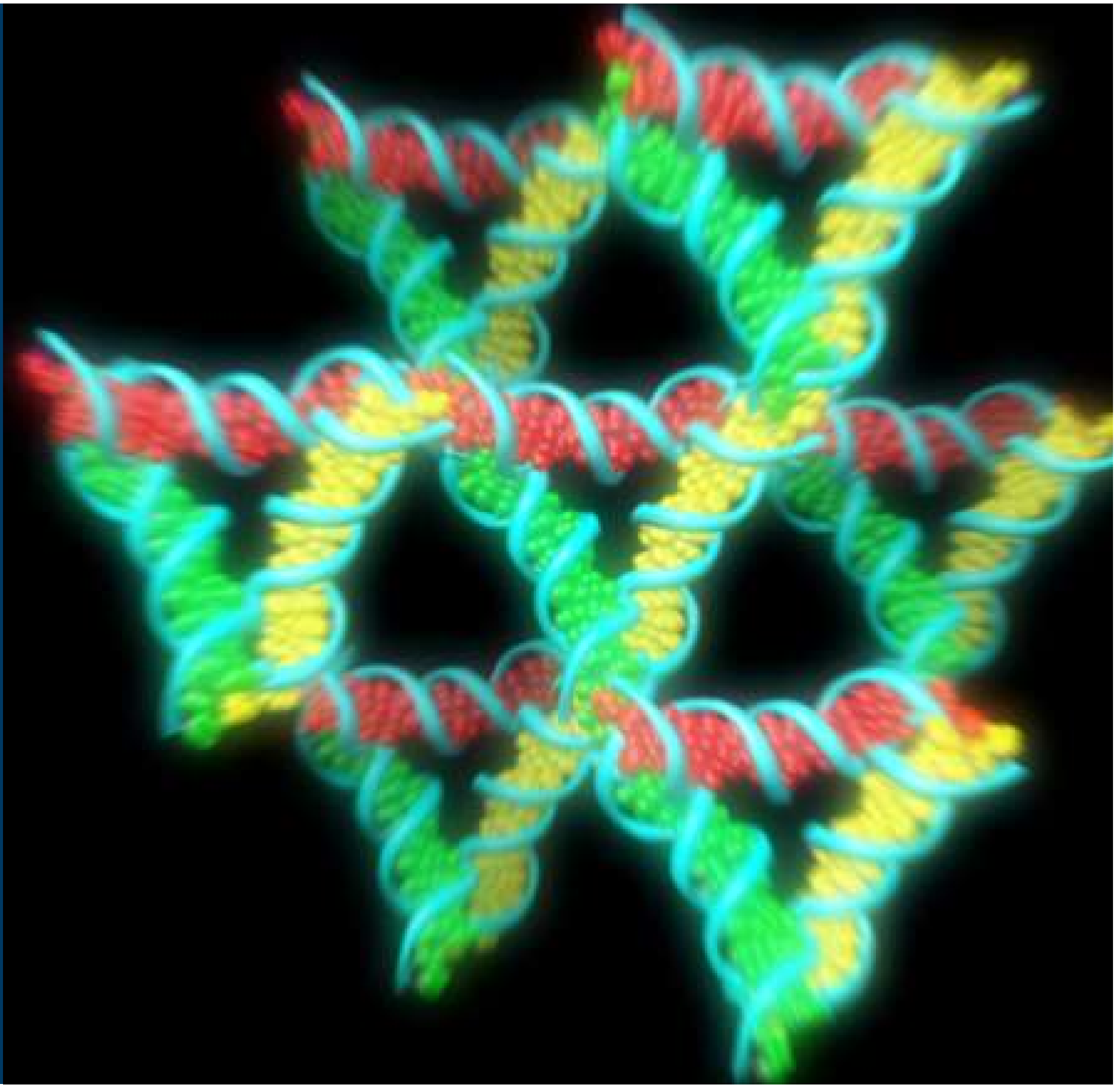
Molecular Programming

Luca Cardelli

Microsoft Research &
University of Oxford

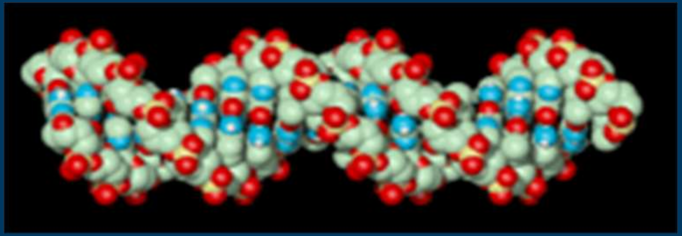
2018-06-26, UPTEC Porto



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
  - Example of a molecular algorithm



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.

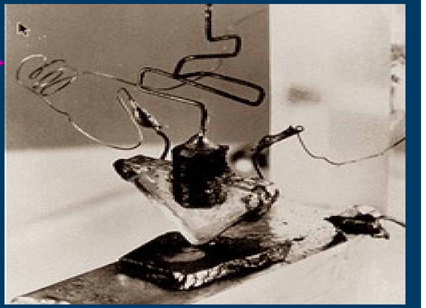
50+ years later

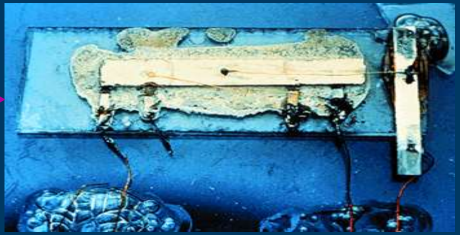
Jan 2010 25nm NAND flash
Intel&Micron. ~50atoms

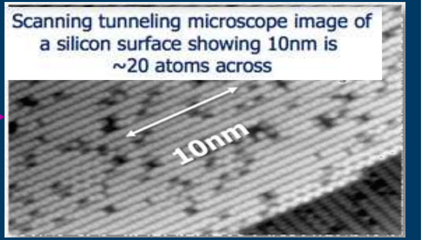
Jun 2018 7nm (54nm pitch)
TSMC, Intel, Samsung, GlobalFoundries - mass production

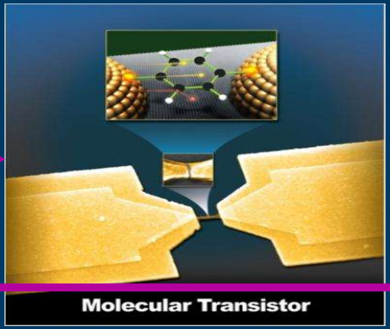
Single molecule transistor
Observation of molecular orbital gating
Nature, 2009; 462 (7276): 1039

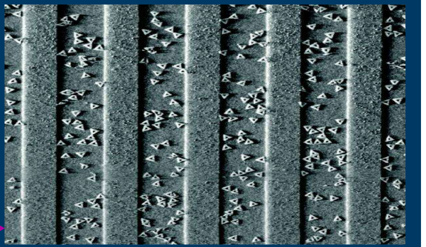
Molecules on a chip






Scanning tunneling microscope image of a silicon surface showing 10nm is ~20 atoms across
10nm





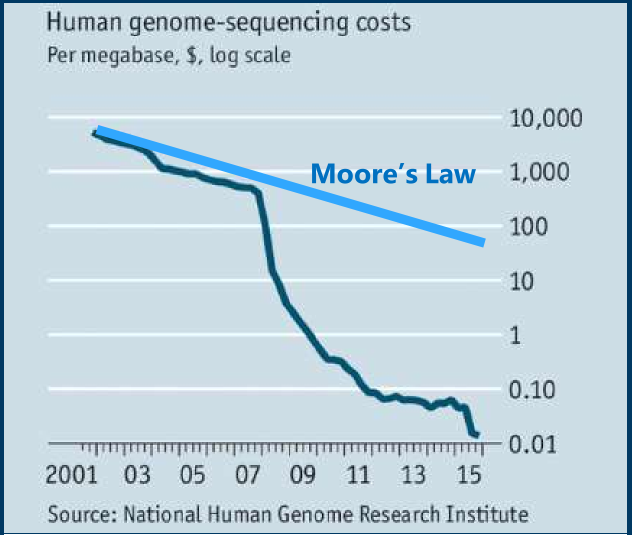
Placement and orientation of individual DNA shapes on lithographically patterned surfaces. Nature Nanotechnology 4, 557 - 561 (2009).

Race to the Bottom

Moore's Law is approaching the single-molecule limit

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

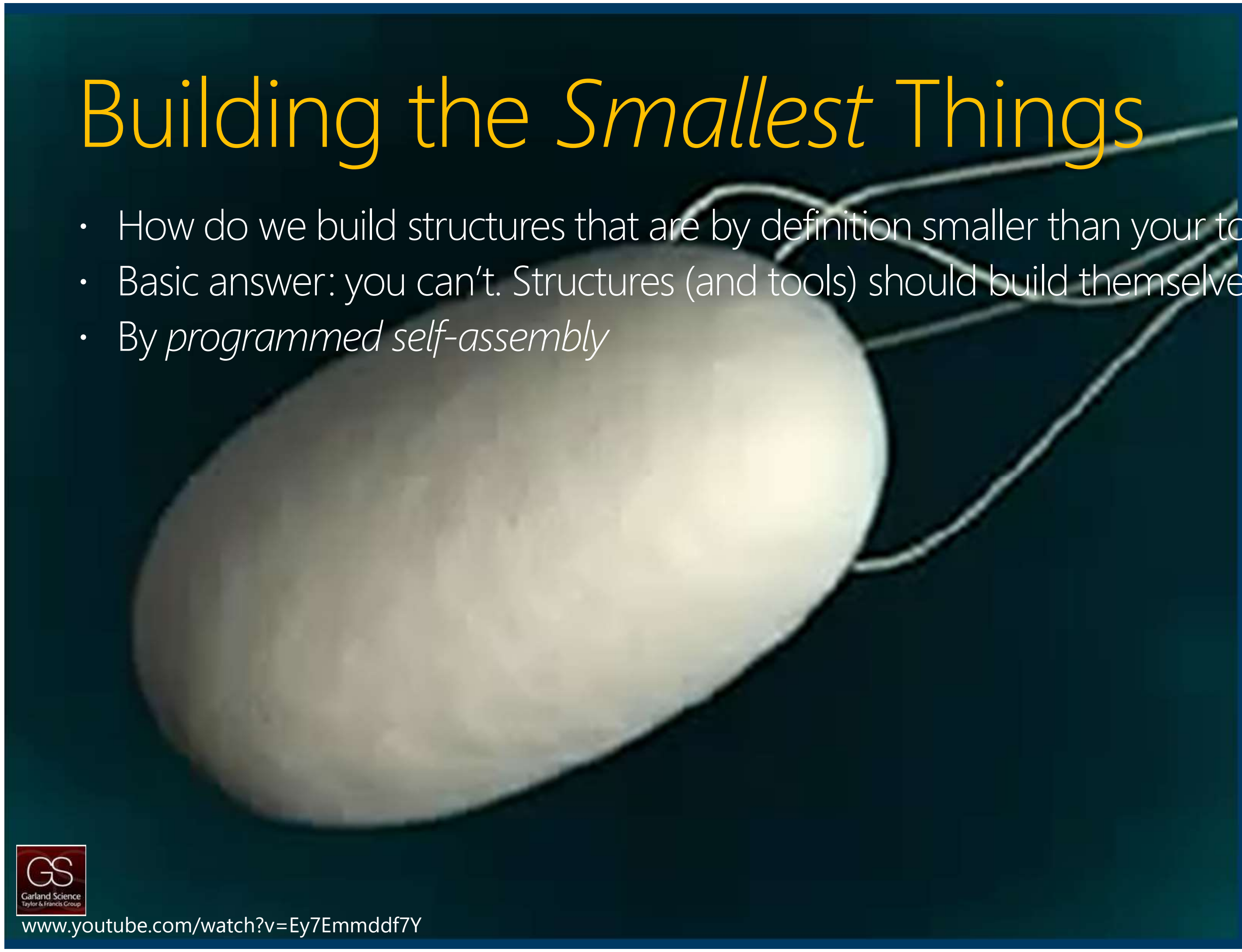
In both cases, we are now down to *molecules*



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

The SmidgION: A portable DNA sequencer that runs on an Iphone

Oxford Nanopore



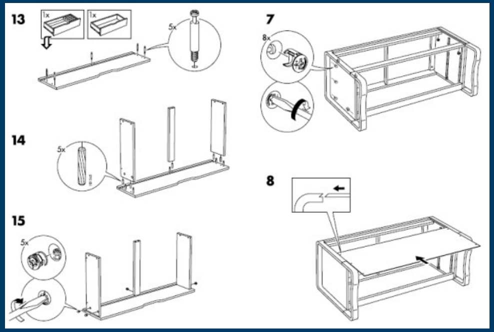
Building the *Smallest* Things

- 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*

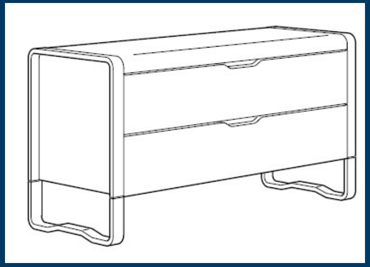
www.youtube.com/watch?v=Ey7Emmddf7Y

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

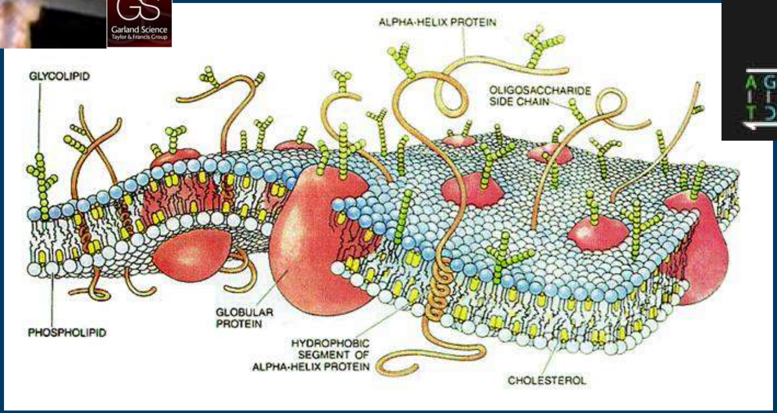


http://www.ikea.com/ms/en_US/customer_service/assembly_instructions.html

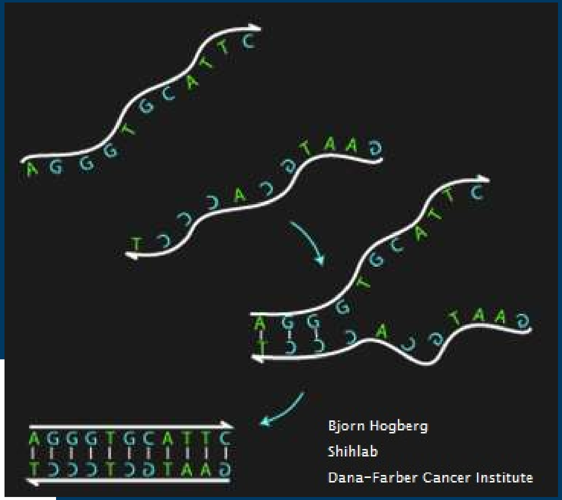
Programmed Self-Assembly

Proteins

Membranes


ALPHA-HELIX PROTEIN
GLYCOLIPID
OLIGOSACCHARIDE SIDE CHAIN
PHOSPHOLIPID
GLOBULAR PROTEIN
HYDROPHOBIC SEGMENT OF ALPHA-HELIX PROTEIN
CHOLESTEROL

DNA/RNA



Molecular Programming:
The Software Aspect

Smaller and smaller things can be programmed

We can program...

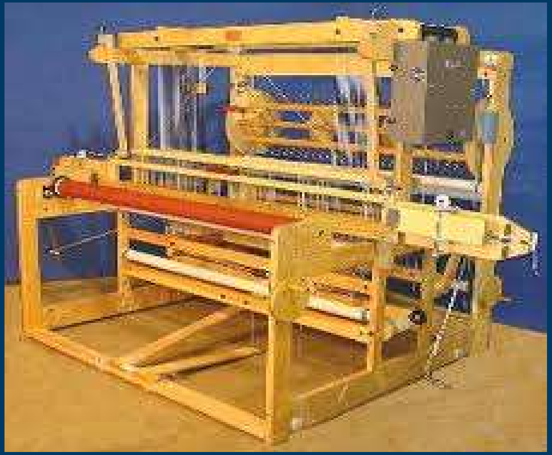
- Information
  - Completely!

$$\begin{tikzcd}
\text{Information} \arrow[d] \\
\text{Computing} \arrow[d] \\
\text{Information}
\end{tikzcd}$$

We can program...

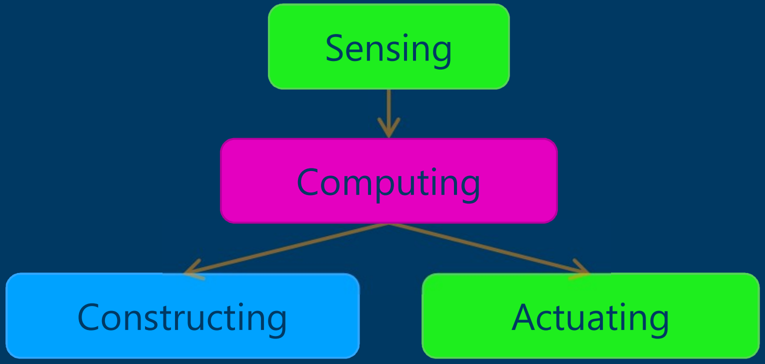
- Forces
  - Completely!
    (Modulo sensors/actuators)

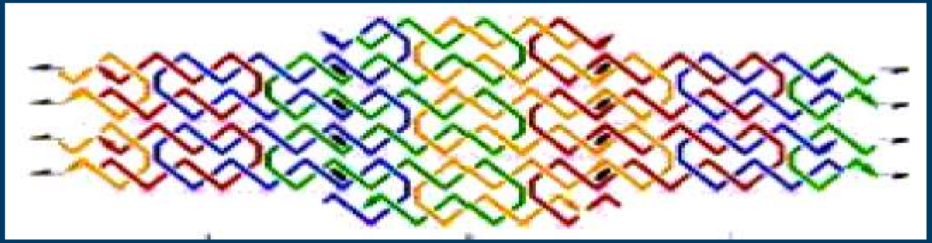
$$\begin{tikzcd}
\text{Sensing} \arrow[d] \\
\text{Computing} \arrow[d] \\
\text{Actuating}
\end{tikzcd}$$



We can program…

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

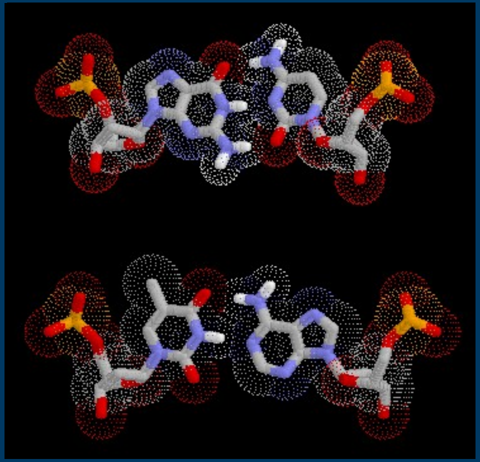




  - But DNA is an amazing *material*

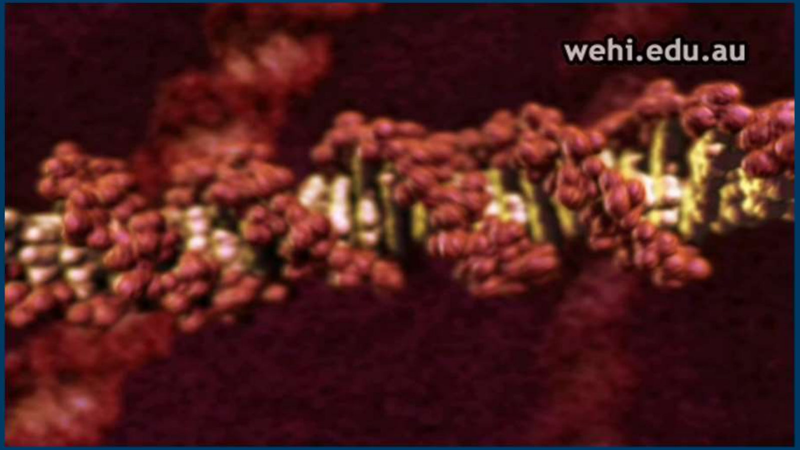
It's like a 3D printer without the printer!
[Andrew Hellington]

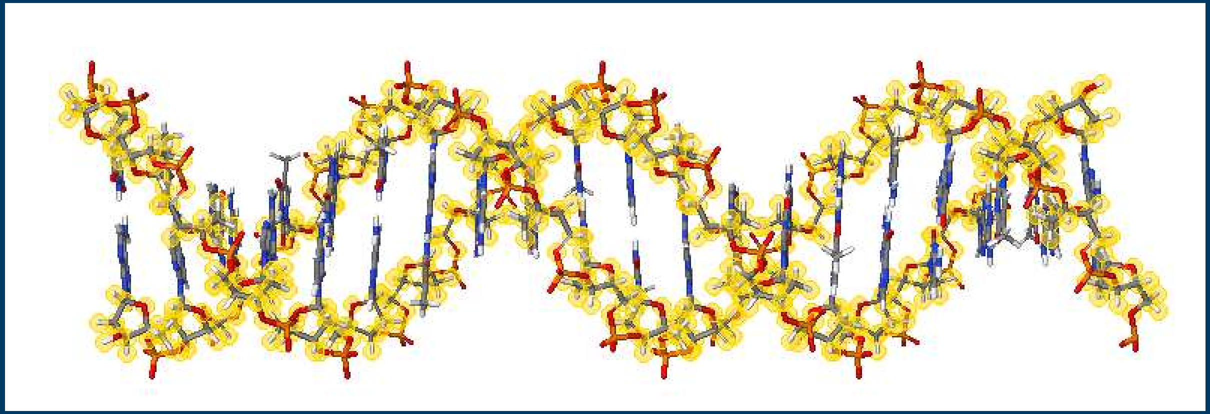
DNA



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)

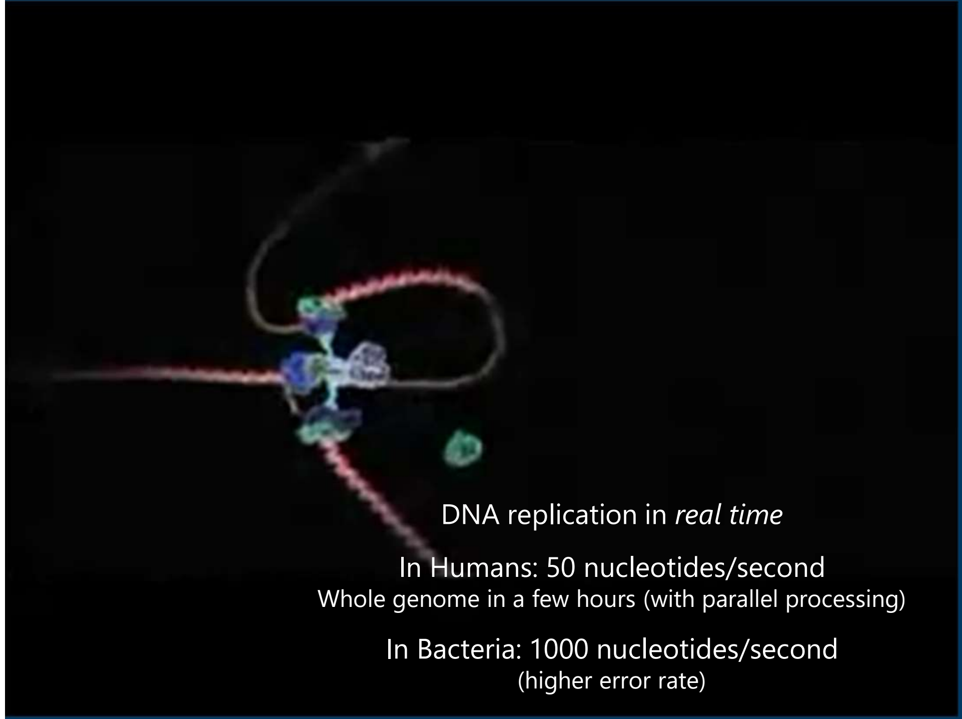
Structure

- DNA in each human cell:
  - 3 billion base pairs
  - 2 meters long, 2nm thick
  - 750 megabytes
  - folded into a 6μm ball,
    140 exabytes (million terabytes)/mm^3
    => *all* the data on the internet fits in a shoebox!
- A huge amount for a cell
  - Every time a cell replicates it has to copy *2 meters of DNA reliably*.
  - Or else!
- DNA in human body
  - 10 trillion cells
  - 133 Astronomical Units long
  - 7.5 octabytes
- DNA in human population
  - 20 million light years long

Andromeda Galaxy
2.5 million light years

Function

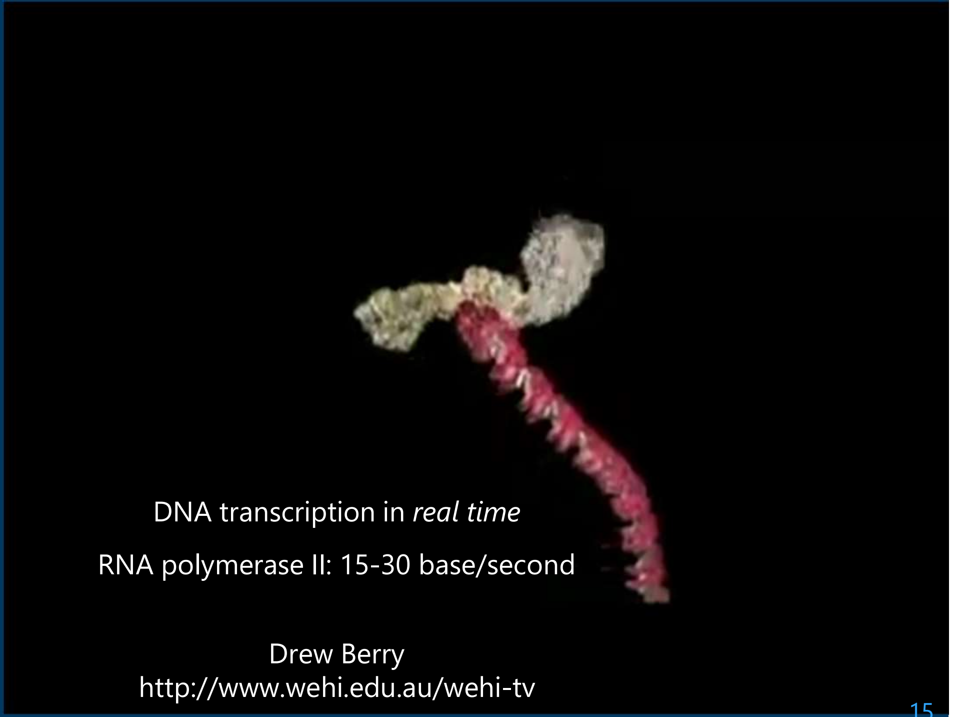
- DNA can support structural and computational complexity.



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*

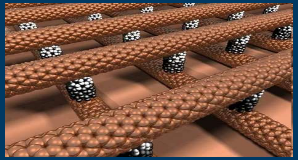
RNA polymerase II: 15-30 base/second

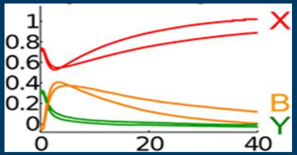
Drew Berry
http://www.wehi.edu.au/wehi-tv

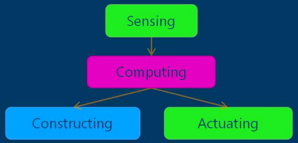
What is special about DNA?

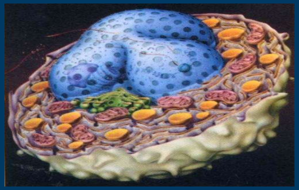
- There are many, many nanofabrication techniques and materials

- 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





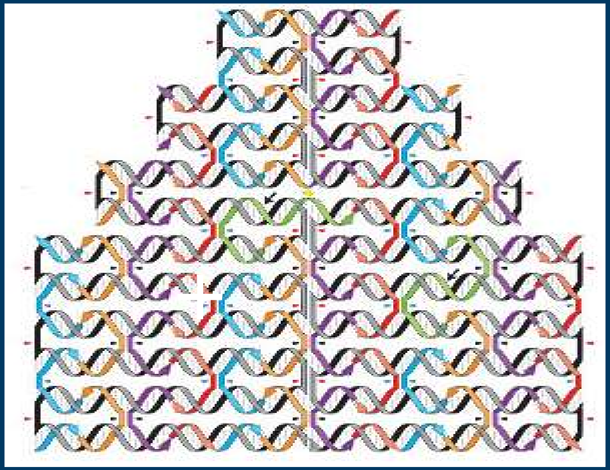



H.Lodish & al. Molecular Cell Biology 4th ed.

Organizing Any Matter

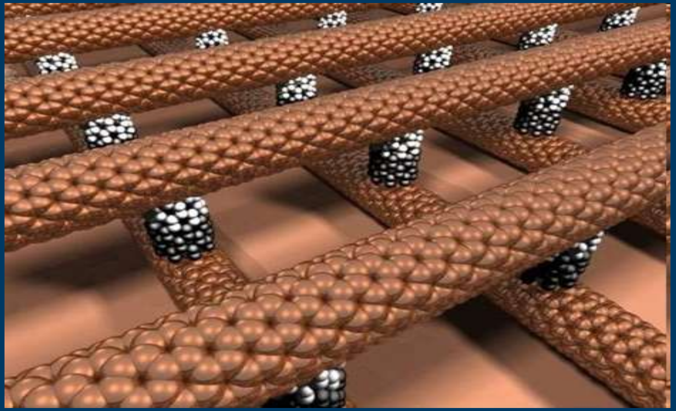
- Use one kind of programmable matter (e.g. DNA).
- To organize (almost) ANY matter through it.

6 nm grid of individually addressable DNA pixels



PWK Rothemund, *Nature* 440, 297 (2006)

→

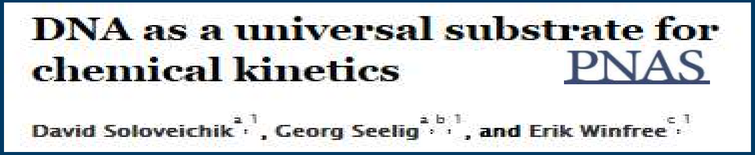


European Nanoelectronics Initiative Advisory Council

"What we are really making are tiny DNA circuit boards that will be used to assemble other components."
Greg Wallraff, IBM

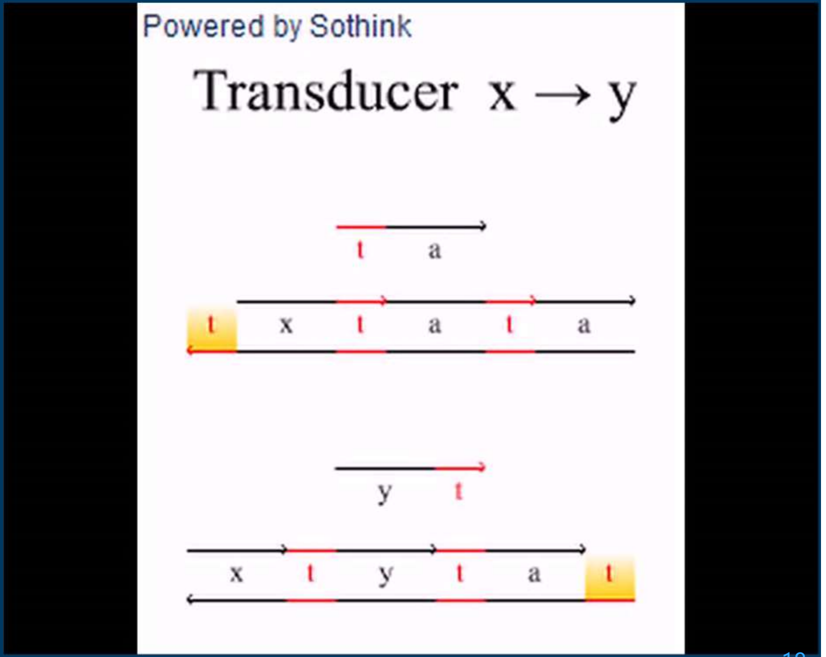
Executing Any Kinetics

- The kinetics of any finite network of chemical reactions, can be implemented (physically) with especially programmed DNA molecules.
- Chemical reactions as an executable programming language for dynamical systems!



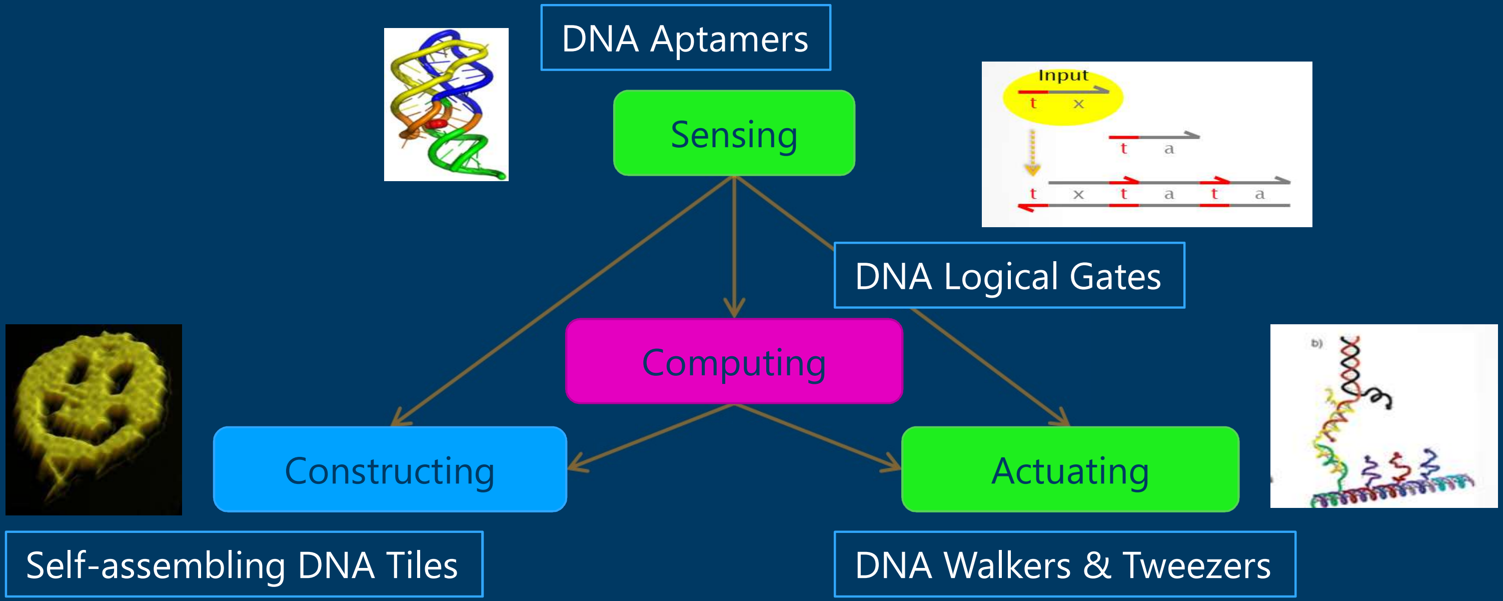
DNA as a universal substrate for chemical kinetics PNAS

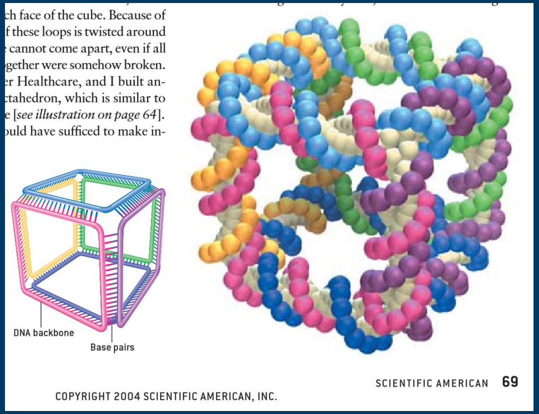
David Soloveichik[a,1], Georg Seelig[a,b,1], and Erik Winfree[c,1]



Building Nano-Control Devices

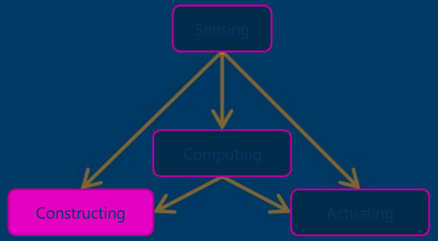
- All the components of nanocontrollers can already be built entirerly and solely with DNA, and interfaced to the environment




ch face of the cube. Because of
f these loops is twisted around
cannot come apart, even if all
gether were somehow broken.
r Healthcare, and I built an-
tahedron, which is similar to
[*see illustration on page 64*].
uld have sufficed to make in-

SCIENTIFIC AMERICAN 69

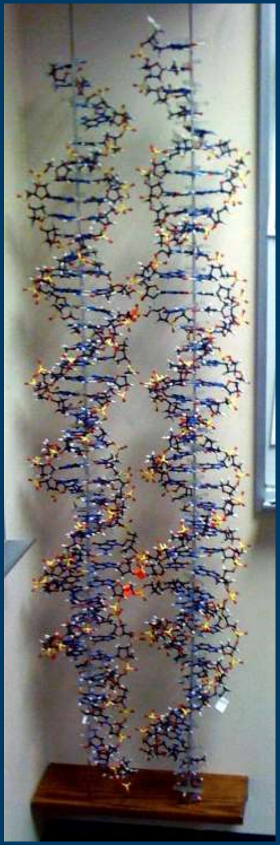
COPYRIGHT 2004 SCIENTIFIC AMERICAN, INC.

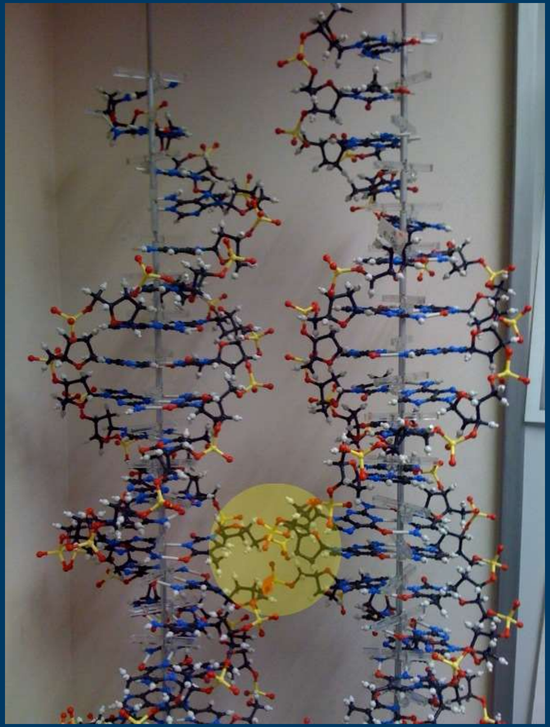


Constructing

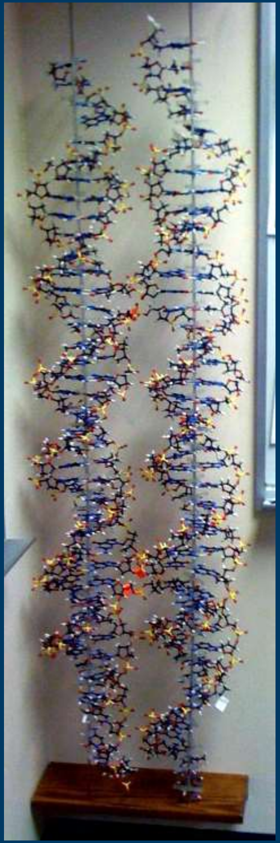
• • •

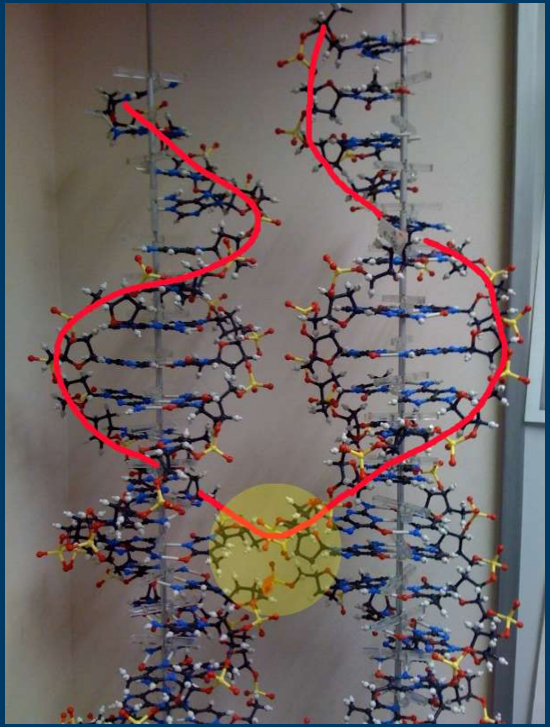
Crosslinking



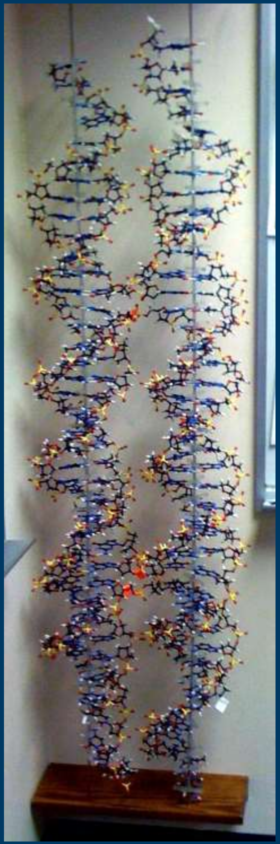


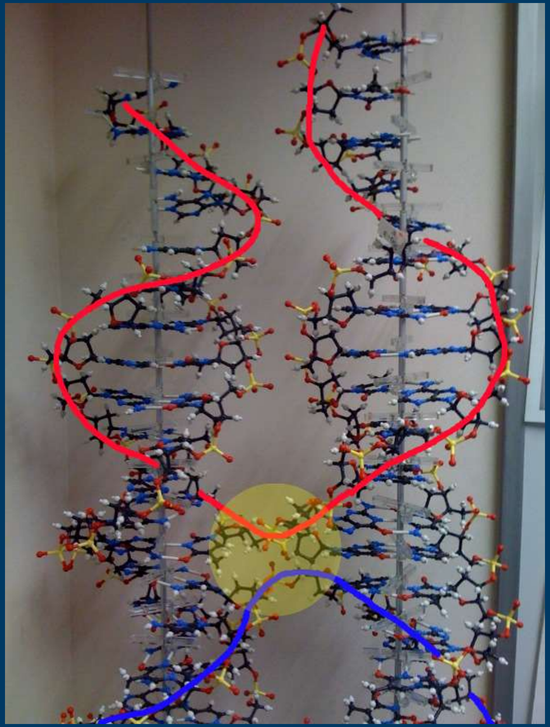
Crosslinking



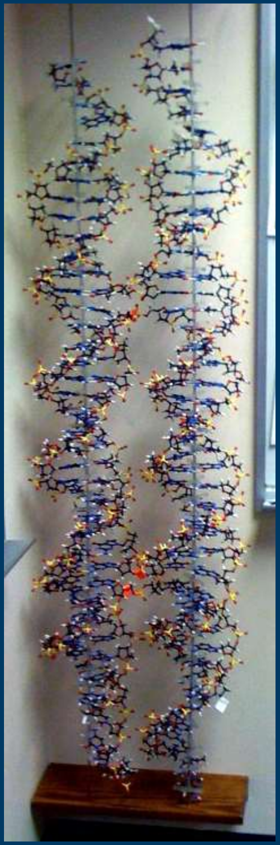


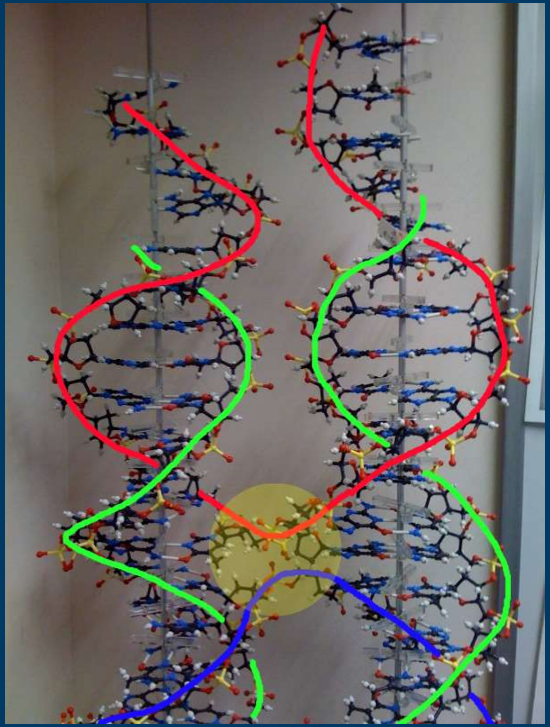
Crosslinking



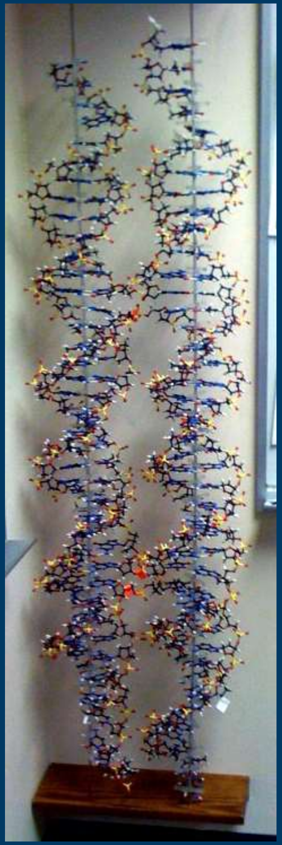


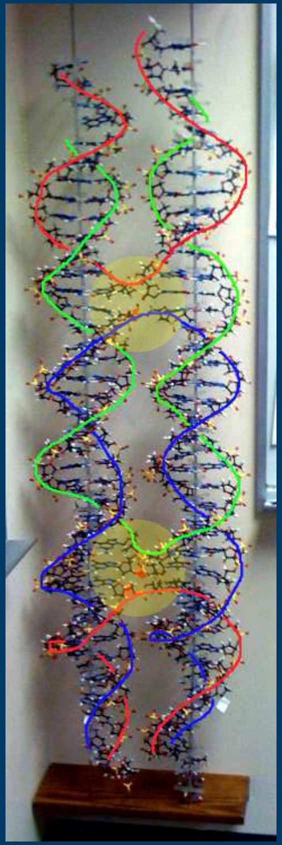
Crosslinking



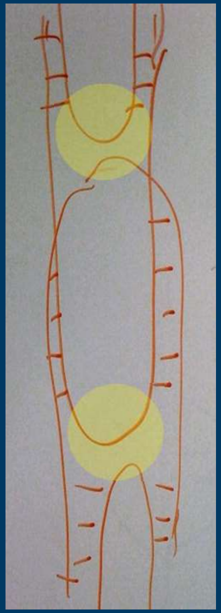


Crosslinking





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

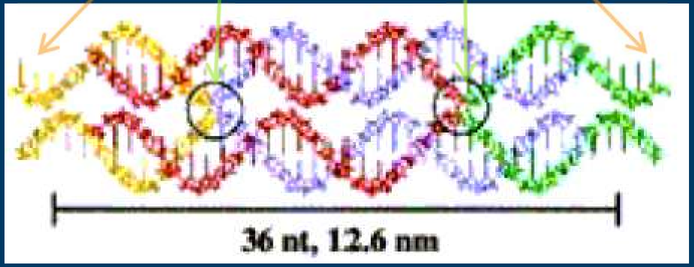


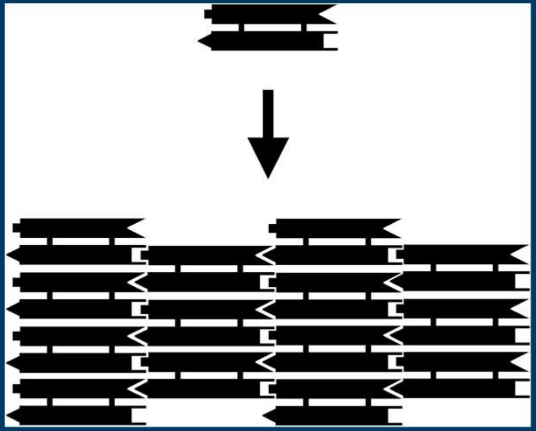
In engineering, crosslinking is the key to using DNA as a construction material.

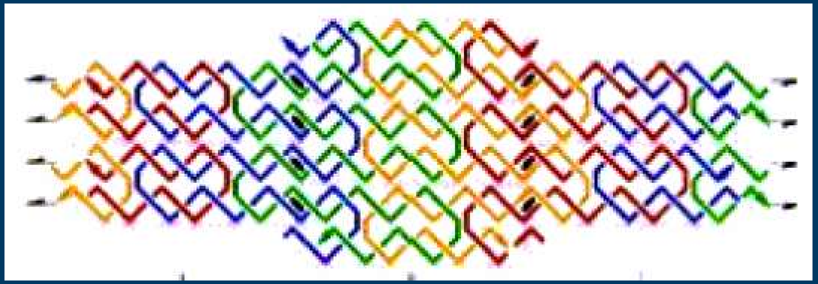
DNA Tiling

4 sticky ends

crosslinking







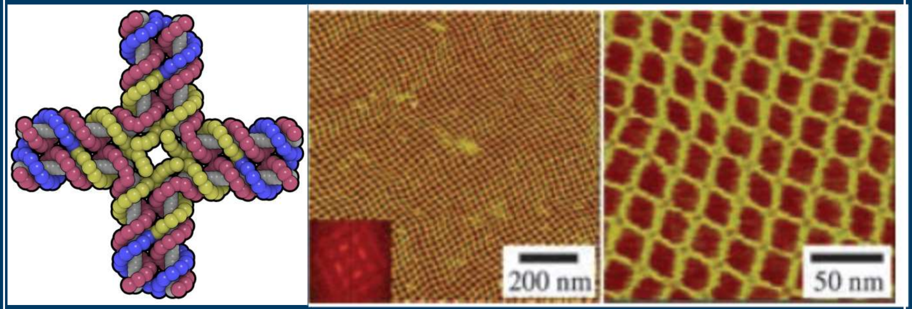
Construction and manipulation of DNA tiles in free space

Pankhudi

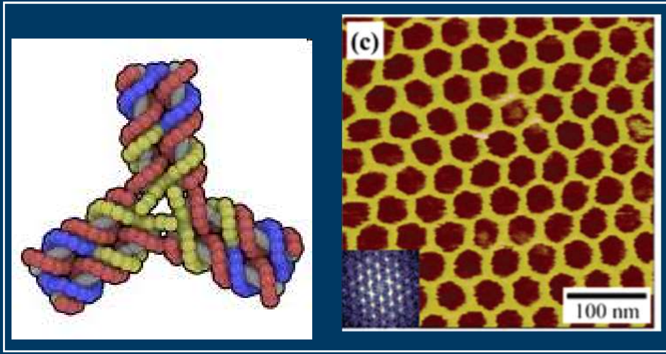
2D DNA Lattices

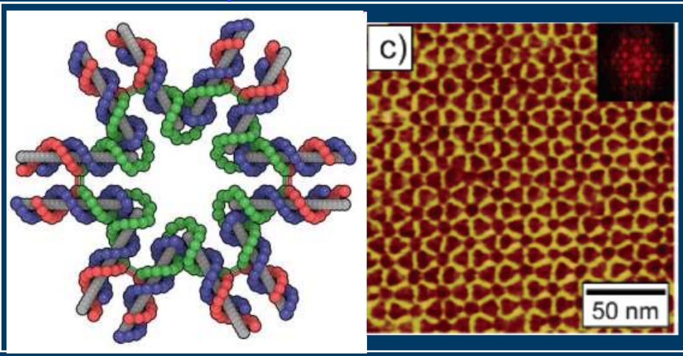
Chengde Mao

Purdue University, USA



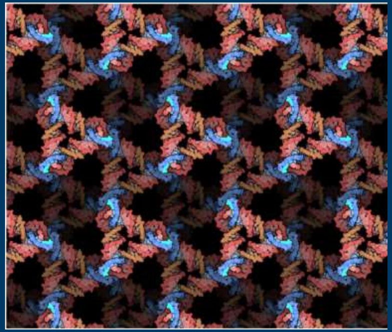
N-point Stars

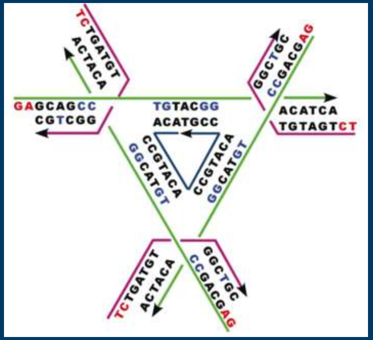




3D DNA Structures

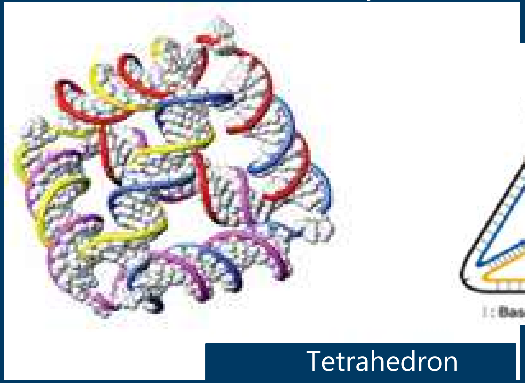
Ned Seeman
NYU

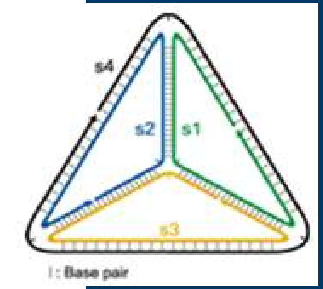




3D Cyrstal

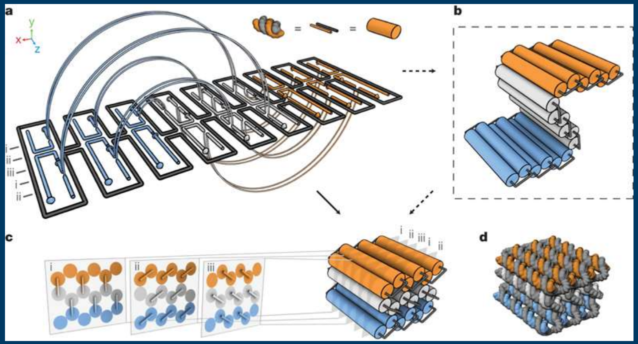
AndrewTuberfield
Oxford

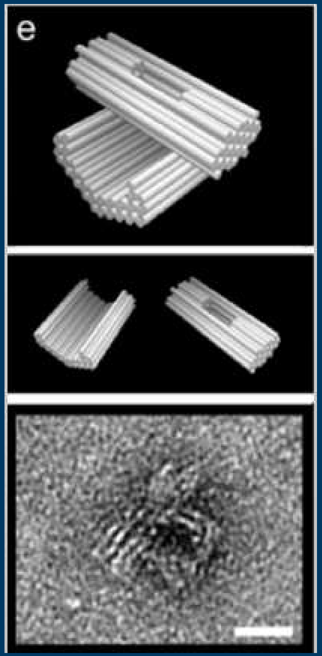


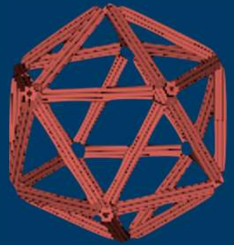


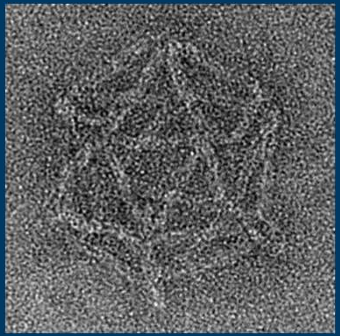
Tetrahedron

CADnano

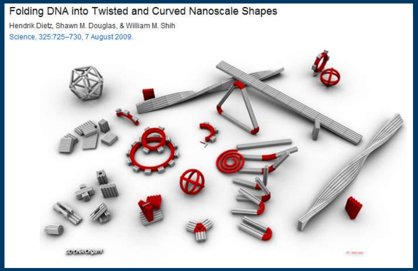








William Shih
Harvard


Folding DNA into Twisted and Curved Nanoscale Shapes
Hendrik Dietz, Shawn M. Douglas, & William M. Shih
Science, 325:725–730, 7 August 2009.

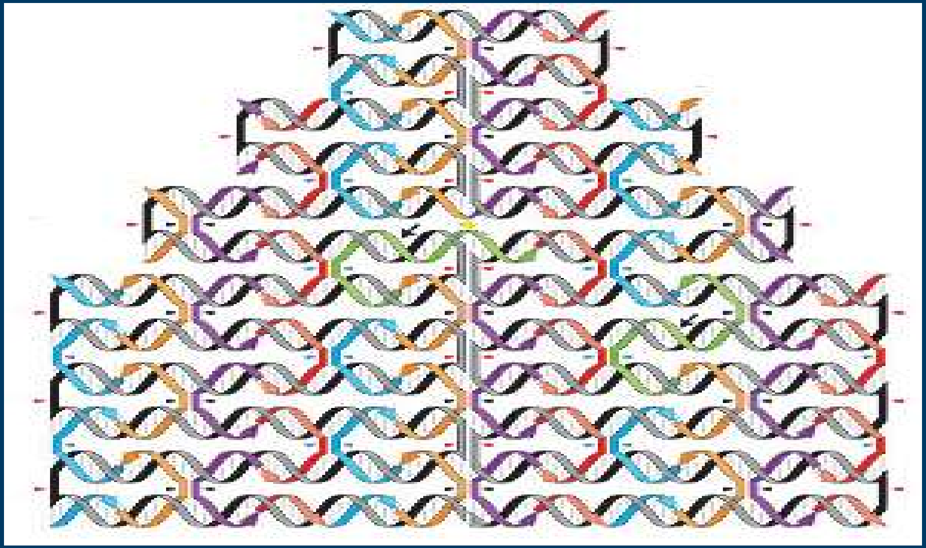
https://www.youtube.com/watch?v=Ek-FDPymyyg

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)

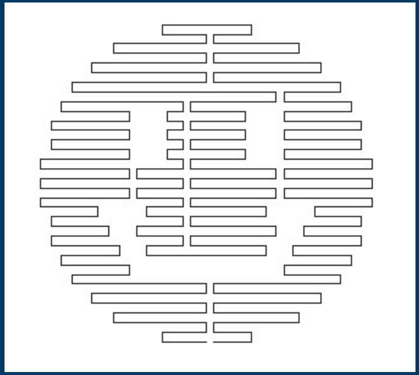
DNA Origami

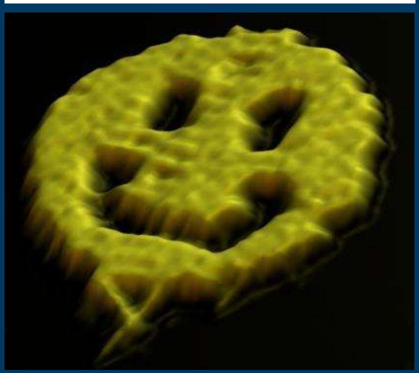
Folding long (7000bp) naturally occurring (viral) ssDNA
By lots of short 'staple' strands that constrain it

Paul W K Rothemund
California Institute of Technology


PWK Rothemund, *Nature* 440, 297 (2006)

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



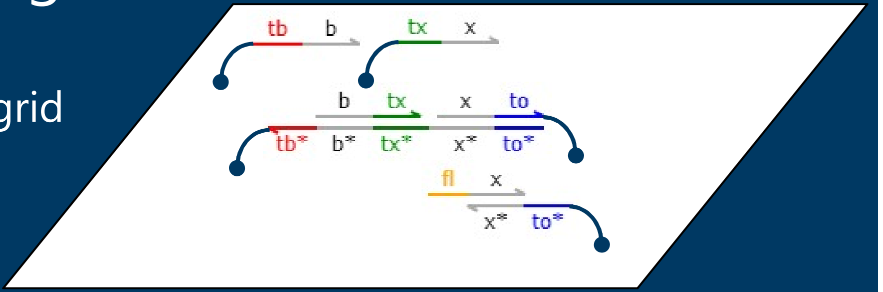

Paul Rothemund's "Disc with three holes" (2006)

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

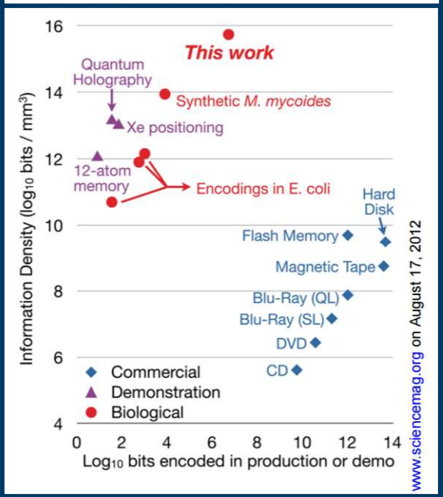


Dalchau, Chandran, Gopalkrishnan, Reif, Phillips. 2014

DNA Storage (Read/Write)

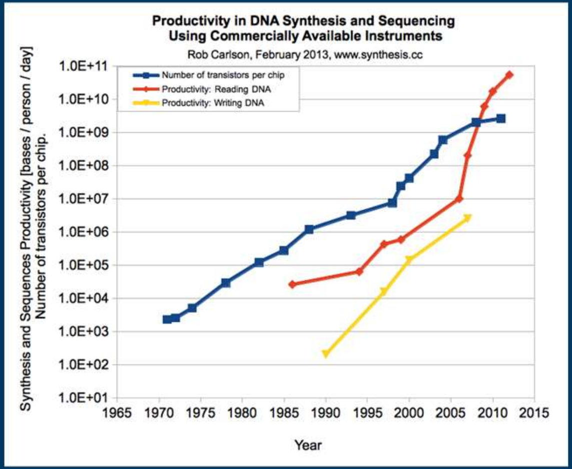
Next-Generation Digital Information Storage in DNA

George M. Church,[1,2] Yuan Gao,[3] Sriram Kosuri[1,2*]



Information-rich physical structures can be used for storage.

DNA has a data density of 140 exabytes (1.4×10^{20} bytes) per mm^3 compared to state-of the art storage media that reaches ~500 megabytes (5×10^8 bytes) per mm^3
DNA has been shown to be stable for millions of years

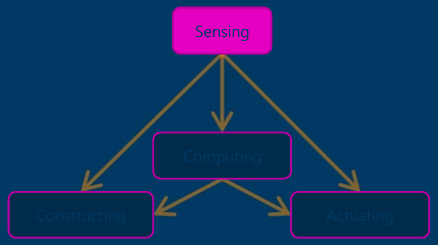


The Pace and Proliferation of Biological Technologies

March 4, 2004 by Rob Carlson

We have machines that can read (sequence) and write (synthesize) DNA. The Carslon 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]

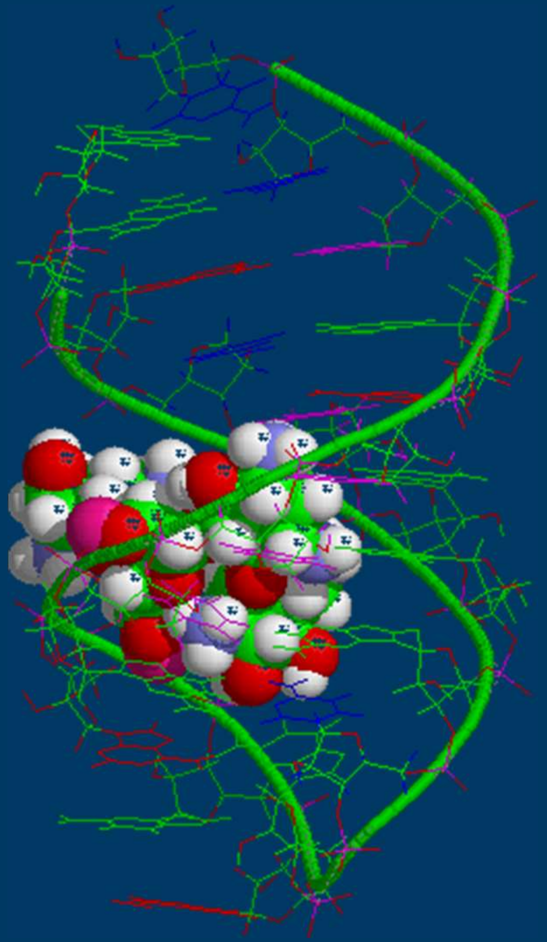


Sensing

• • •

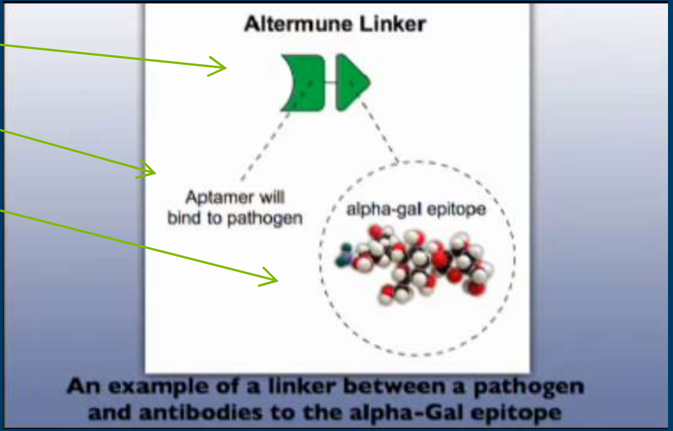
Aptamers

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



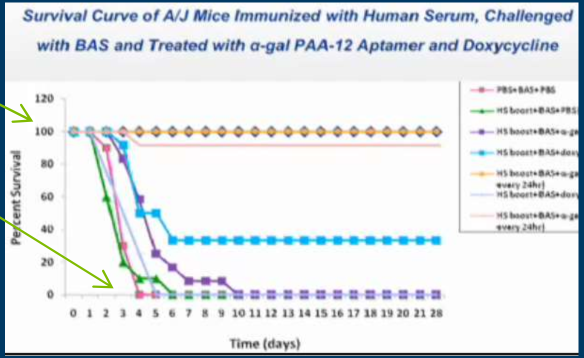
Pathogen Spotlights

- 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!



Altermune Linker
Aptamer will bind to pathogen
alpha-gal epitope
An example of a linker between a pathogen and antibodies to the alpha-Gal epitope

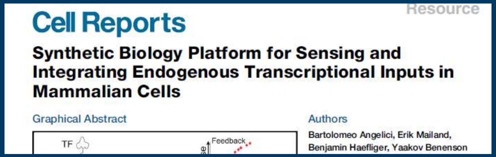
- Result: instant immunity
  - Mice poisoned with Anthrax plus aptamer (100% survival)
  - Mice poinsoned with Anthrax (not so good)



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

Transcriptional Sensors

"One of the goals of synthetic biology is to develop programmable artificial gene networks that can transduce multiple endogenous molecular cues to precisely control cell behavior. "


Cell Reports

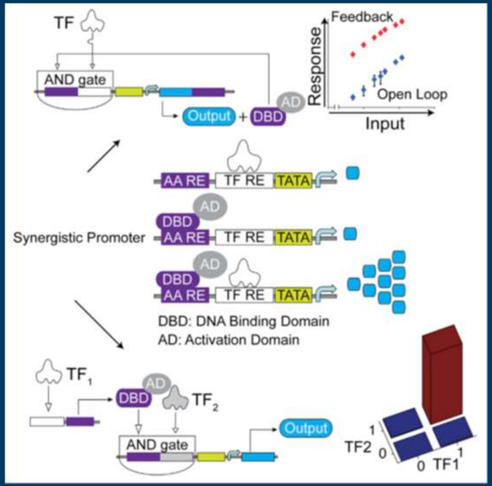
Resource

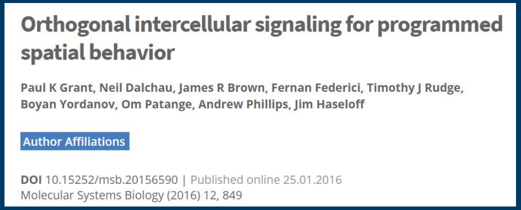
Synthetic Biology Platform for Sensing and Integrating Endogenous Transcriptional Inputs in Mammalian Cells

Graphical Abstract

Authors

Bartolomeo Angelici, Erik Mailand, Benjamin Haefliger, Yaakov Benenson



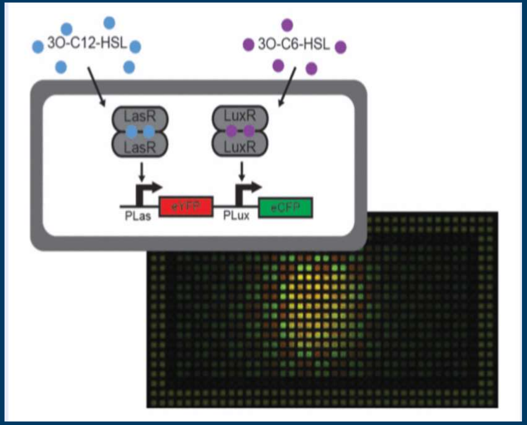

Orthogonal intercellular signaling for programmed spatial behavior

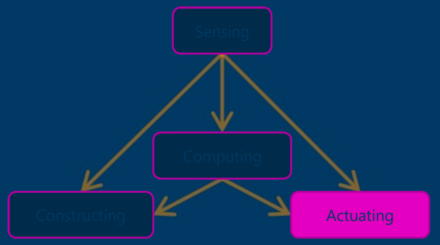
Paul K Grant, Neil Dalchau, James R Brown, Fernan Federici, Timothy J Rudge, Boyan Yordanov, Om Patange, Andrew Phillips, Jim Haseloff

Author Affiliations

DOI 10.15252/msb.20156590 | Published online 25.01.2016

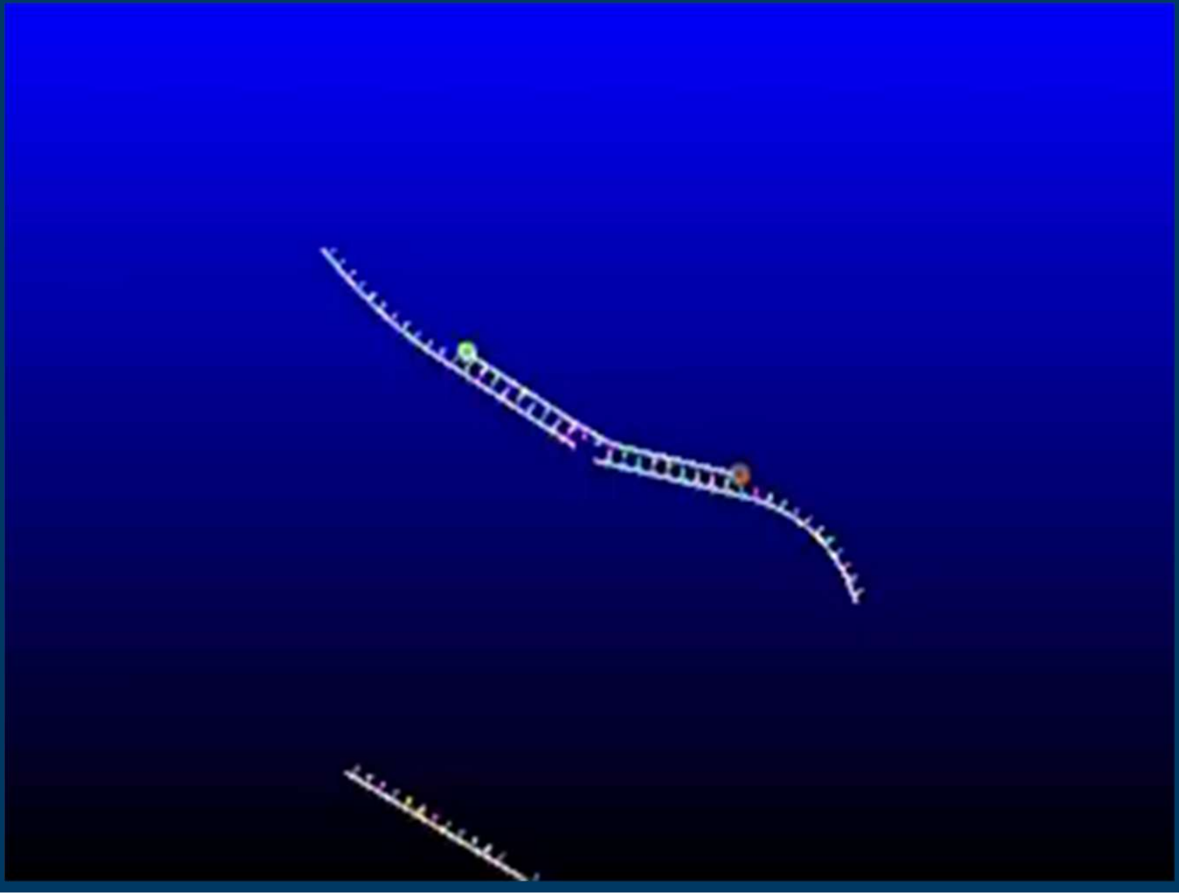
Molecular Systems Biology (2016) 12, 849

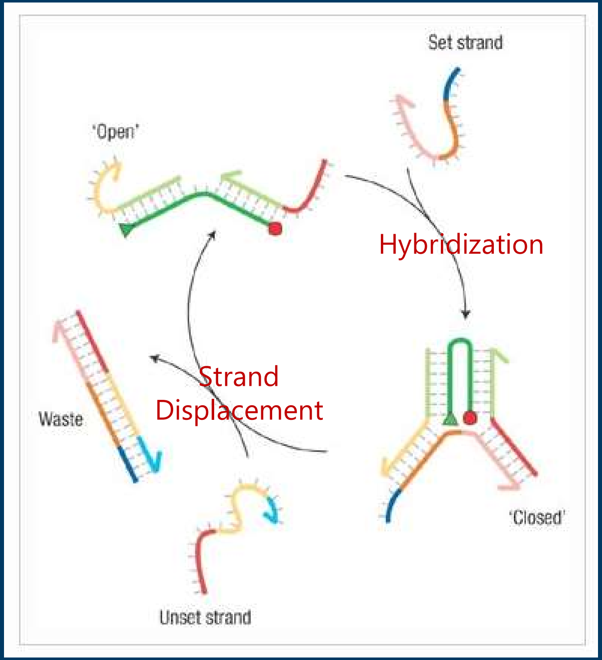




Actuating

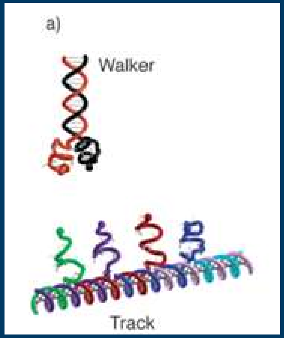
DNA Tweezers

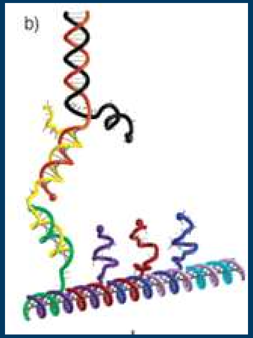


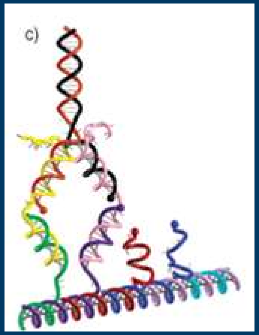


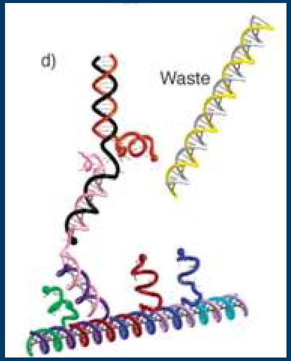
DNA nanomachines
Jonathan Bath & Andrew J. Turberfield
Nature Nanotechnology **2**, 275 - 284 (2007)
doi:10.1038/nnano.2007.104

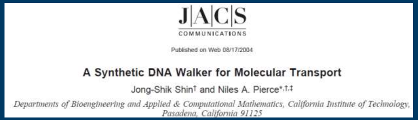
DNA Walkers










J|A|C|S
COMMUNICATIONS

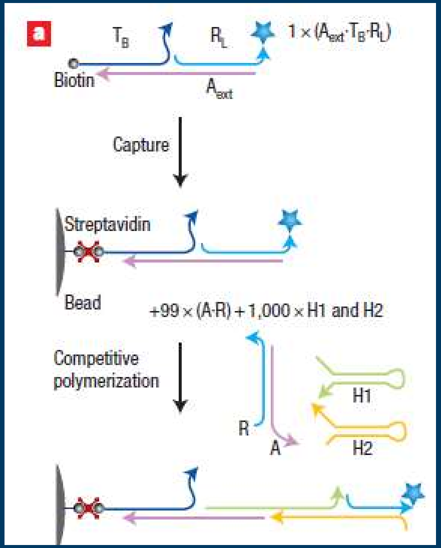
Published on Web 08/17/2004

A Synthetic DNA Walker for Molecular Transport

Jong-Shik Shin[†] and Niles A. Pierce[*,†,‡]

Departments of Bioengineering and Applied & Computational Mathematics, California Institute of Technology, Pasadena, California 91125

Polymerization Motor


a
T_B
R_L
$1 \times (A_{ext} \cdot T_B \cdot R_L)$
Biotin
A_{ext}
Capture
Streptavidin
Bead
$+99 \times (A \cdot R) + 1{,}000 \times$ H1 and H2
Competitive polymerization
R
A
H1
H2

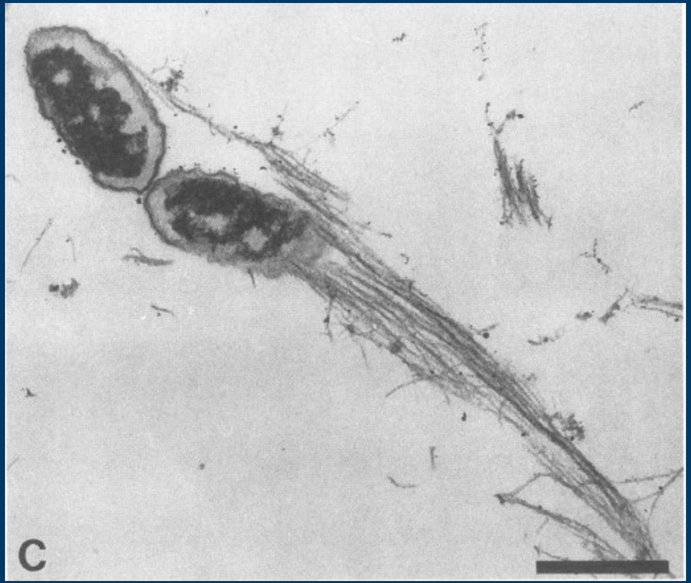
An autonomous polymerization motor powered by DNA hybridization

SUVIR VENKATARAMAN[1], ROBERT M. DIRKS[1], PAUL W. K. ROTHEMUND[2,3], ERIK WINFREE[2,3] AND NILES A. PIERCE[1,4*]

Triggered amplification by hybridization chain reaction

Robert M. Dirks† and Niles A. Pierce‡,§

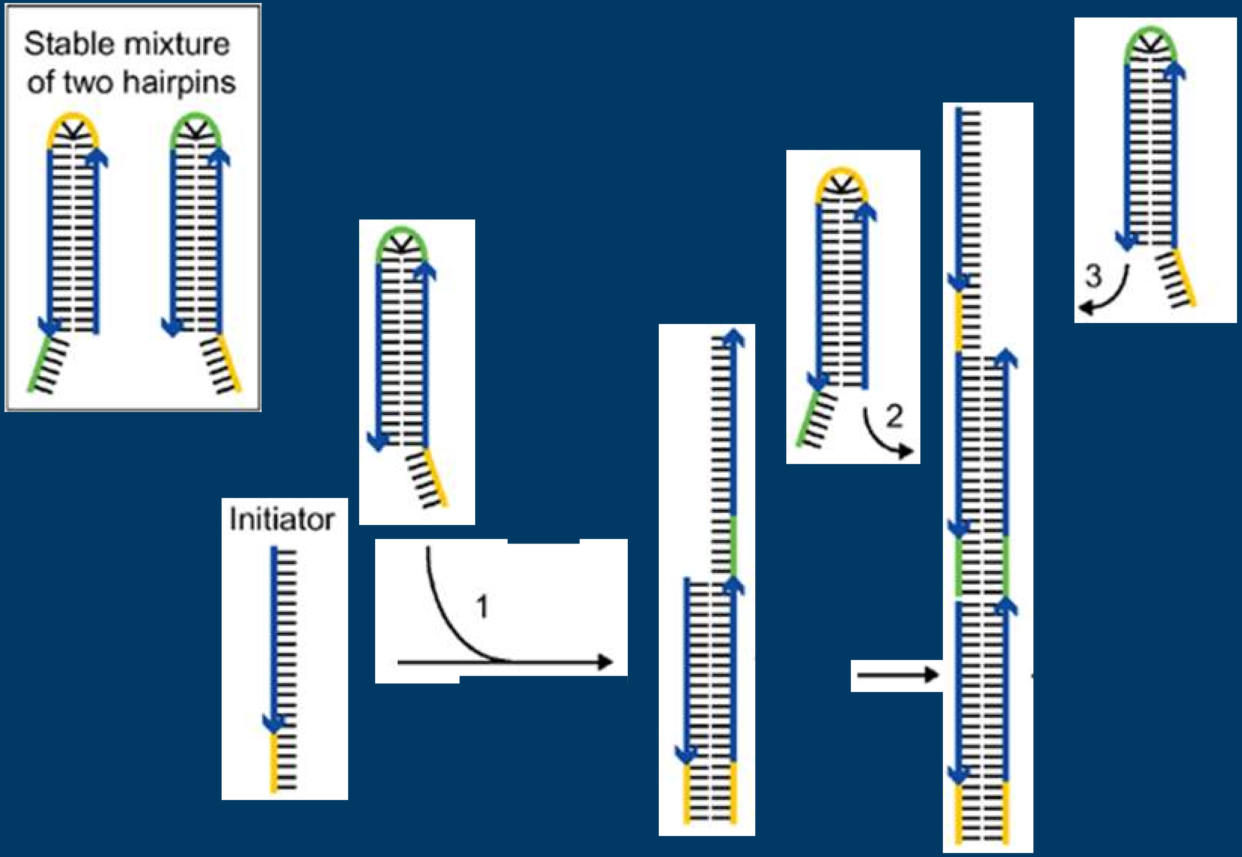
Rickettsia (spotted fever)

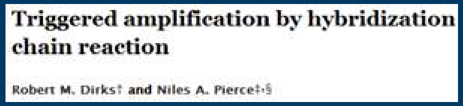


Directional Actin Polymerization Associated with Spotted Fever Group Rickettsia Infection of Vero Cells

ROBERT A. HEINZEN, STANLEY F. HAYES, MARIUS G. PEACOCK, AND TED HACKSTADT*

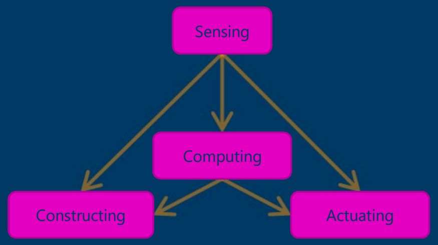
Hybridization Chain Reaction





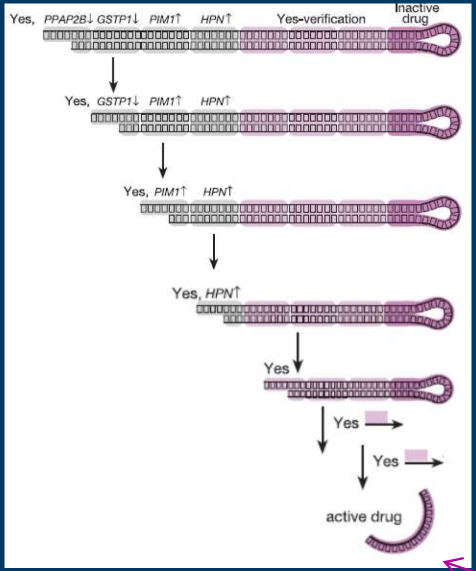
Triggered amplification by hybridization chain reaction

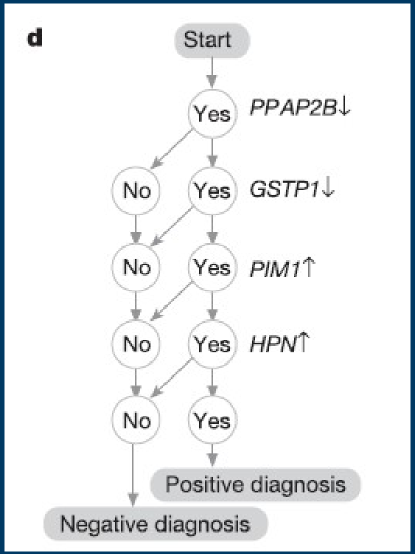
Robert M. Dirks† and Niles A. Pierce‡,§



Curing

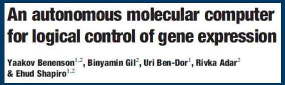
Computational Drugs





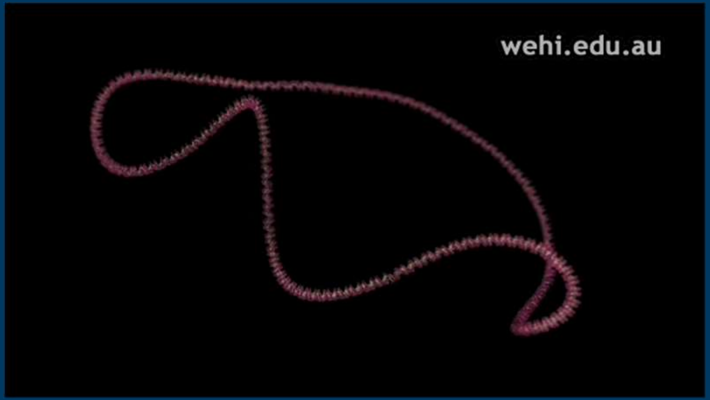
Vitravene (GCGTTTGCTCTTCTTCTTGCG)

- An automaton sequentially reading the string PPAP2B, GSTP1, PIM1, HPS (known cancer indicators) and sequentially cutting the DNA hairpin until a ssDNA drug (Vitravene) is released.

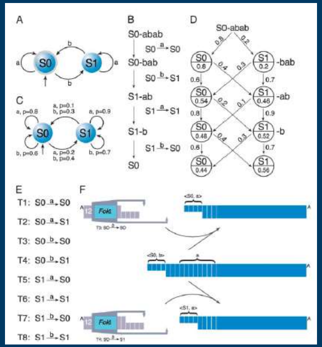


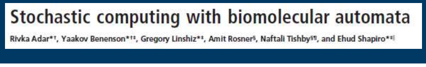
An autonomous molecular computer for logical control of gene expression

Yaakov Benenson[1,2], Binyamin Gil[2], Uri Ben-Dor[1], Rivka Adar[2] & Ehud Shapiro[1,2]



Based on restriction enzymes





Stochastic computing with biomolecular automata

Rivka Adar*†, Yaakov Benenson*†‡, Gregory Linshiz*‡, Amit Rosner§, Naftali Tishby§¶, and Ehud Shapiro*‡‖

Interfacing to Biology

- A doctor in each cell

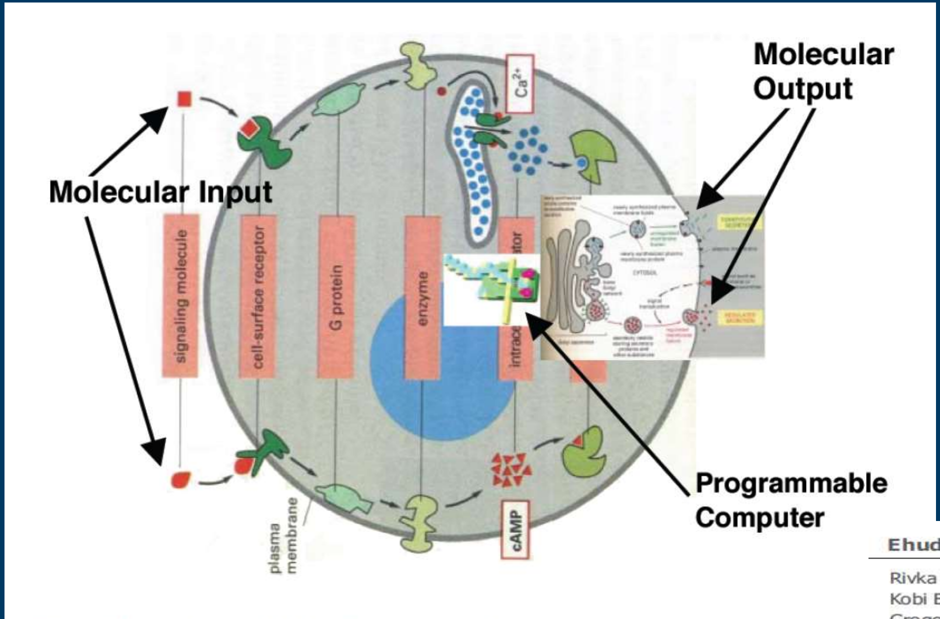
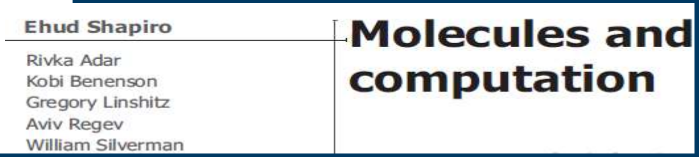


Fig. 1 Medicine in 2050: "Doctor in a Cell"



Ehud Shapiro
Rivka Adar
Kobi Benenson
Gregory Linshitz
Aviv Regev
William Silverman

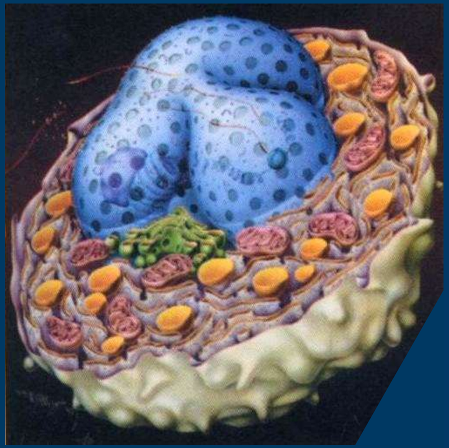
Molecules and computation

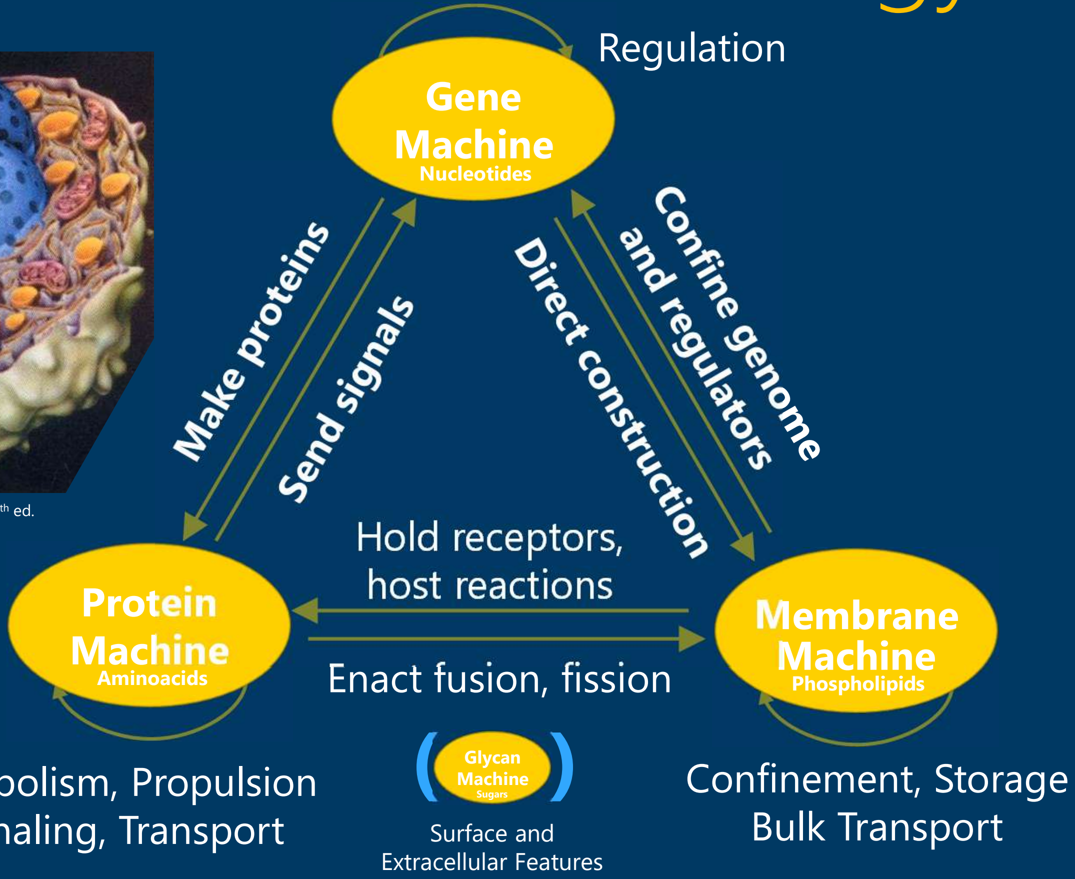
~2002

Molecular Programming:
The Biological Aspect

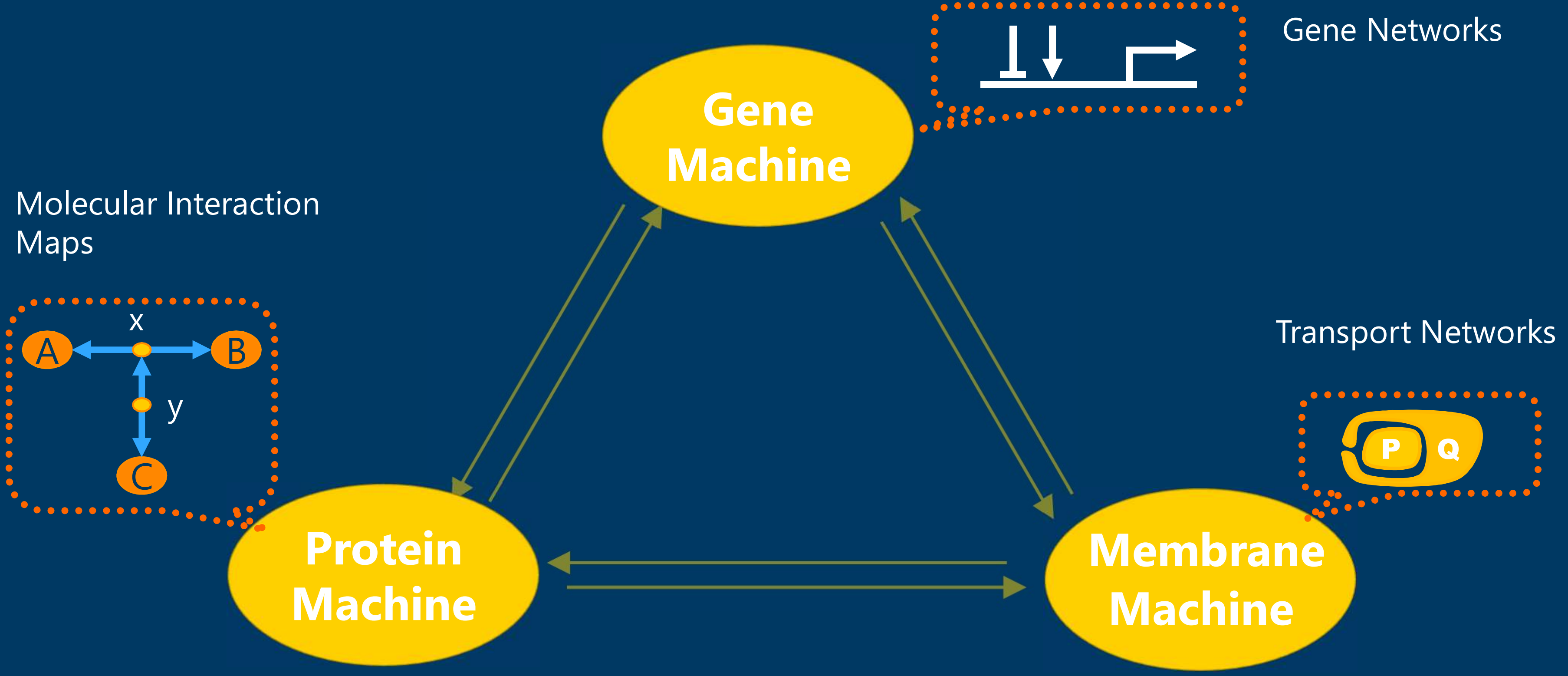
Biological systems are already ‘molecularly programmed’

Abstract Machines of Biology


H.Lodish & al. Molecular Cell Biology 4th ed.



Biological Languages



But …

- Biology is programmable, but (mostly) not by us!

- Still work in progress:
  - Gene networks are being programmed in synthetic biology, but using existing ‘parts’
  - Protein networks are a good candidate, but we cannot yet effectively design proteins
  - Transport networks are being investigated for programming microfluidic devices that manipulate vesicles

Molecular Languages

… that we can execute

(more easily than what nature provides)

Our Programming Language: Chemistry

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

Chemical Programming Examples

specification	*program*	
Y := min(X1, X2)	X1 + X2 -> Y	
Y := max(X1, X2)	X1 -> L1 + Y X2 -> L2 + Y L1 + L2 -> K Y + K -> 0	max(X1,X2)= (X1+X2)-min(X1,X2) (but is not computed "sequentially": it is a form of concurrent computation)

chemical reaction network

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?

Molecular Programming with DNA

Building the cores of programmable molecular controllers

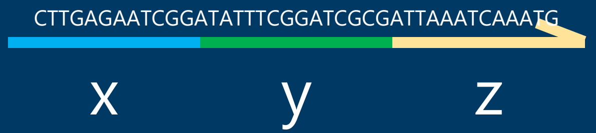
The role of DNA Computing

- 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

Domains

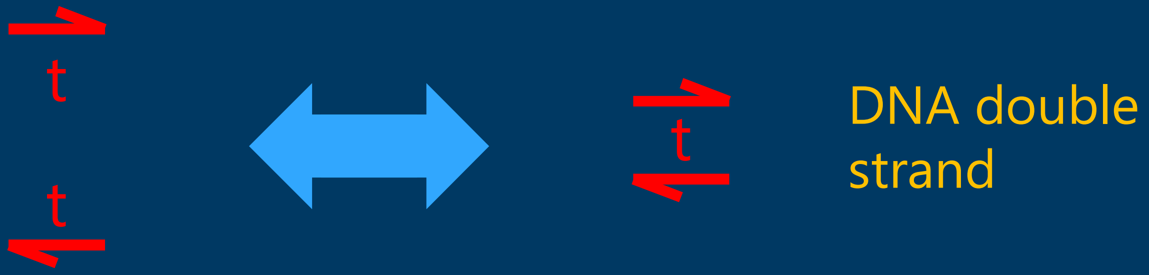
- Subsequences on a DNA strand are called domains
  - *provided* they are “independent” of each other



oriented DNA
single strand

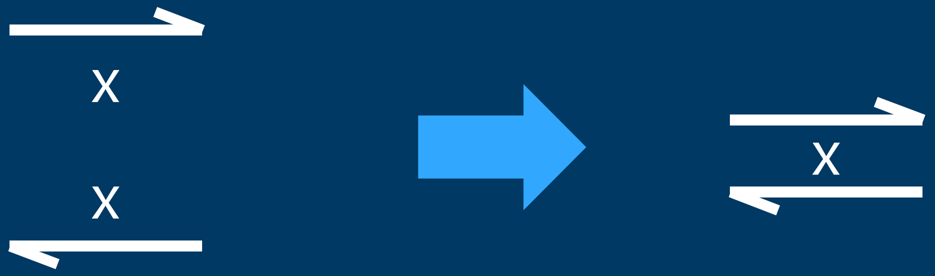
- Differently named domains must not hybridize
  - With each other, with each other’s complement, with subsequences of each other, with concatenations of other domains (or their complements), etc.

Short Domains



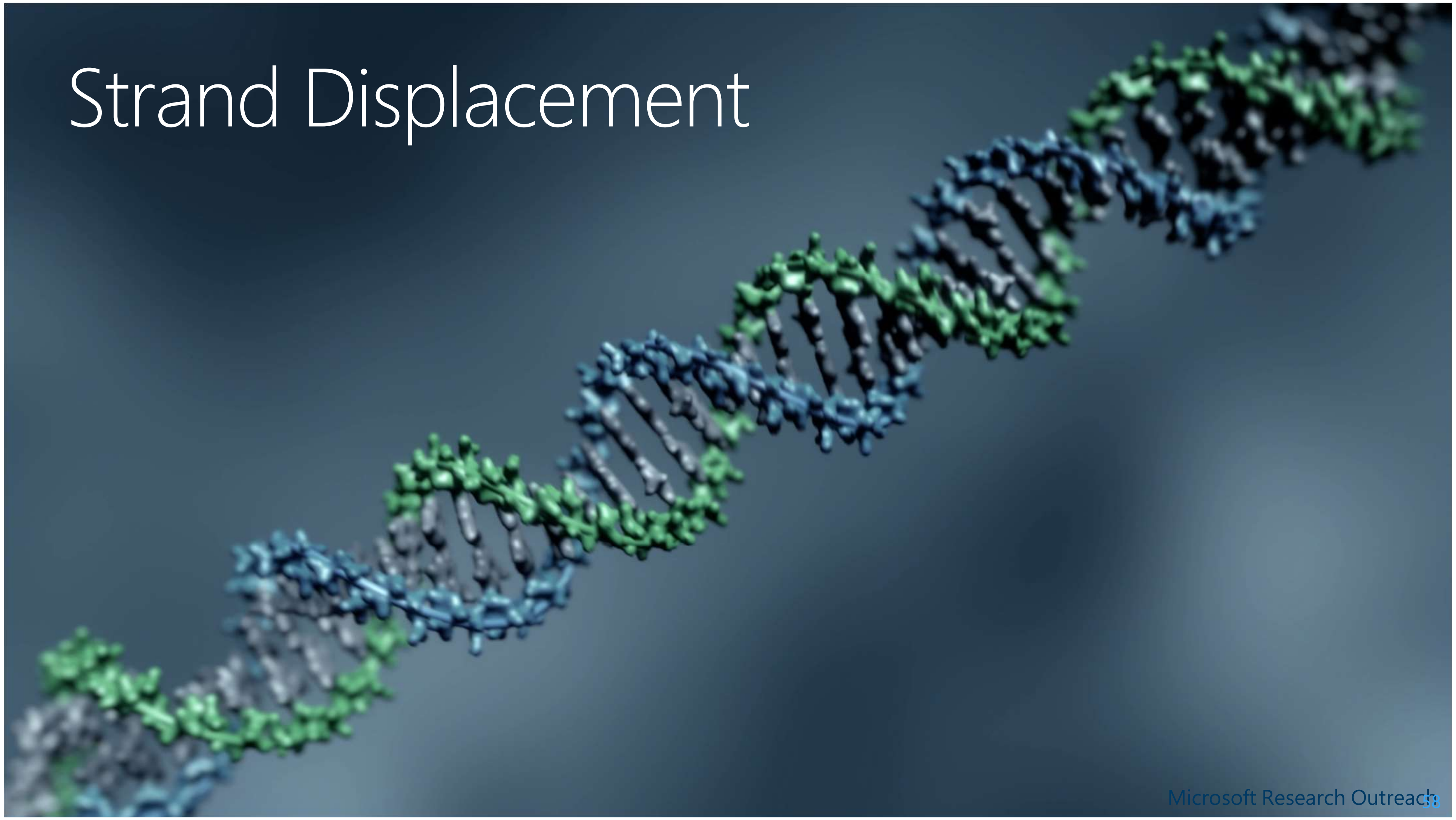
Reversible Hybridization

Long Domains

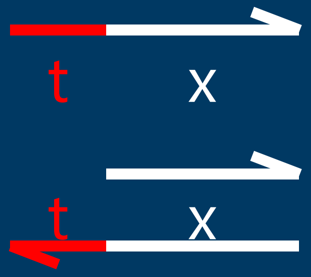


Irreversible Hybridization

Strand Displacement

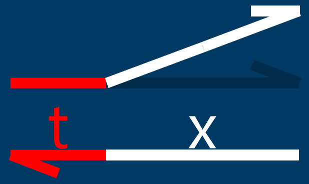


Strand Displacement



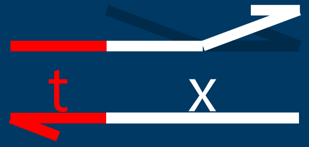
“Toehold Mediated”

Strand Displacement



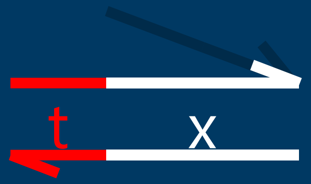
Toehold Binding

Strand Displacement



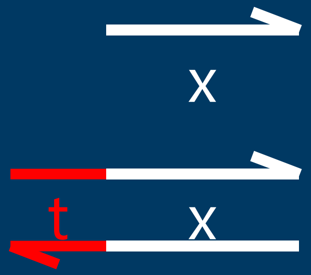
Branch Migration

Strand Displacement



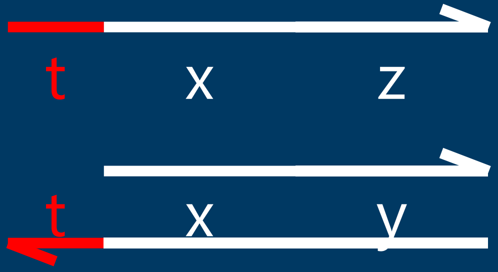
Displacement

Strand Displacement

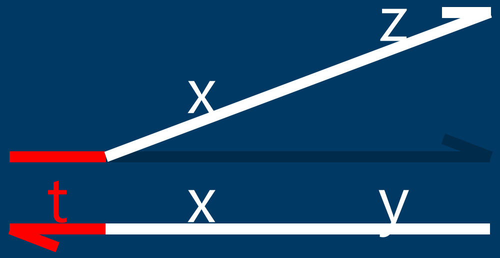


Irreversible release

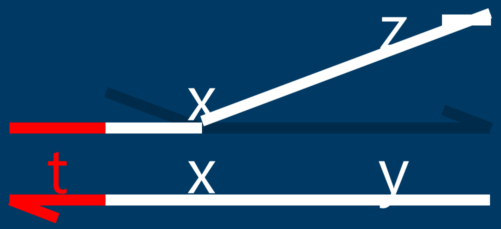
Bad Match



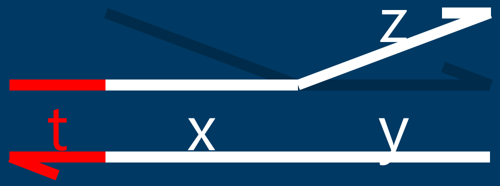
Bad Match



Bad Match



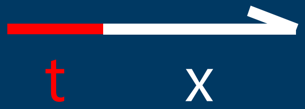
Bad Match



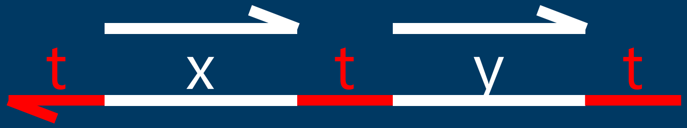
Cannot proceed
Hence will undo

Two-Domain Architecture

- Signals: 1 toehold + 1 recognition region



- Gates: “top-nicked double strands” with open toeholds



Garbage collection “built into” the gate operation

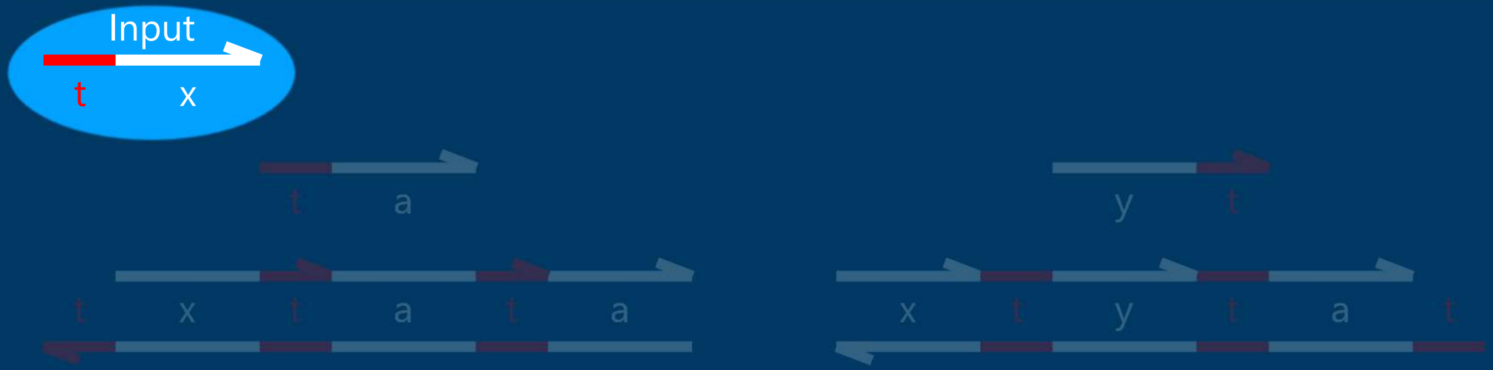
Two-Domain DNA Strand Displacement

Luca Cardelli

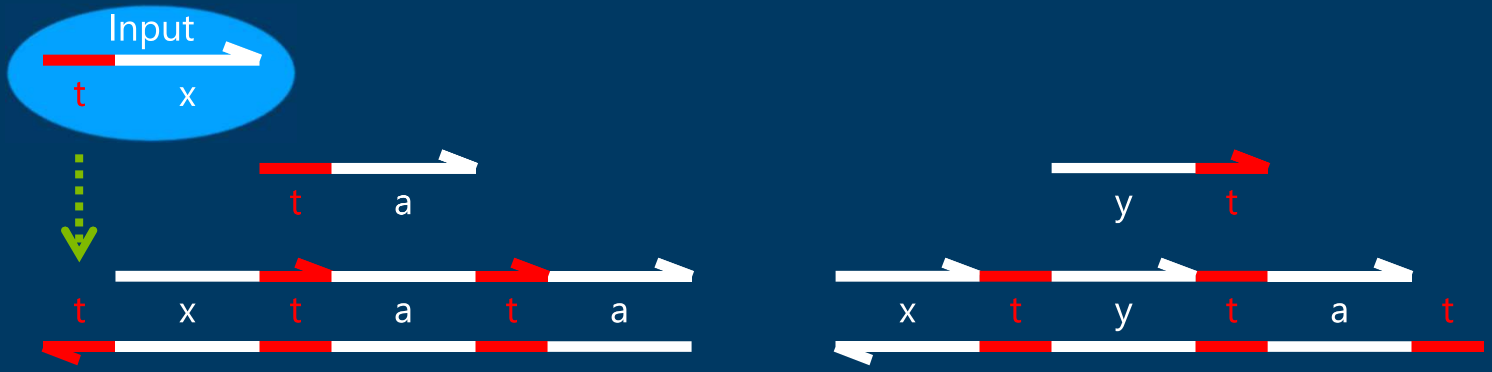
In S. B. Cooper, E. Kashefi, P. Panangaden (Eds.): Developments in Computational Models (DCM 2010). EPTCS 25, 2010, pp. 33-47. May 2010.

Transducer

Transducer x→y



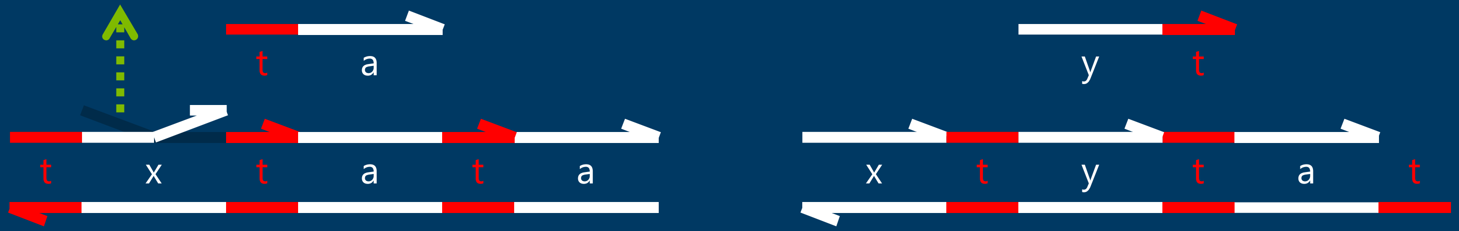
Transducer x→y



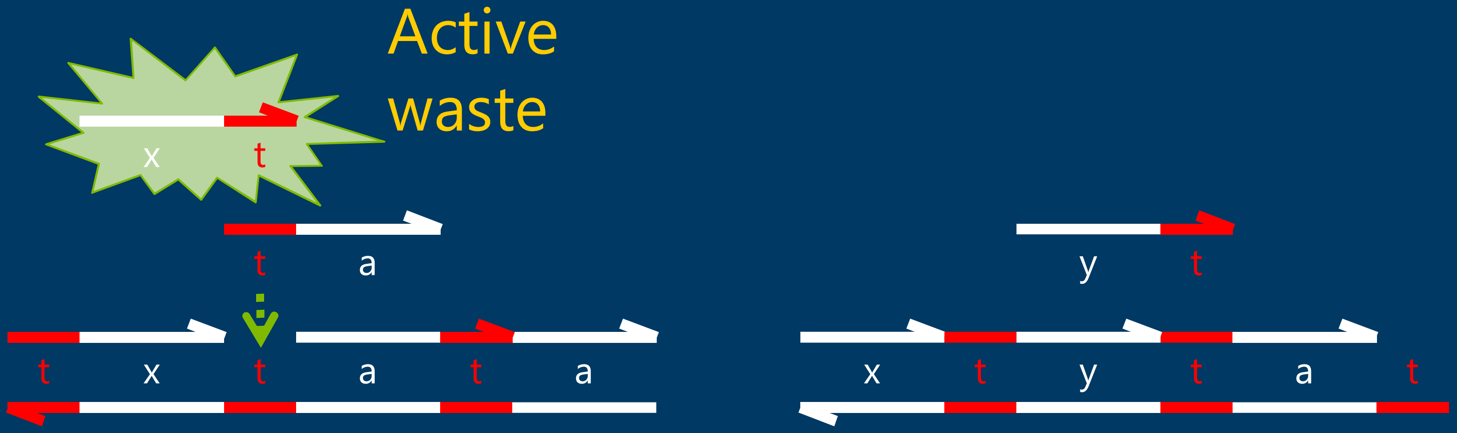
Built by self-assembly!

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

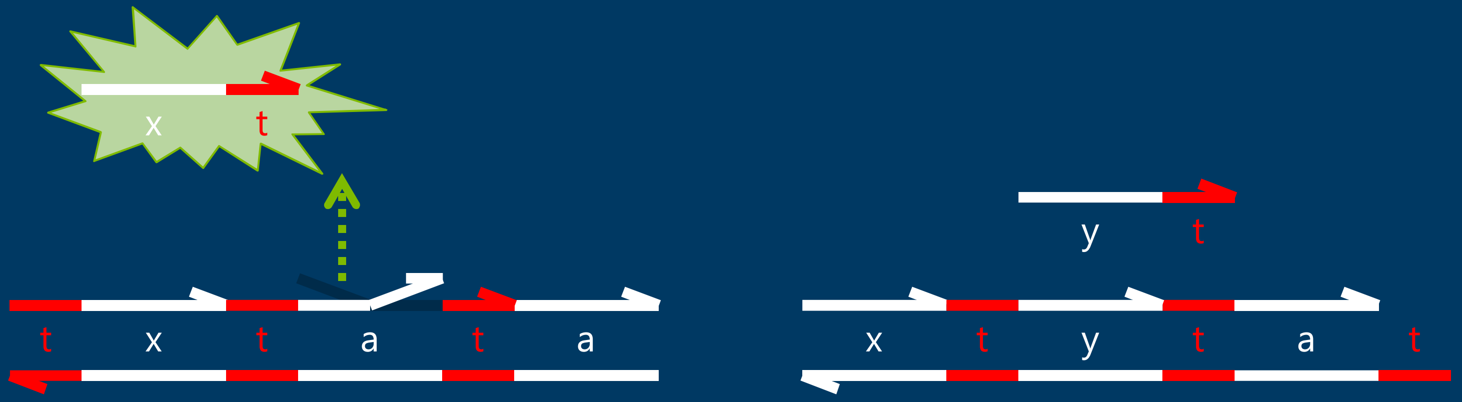
Transducer x→y



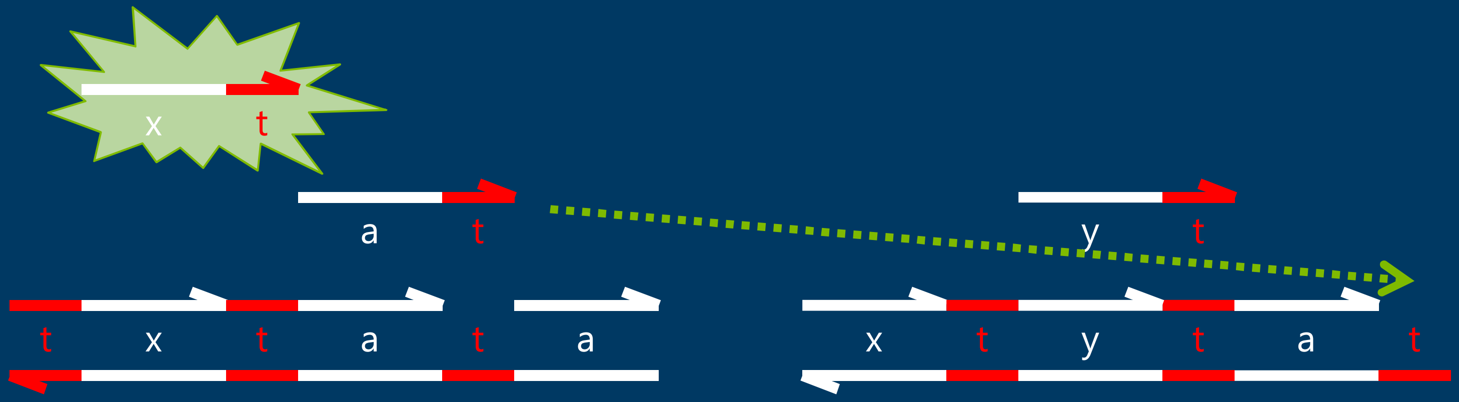
Transducer x⟶y



Transducer x⟶y

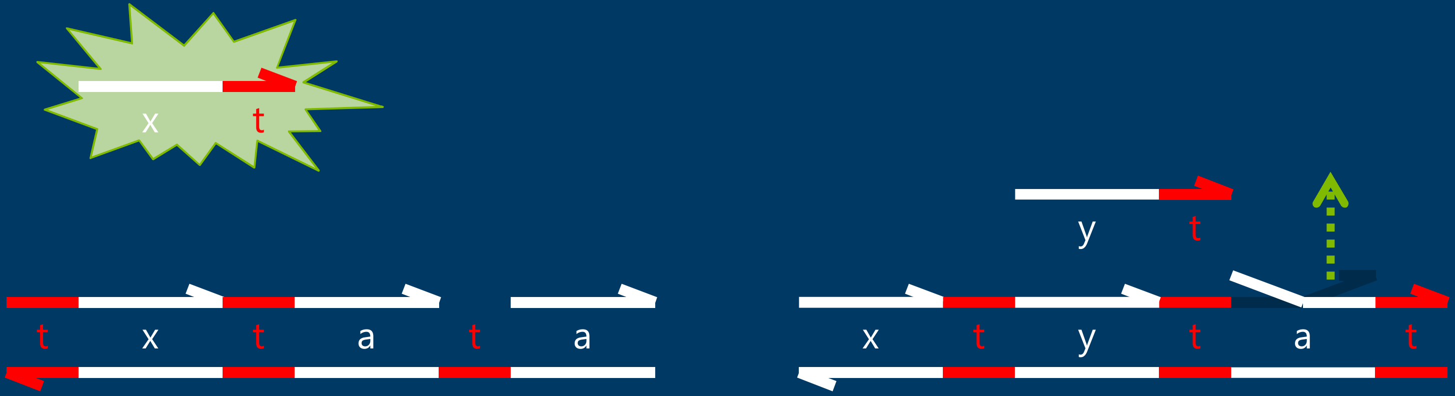


Transducer x→y

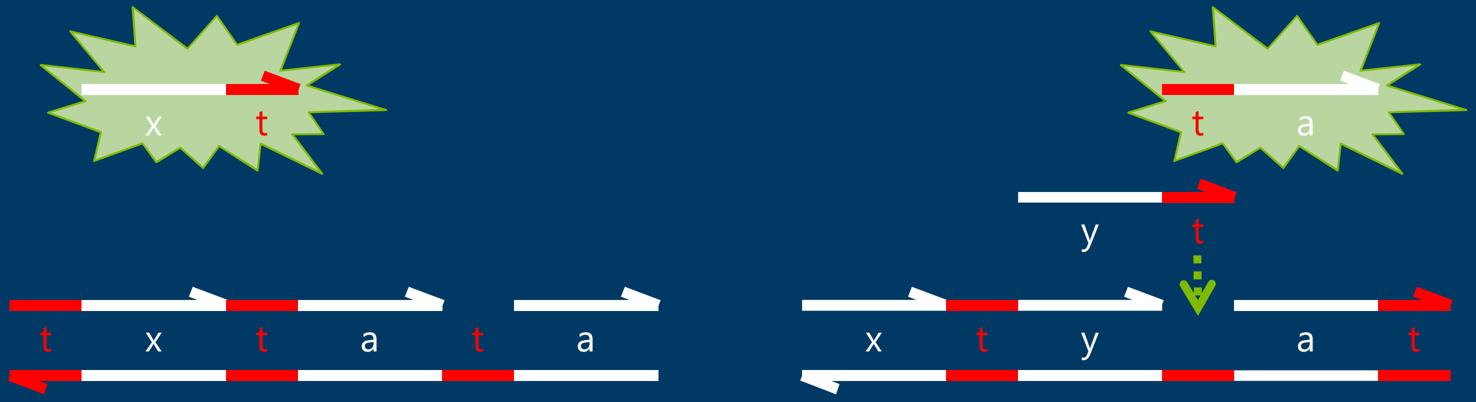


So far, a **tx** *signal* has produced an **at** *cosignal.*
But we want signals as output, not cosignals.

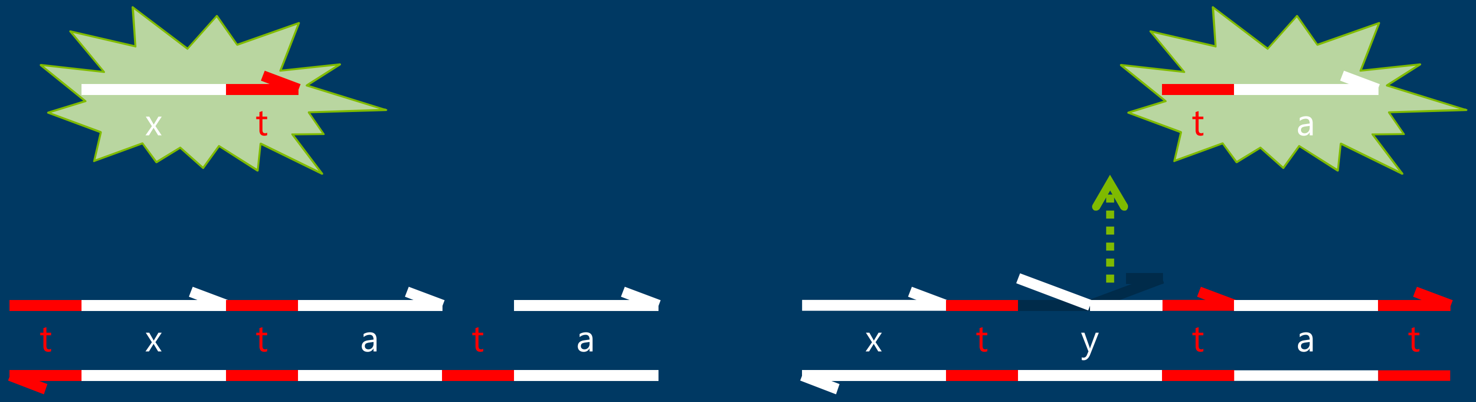
Transducer x→y



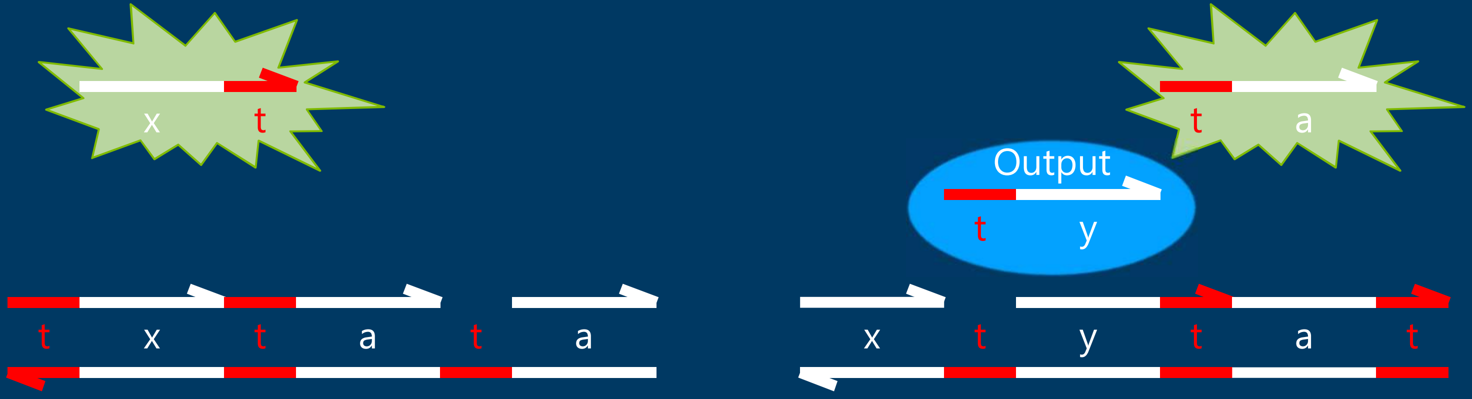
Transducer x→y



Transducer x⟶y



Transducer x→y

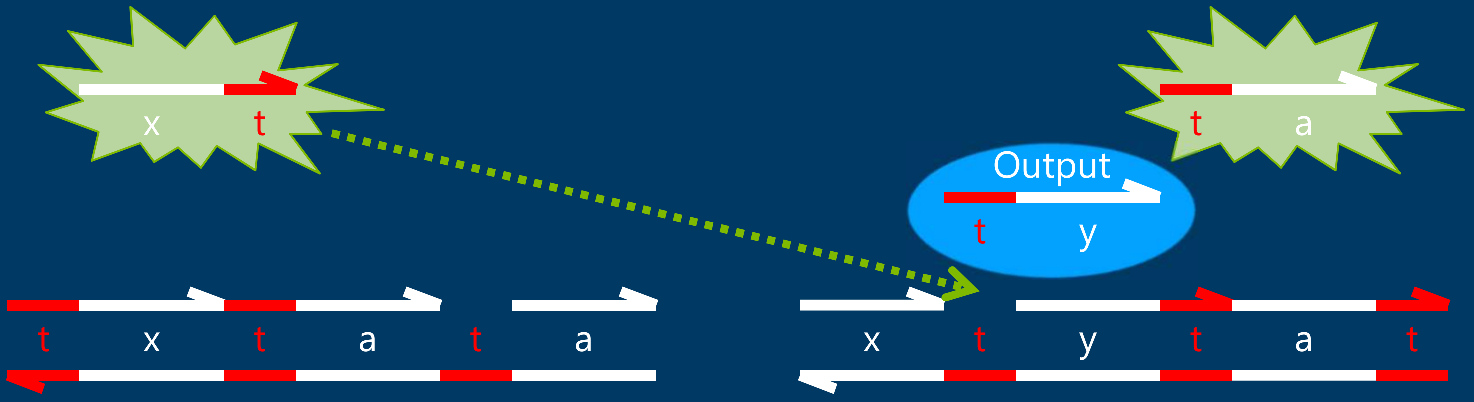


Here is our output **ty** *signal*.
But we are not done yet:

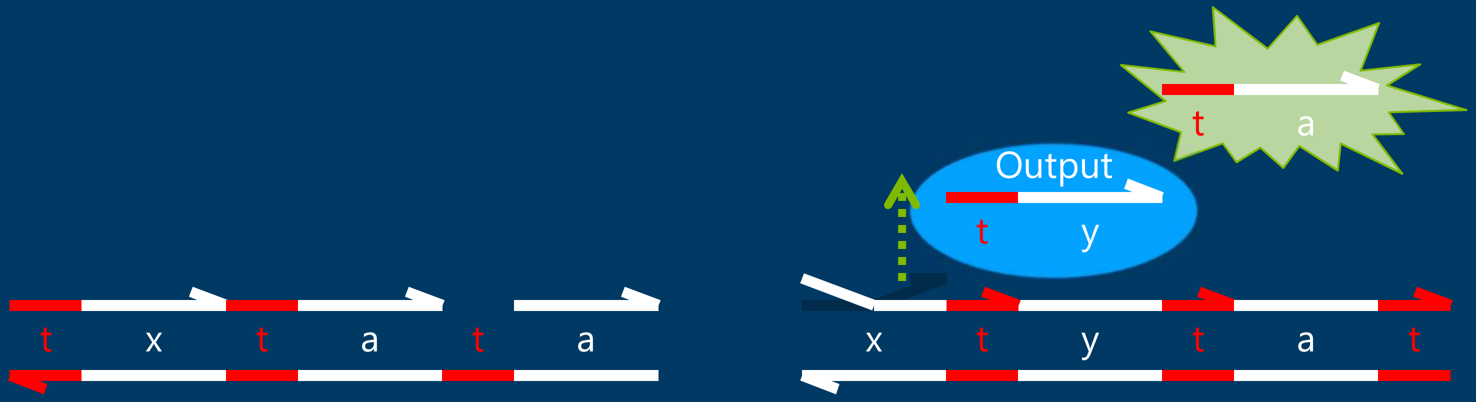
1) We need to make the output irreversible.
2) We need to remove the garbage.

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

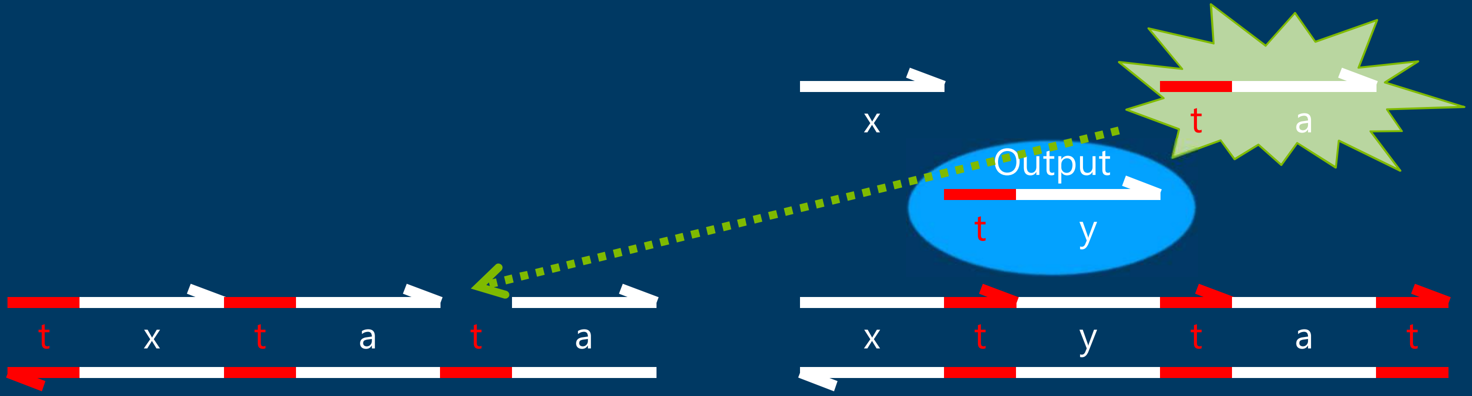
Transducer x→y



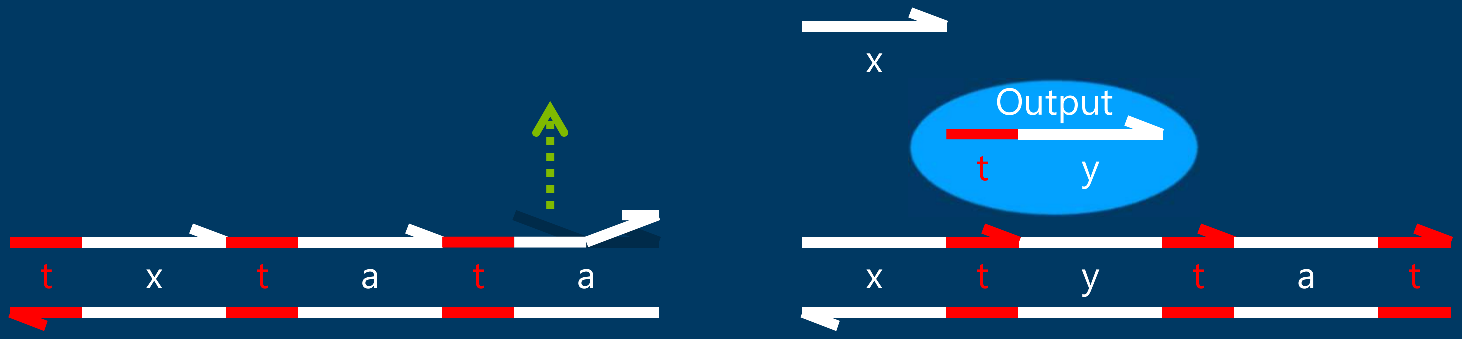
Transducer x→y



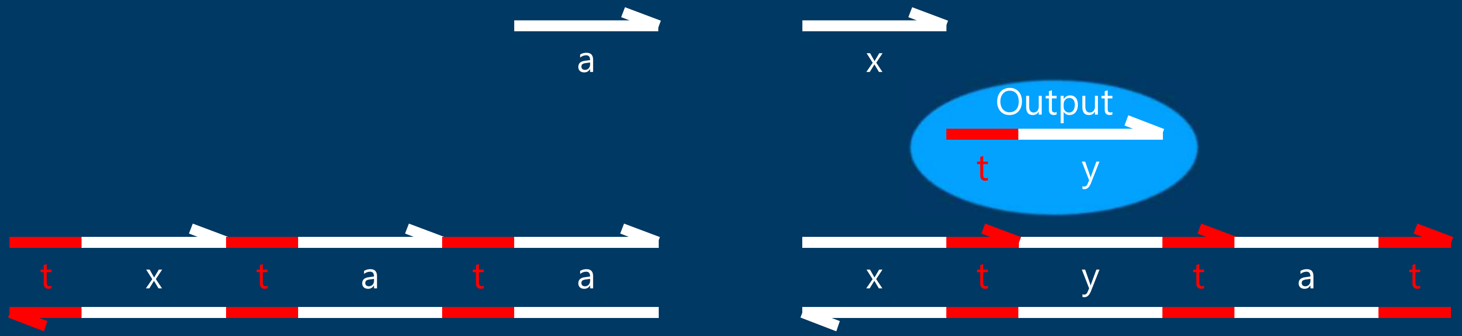
Transducer x→y



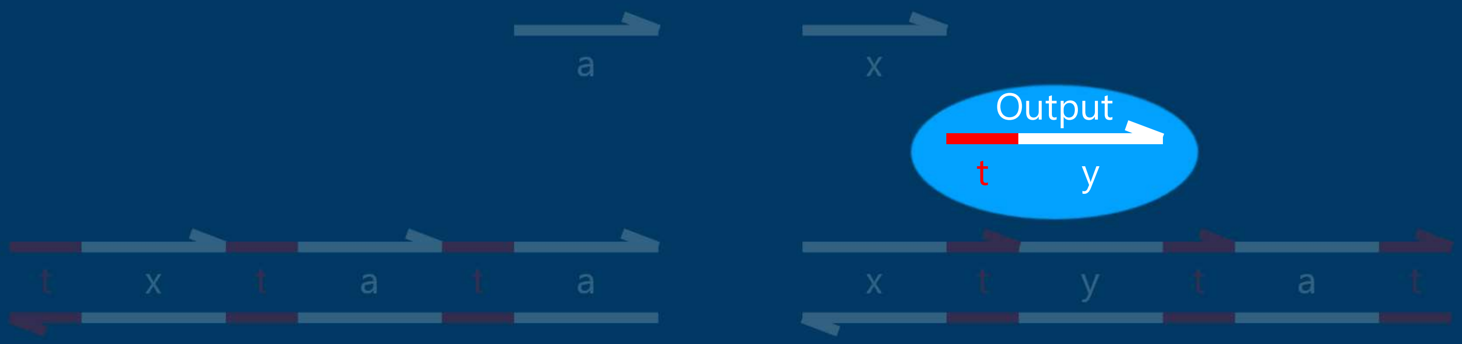
Transducer x⟶y



Transducer x→y



Transducer x→y



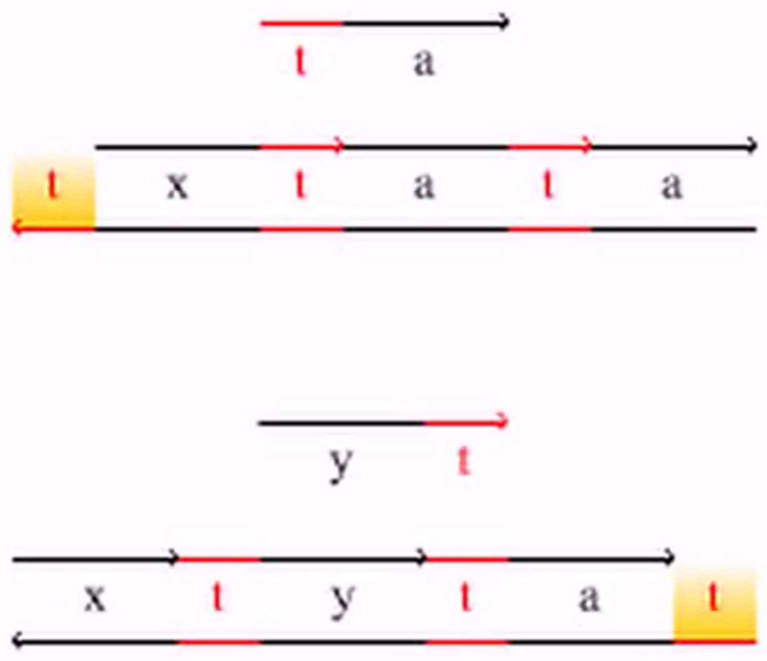
Done.

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

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

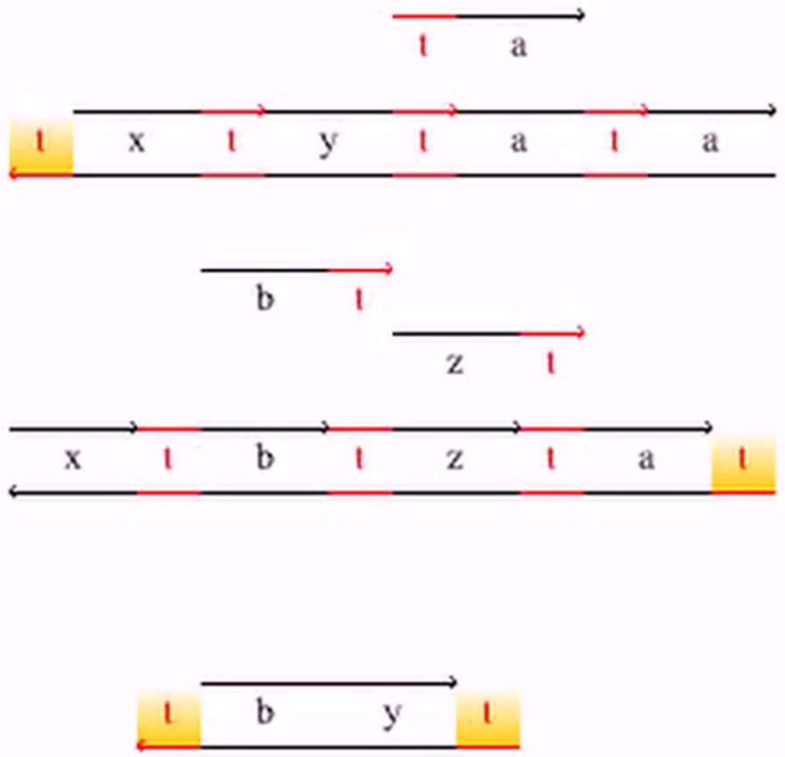
Powered by Sothink

Transducer $x \rightarrow y$



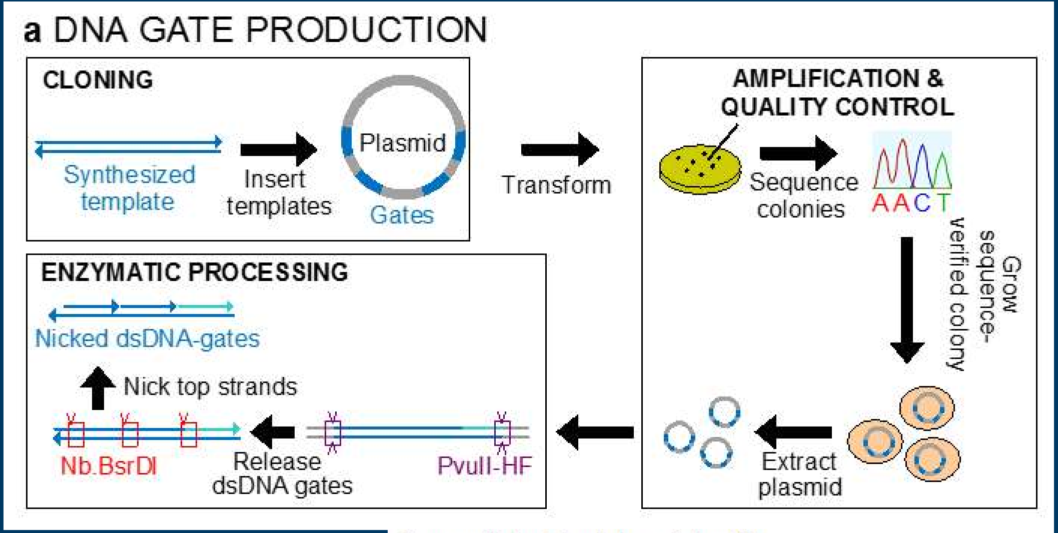
Powered by Sothink

Join $x+y \rightarrow z$



Plasmidic Gate Technology

- Synthetic DNA is length-limited
  - Finite error probability at each nucleotide addition, hence ~ 200nt max
- Bacteria can replicate plasmids for us
  - Loops of DNA 1000's nt, with extremely high fidelity
  - Practically no structural limitations on gate fan-in/fan-out

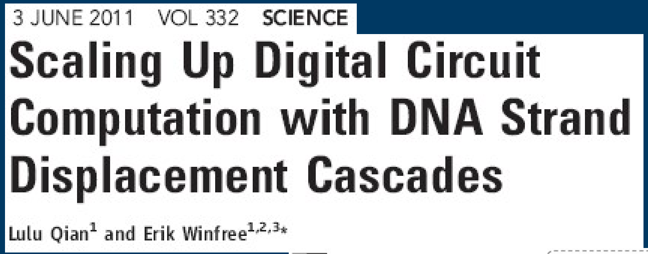


Programmable chemical controllers made from DNA

Yuan-Jyue Chen[1], Neil Dalchau[2], Niranjan Srinivas[3], Andrew Phillips[2], Luca Cardelli[2], David Soloveichik[4], and Georg Seelig[1,5]

Only possible with two-domain architecture

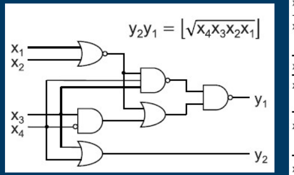
Large-scale Circuits (so far...)

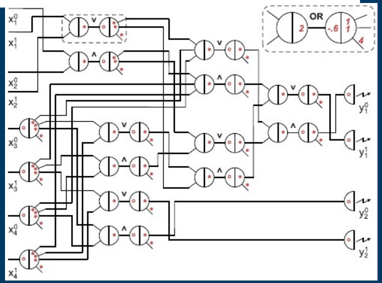


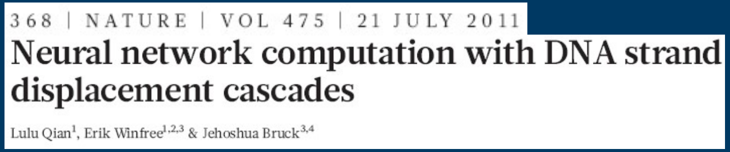
3 JUNE 2011 VOL 332 SCIENCE

Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades

Lulu Qian[1] and Erik Winfree[1,2,3*]



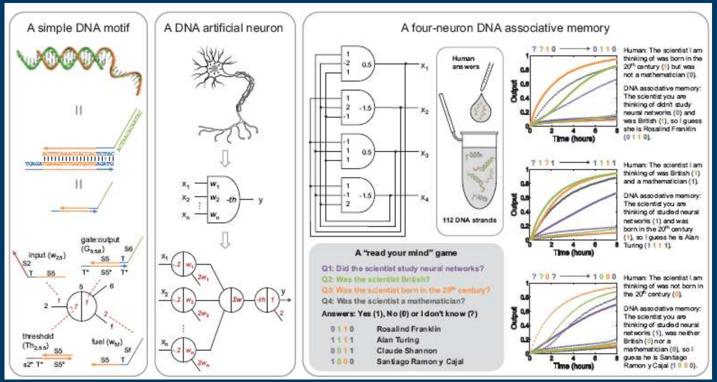




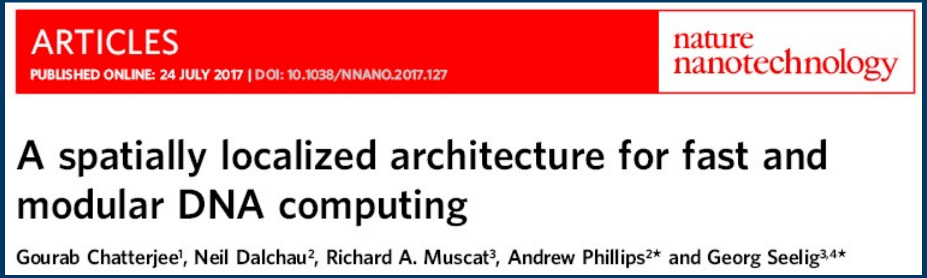
368 | NATURE | VOL 475 | 21 JULY 2011

Neural network computation with DNA strand displacement cascades

Lulu Qian[1], Erik Winfree[1,2,3] & Jehoshua Bruck[3,4]


A simple DNA motif
A DNA artificial neuron
A four-neuron DNA associative memory
Human answers
112 DNA strands
A "read your mind" game
Q1: Did the scientist study neural networks?
Q2: Was the scientist British?
Q3: Was the scientist born in the 20th century?
Q4: Was the scientist a mathematician?
Answers: Yes (1), No (0) or I don't know (?)
0 1 1 0 Rosalind Franklin
1 1 1 1 Alan Turing
0 0 1 1 Claude Shannon
1 0 0 0 Santiago Ramon y Cajal
Output
Time (hours)

Scaling up: DNA Circuit Boards



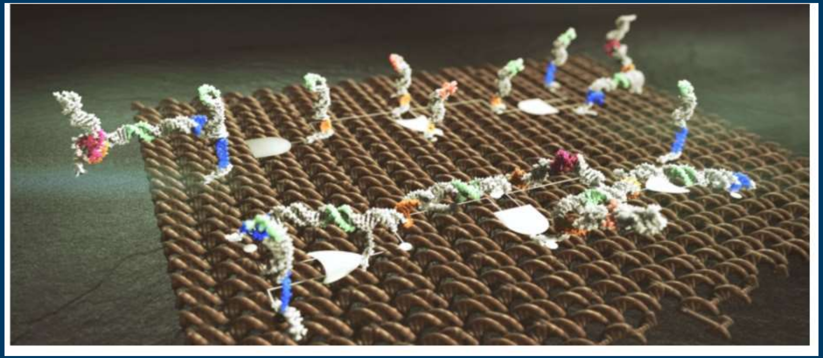
ARTICLES

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

nature nanotechnology

A spatially localized architecture for fast and modular DNA computing

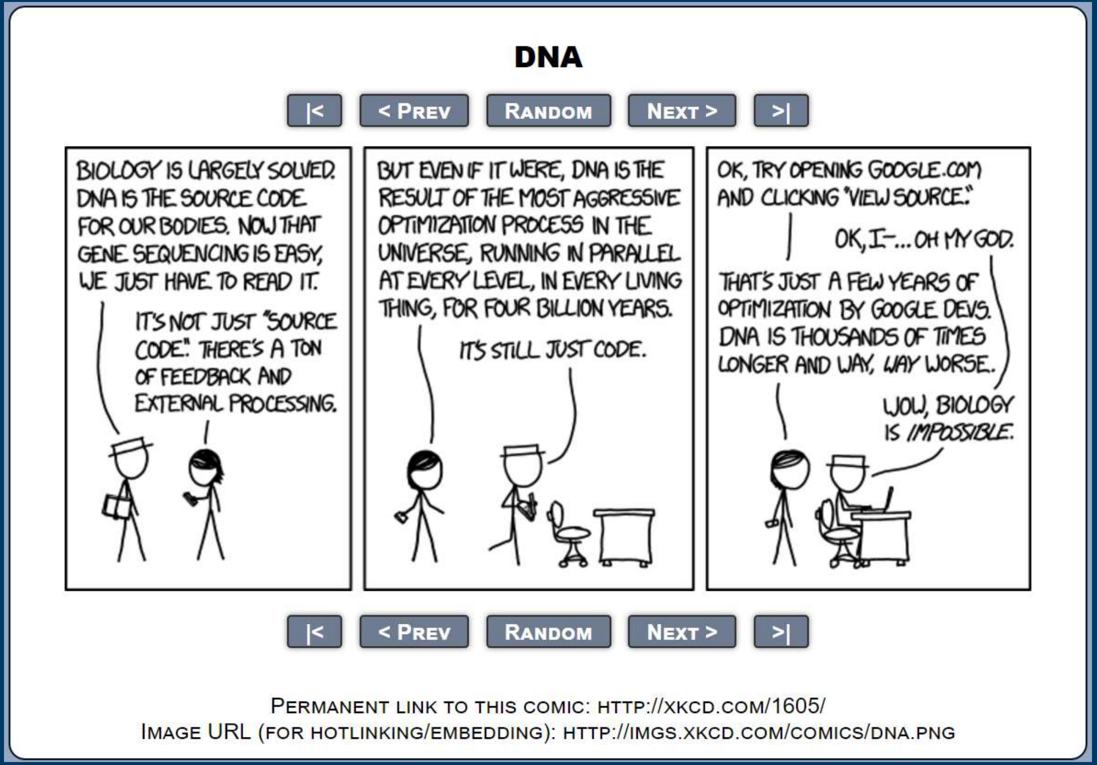
Gourab Chatterjee[1], Neil Dalchau[2], Richard A. Muscat[3], Andrew Phillips[2]* and Georg Seelig[3,4]*



The first computational circuit boards made of DNA
https://www.microsoft.com/en-us/research/blog/researchers-build-nanoscale-computational-circuit-boards-dna

Questions?

BREAK

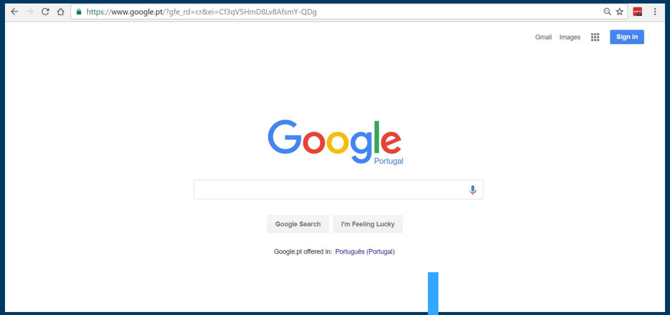

DNA

|< < PREV RANDOM NEXT > >|

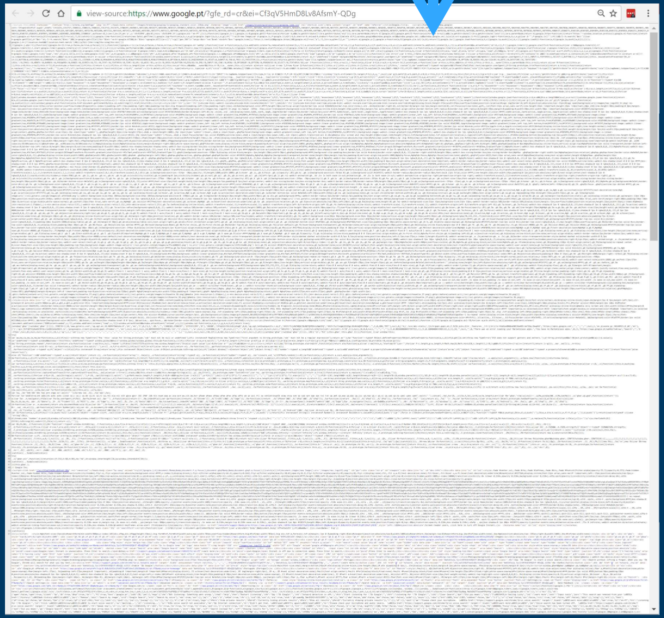
BIOLOGY IS LARGELY SOLVED. DNA IS THE SOURCE CODE FOR OUR BODIES. NOW THAT GENE SEQUENCING IS EASY, WE JUST HAVE TO READ IT.
IT'S NOT JUST "SOURCE CODE." THERE'S A TON OF FEEDBACK AND EXTERNAL PROCESSING.
BUT EVEN IF IT WERE, DNA IS THE RESULT OF THE MOST AGGRESSIVE OPTIMIZATION PROCESS IN THE UNIVERSE, RUNNING IN PARALLEL AT EVERY LEVEL, IN EVERY LIVING THING, FOR FOUR BILLION YEARS.
IT'S STILL JUST CODE.
OK, TRY OPENING GOOGLE.COM AND CLICKING "VIEW SOURCE."
OK, I—...OH MY GOD.
THAT'S JUST A FEW YEARS OF OPTIMIZATION BY GOOGLE DEVS. DNA IS THOUSANDS OF TIMES LONGER AND WAY, WAY WORSE.
WOW, BIOLOGY IS IMPOSSIBLE.

|< < PREV RANDOM NEXT > >|

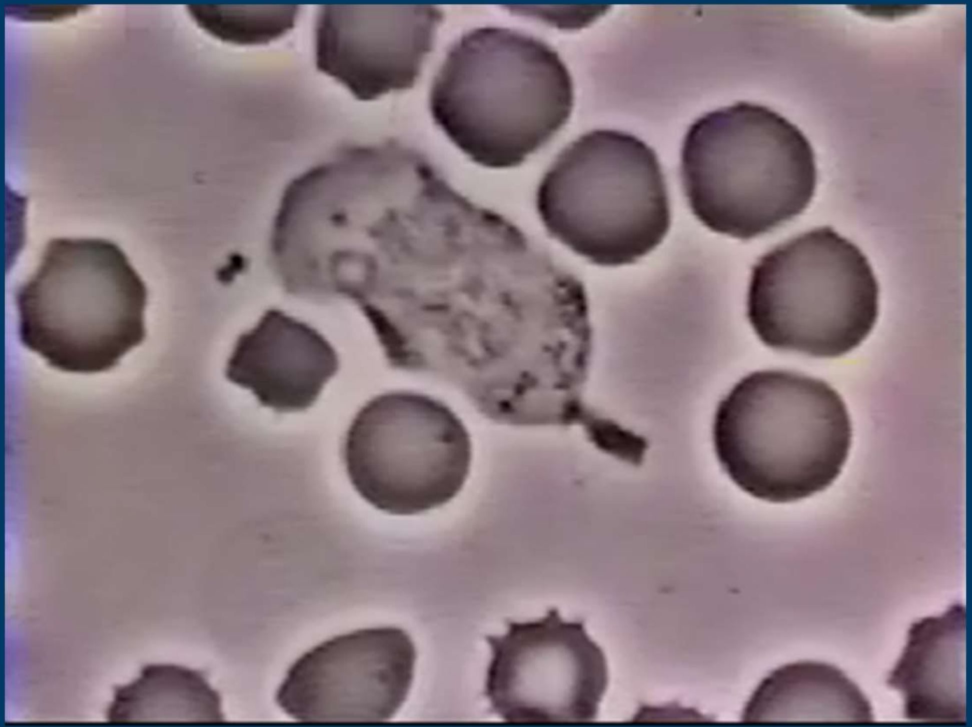
PERMANENT LINK TO THIS COMIC: HTTP://XKCD.COM/1605/
IMAGE URL (FOR HOTLINKING/EMBEDDING): HTTP://IMGS.XKCD.COM/COMICS/DNA.PNG


https://www.google.pt/?gfe_rd=cr&ei=Cf3qV5HmD8Lv8AfsmY-QDg
Gmail
Images
Sign in
Google
Portugal
Google Search
I'm Feeling Lucky
Google.pt offered in: Português (Portugal)

View Source


view-source:https://www.google.pt/?gfe_rd=cr&ei=Cf3qV5HmD8Lv8AfsmY-QDg

Some kind of computation



A Molecular Algorithm

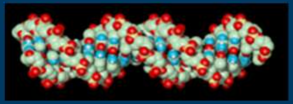
Running something interesting with DNA

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]

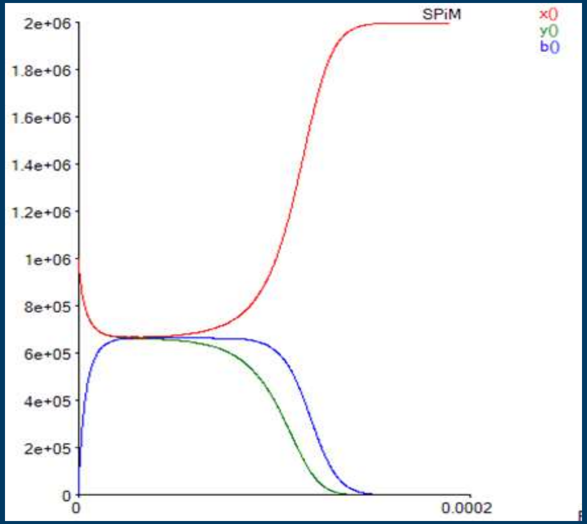
- 3 rules of agent (or molecule) interaction
  - $X + Y \rightarrow B + B$
  - $B + X \rightarrow X + X$
  - $B + Y \rightarrow Y + Y$

"our program" ⟶



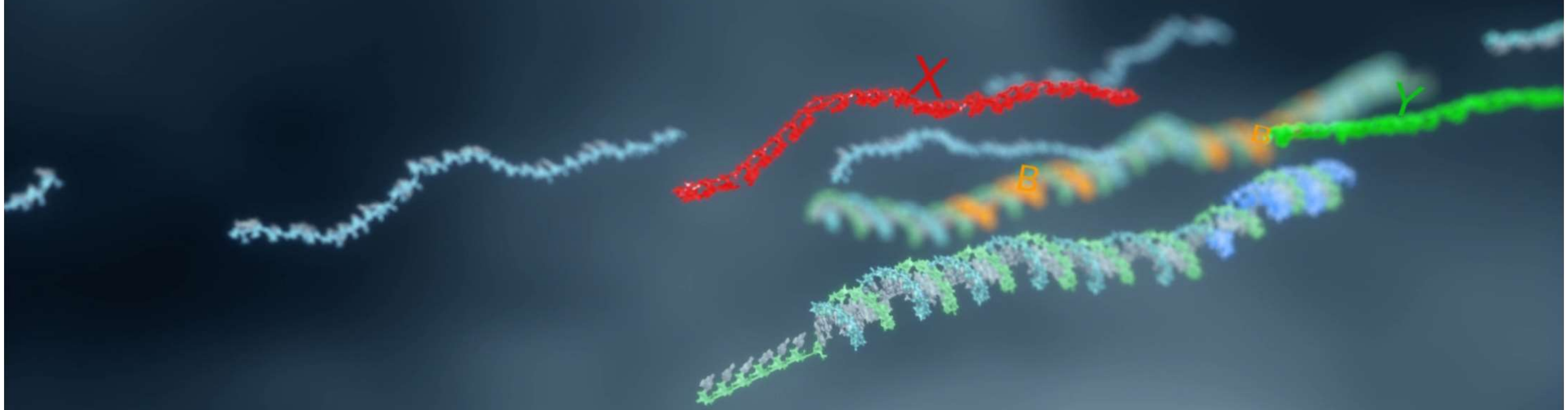
Surprisingly good (in fact, optimal)

- 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 ω(√n log n)
  - Hence the agreement state is stable

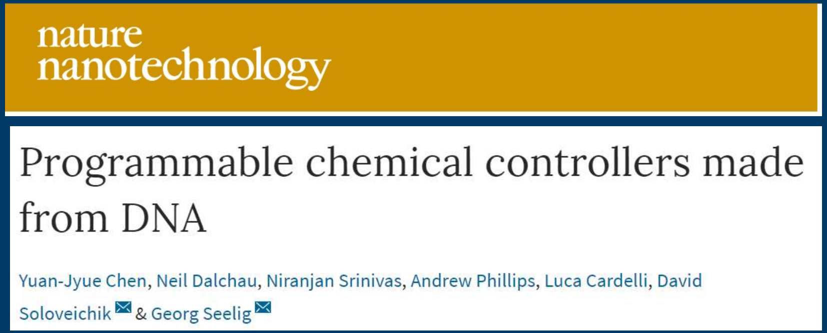


Stochastic simulation of worst-case scenario with initially #X = #Y

Circuit component $X + Y \rightarrow 2B$

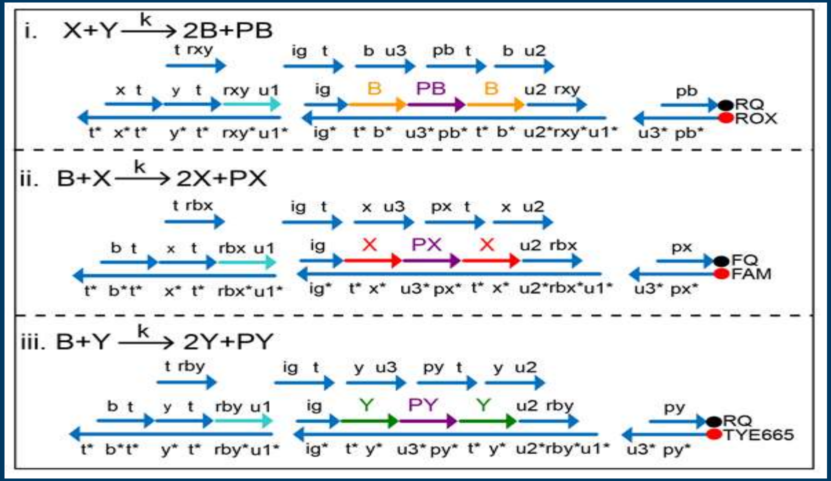


DNA Implementation of AM

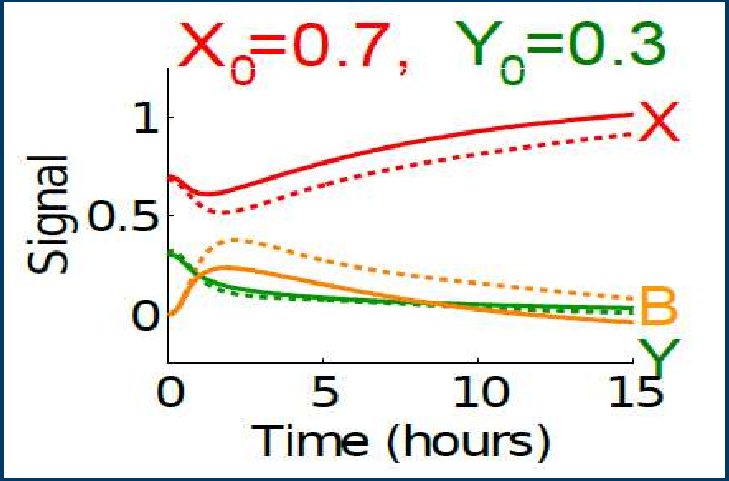

nature nanotechnology

Programmable chemical controllers made from DNA

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



i. $X+Y \xrightarrow{k} 2B+PB$
t rxy
ig t
b u3
pb t
b u2
x t
y t
rxy u1
ig
B
PB
B
u2 rxy
pb
RQ
ROX
t* x* t* y* t* rxy* u1* ig* t* b* u3* pb* t* b* u2* rxy* u1* u3* pb*
ii. $B+X \xrightarrow{k} 2X+PX$
t rbx
ig t
x u3
px t
x u2
b t
x t
rbx u1
ig
X
PX
X
u2 rbx
px
FQ
FAM
t* b* t* x* t* rbx* u1* ig* t* x* u3* px* t* x* u2* rbx* u1* u3* px*
iii. $B+Y \xrightarrow{k} 2Y+PY$
t rby
ig t
y u3
py t
y u2
b t
y t
rby u1
ig
Y
PY
Y
u2 rby
py
RQ
TYE665
t* b* t* y* t* rby* u1* ig* t* y* u3* py* t* y* u2* rby* u1* u3* py*



Carbon-based Computing

How to get there

Action Plan

- 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]

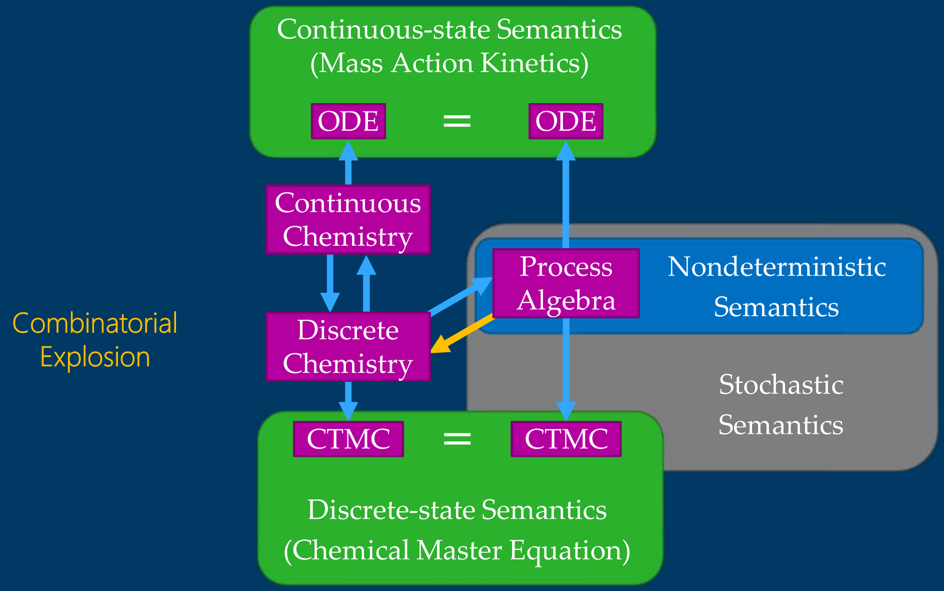
- To realize the potential of Molecular Programming

- "With *no alien technology*" [David Soloveichik]

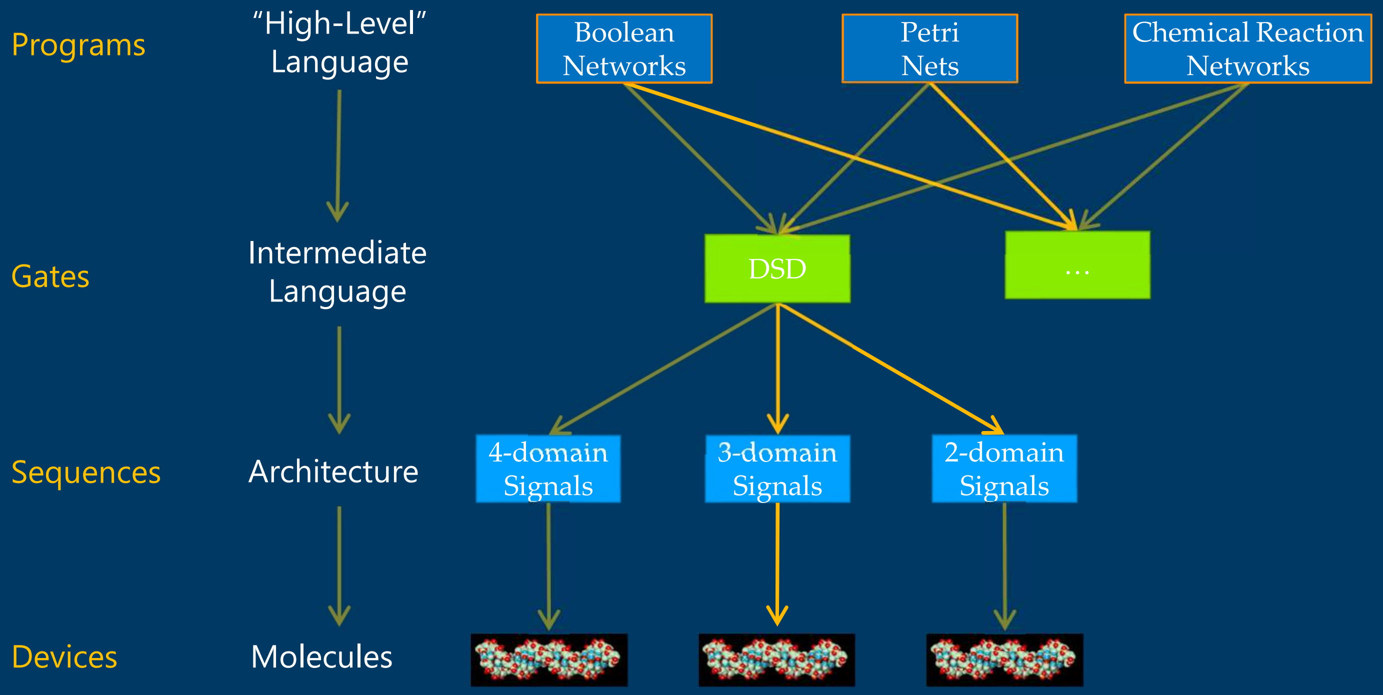
- This is largely a 'software problem' even when working on device design

Chemistry as a Concurrent Language

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

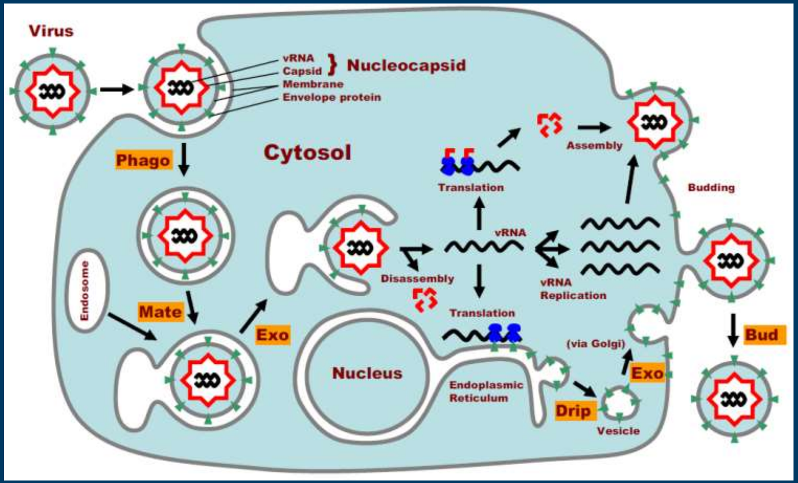


Molecular Compilation

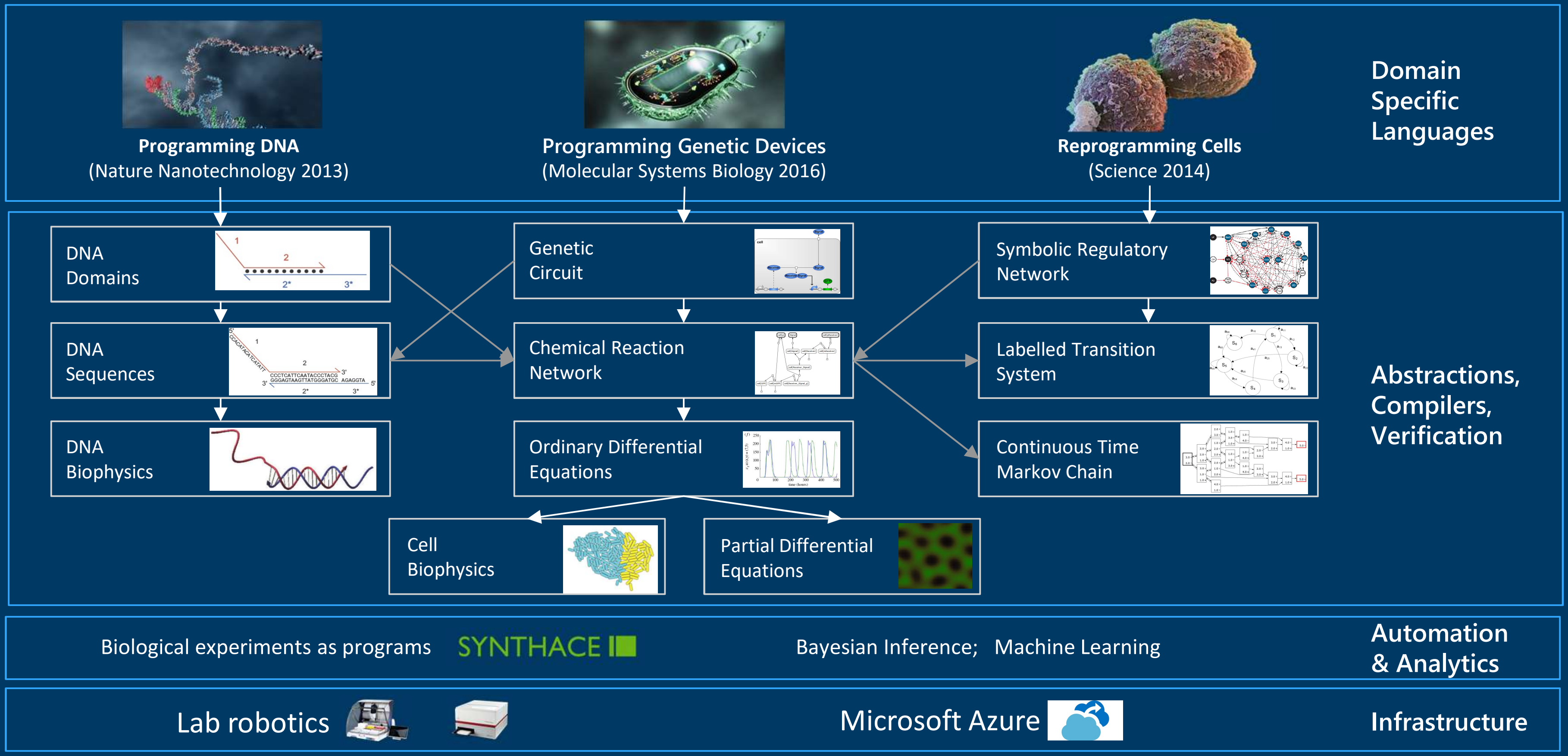


Towards High(er)-Level Languages

- Gene Networks
  - Synchronous Boolean networks
    - Stewart Kauffman, etc.
  - Asynchronous Boolean networks
    - René Thomas, etc.
- Protein Networks
  - Process Algebra (stochastic π-calculus etc.)
    - Priami, Regev-Shapiro, etc.
  - Graph Rewriting (kappa, BioNetGen etc.)
    - Danos-Laneve, Fontana & al., etc.
- Membrane Networks
  - Membrane Computing
    - Gheorghe Păun, etc.
  - Brane Calculi
    - Luca Cardelli, etc.
- Waiting for an architecture to run on…



A platform for programming biology

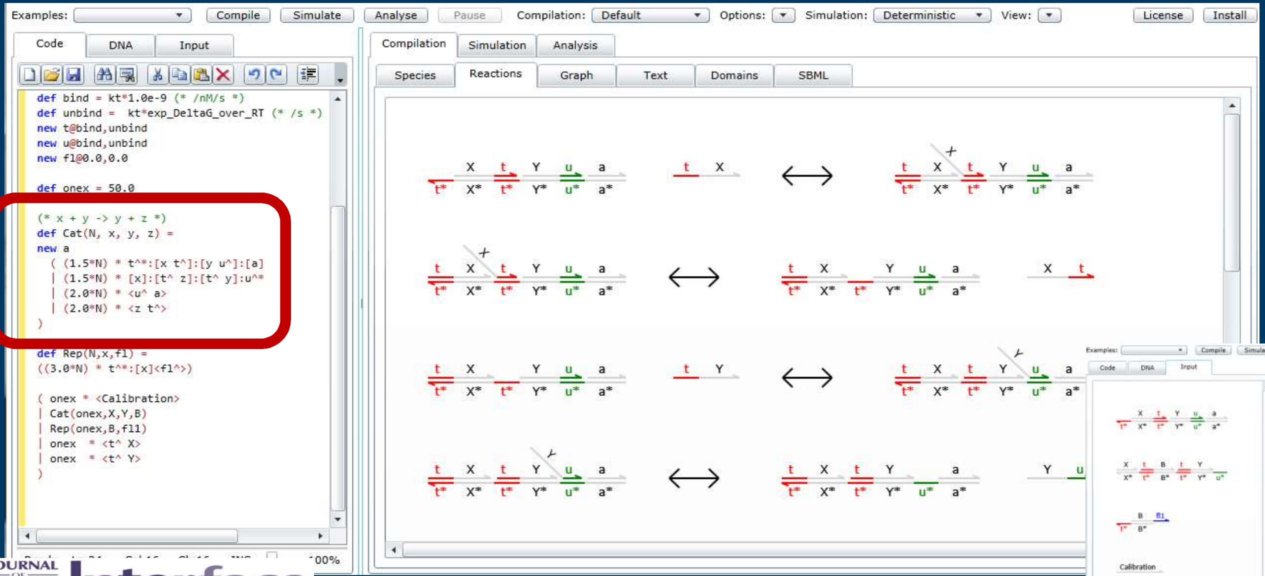

Programming DNA
(Nature Nanotechnology 2013)
Programming Genetic Devices
(Molecular Systems Biology 2016)
Reprogramming Cells
(Science 2014)
Domain Specific Languages
DNA Domains
DNA Sequences
DNA Biophysics
Genetic Circuit
Chemical Reaction Network
Ordinary Differential Equations
Symbolic Regulatory Network
Labelled Transition System
Continuous Time Markov Chain
Cell Biophysics
Partial Differential Equations
Abstractions, Compilers, Verification
Biological experiments as programs
SYNTHACE
Bayesian Inference; Machine Learning
Automation & Analytics
Lab robotics
Microsoft Azure
Infrastructure

Algorithm Design

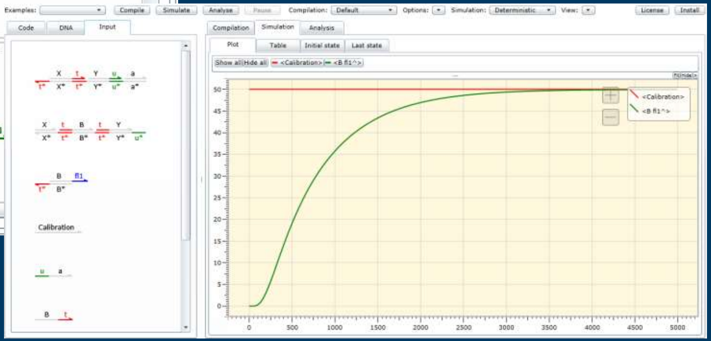
A software pipeline for Molecular Programming

Development Tools

MSRC Biological Computation Group



```
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 fl@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 t^]:[y u^]:[a]
  | (1.5*N) * [x]:[t^ z]:[t^ y]:u^*
  | (2.0*N) * <u^ a>
  | (2.0*N) * <z t^>
)

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

( onex * <Calibration>
| Cat(onex,X,Y,B)
| Rep(onex,B,fl1)
| onex  * <t^ X>
| onex  * <t^ Y>
)
```



Visual DSD

A Development Environment for DNA Strand Displacement

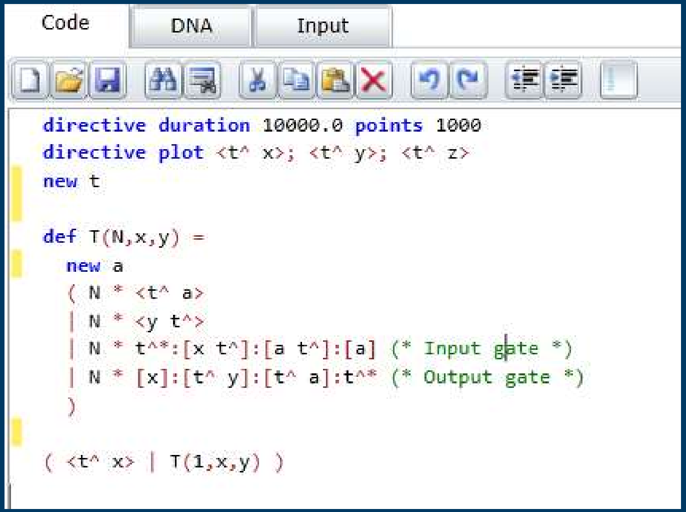
JOURNAL OF THE ROYAL SOCIETY Interface

A programming language for composable DNA circuits

Andrew Phillips and Luca Cardelli

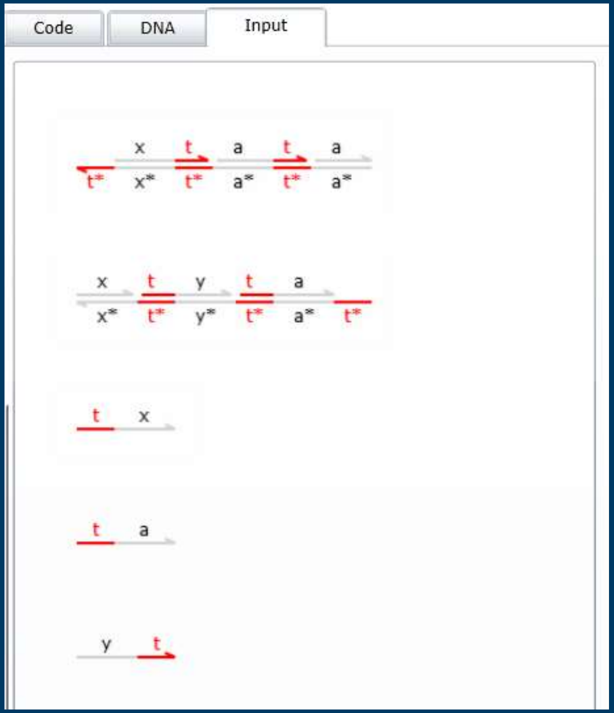
A Language for DNA Structures

- Describe the initial *structures* (not behavior)



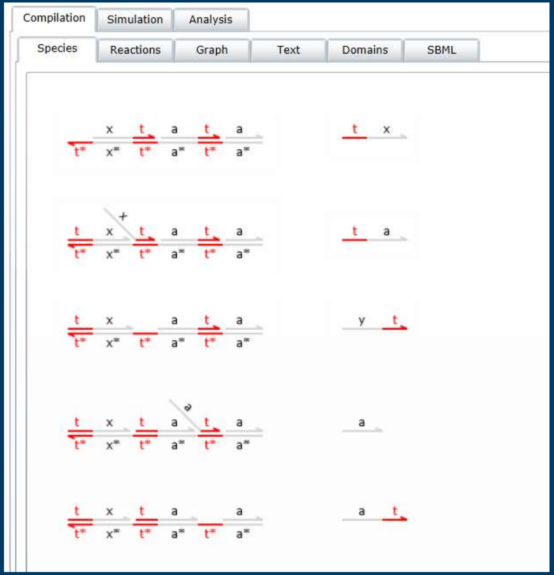
```
directive duration 10000.0 points 1000
directive plot <t^ x>; <t^ y>; <t^ z>
new t

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) )
```

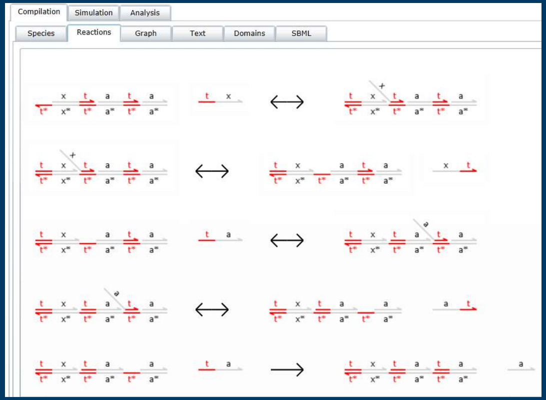
=

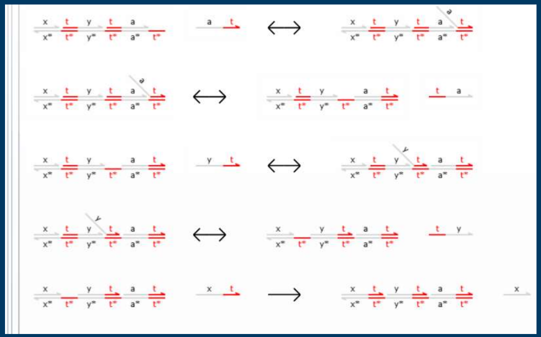


Compute Species and Reactions

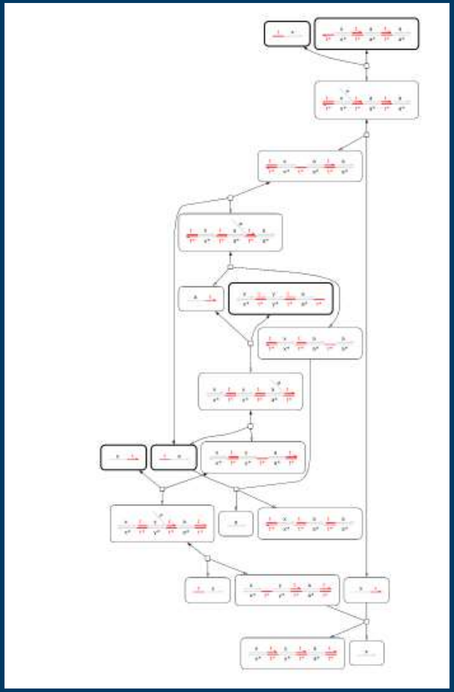
- Recursively computed from the initial structures

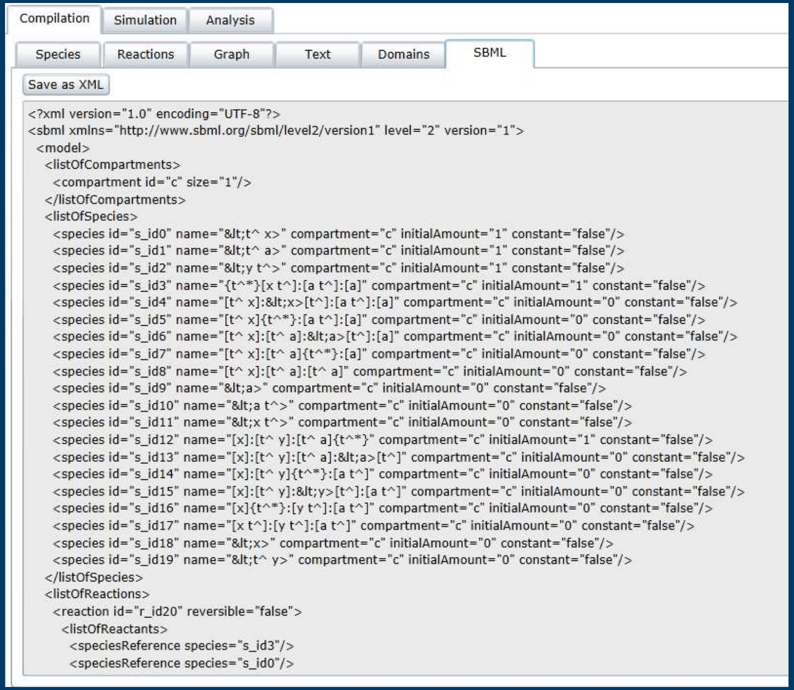






Reaction Graph and Export

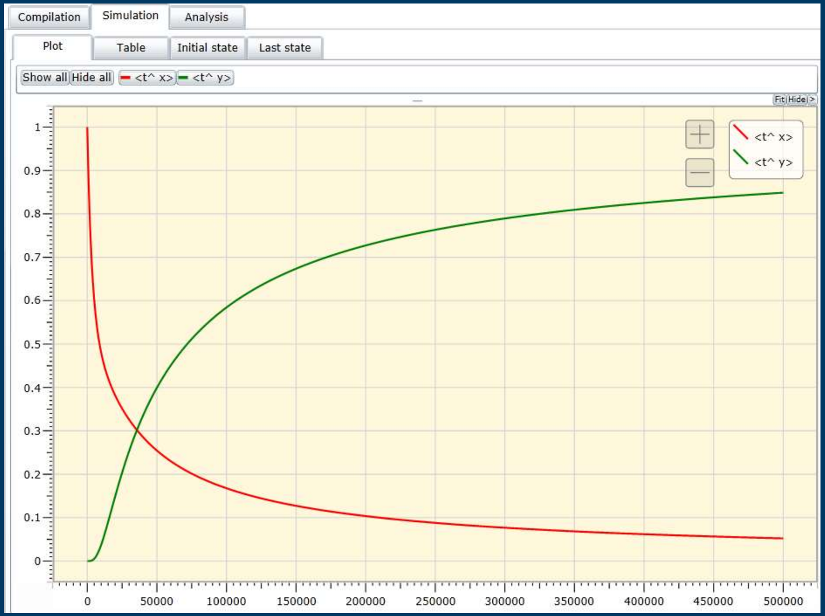




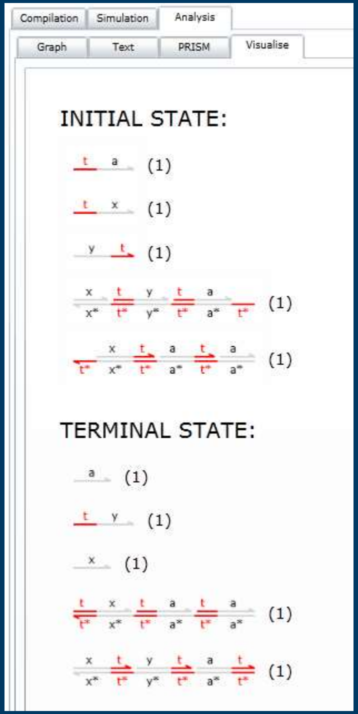
```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level2/version1" level="2" version="1">
  <model>
    <listOfCompartments>
      <compartment id="c" size="1"/>
    </listOfCompartments>
    <listOfSpecies>
      <species id="s_id0" name="&lt;t^ x>" compartment="c" initialAmount="1" constant="false"/>
      <species id="s_id1" name="&lt;t^ a>" compartment="c" initialAmount="1" constant="false"/>
      <species id="s_id2" name="&lt;y t^>" compartment="c" initialAmount="1" constant="false"/>
      <species id="s_id3" name="{t^*}[x t^]:[a t^]:[a]" compartment="c" initialAmount="1" constant="false"/>
      <species id="s_id4" name="[t^ x]:&lt;x>[t^]:[a t^]:[a]" 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]:&lt;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="&lt;a>" compartment="c" initialAmount="0" constant="false"/>
      <species id="s_id10" name="&lt;a t^>" compartment="c" initialAmount="0" constant="false"/>
      <species id="s_id11" name="&lt;x t^>" compartment="c" initialAmount="0" constant="false"/>
      <species id="s_id12" name="[x]:[t^ y]:[t^ a]{t^*}" compartment="c" initialAmount="1" constant="false"/>
      <species id="s_id13" name="[x]:[t^ y]:[t^ a]:&lt;a>[t^]" compartment="c" initialAmount="0" constant="false"/>
      <species id="s_id14" name="[x]:[t^ y]{t^*}:[a t^]" compartment="c" initialAmount="0" constant="false"/>
      <species id="s_id15" name="[x]:[t^ y]:&lt;y>[t^]:[a t^]" compartment="c" initialAmount="0" constant="false"/>
      <species id="s_id16" name="[x]{t^*}:[y t^]:[a t^]" compartment="c" initialAmount="0" constant="false"/>
      <species id="s_id17" name="[x t^]:[y t^]:[a t^]" compartment="c" initialAmount="0" constant="false"/>
      <species id="s_id18" name="&lt;x>" compartment="c" initialAmount="0" constant="false"/>
      <species id="s_id19" name="&lt;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

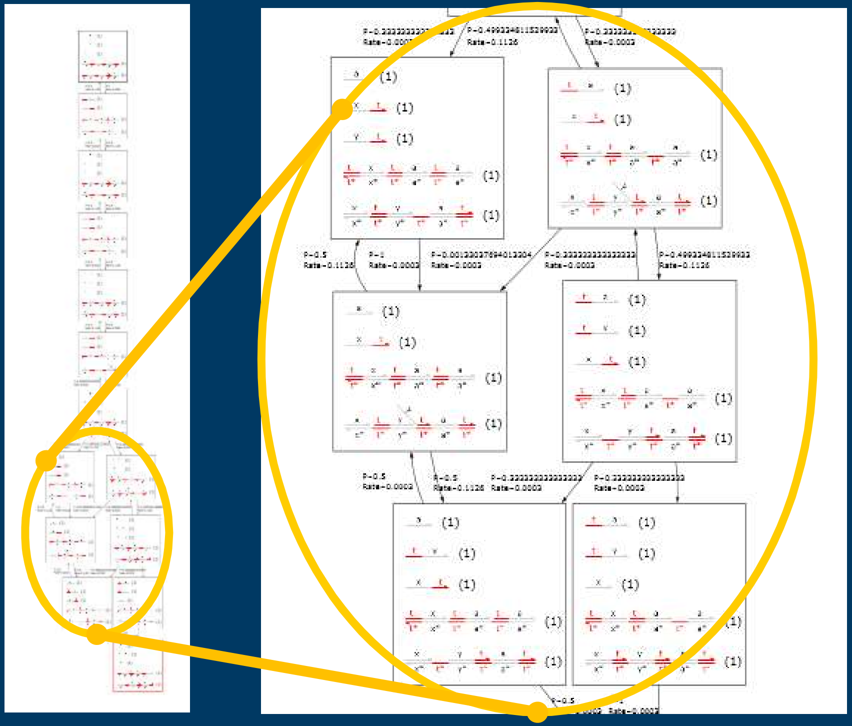
- Deterministic
- Stochastic (Gillespie)
- Probabilistic (CME)
- Linear Noise Approximation
- "JIT"



State Space Analysis

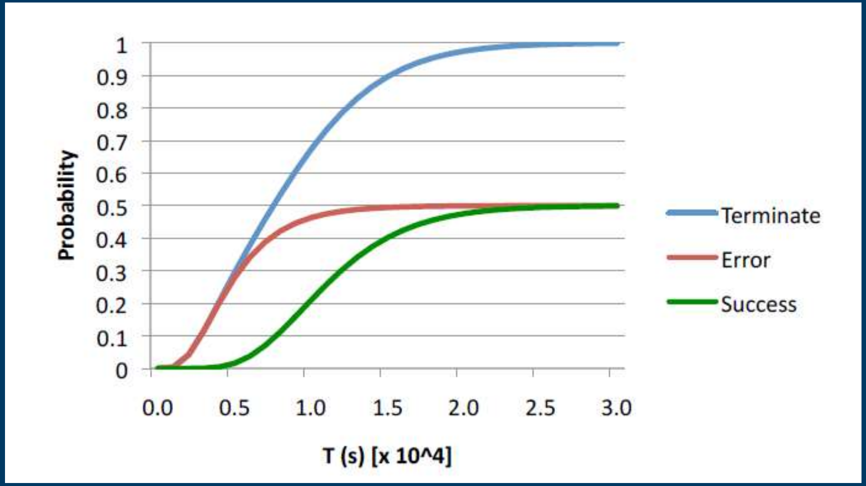


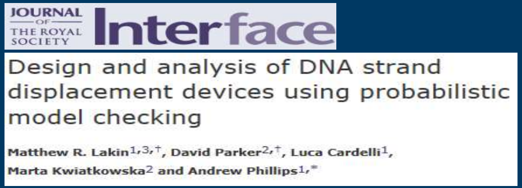
CTMC



Modelchecking

- Export to PRISM probabilistic modelchecker



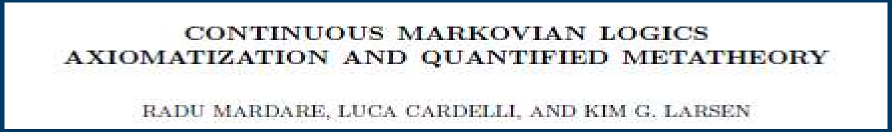

JOURNAL OF THE ROYAL SOCIETY Interface

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

Matthew R. Lakin[1,3,†], David Parker[2,†], Luca Cardelli[1], Marta Kwiatkowska[2] and Andrew Phillips[1,*]

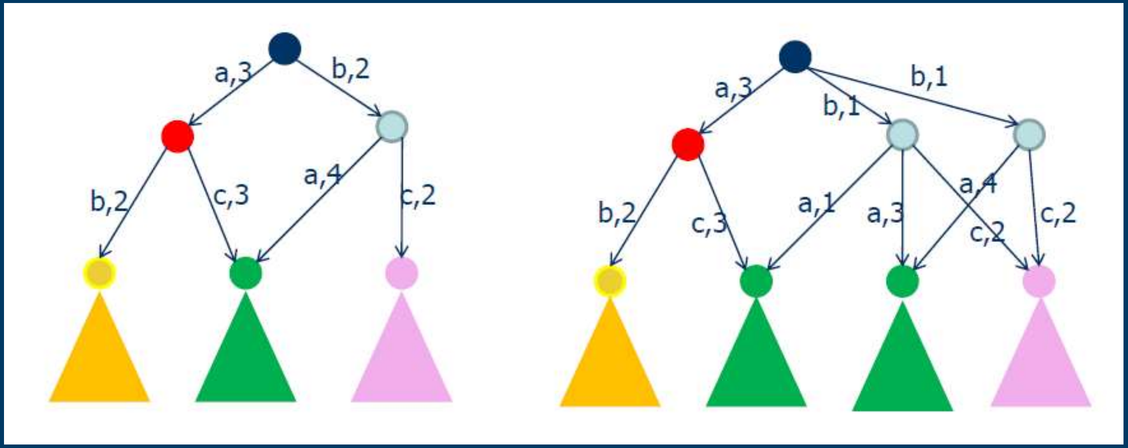
Verification

- Quantitative theories of system equivalence and approximation.

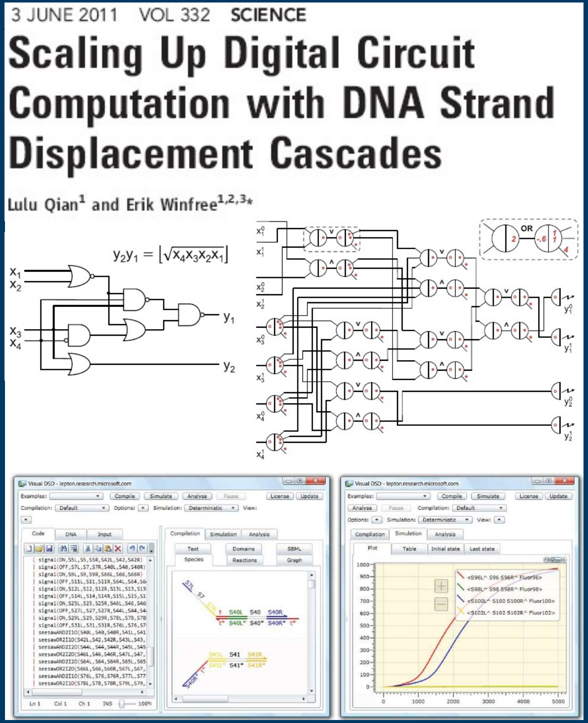


CONTINUOUS MARKOVIAN LOGICS
AXIOMATIZATION AND QUANTIFIED METATHEORY

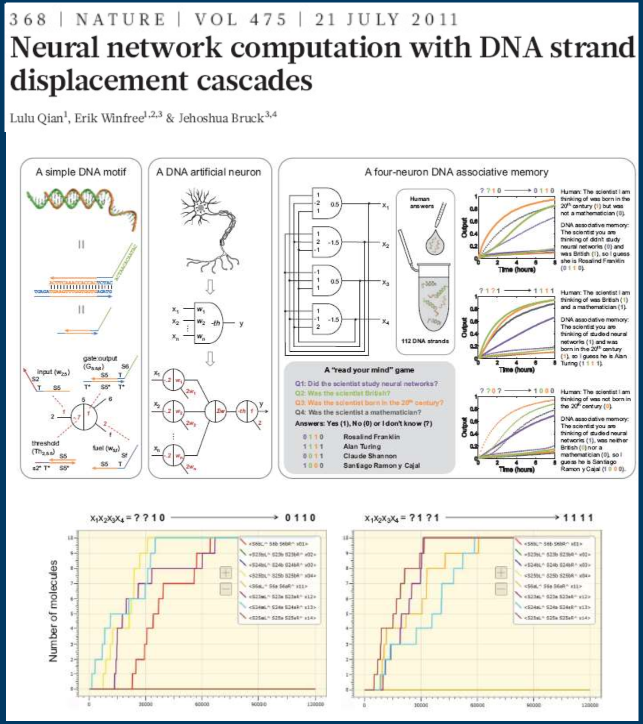
RADU MARDARE, LUCA CARDELLI, AND KIM G. LARSEN



Related Work Supporter by our Tools


3 JUNE 2011 VOL 332 SCIENCE
Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades
Lulu Qian1 and Erik Winfree1,2,3*
$y_2y_1 = \lfloor\sqrt{x_4x_3x_2x_1}\rfloor$
x1
x2
x3
x4
y1
y2
OR

Square root of a 4-bit number


368 | NATURE | VOL 475 | 21 JULY 2011
Neural network computation with DNA strand displacement cascades
Lulu Qian1, Erik Winfree1,2,3 & Jehoshua Bruck3,4
A simple DNA motif
A DNA artificial neuron
A four-neuron DNA associative memory
A "read your mind" game
112 DNA strands
Time (hours)
Number of molecules
x1x2x3x4 = ? ? 1 0 → 0 1 1 0
x1x2x3x4 = ? 1 ? 1 → 1 1 1 1

Associative memory

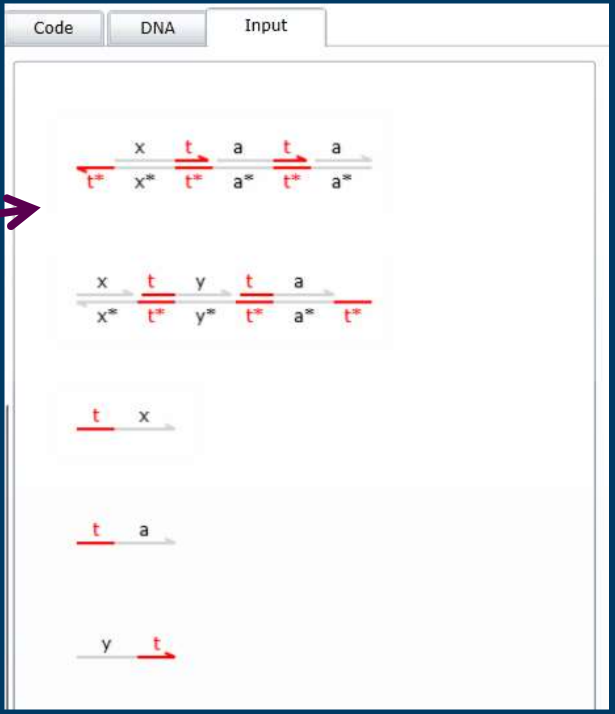
Algorithm Execution

A wetlab pipeline for Molecular Programming

Output of Design Process

- Domain structures
  - (DNA sequences to be determined)

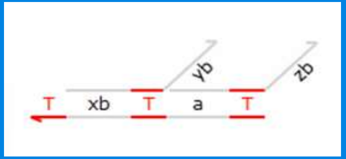
"Ok, how do I run this for real"

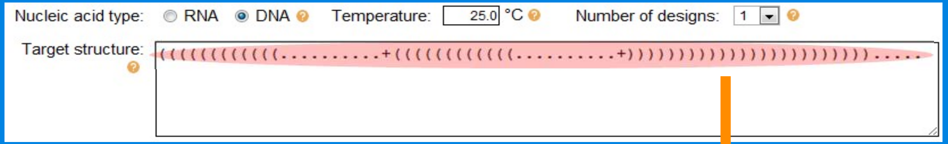


From Structures to Sequences

www.nupack.org

DSD Structure → "Dot-Paren" representation

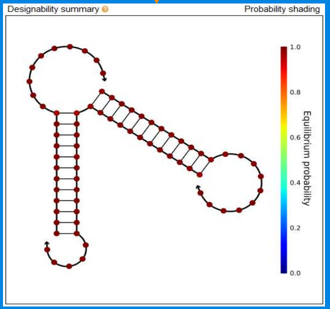




Nucleic acid type: RNA [✓ DNA] Temperature: 25.0 °C Number of designs: 1

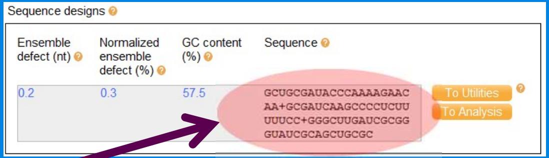
Target structure:

```
((((((((((((..........+((((((((((((..........+))))))))))))))))))))))))).....
```



Thermodynamic Synthesis

Output Sequences

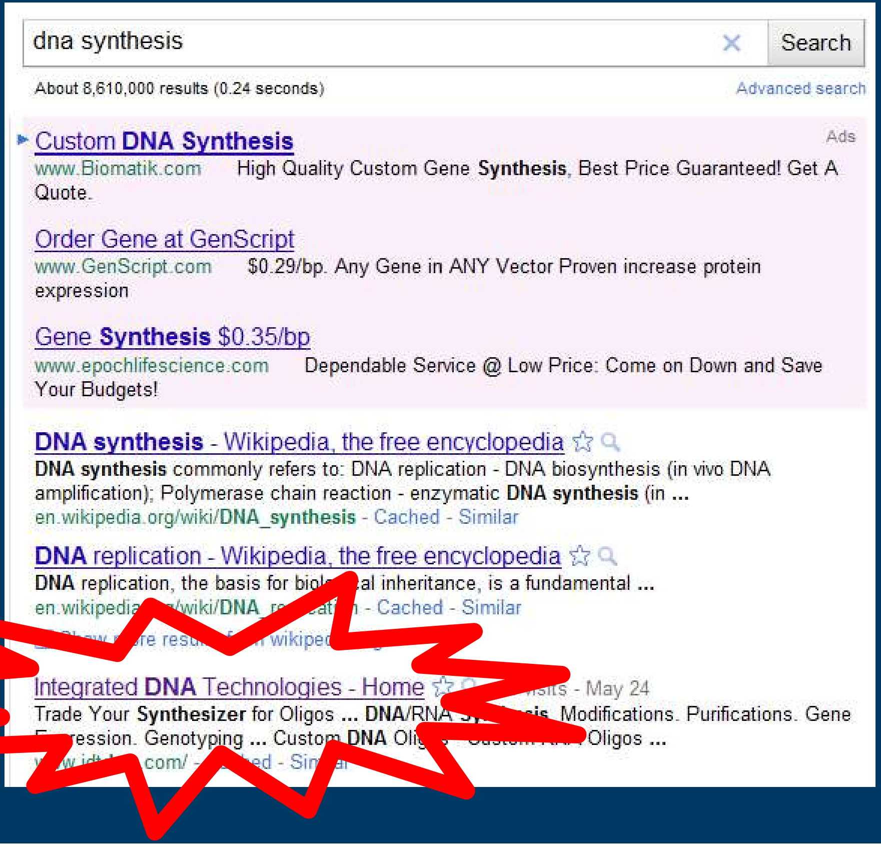


Sequence designs

Ensemble defect (nt)	Normalized ensemble defect (%)	GC content (%)	Sequence
0.2	0.3	57.5	GCUGCGAUACCCAAAAGAACAA+GCGAUCAAGCCCCUCUUUUUCC+GGGCUUGAUCGCGGGUAUCGCAGCUGCGC

"Ok, where do I buy these?"

“DNA Synthesis”



dna synthesis Search

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

Ads

Custom DNA Synthesis
www.Biomatik.com High Quality Custom Gene **Synthesis**, Best Price Guaranteed! Get A Quote.

Order Gene at GenScript
www.GenScript.com $0.29/bp. Any Gene in ANY Vector Proven increase protein expression

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

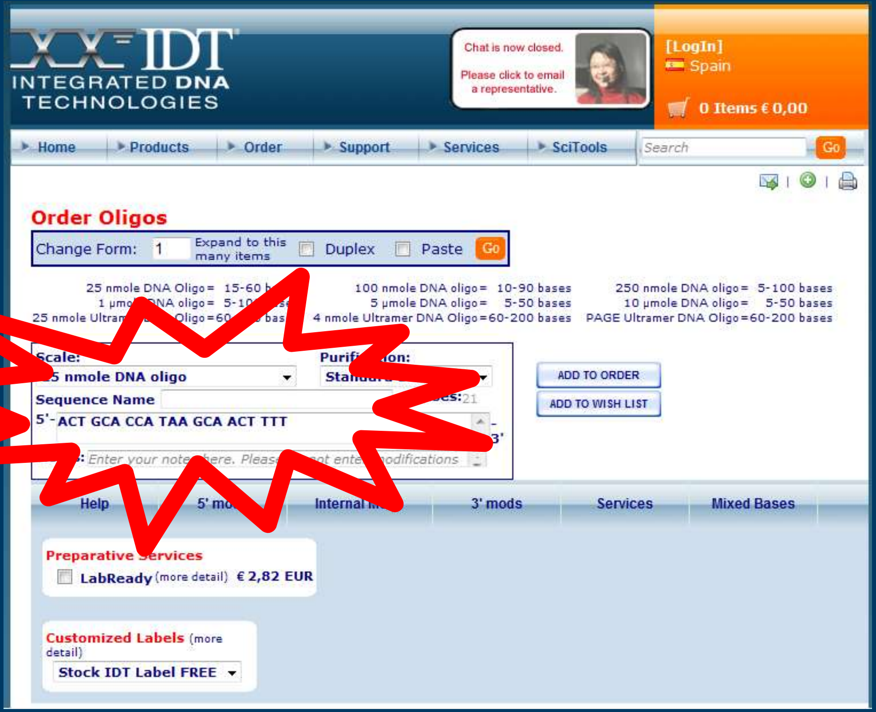
DNA synthesis - Wikipedia, the free encyclopedia
DNA synthesis commonly refers to: DNA replication - DNA biosynthesis (in vivo DNA amplification); Polymerase chain reaction - enzymatic **DNA synthesis** (in ...
en.wikipedia.org/wiki/DNA_**synthesis** - Cached - Similar

DNA replication - Wikipedia, the free encyclopedia
DNA replication, the basis for biol[illegible]al inheritance, is a fundamental ...
en.wikipedia[illegible]/wiki/DNA_re[illegible] - Cached - Similar
[illegible]re resu[illegible] wikiped[illegible]

Integrated **DNA** Technologies - Home [illegible]visits - May 24
Trade Your **Synthesizer** for Oligos ... **DNA**/RNA [illegible]is. Modifications. Purifications. Gene E[illegible]ression. Genotyping ... Custom **DNA** Oli[illegible] Oligos ...
www.idt[illegible].com/ - [illegible]ed - Sim[illegible]

From Sequences to Molecules

- Copy&Paste from nupack



INTEGRATED DNA TECHNOLOGIES

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

[LogIn]
Spain
0 Items € 0,00

Order Oligos

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

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

Scale: 5 nmole DNA oligo
Purifi ion: Stan
Sequence Name
es: 21
5'- ACT GCA CCA TAA GCA ACT TTT -3'
Enter your note here. Please ot ente odifications

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

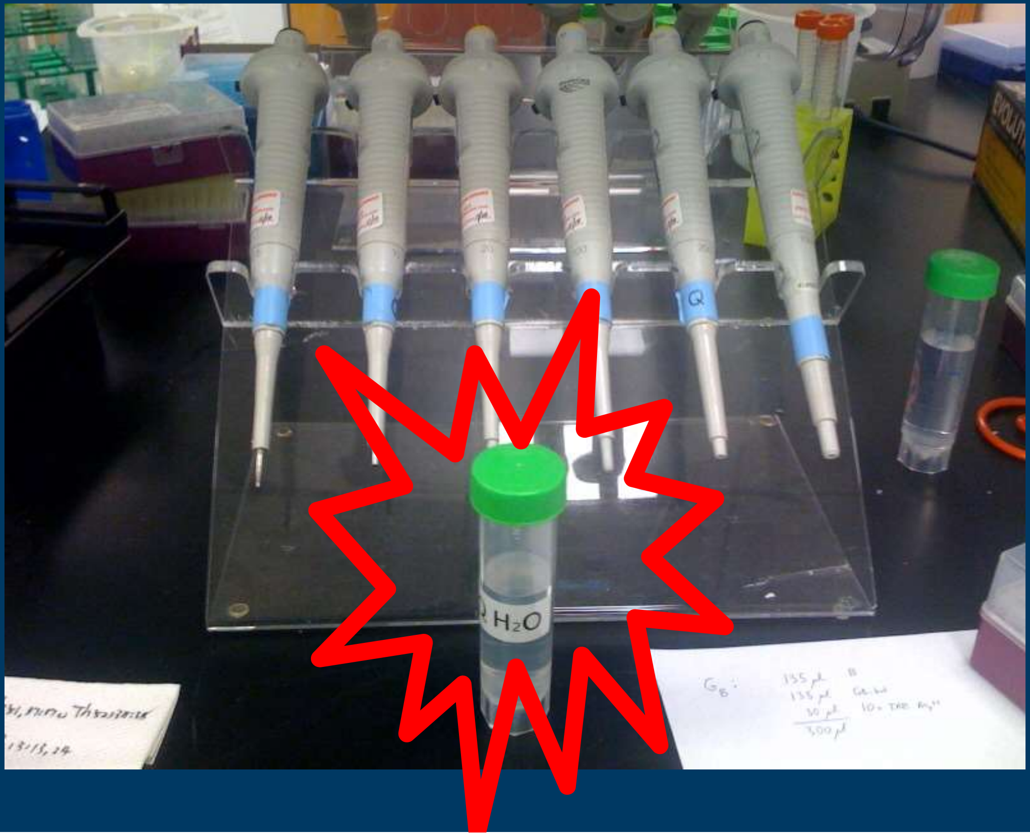
Customized Labels (more detail)
Stock IDT Label FREE

Molecules by FedEx

IDT
INTEGRATED DNA TECHNOLOGIES, INC.
Custom Oligonucleotide Synthesis
Innovation and Precision in Nucleic Acid Synthesis
www.idtdna.com

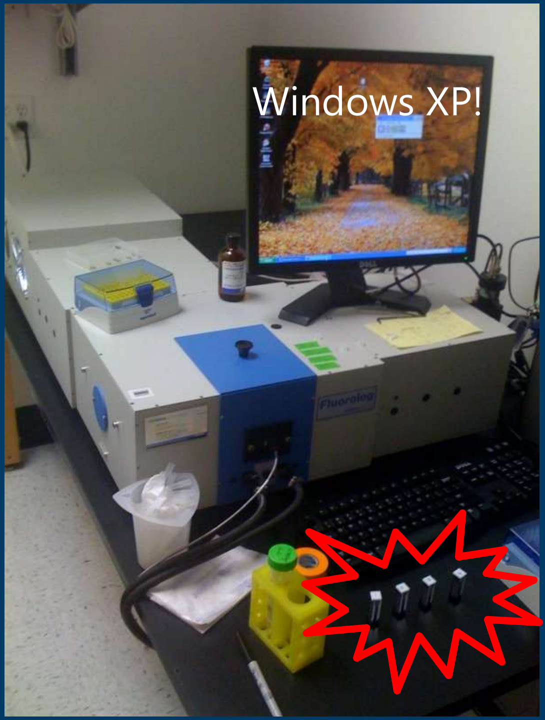
"Ok, how do I run these?"

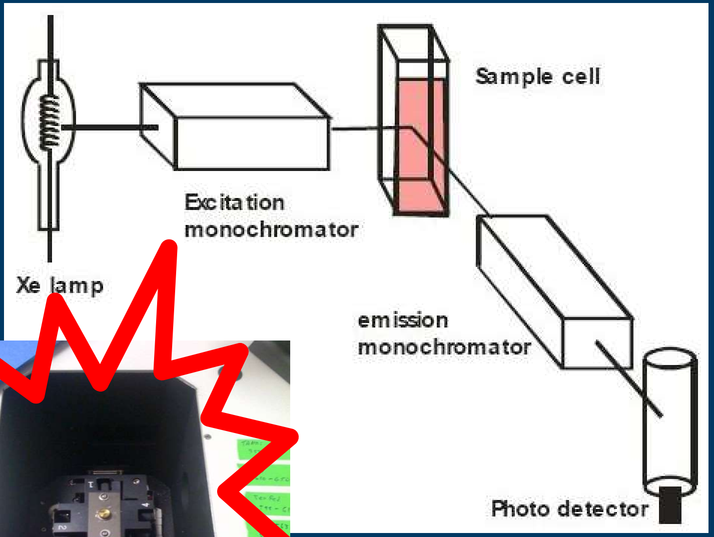
Add Water



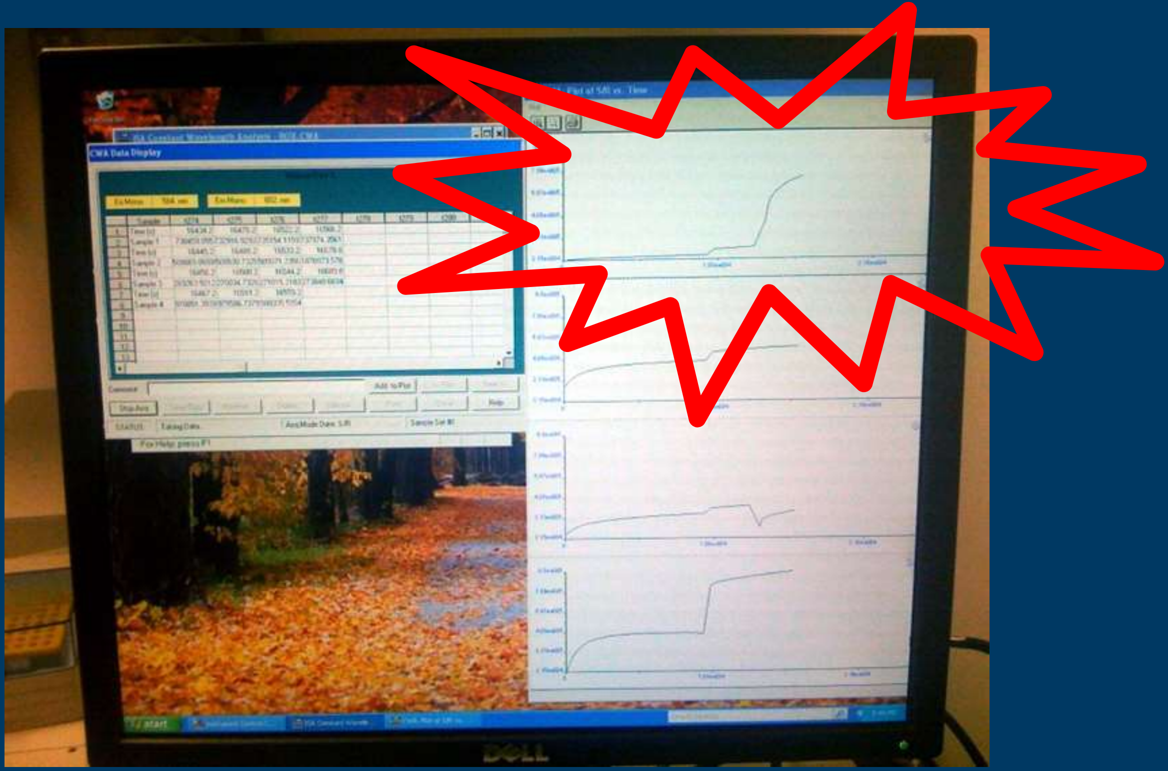
Execute (finally!)

- Fluorescence is your one-bit 'print' statement



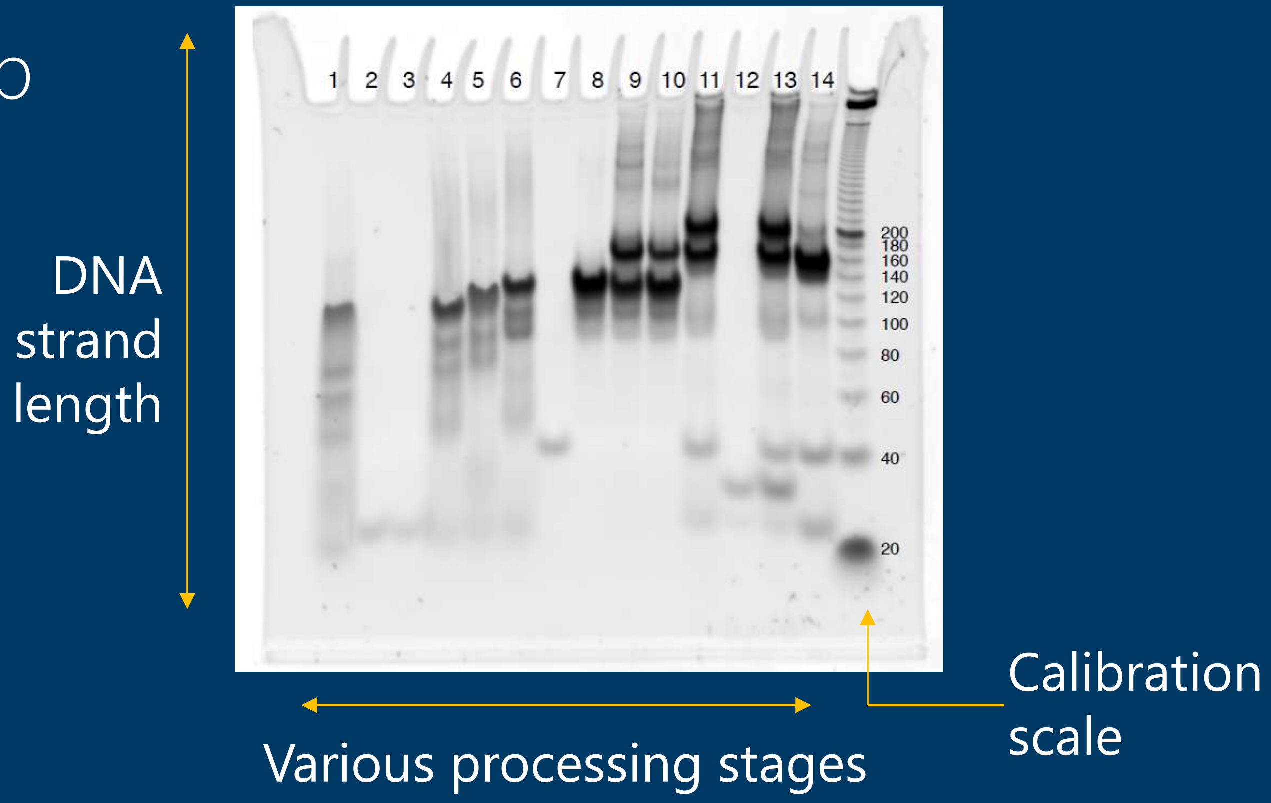


Output



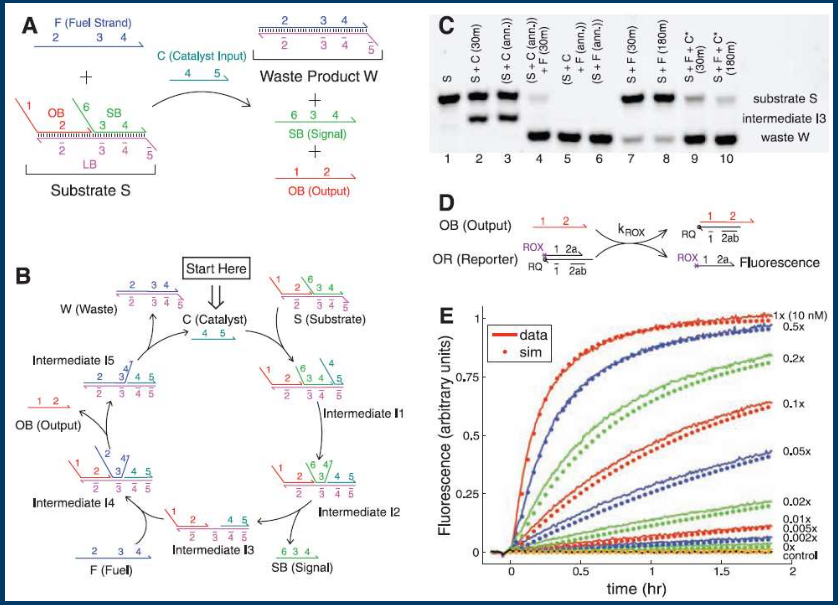
Debugging

- A core dump



Delivery!

Engineering Entropy-Driven Reactions and Networks Catalyzed by DNA
David Yu Zhang, *et al.*
Science **318**, 1121 (2007);
DOI: 10.1126/science.1148532



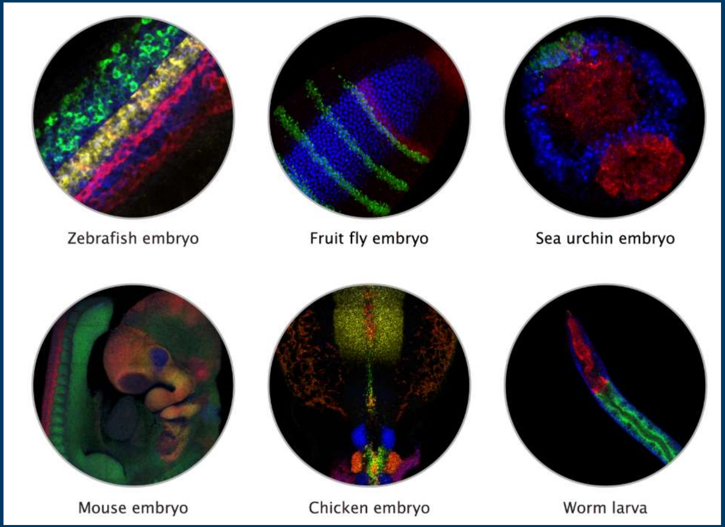
Fun Applications

RNA Rewiring

- Using RNA gates to detect, intercept, and replace messenger RNA
  - to "hotwire" cells without changing their genetic code
  - there is a similar natural process called RNA Interference, used by cells to fight viruses

Cell Staining

- Using Hybridization Chain Reaction
  - to simultaneously stain tissues in multiple colors



http://www.moleculartechnologies.org/

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[1,2], Lining Yao[2], Chin-Yi Cheng[2,3], Teng Zhang[4], Hiroshi Atsumi[5], Luda Wang[4], Guanyun Wang[2], Oksana Anilionyte...
+ See all authors and affiliations

Hacking Yoghurt

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

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

The iGEM Competition

- The Hackaton of Synthetic Biology

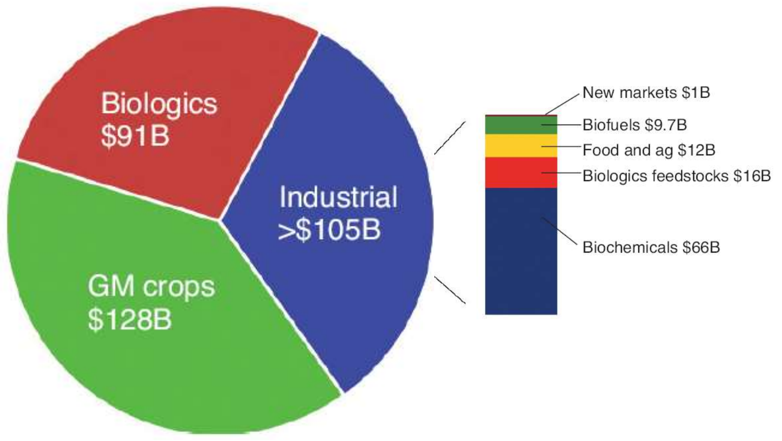
  The International Genetically Engineered Machine (iGEM) competition is a worldwide synthetic biology competition that was initially aimed at undergraduate university students, but has since expanded to include divisions for high school students, entrepreneurs, and community laboratories, as well as 'overgraduates'. *https://en.wikipedia.org/wiki/International_Genetically_Engineered_Machine*

  - Don't like how *E. coli* smell? Make them smell like bananas!
  - Fruit freshness detector
  - Gold mining bacteria in Ghana
  - etc.

Markets
Scientific Discovery
Molecular Computability

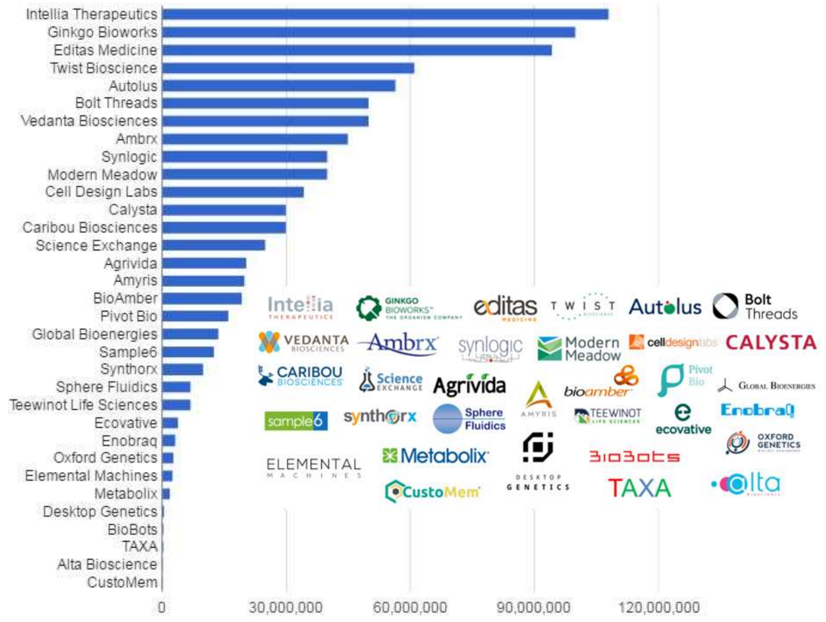
Synthetic Biology Market

Annual revenue from GMOs in the US exceeds $324Bn



Source: Rob Carlson, Nature Biotechnology, 2016

33 Programming Biology companies raised $900M in 2016



Source: SynBioBeta.com, 2016

Some (ongoing) successes stories

JUNO THERAPEUTICS

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

AMYRIS

- ($300M) Reprogram yeast to synthesise chemicals
- Antimalarial drug in production (with Sanofi)
- Jet fuel used in commercial flights (with Total)

GINKGO BIOWORKS™ THE ORGANISM COMPANY

- Supply custom organisms for bio fabrication

Modern Meadow

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

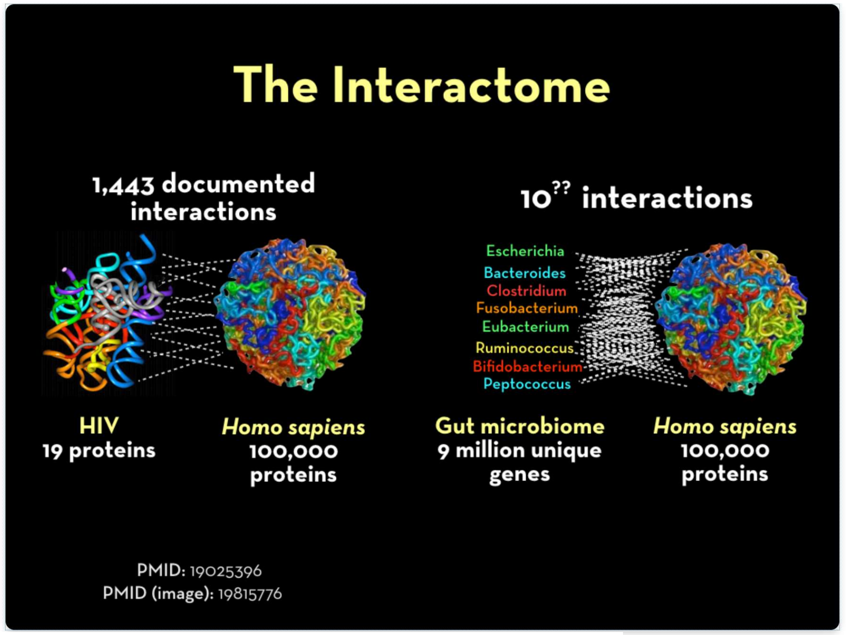
Scaling up Science

Developing these markets requires dramatically scaling up scientific discovery

Because we know so very little about biology

And there are way too many proteins to study!

Fortunately, a new virtuous circle is developing.

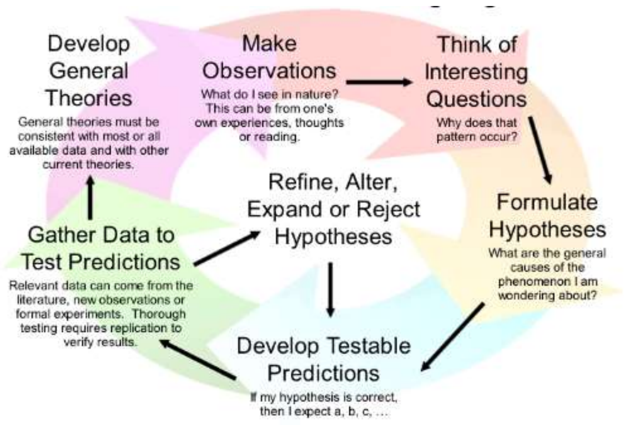


Human immunodeficiency virus type 1, human protein interaction database at NCBI.

Fu W[1], Sanders-Beer BE, Katz KS, Maglott DR, Pruitt KD, Ptak RG.

Molecular Programming and Scientific Discovery

As we learn to program physical and biological matter the process of scientific discovery will be transformed



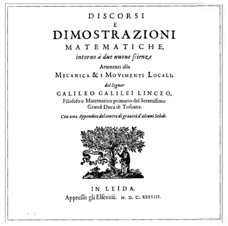
Develop General Theories
General theories must be consistent with most or all available data and with other current theories.
Make Observations
What do I see in nature? This can be from one's own experiences, thoughts or reading.
Think of Interesting Questions
Why does that pattern occur?
Formulate Hypotheses
What are the general causes of the phenomenon I am wondering about?
Develop Testable Predictions
If my hypothesis is correct, then I expect a, b, c, ...
Gather Data to Test Predictions
Relevant data can come from the literature, new observations or formal experiments. Thorough testing requires replication to verify results.
Refine, Alter, Expand or Reject Hypotheses

Garland, Jr., Theodore. "The Scientific Method as an Ongoing Process". U C Riverside.

Discovery through Observation

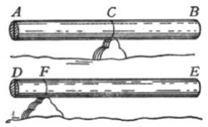
The Scientific Method ~ 1638

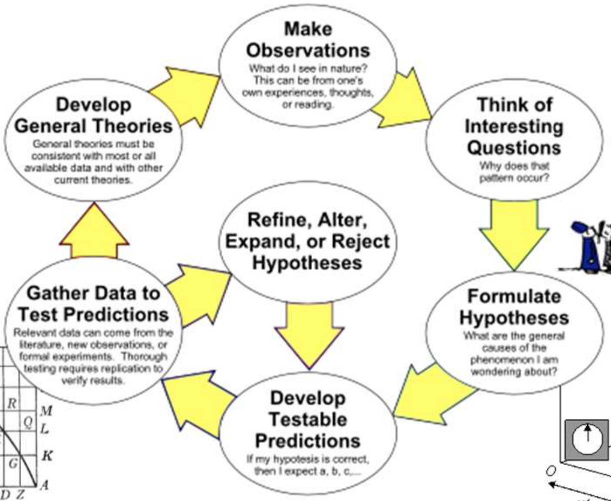
1 Guy



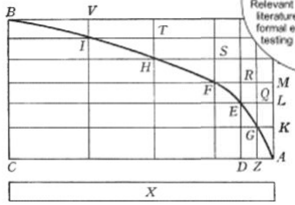
DISCORSI
E
DIMOSTRAZIONI
MATEMATICHE,
intorno à due nuoue ſcienze
Attenenti alla
MECANICA & I MOVIMENTI LOCALI,
del Signor
GALILEO GALILEI LINCEO,
Filoſofo e Matematico primario del Sereniſſimo
Grand Duca di Toſcana.
Con vna Appendice del centro di grauità d'alcuni Solidi.

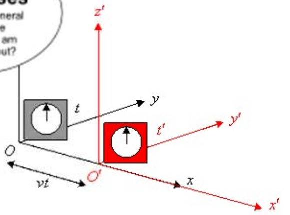
IN LEIDA,
Appreſſo gli Elſevirii. M. D. C. XXXVIII.




Make Observations
What do I see in nature? This can be from one's own experiences, thoughts, or reading.
Think of Interesting Questions
Why does that pattern occur?
Formulate Hypotheses
What are the general causes of the phenomenon I am wondering about?
Develop Testable Predictions
If my hypotesis is correct, then I expect a, b, c,...
Gather Data to Test Predictions
Relevant data can come from the literature, new observations, or formal experiments. Thorough testing requires replication to verify results.
Develop General Theories
General theories must be consistent with most or all available data and with other current theories.
Refine, Alter, Expand, or Reject Hypotheses

Garland, Jr., Theodore. "The Scientific Method as an Ongoing Process". U C Riverside.





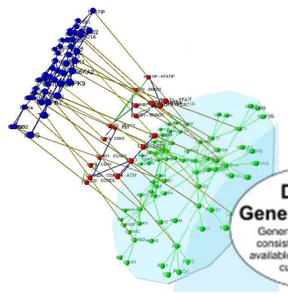
Discovery through Collaboration

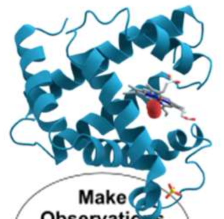
The Scientific Method ~ 2000's

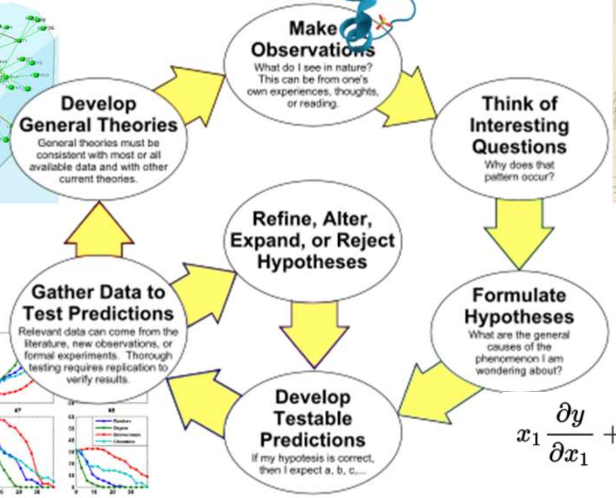
1 Lab

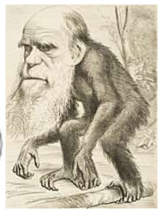
1 protein = 30 people / 30 years

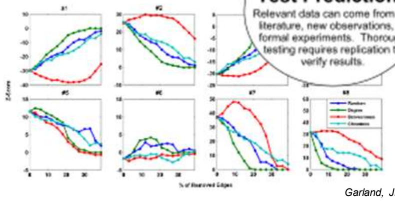
Humans have >100,000 proteins ☹






Make Observations
What do I see in nature? This can be from one's own experiences, thoughts, or reading.
Think of Interesting Questions
Why does that pattern occur?
Formulate Hypotheses
What are the general causes of the phenomenon I am wondering about?
Develop Testable Predictions
If my hypotesis is correct, then I expect a, b, c,...
Refine, Alter, Expand, or Reject Hypotheses
Gather Data to Test Predictions
Relevant data can come from the literature, new observations, or formal experiments. Thorough testing requires replication to verify results.
Develop General Theories
General theories must be consistent with most or all available data and with other current theories.





$$x_1 \frac{\partial y}{\partial x_1} + x_2 \frac{\partial y}{\partial x_2} = y$$

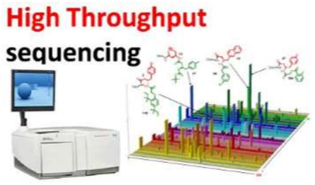
Garland, Jr., Theodore. "The Scientific Method as an Ongoing Process". U C Riverside.

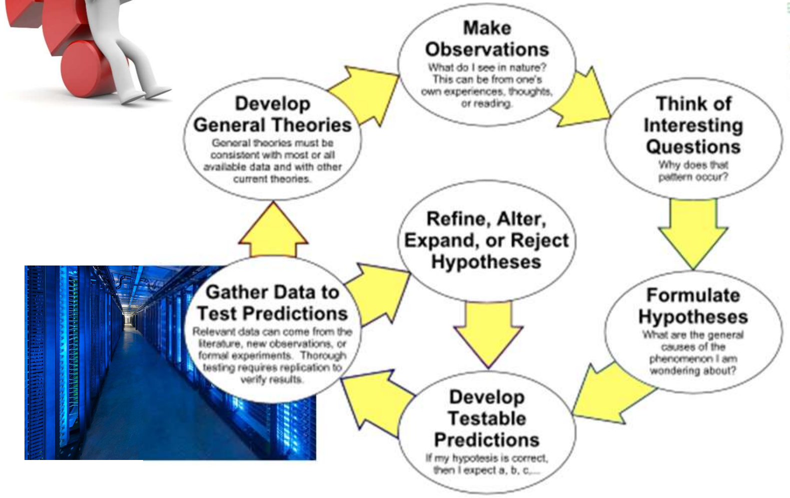
Discovery through closed-loop Automation

The Scientific Method ~ 2020's

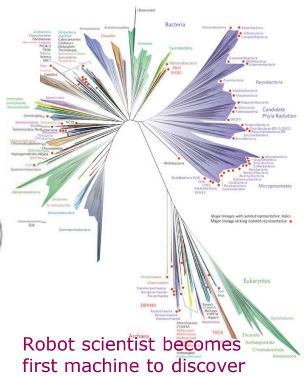
1 Program

```
while (true) {
      predict();
      falsify();
}
```




Make Observations
What do I see in nature? This can be from one's own experiences, thoughts, or reading.
Think of Interesting Questions
Why does that pattern occur?
Formulate Hypotheses
What are the general causes of the phenomenon I am wondering about?
Develop Testable Predictions
If my hypotesis is correct, then I expect a, b, c,...
Gather Data to Test Predictions
Relevant data can come from the literature, new observations, or formal experiments. Thorough testing requires replication to verify results.
Develop General Theories
General theories must be consistent with most or all available data and with other current theories.
Refine, Alter, Expand, or Reject Hypotheses

Garland, Jr., Theodore. "The Scientific Method as an Ongoing Process". U C Riverside.

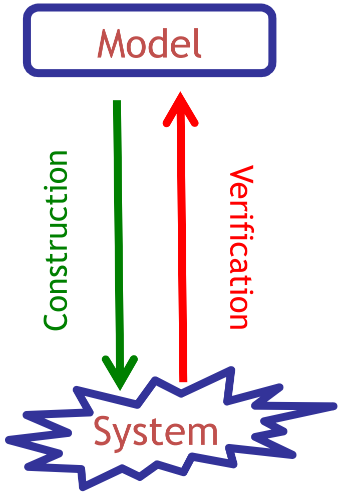


Robot scientist becomes first machine to discover new scientific knowledge

Ross King

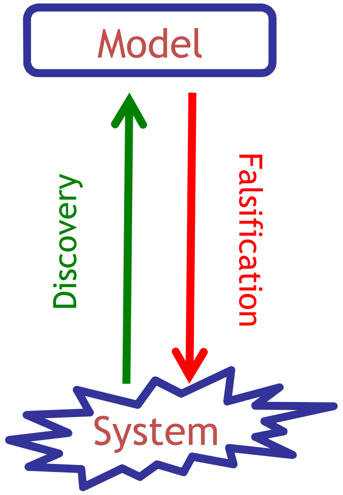
Scientific Method vs. Engineering Method

Engineering Method



Direct Engineering
(Synthetic Biology)

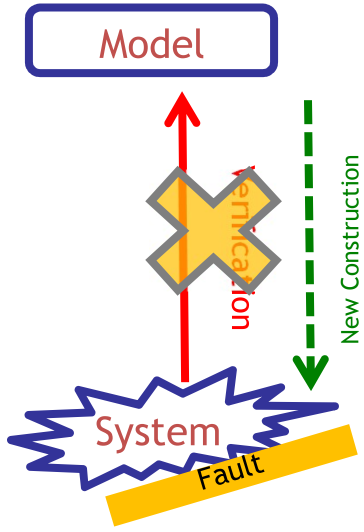
Scientific Method



Reverse Engineering
(Systems Biology)

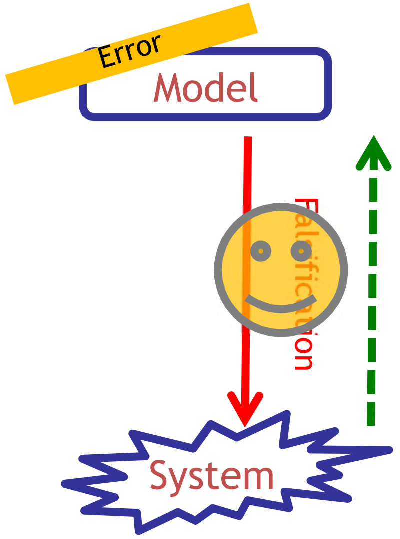
Scientific Method vs. Engineering Method

Engineering Method



Direct Engineering

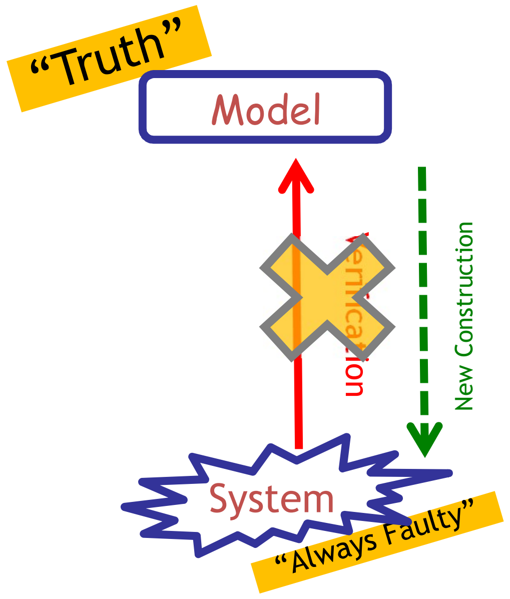
Scientific Method



Reverse Engineering

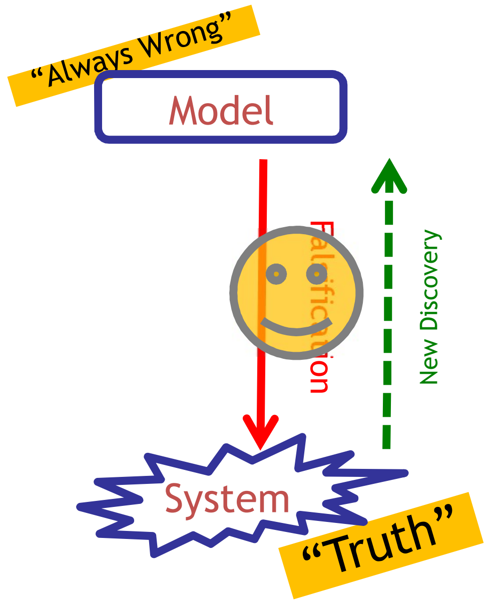
Scientific Method vs. Engineering Method

Engineering Method



Direct Engineering

Scientific Method



Reverse Engineering

Scientific Method vs. Engineering Method

When the models and the systems are *both* too complex to *either* be the full Truth

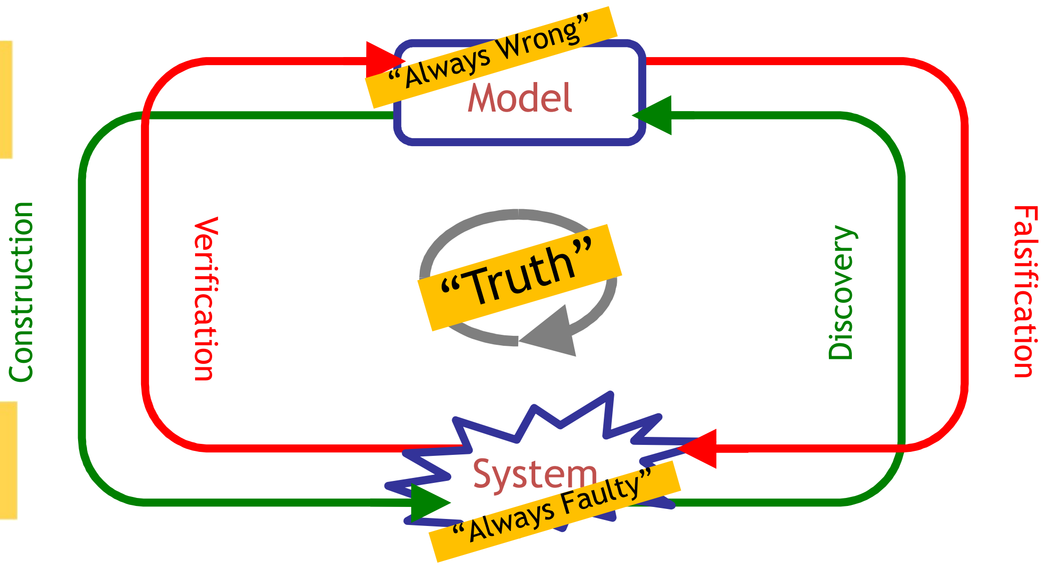
Combined Method

Now we are in Feynman territory:

What I cannot create, I do not understand.

The model is always somewhat wrong in its predictions

The system is always somewhat faulty in its execution



The Truth is not something you ever “have” but something you “maintain”

(we need to "instrument the model": change what we believe)

The models that we discover should be suitable for construction

The systems that we build should be suitable for discovery

(we need to "instrument the system": change what we study)

Theory of Molecular Computability

Those single closed-loop programs run "*half in the computer*" (the controlling software) and "*half in the organism*" (the gene network).

Shaping bacterial population behavior through computer-interfaced control of individual cells

Remy Chait[1], Jakob Ruess[1,2,3], Tobias Bergmiller[1], Gašper Tkačik[1] & Călin C. Guet[1]

In particular, we need to understand biochemical algorithms and computability from a software engineering point of view.

Today, we fundamentally understand how to program digital computers
- Classical theory of algorithms and computability

Do we fundamentally understand how to program molecular systems?
- A *different* theory of algorithms and computability (still being developed)
- To design new systems *and* understand what's there
- How biological systems can, might, and do compute

Programming with chemical reactions

$$X + Y \ \text{->}^{r} \ Z + W$$

- A fundamental model of kinetics (i.e. “behavior”) in the natural sciences
- A fundamental mathematical structure, rediscovered in many forms
  - Vector Addition Systems, Petri Nets, Bounded Context-Free Languages, Population Protocols, …
- A programming language (coded up in the genome) by which living things manage the processing of matter and information

Chemical Reaction Networks: Discrete-State Semantics

Programming Examples

Discrete (-state) Semantics

- A *state* of the system is a finite multiset of molecules; each molecule belongs to one of a finite set of *species*.
- A fixed finite set of *reactions* over species performs multiset-rewriting over those states.
- Reactions have rates: the state space is a Continuous-Time Markov Chain (a labeled transition system where labels are transition speeds).
- Hence the semantics is discrete and stochastic = atomic theory of matter.

- Issues:
  - Computing Kinetics (distribution of outcomes over time)
  - Analyzing mean, variance, and other moments
  - State reachability

Programming Examples

spec	*program*
$Y := 2X$	X -> Y + Y
$Y := \lfloor X/2 \rfloor$	X + X -> Y
$Y := X1 + X2$	X1 -> Y X2 -> Y
$Y := \min(X1, X2)$	X1 + X2 -> Y

Advanced Programming Examples

spec

Y := max(X1, X2)

program

```
X1 -> L1 + Y
X2 -> L2 + Y
L1 + L2 -> K
Y + K -> 0
```

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

(but is not computed "sequentially")

Approximate Majority

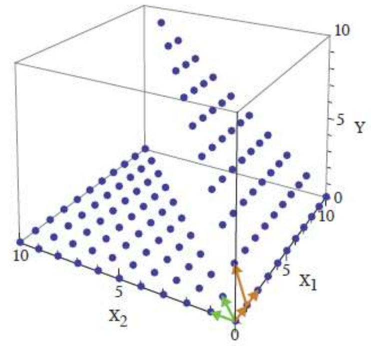
```
(X,Y) :=
        if X≥Y then (X+Y, 0)
        if Y≥X then (0, X+Y)
```

```
X + Y -> Y + B
Y + X -> X + B
B + X -> X + X
B + Y -> Y + Y
```

What can we compute this way?

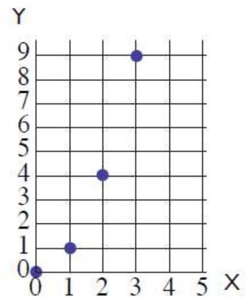
- The semilinear functions
  - Those whose graph is a finite union of linearly-bounded regions

$f(x_1, x_2)=x_2$ if $x_1>x_2$ and 0 otherwise



$\{n_1 \cdot (1,1,0) + n_2 \cdot (0,1,0) \mid n_1, n_2 \in \mathbb{N}\} \cup$
$\{ (1,0,0) + n_1 \cdot (1,1,1) + n_2 \cdot (1,0,0) \mid n_1, n_2 \in \mathbb{N}\}$

$f(x) = X^2$



not semilinear

Chen, Doty, Soloveichik, "Deterministic Function Computation with Chemical Reaction Networks" (2013)

But also Register Machines (almost…)

i: INC R_1; JMP j	$PC_i \rightarrow R_1 + PC_j$
i: DEC R_1; JMP j	$PC_i + R_1 \rightarrow PC_j$
i: IF R_2>0 {INC R_1; JMP j}	$PC_i + R_2 \rightarrow R_2 + R_1 + PC_j$
i: IF R_2=0 …	??? Whatever trick we use will have some error

- Turing-complete up to an arbitrarily small error
  - The error bound is set in advance uniformly for any computation of arbitrary length (because we cannot know how long the computation will last), and the machine will progressively "slow down" to always stay below that bound.

- David Soloveichik, Matt Cook, Erik Winfree, Shuki Bruck, "Computation with Finite Stochastic Chemical Reaction Networks". [Natural Computing, (online Feb 2008), or Technical Report: CaltechPARADISE:2007.ETR085: .pdf]

Chemical Reaction Networks: Continuous-State Semantics

Programming Examples

Continuous (-state) Semantics

- A state of the system is a (real-valued) concentration for each species.
- A fixed finite set of reactions act (continuously) on such states.
- The Law of Mass Action describes how the system evolves in continuous time.
  - Each reaction acts with a "speed" that is proportional to the product of the concentrations on its left-hand-side, multiplied by its rate.
  - Each species concentration increases or decreases according to the sum of the effects of all the reactions.
- Issues:
  - Computing Kinetics (outcomes over time)
  - Analyzing Equilibria (steady-states, etc.)
  - Model Reduction

Sniffers, buzzers, toggles and blinkers

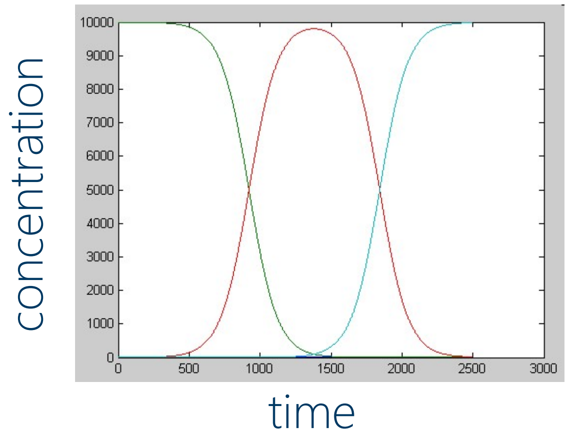
- Sigmoidal response (*buzzer*)
- Perfect adaptation (*sniffer*)
- Positive feedback
  - – Mutual activation (*one way switch*)
  - – Mutual inhibition (*toggle switch*)
- Negative feedback
  - – homeostasis
  - – oscillations (*Blinker*)

Tyson JJ - *Sniffers, buzzers, toggles and blinkers.* Curr Opin Cell Biol. 2003 Apr;15(2):221-31.

http://www.inf.ed.ac.uk/teaching/courses/csb/CSB_lecture_dynamic_signalling_and_gene_expression.pdf

Making Waves

How to program a *symmetric* wave?



A+B -> B+B
B+C -> C+C

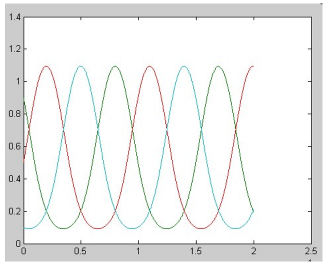
$dA/dt = -AB$
$dB/dt = AB-BC$
$dC/dt = BC$

Synthesizing programs such as this from specifications

Syntax-Guided Optimal Synthesis for Chemical Reaction Networks. Luca Cardelli, Milan Ceska, Martin Fränzle, Marta Kwiatkowska, Luca Laurenti, Nicola Paoletti, Max Whitby.
Computer Aided Verification, CAV'17.

Making Clocks

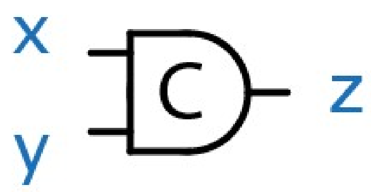
- Large literature going back to Lotka in the 1920's
- *Minimal* oscillators still a topic of interest
  - How many species? How many reactions? How symmetrical?
  - How sensitive to parameters?
  - Free running or self-regulating (limit-cycle)?
- Ex: one built with DNA strand displacement



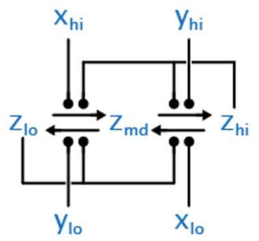
A+B -> B+B
B+C -> C+C
C+A -> A+A

Niranjan Srinivas, James Parkin, Georg Seelig, Erik Winfree, David Soloveichik, "Enzyme-free nucleic acid dynamical systems".
[Preprint: bioRxiv: .pdf paper and .pdf supplementary information]

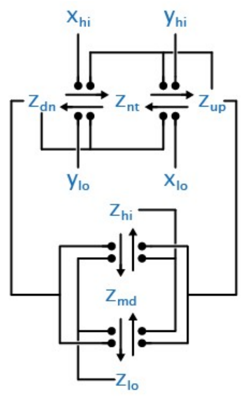
Avoiding Clocks



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



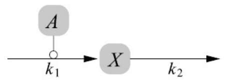
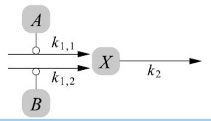
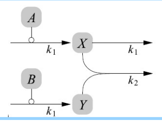
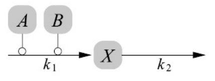
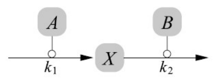
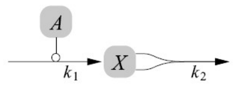
Core C-Element
(AM with external inputs)



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.

Steady-State Arithmetic

Copy	[X] := [A]		$A \xrightarrow{k_1} A+X,$ $X \xrightarrow{k_2} .$
Add	[X] := [A]+[B]		$A \xrightarrow{k_{1,1}} A+X,$ $B \xrightarrow{k_{1,2}} B+X,$ $X \xrightarrow{k_2} .$
Subtract	[X] := [A]-[B] (or 0)		$A \xrightarrow{k_1} A+X,$ $B \xrightarrow{k_1} B+Y,$ $X \xrightarrow{k_1},$ $X+Y \xrightarrow{k_2} .$
Multiply	[X] := [A]*[B]		$A+B \xrightarrow{k_1} A+B+X,$ $X \xrightarrow{k_2} .$
Divide	[X] := [A]/[B]		$A \xrightarrow{k_1} A+X,$ $B+X \xrightarrow{k_2} B.$
Root	[X] := sqrt[A]		$A \xrightarrow{k_1} A+X,$ $X+X \xrightarrow{k_2} .$

H. J. Buisman et al. Computing Algebraic Functions with Biochemical Reaction Networks

Computing Algebraic Functions

H. J. Buisman et al. Computing Algebraic Functions with Biochemical Reaction Networks

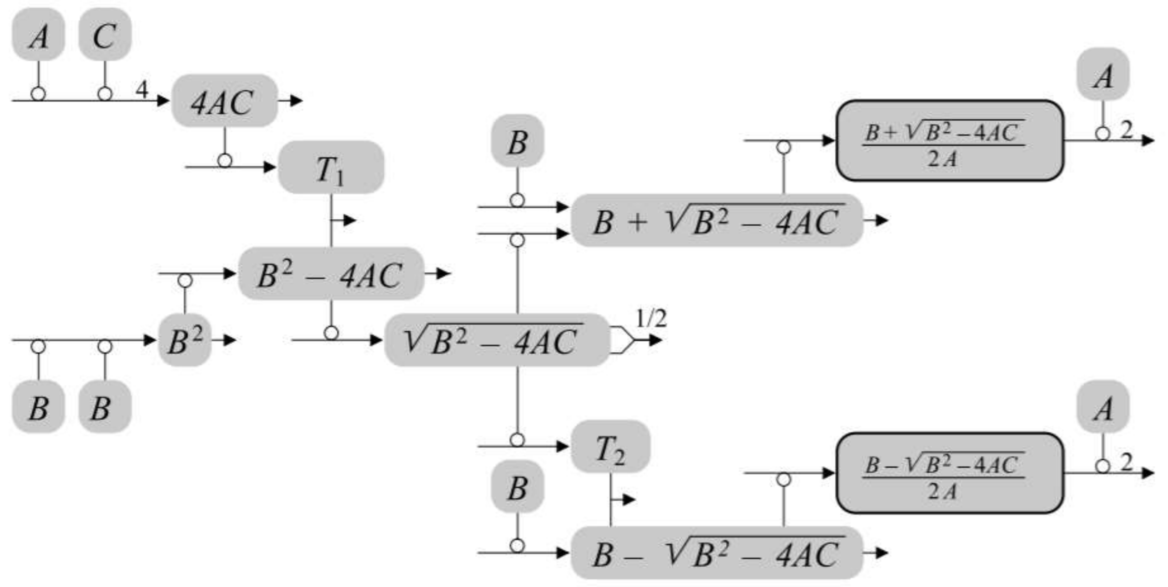


Figure 8. The quadratic formula for finding (the positive real parts of) the roots of $ax^2 - bx + c = 0$. Each of the species in the network has been given a name that represents its steady state concentration. The output species of the computation are highlighted with a black border.

Solving Algebraic Equations

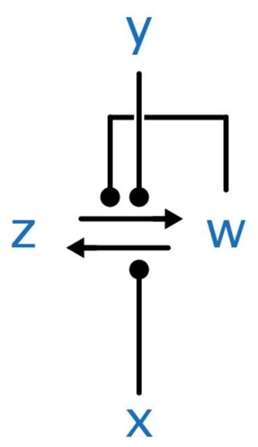
$1/\varphi = \varphi - 1$

Golden Ratio (-conjugate)

$Z + Y \rightarrow Y + W$
$W + X \rightarrow X + Z$
$Z + W \rightarrow W + W$

Init x=y=w=1.0
Init z = 0.0
all rates 1.0

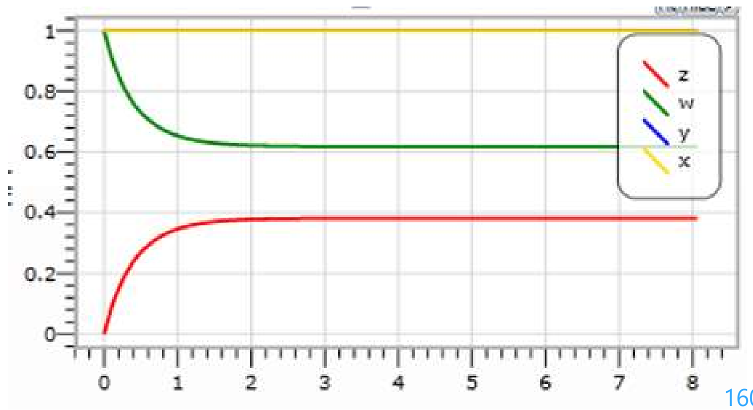
Golden Ratio



Then (we can easily show analytically by the mass action ODEs that) at steady state:

$$1/w = w - 1$$

hence $W = \varphi = 0.61803\ldots$

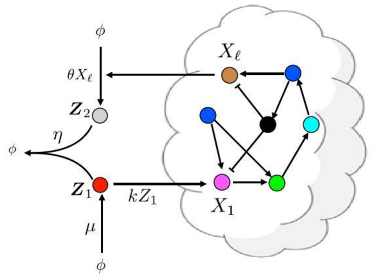


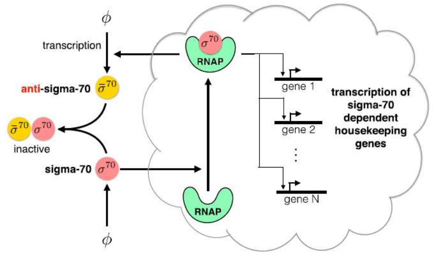
All algebraic equations can be solved [Ref]

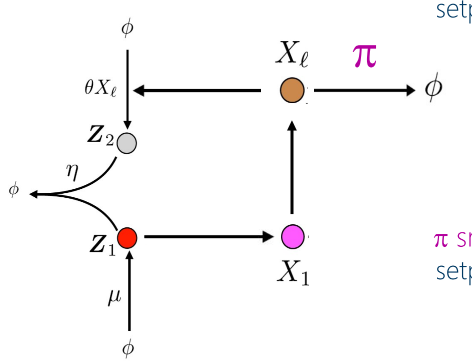
Antithetic Integral Feedback Controller

Antithetic Integral Feedback Ensures Robust Perfect Adaptation in Noisy Biomolecular Networks

Corentin Briat,[1,2] Ankit Gupta,[1,2] and Mustafa Khammash[1,*]
[1]Department of Biosystems Science and Engineering (D-BSSE), ETH–Zürich, Mattenstrasse 26, 4058 Basel, Switzerland
[2]Co-first author

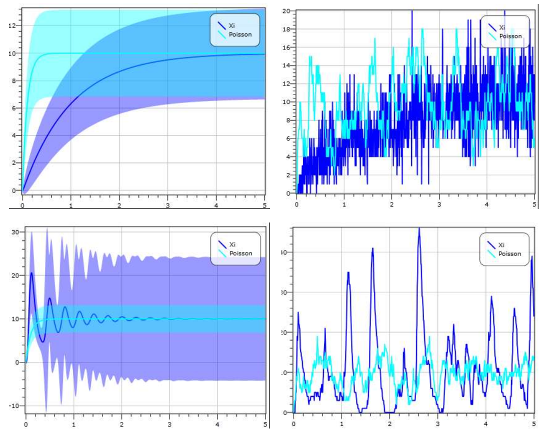






π large
setpoint ->

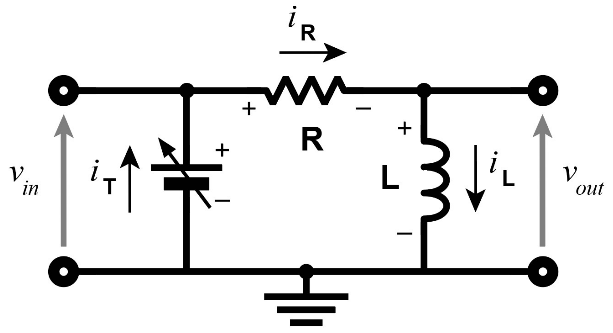
π small
setpoint ->



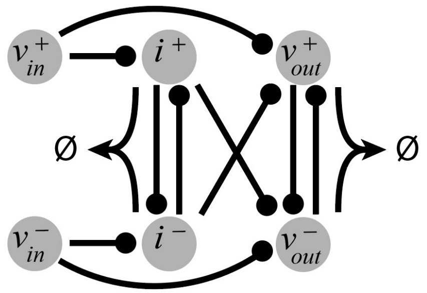
The difference between Z_1 and Z_2
is proportional to the integral of the error.

From Electrical Circuits to Chemical Networks

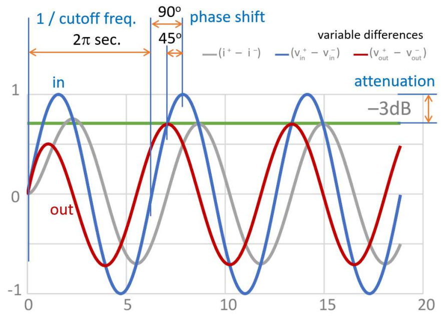
Take any textbook electric circuit:
(or, technically, a linear Differential Algebraic Equation system)

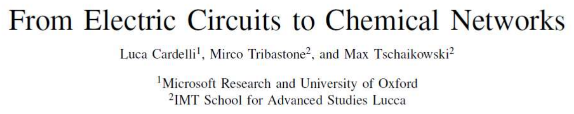


And algorithmically produce a Chemical Reaction Network :



That does the same exact thing:





From Electric Circuits to Chemical Networks

Luca Cardelli[1], Mirco Tribastone[2], and Max Tschaikowski[2]

[1]Microsoft Research and University of Oxford
[2]IMT School for Advanced Studies Lucca

Finally, Some *Bad* Programs

X -> Y

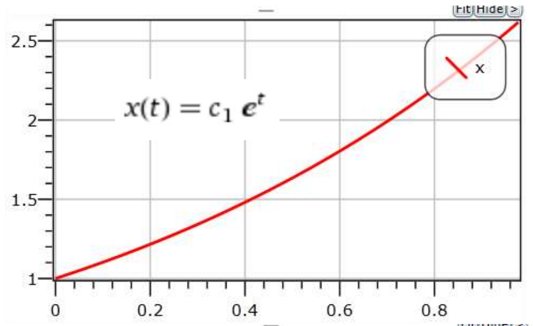
Violates thermodynamics.
(Assume there is a tiny reverse reaction.)

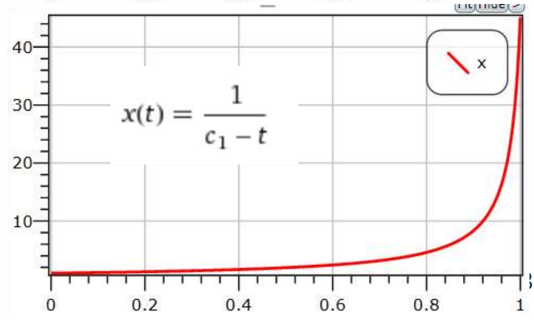
X -> X + X

Violates conservation of mass.
(No biggie, assume there is inflow/outflow.)

X + X -> X + X + X

Violates finite density.
(This is really bad.)





Chemical Reaction Networks: What do they mean?

Wait, there are *two* semantics?

- In a given volume are there
  - (A) A finite number of molecules? or
  - (B) A continuous concentration of <something>?

- Does it make a difference?
  - Related by Avogadro's number: #molecules = concentration * Avogadro
  - But finite density issues: concentration is not unbounded in the discrete model: the program 2X -> 3X will stop when there is no more "space" for molecules

Are these programs equivalent? (YES!)

AM with 4 reactions

$X + Y \to Y + B$
$Y + X \to X + B$
$B + X \to X + X$
$B + Y \to Y + Y$

AM with 3 reactions

$X + Y \to B + B$
$B + X \to X + X$
$B + Y \to Y + Y$

Same *identical* ODEs => EQUIVALENT

$$dX/dt = -XY + BX$$
$$dY/dt = -YX + BY$$
$$dB/dt = 2XY - BX - BY$$

Are these programs equivalent? (NO!)

- With 3 reactions:
  - {X, Y} -> {B, B} in one step, then stop

```
X + Y -> B + B
B + X -> X + X
B + Y -> Y + Y
```

- With 4 reactions:
  - {X, Y} -> ({X, B} or {Y, B}) -> ({X, X} or {Y, Y}), then stop
  - (no {B, B} final state)

```
X + Y -> Y + B
Y + X -> X + B
B + X -> X + X
B + Y -> Y + Y
```

- Different final states => NOT EQUIVALENT
  - The 3-reaction version fails the requirement that in the end one of the outputs should be the sum of the inputs.

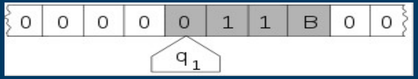
Who is right?

- #1: Believe the discrete nature of atoms (and cells): there are no continuous concentrations
- #2: Believe the analytical power of calculus: a useful approximation in appropriate conditions
- Biologists have (quite recently) realized that #1 must be taken seriously, because of advances in laboratory equipment that allow examining single molecules and single cells.

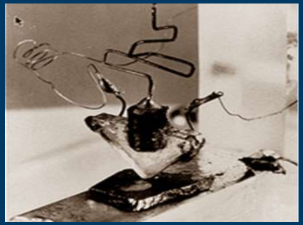
Final Remarks

A Brief History of DNA

Turing Machine, 1936

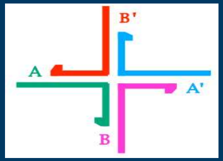


Transistor, 1947

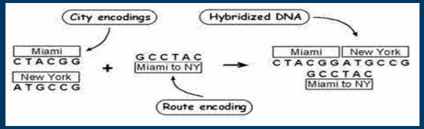


DNA, -3,800,000,000

Structural DNA Nonotech, 1982



DNA Algorithm, 1994



Computer programming

Systematic manipulation of information

Systematic manipulation of matter

Molecular programming

20th century

21th century

Acknowledgments

- Microsoft Research
  - Andrew Phillips, Biological Computation Group
- Caltech
  - Winfree Lab
- U.Washington
  - Seelig Lab

Biological Computation Group

Microsoft Research Cambridge

Andrew Phillips

Luca Cardelli

Neil Dalchau

Boyan Yordanov

Sara-Jane Dunn

Colin Gravill

Paul Grant

Carlo Spaccasassi

Filippo Polo

Collaborators

Programming DNA

University of Washington: Georg Seelig (GS), Gourab Chatterjee (GC), Suzie Pun (SP)
University of New Mexico: Matthew Lakin (ML)
Rice University: Dave Zhang (DZ)
University of Cambridge: Ulrich Keyser (UK), Elisa Hemmig (EH)
Microsoft Research: Karin Strauss (KS), Yuan Chen (YC), Alex Gaunt (AG), Ryota Tomioka (RT), Ted Meeds (TM).
Caltech: Frits Dannenberg (FD)

Programming Genetic Devices

University of Cambridge: James Locke (JL), Niall Murphy (NM), Jim Ajioka (JA), Jim Haseloff (JH), Om Patange (OP), Eugene Nadezhdin (EN)
UCL: Chris Barnes (CB), Luca Rosa (LR)

Reprogramming Stem Cells

University of Cambridge: Austin Smith (AS), Amy Li (AL), Brian Hendrich (BH), Nicola Reynolds (NR), Bertille Montibus (BM)
University of Padova: Graziano Martello (GM)
Microsoft Research: Christoph Wintersteiger (CW)
Kings College London: Angela Oliveira Pisco (AOP), Fiona Watt (FW)
University of Toronto: Peter Zandstra (PZ)

Microsoft Research: Agile Projects Team (APT),
Industry: Horizon, Twist, Synthace.

Resources

- Biological Computation Group at MSR
  https://www.microsoft.com/en-us/research/group/biological-computation/
- Molecular Programming Project at Caltech
  http://molecular-programming.org/
- Georg Seelig's DNA Nanotech Lab at U.W. CS&E
  http://homes.cs.washington.edu/~seelig/
- "DNA Computing and Molecular Programming" Conference Proceedings
  http://www.dna-computing.org/