

Microsoft Engineering Excellence

Molecular Programming

Luca Cardelli, Principal Researcher, MSR Cambridge
Programming Principles and Tools

2012-12-06

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

Microsoft Engineering Excellence

The Hardware Argument

Smaller and smaller things can be built

Smaller and Smaller

First working transistor

John Bardeen and Walter Brattain , Dec. 23, 1947

First integrated circuit

Jack Kilby, Sep. 1958.

50 years later

25nm NAND flash

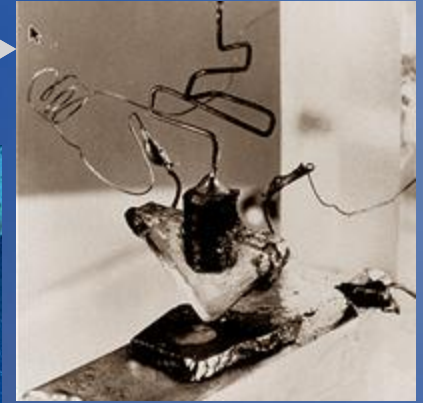
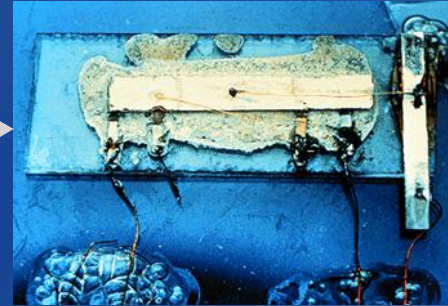
Intel&Micron, Jan. 2010. ~50atoms

Single molecule transistor

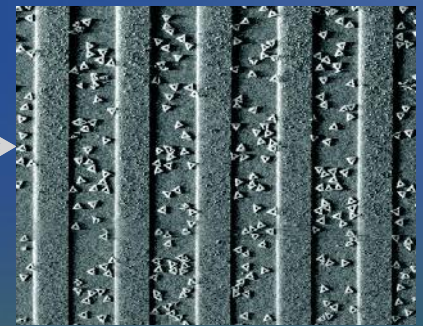
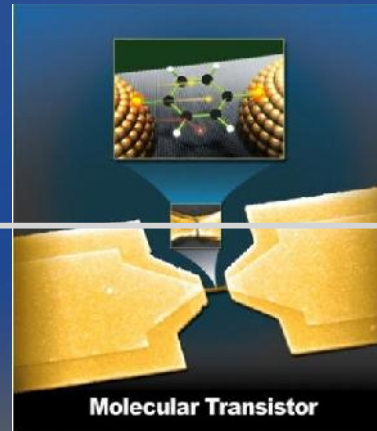
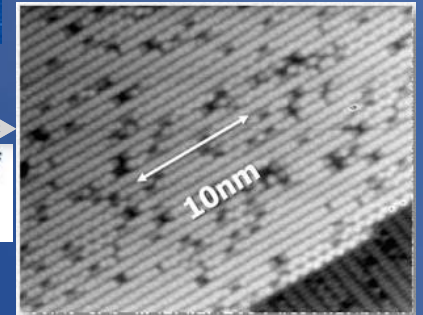
Observation of molecular orbital gating
Nature, 2009; 462 (7276): 1039

Molecules on a chip

~10 Moore's Law cycles left!



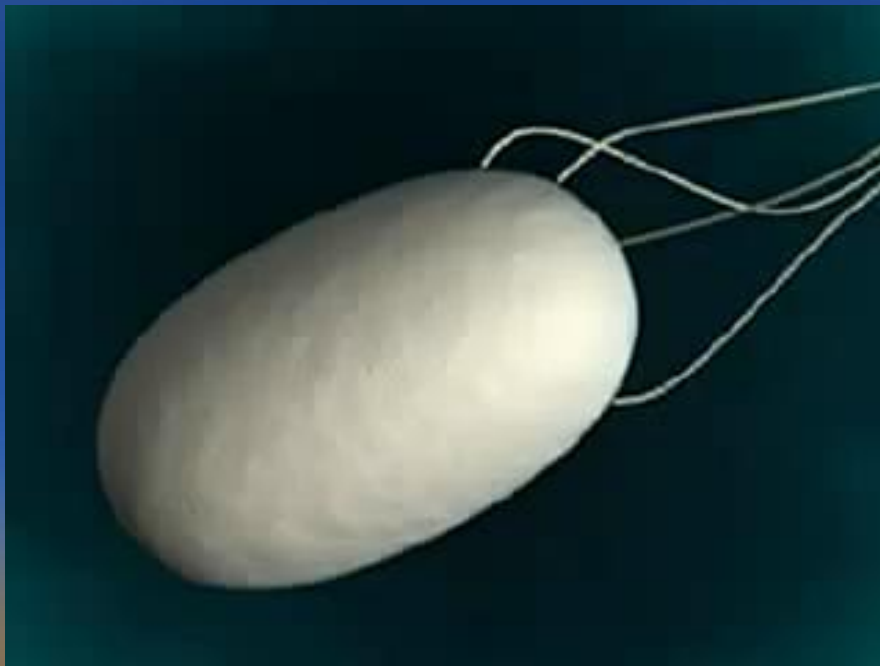
Scanning tunneling microscope image of a silicon surface showing 10nm is ~20 atoms across



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

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*



Garland Science
Taylor & Francis Group

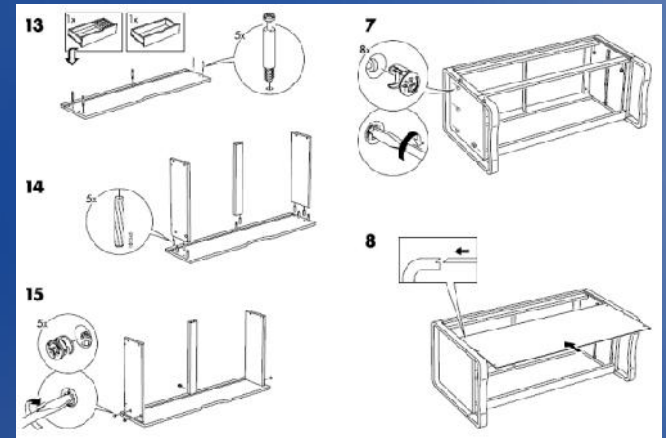
Microsoft Engineering Excellence

Microsoft Confidential

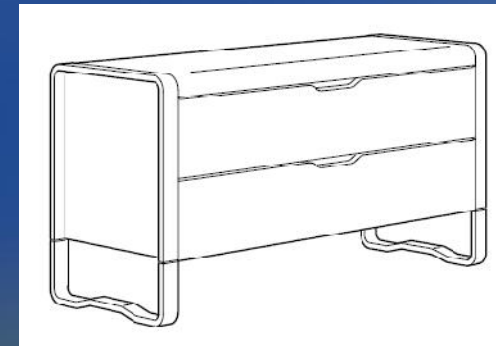
www.youtube.com/watch?v=Ey7Emmddf7Y

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...

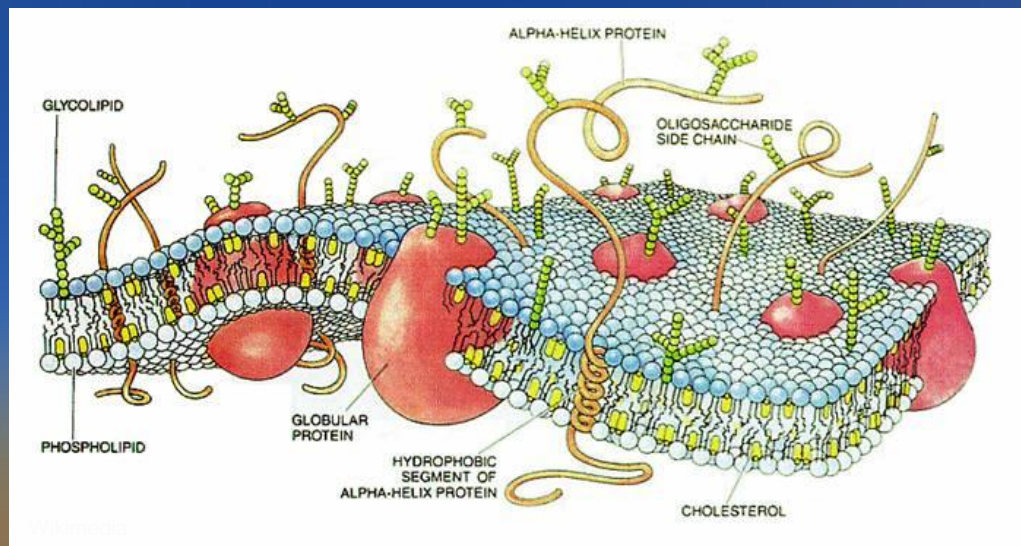


↓ Add water

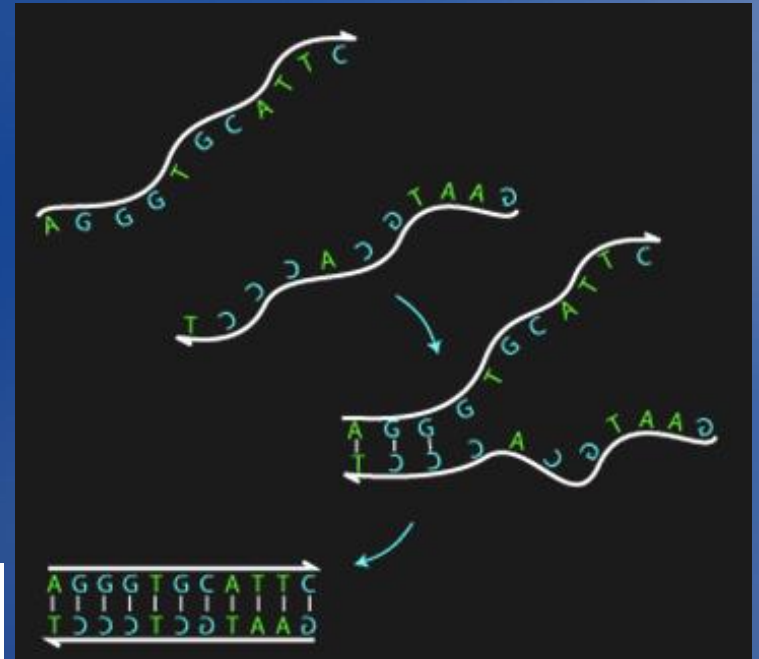


Programmed Self-Assembly

Proteins



DNA/RNA



Membranes

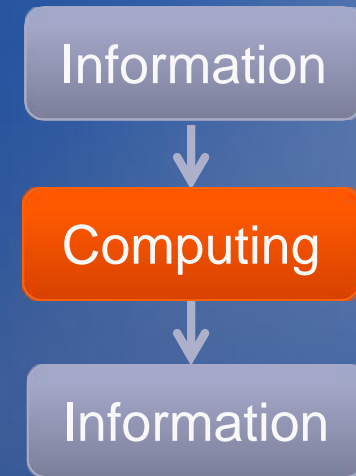
Microsoft Engineering Excellence

The Software Argument

Smaller and smaller things can be programmed

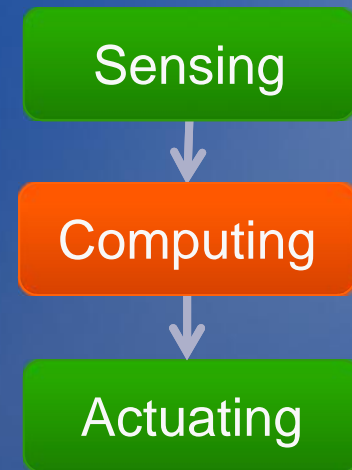
We can program...

- Computers
 - Completely!



We can program...

- Physical systems
 - Completely!
 - Modulo sensors/actuator capabilities



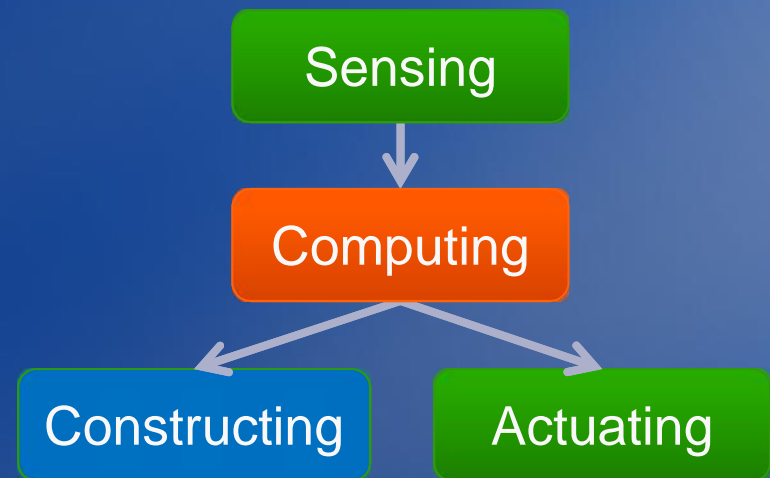
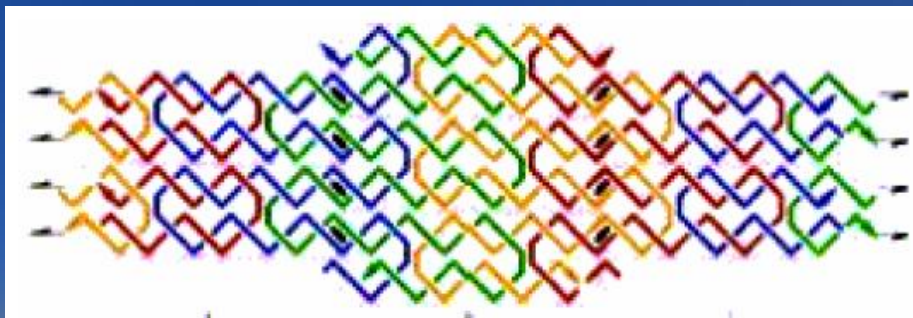
We can program...

- **Matter**

- Completely
- Directly!

- **Which matter?**

- Currently: only DNA/RNA
- But this is not so limiting...

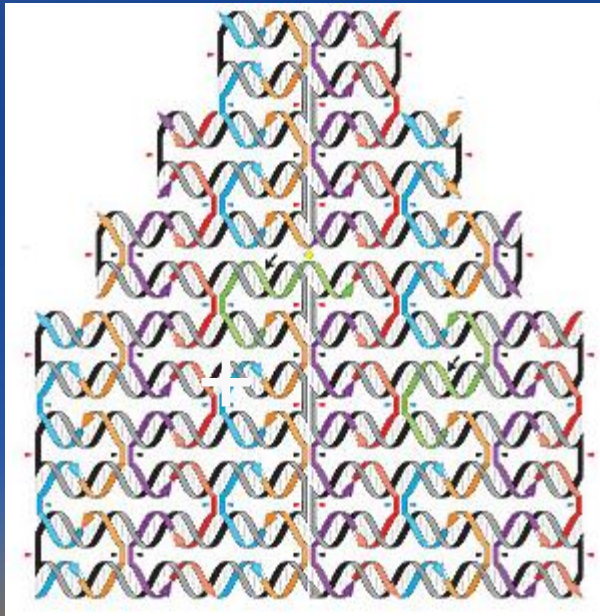


What can we do with it?

- Organize ANY matter [caveats apply]
- Execute ANY kinetics [caveats: up to time scaling]
- Control molecular systems
- Interface to Biology

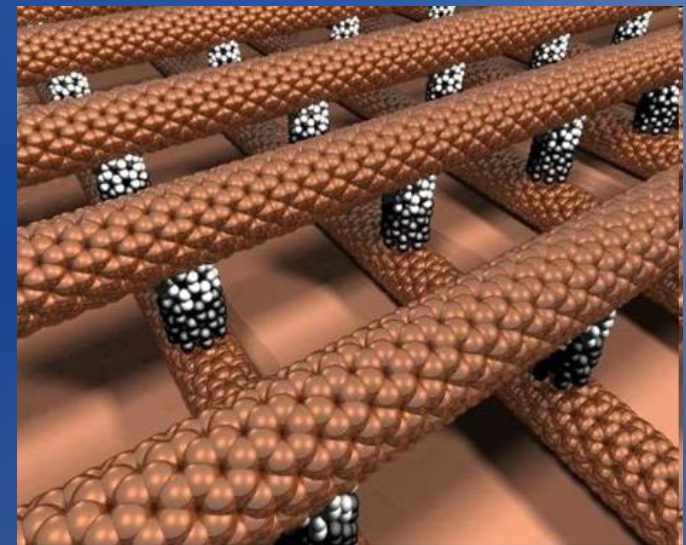
Organizing Any Matter

- 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 Desired Kinetics

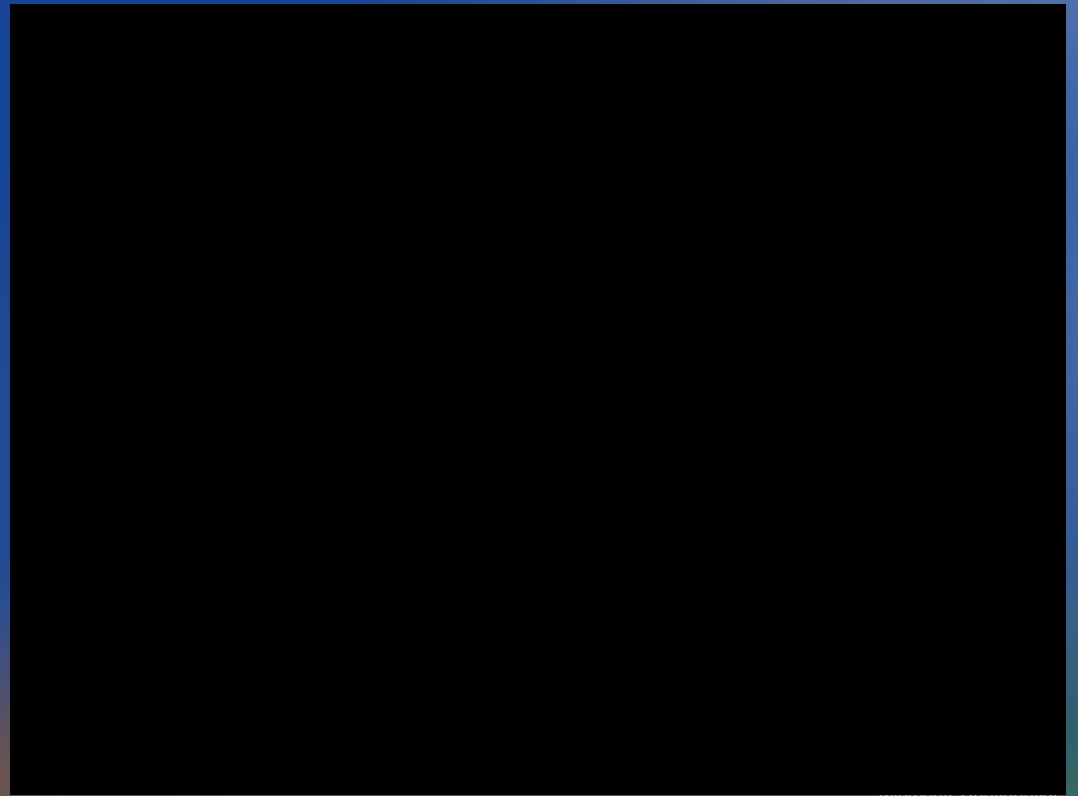
- The kinetics of any finite network of chemical reactions among abstract species, can be executed (physically) with especially programmed DNA molecules
- Chemical reactions as an executable programming language for dynamical systems!

Two-Domain DNA Strand Displacement

Luca Cardelli (Microsoft Research)

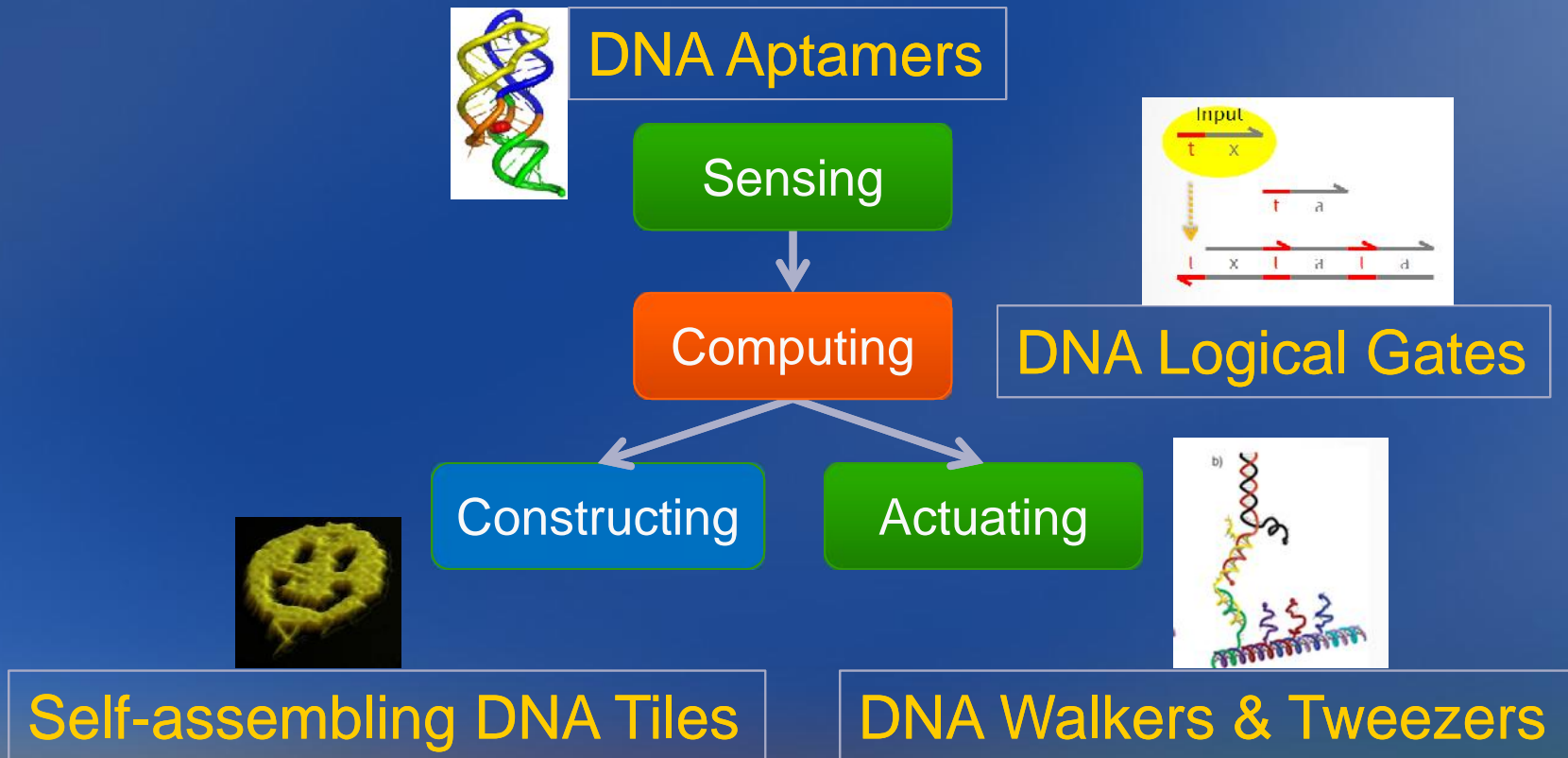
DNA as a universal substrate for chemical kinetics

David Soloveichik^{a,1}, Georg Seelig^{a,b,1}, and Erik Winfree^{c,1}



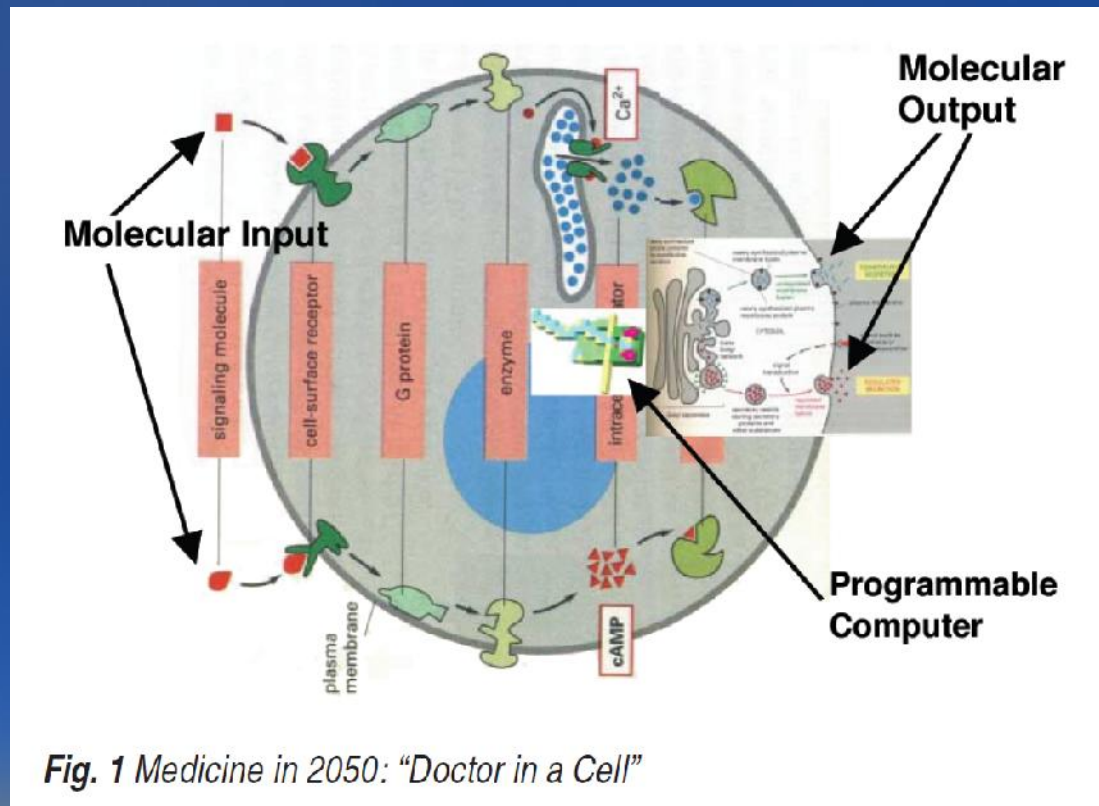
Building Molecular Controllers

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



Interfacing to Biology

A doctor in each cell



Ehud Shapiro

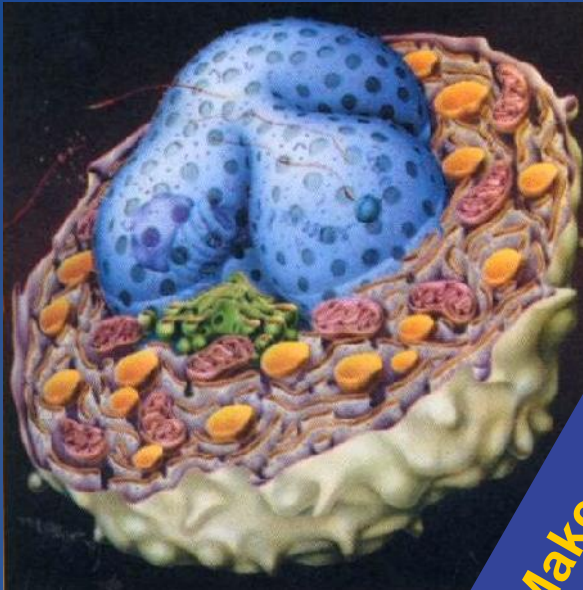
Rivka Adar
Kobi Benenson
Gregory Linshitz
Aviv Regev
William Silverman

**Molecules and
computation**

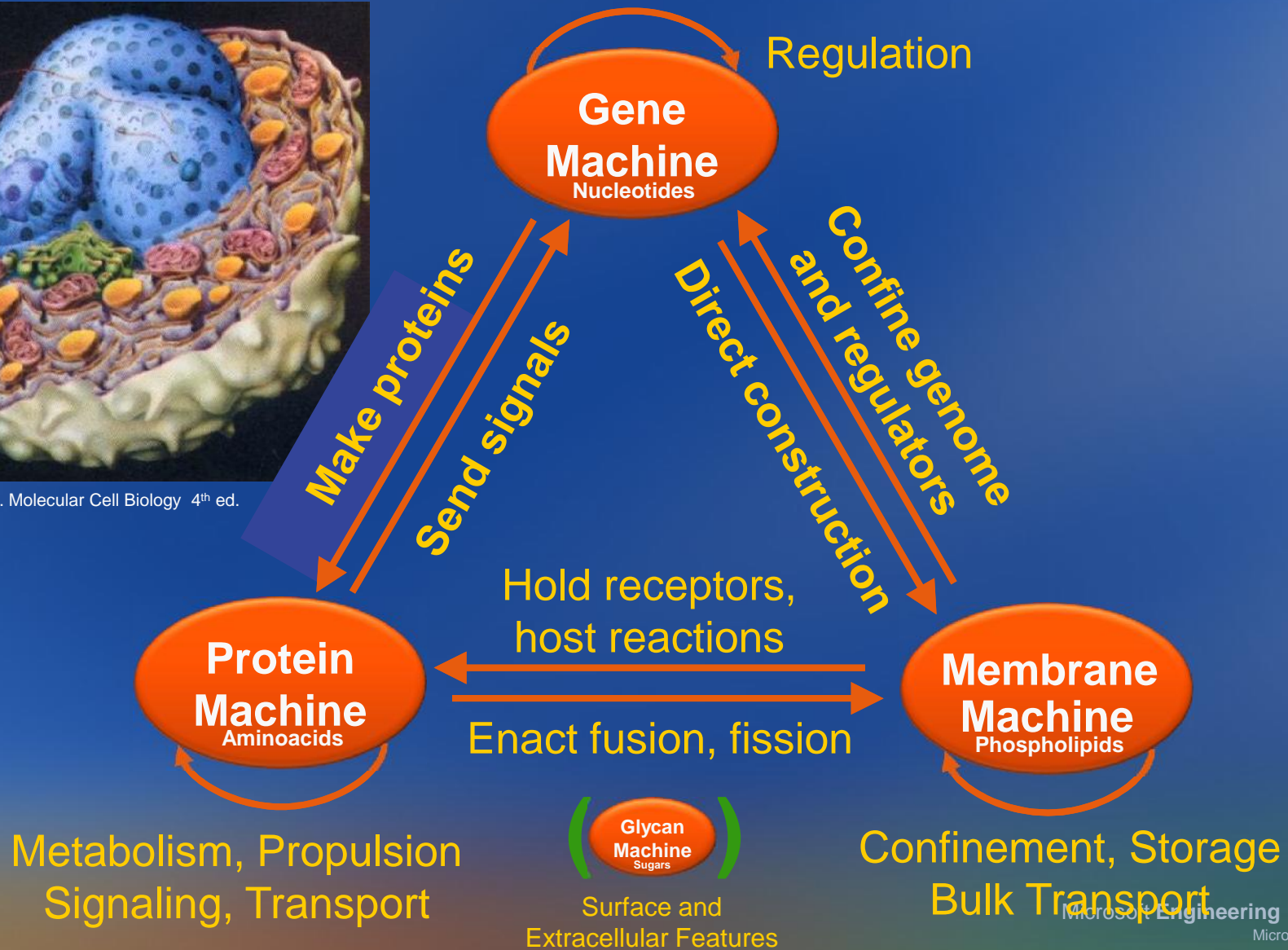
The Biological Argument

Biological systems are already
'molecularly programmed'

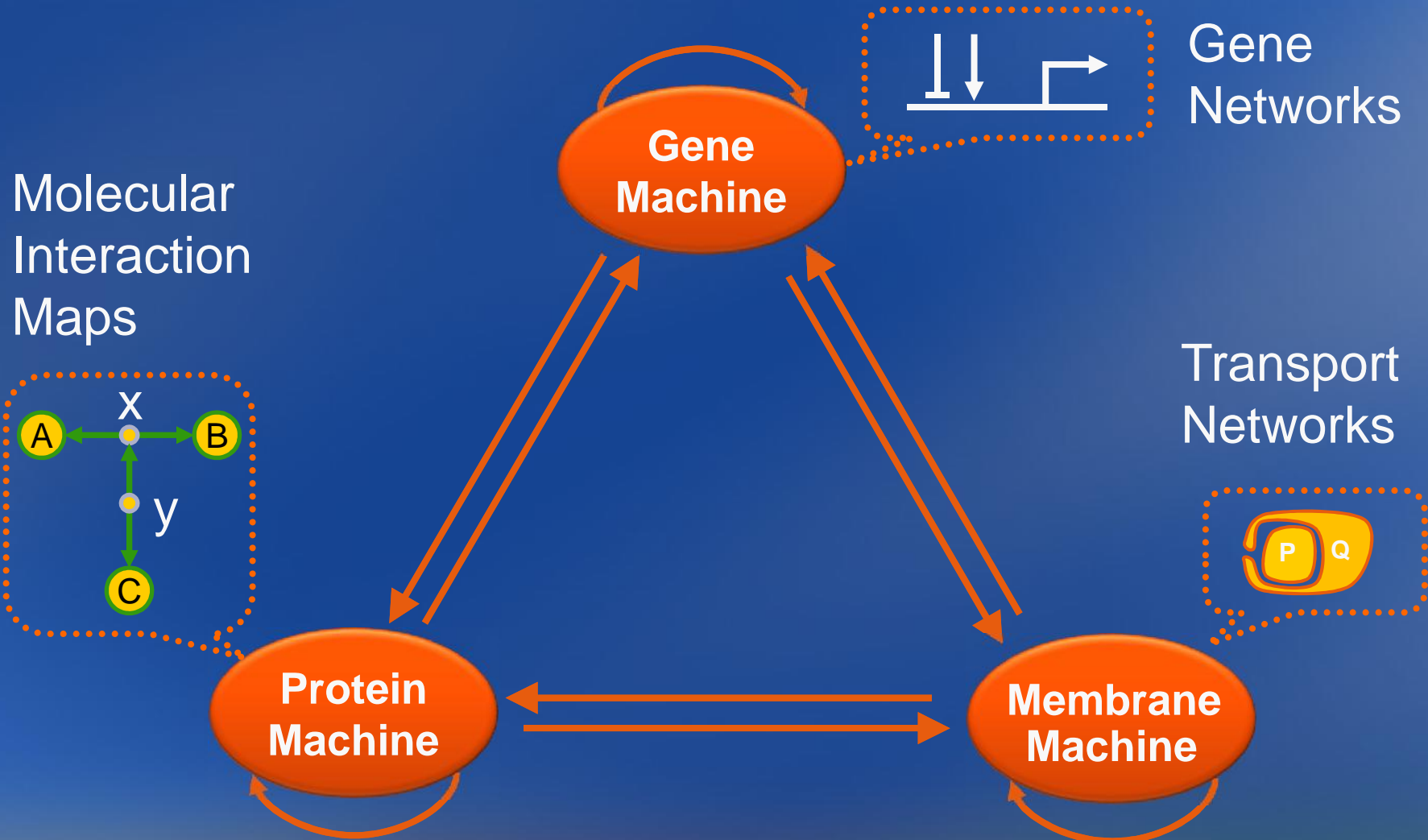
Abstract Machines of Biology



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



Languages



But ...

- Biology is programmable, but 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, unfortunately we cannot yet effectively design proteins
 - Transport networks are being looked at for programming microfluidic devices manipulating vesicles

Microsoft Engineering Excellence

Molecular Languages

... that we can deal with

Long-Term Action Plan

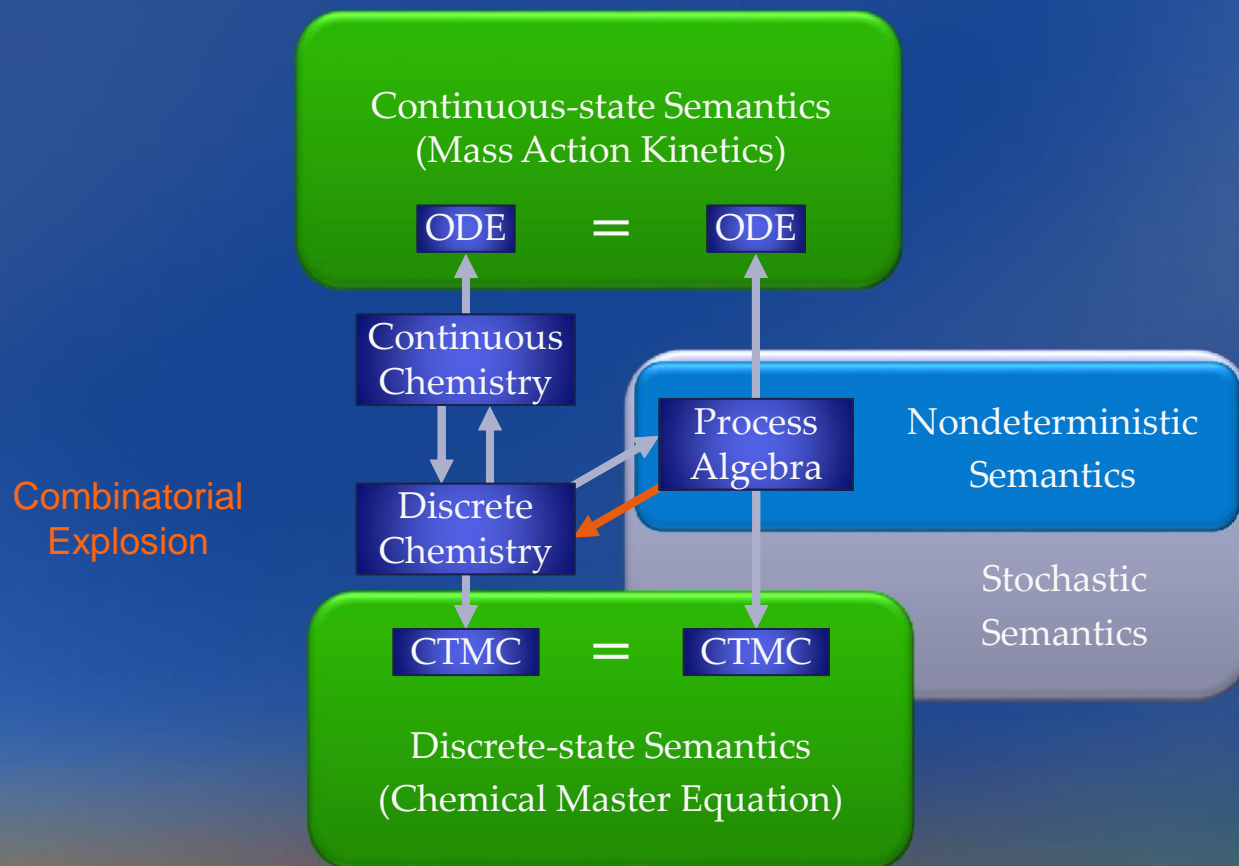
- Building a full pipeline
 - Mathematical Foundations [~ concurrency theory in the 80's]
 - Programming Languages [~ software engineering in the 70's]
 - Analytical Methods and Tools [~ formal methods in the 90's]
 - Device Architecture and Manufacturing [~ electronics in the 60's]
- Molecular Compilers
 - Front end: theory-backed analyzable programming languages
 - Back end: executable molecular systems
 - Requiring techniques for mastering complexity and analyzing system performance/safety ... mostly familiar to us
- No “alien technology”!
 - Do not use components (from Biology) we do not understand how to build ourselves. [David Soloveichik]

Our Assembly 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
 - $d[A]/dt = -r[A][B] \dots$ (the behavior)
- Rich analytical techniques based on Calculus
- But prone to combinatorial explosion
 - E.g., due to the peculiarities of protein interactions

Chemistry as a Concurrent Language

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



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

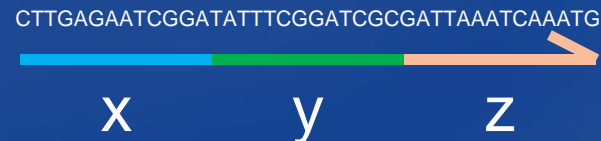
Building the cores of programmable molecular controllers

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
 - It is possible that other such materials will be developed

Domains

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



- That is, 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.
- Still somewhat of an open problem
 - A large literature
 - Can work in practice
 - Domain sequences often designed “by hand”

Short Domains



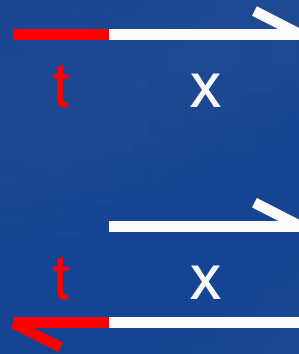
Reversible Hybridization

Long Domains



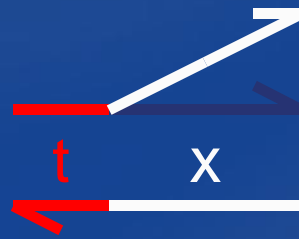
Irreversible Hybridization

Strand Displacement



“Toehold Mediated”

Strand Displacement



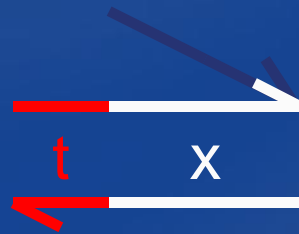
Toehold Binding

Strand Displacement



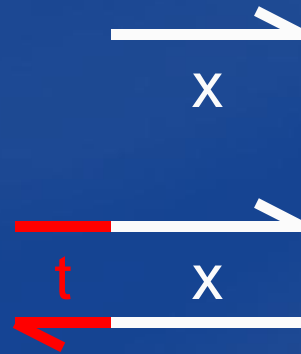
Branch Migration

Strand Displacement



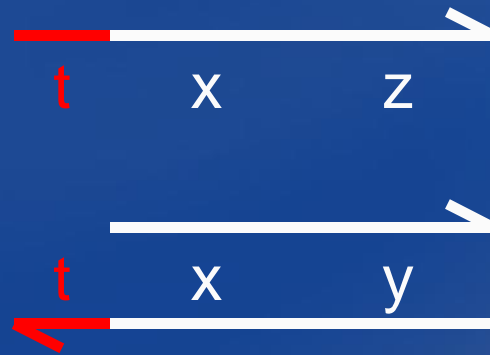
Displacement

Strand Displacement

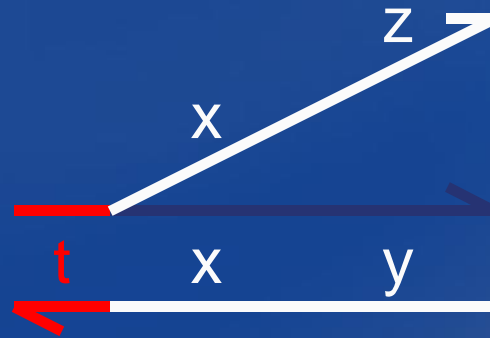


Irreversible release

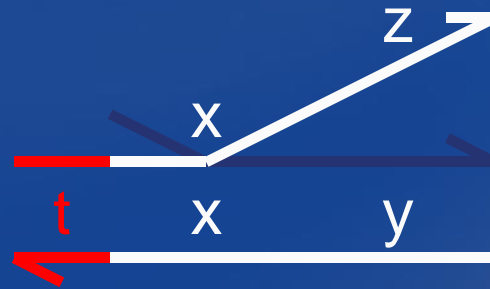
Bad Match



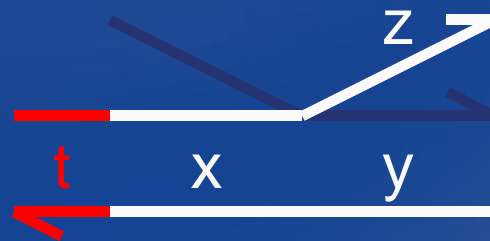
Bad Match



Bad Match



Bad Match



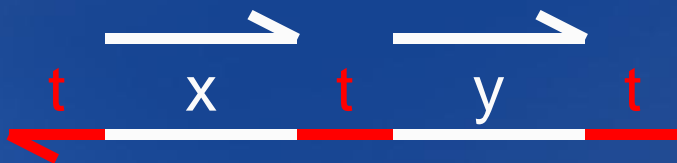
Cannot proceed
Hence will undo

Two-Domain Architecture

- Signals: 1 toehold + 1 recognition region



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



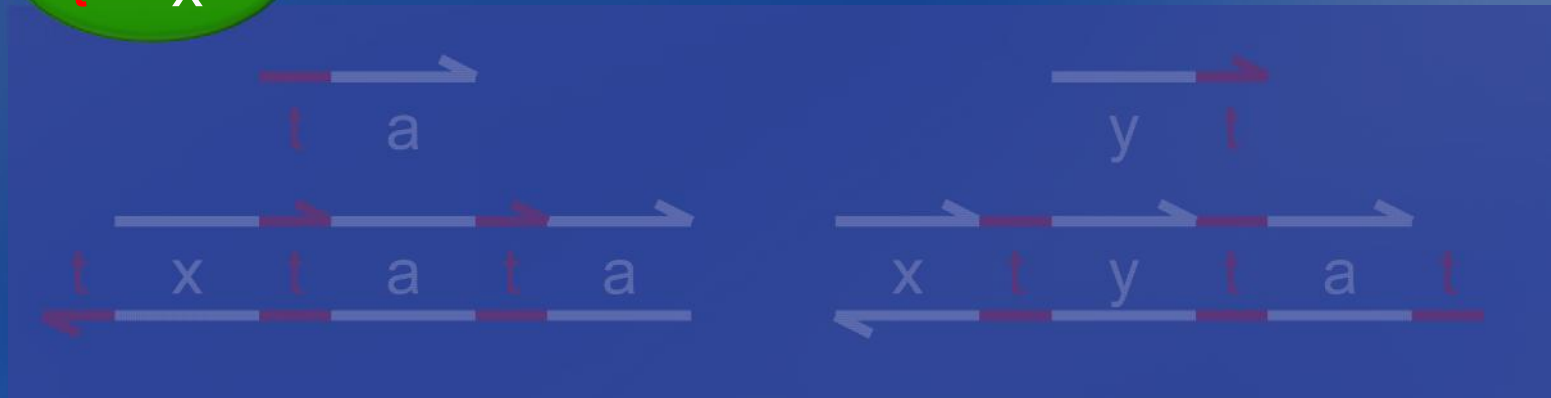
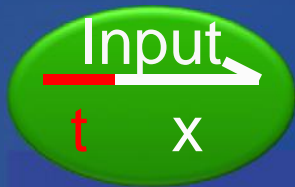
Garbage collection
“built into” the gates

Two-Domain DNA Strand Displacement

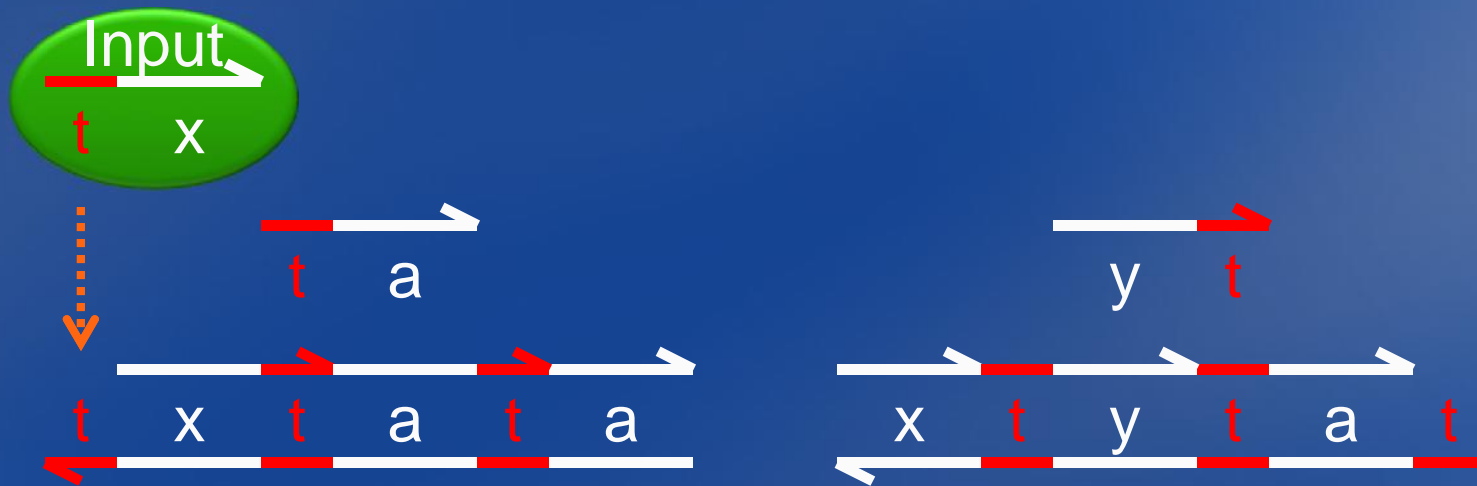
Luca Cardelli

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

Transducer $x \rightarrow y$



Transducer $x \rightarrow y$



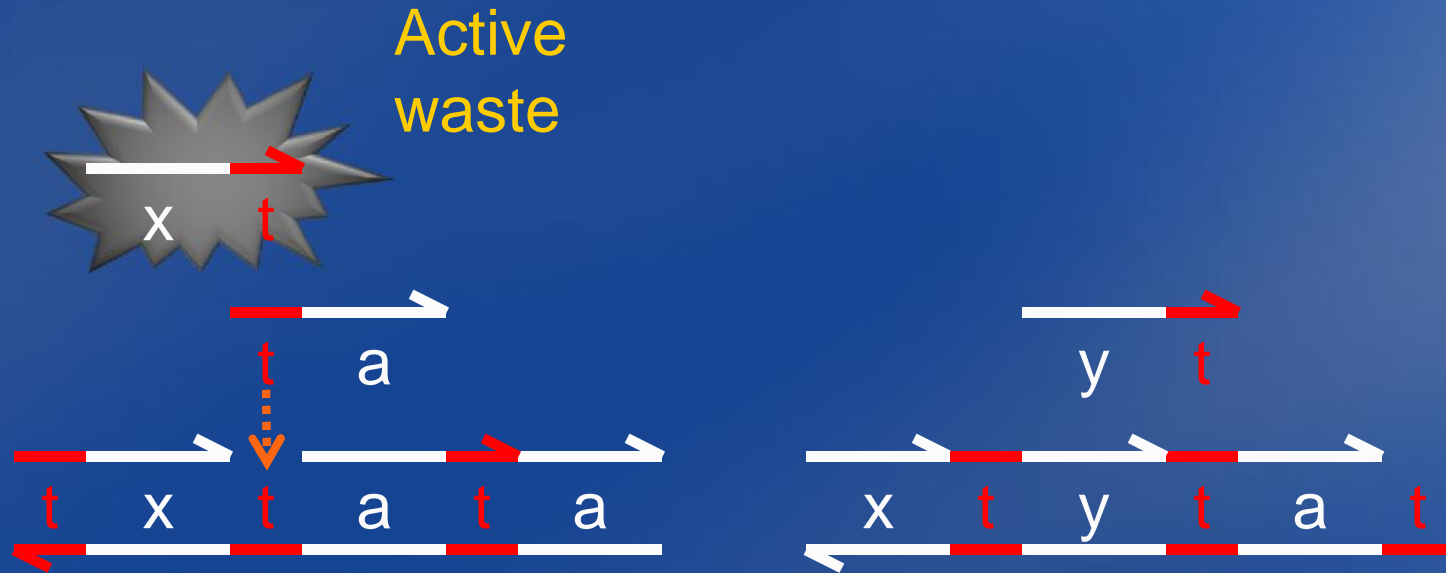
Built by self-assembly!

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

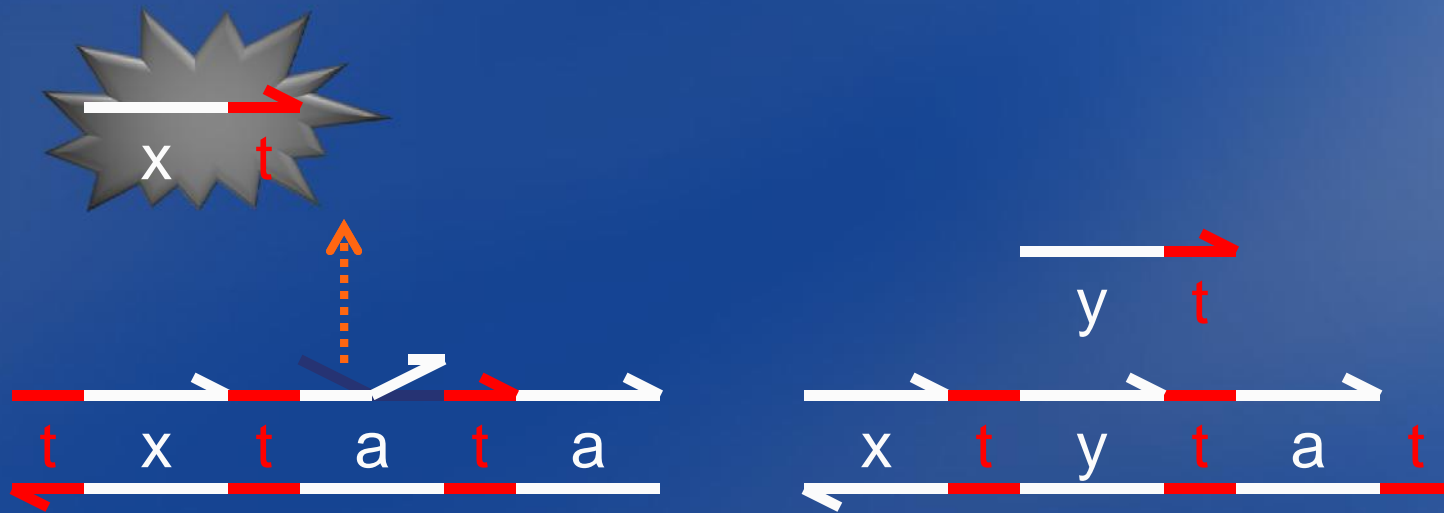
Transducer $x \rightarrow y$



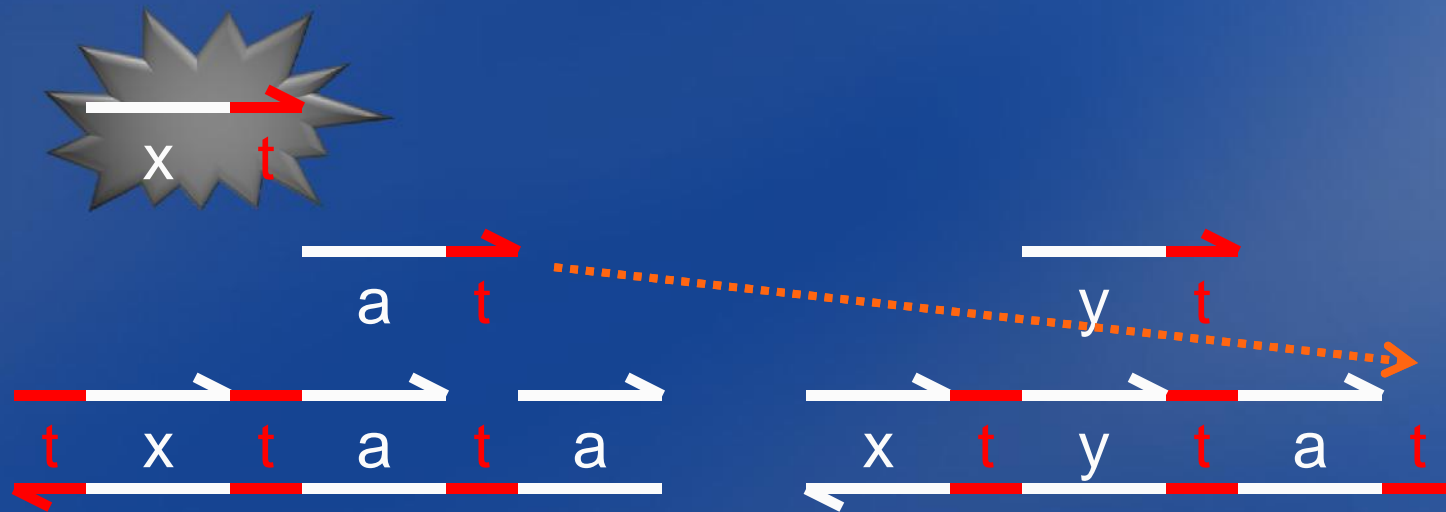
Transducer $x \rightarrow y$



Transducer $x \rightarrow y$



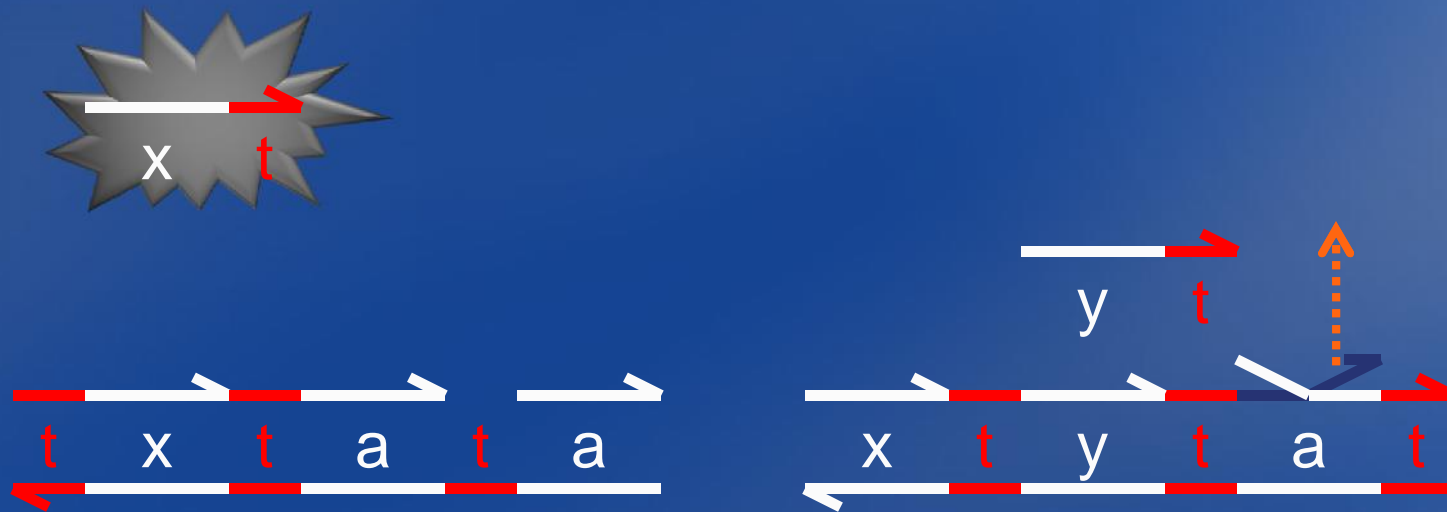
Transducer $x \rightarrow y$



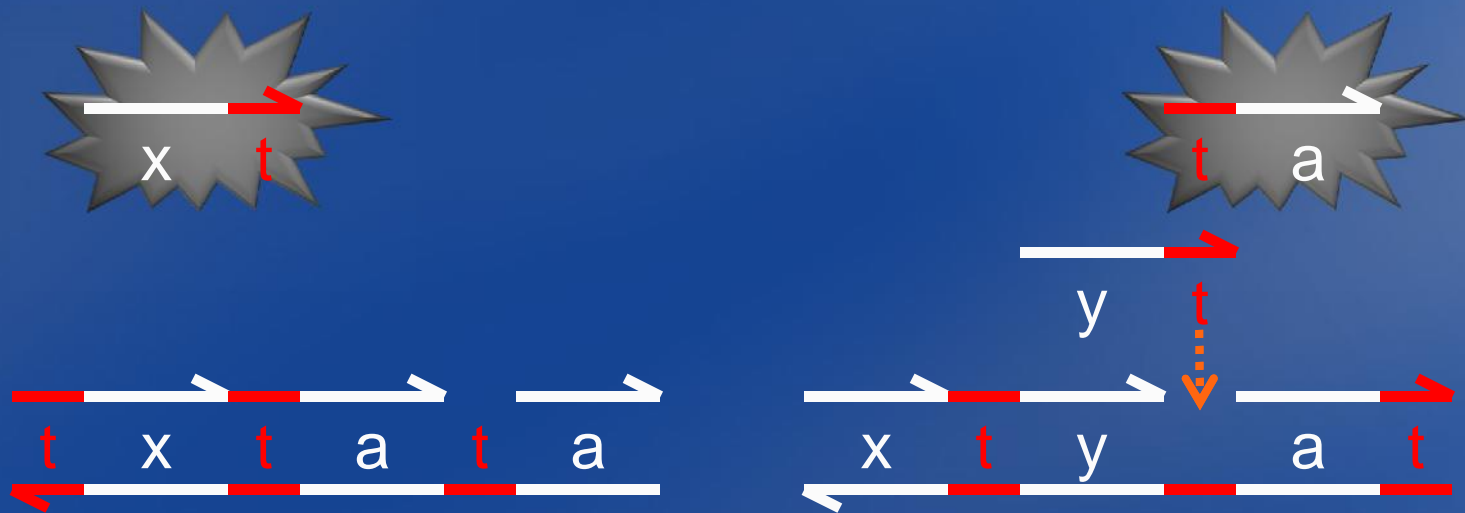
So far, a tx signal has produced an at cosignal.

But we want signals as output, not cosignals.

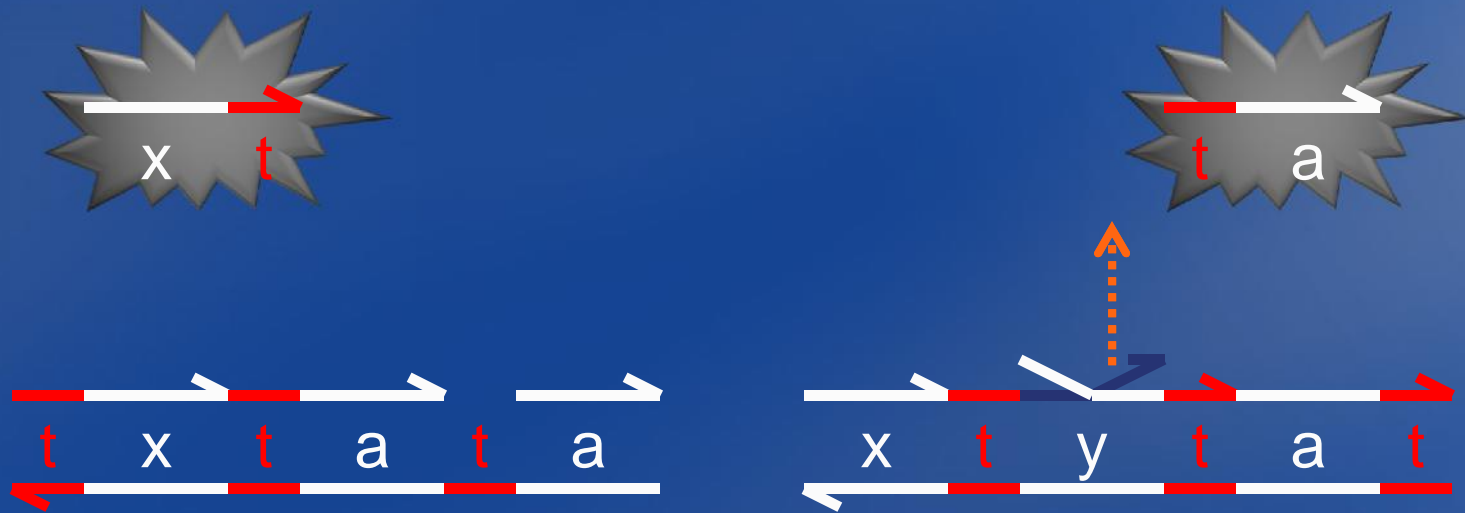
Transducer $x \rightarrow y$



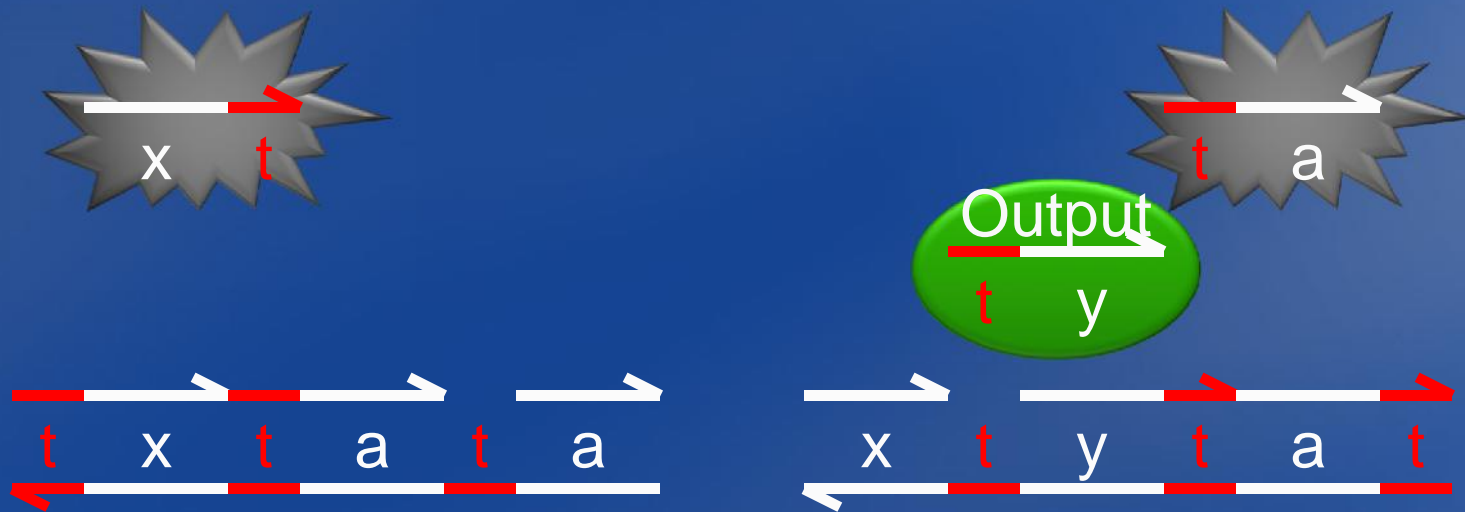
Transducer $x \rightarrow y$



Transducer $x \rightarrow y$



Transducer $x \rightarrow y$



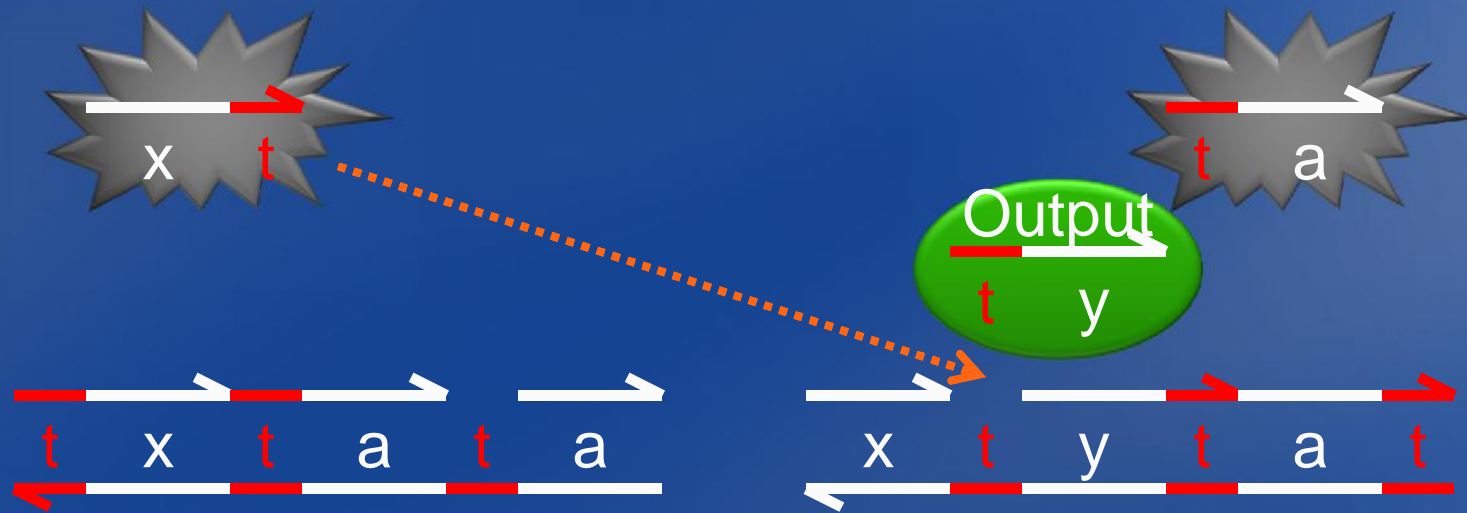
Here is our output *ty* signal.

But we are not done yet:

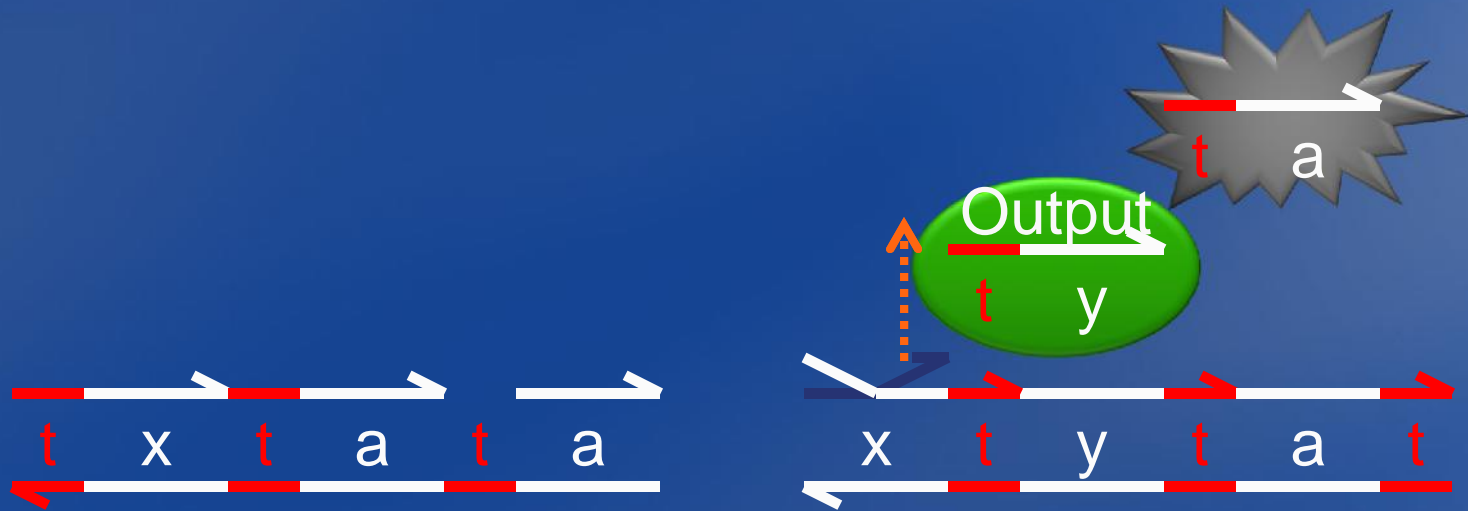
- 1) We need to make the output irreversible.
- 2) We need to remove the garbage.

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

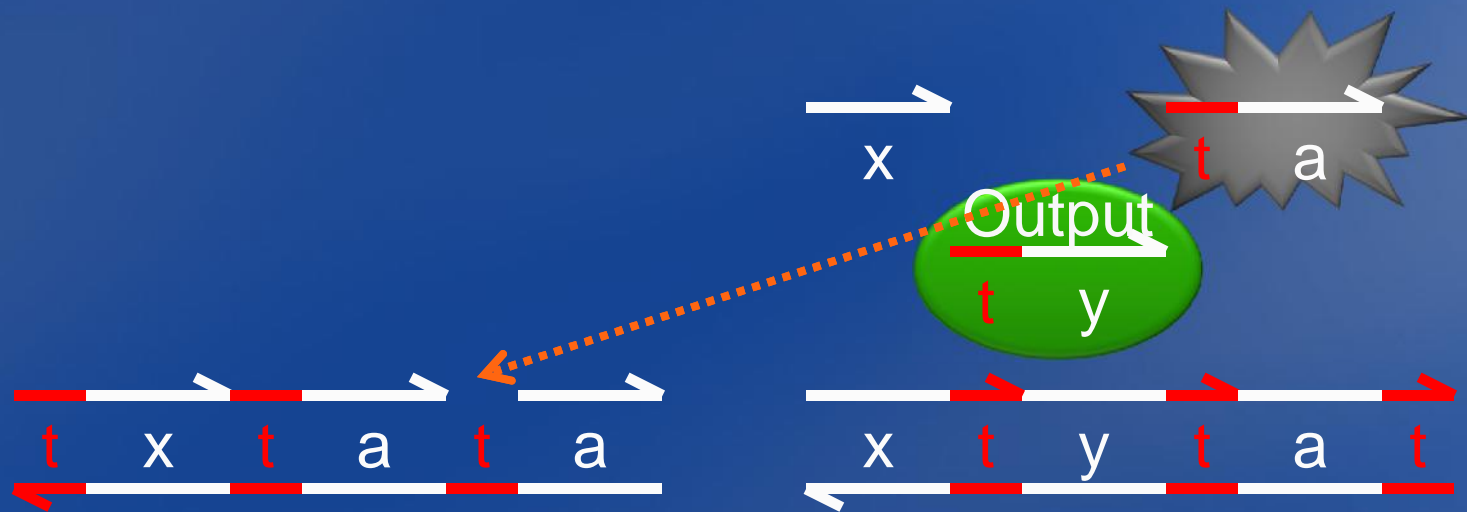
Transducer $x \rightarrow y$



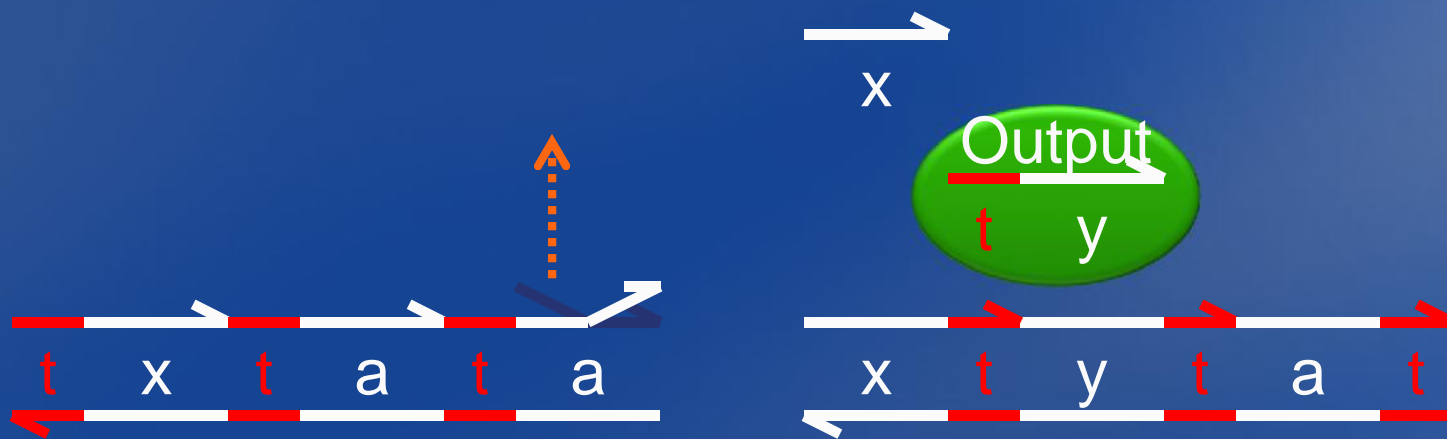
Transducer $x \rightarrow y$



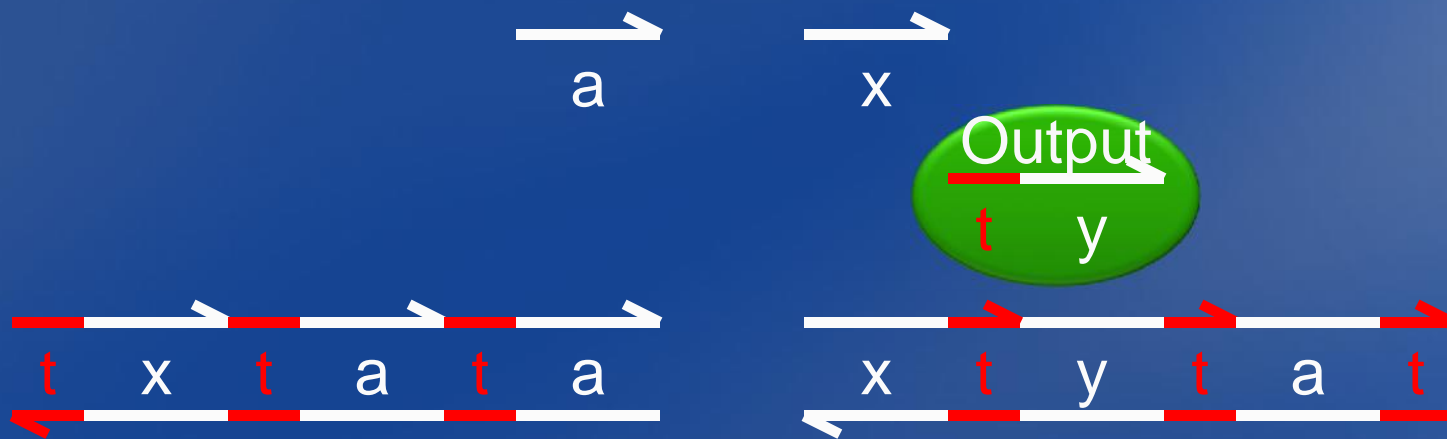
Transducer $x \rightarrow y$



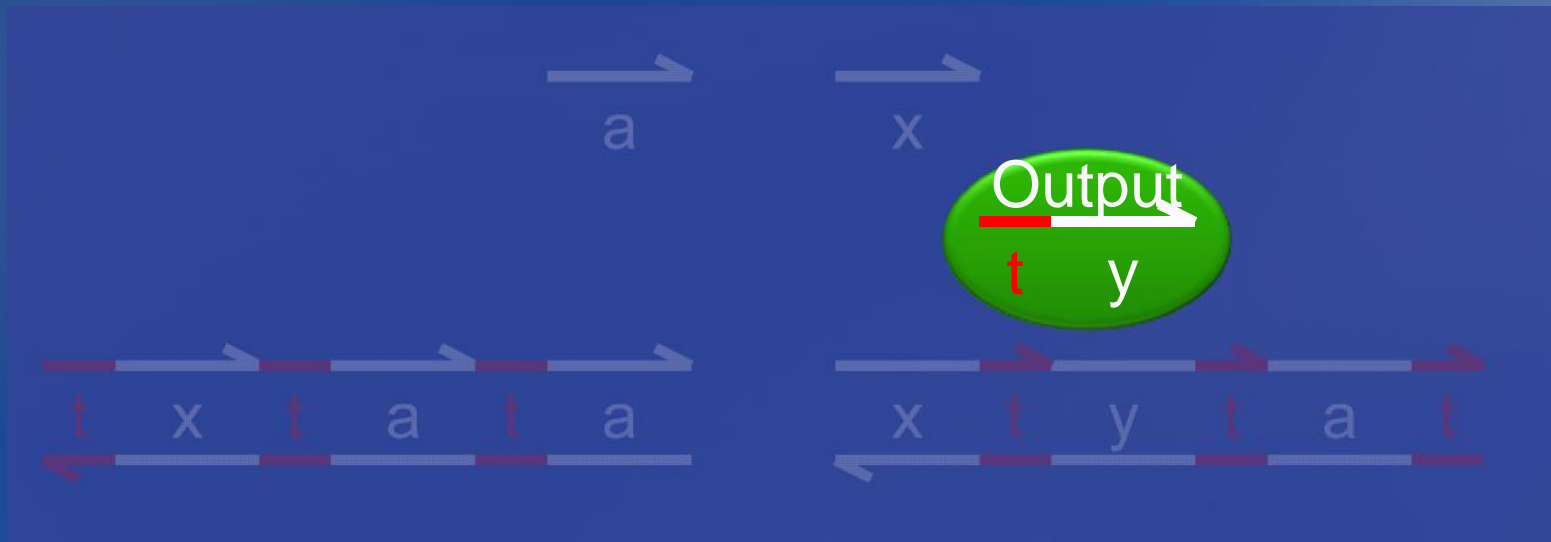
Transducer $x \rightarrow y$



Transducer $x \rightarrow y$



Transducer $x \rightarrow y$



Done.

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

Microsoft Engineering Excellence

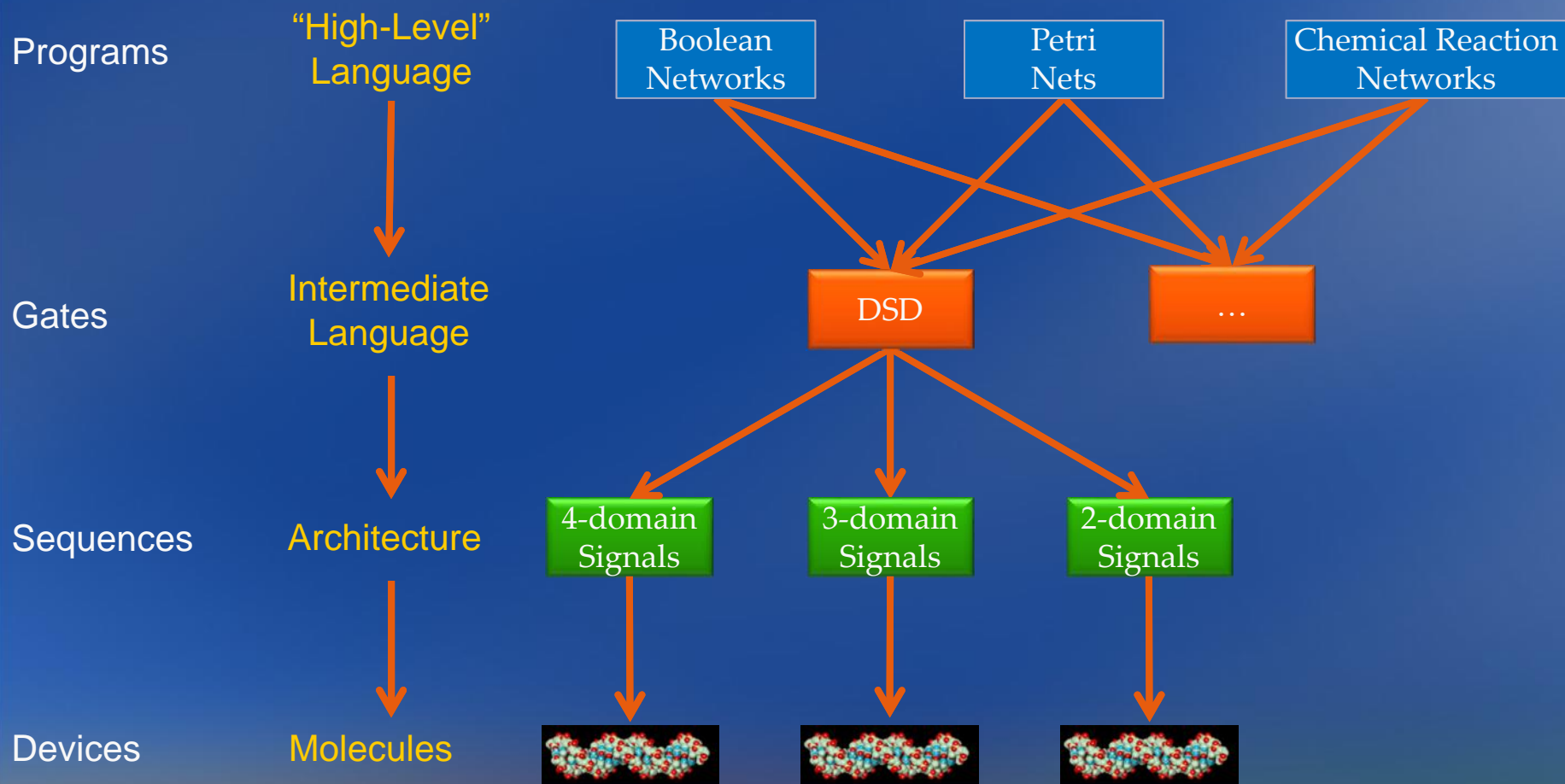
Tools and Techniques

A software pipeline for Molecular Programming

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...

Molecular Compilation



Visual DSD (DNA Strand Displacement)

- <http://research.microsoft.com/apps/pubs/default.aspx?id=157262>

Andrew Phillips
MSR Cambridge

The screenshot shows the Microsoft Research website interface. At the top, there is a search bar and navigation links for Videos, Projects, Publications, People, and Downloads. Below this is a secondary navigation bar with links for Home, Our Research, Connections, Careers, and Hub. The main content area displays the article title 'Visual DSD: a design and analysis tool for DNA strand displacement systems' by Matthew Lakin, Simon Youssef, Filippo Polo, Stephen Emmott, and Andrew Phillips, dated November 2011. The article summary states: 'SUMMARY: The Visual DSD (DNA Strand Displacement) tool allows rapid prototyping and analysis of computational devices implemented using DNA strand displacement, in a convenient web-based graphical interface. It is an implementation of the DSD programming language and compiler described by Lakin et al. (2011) with additional features such as support for polymers of unbounded length. It also supports stochastic and deterministic simulation, construction of continuous-time Markov chains and various export formats which allow models to be analysed using third-party tools.' Below the summary, it lists 'In: Bioinformatics' and 'Publisher: Oxford University Press'. A 'Details' section provides metadata: Type: Article, URL: <http://dx.doi.org/10.1093/bioinformatics/btr543>, Pages: 3211-3213, Volume: 27, Number: 22, and Institution: Biological Computation Group, Microsoft Research, Cambridge CB3 0FB, UK. On the right side, there are sections for 'Related Projects' (A programming language for composable DNA circuits), 'Related People' (Andrew Phillips, Stephen Emmott), 'Related Groups' (Biological Computation, Computational Science Laboratory), 'Related Labs' (Microsoft Research Cambridge), and 'Related Research Areas' (Computational sciences). The footer contains contact information, terms, and the Microsoft xcellence logo.

A Development Environment for DNA Gates

The screenshot displays a software development environment for DNA gates. The interface is divided into several panels:

- Code Panel (Left):** Contains a code editor with the following text:

```
directive duration 50000.0 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) )
```
- Compilation Panel (Top Right):** Includes tabs for 'Compilation', 'Simulation', and 'Analysis'. Below these are sub-tabs for 'Species', 'Reactions', 'Graph', 'Text', 'Domains', and 'SBML'. The 'Reactions' sub-tab is currently selected.
- Reaction Network Diagram (Right):** Shows a series of chemical reactions between DNA species. The species are represented as horizontal lines with segments labeled 'x', 'y', 'a', and 't'. The reactions are shown with double-headed arrows (reversible) or single-headed arrows (irreversible). The reactions illustrate the assembly and disassembly of DNA structures, such as the formation of a double-stranded complex from two single strands and the subsequent cleavage of that complex.

The status bar at the bottom indicates 'Ready', 'Ln 7', 'Col 15', 'Ch 15', 'INS', and '100%' zoom.

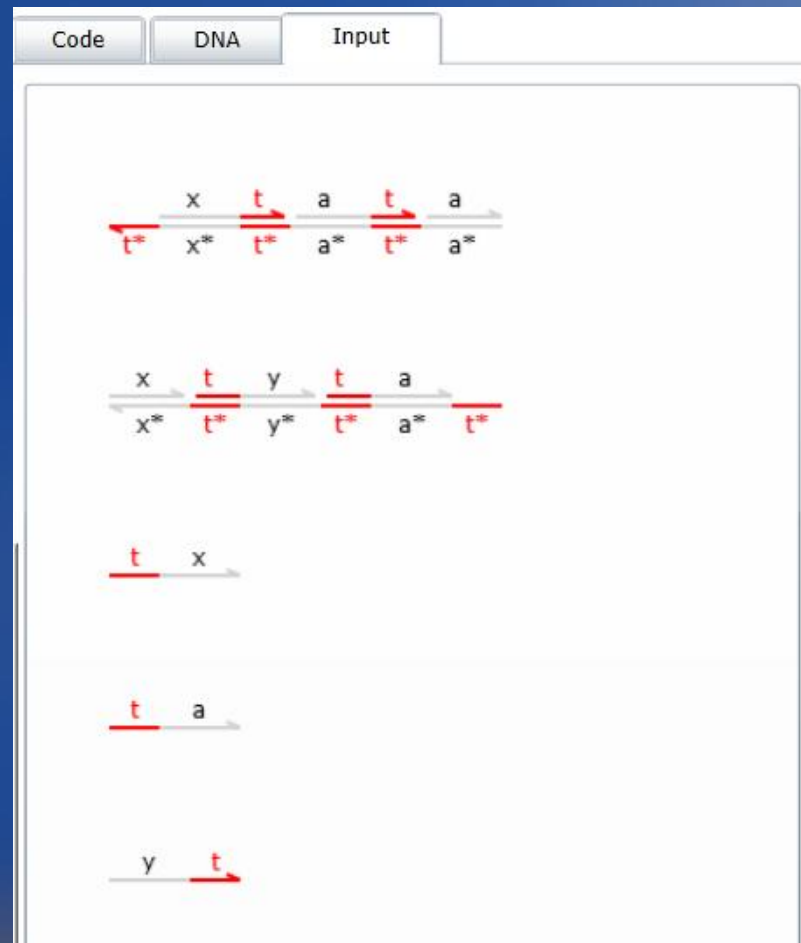
A Language for DNA Structures

- Describe the initial structures

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

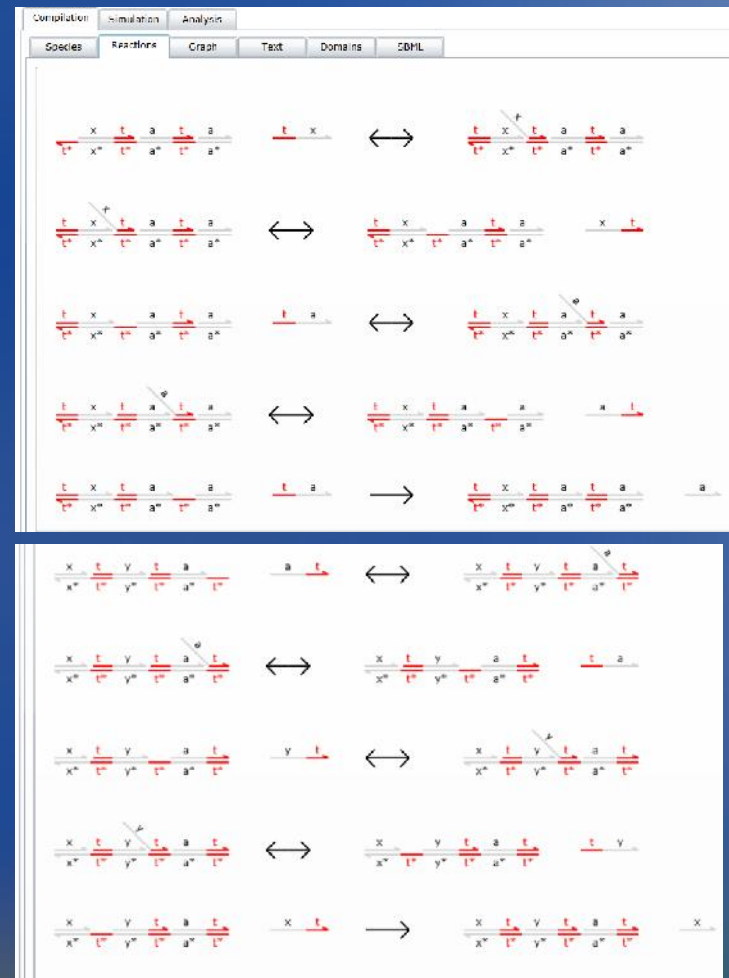
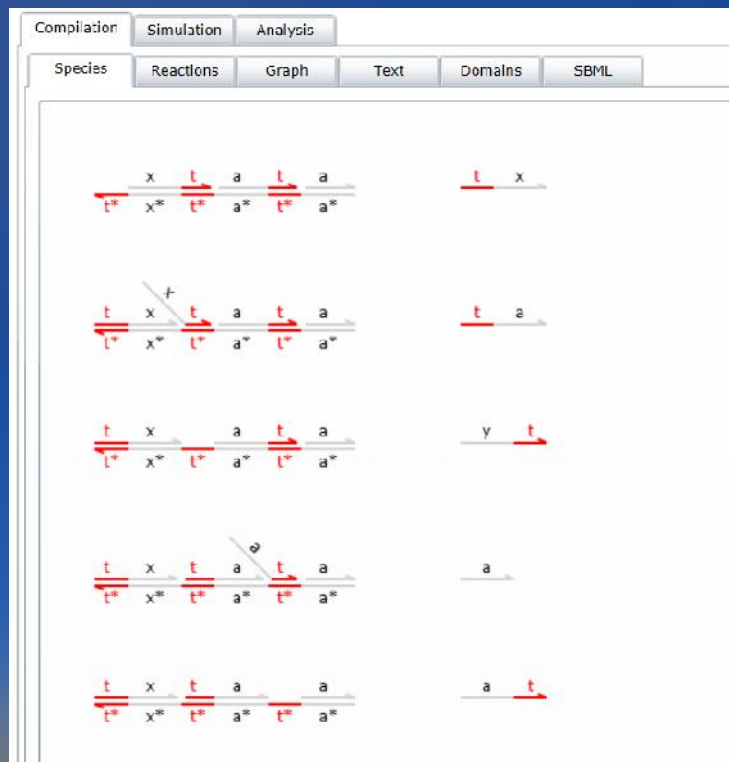
def T(N,x,y) =
  new a
  ( N * <t^ a>
  | N * <y t^>
  | N * t^*: [x t^]: [a t^]: [a] (* Input gate *)
  | N * [x]: [t^ y]: [t^ a]: t^* (* Output gate *)
  )

( <t^ x> | T(1,x,y) )
```

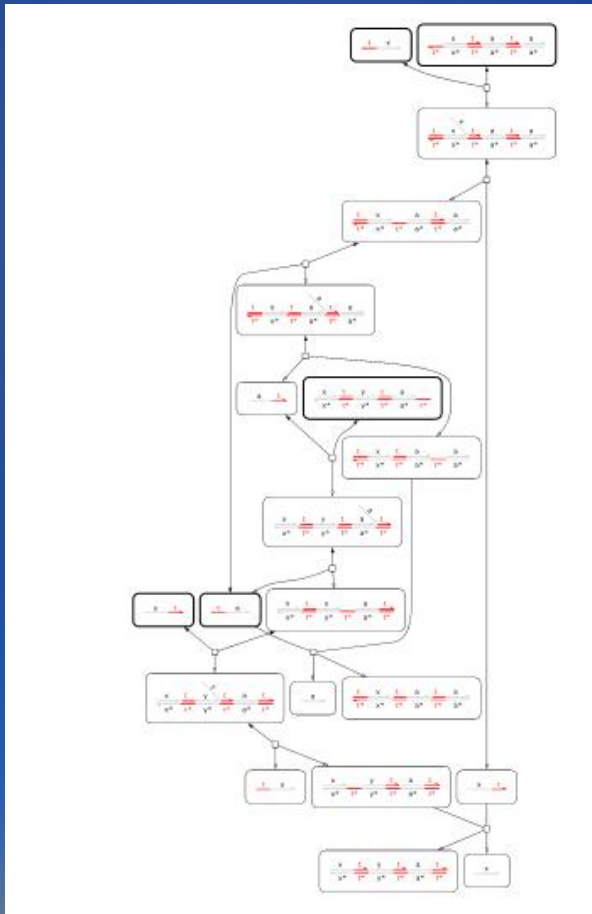


Compute Species and Reactions

- Recursively computed from the initial structures



Reaction Graph and Export



Compilation Simulation Analysis

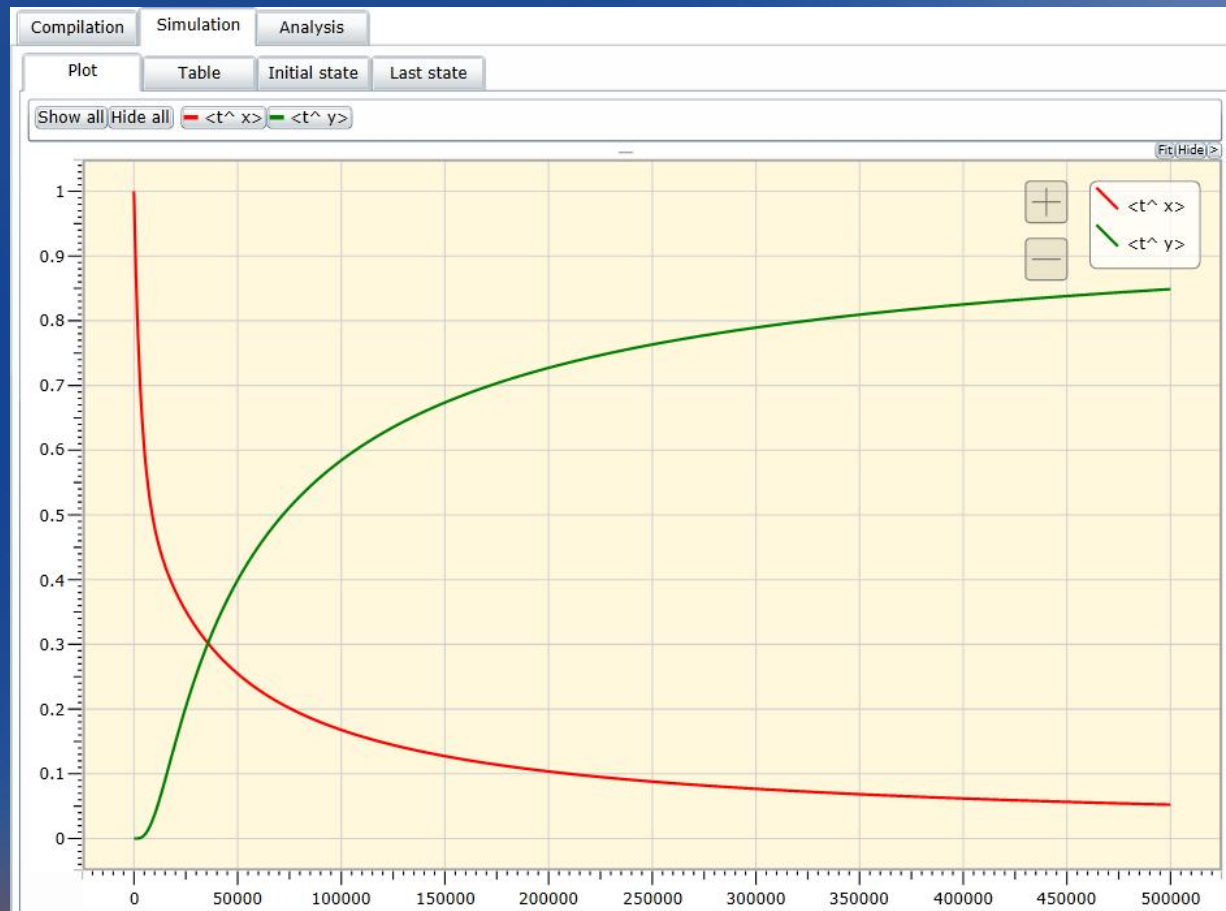
Species Reactions Graph Text Domains SBML

Save as XML

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level2/version1" level="2" version="1">
  <model>
    <listOfCompartments>
      <compartment id="c" size="1"/>
    </listOfCompartments>
    <listOfSpecies>
      <species id="s_id0" name="&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"/>
        </listOfReactants>
      </reaction>
    </listOfReactions>
  </model>
</sbml>
```

Simulation

- Stochastic
- Deterministic
- “JIT”



Analysis

Compilation Simulation Analysis
Graph Text PRISM Visualise

INITIAL STATE:

$\frac{t}{t} \frac{a}{a} (1)$

$\frac{t}{t} \frac{x}{x} (1)$

$\frac{y}{y} \frac{t}{t} (1)$

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

$\frac{x}{t^*} \frac{t}{x^*} \frac{a}{t^*} \frac{t}{a^*} \frac{a}{t^*} (1)$

TERMINAL STATE:

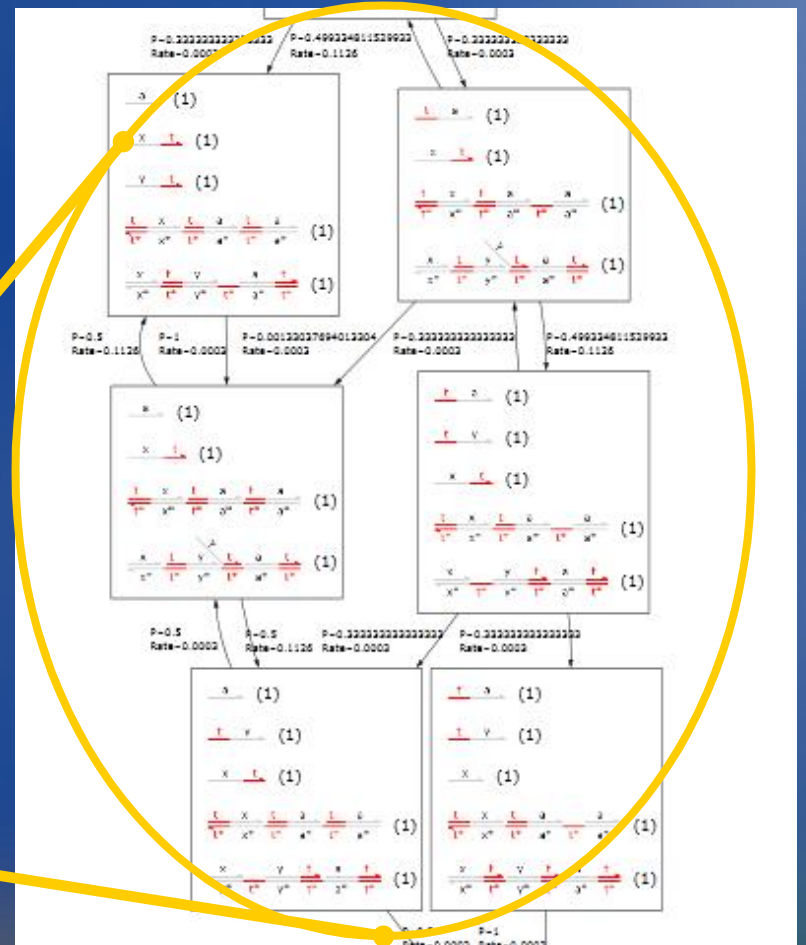
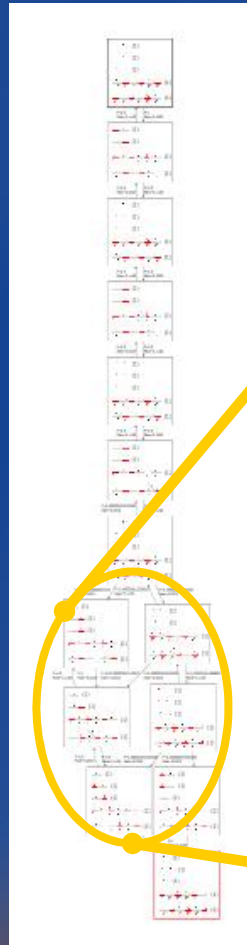
$\frac{a}{a} (1)$

$\frac{t}{t} \frac{y}{y} (1)$

$\frac{x}{x} (1)$

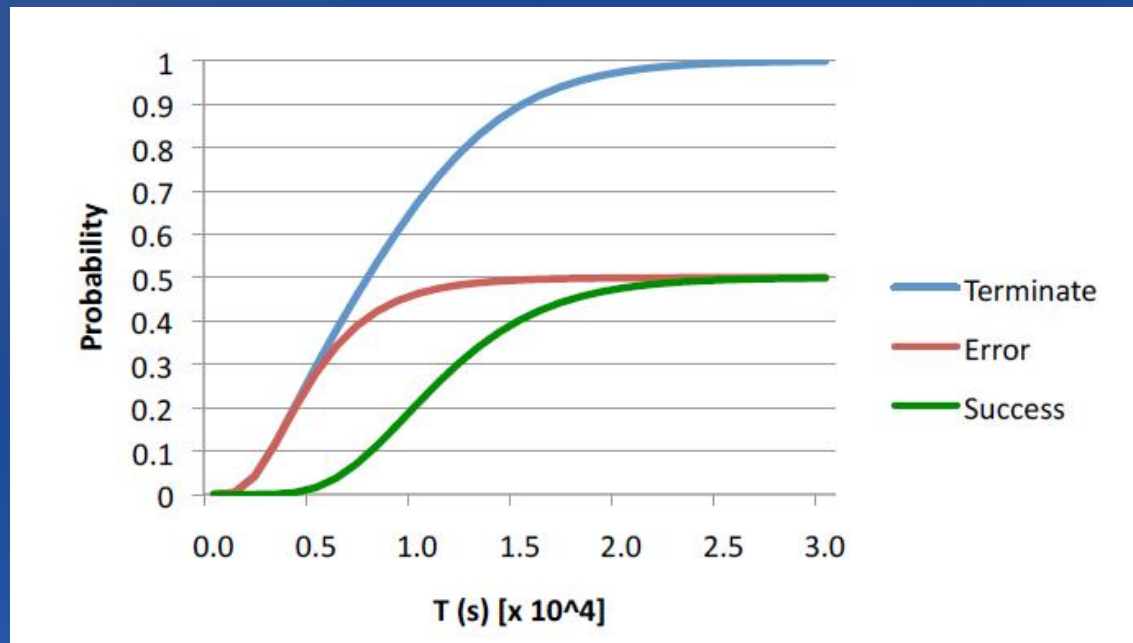
$\frac{t}{t^*} \frac{x}{x^*} \frac{t}{t^*} \frac{a}{a^*} \frac{t}{t^*} \frac{a}{a^*} (1)$

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



Modelchecking

- Export to PRISM probabilistic modelchecker



Design and Analysis of DNA
Strand Displacement Devices
using Probabilistic Model
Checking

Matthew R. Lakin ^{*†} David Parker ^{‡§}

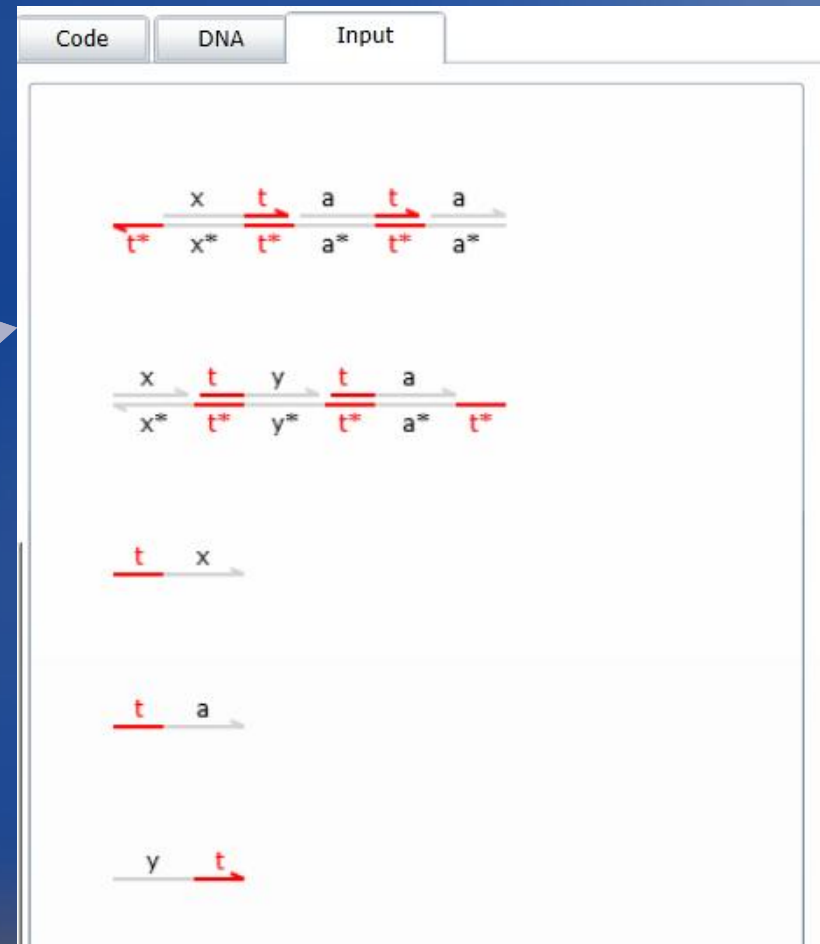
Luca Cardelli^{*} Marta Kwiatkowska[‡]

Andrew Phillips^{*§}

Tool Output: Domain Structures

- Abstract structures
 - (no DNA sequences)

“Ok, I want to run this for real”

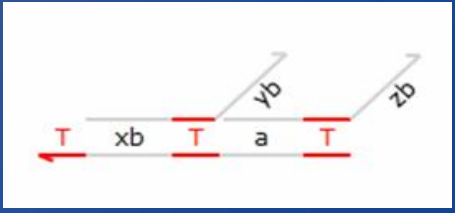


From Structures to Sequences



www.nupack.org

DSD Structure → “Dot-Paren” representation



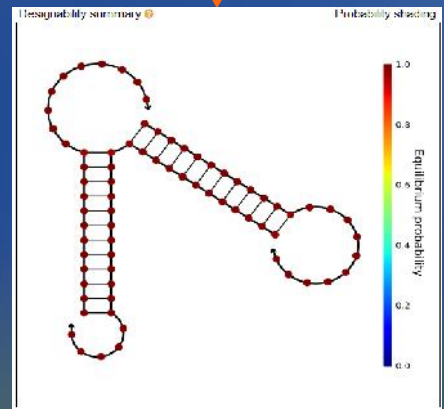
Nucleic acid type: RNA DNA Temperature: °C Number of designs:

Target structure:

Output Sequences

Ensemble	Normalized	GC content:	Sequence
dctct (nt)	ensemble	(%)	
	defect (%)		
0.2	0.3	57.5	GCUGCGAUCGCCAAAGAAC AA+GGGAUCAAGCCOCUCUU UUUCC+GGGCUUGAUCCGGG GUAUCGCGCUGCGC

Thermodynamic Synthesis



“Ok, where do I buy these”



DNA Synthesis

dna synthesis × Search

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

▶ **Custom DNA Synthesis** Ads
www.Biomatik.com 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

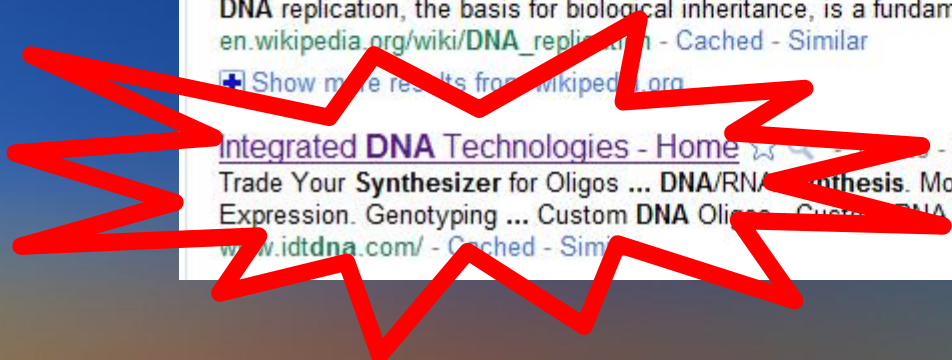
[Gene **Synthesis** \\$0.35/bp](#)
www.epochlifescience.com Dependable Service @ Low Price: Come on Down and Save Your Budgets!

[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 biological inheritance, is a fundamental ...
en.wikipedia.org/wiki/DNA_replication - Cached - Similar

[Show more results from wikipedia.org](#)

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



From Sequences to Molecules

- Copy&Paste from nupack

XX IDT
INTEGRATED DNA
TECHNOLOGIES

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

[Log In]
Spain
0 Items € 0,00

Home Products Order Support Services SciTools Search Go

Order Oligos

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

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

Scale: 25 nmole DNA oligo Purification: Standard
Sequence Name: 5'-ACT GCA CCA TAA GCA ACT TTT
Notes: Enter your notes here. Please do not enter modifications

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

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

Customized Labels (more detail)
Stock IDT Label FREE

Molecules by FedEx



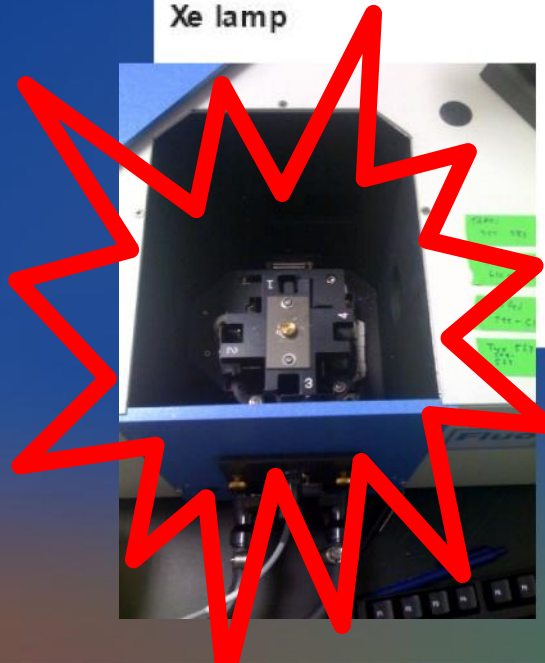
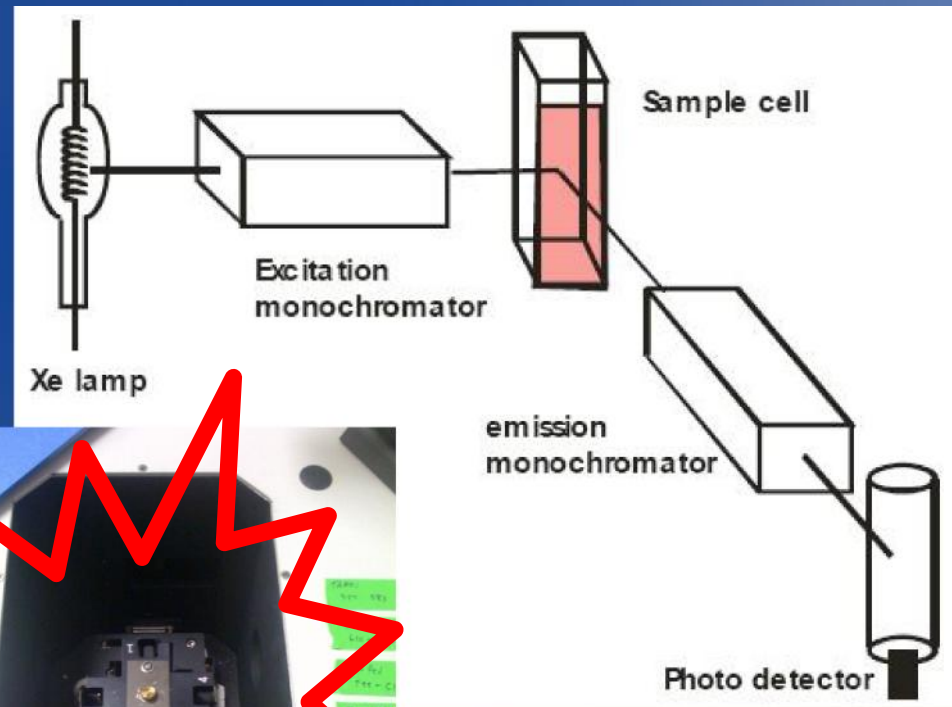
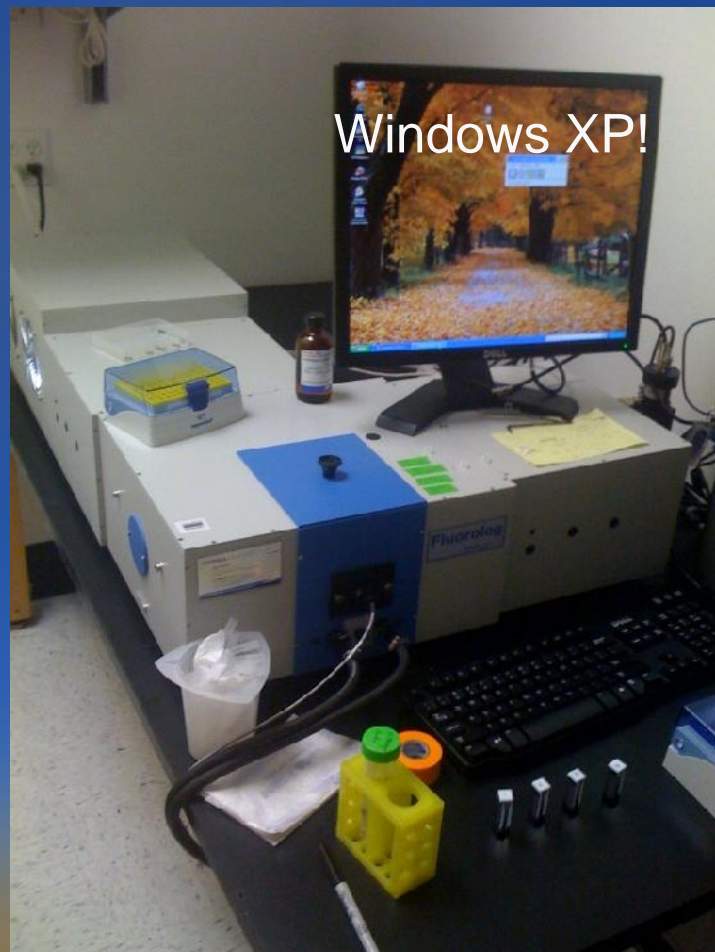
“Ok, how do I run these?”

Add Water

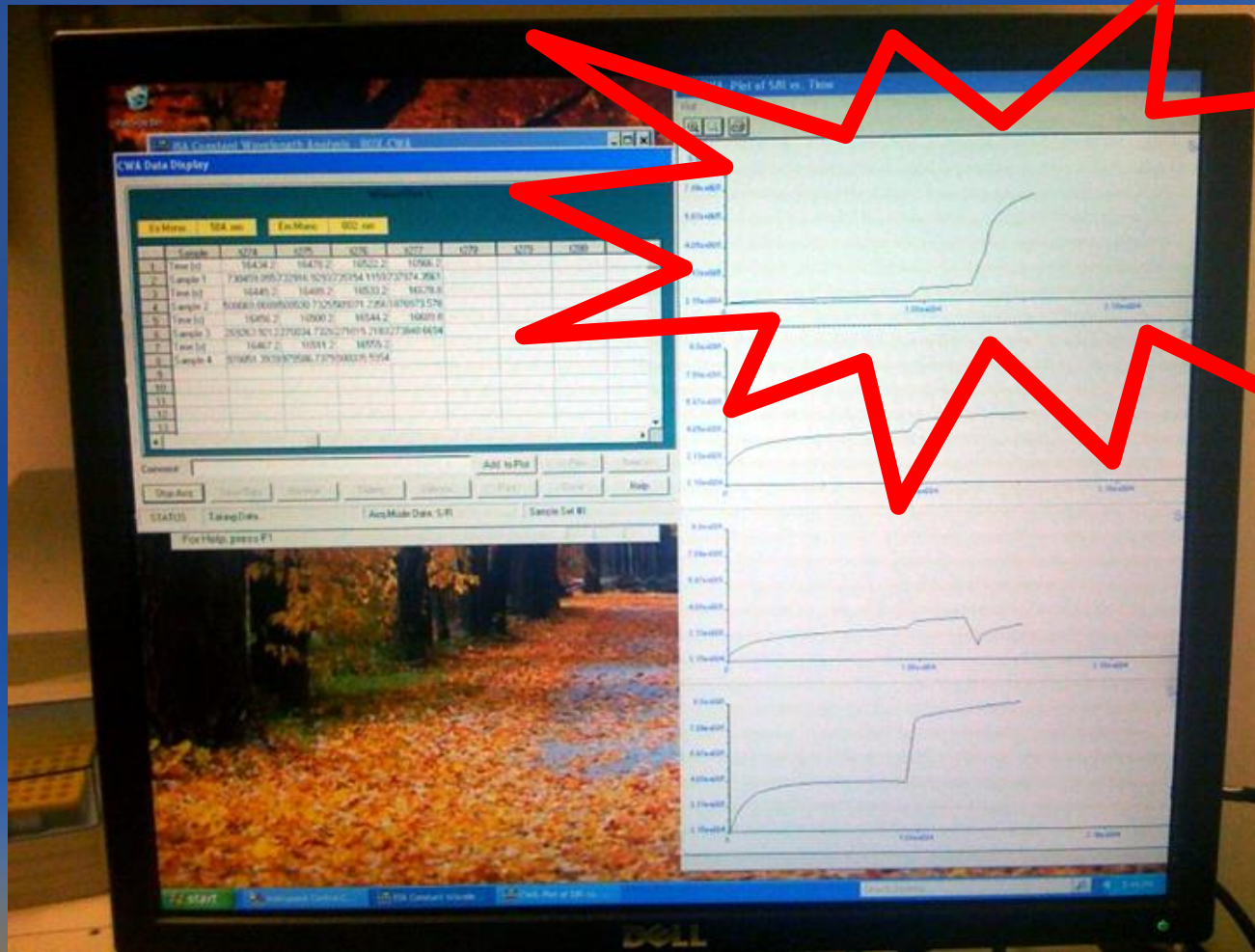


Execute (finally!)

- Fluorescence is your one-bit 'print' statement

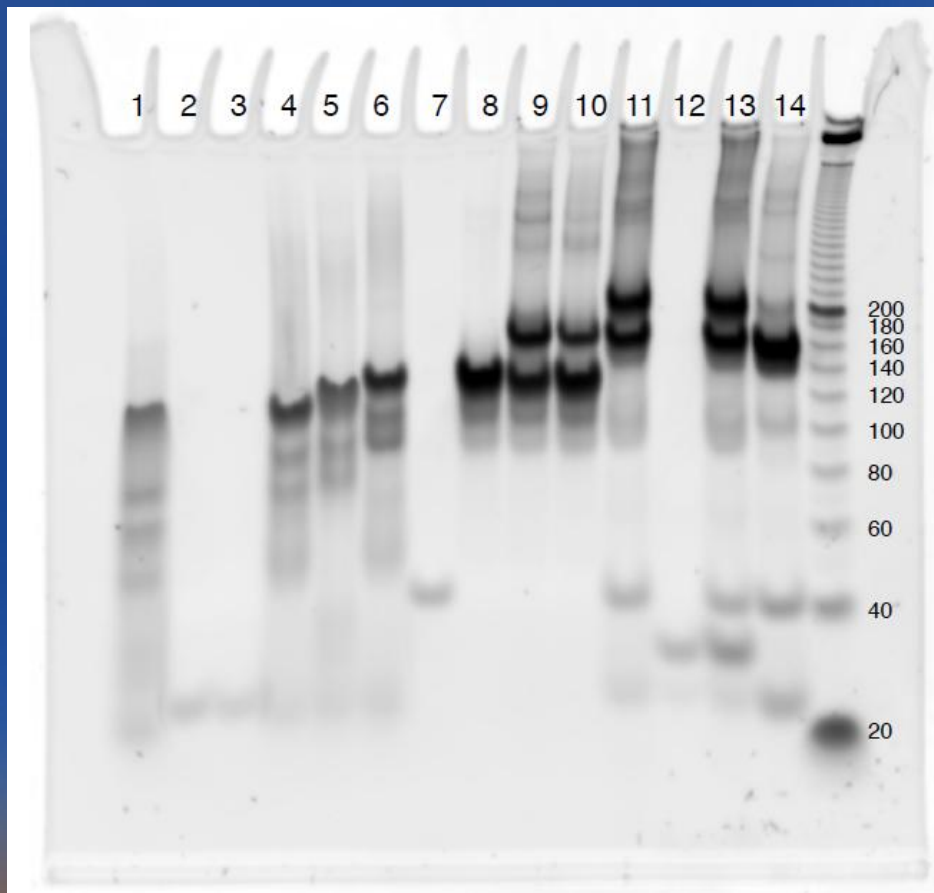


Output



Debugging

- A core dump



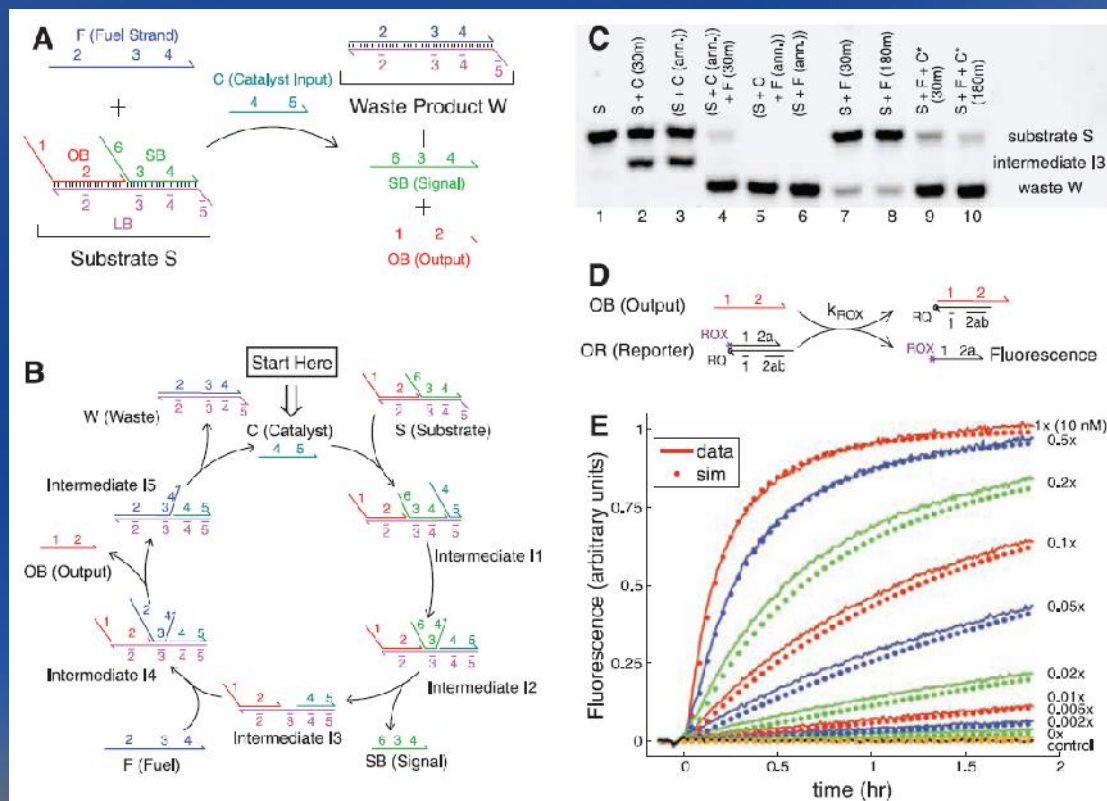
Delivering!

Engineering Entropy-Driven Reactions and Networks Catalyzed by DNA

David Yu Zhang, *et al.*

Science **318**, 1121 (2007);

DOI: 10.1126/science.1148532



Microsoft Engineering Excellence

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

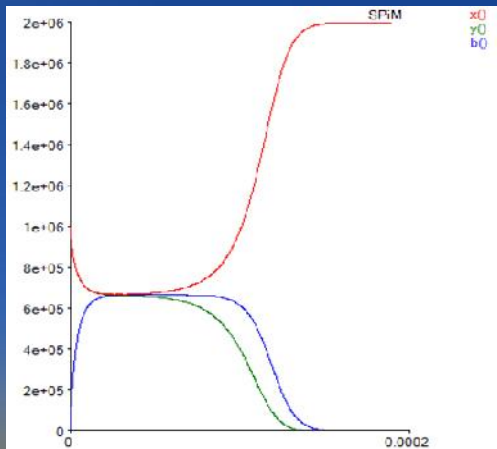
[Angluin et al., Distributed Computing, 2007]
- Could be used to restore a signal to full strength

- A chemical implementation



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 $\omega(\sqrt{n \log n})$
 - Hence the agreement state is stable

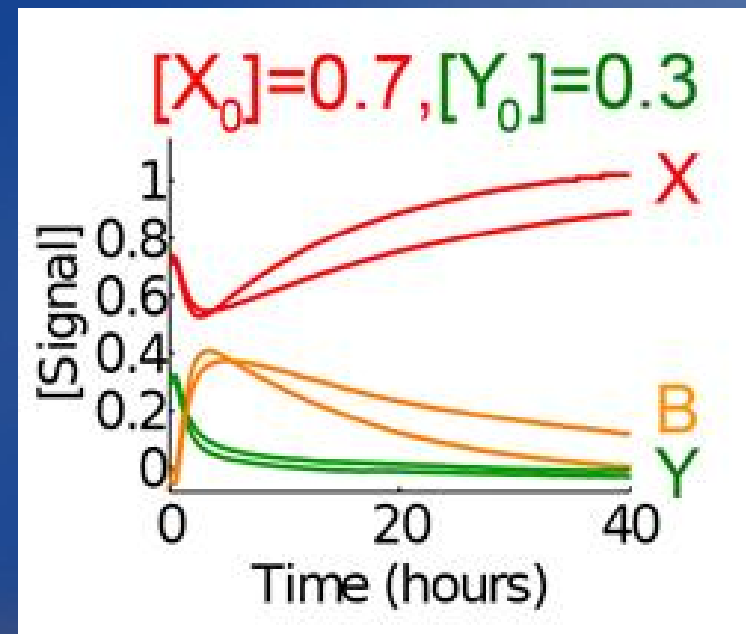
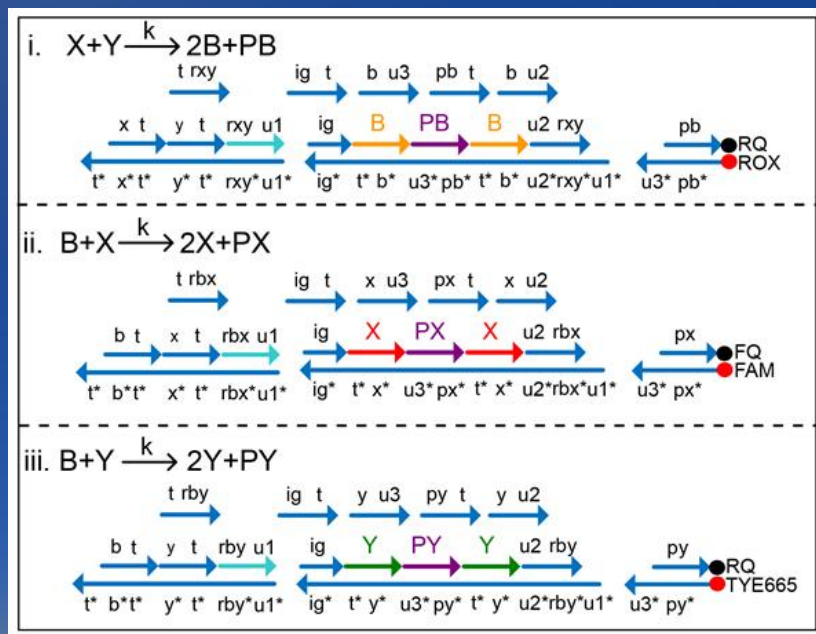


Stochastic simulation of worst-case,
with initially $\#X = \#Y$

DNA Implementation, at U.W.

● A DNA Realization of Chemical Reaction

Networks [Yuan-Jyue Chen, Neil Dalchau, Niranjan Srinivas, Andrew Phillips, Luca Cardelli, David Soloveichik and Georg Seelig]

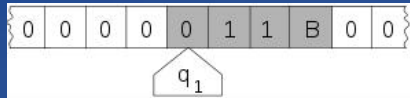


Microsoft Engineering Excellence

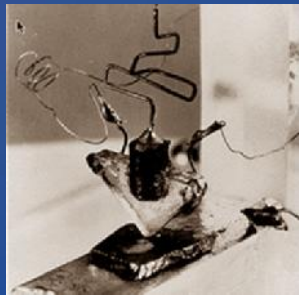
Final Remarks

A Brief History of DNA

Turing Machine, 1936



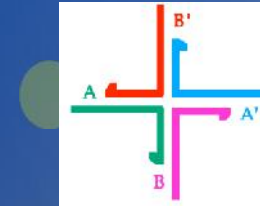
Transistor, 1947



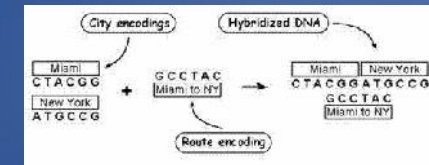
DNA, -3,800,000,000



Structural DNA
Nanotech, 1982



DNA Algorithm, 1994



~~Digital Computers~~
Computer programming

Software
systematic manipulation of information
20th century

??
systematic manipulation of matter
21th century

~~DNA Computers~~
Molecular programming

Acknowledgments

- Microsoft Research

- Andrew Phillips

- Caltech

- Winfree Lab

- U. Washington

- Seelig Lab

Microsoft Engineering Excellence

Questions?

Microsoft

Resources

- Visual DSD at MSR

<http://research.microsoft.com/apps/pubs/default.aspx?id=157262>

- 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/>

- This slide deck and related resources:

[http://lucacardelli.name/Talks/2012-12-06 Molecular Programming \(Redmond\).pptx](http://lucacardelli.name/Talks/2012-12-06%20Molecular%20Programming%20(Redmond).pptx)