.com](http://www.epochlifescience.com) Dependable Service @ Low Price: Come on Down and Save Your Budgets!

**DNA synthesis - Wikipedia, the free encyclopedia** ☆ 🔍  
DNA **synthesis** commonly refers to: DNA replication - DNA biosynthesis (in vivo DNA amplification); Polymerase chain reaction - enzymatic **DNA synthesis** (in ...  
[en.wikipedia.org/wiki/DNA\\_synthesis](http://en.wikipedia.org/wiki/DNA_synthesis) - Cached - Similar

**DNA replication - Wikipedia, the free encyclopedia** ☆ 🔍  
DNA replication, the basis for biological inheritance, is a fundamental ...  
[en.wikipedia.org/wiki/DNA\\_replication](http://en.wikipedia.org/wiki/DNA_replication) - Cached - Similar

Show more results from wikipedia

**Integrated DNA Technologies - Home** ☆ 🔍 Visits - May 24  
Trade Your **Synthesizer** for Oligos ... **DNA/RNA Synthesis**. Modifications. Purifications. Gene Expression. Genotyping ... Custom **DNA Oligos** ... Custom **RNA Oligos** ...  
[www.idt.com/](http://www.idt.com/) - Cached - Similar



# From Sequences to Molecules

- Copy&Paste from nupack

**XX-IDT**  
INTEGRATED DNA  
TECHNOLOGIES

Chat is now closed.  
Please click to email  
a representative.

[LogIn]  
Spain

0 Items € 0,00

Home Products Order Support Services SciTools Search Go

### Order Oligos

Change Form: 1 Expand to this many items Duplex Paste Go

25 nmole DNA Oligo = 15-60 bases  
100 nmole DNA oligo = 10-90 bases  
250 nmole DNA oligo = 5-100 bases  
1 µmole DNA oligo = 5-100 bases  
5 µmole DNA oligo = 5-50 bases  
10 µmole DNA oligo = 5-50 bases  
25 nmole Ultramer DNA Oligo = 60-200 bases  
4 nmole Ultramer DNA Oligo = 60-200 bases  
PAGE Ultramer DNA Oligo = 60-200 bases

Scale: 25 nmole DNA oligo Purification: Standard  
Sequence Name: 5'-ACT GCA CCA TAA GCA ACT TTT  
3'

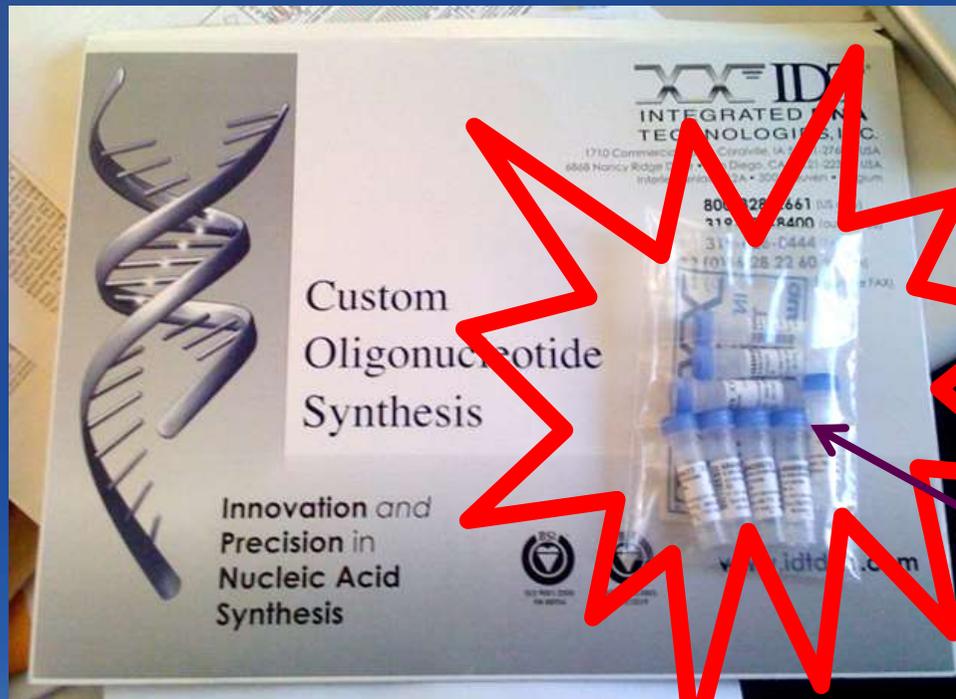
Enter your notes here. Please do not enter modifications.

Help 5' mods Internal mods 3' mods Services Mixed Bases

**Preparative Services**  
 LabReady (more detail) € 2,82 EUR

**Customized Labels** (more detail)  
Stock IDT Label FREE

# Molecules by FedEx



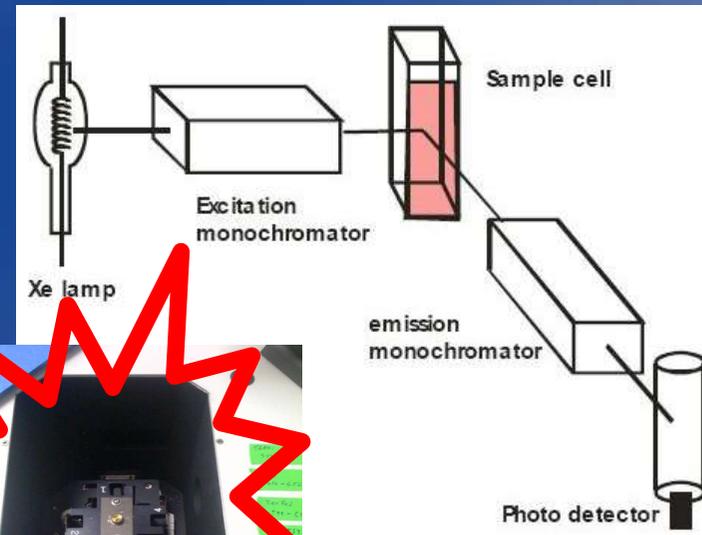
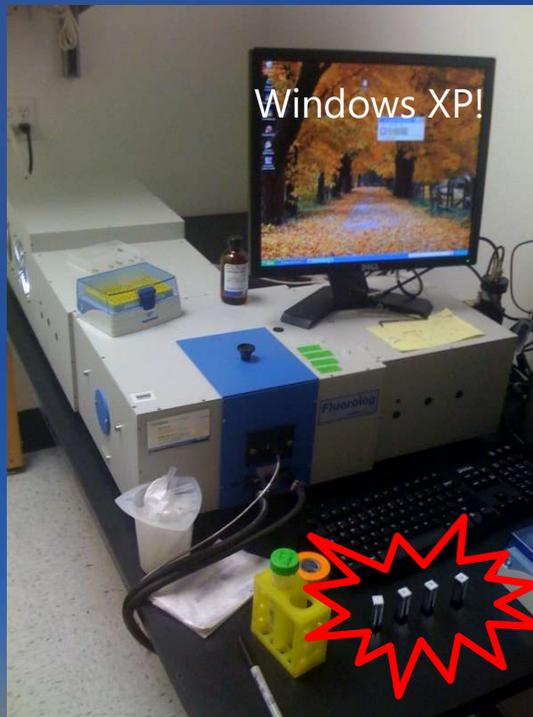
"Ok, how do I run these?"

# Add Water

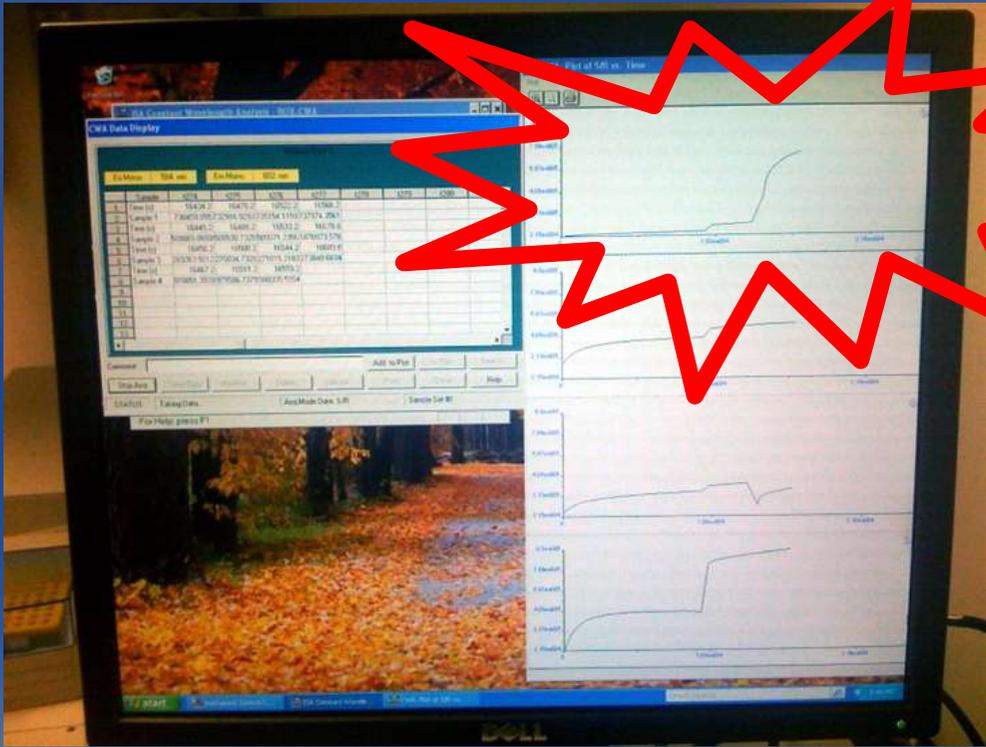


# Execute (finally!)

- Fluorescence is your one-bit 'print' statement



# Output

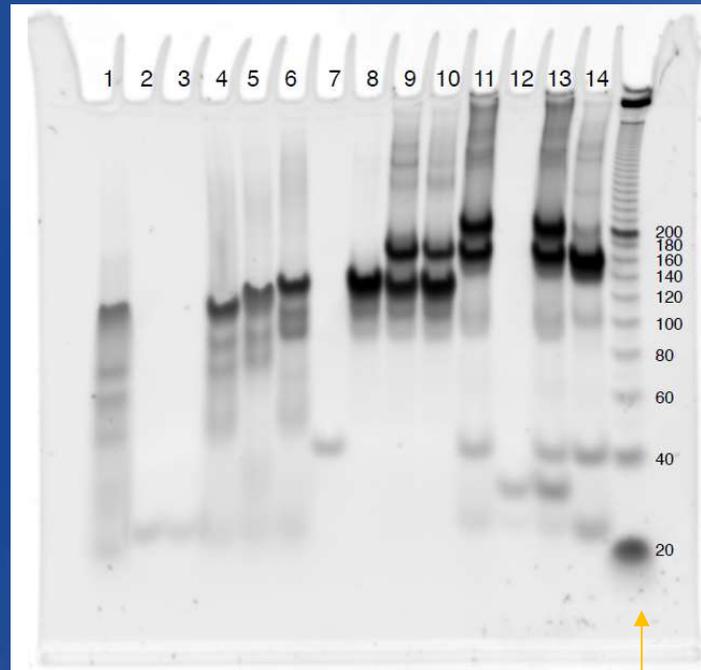


# Debugging

- A core dump

DNA strand length

polyacrylamide gel electrophoresis



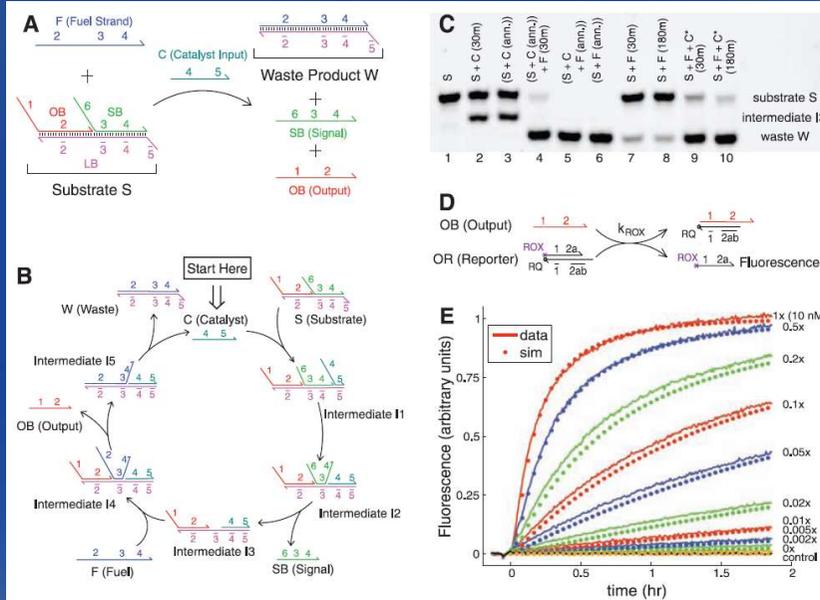
Various processing stages

Calibration scale

# Delivery!

## Engineering Entropy-Driven Reactions and Networks Catalyzed by DNA

David Yu Zhang, *et al.*  
*Science* **318**, 1121 (2007);  
 DOI: 10.1126/science.1148532



# Final Remarks

# "Modern" DNA Computing

- Non-goals
  - Not to solve NP-complete problems with large vats of DNA
  - Not to replace silicon
- Bootstrapping a carbon-based technology
  - To precisely control the organization and dynamics of matter and information at the molecular level
  - DNA is our engineering material
    - Its biological origin is "accidental" (but convenient)
    - It is an information-bearing programmable material
    - Other such materials will be (are being) developed

# State of the art

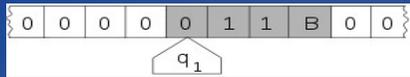
- Building a full software/hardware pipeline for a new fundamental technology
  - Mathematical Foundations [~ concurrency theory in the 80's]
  - Programming Languages [~ software engineering in the 70's]
  - Analytical Methods and Tools [~ formal methods in the 90's]
  - Device Architecture and Manufacturing [~ electronics in the 60's]
- To realize the potential of Molecular Programming
- “With *no alien technology*” [David Soloveichik]
- We have some good strategies. Device design is now largely a ‘software problem’ but with a significant ‘engineering scaleup and integration’ problem

# Ongoing Challenges

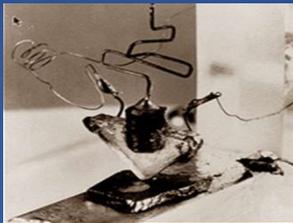
- In-vivo DNA survivability
- Complexity (and crosstalk)
- Manufacturing
- Speed
- Energy

# A Brief History of DNA

Turing Machine, 1936



Transistor, 1947



Computer programming

20<sup>th</sup> century

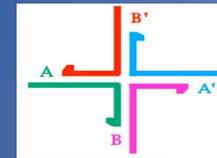
DNA, -3,800,000,000



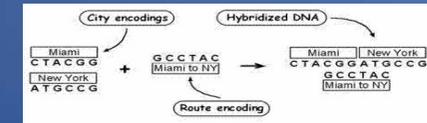
*Systematic manipulation of information*

*Systematic manipulation of matter*

Structural DNA Nanotech, 1982



DNA Algorithm, 1994



Molecular programming

21<sup>th</sup> century

# Resources

- DNA Computing and Molecular Programming Conference - incarnations since 1995

<http://www.dna-computing.org/>

- Molecular Programming Project (Caltech - U.W. - Harvard - UCSF)

<http://molecular-programming.org/> (2008-2018 NSF Expeditions in Computing)

- Georg Seelig's DNA Nanotech Lab at U.W. CS&E

<http://homes.cs.washington.edu/~seelig/>

- Biological Computation Group at Microsoft

<https://www.microsoft.com/en-us/research/group/biological-computation/>