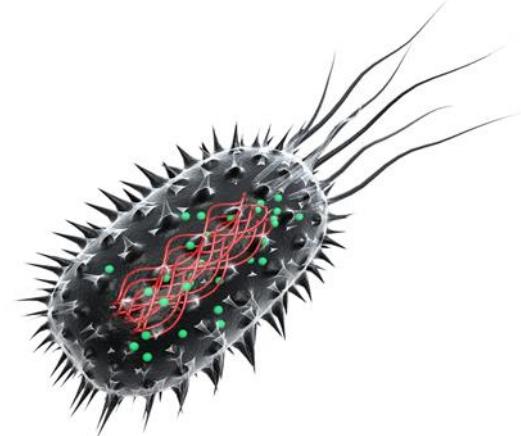


Programming Cells

(/etc)

Colin Gravill
Biological Computation Group
Microsoft Research



Biological Computation

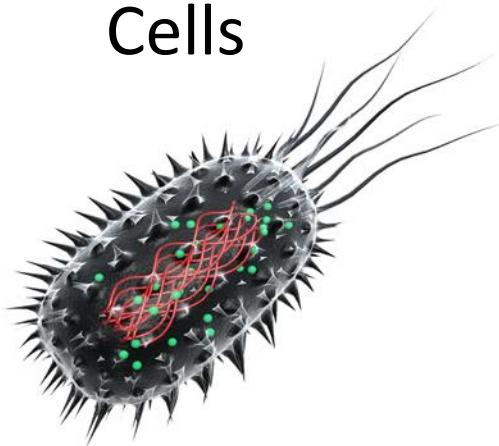
- Cells as computation devices
- Systems of chemical reactions
- The “transcriptor”

Programming (biology)

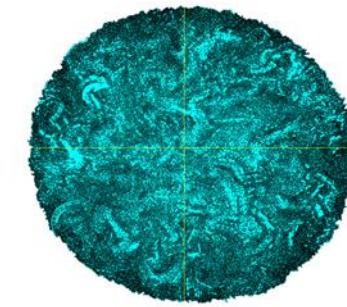
Molecules



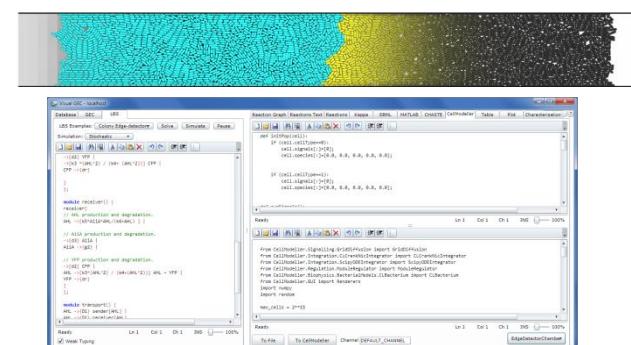
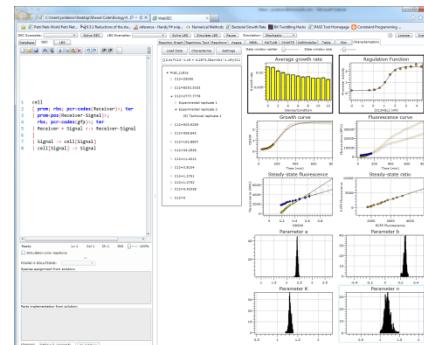
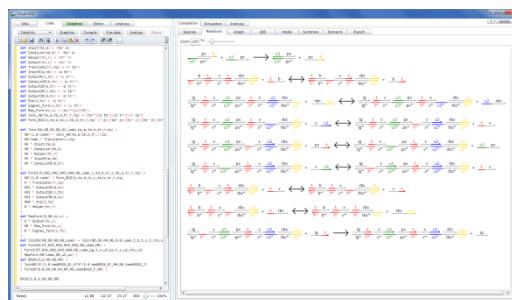
Cells



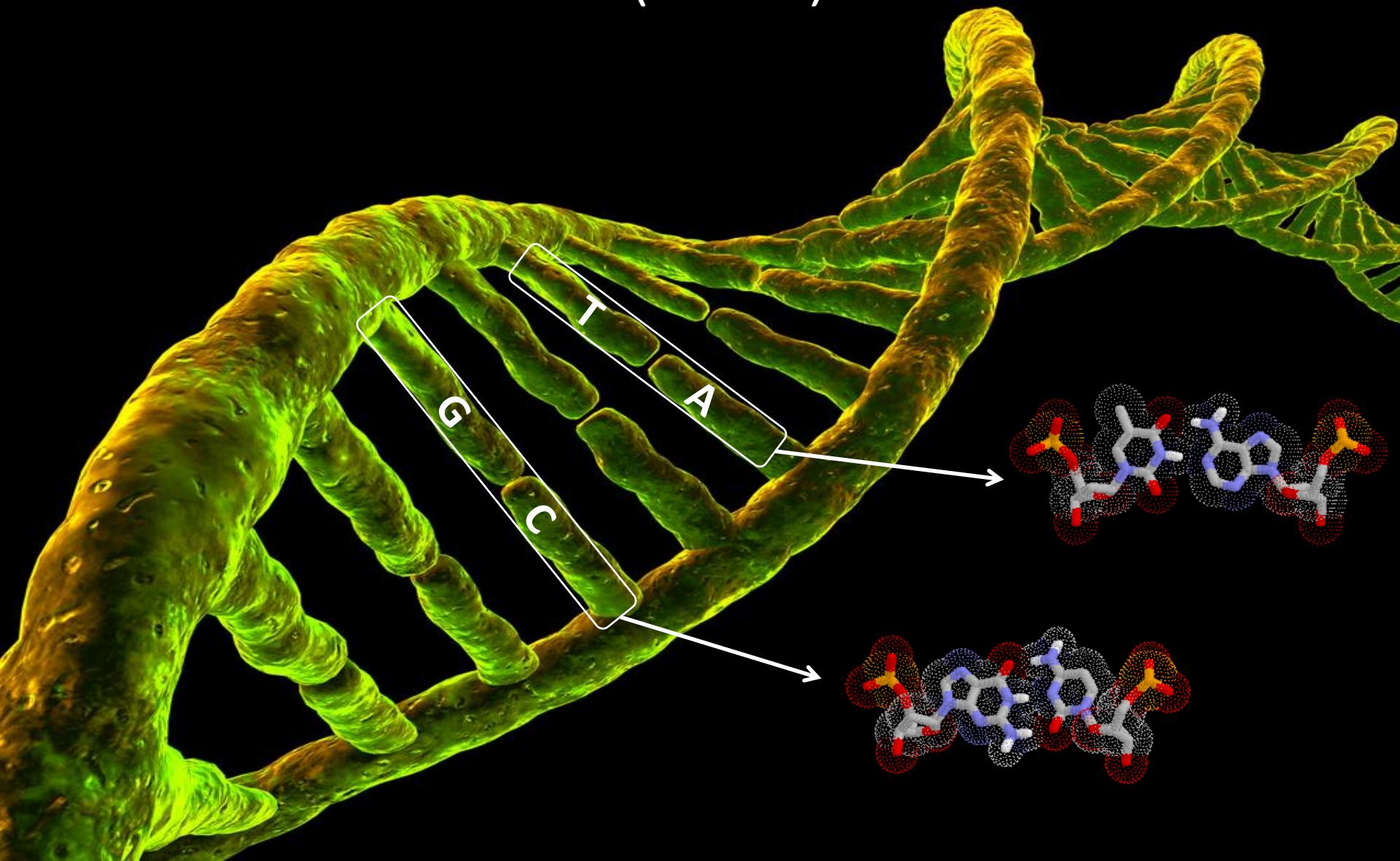
Colonies



Computer Aided Design software

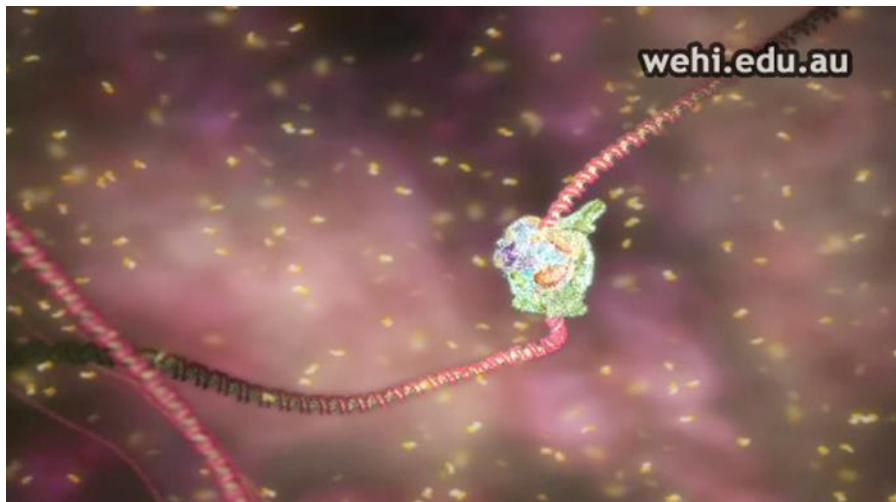


Machine Code (DNA)

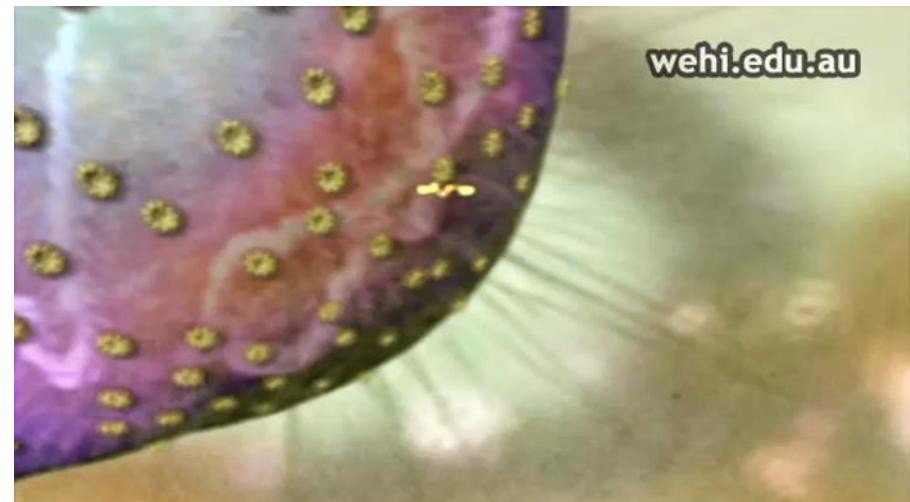


DNA codes the proteins

DNA => RNA => proteins



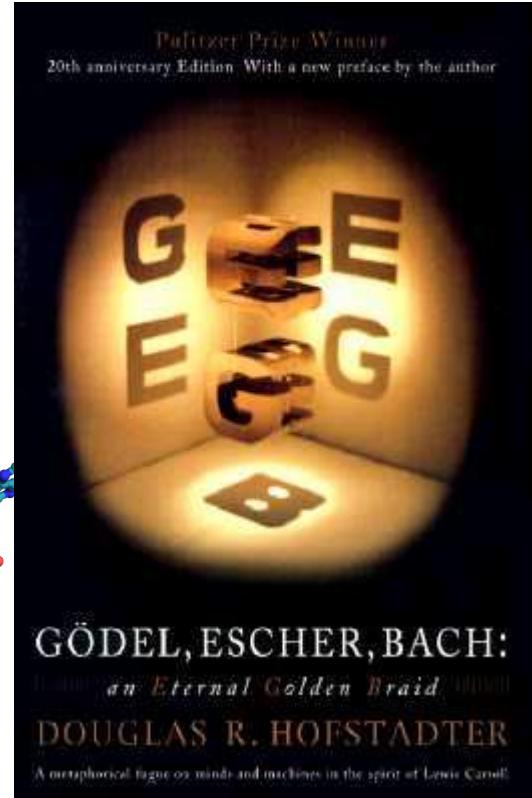
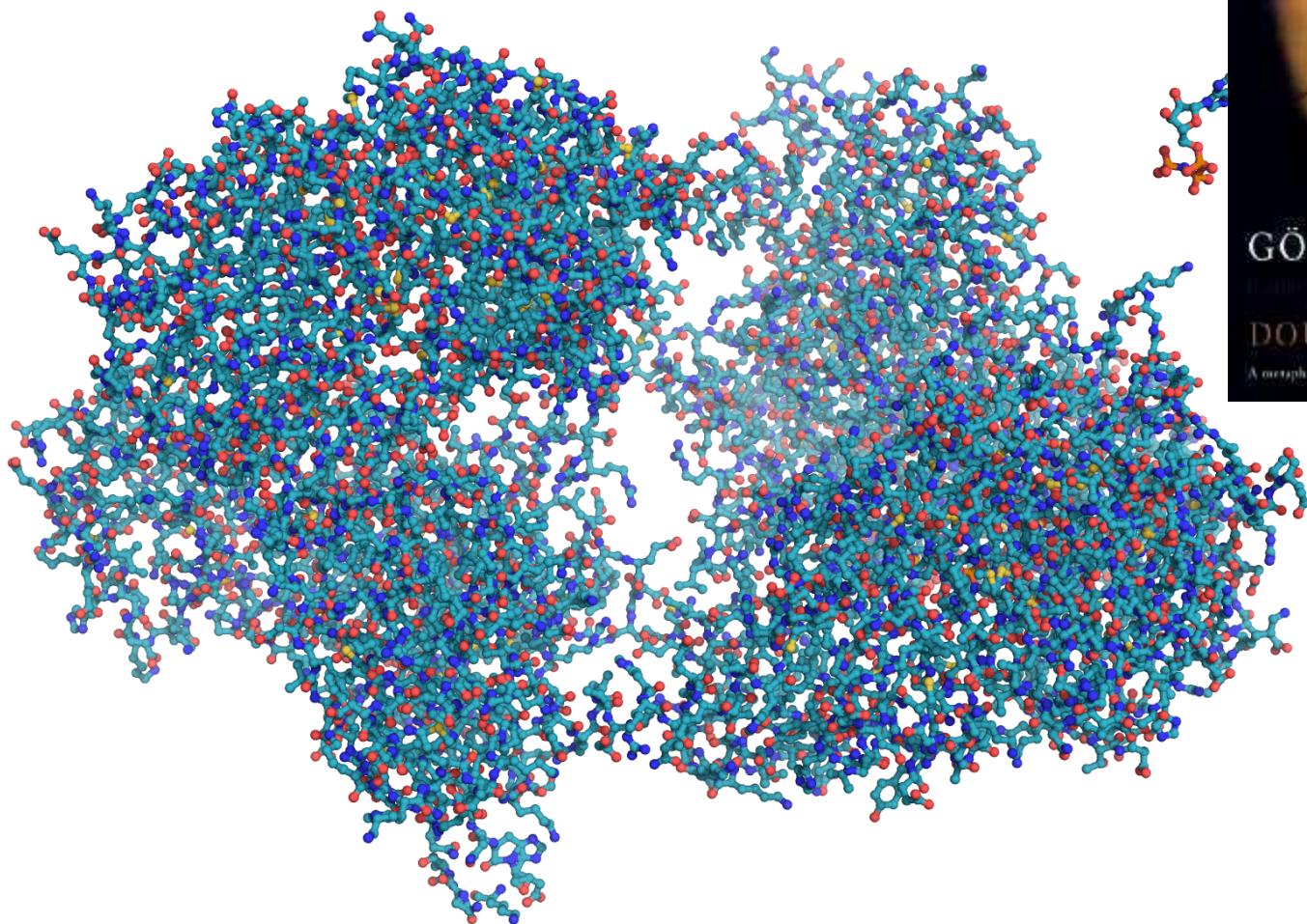
DNA transcription



Messenger RNA translation

Strange loops

Chapter XVI: Self-Ref and Self-Rep

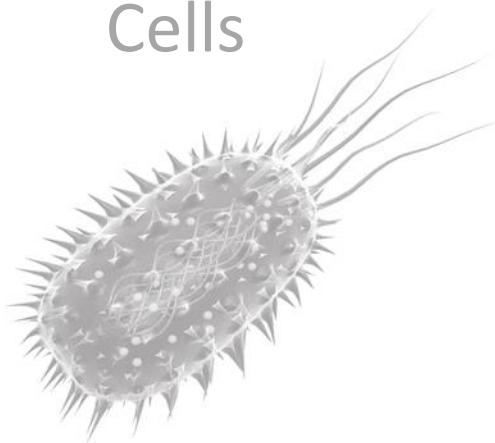


Programming /etc

Molecules



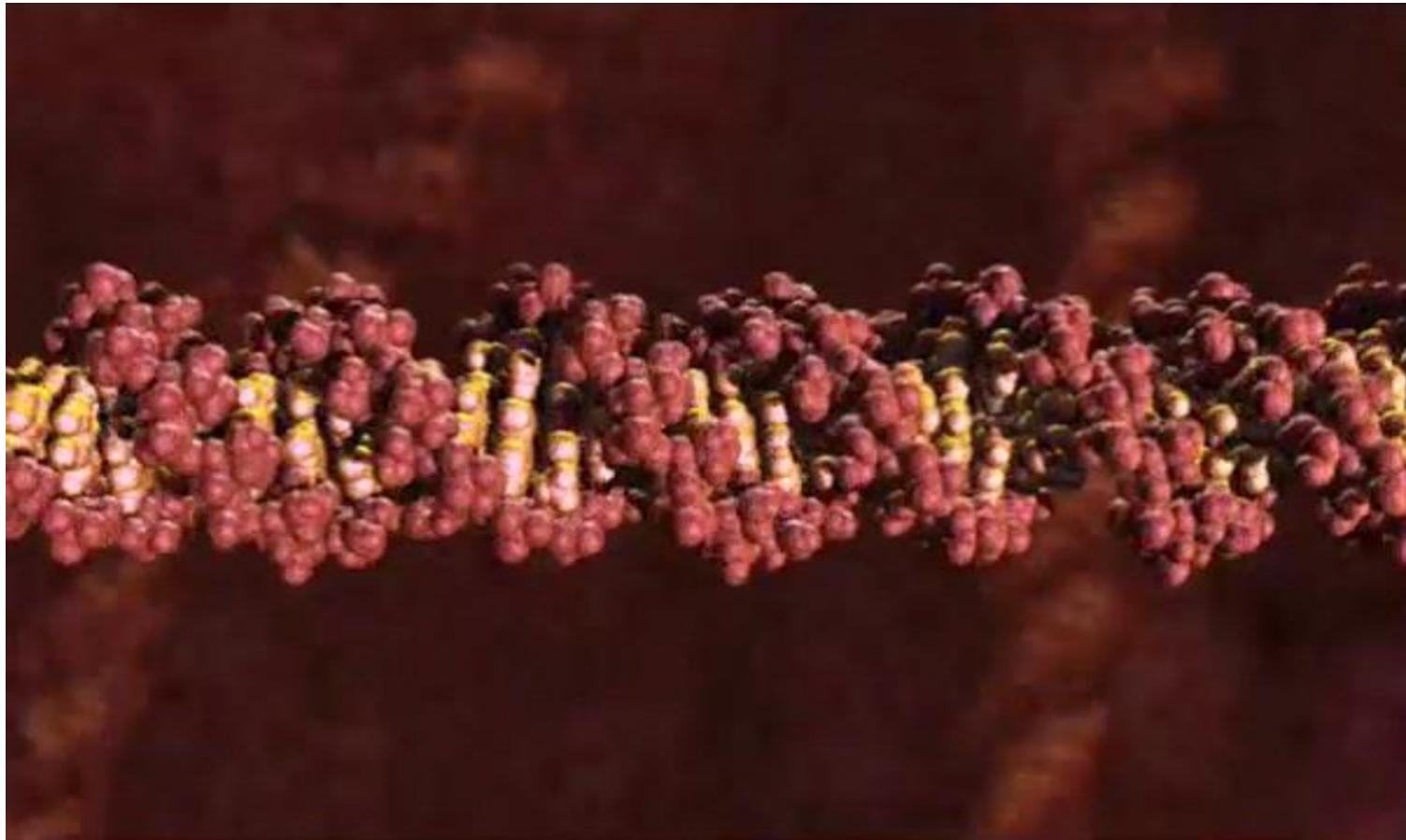
Cells



Colonies

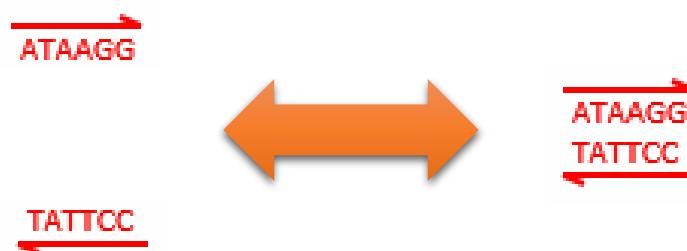


DNA strands

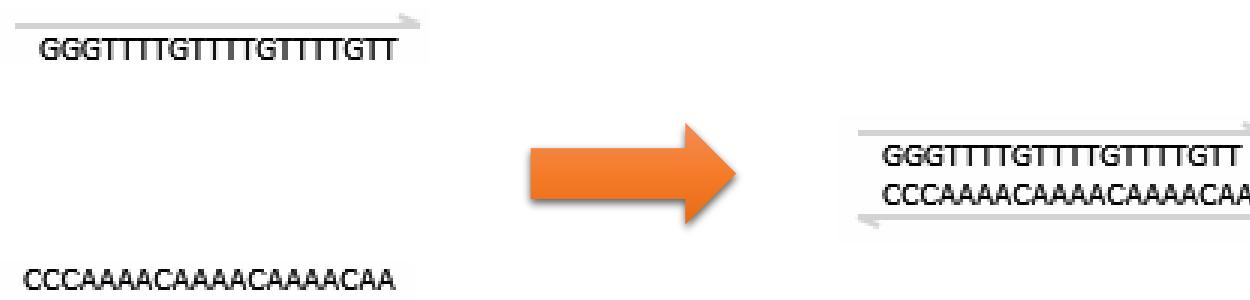


DNA Strand Affinity

Short complementary segments bind *reversibly*

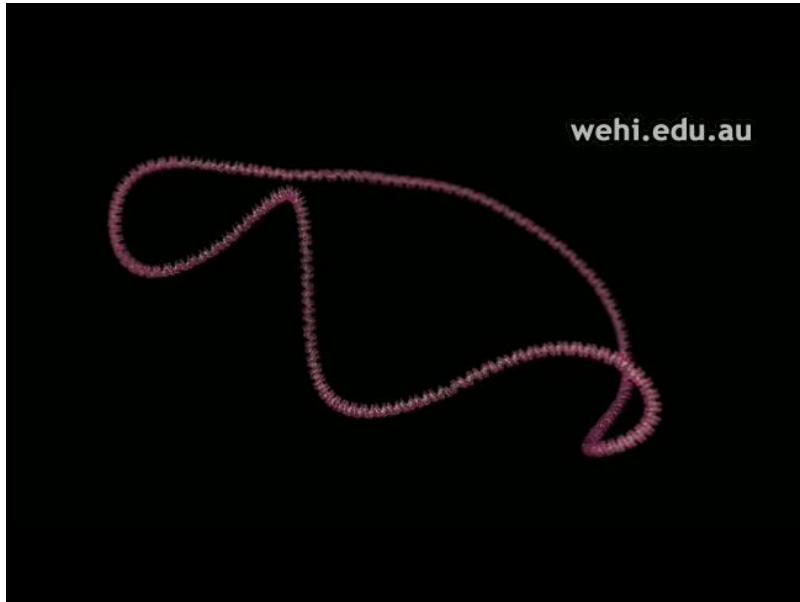


Long complementary segments bind *irreversibly*



DNA Computing with just strands

With Restriction Enzymes



Without Restriction Enzymes



Bernard Yurke

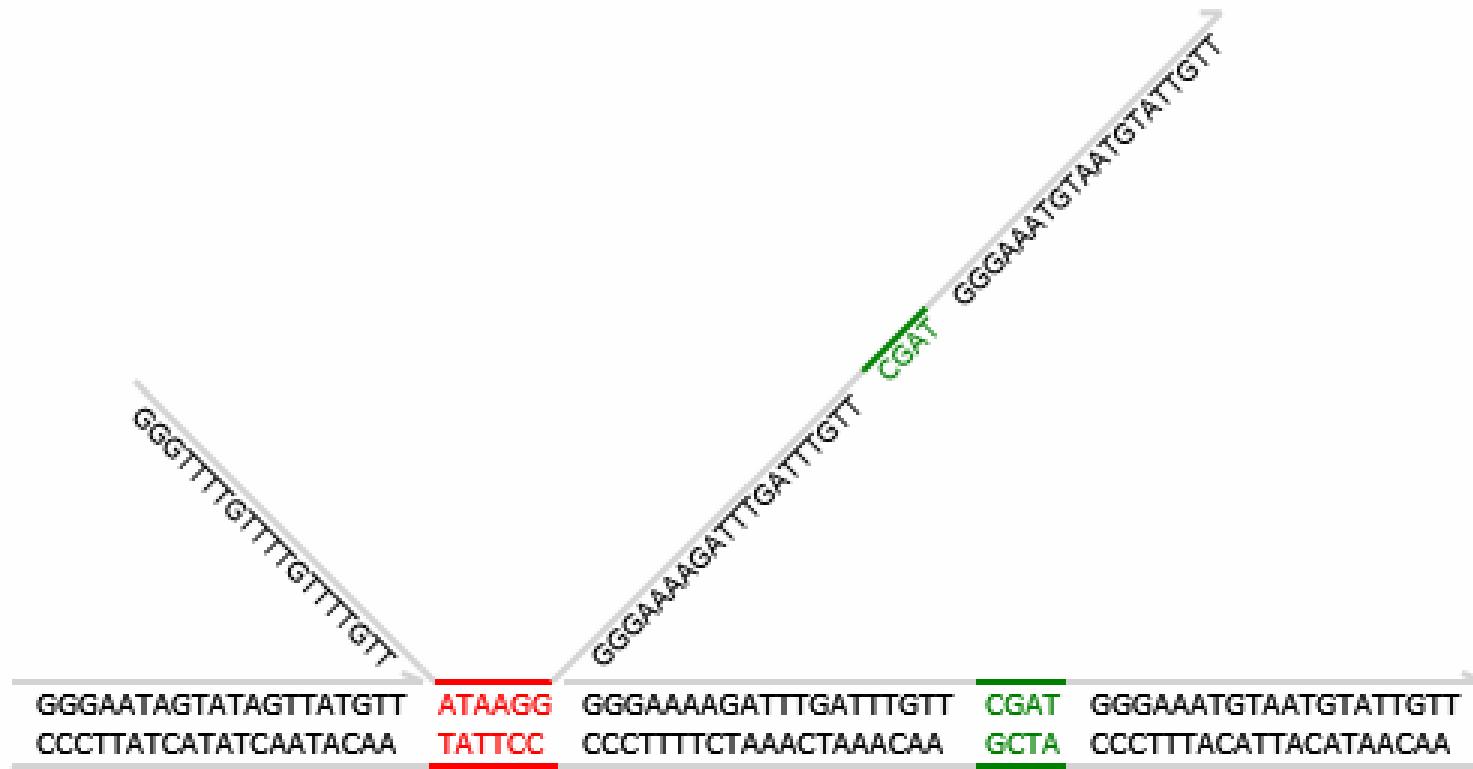
This simple principle can be used to compute with DNA

Bind, Migrate, Displace

GGGTTTTGTTTGTTTGTT **ATAAGG** GGGAAAAGATTGATTGTT **CGAT** GGGAAATGTAATGTATTGTT

GGGAATAGTATAGTTATGTT **TATTCC** GGGAAAAGATTGATTGTT **CGAT** GGGAAATGTAATGTATTGTT
CCCTTATCATATCAATACAA

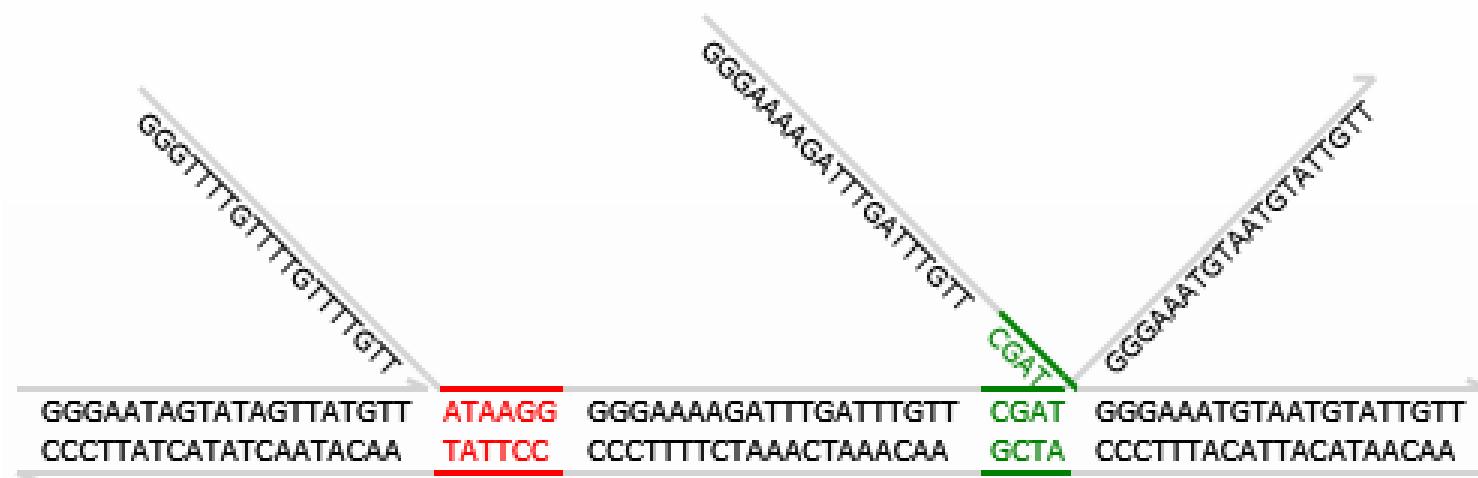
Bind, Migrate, Displace



Bind, Migrate, Displace



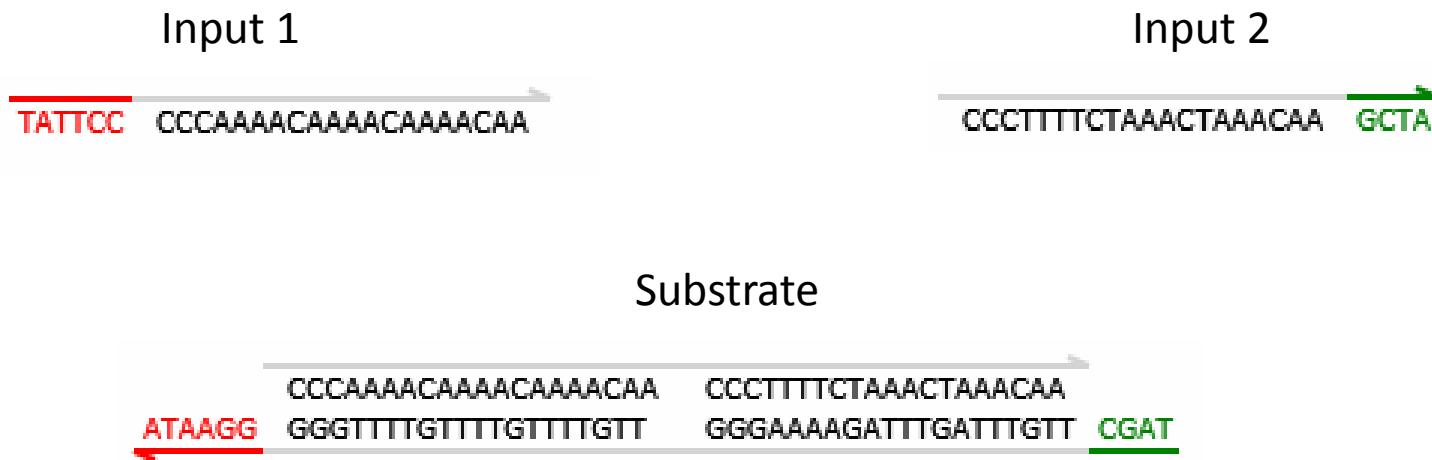
Bind, Migrate, Displace



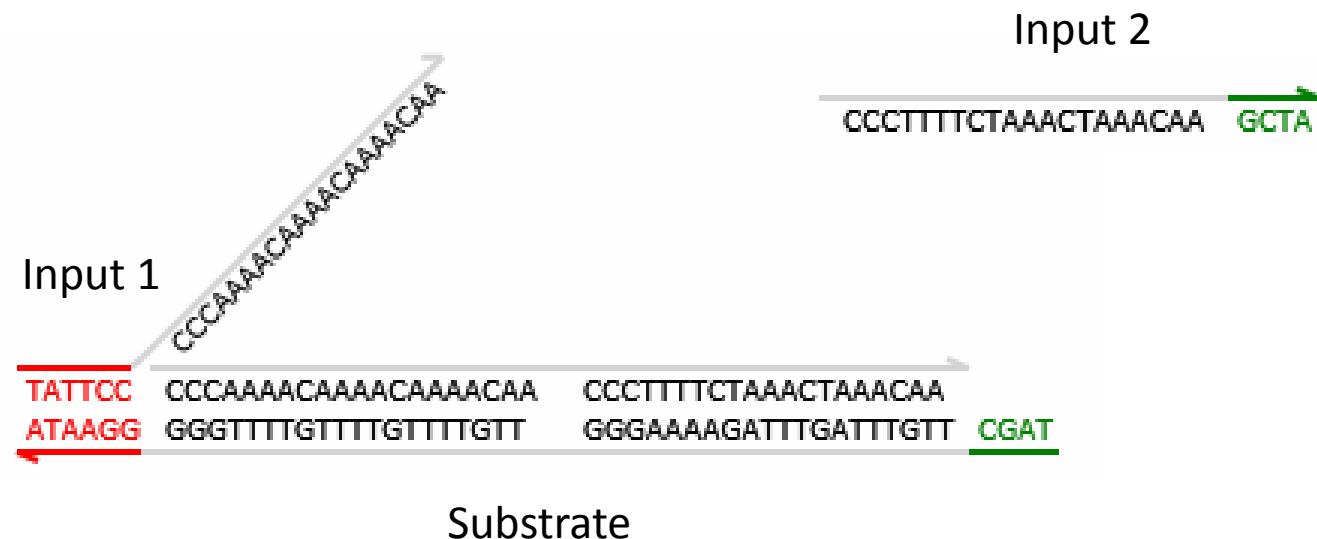
Bind, Migrate, Displace



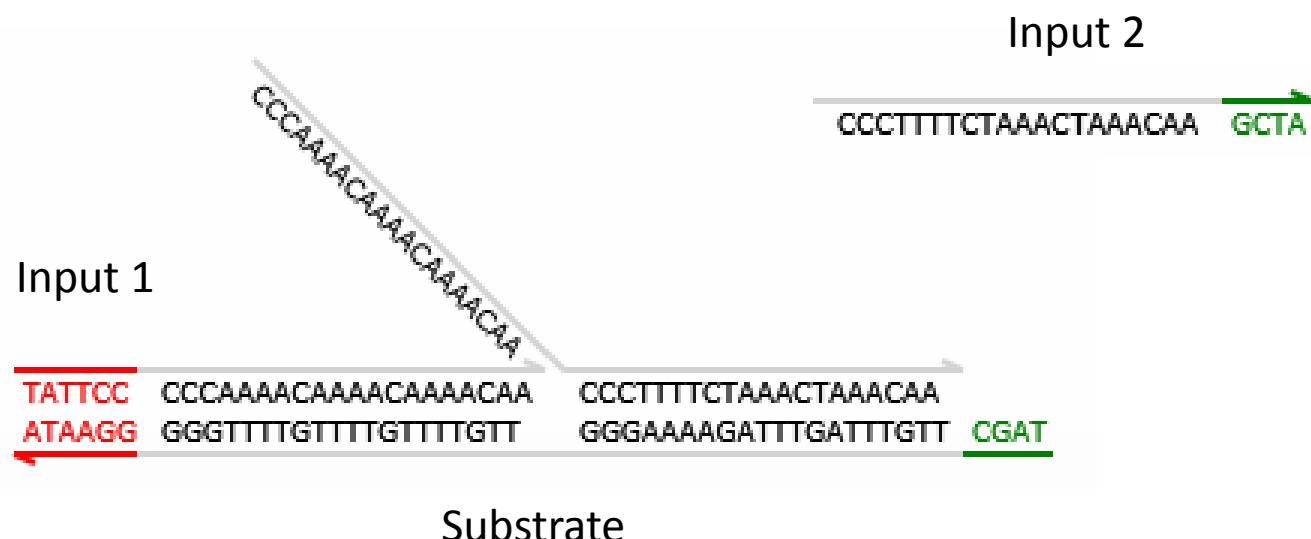
Output = Input1 AND Input2



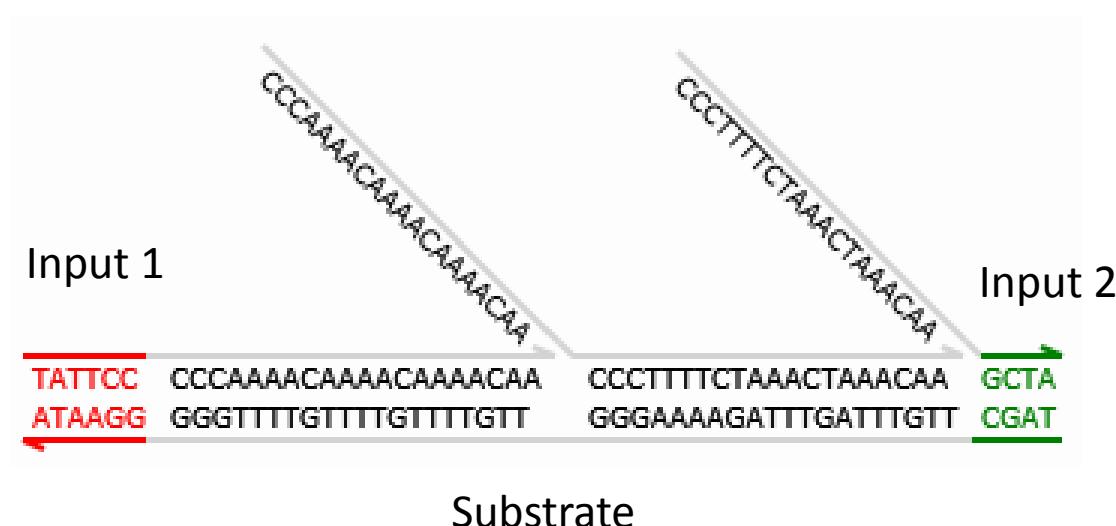
Output = Input1 AND Input2



Output = Input1 AND Input2



Output = Input1 AND Input2



Output = Input1 AND Input2

Output

CCCAAAACAAAACAAAACAA CCCTTTCTAAACTAAACAA

Input 1

TATTCC
ATAAGG

Input 2

GCTA
CGAT

Substrate

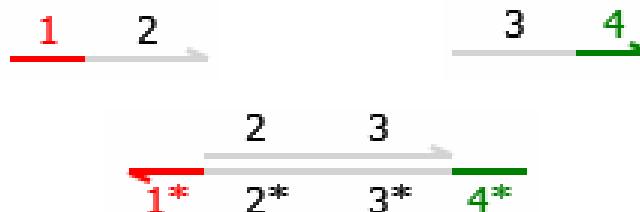
Domain abstraction (assembler)

TATTCC CCCAAAAACAAAACAAAACAA

CCCTTTCTAAACTAAACAA GCTA

ATAAGG CCCAAAAACAAAACAAAACAA
GGGTTTTGTTTGTTTGTT CCCTTTCTAAACTAAACAA
GGGAAAAGATTTGATTGGTT CGAT

- 1 --> (5') TATTCC (3')
4 --> (5') GCTA (3')
2 --> (5') CCCAAAAACAAAACAAAACAA (3')
3 --> (5') CCCTTTCTAAACTAAACAA (3')



```
def Input1() = <1^ 2>
def Input2() = <3 4^>
def AND() = {1^*}[2 3]{4^*}
```

DSD Tool

<http://research.microsoft.com/dna>

Search: Visual DSD

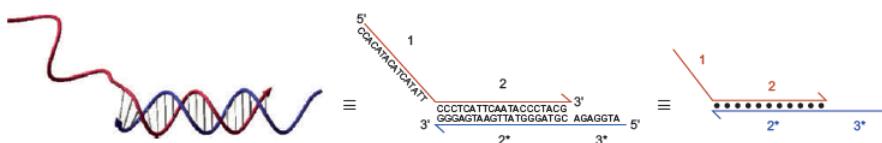
Biological Computation Group
Microsoft Research

computational
science
laboratory

DNA strand displacement

Dynamic DNA nanotechnology using strand displacement reactions

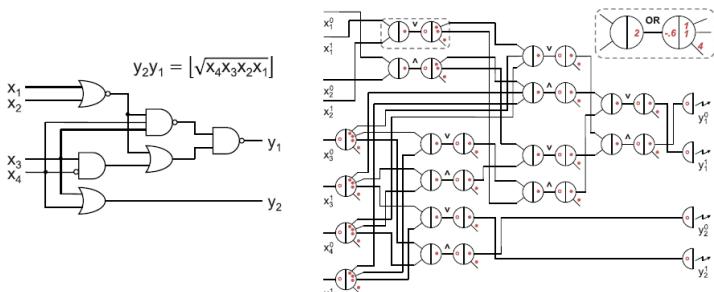
David Yu Zhang¹ and Georg Seelig²



NATURE CHEMISTRY | VOL 3 | FEBRUARY 2011

Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades

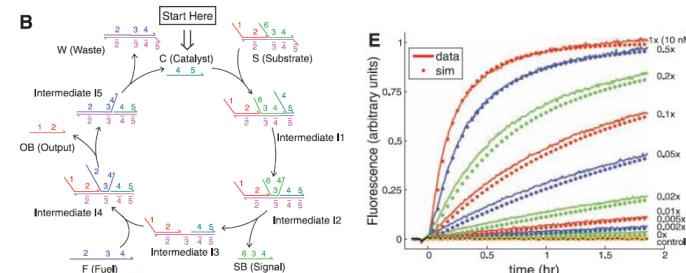
Lulu Qian¹ and Erik Winfree^{1,2,3*}



3 JUNE 2011 VOL 332 SCIENCE

Engineering Entropy-Driven Reactions and Networks Catalyzed by DNA

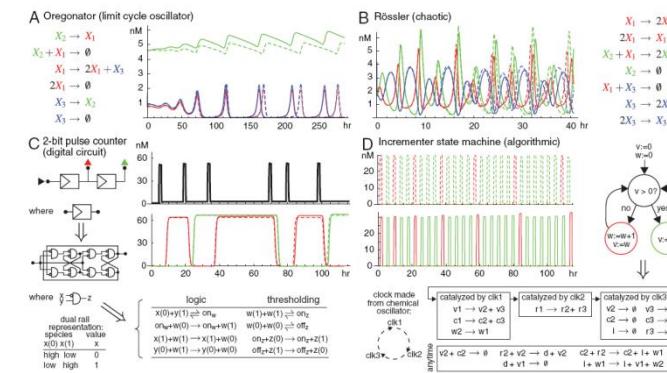
David Yu Zhang,^{1†} Andrew J. Turberfield,² Bernard Yurke,^{3*} Erik Winfree^{1†}



SCIENCE VOL 318 16 NOVEMBER 2007

DNA as a universal substrate for chemical kinetics

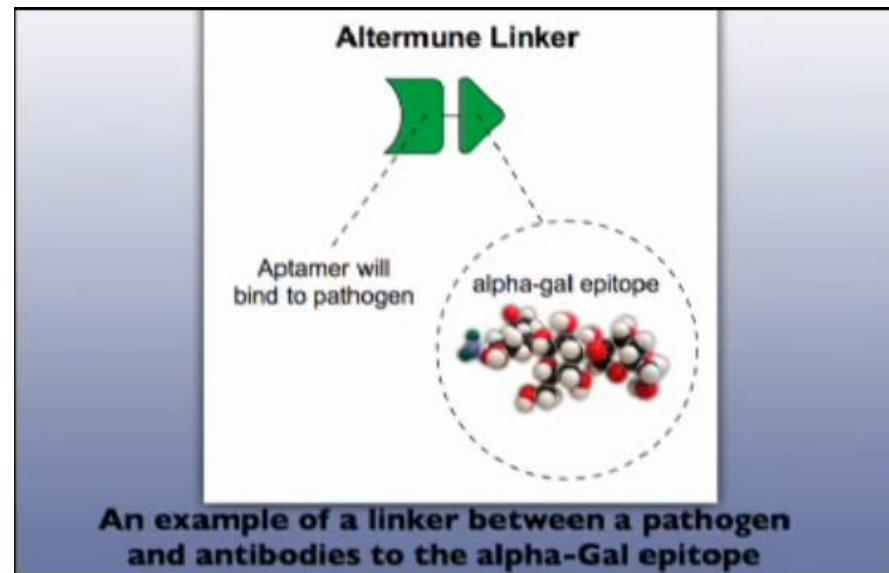
David Soloveichik^{a,b,1}, Georg Seelig^{a,b,1}, and Erik Winfree^{c,1}



PNAS | March 23, 2010 | vol. 107 | no. 12 | 5393–5398

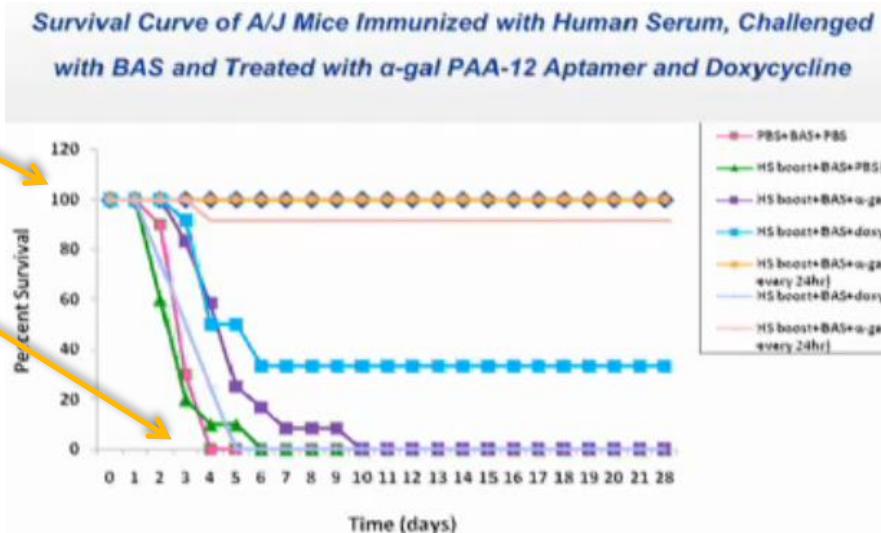
DNA Aptamers

- DNA aptamer binds to:
 - a pathogen (something bad)
 - a molecule our immune system already recognizes and immediately removes (eats) along with anything attached to it



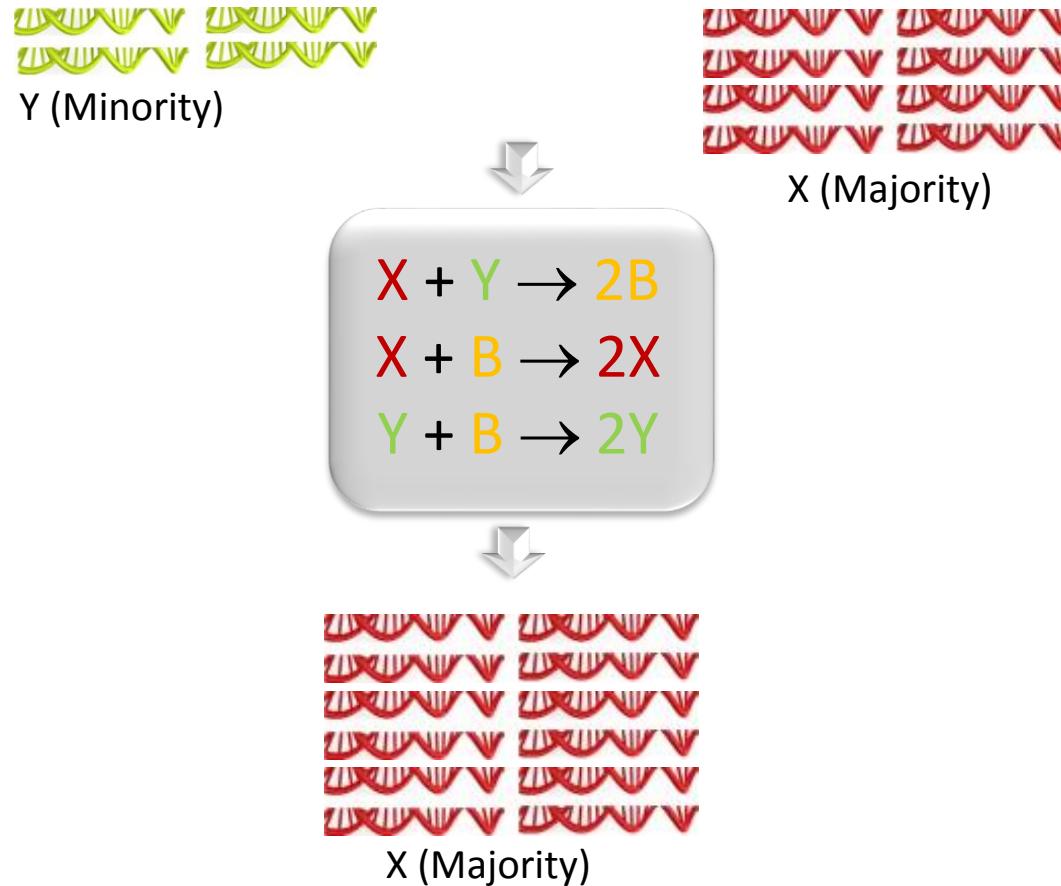
- Result: instant immunity

- Mice poisoned with Anthrax plus aptamer (100% survival)
- Mice poisoned with Anthrax (not so good)

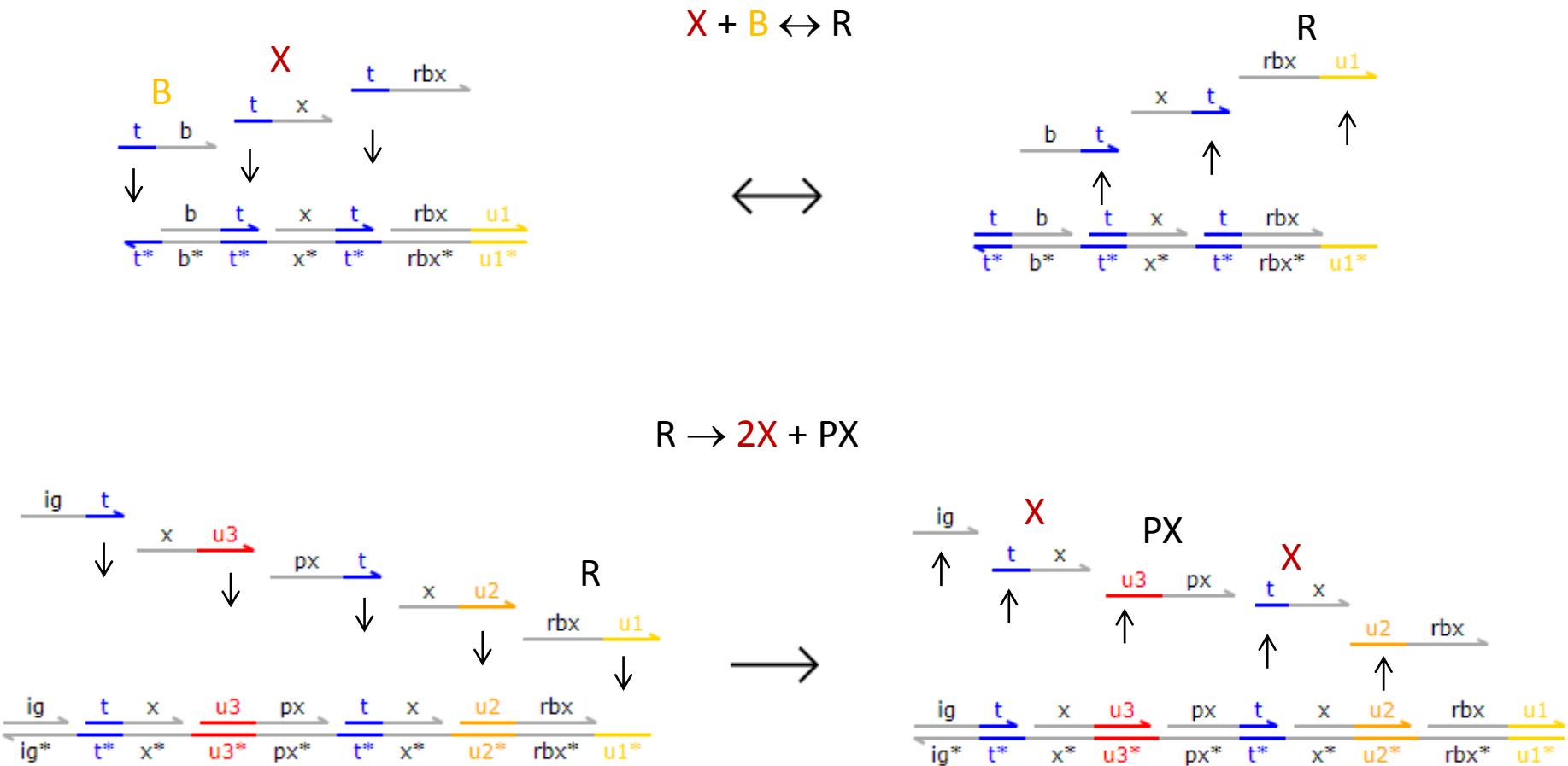
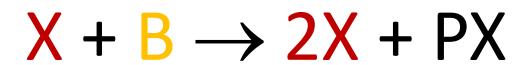


Kary Mullis (incidentally, also Nobel prize for inventing the Polymerase Chain Reaction)

A DNA Consensus Algorithm

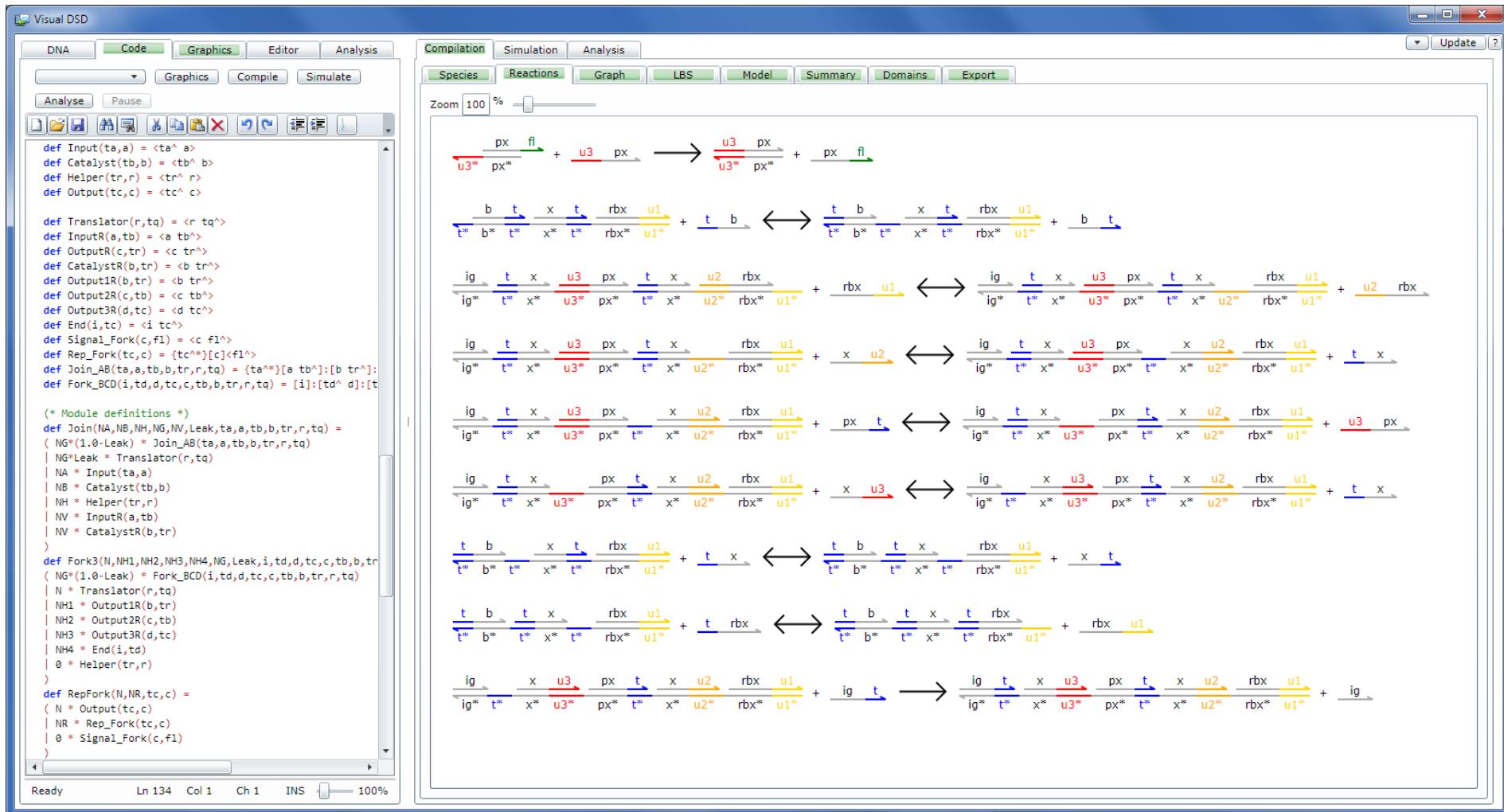


Molecular components



Molecular components

X + B → 2X + PX



Analysing function: Z3 SMT Solver

$$P_{\text{SQRT}_i}(q_0, q) \leftrightarrow O(q) = \lfloor \sqrt{I(q_0)} \rfloor$$

Correct function?

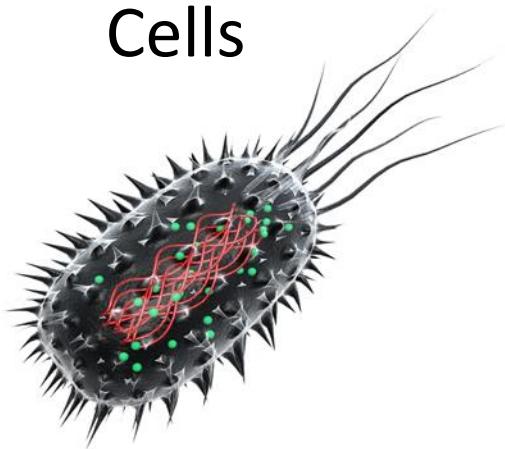
Boyan Yordanov, Christoph Wintersteiger, Youssef Hamadi,
Andrew Phillips, and Hillel Kugler, [Functional Analysis of Large-scale DNA Strand Displacement Circuits](#), in *International Conference on DNA Computing and Molecular Programming (DNA 19)*, Springer, September 2013

Programming cells

Molecules



Cells

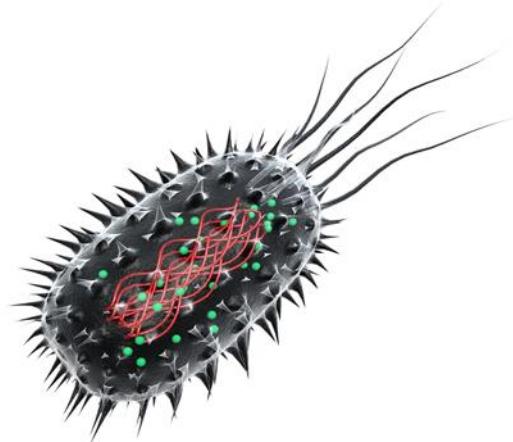


Colonies



Spec sheet: the cell

- 1GB in a millionth of a mm³
- Massively parallel
- Robust to failure
- Self-repairing
- Efficient power supply
- Can reproduce itself



Cross-platform coding



Glowing jellyfish

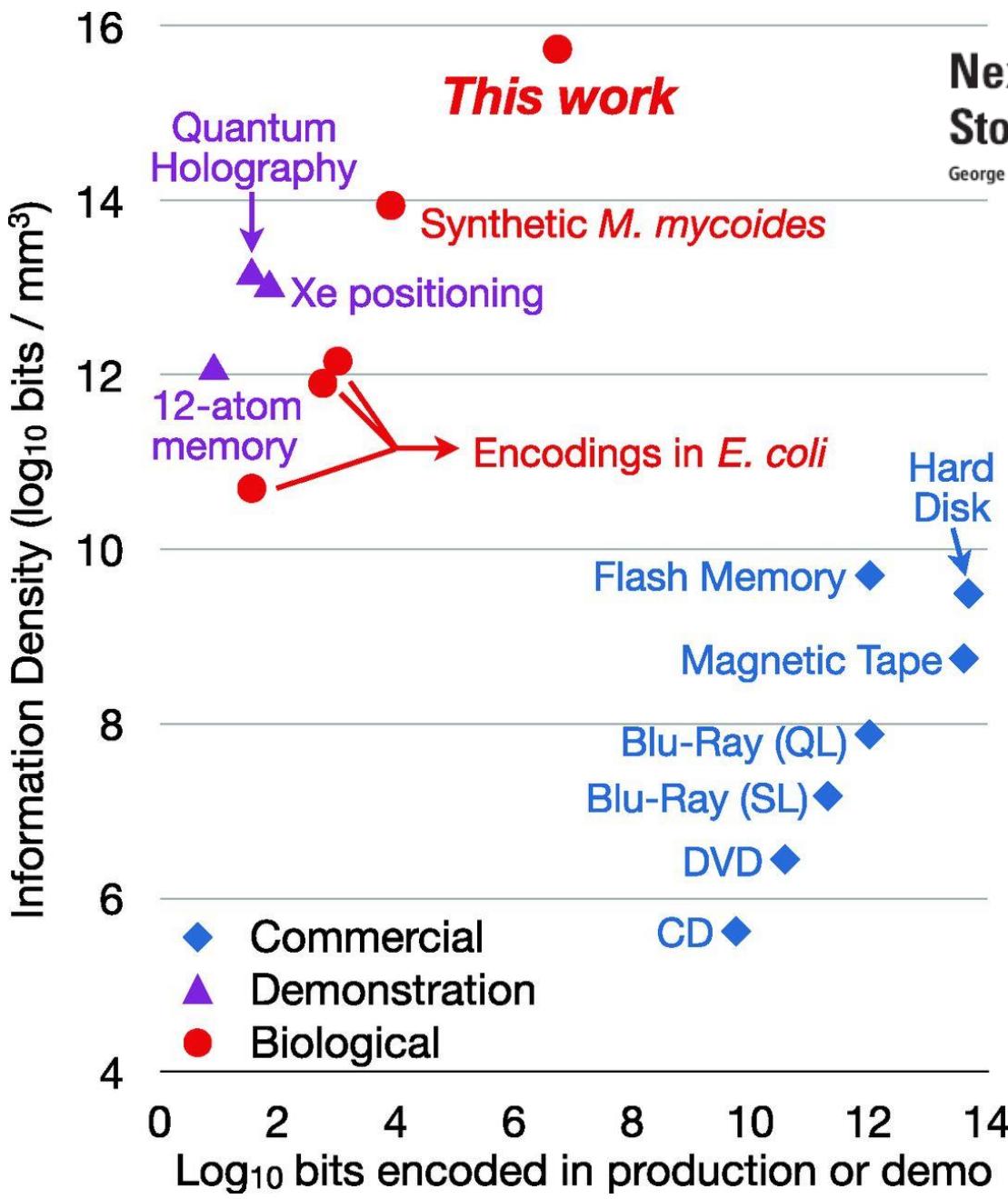


Glowing bacteria

DNA can function across species

Next-Generation Digital Information Storage in DNA

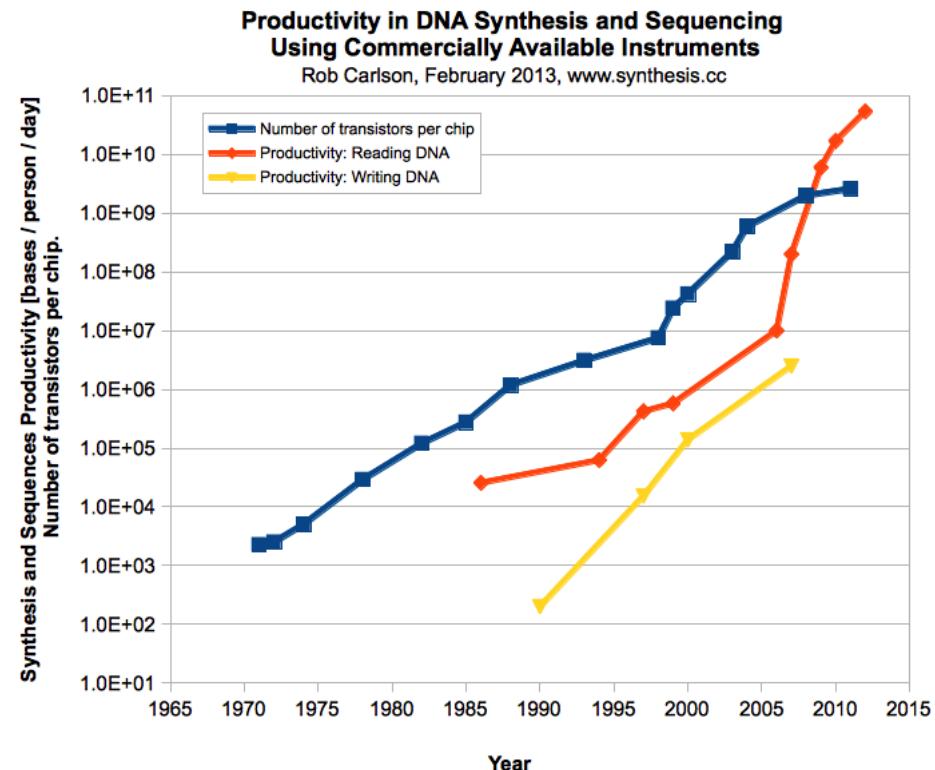
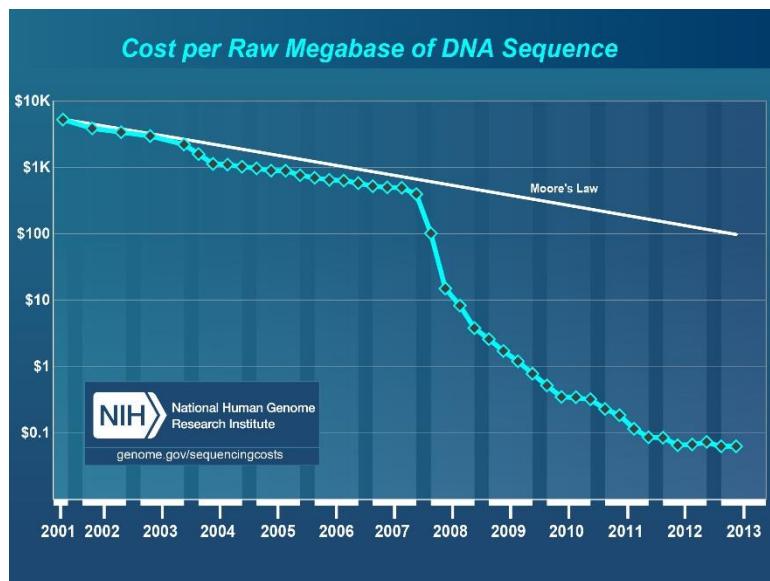
George M. Church,^{1,2} Yuan Gao,³ Sriram Kosuri^{1,2*}



Science

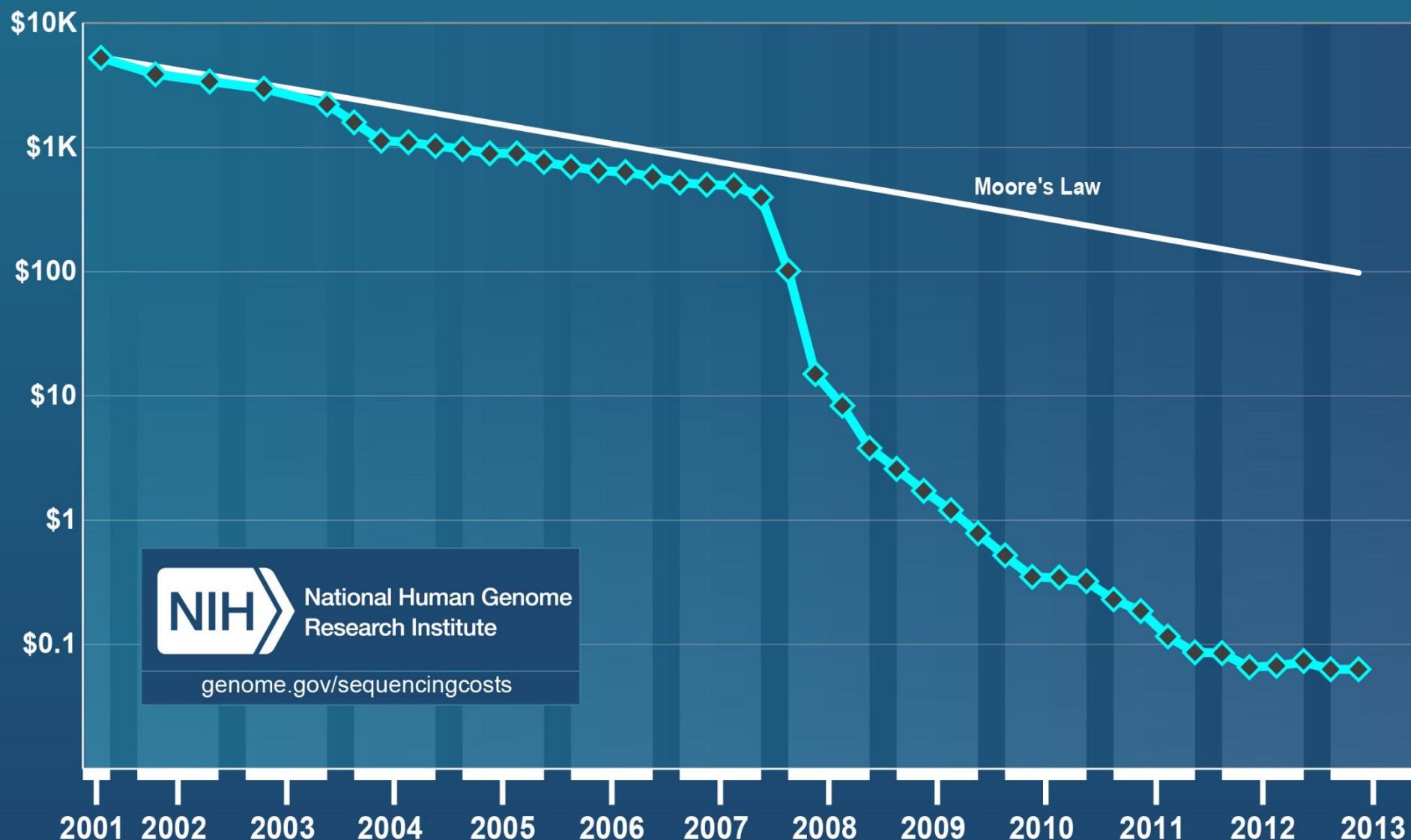
AAAS

Cellular programming economics



<http://www.synthesis.cc/2013/04/updated-dna-cost-and-productivity-curves-plus-a-few-more-thoughts-on-moores-law.html>

Cost per Raw Megabase of DNA Sequence

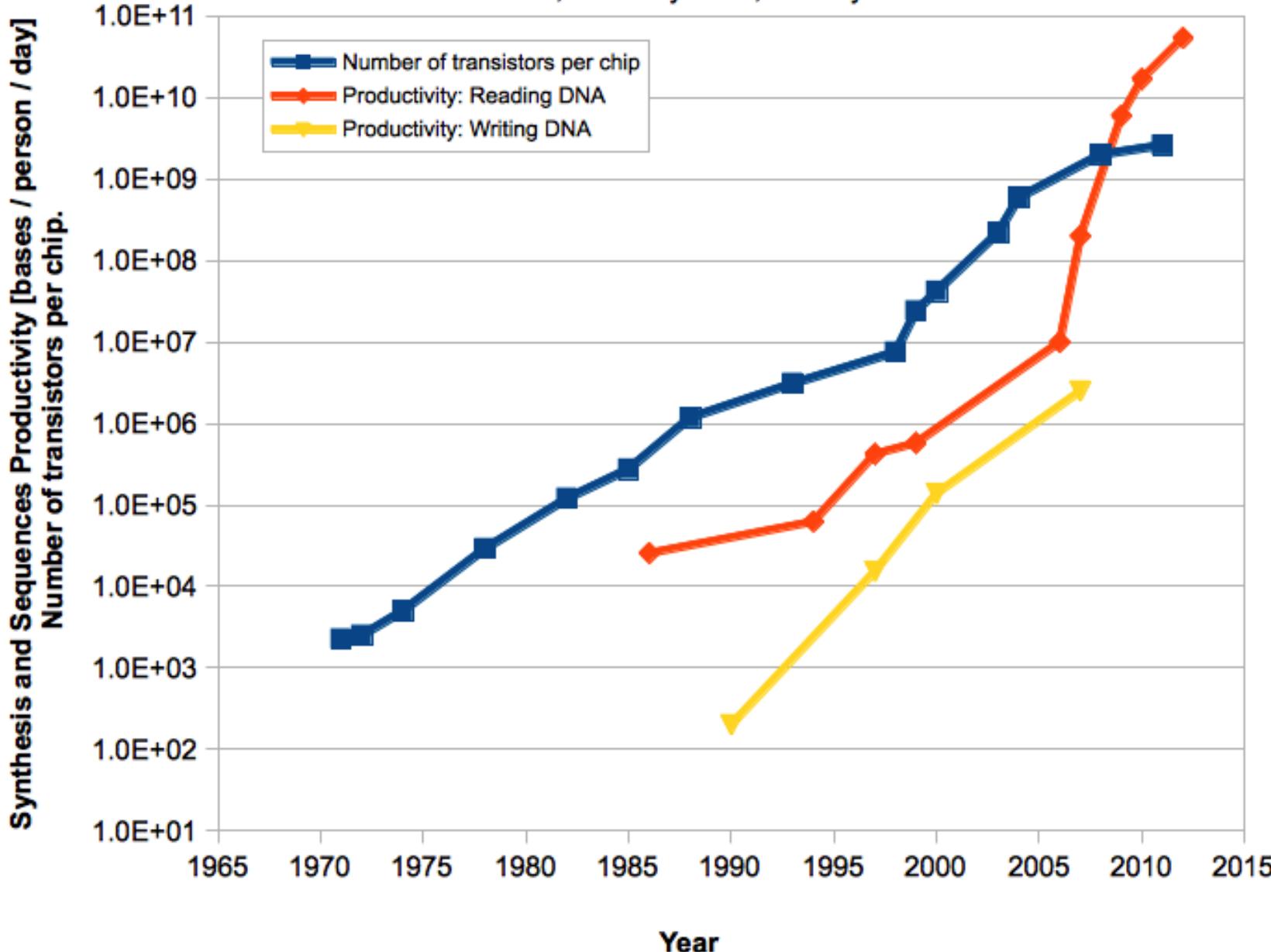


National Human Genome
Research Institute

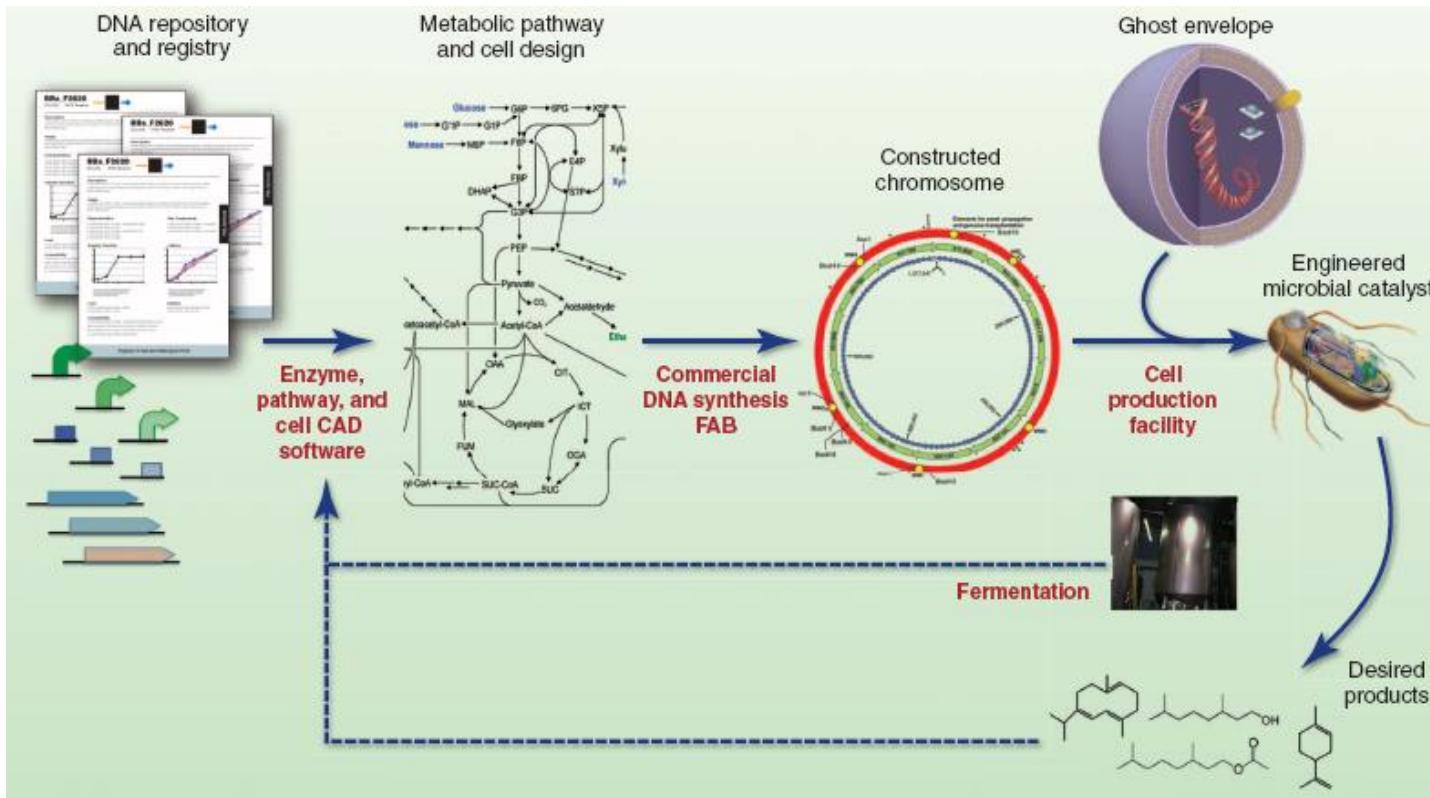
genome.gov/sequencingcosts

Productivity in DNA Synthesis and Sequencing Using Commercially Available Instruments

Rob Carlson, February 2013, www.synthesis.cc



Synthetic Biology Software



**Manufacturing Molecules
Through Metabolic Engineering**

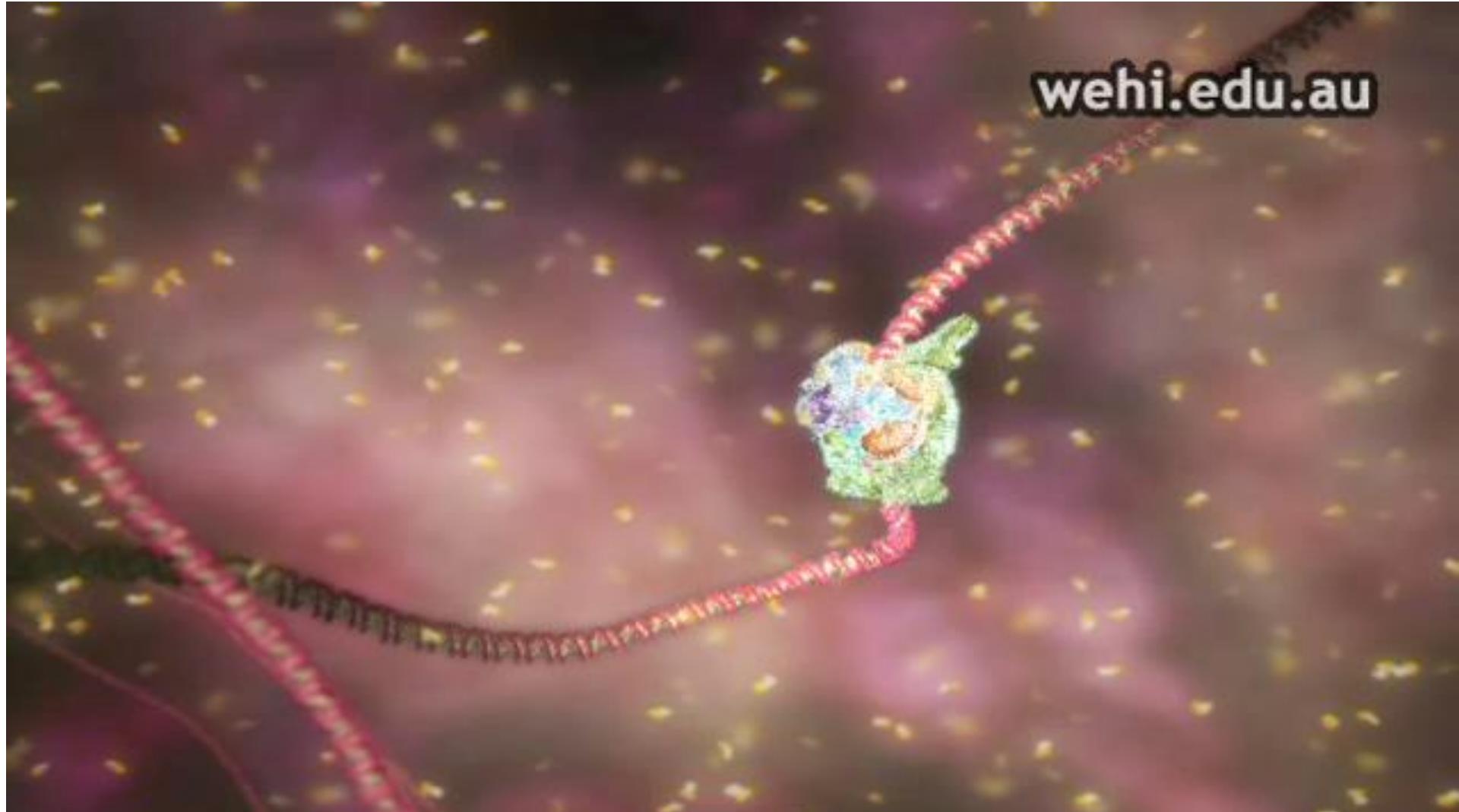
Jay D. Keasling^{1,2,3}

Cell programming

- Complex and error-prone
 - Still learning how to write cell programs (lost the source)
 - Resulting cells do not behave as we intend them to
 - Issues of reliability, toxicity, strain on the host cell
- Programs are increasingly complex
 - Can no longer be designed by trial and error
 - Computer software is needed to accelerate progress

Genetic Engineering of Cells (GEC)

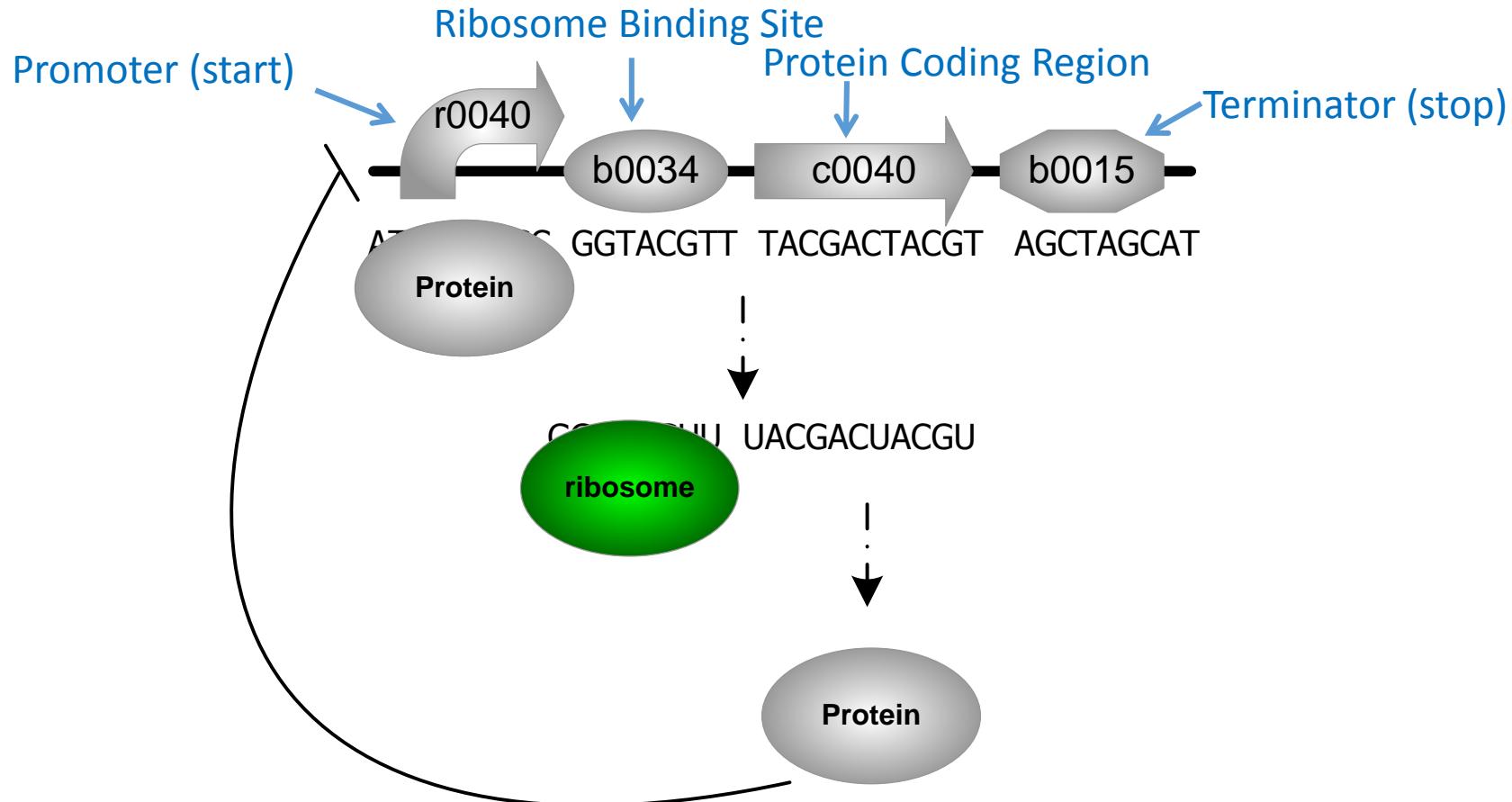
Transcription



wehi.edu.au

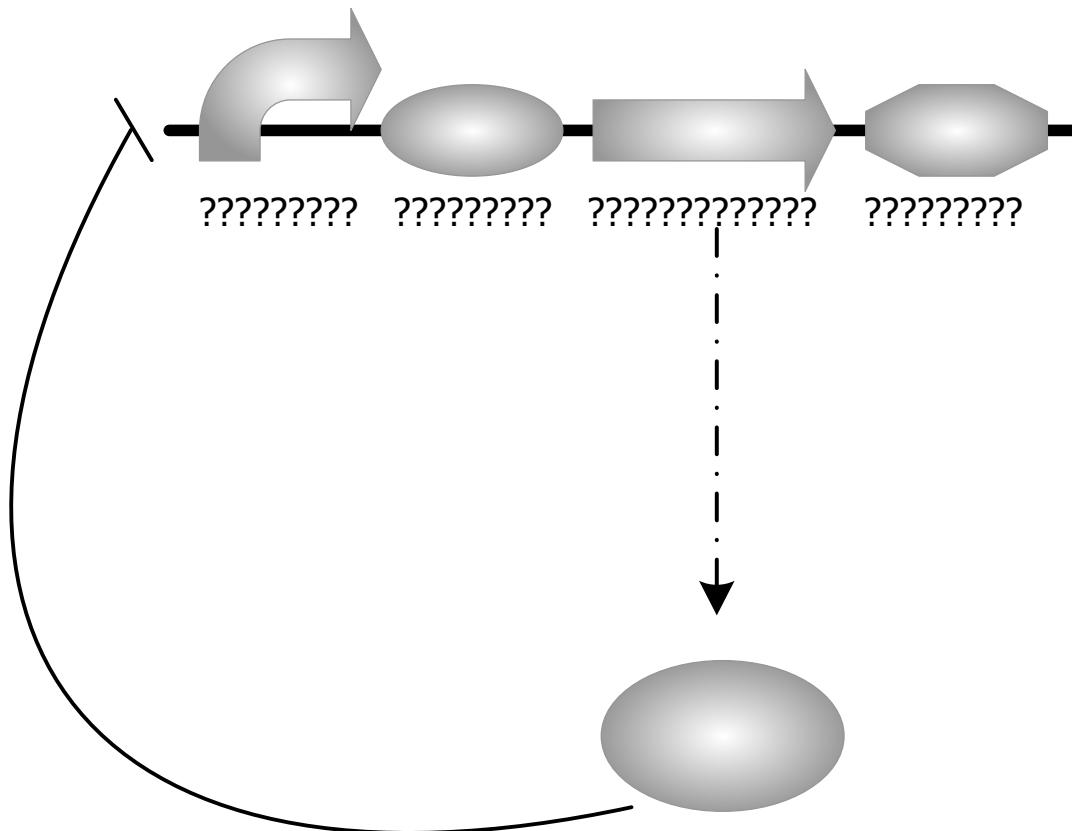
Low-level DNA Language

A simplified view of DNA instructions

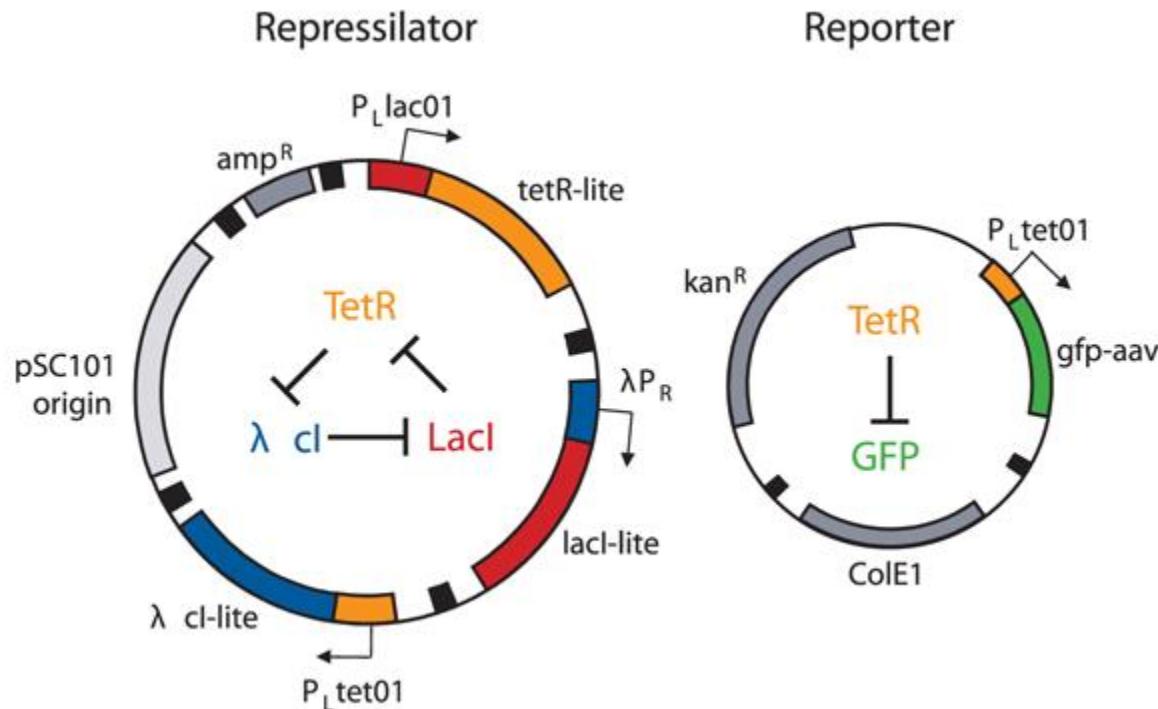


High-level DNA Language

Given a design, automatically determine the DNA



Repressor



GEC Tool

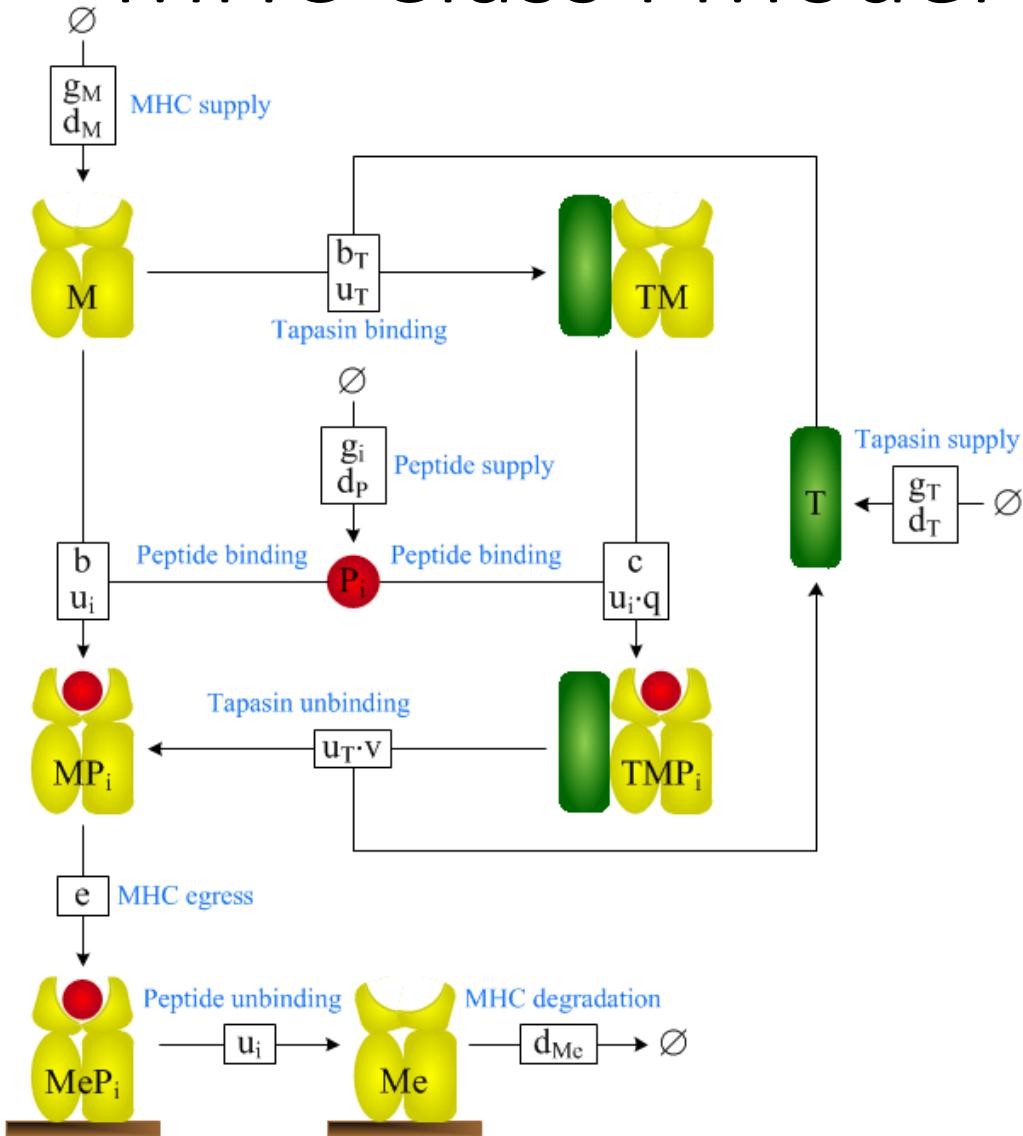
<http://research.microsoft.com/gec>

Search: Visual GEC

Biological Computation Group
Microsoft Research

computational
science
laboratory

MHC Class I Model



Kinetic Rates:

g generation
 d degradation
 b binding
 c binding
 u unbinding
 e egress

Effects:

v increased tapasin unbinding
 q increased peptide unbinding

Components:

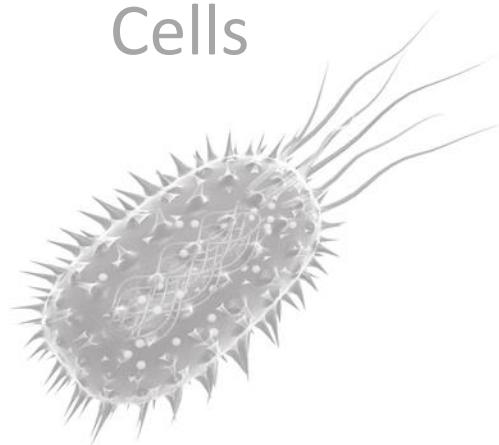
M	MHC
P_i	Peptide
MP_i	MHC-Peptide complex
T	Tapasin
TM	Tapasin-MHC complex
TMP_i	Tapasin-MHC-Peptide complex
MeP_i	Egressed MHC-Peptide complex
Me	Egressed MHC

Programming (macro)

Molecules



Cells



Colonies

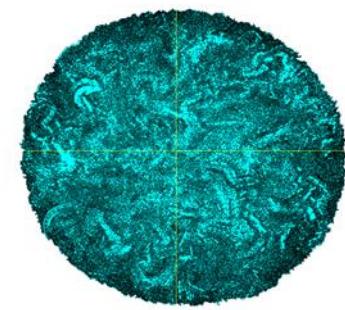
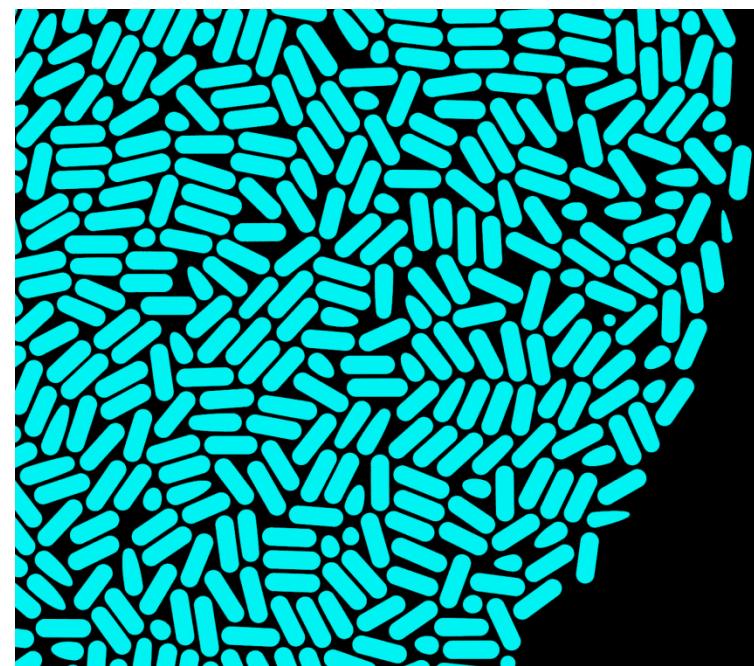
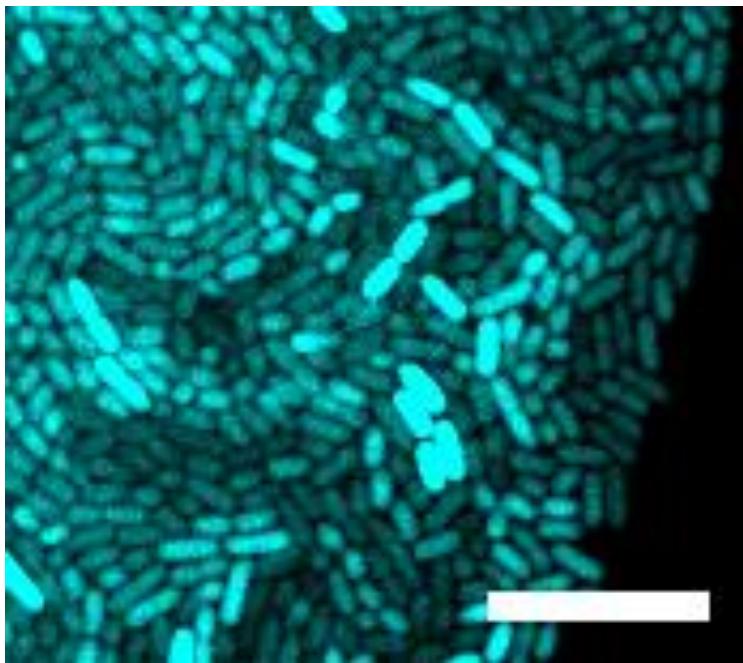




Image by Fernan Federici, Haseloff Lab

Biophysical Modelling



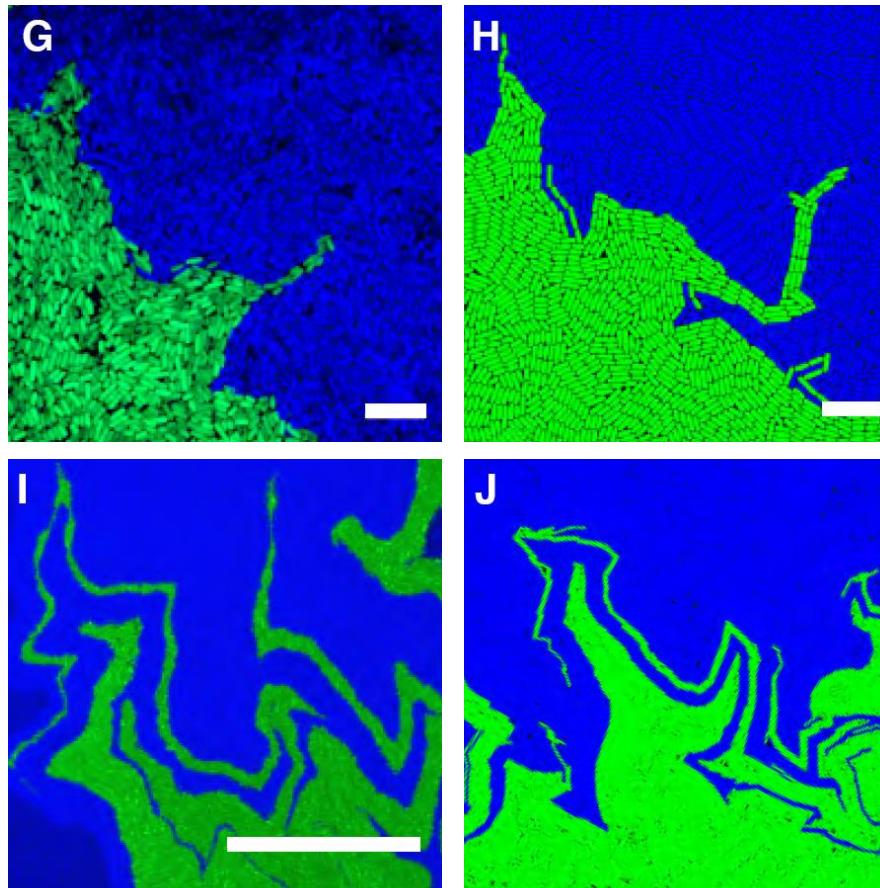
Biophysical patterning



Programming cell colonies

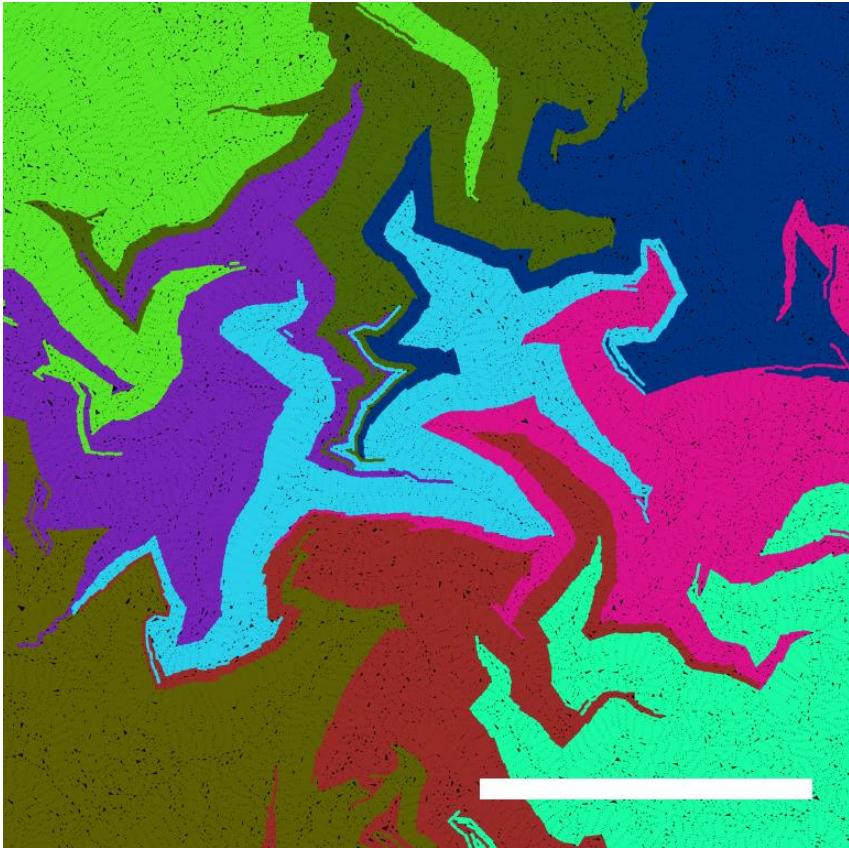
Predicting colony morphology

Experiments Simulations

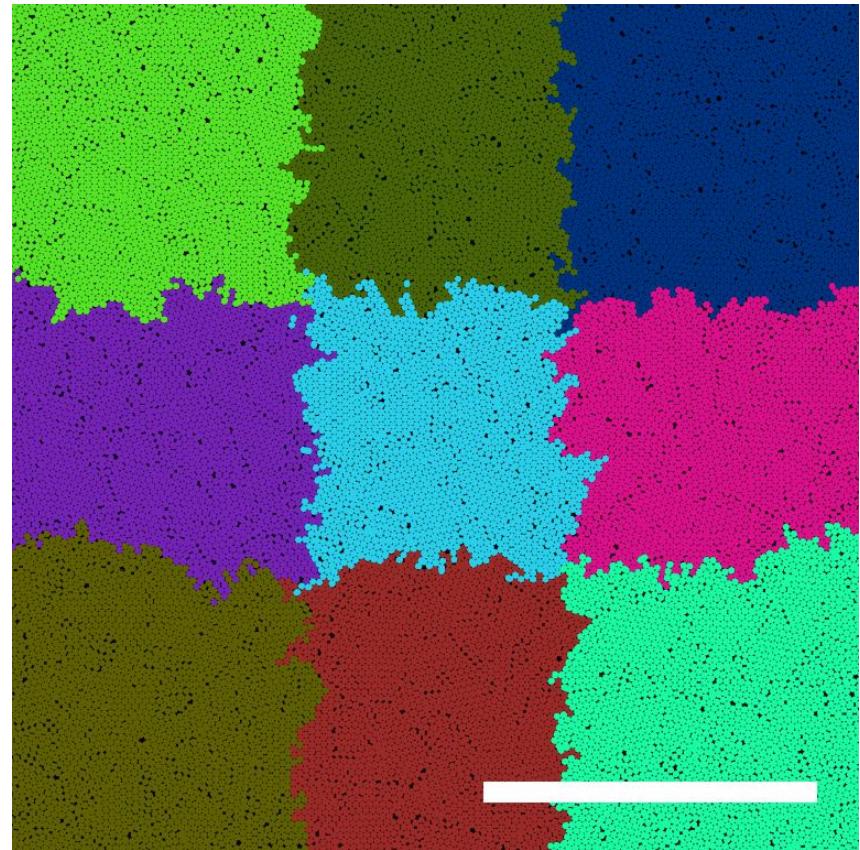


Round vs rod-shaped cells

Rod-shaped cells (simulation)

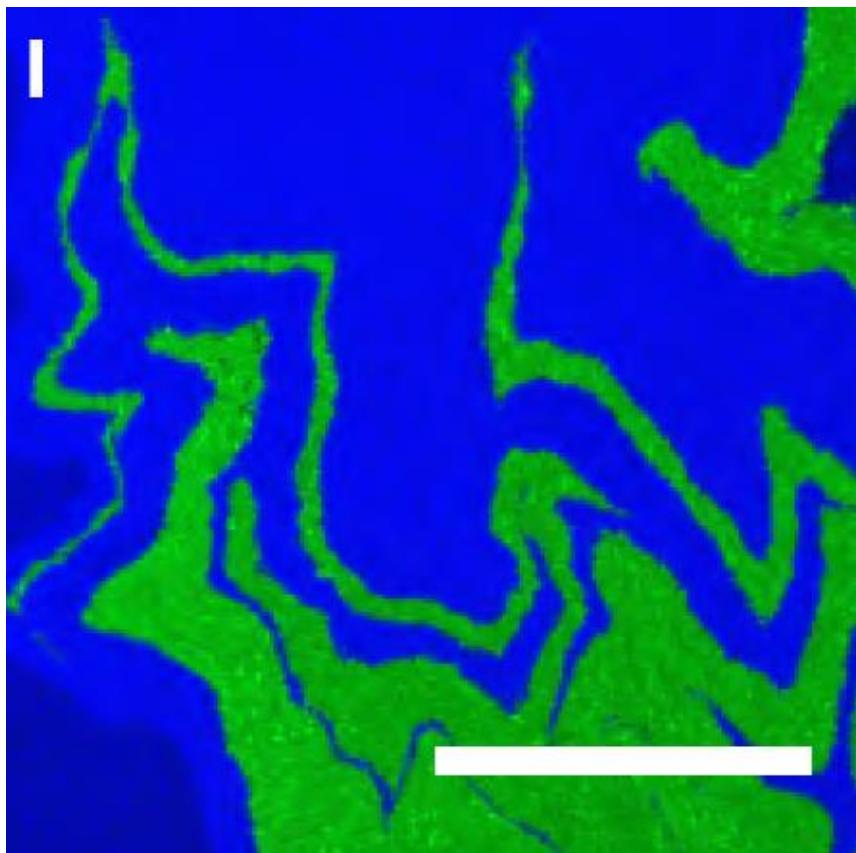


Round cells (simulation)

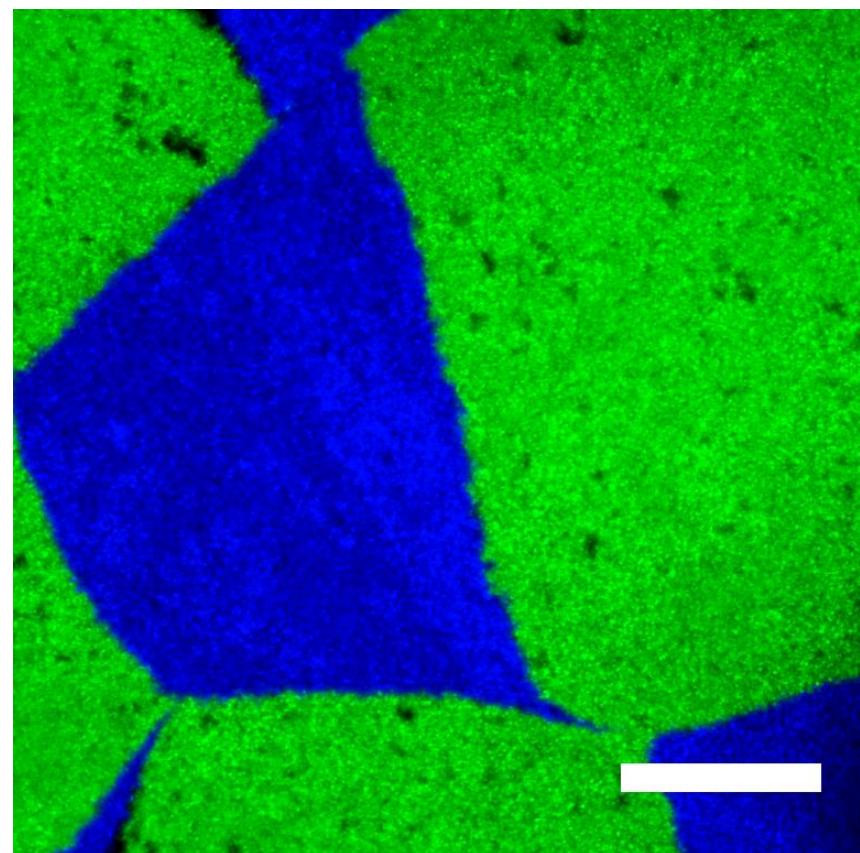


Round vs rod-shaped cells

Rod-shaped cells (experiment)



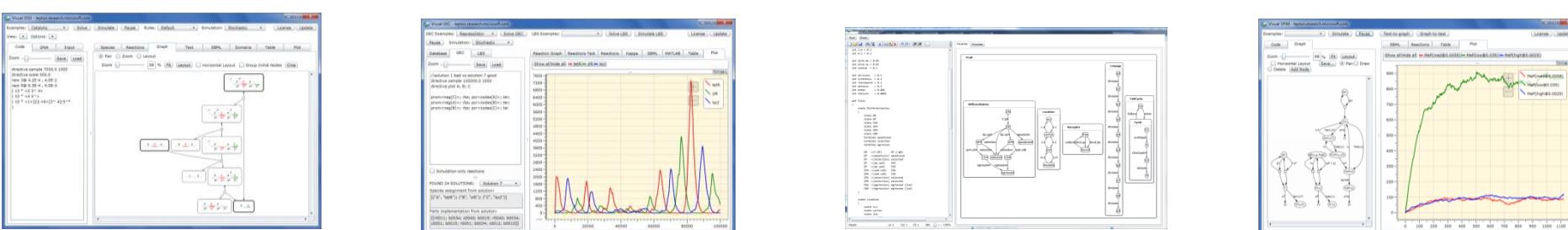
Round cells (experiment)



Biological Computation Group



DNA Computing Synthetic Biology Development Immunology



DSD GEC Biocharts SPiM

Modelling Engine

Simulation (Stochastic, ODE, PDE);
Analysis (probabilistic model-checking, verification with Z3);
Parameter estimation (Filzbach); Visualisation (DDD,MSAGL);
Unbounded computation
Interoperability (SBML,C#, PRISM, Matlab, Chaste, CellModeller)

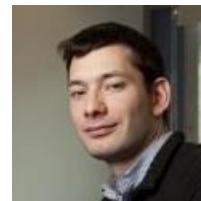
Biological Computation Group



Hillel
Kugler



Neil
Dalchau



Boyan
Yordanov



Chris
McEwan



Sara-Jane
Dunn



Michael
Pedersen



Paul
Grant



Filippo
Polo



Rasmus
Petersen



Andrew
Phillips



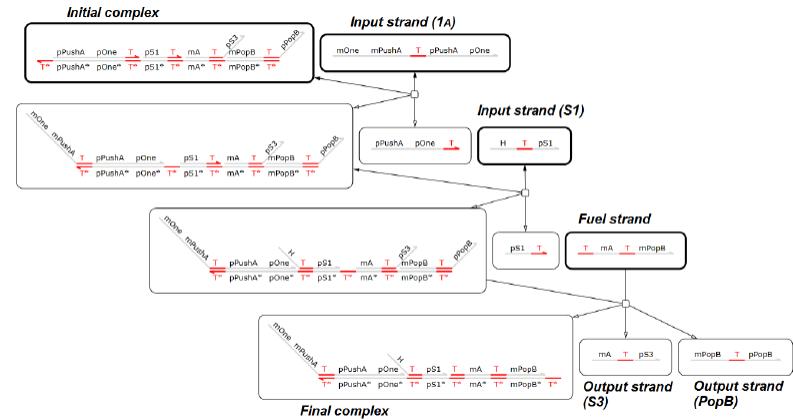
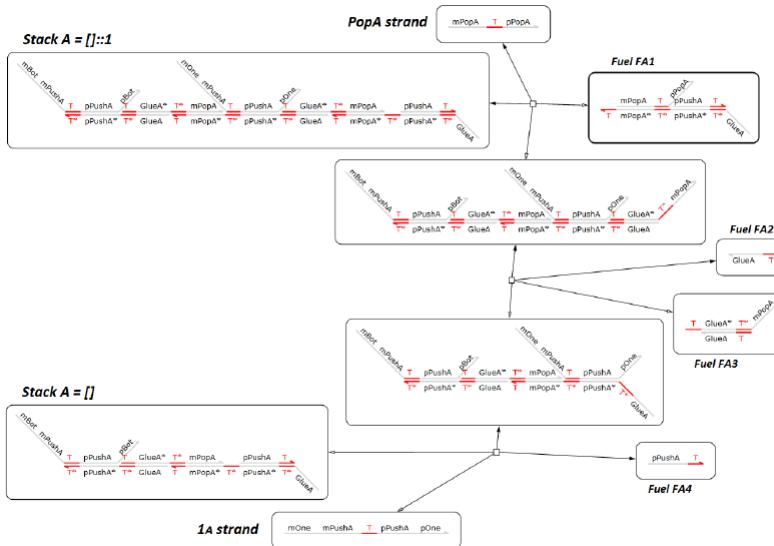
Luca
Cardelli



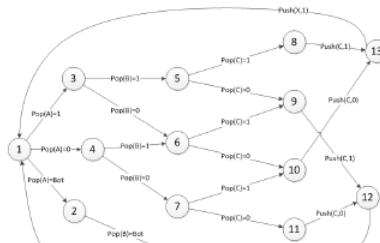
Stephen
Emmott

A DNA Stack Machine

Encoding state transitions



Encoding a Stack



Model-Checking a DNA Ripple Carry Adder

Input A MSB LSB Value	Input B MSB LSB Value	Output X LSB MSB Value	Output C Value	Result Value
0 0 0	0 0 0	0 0	0	0
0 0 0	0 1 1	1 0	0	1
0 0 0	1 0 2	0 1	0	2
0 0 0	1 1 3	1 1	0	3
0 1 1	0 0 0	1 0	0	1
0 1 1	0 1 1	0 1	0	2
0 1 1	1 0 2	1 1	0	3
0 1 1	1 1 3	0 0	1	4
1 0 2	0 0 0	0 1	0	2
1 0 2	0 1 1	1 1	0	3
1 0 2	1 0 2	0 0	1	4
1 0 2	1 1 3	1 0	1	5
1 1 3	0 0 0	1 1	0	3
1 1 3	0 1 1	0 0	1	4
1 1 3	1 0 2	1 0	1	5
1 1 3	1 1 3	0 1	1	6