# Speaking the Language of Molecules

**Luca Cardelli**
Microsoft Research

**Dublin 2012-05-25**
**http://lucacardelli.name**

# Outline

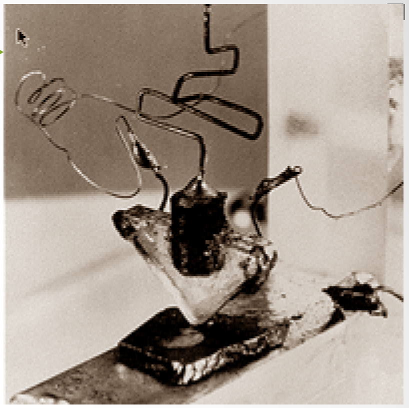
- Molecular Structures
    - In Science and Engineering
    - Self-assembly
- Molecular Languages
    - Natural languages: proteins, genes, membranes
    - Modeling languages (systems biology)
    - Executable languages (nano-engineering)
- Molecular Computation
    - Molecular Programming
    - Molecular Compilation

# Molecular Structures

# Smaller and Smaller

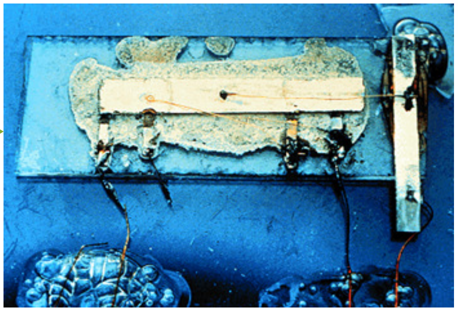
First working transistor

John Bardeen and Walter Brattain , Dec. 23, 1947.



First integrated circuit

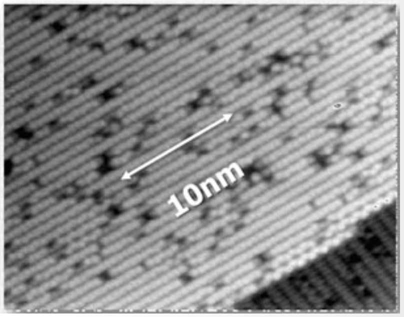
Jack Kilby, Sep. 1958.



50 years later

25nm NAND flash

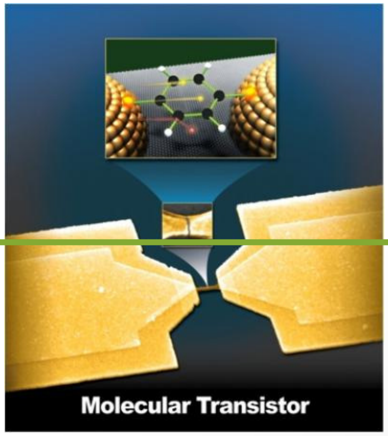
Intel&Micron, Jan. 2010. ~50atoms.



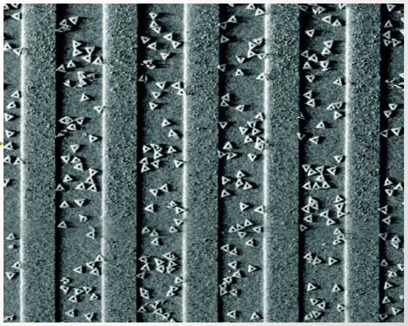
Scanning tunneling microscope image of a silicon surface showing 10nm is ~20 atoms across

Single molecule transistor

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



Molecules on a chip

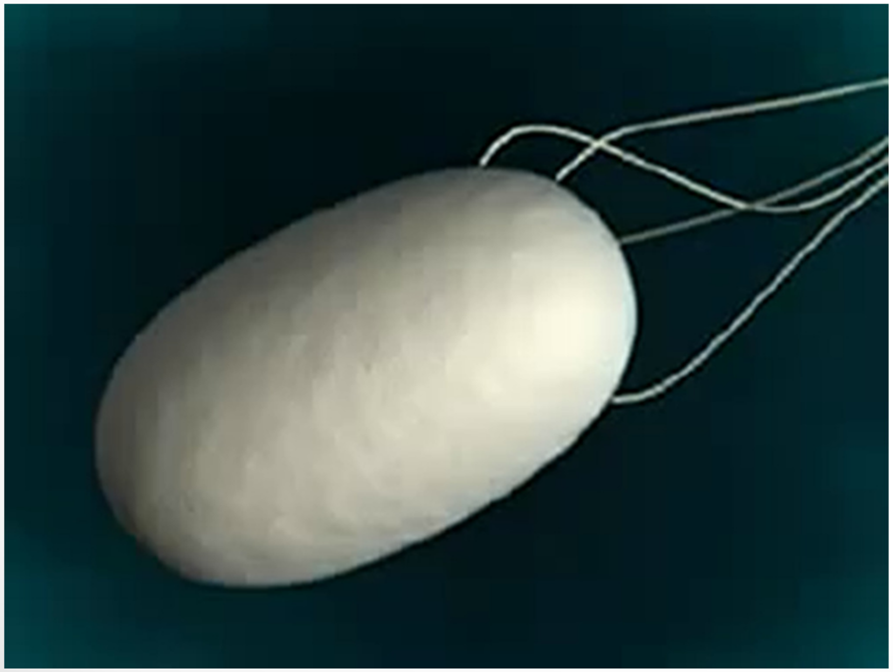


Placement and orientation of individual DNA shapes on lithographically patterned surfaces. Nature Nanotechnology 4, 557 – 561 (2009).

~10 Moore's Law cycles left!

# Building The Small*est* 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*.

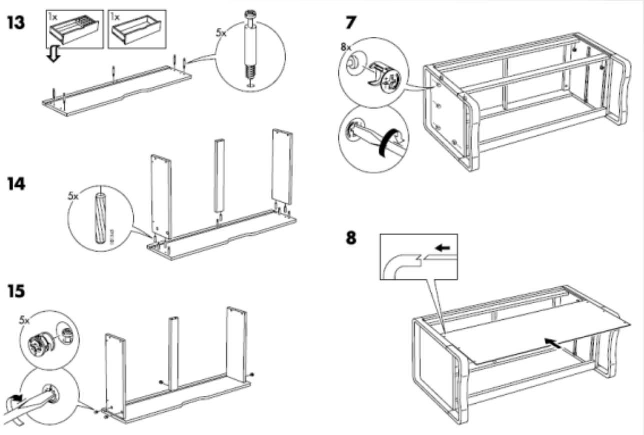


Garland Science
Taylor & Francis Group

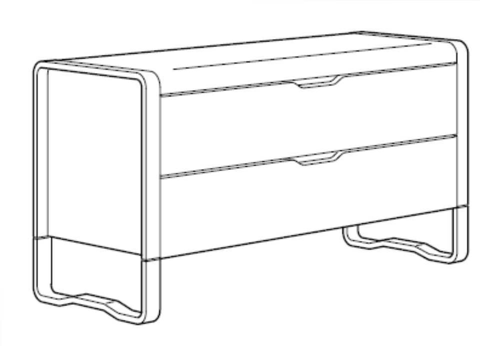
www.youtube.com/watch?v=Ey7Emmddf7Y

# 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…



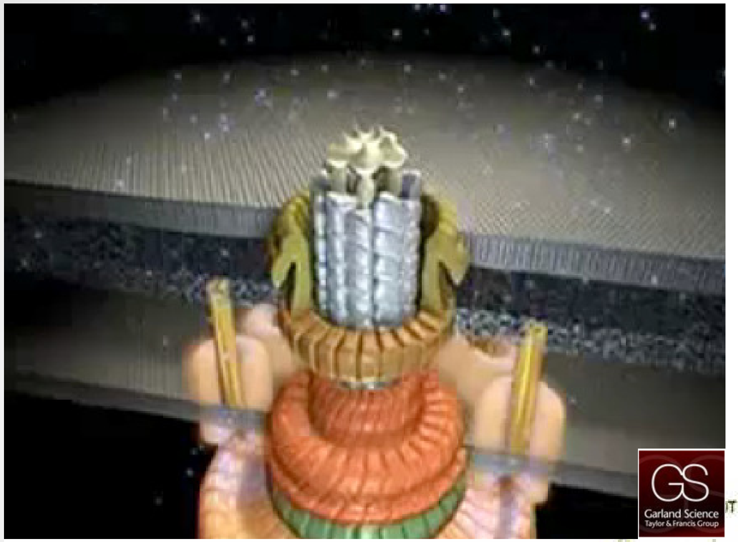
Add water



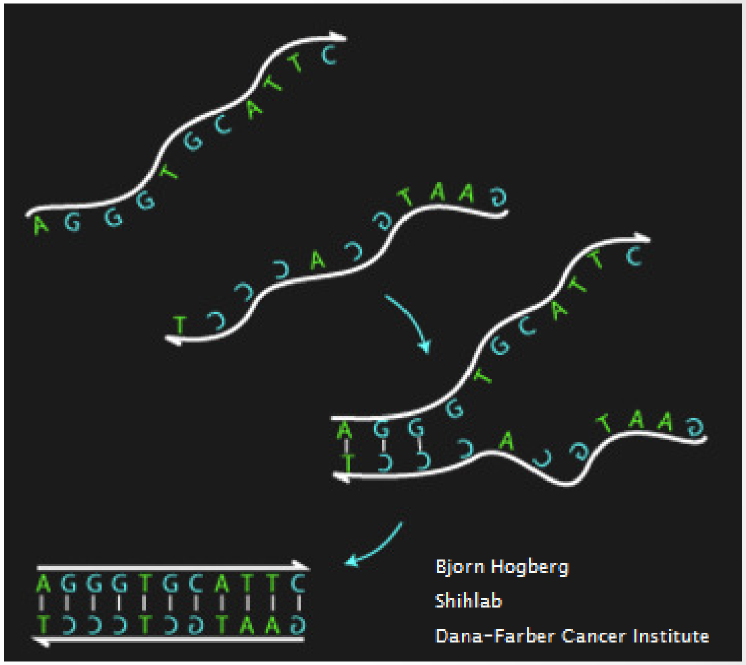
http://www.ikea.com/ms/en_US/customer_service/assembly_instructions.html

# Programmed Self-Assembly

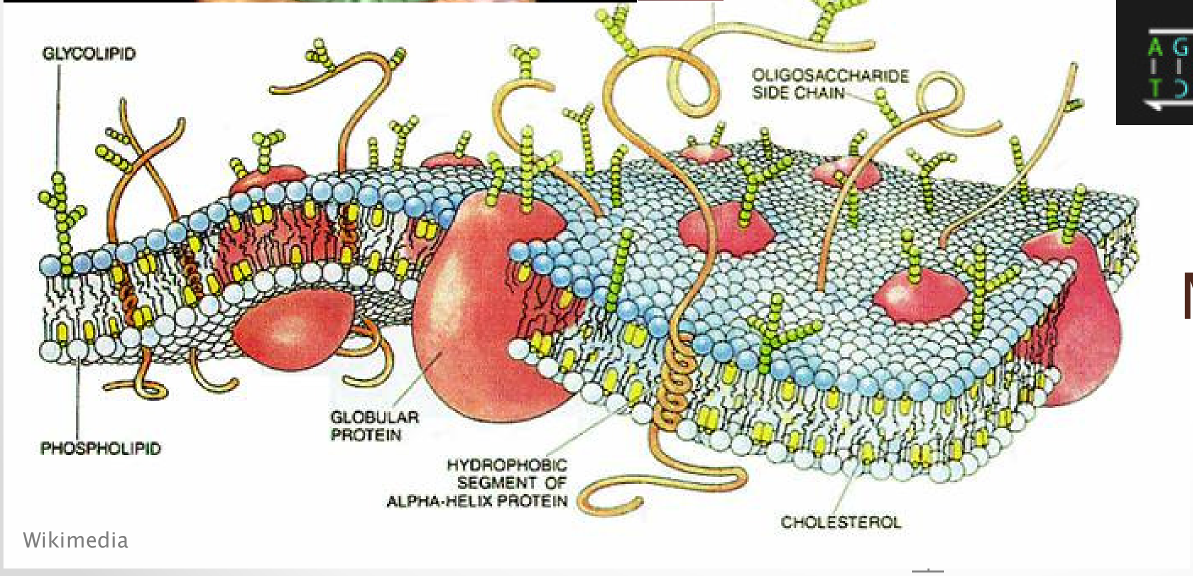
## Proteins



## DNA/RNA



Bjorn Hogberg
Shihlab
Dana-Farber Cancer Institute



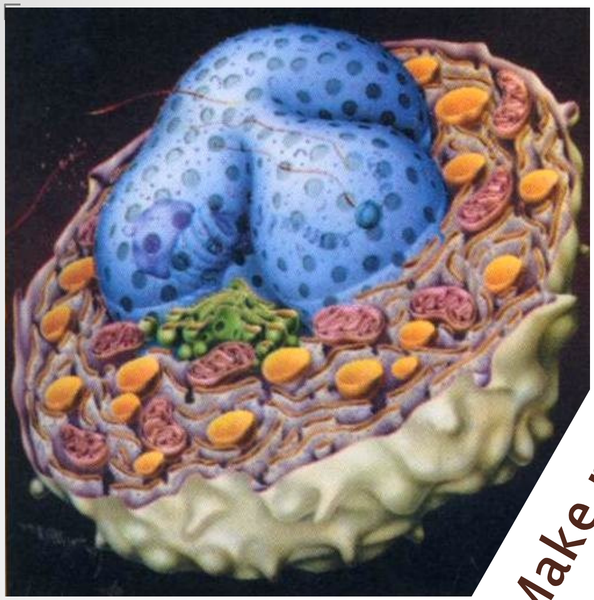
Wikimedia

## Membranes

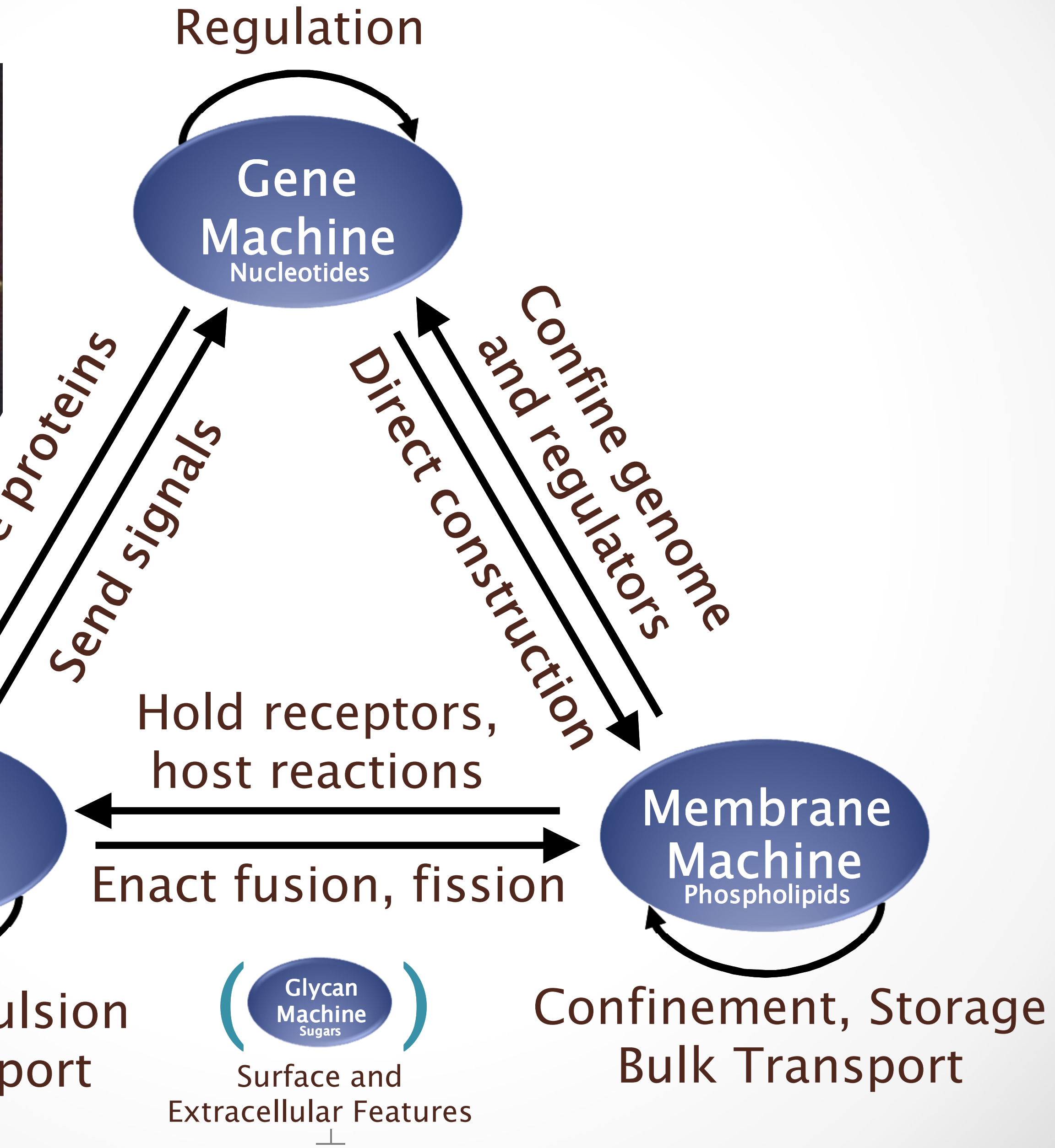
# Molecular Languages

## - natural languages -

# Abstract Machines of Biochemistry



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



Regulation

**Gene Machine**
Nucleotides

Make proteins

Send signals

Direct construction

Confine genome and regulators

Hold receptors, host reactions

Enact fusion, fission

**Protein Machine**
Aminoacids

**Membrane Machine**
Phospholipids

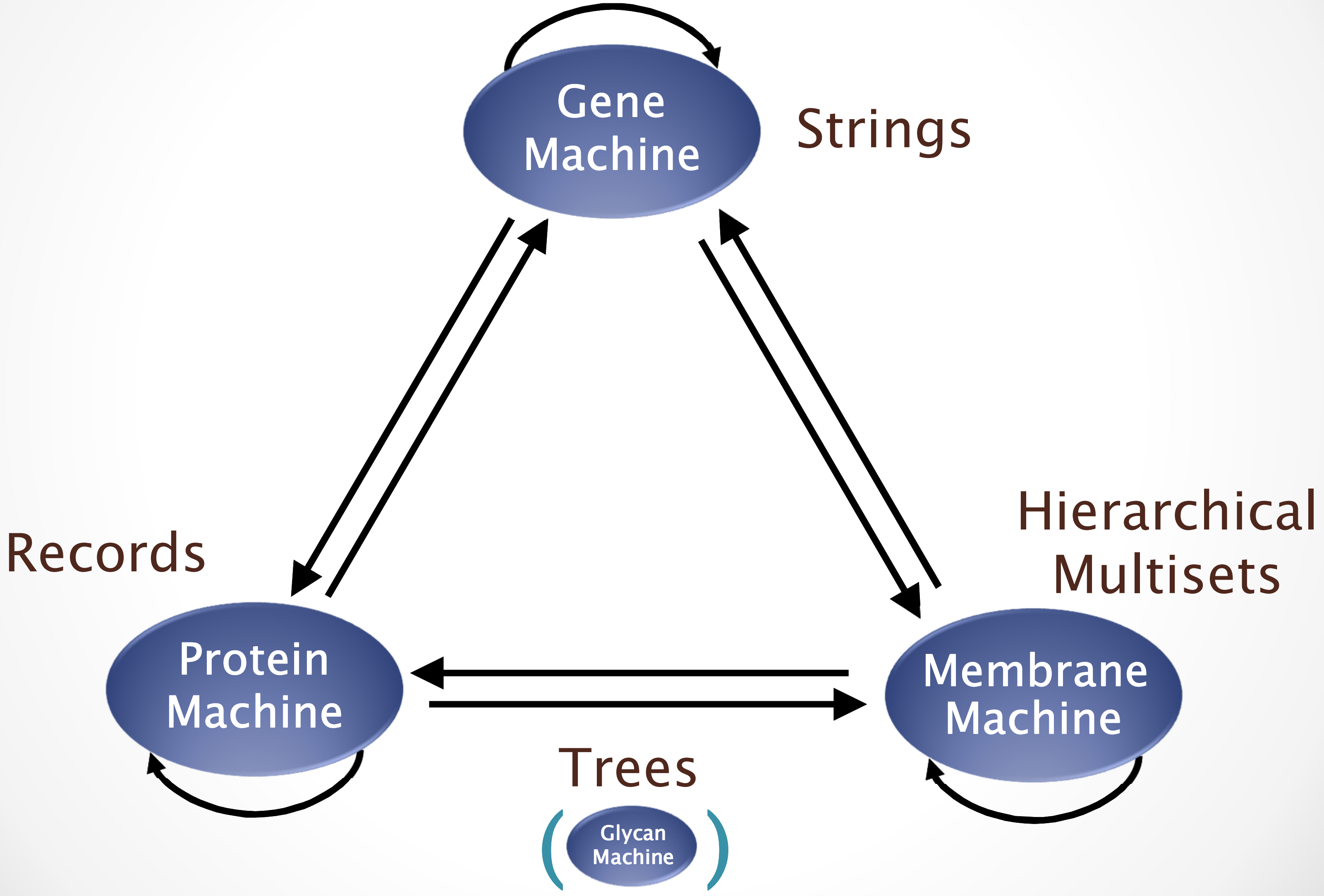
Metabolism, Propulsion
Signaling, Transport

**Glycan Machine**
Sugars

Surface and
Extracellular Features

Confinement, Storage
Bulk Transport

# Bioinformatics (Data Structures)



Gene Machine

Strings

Records

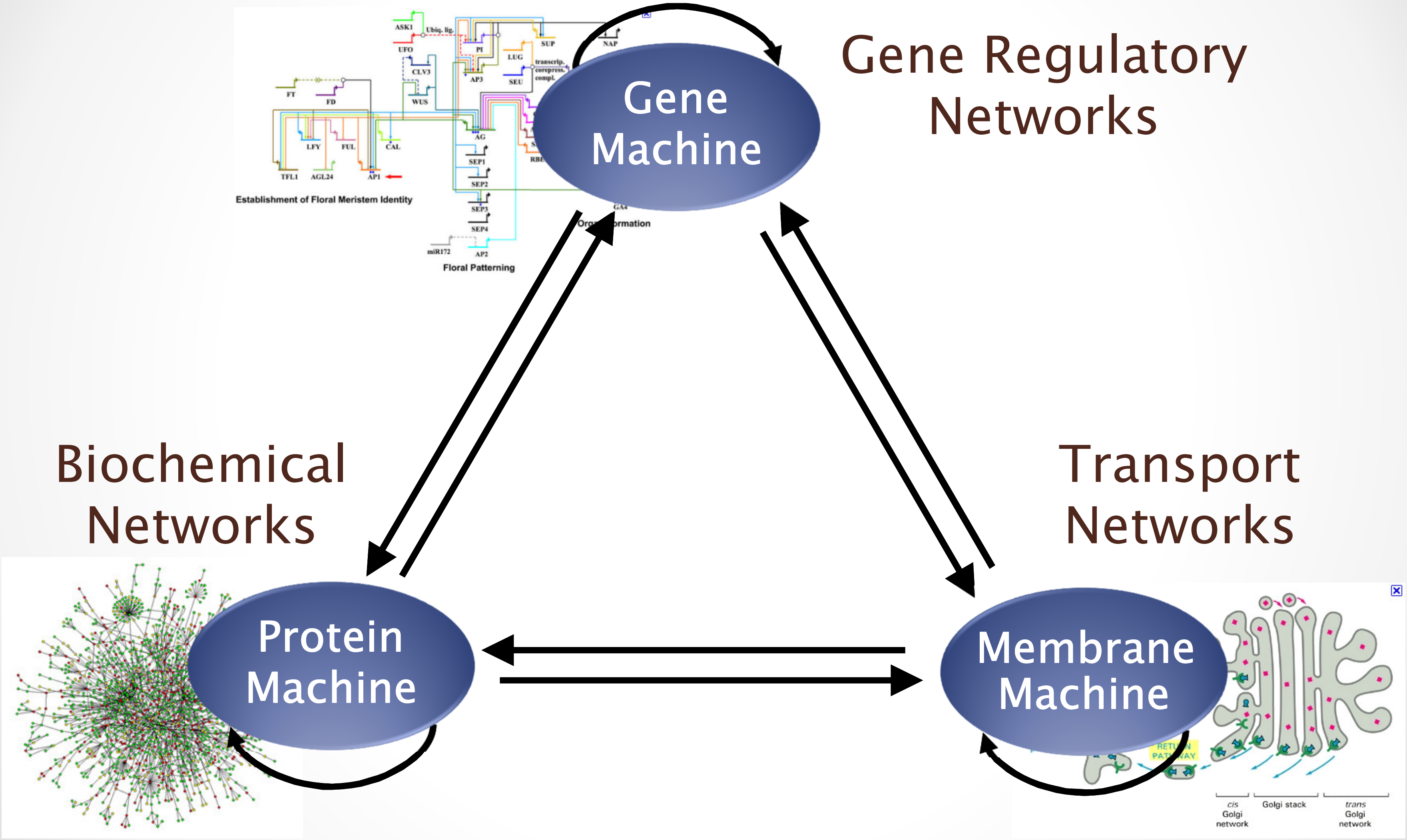
Protein Machine

Hierarchical Multisets

Membrane Machine

Trees

# Systems Biology (Networks)



Gene Machine

Gene Regulatory Networks

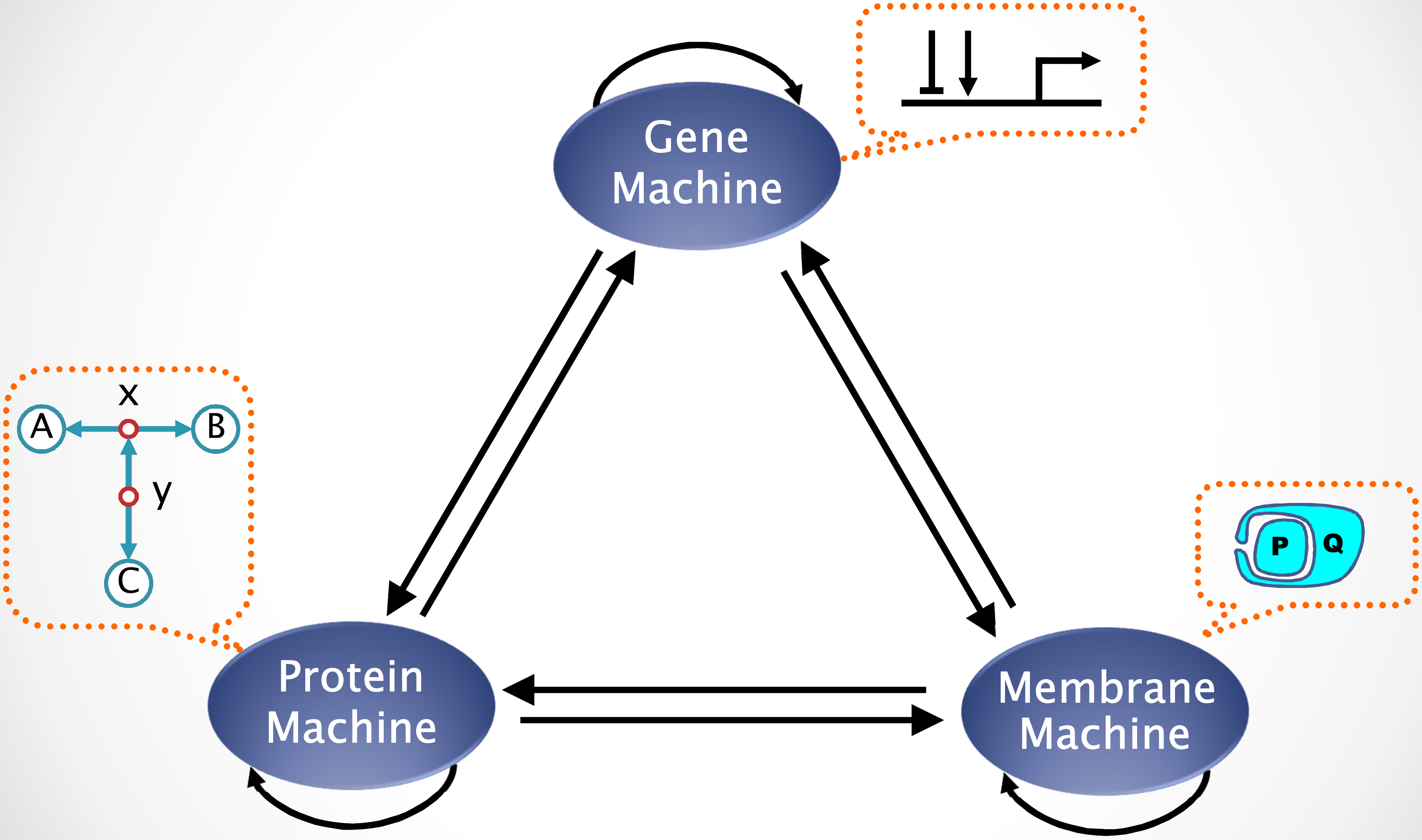
Protein Machine

Biochemical Networks

Membrane Machine

Transport Networks

# Computation (Languages)



Gene Machine

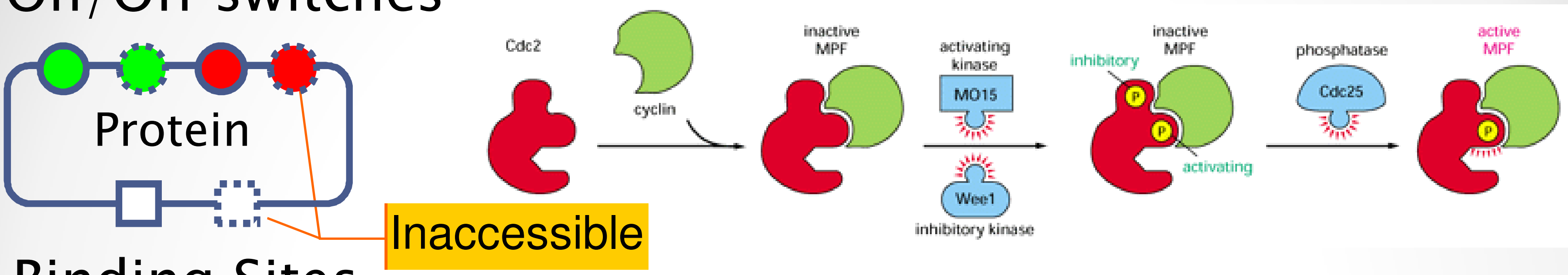
Protein Machine

Membrane Machine

# The Protein Machine

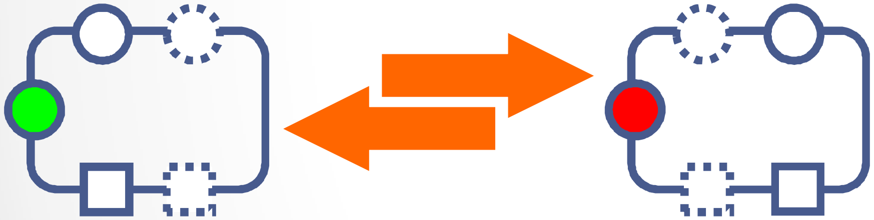
*cf.* BioCalculus [Kitano&Nagasaki], κ–calculus [Danos&Laneve]

## On/Off switches



Binding Sites

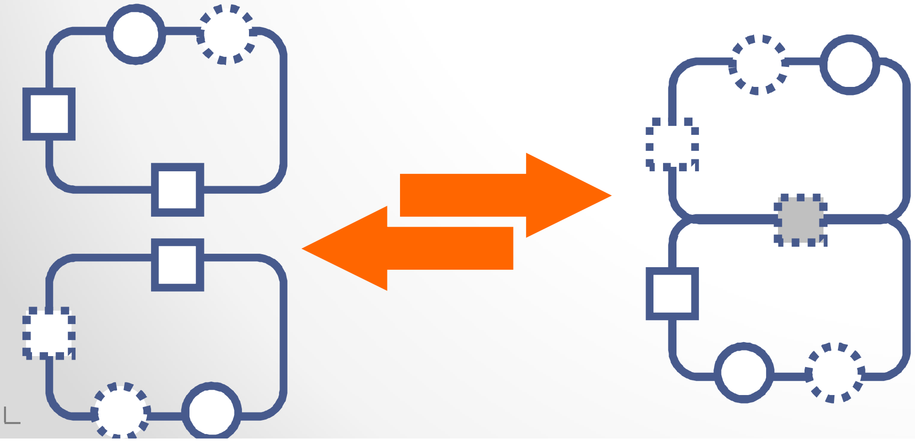
---



## Switching accessible switches

– May cause other switches and binding sites to become (in)accessible.

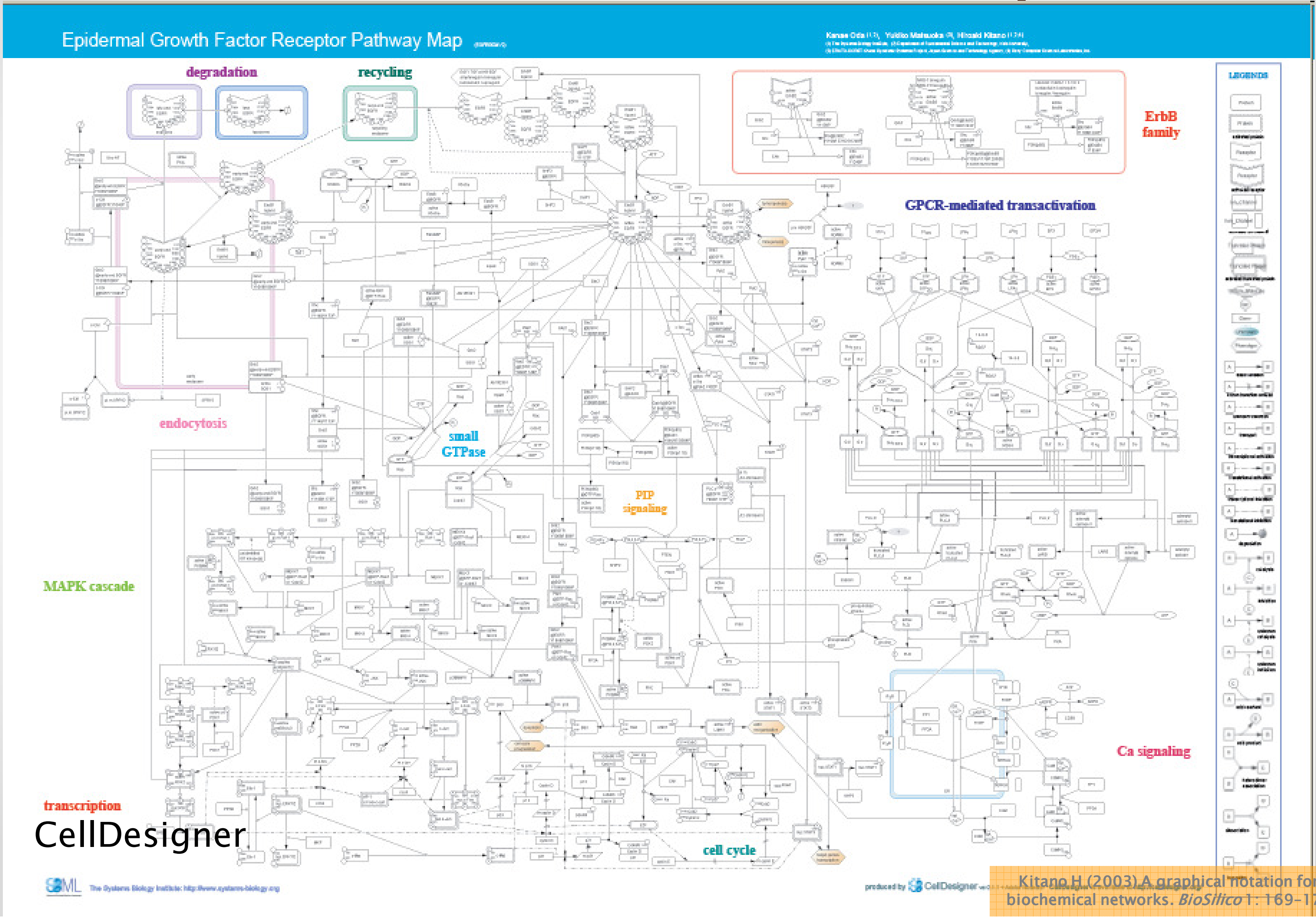
---



## Binding accessible sites

– May cause other switches and binding sites to become (in)accessible.

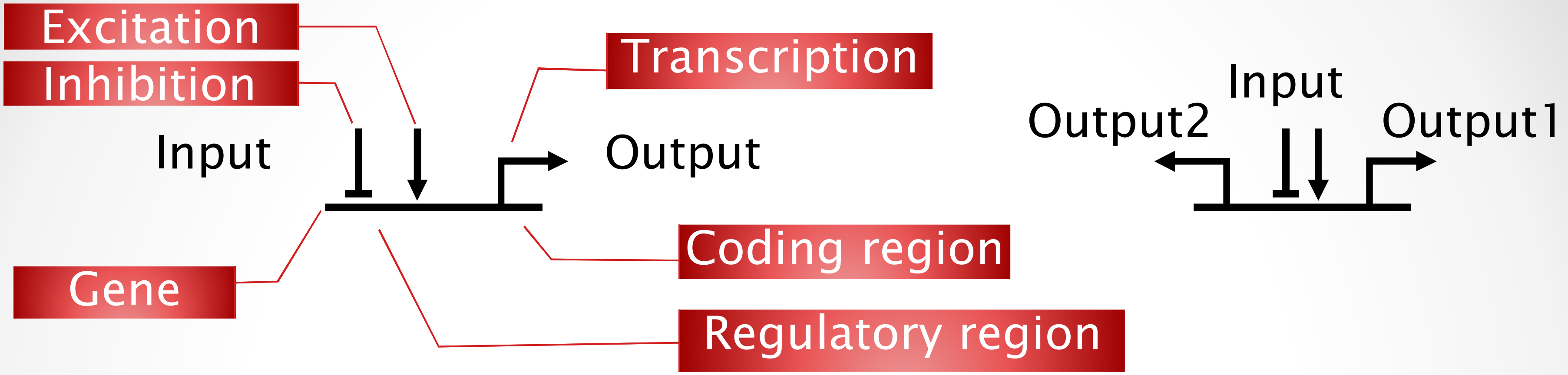
# Molecular Interaction Maps (Kohn/Kitano)



CellDesigner

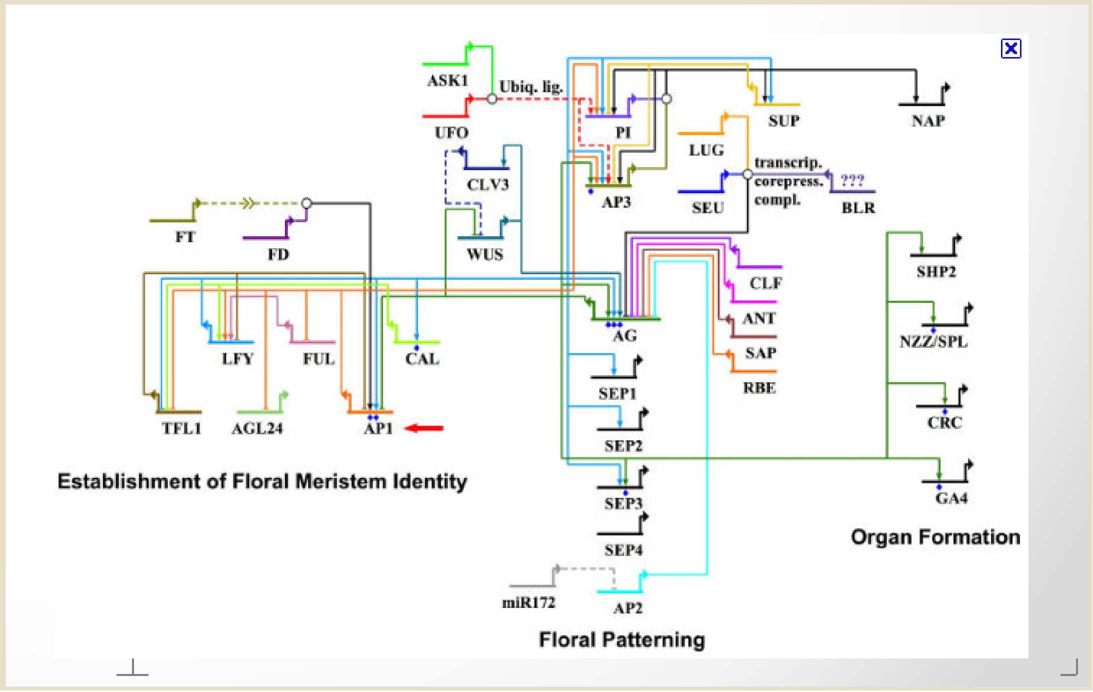
Kitano H (2003) A graphical notation for biochemical networks. *BioSilico* 1: 169-176

# The Gene Machine

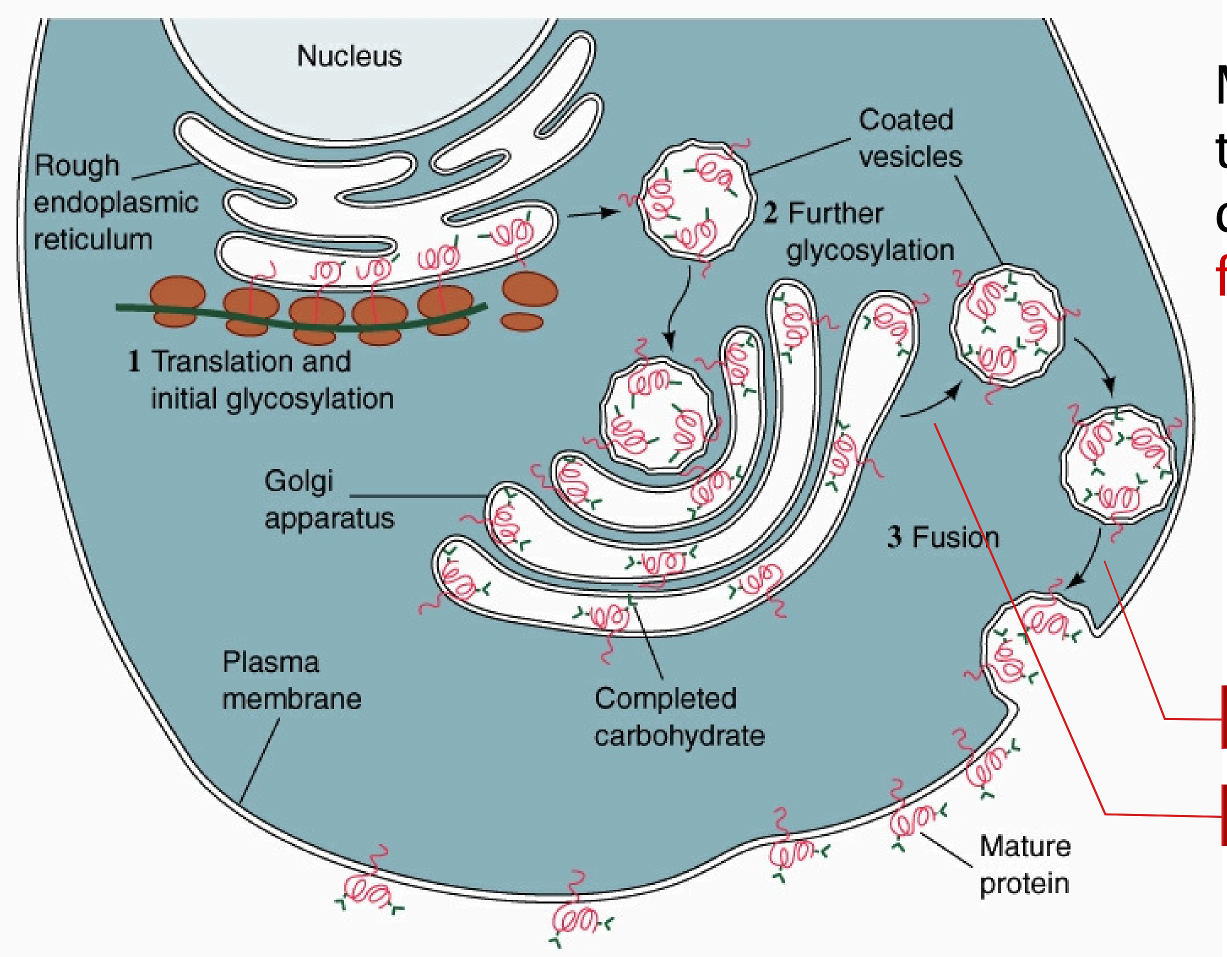


Regulation of a gene influences transcription. The regulatory region has precise DNA sequences meant for binding regulators.

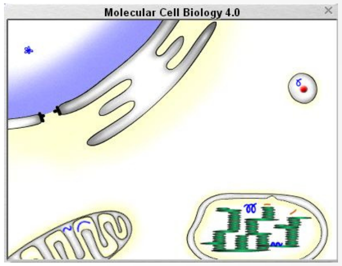
Transcription produces molecules (RNA or, through RNA, proteins) that bind to regulatory region of other genes (or that are end-products).



# The Membrane Machine



Molecular transport and transformation through dynamic compartment fusion and fission.



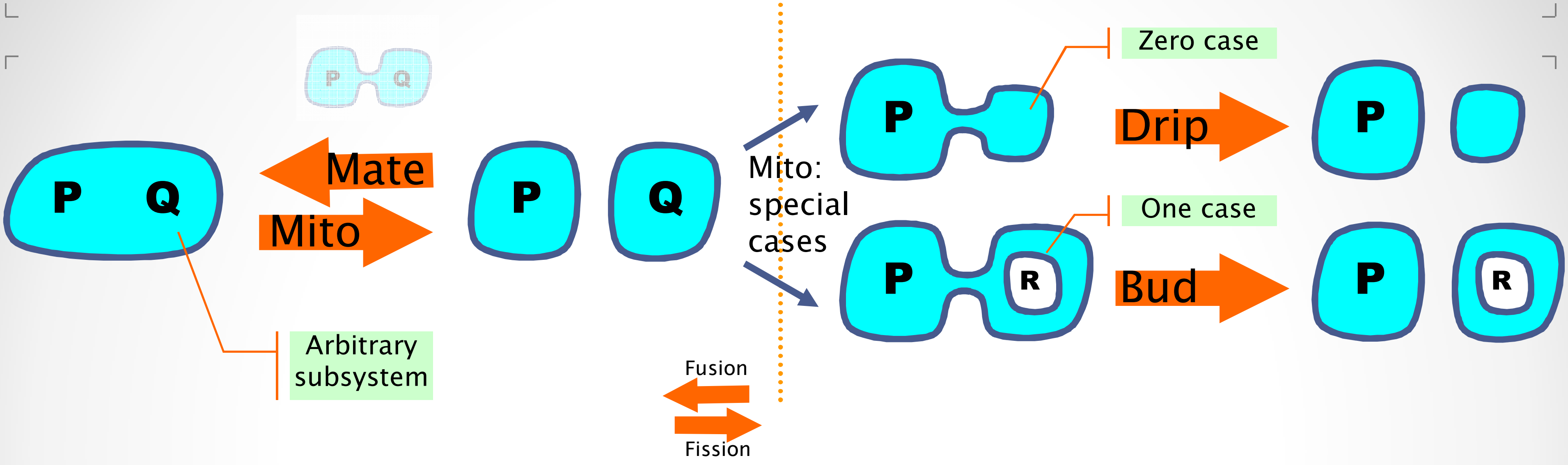
Taken from MCB CD

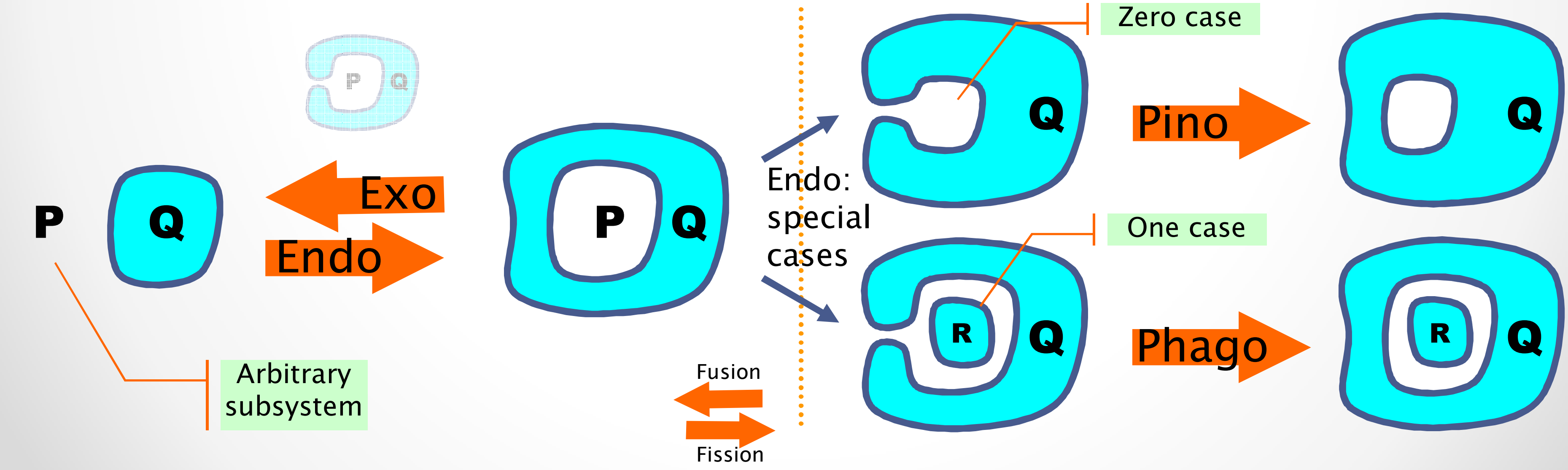
**Fusion**

**Fission**

Voet, Voet & Pratt
Fundamentals of Biochemistry
Wiley 1999. Ch10 Fig 10-22.

# Bitonal Diagrams





# Molecular Languages

## - modeling languages -

# From Instructions to Programs

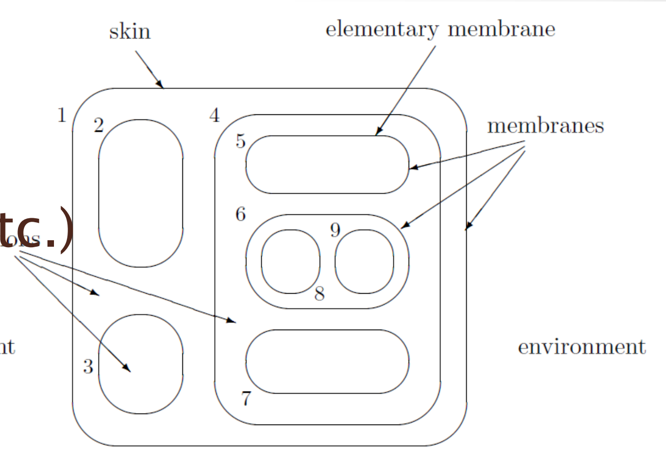
- We have seen the instruction sets:
  - Proteins – complexation, phosphorilation
  - Genes – activation, inhibition
  - Membranes – fusion, fission
- How do we combine them into programs?
  - I.e., into models (quantitative programs)
- How do we study their semantics?
  - I.e., their kinetics (quantitative semantics)

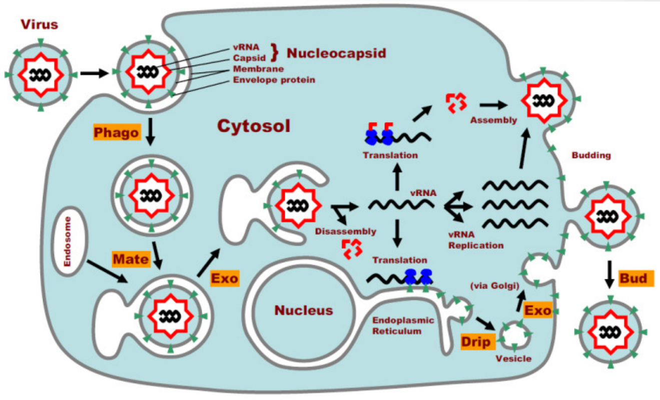
# Chemistry

- Chemical reactions
    - o $A + B \rightarrow_r C + D$ (a program)
- Ordinary Differential Equations
    - o $d[A]/dt = -r[A][B]$ … (a semantics)
- Rich analytical techniques based on Calculus
- But prone to combinatorial explosion
    - o E.g., due to the peculiarities of protein interactions

# 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.



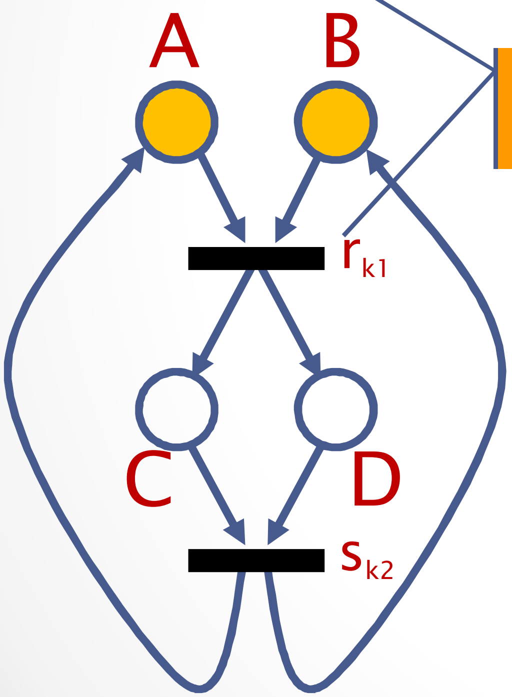


# Reactions vs. Reagents

Says what "A" *does*.

r: $A + B \rightarrow_{k1} C + D$
s: $C + D \rightarrow_{k2} A + B$

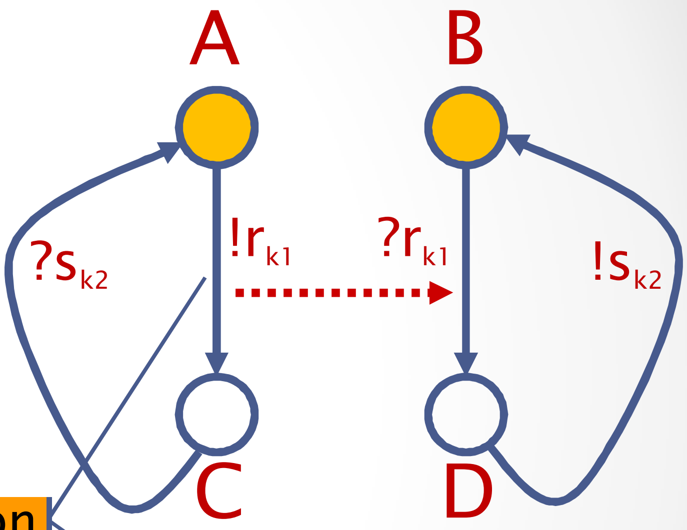
Does A become C or D?



**Reaction oriented**

1 line per reaction

Says what "A" *is*.



**Interaction oriented**

1 line per agent

$A = !r_{k1};\ C$
$C = ?s_{k2};\ A$
$B = ?r_{k1};\ D$
$D = !s_{k2};\ B$

A becomes C not D!

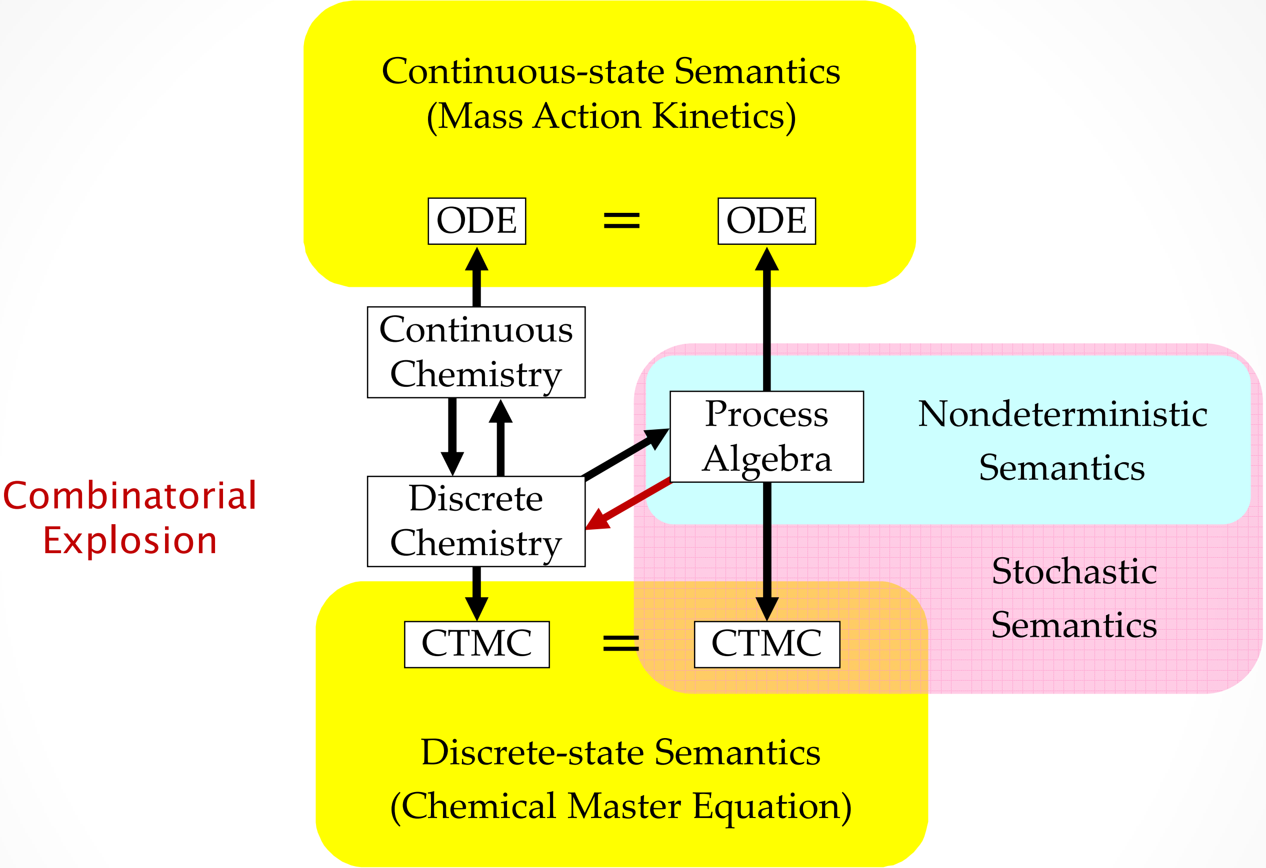
The same "math model"

CTMC

# Molecular Languages

- Reaction-Based (A + B → C + D) (Chemistry)
    - Limited to finite set of species (no polymerization)
    - Practically limited to small number of species (no run-away complexation)
- Interaction-Based (A = !r; C) (Process Algebra)
    - Reduces combinatorial complexity of models by combining independent submodels connected by interactions.
- Rule-Based (A{-}:B{p} → A{p}:B{-}) (Logic, Graph Rewriting)
    - Further reduces model complexity by describing molecular state, and by allowing one to 'ignore the context': a *rule* is a reaction in an unspecified (complexation/phosphorylatio) context.
    - Similar to informal descriptions of biochemical events ("narratives").
- Syntactic connections
    - The latter two can be translated (to each other and) to the first, but doing so may introduce an infinite, or anyway *extremely large*, number of species.

# Semantic Connections



These diagrams commute via appropriate maps.

L. Cardelli: “On Process Rate Semantics” (TCS)

L. Cardelli: “A Process Algebra Master Equation” (QEST’07)

# But what about Execution?

- Chemistry is not easily executable
    - Please Mr Chemist, execute me these reactions that I just made up

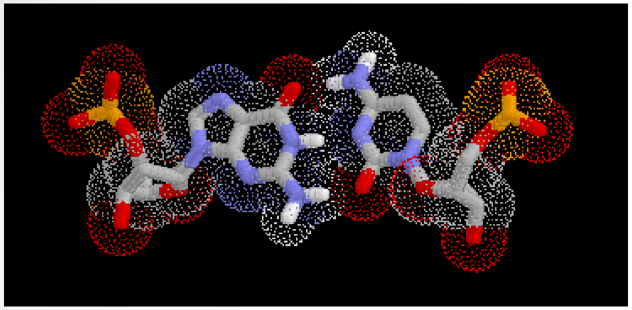
- Similarly, the molecular languages seen so fare are descriptive (modeling) languages

- How can we actually execute molecular languages? With real molecules?

# Molecular Languages

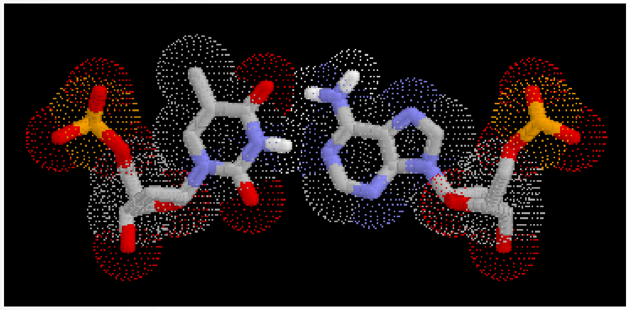
## - executable languages -

# DNA



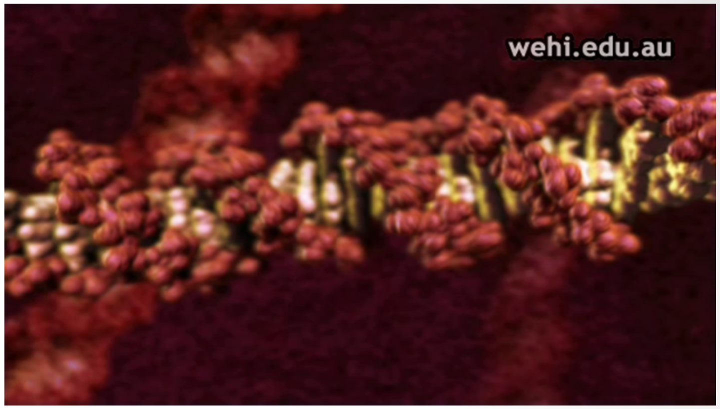
**GC Base Pair**

Guanine–Cytosine



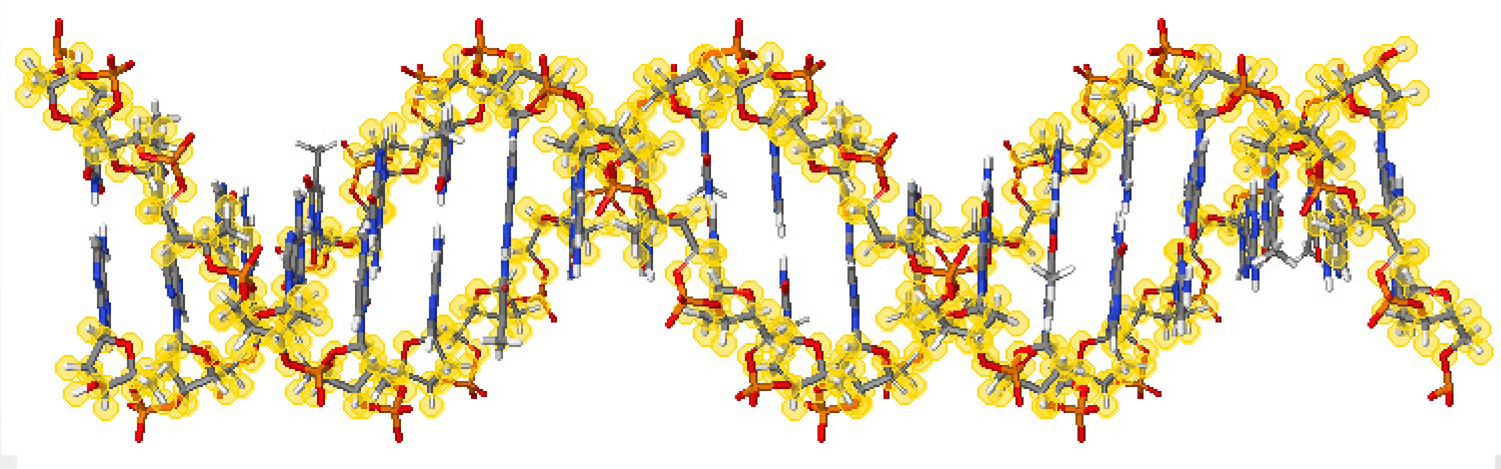
**TA Base Pair**

Thymine–Adenine



Interactive DNA Tutorial

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



Sequence of Base Pairs (GACT alphabet)

# Robust, and *Long*

- DNA in each human cell:
  - 3 billion base pairs
  - 2 meters long, 2nm thick
  - folded into a 6μm ball
  - 750 MegaBytes
- A huge amount for a cell
  - Every time a cell replicates it has tc copy *2 meters of DNA* reliably.
  - To get a feeling for the scale disparity, compute:
- DNA in human body
  - 10 trillion cells
  - 133 Astronomical Units long
  - 7.5 OctaBytes
- DNA in human population
  - 20 million light years long

DNA wrapping into chromosomes

wehi.edu.au

Andromeda Galaxy
2.5 million light years away

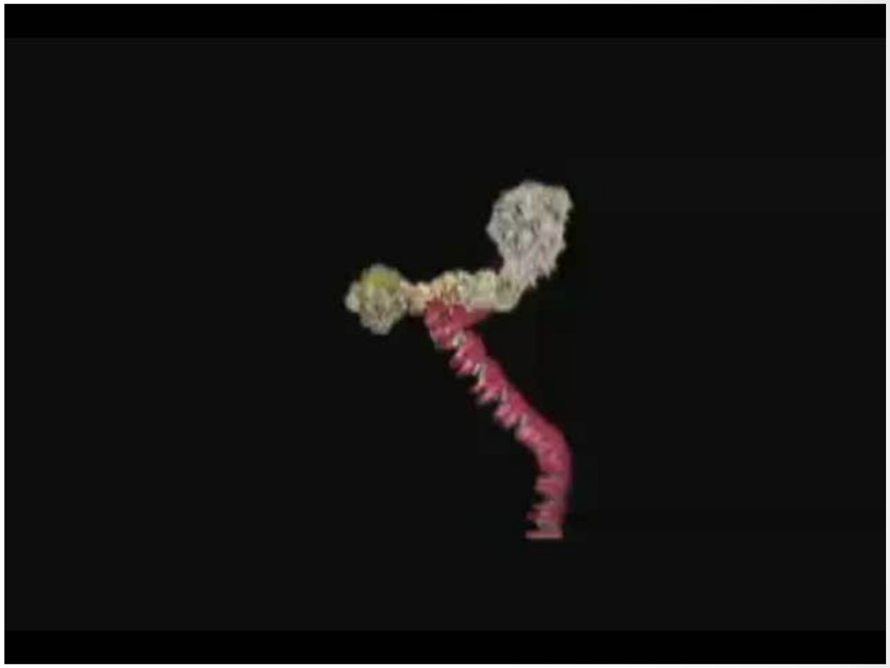
# Natural DNA Operation

- 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*

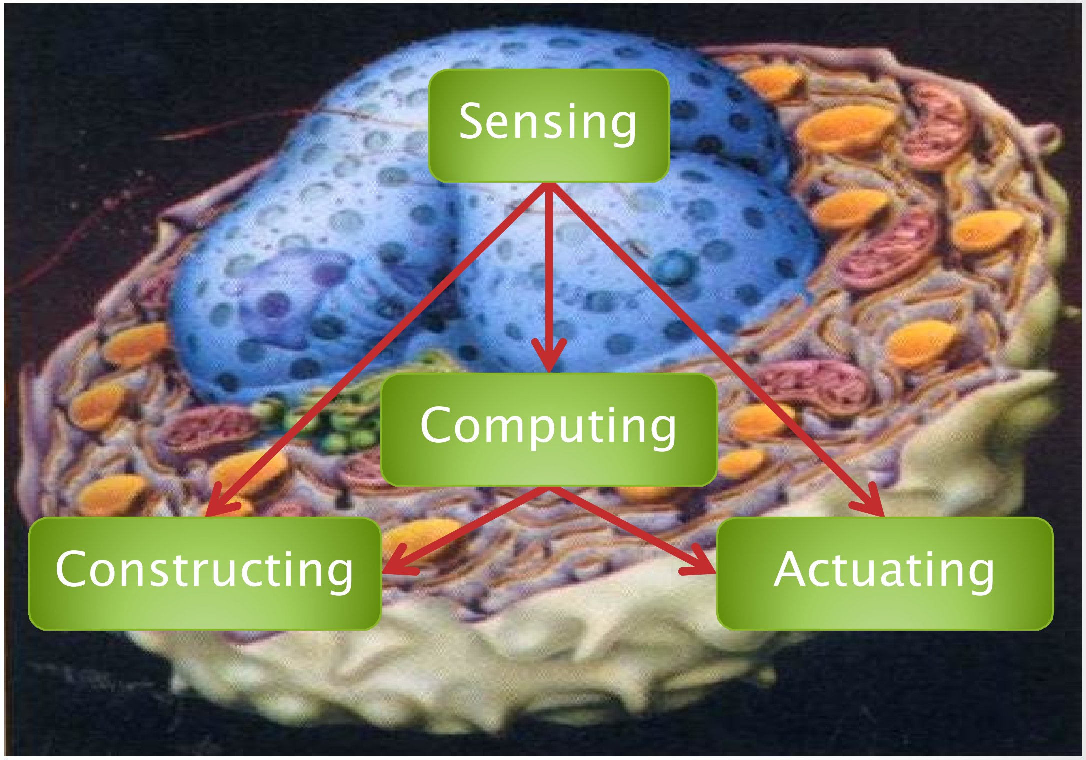
RNA polymerase II:
15-30 bases/second

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

# Unnatural DNA Operation

- Sensing
  - Reacting to forces
  - Binding to molecules
- Actuating
  - Releasing molecules
  - Producing forces
- Constructing
  - Chassis
  - Growth
- Computing
  - Signal Processing
  - Decision Making

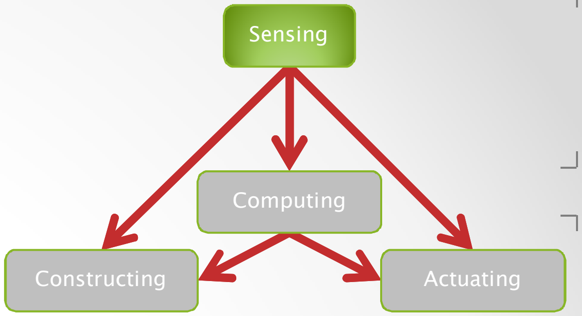
Nanoscale Control Systems



Nucleic Acids can do all this.

And interface to biology.

# Sensing

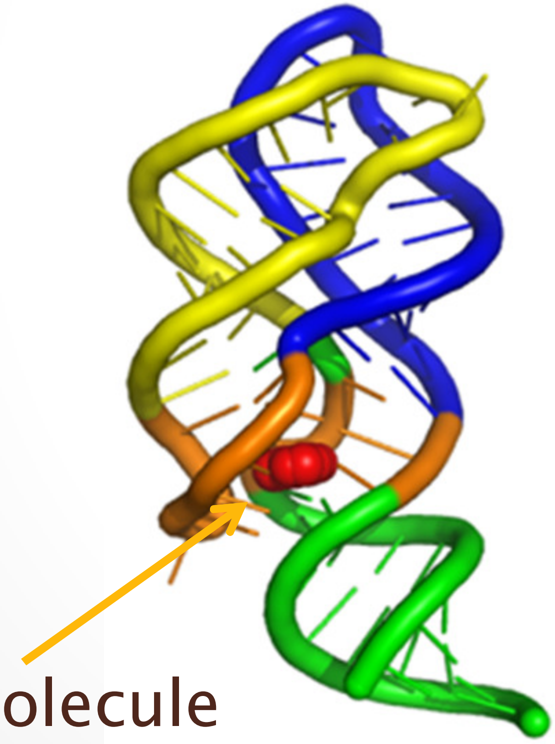

Sensing

Computing

Constructing

Actuating

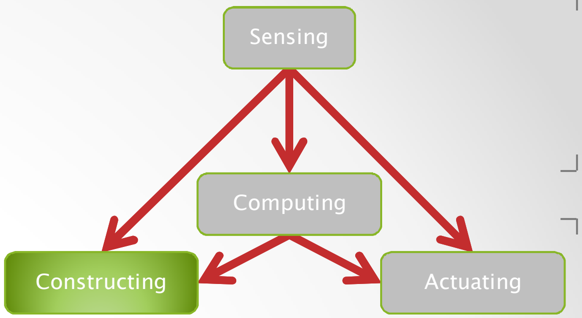
Aptamers: natural or artificially evolved DNA molecules that stick to other molecules (highly selectively).


Target molecule

## Adenine riboswitch aptamer

Structural basis for discriminative regulation of gene expression by adenine- and guanine-sensing mRNAs. Chem Biol. 2004 Dec;11(12):1729-41.

# Constructing

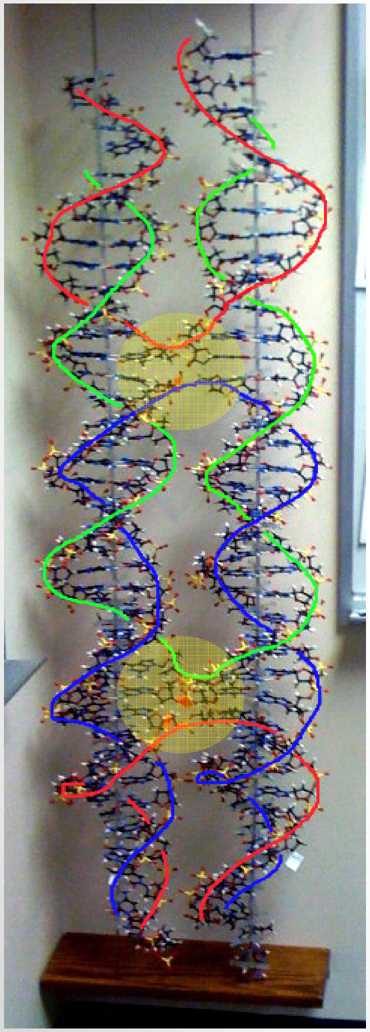

Sensing

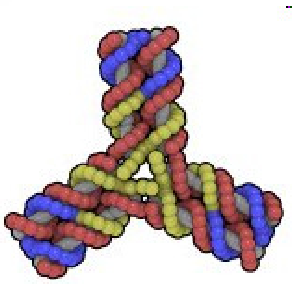
Computing

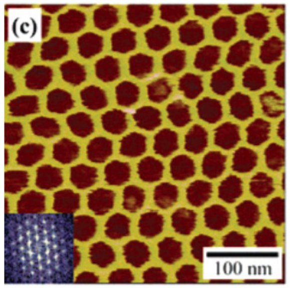
Constructing

Actuating

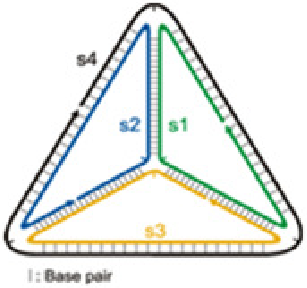
## Crosslinking

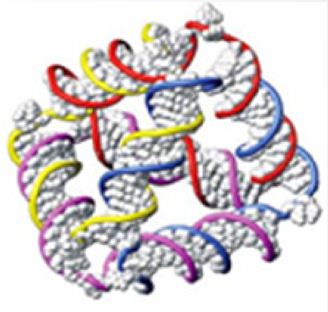






Chengde Mao, Purdue





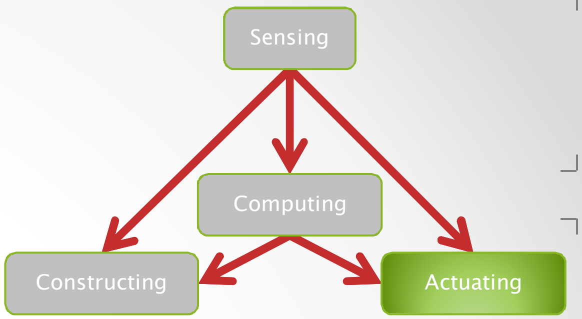
Andrew Turberfield, Oxford

**Folding DNA into Twisted and Curved Nanoscale Shapes**

Hendrik Dietz, Shawn M. Douglas, & William M. Shih

Science, 325:725–730, 7 August 2009.

# Actuating



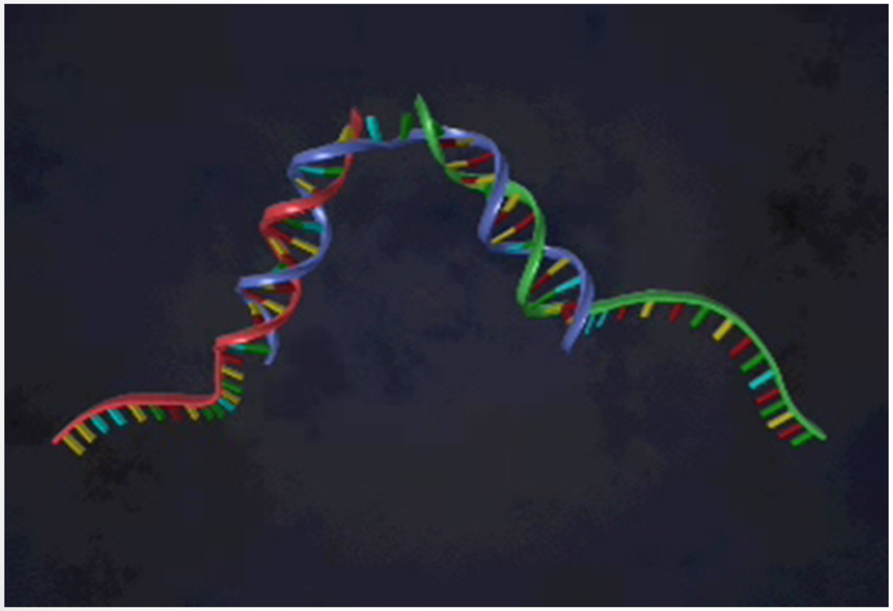
Sensing

Computing

Constructing

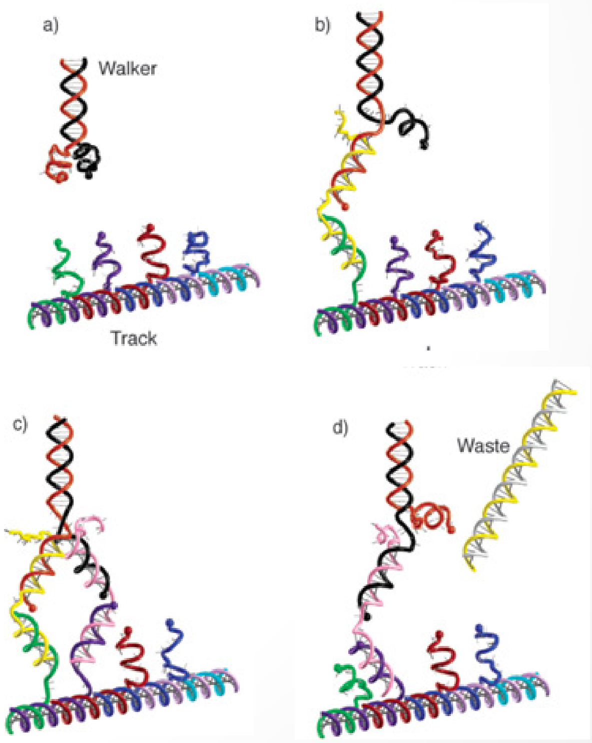
Actuating

## DNA tweezers

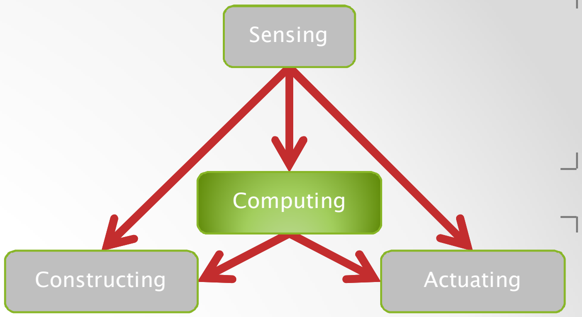


Bernard Yurke, Boise State

## DNA walkers



# Computing

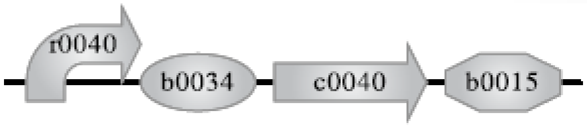


- Sensors and Actuators
  at the 'edge' of the system
  - They can use disparate technologies and phenomena
- Computation in the 'kernel' of the system
- Compositionality in the kernel
  - The components should use uniform inputs and outputs
  - The components should be ‘computationally complete’

# “Embedded” Computing

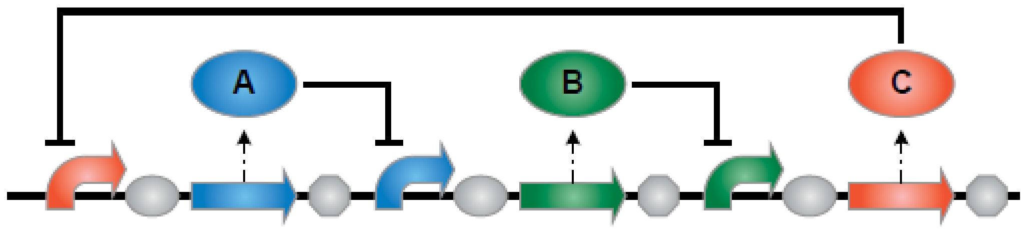
(Synthetic Biology)

- Using bacterial machinery (e.g.) as the hardware. Using embedded gene networks as the software.
- MIT Registry of Standard Biological Parts
- GenoCAD
  - Meaningful sequences [Cai et al.]



```
r0040:prom; b0034:rbs; c0040:pcr; b0015:ter
```

- GEC
  - [Pedersen & Phillips]



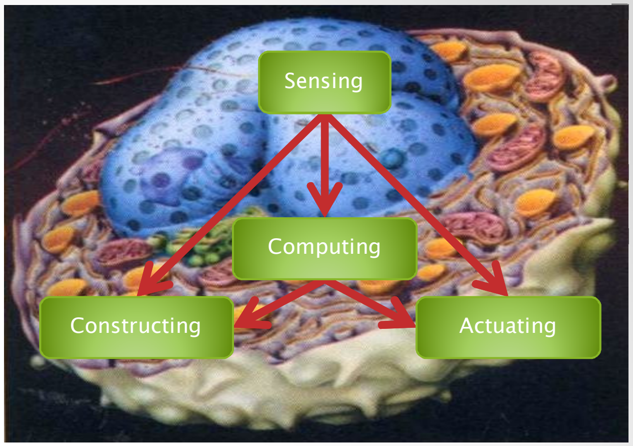
```
prom<neg(C)>; rbs; pcr<codes(A)>; ter;
prom<neg(A)>; rbs; pcr<codes(B)>; ter;
prom<neg(B)>; rbs; pcr<codes(C)>; ter
```

# “Autonomous” Computing

(Nano-engineering)

- Mix & go
  - All (or most) parts are synthesized
  - No manual cycling (cf. early DNA computing)
  - In some cases, all parts are made of DNA (no enzyme/proteins)
- Self-assembled and self-powered
  - Can run on its own (e.g. environmental sensing)
  - Or be embedded into organisms, but running ‘separately’

# Curing



## A doctor in each cell

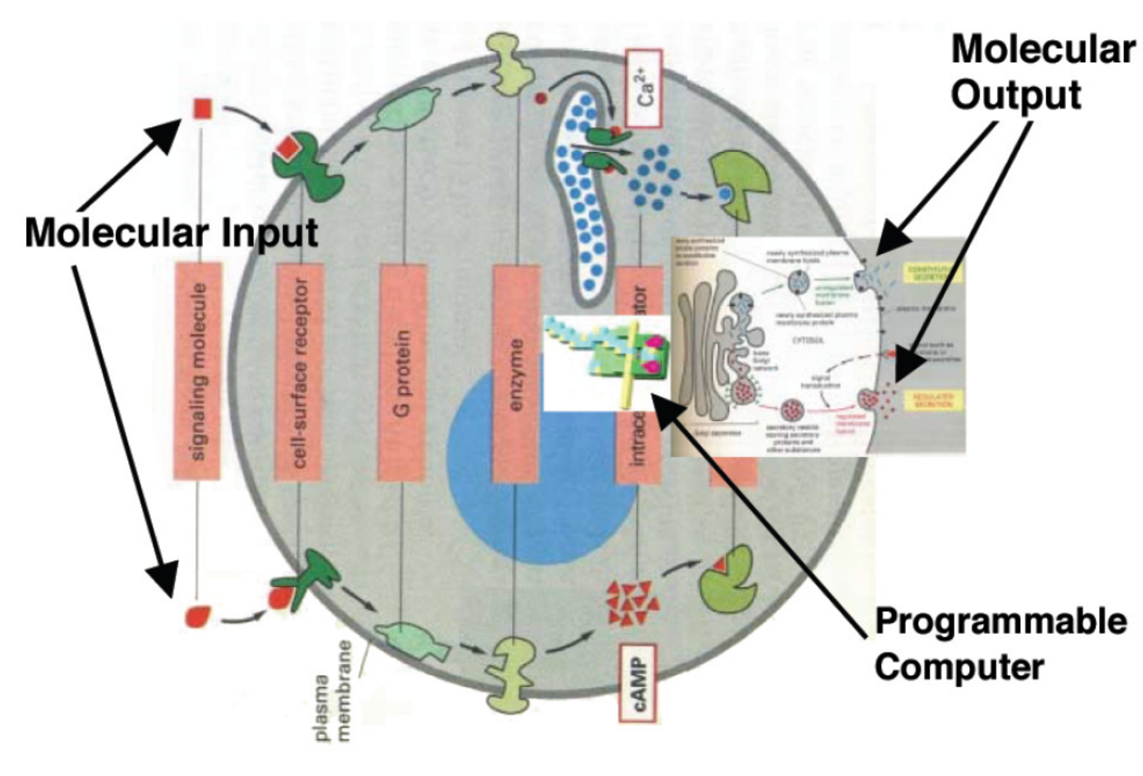


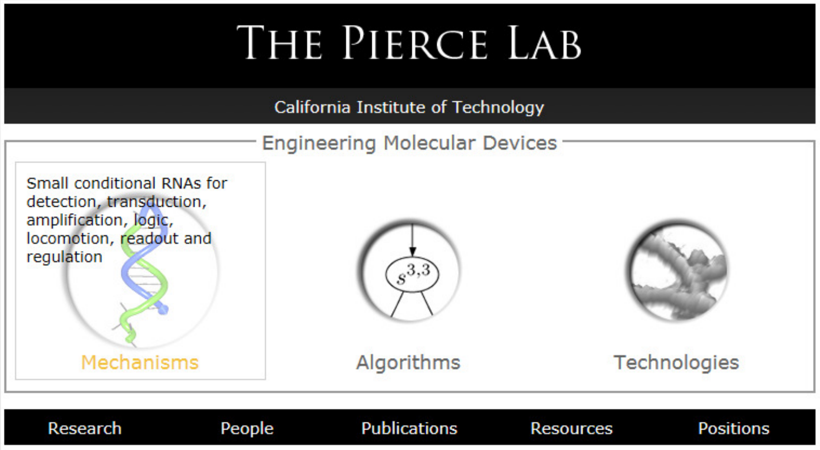
***Fig. 1*** *Medicine in 2050: "Doctor in a Cell"*

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

**Molecules and computation**

# RNA operation in (dead) cells

- Using RNA Hybridization Chain Reaction for imaging of mRNA expression.
  - The programmability of orthogonal RNA reactions enables spatial imaging with 5 simultaneous targets.

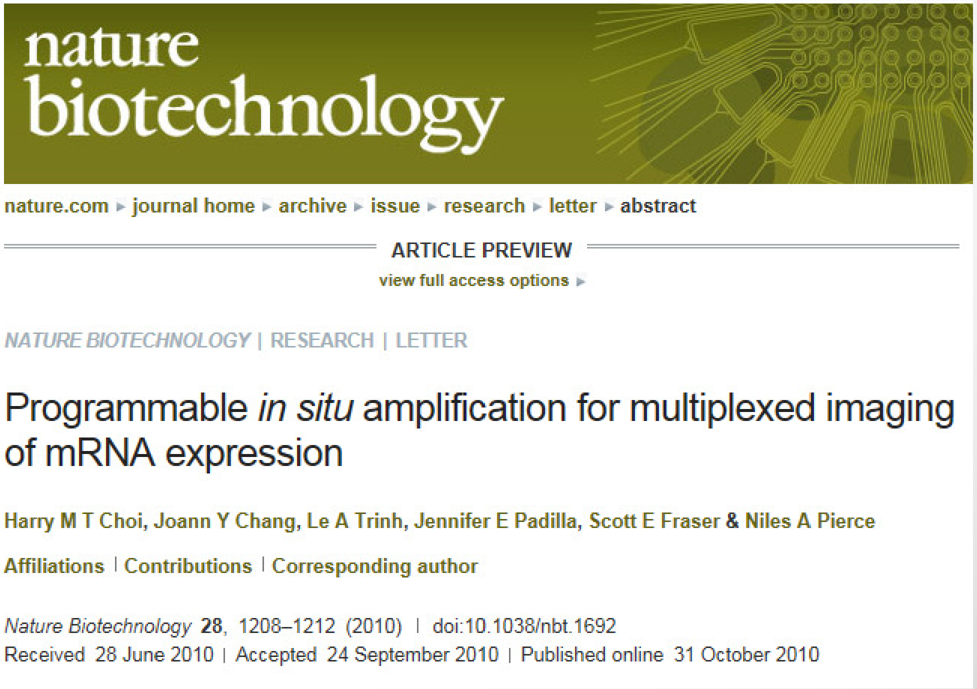


THE PIERCE LAB

California Institute of Technology

Engineering Molecular Devices

Small conditional RNAs for detection, transduction, amplification, logic, locomotion, readout and regulation



*NATURE BIOTECHNOLOGY* | RESEARCH | LETTER

Programmable *in situ* amplification for multiplexed imaging of mRNA expression

Harry M T Choi, Joann Y Chang, Le A Trinh, Jennifer E Padilla, Scott E Fraser & Niles A Pierce

Affiliations | Contributions | Corresponding author

*Nature Biotechnology* **28**, 1208–1212 (2010) | doi:10.1038/nbt.1692
Received 28 June 2010 | Accepted 24 September 2010 | Published online 31 October 2010

# Molecular Computation

# DNA Computing

- Non–goals
  - Not to solve NP–complete problems.
  - Not to replace electronics.
  - Not necessarily using genes or producing proteins.

- For general ‘molecular programming’
  - To precisely control the organization and dynamics of matter and information at the molecular level.
  - To interact algorithmically with biological entities.
  - The use of DNA is “accidental”: no genes involved.
  - In fact, no material of biological origin.

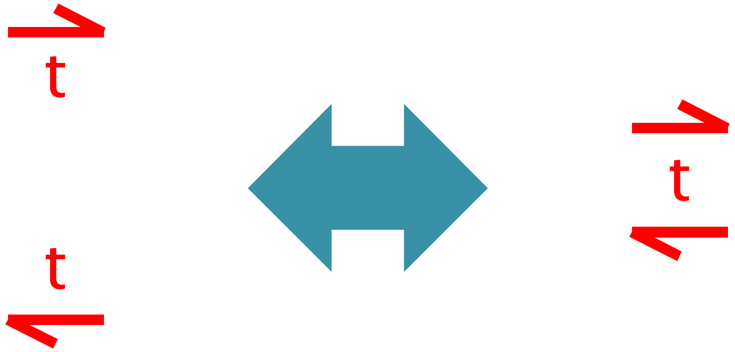
# Domains

- Subsequences on a DNA strand are called domains. *PROVIDED* they are "independent" of each other.

- I.e., 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.

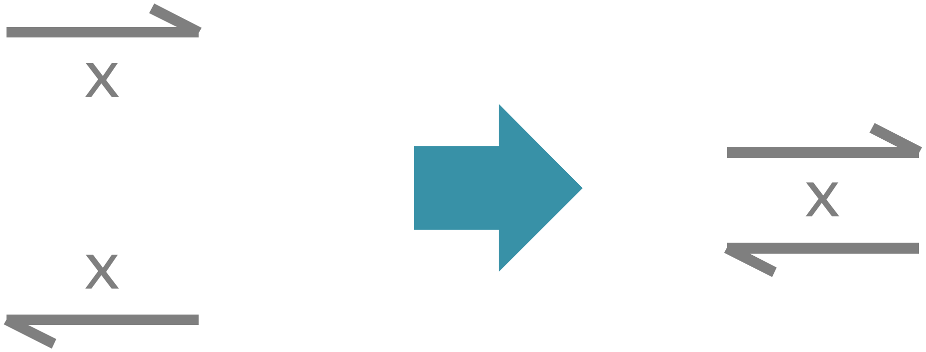
- Choosing domains (subsequences) that are suitably independent is a tricky issue that is still somewhat of an open problem (with a vast literature). But it can work in practice.

# Short Domains



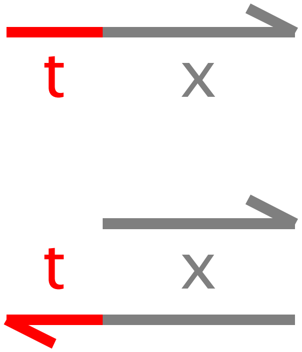
Reversible Hybridization

# Long Domains



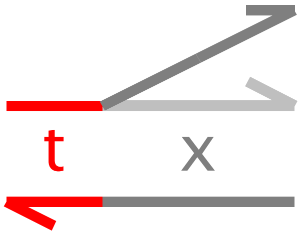
Irreversible Hybridization

# Strand Displacement



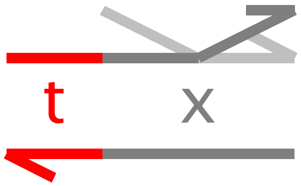
“Toehold Mediated”

# Strand Displacement



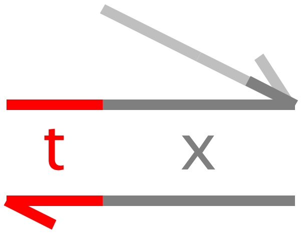
Toehold Binding

# Strand Displacement



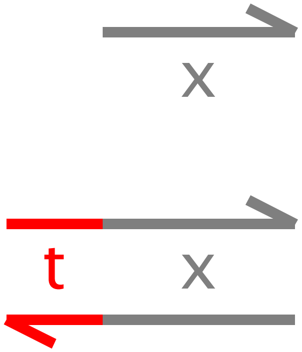
Branch Migration

# Strand Displacement



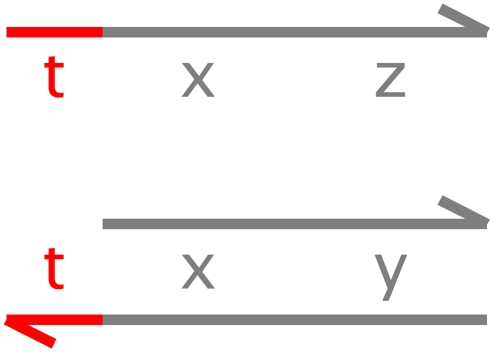
Displacement

# Strand Displacement

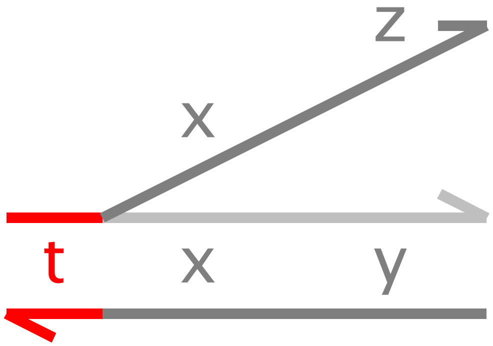


Irreversible release

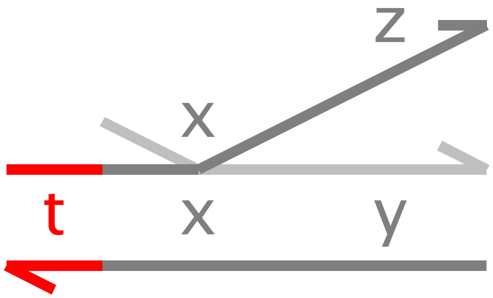
# Bad Match



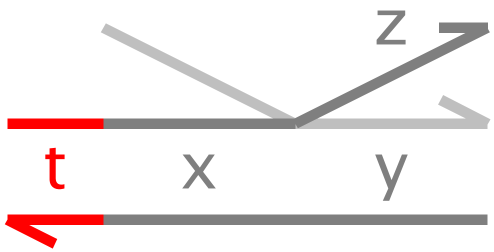
# Bad Match



# Bad Match



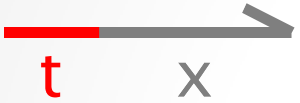
# Bad Match



Cannot proceed
Hence will undo

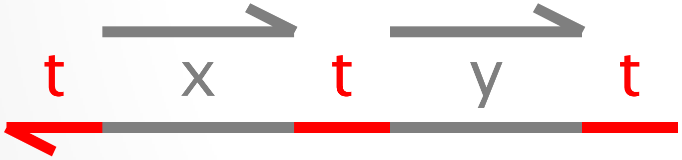
# Two-Domain Architecture

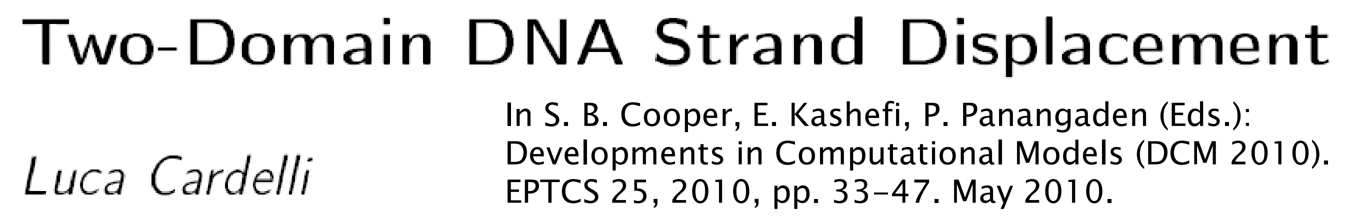
- Signals: 1 toehold + 1 recognition region



Garbage collection
“built into” the gates

- Gates: “top-nicked double strands”
(or equivalently double strands with open toeholds)



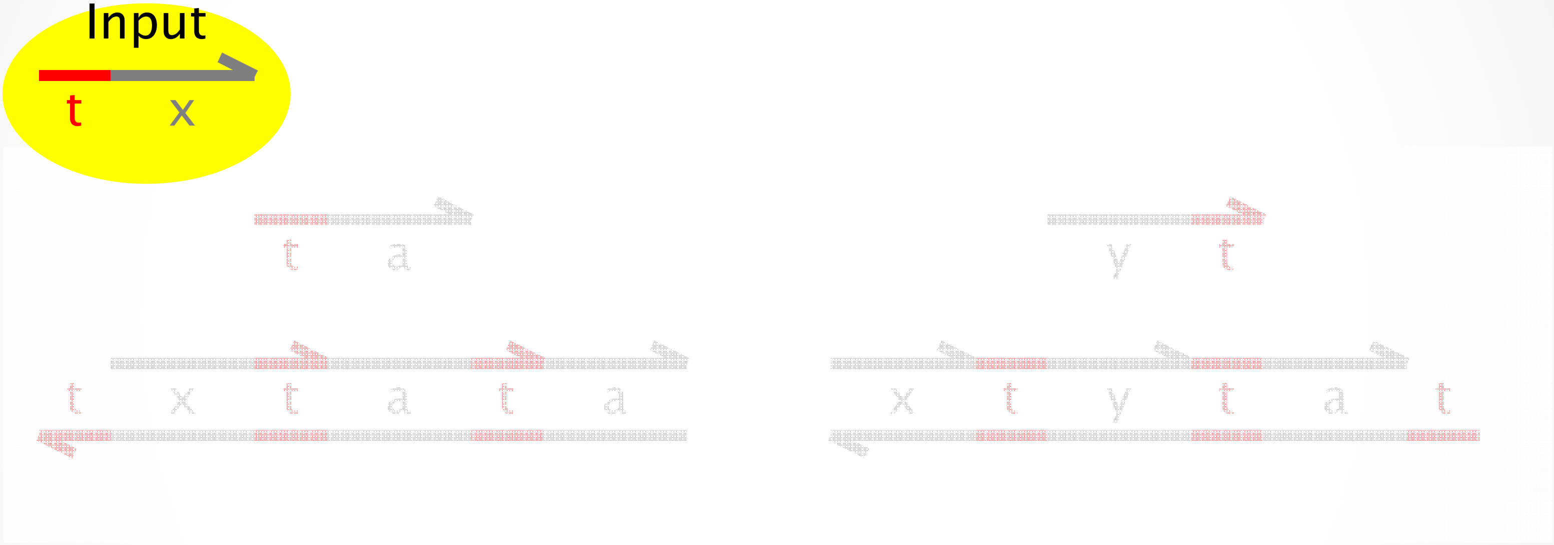


**Two-Domain DNA Strand Displacement**

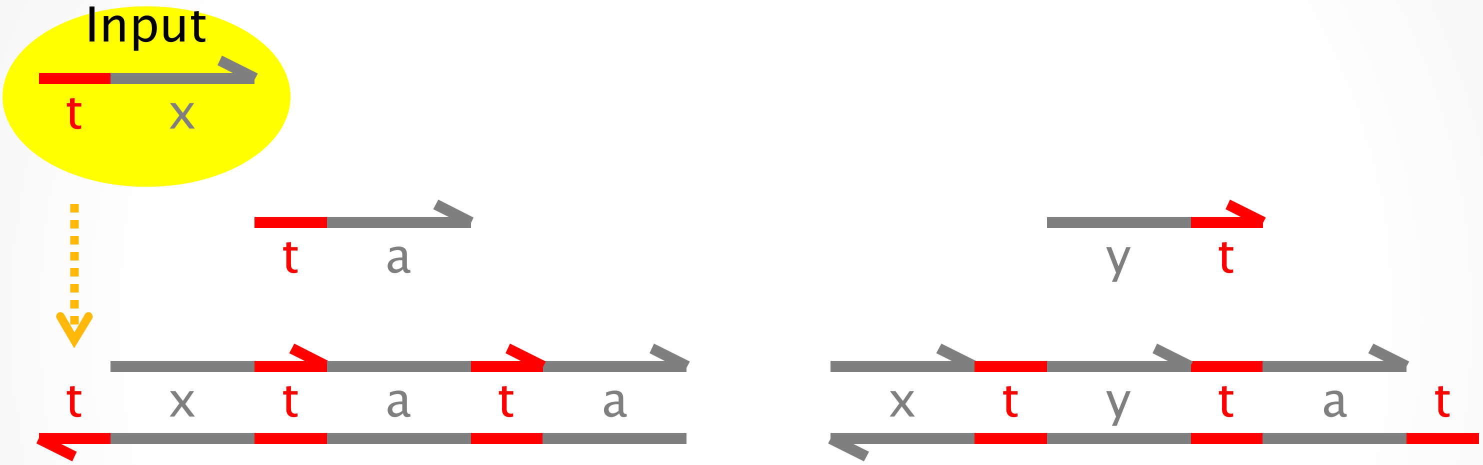
*Luca Cardelli*

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

# Transducer x→y



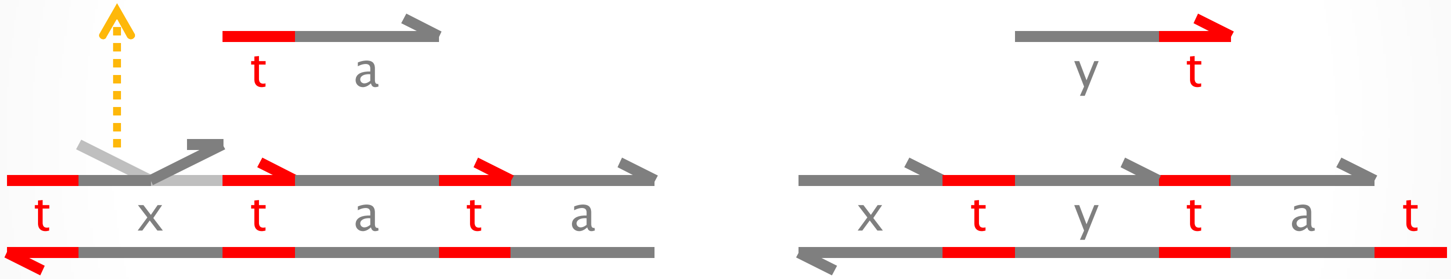
# Transducer x→y



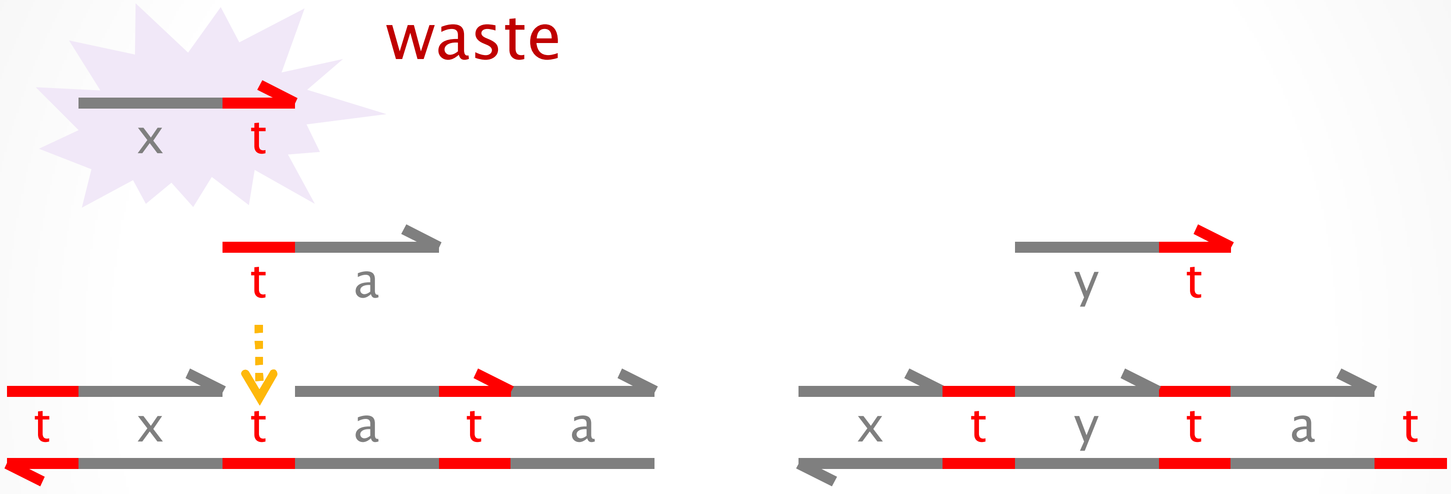
Built by self-assembly!

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

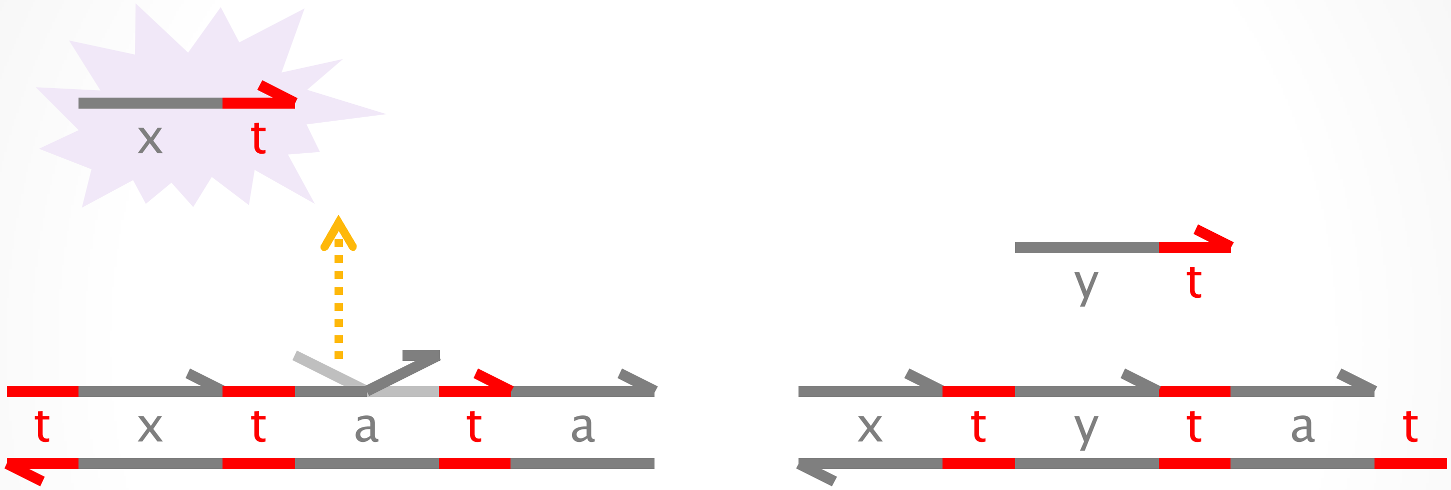
# Transducer x→y



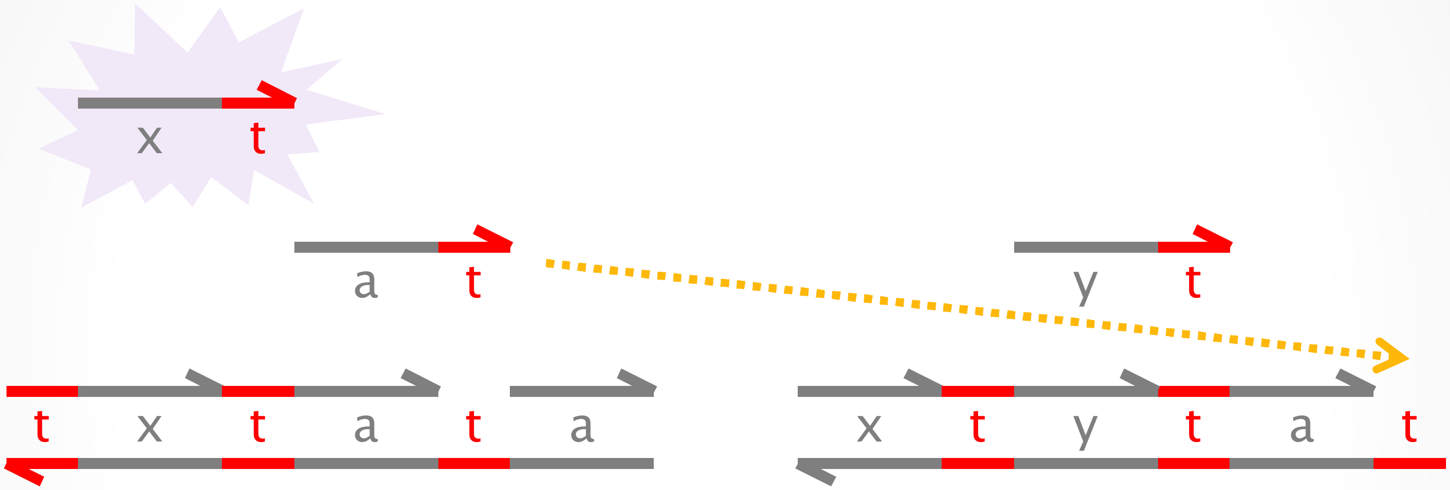
# Transducer x→y


Active
waste

# Transducer x→y

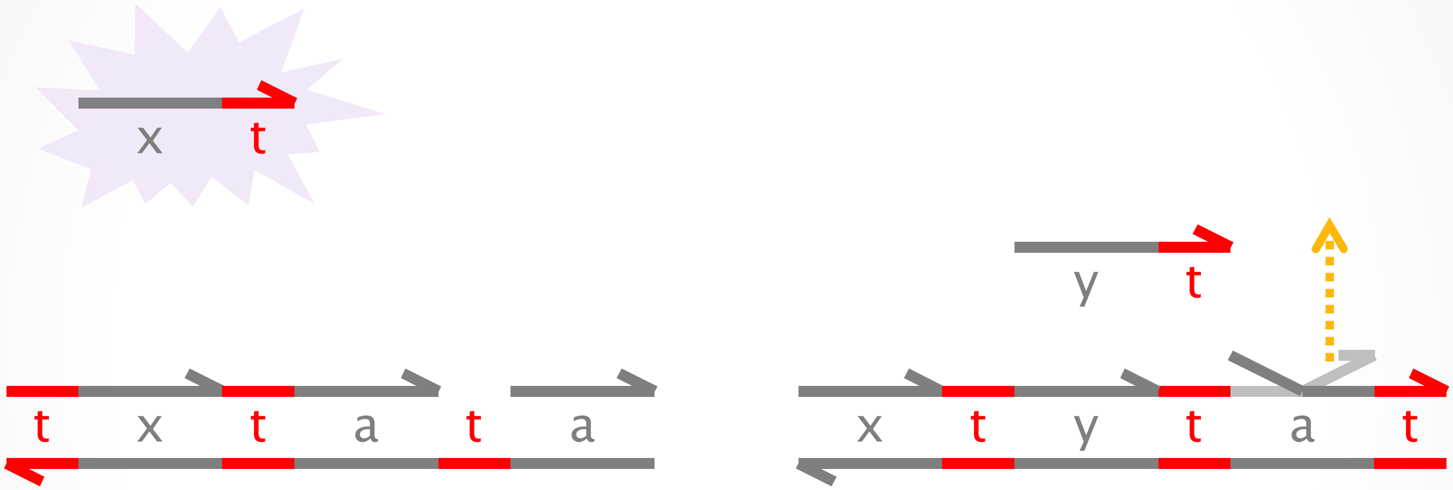


# Transducer x→y

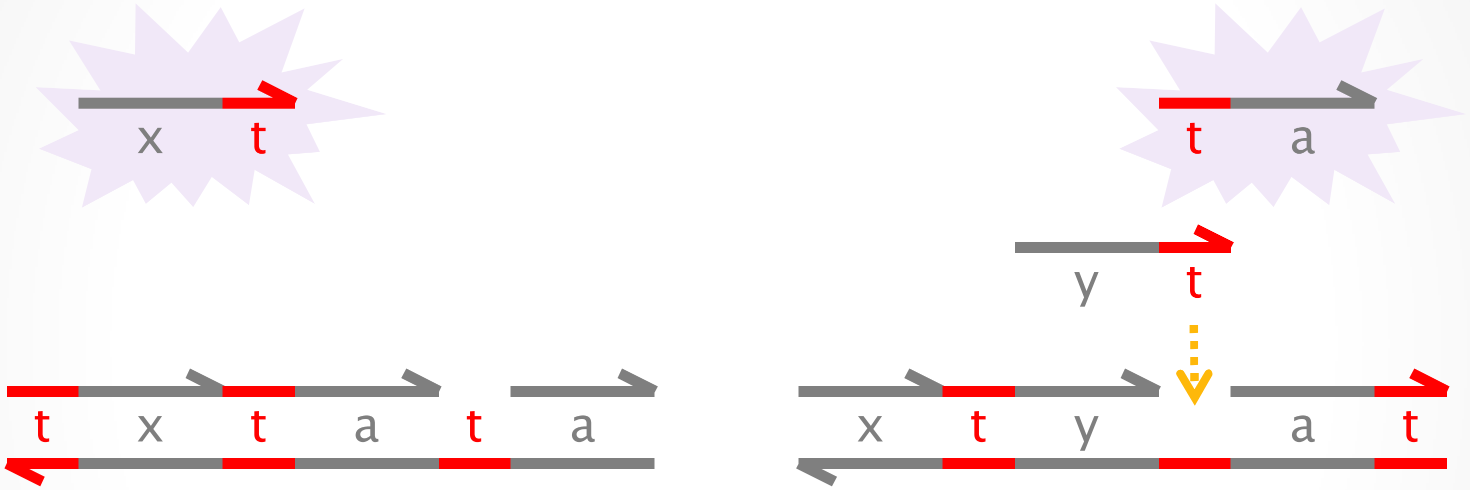


So far, a tx *signal* has produced an at *cosignal.*
But we want signals as output, not cosignals.

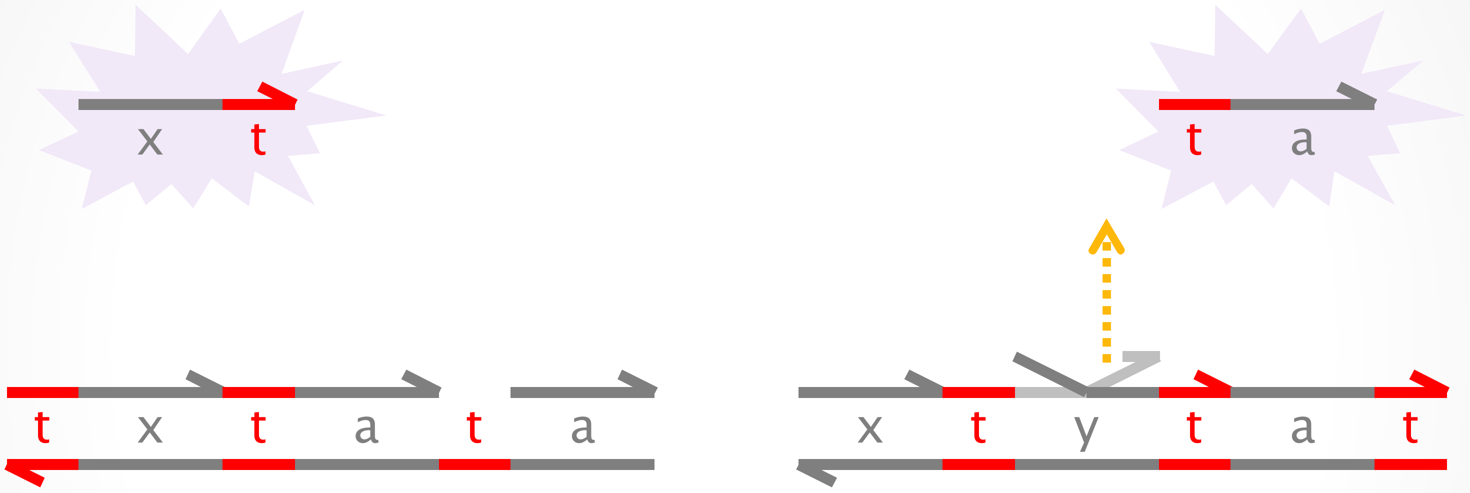
# Transducer x→y



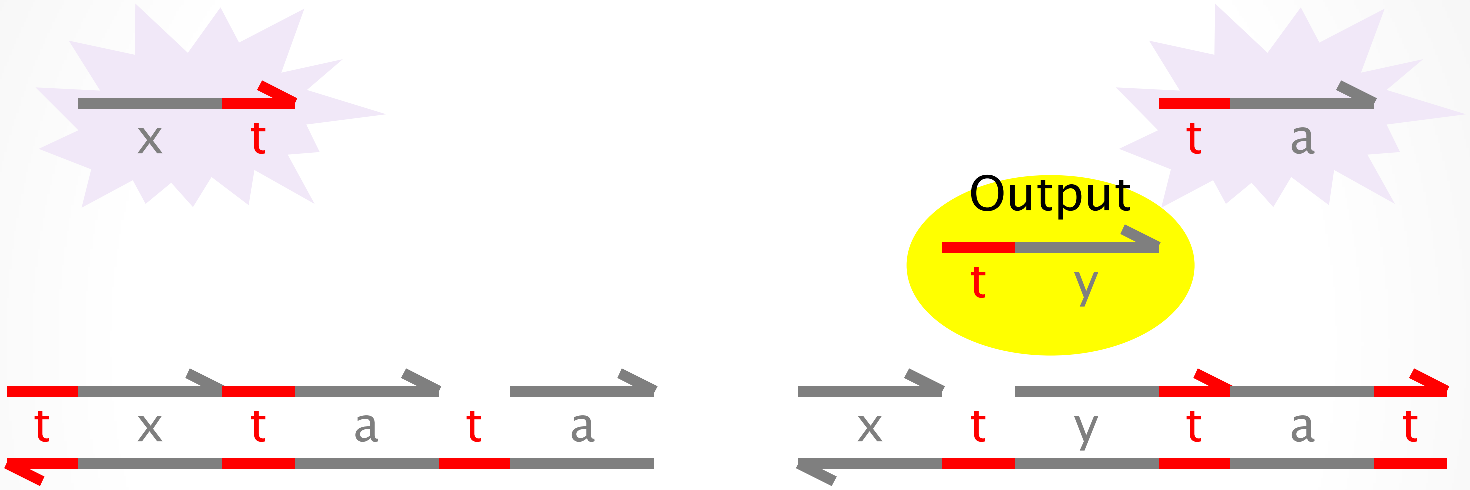
# Transducer x→y



# Transducer x→y



# Transducer x→y



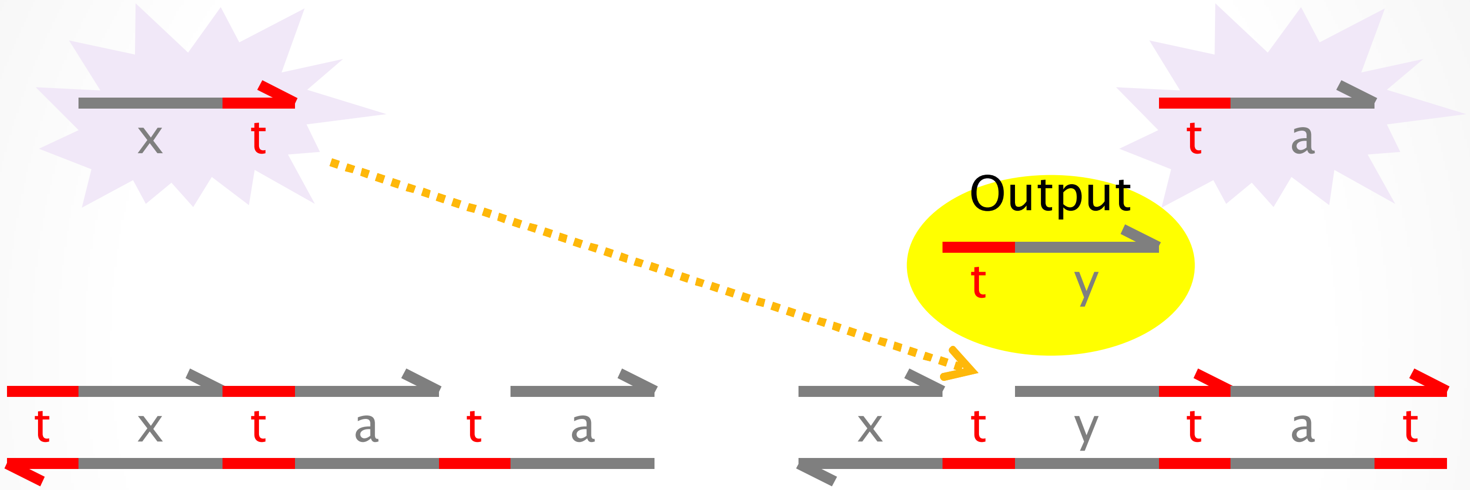
Here is our output ty *signal*.

But we are not done yet:

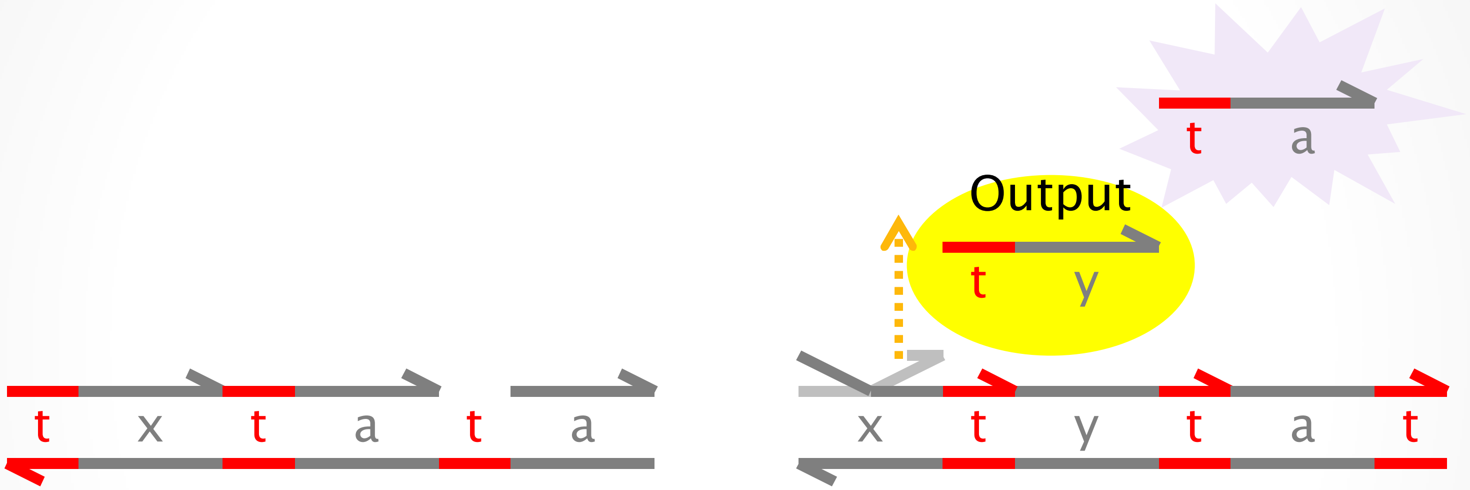
1) We need to make the output irreversible.
2) We need to remove the garbage.

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

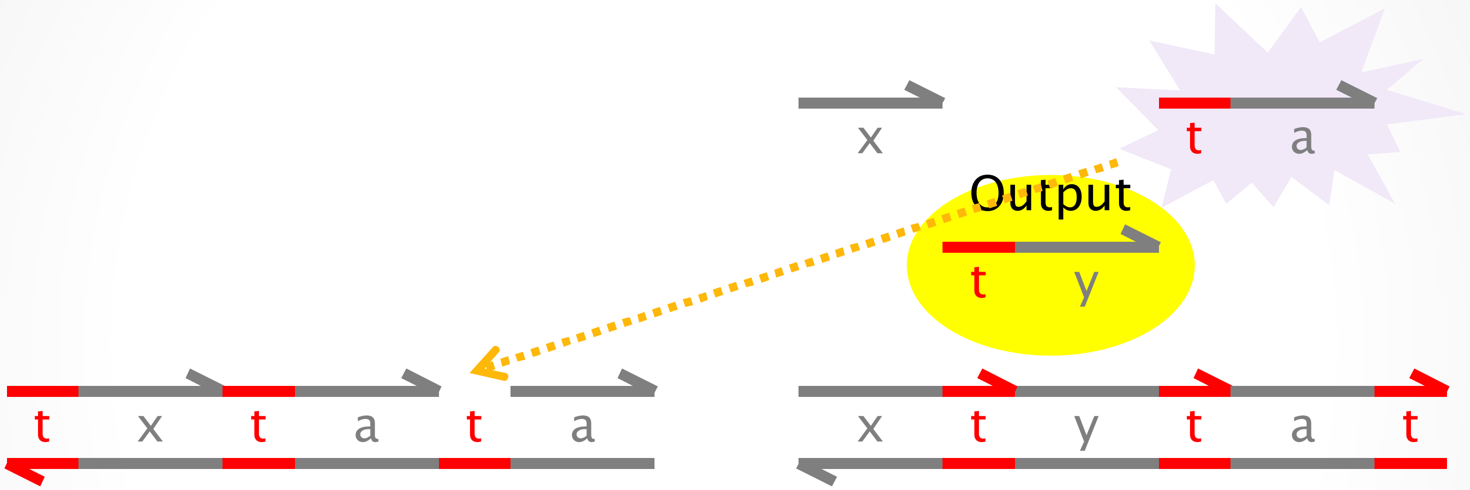
# Transducer x→y



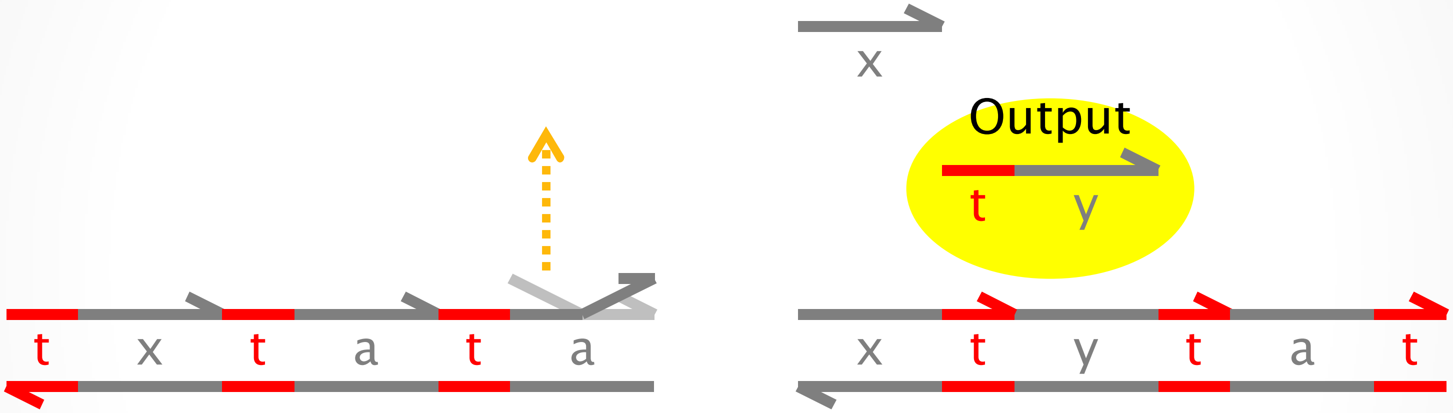
# Transducer x→y



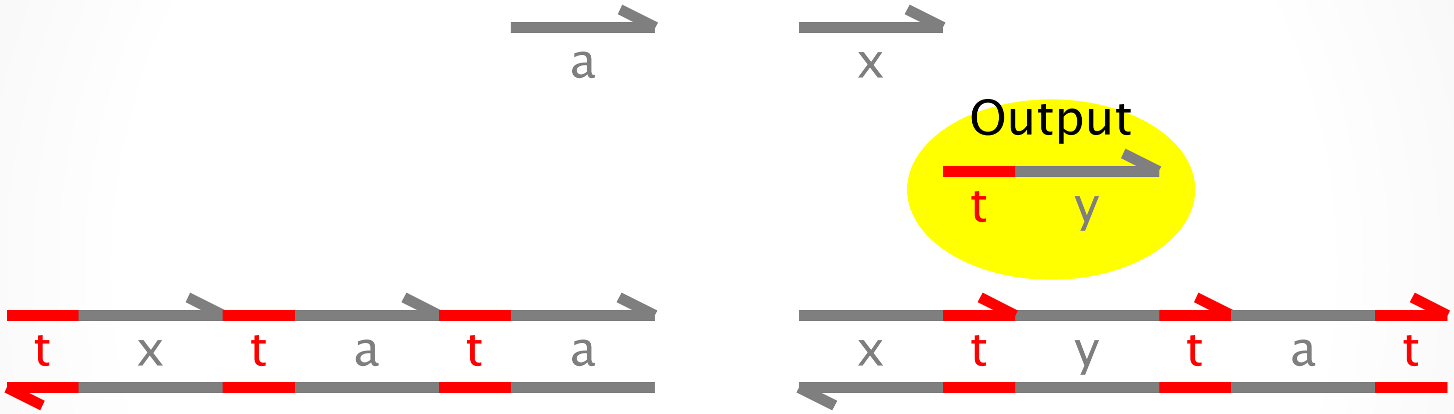
# Transducer x→y



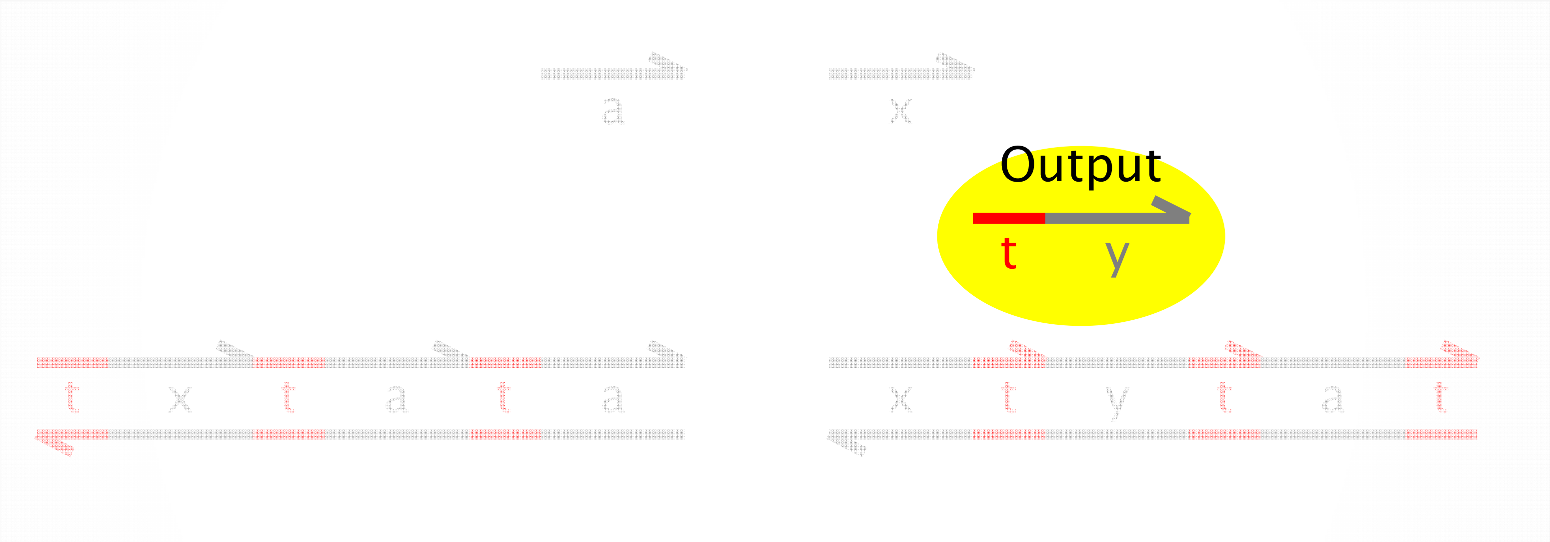
# Transducer x→y



# Transducer x→y



# Transducer x→y



Done.

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

# General n×m Join–Fork

- Easily generalized to 2+ inputs (with 1+ collectors).
- Easily generalized to 2+ outputs.

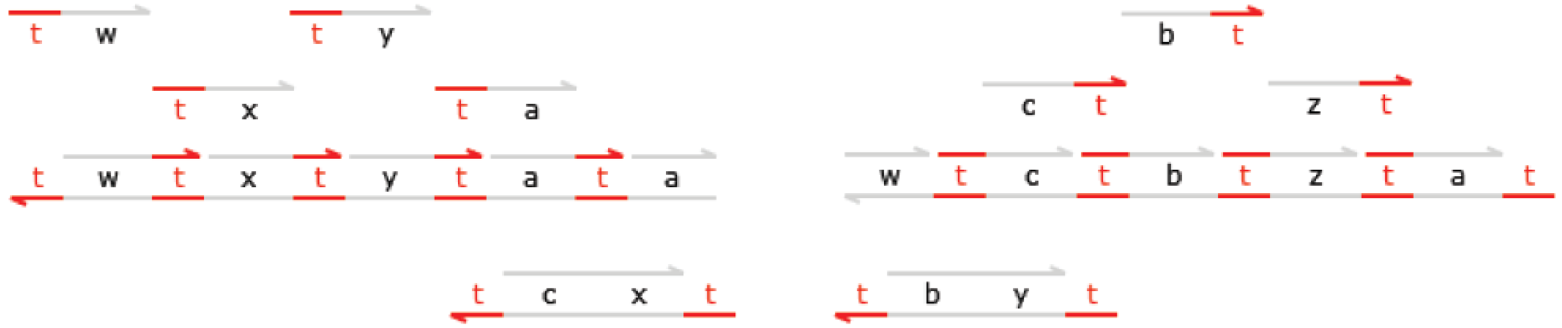
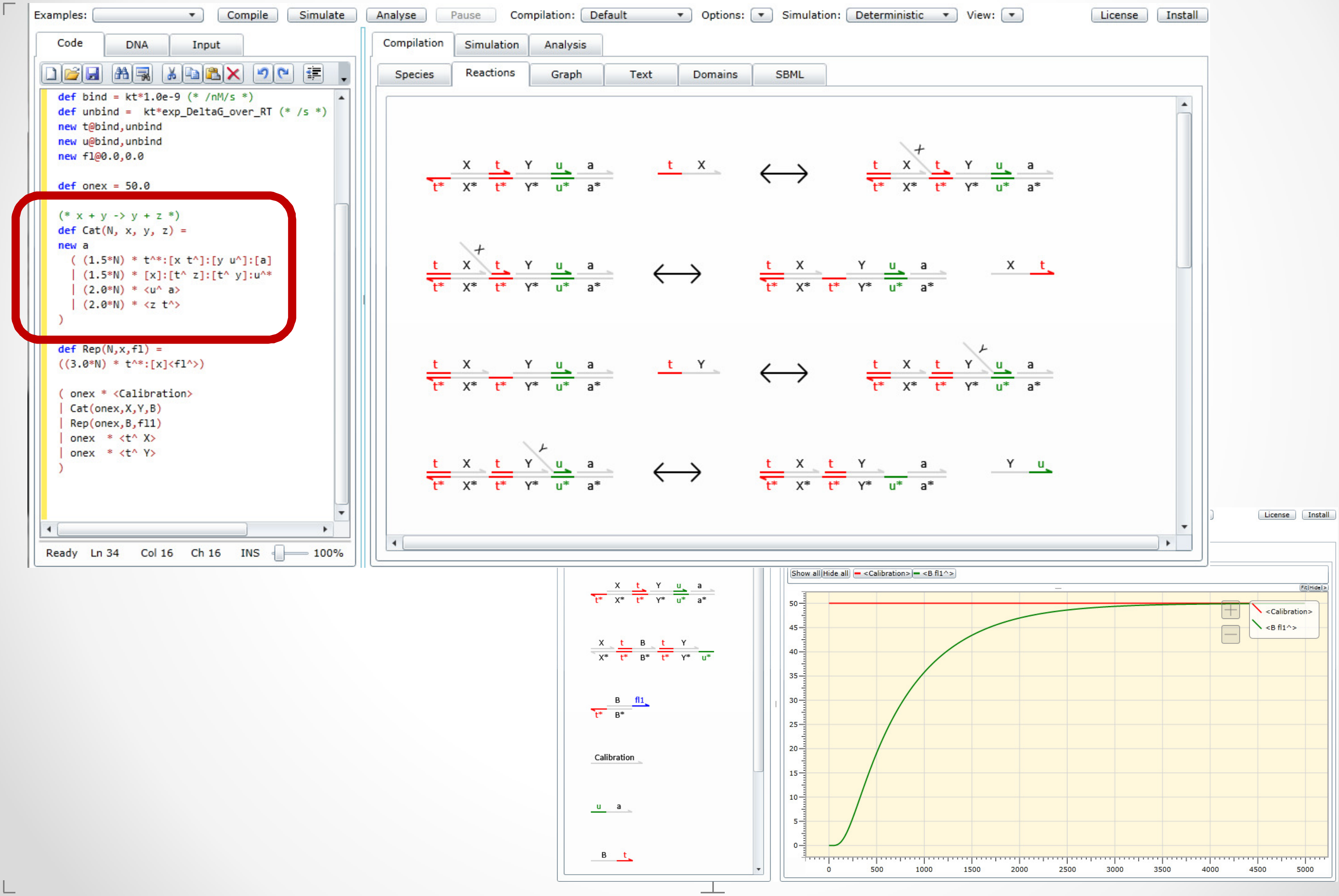


Figure 9: 3-Join $J_{wxyz} \mid tw \mid tx \mid ty \to tz$: initial state plus inputs $tw$, $tx$, $ty$.

# DNA Programming



Compilation: Default

Options:

Simulation: Deterministic

View:

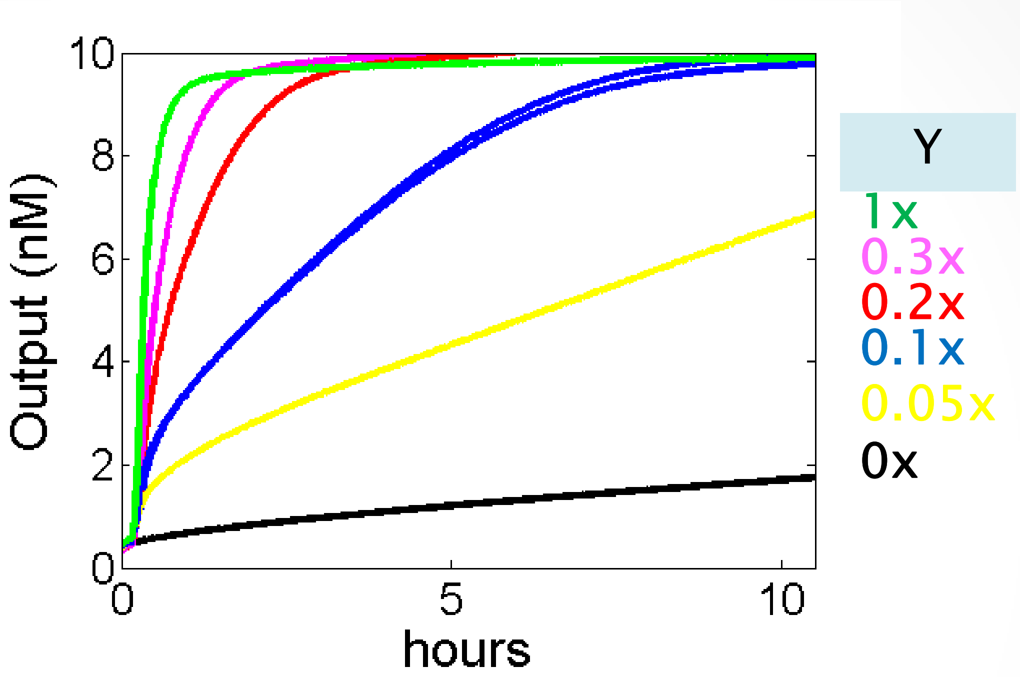
```
def bind = kt*1.0e-9 (* /nM/s *)
def unbind =  kt*exp_DeltaG_over_RT (* /s *)
new t@bind,unbind
new u@bind,unbind
new fl@0.0,0.0

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>
)
```

Ready   Ln 34   Col 16   Ch 16   INS   100%

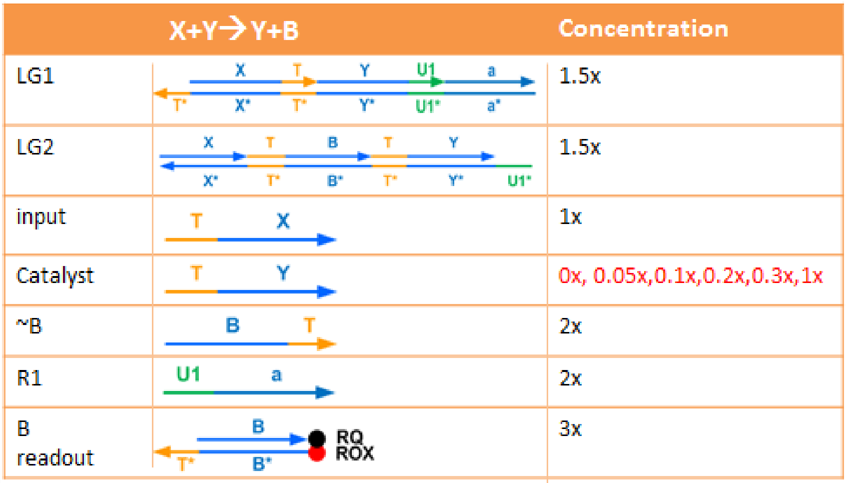
# Experiments

Two-domain gate
for X+Y → Y+B

X+Y→Y+B
35C
1x = 50nM



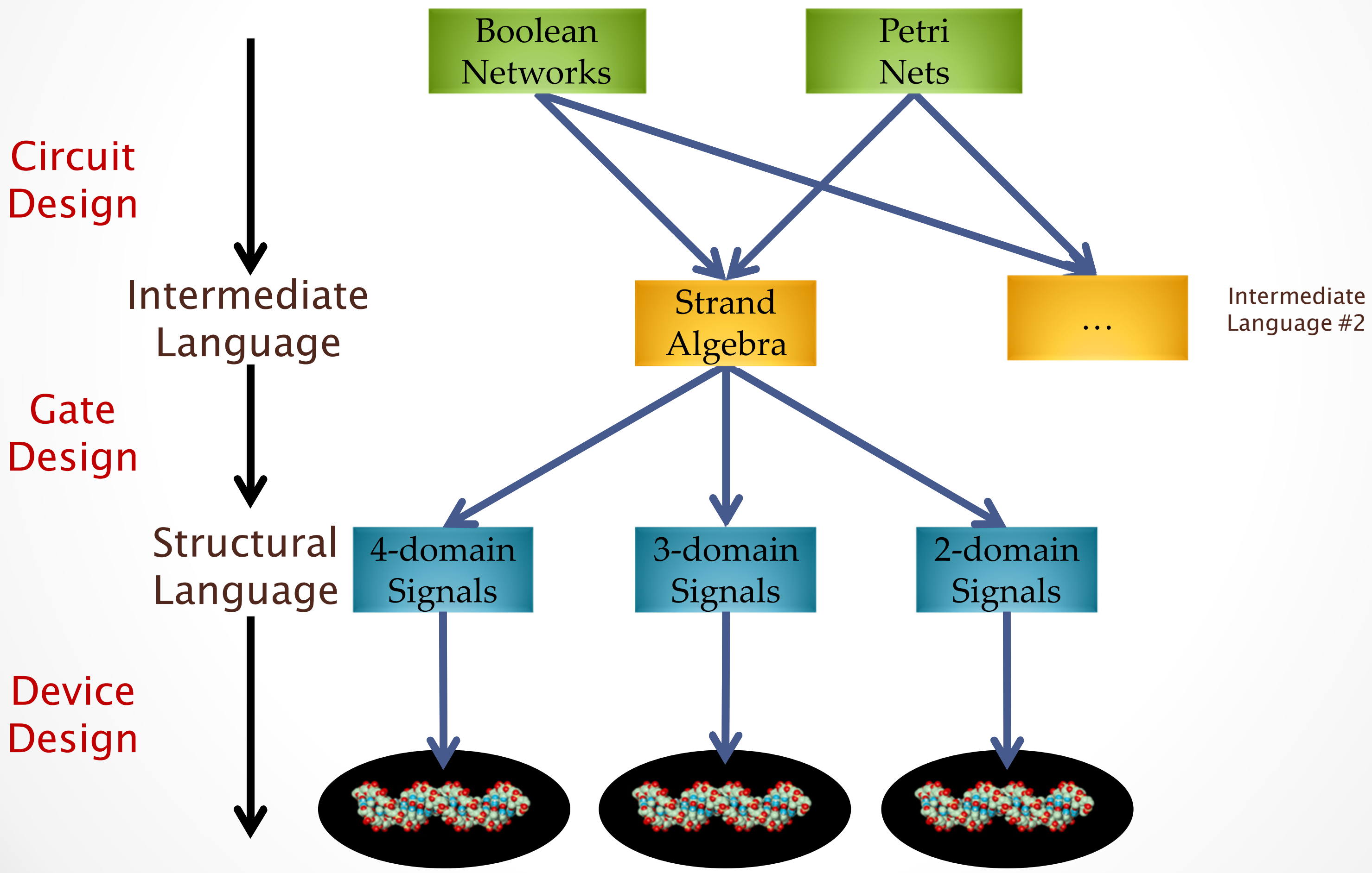
Yuan-Jyue Chen and Georg Seelig
U.Washingon.



| | X+Y→Y+B | Concentration |
|---|---|---|
| LG1 | X T Y U1 a<br>T* X* T* Y* U1* a* | 1.5x |
| LG2 | X T B T Y<br>X* T* B* T* Y* U1* | 1.5x |
| input | T X | 1x |
| Catalyst | T Y | 0x, 0.05x,0.1x,0.2x,0.3x,1x |
| ~B | B T | 2x |
| R1 | U1 a | 2x |
| B readout | B RQ<br>T* B* ROX | 3x |

# Molecular Programming Workflow

# Molecular Compilation



Circuit Design

Intermediate Language

Gate Design

Structural Language

Device Design

Boolean Networks

Petri Nets

Strand Algebra

…

Intermediate Language #2

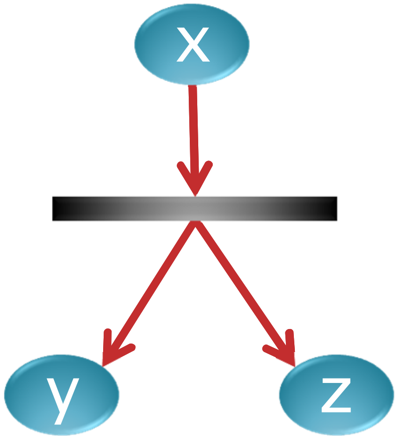
4-domain Signals

3-domain Signals

2-domain Signals

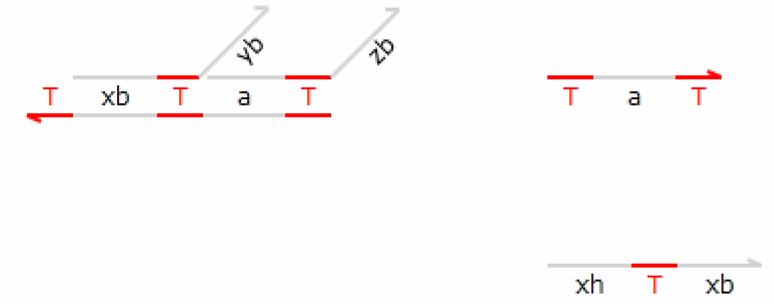
# Circuits to Signals and Gates

- E.g., a simple Petri Net fork transition
- In Strand Algebra: x | ([x].[y,z])*



# Signals and Gates to Structures

- Visual DSD [Andrew Phillips]



[x].[y,z] 3-domain gate

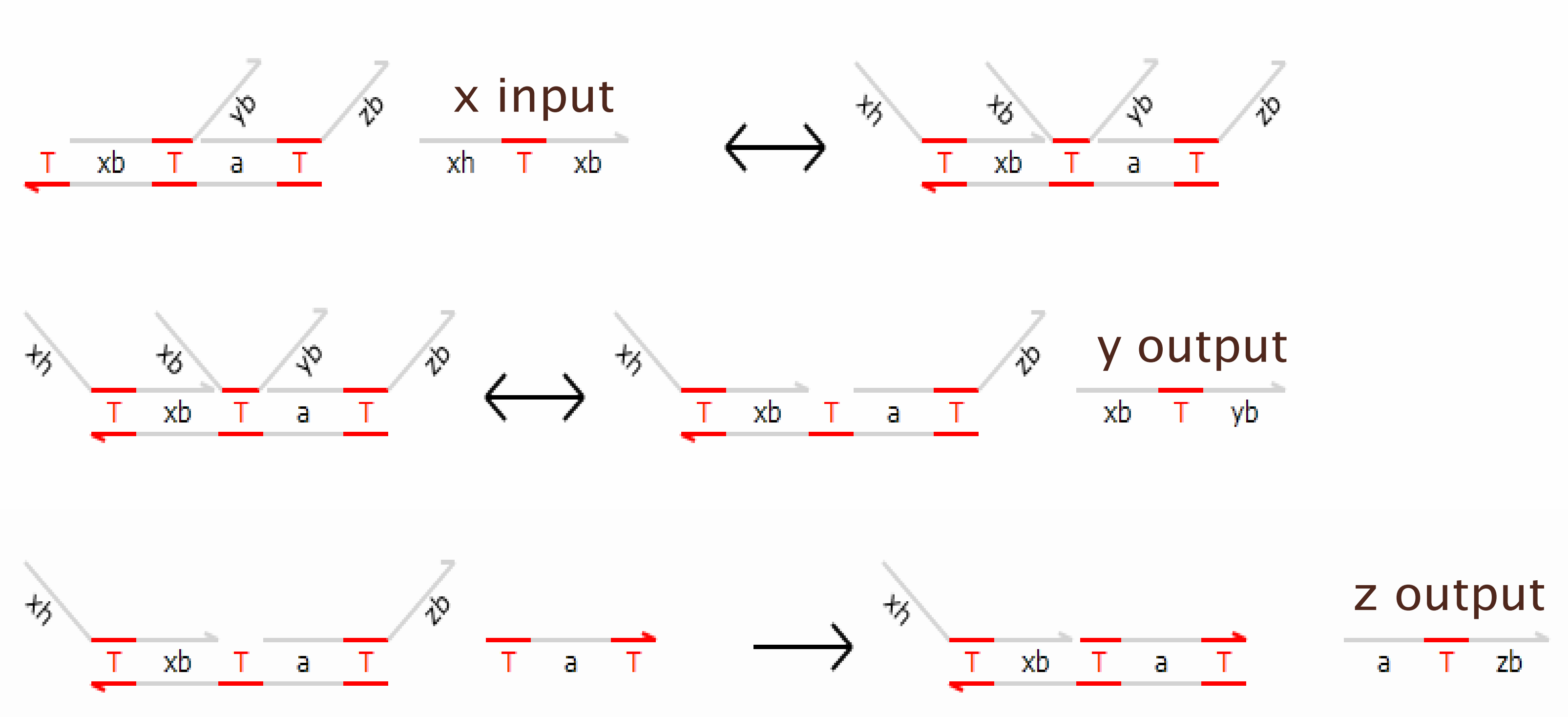
x 3-domain input

```
directive sample 5000.0 1000
directive plot sum(<_ T^ xb>); sum(<_ T^ yb>); sum(<_ T^ zb>)
def scaling = 1000
def bind = 0.0003/(float_of_int scaling) (* /nM/s *) (* =3*10^5 /M/s *)
def unbind =  0.1126 (* /s *)
new T@bind,unbind

def F1x2(N,Xb,Yb,Zb) =
new a
( N * T^:[Xb T^]<Yb>:[a T^]<Zb>
| N * <T^ a T^>
)

(   F1x2(10*scaling,xb,yb,zb)
|  (1*scaling)* <xh T^ xb>
)
```

Actual script

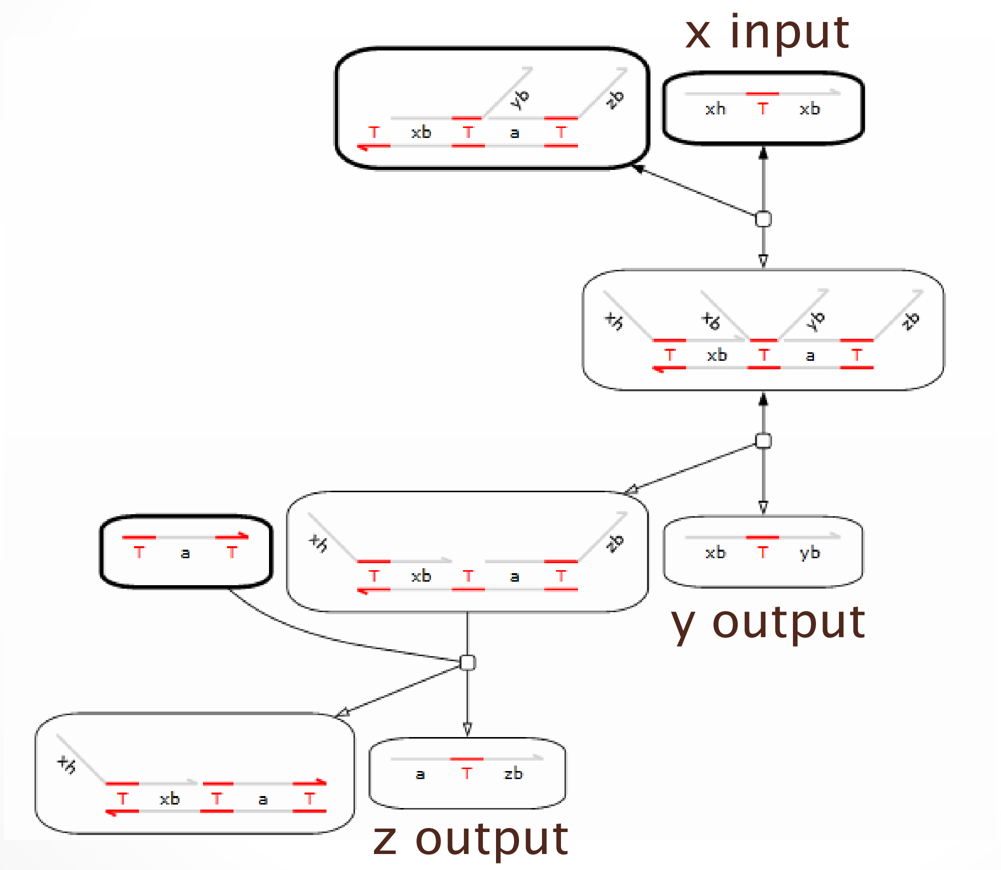
# Signals and Gates to Structures

- Fork gate: the reactions



# Signals and Gates to Structures

- Fork gate: the reaction graph

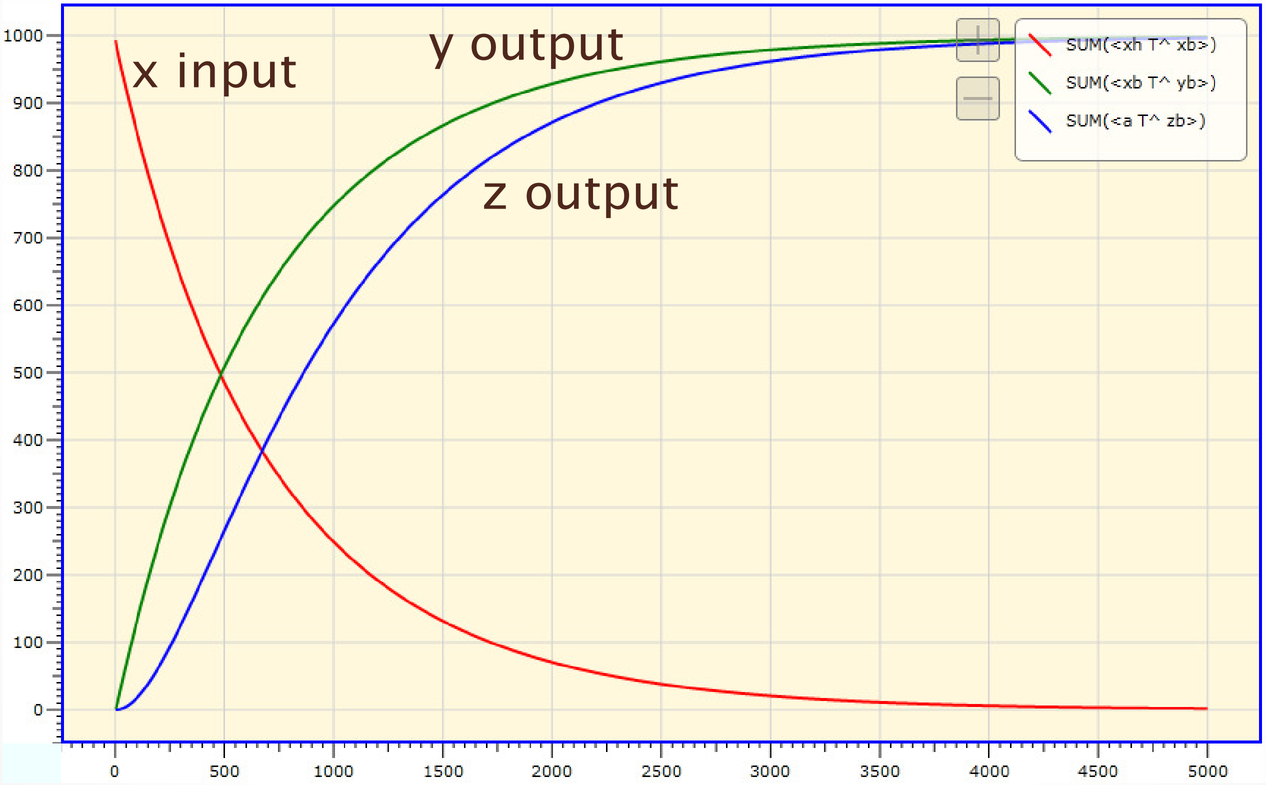

x input

y output

z output

# Signals and Gates to Structures

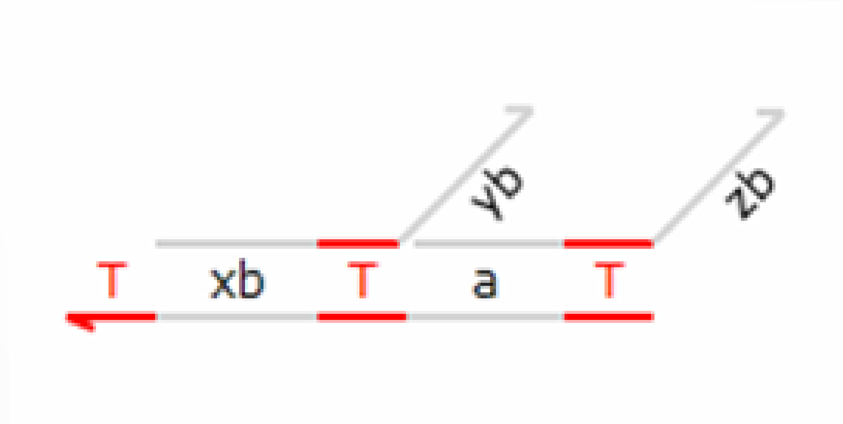
- Fork gate: the behavior



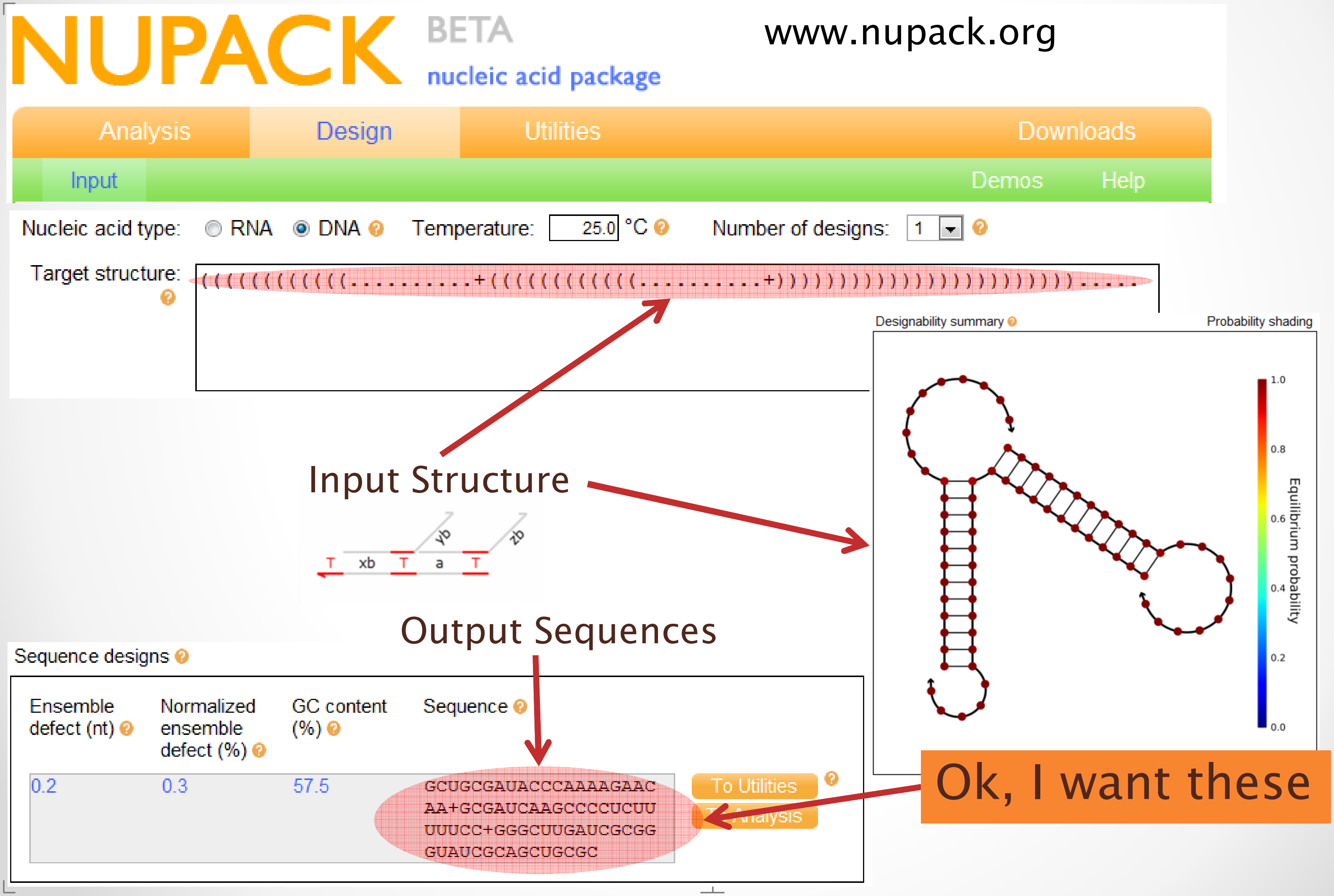
# Signals and Gates to Structures

- Fork gate: check

Ok, I want this



# Structures to Sequences



www.nupack.org

Nucleic acid type: ○ RNA ● DNA Temperature: 25.0 °C Number of designs: 1

Target structure:

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

Input Structure

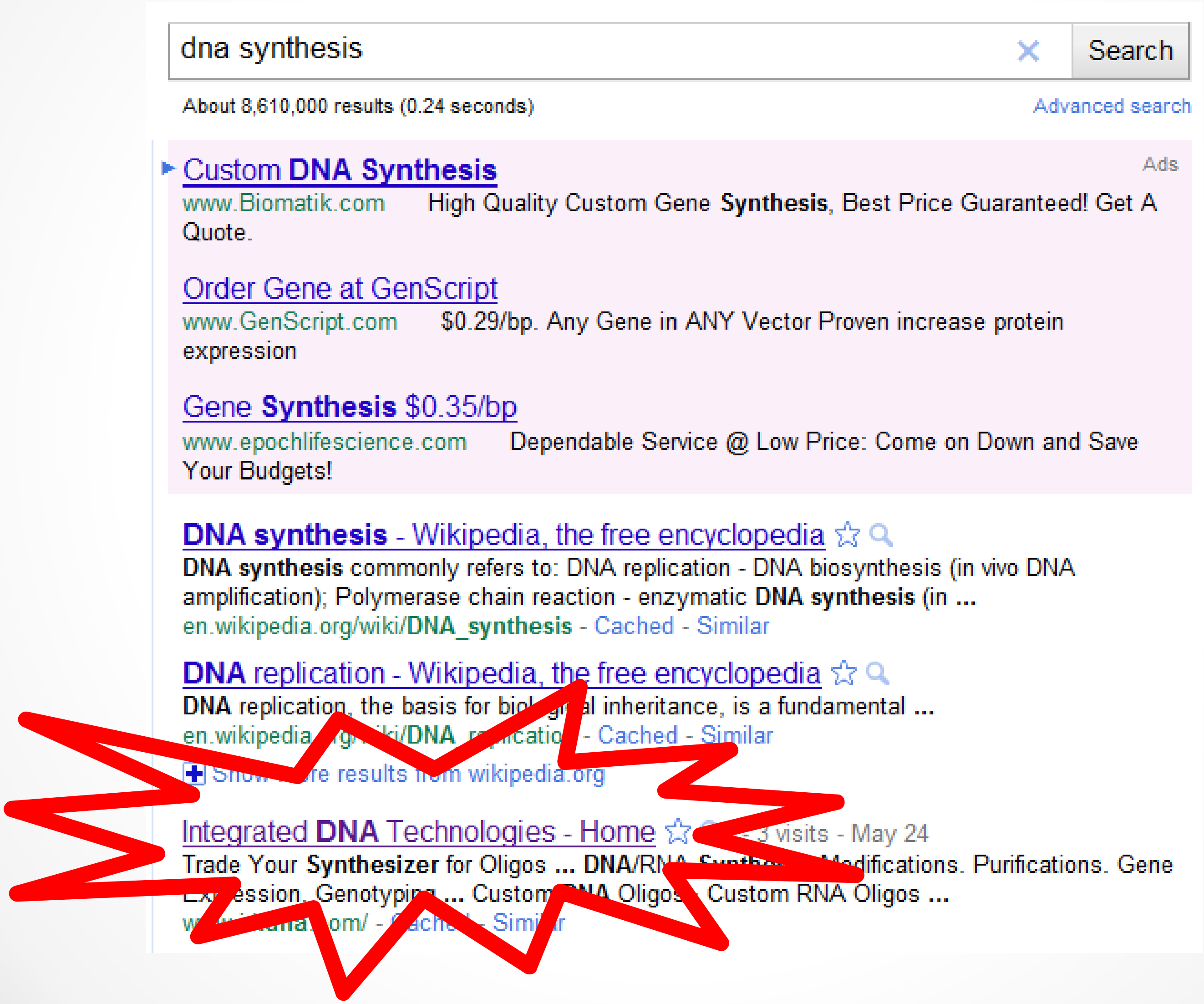
Output Sequences

Sequence designs

| Ensemble defect (nt) | Normalized ensemble defect (%) | GC content (%) | Sequence |
|---|---|---|---|
| 0.2 | 0.3 | 57.5 | GCUGCGAUACCCAAAAGAAC<br>AA+GCGAUCAAGCCCCUCUU<br>UUUCC+GGGCUUGAUCGCGG<br>GUAUCGCAGCUGCGC |

Ok, I want these

# DNA Synthesis



About 8,610,000 results (0.24 seconds)

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

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

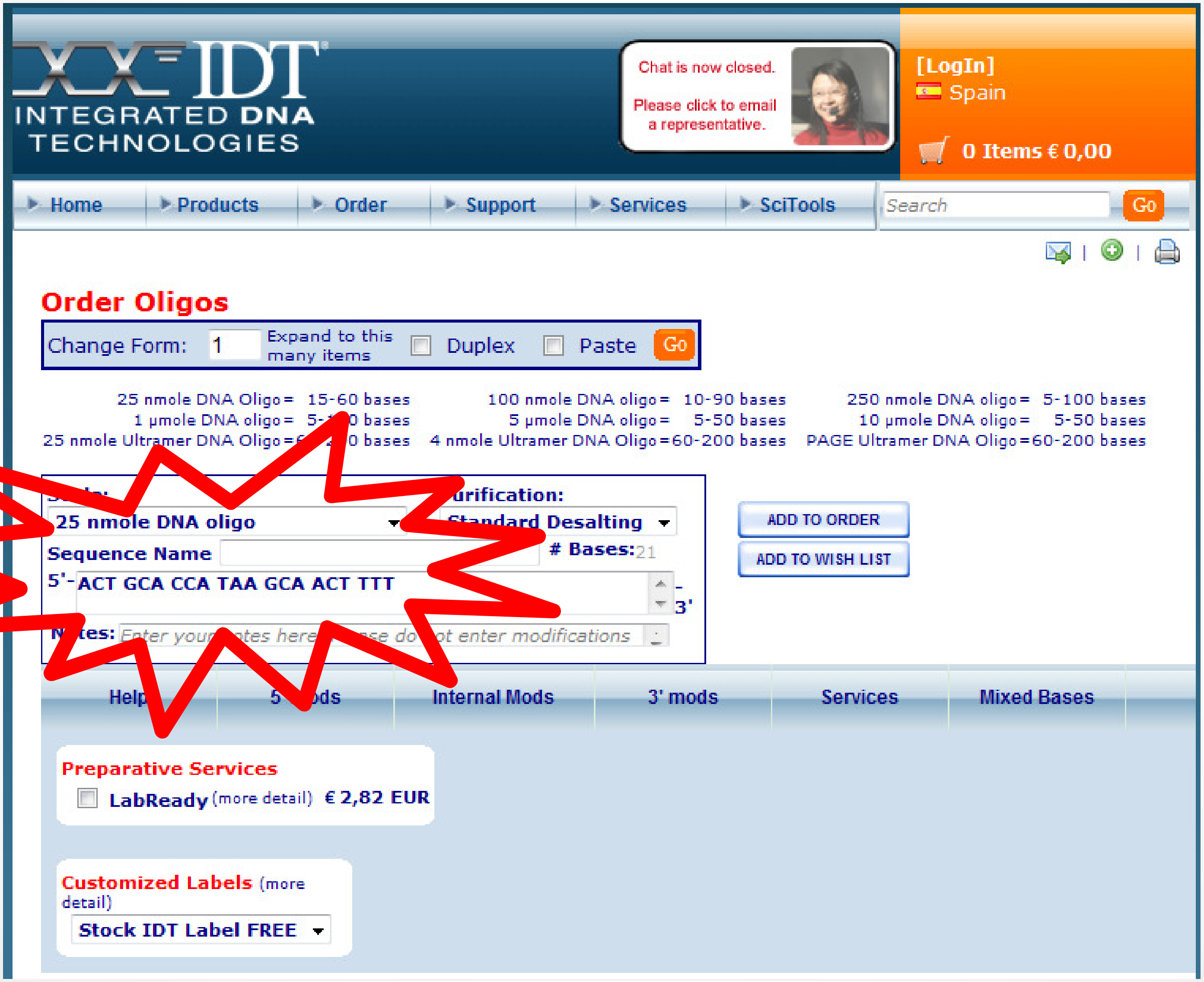
[Gene **Synthesis** $0.35/bp](www.epochlifescience.com)
www.epochlifescience.com Dependable Service @ Low Price: Come on Down and Save Your Budgets!

[**DNA synthesis** - Wikipedia, the free encyclopedia](en.wikipedia.org/wiki/DNA_synthesis)
**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](en.wikipedia.org/wiki/DNA_replication)
**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](www.idtdna.com/)
3 visits - May 24
Trade Your **Synthesizer** for Oligos **...** DNA/RNA **Synthesis**. Modifications. Purifications. Gene Expression. Genotyping **...** Custom DNA Oligos. Custom RNA Oligos **...**
www.idt**dna**.com/ - Cached - Similar

# Sequences to Molecules



## Order Oligos

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

25 nmole DNA Oligo = 15-60 bases
1 µmole DNA oligo = 5- 0 bases
25 nmole Ultramer DNA Oligo = 0 bases
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

**le:** 25 nmole DNA oligo

**urification:** Standard Desalting

**Sequence Name**

**# Bases:** 21

5'- ACT GCA CCA TAA GCA ACT TTT -3'

**tes:** Enter your tes here se do t enter modifications

**Preparative Services**

☐ **LabReady** (more detail) € 2,82 EUR

**Customized Labels** (more detail)

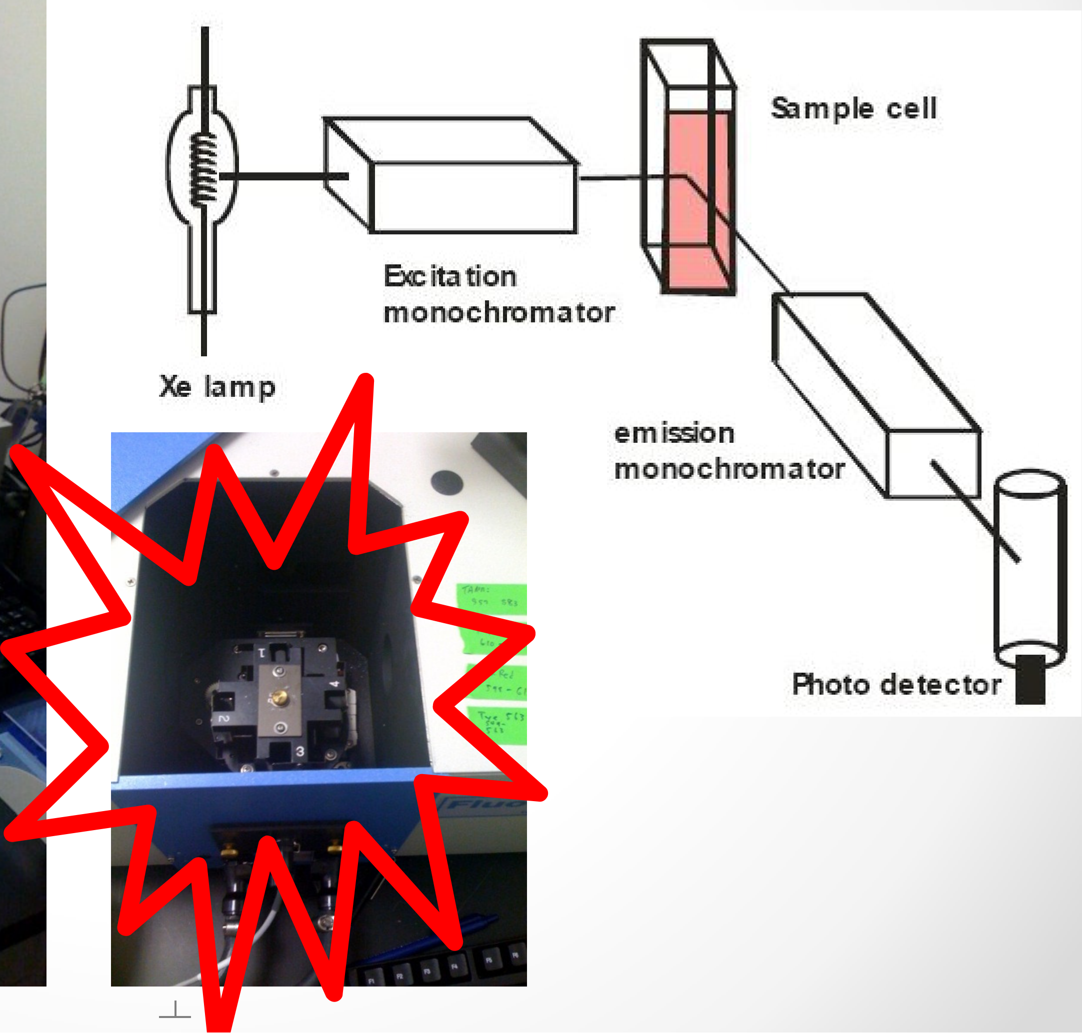
Stock IDT Label FREE

# Molecules by Mail

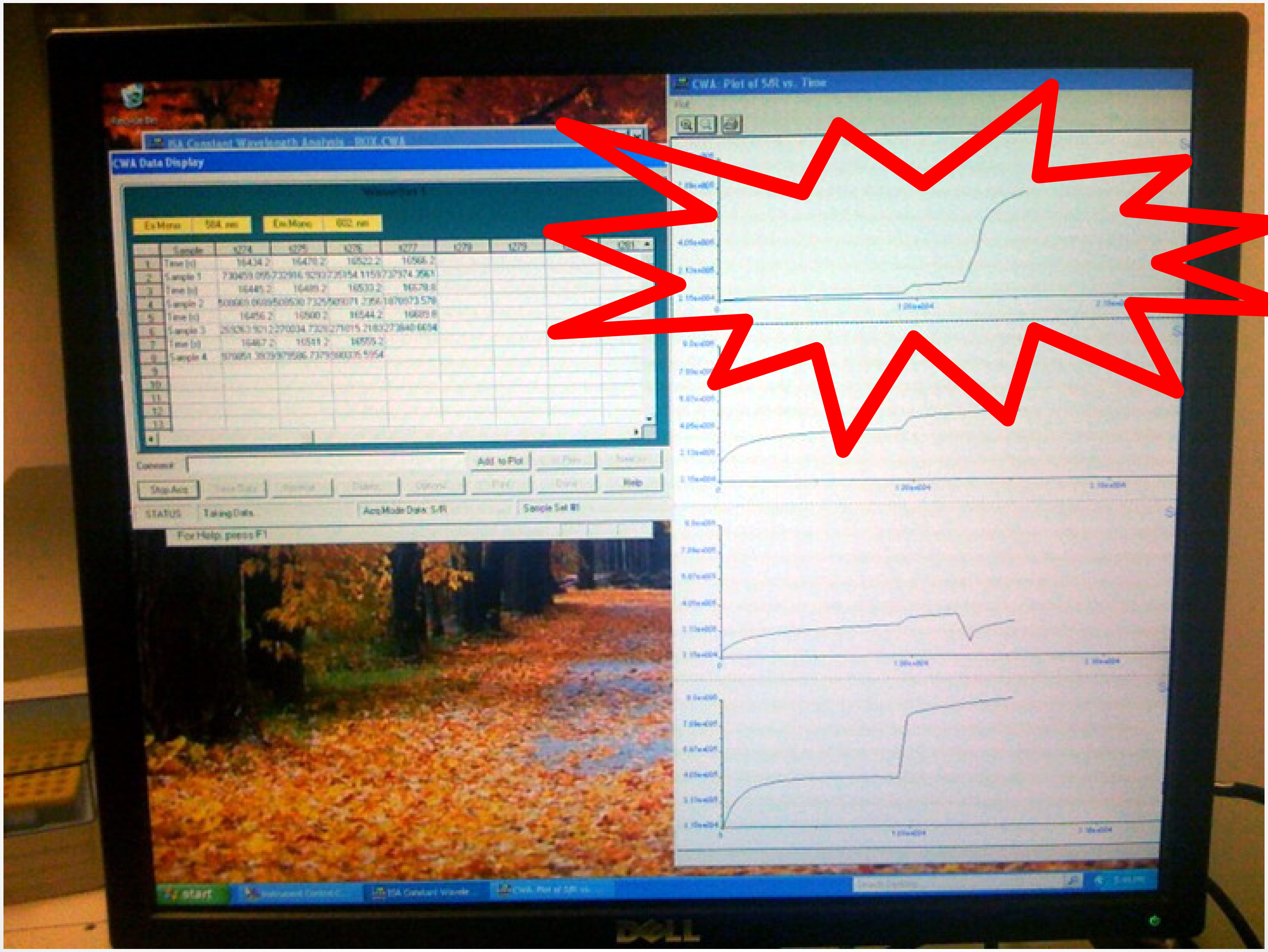
# Add Water

# Execution

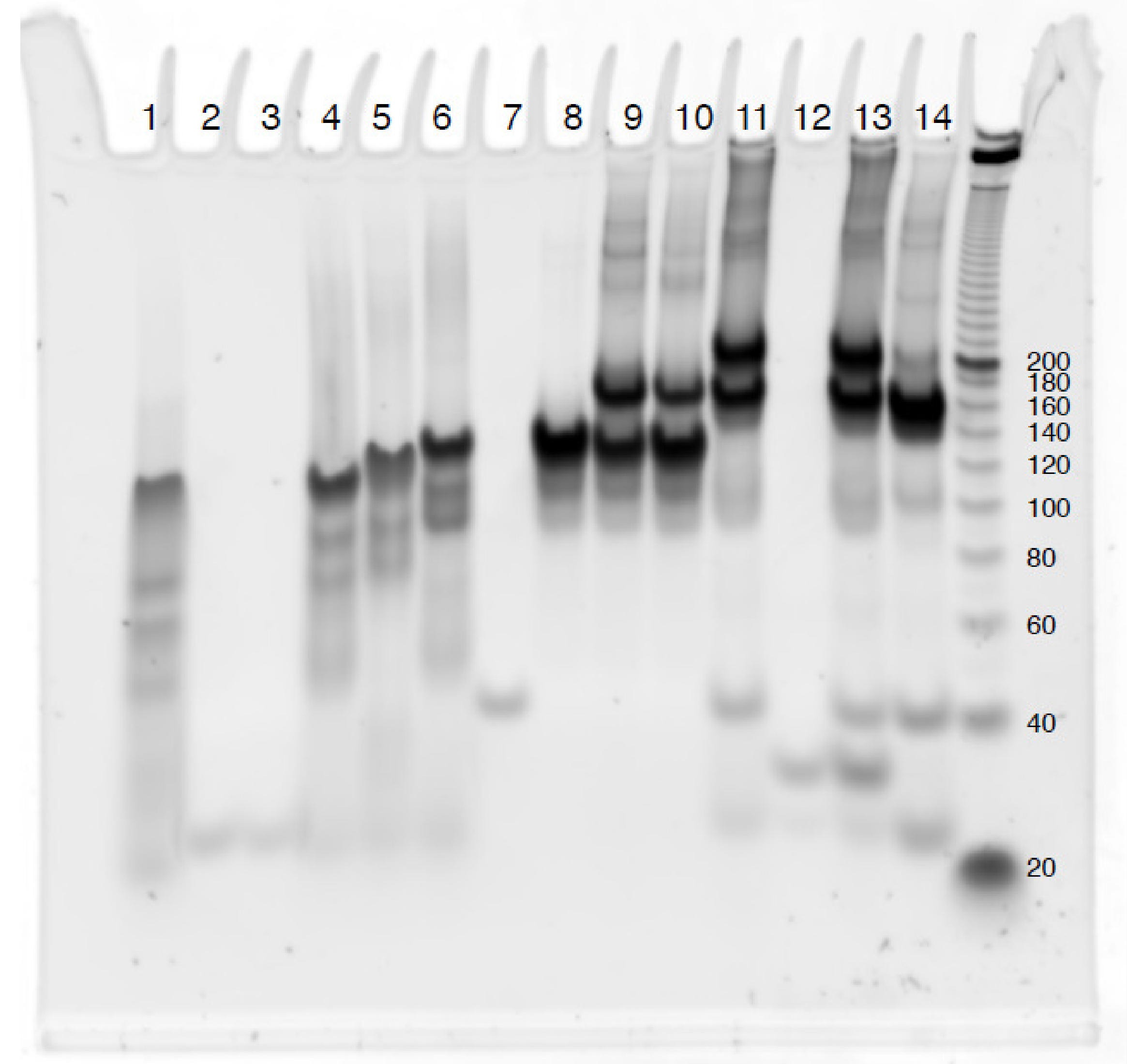
- Fluorescence is your ‘print’ statement



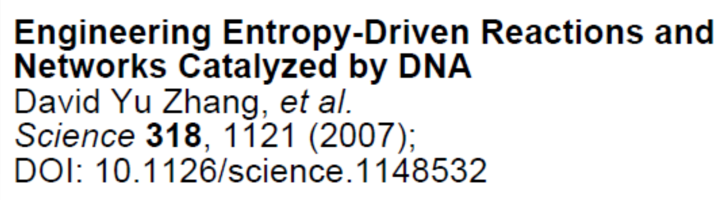
# Output


CWA: Plot of S/R vs. Time
CWA Data Display
Ex Mono
584 nm
Em Mono
602 nm
Sample
Time (s)
Sample 1
Sample 2
Sample 3
Sample 4
Comment
Add to Plot
Stop Acq
Help
STATUS
Taking Data
Acq Mode Data: S/R
Sample Set #1
For Help, press F1
start

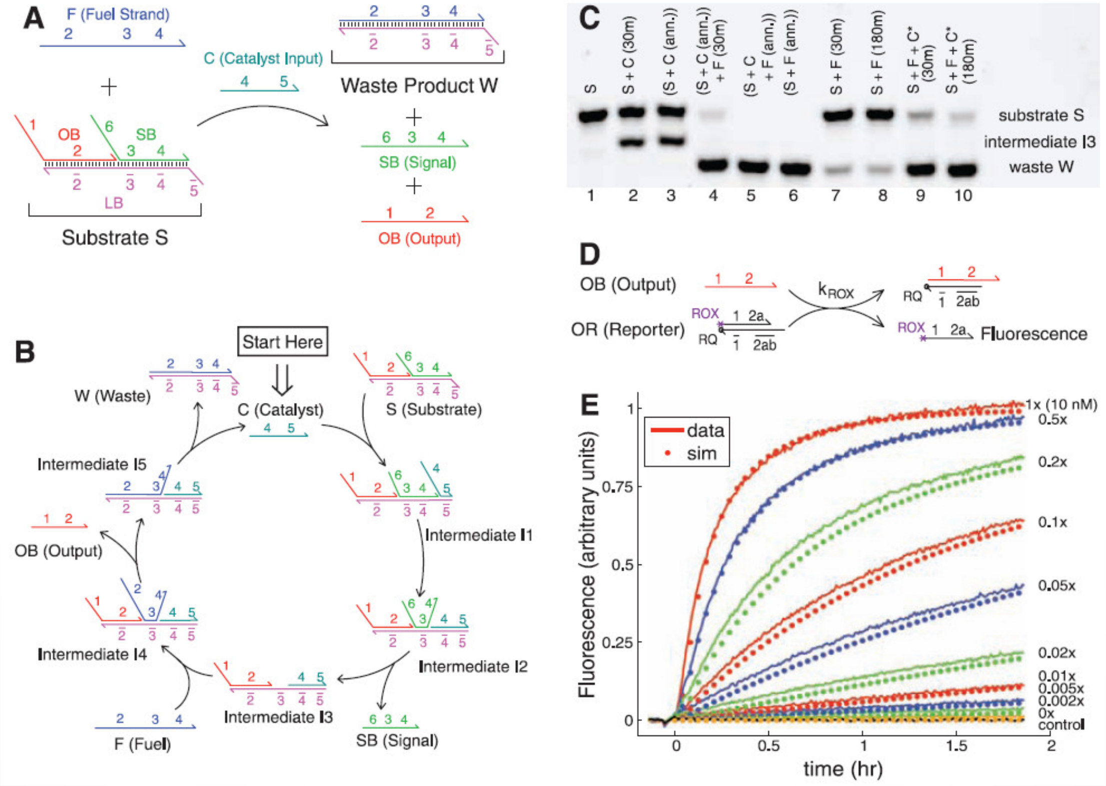
# Debugging



# Publishing!



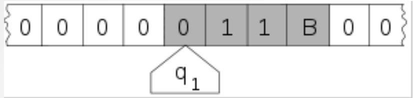
**Engineering Entropy-Driven Reactions and Networks Catalyzed by DNA**
David Yu Zhang, *et al.*
*Science* **318**, 1121 (2007);
DOI: 10.1126/science.1148532



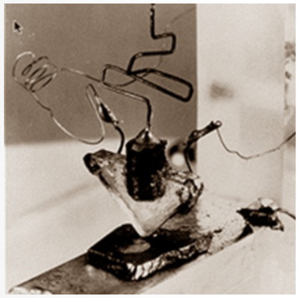
# Conclusions

# A Brief History of DNA

Turing Machine, 1936

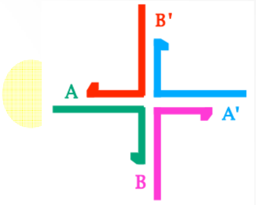


Transistor, 1947

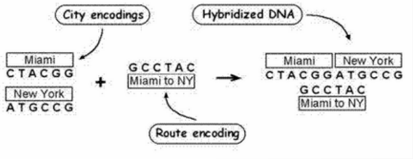


DNA, −3,800,000,000

Structural DNA, 1982



DNA Algorithm, 1994



~~Digital Computers~~

Computer programming

**Software**

*systematic manipulation of information*

20th century

**Matterware??**

*systematic manipulation of matter*

21th century

~~DNA Computers~~

Molecular programming

# Acknowledgments

- Microsoft Research
  - Andrew Phillips
- Caltech
  - Winfree Lab
- U.Washington
  - Seelig Lab