The Molecular Programming Project

molecular-programming.org

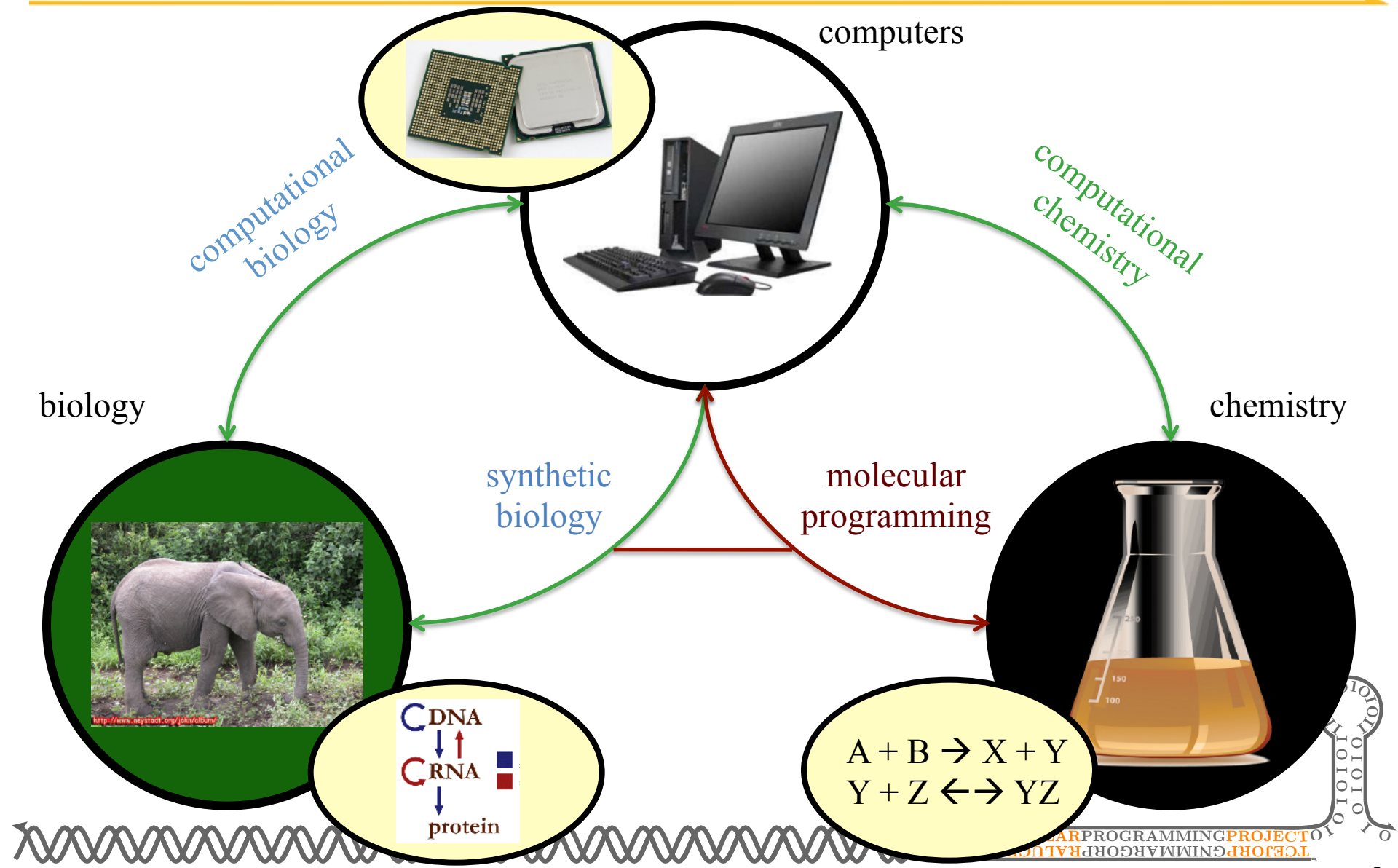
Expeditions in Computing PI Meeting
May 14-16, 2013

PI: Erik Winfree

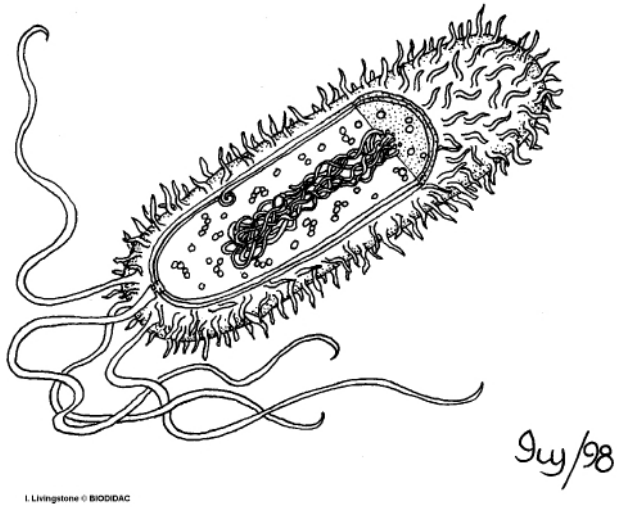
co-PIs: Jehoshua Bruck, Eric Klavins, Richard Murray, Niles Pierce, Paul Rothemund

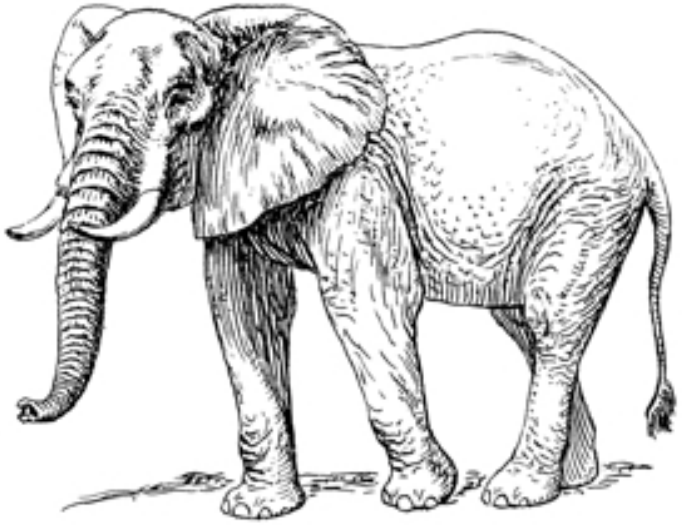
California Institute of Technology and University of Washington

Chemistry as an information technology



The biological inspiration





1 micron^3 volume
4 megabit genome
biochemical circuitry
manufacturing plant
atomic-level design
➔ deep physics & chemistry

grows from a single cell
contains 10^{15} cells
10^{27} macromolecules
spans atomic to macro scales
intelligent behavior
➔ deep algorithmic issues

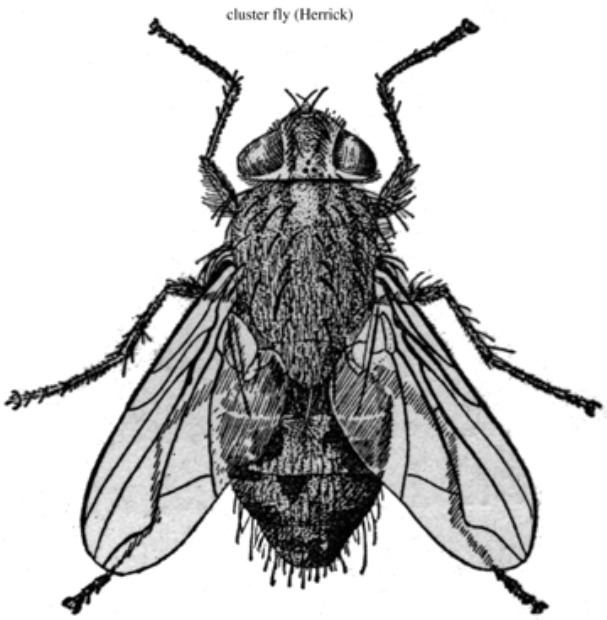
Can we engineer molecular systems with life-like qualities?

The challenge of programming chemical systems

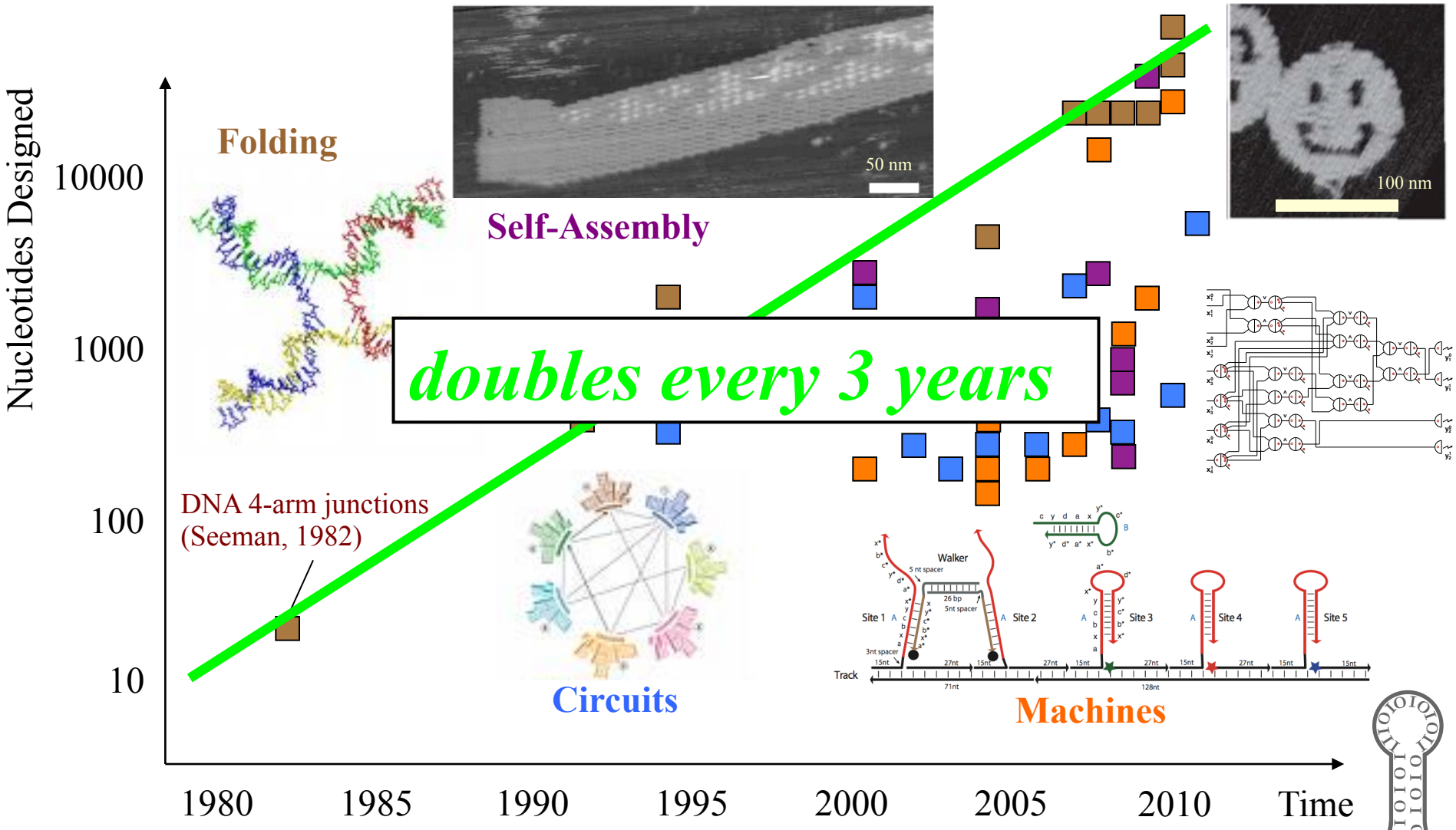
- **What kind of programming language do we need to build a fly?**
  - What are the programmable molecular and biochemical building blocks?
  - New concepts for programming and analyzing such systems?
  - Intrinsic fault-tolerance, adaptation, and learning conceptually built in?
  - How to incorporate geometric and mechanical factors?

```
include gro
alpha := 0.75;
program p() := {
  gfp := 0;
  r := [ t := 0 ];
  selected & just_divided : {
    print ( "At time ", r.t, ":
        After division, cell ", id,
        " has ", gfp, " gfp molecules" )
  }
  rate ( alpha * volume ) : {
    gfp := gfp + 1
  }
  true : {
    r.t := r.t + dt
  }
};
stemcell ( [], program p() );
```

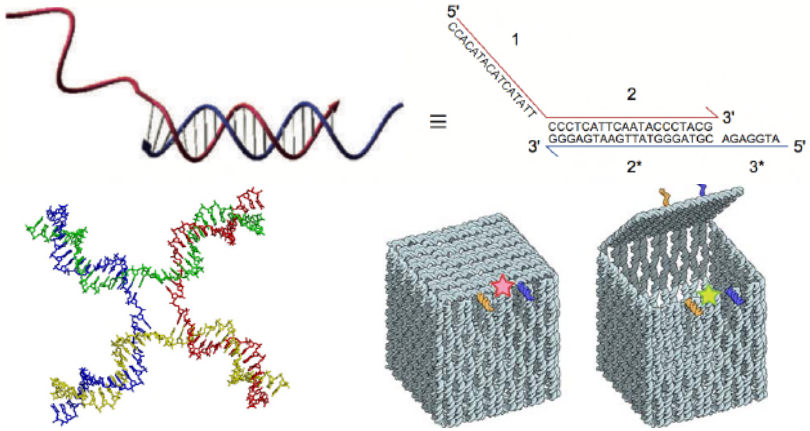
?



Design complexity in DNA nanotechnology



Implementation and applications

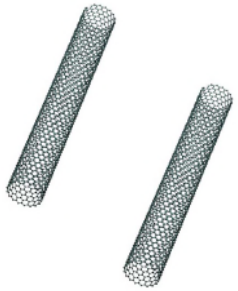


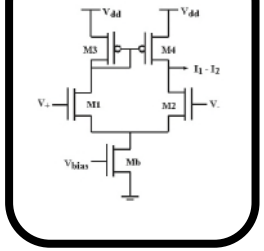
nucleic acids: DNA and RNA

protein enzymes: polymerase, ribosome, …

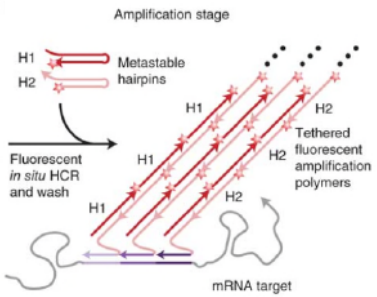
living cells

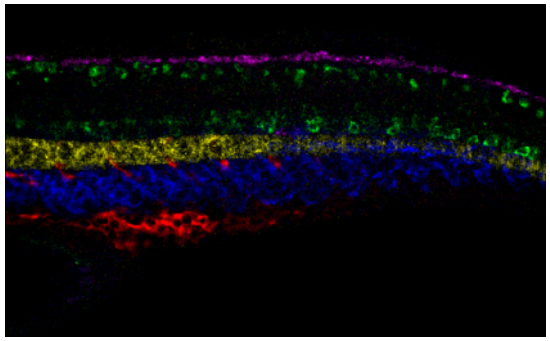
DNA origami breadboard



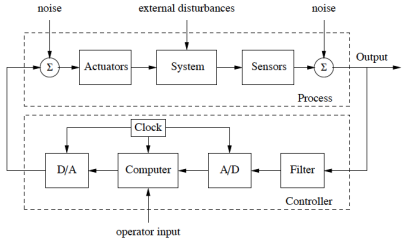


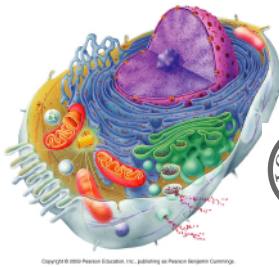
molecular device fabrication by self-assembly





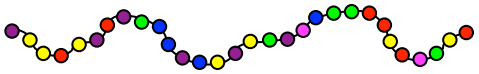
programmable molecular instruments for science



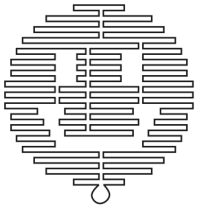


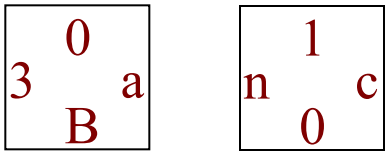
reprogramming cellular circuits and behavior

Molecular programs and compilers

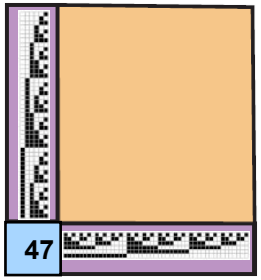


folding



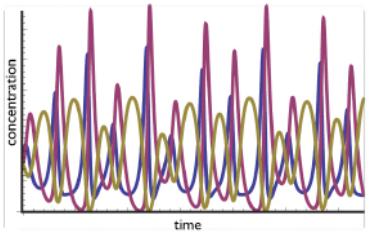


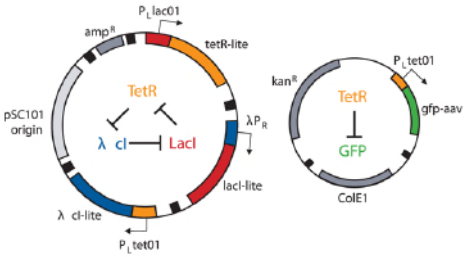
self-assembly



$$A + C \xrightarrow{1} \quad C \xrightarrow{16.5} 2C \quad 2C \xrightarrow{0.5} C$$

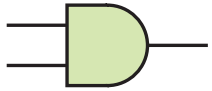
reaction networks





multicellular systems

4. Abstract Models
(components and algorithms)

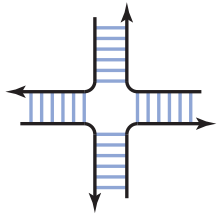


compile ↓ ↑ analyze

3. Chemical State Models
(reaction mechanism)

$A + B \rightleftharpoons C$

compile ↓ ↑ analyze



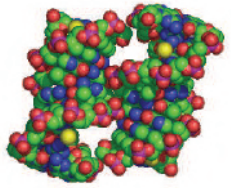
2. Secondary Structure Models
(structural mechanisms)

compile ↓ ↑ analyze

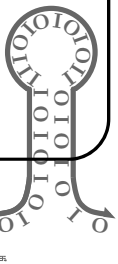
1. Primary Structure
(DNA sequence)

...ACGTGGCCA...

compile ↓ ↑ analyze



0. Molecular Execution
(laboratory implementation)



The question when we started, five years ago…

Are we only dreaming?

Can chemistry be programmed? At all? Is chemistry a genuine information technology? It sounds good, but what does it mean?

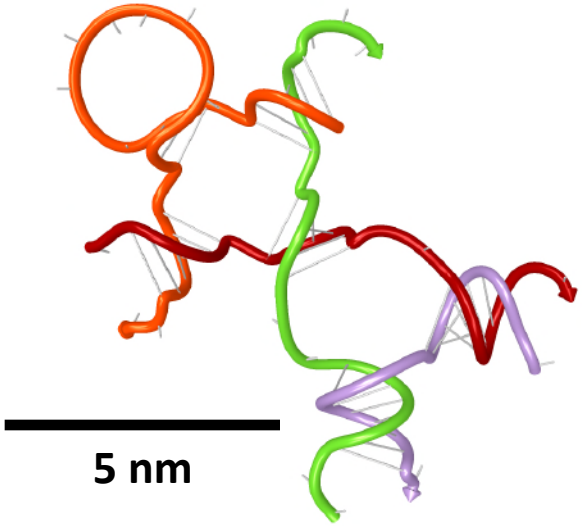
We need abstractions that work. We need systematic methods to design molecules and systems. We need "wet" synthesis and analysis methods that work – and work at a large scale.

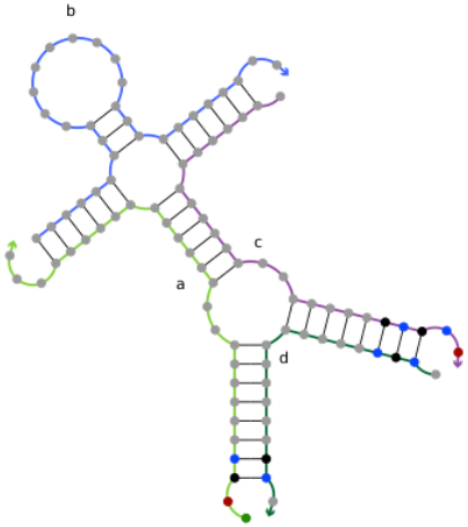
Should we wait for the biologists and bioengineers to give us working scalable molecular technology? Or should we make it happen ourselves?

No, we're not dreaming!

NUPACK: nucleic acid system design and analysis

- **Rigorous thermodynamic sequence design for structures**






```
material = rna
temperature[C] = 23.0
sodium[M] = 1.0

structure stickfigure =
    U2D8(U2D6(D6(U3+
      )D3U9D6(U2+U1))U2D8(U2+U1))U1

domain a = AUGC N23
domain b = N29
domain c = N20 GCGCU
domain d = N18

stickfigure.seq = a b c d
```

sequence structure

$$\Delta G(\phi, s) = \sum_{\text{loop} \in s} \Delta G(\text{loop})$$

Equilibrium probability

$$p(\phi, s) = \frac{1}{Q(\phi)} e^{-\Delta G(\phi,s)/k_B T}$$

Partition function

$$Q(\phi) = \sum_{s \in \Gamma} e^{-\Delta G(\phi,s)/k_B T}$$

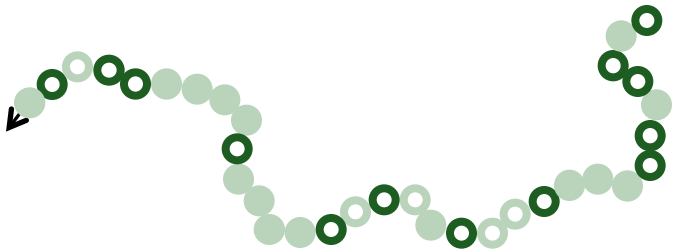
Dynamic programming: $\Theta(N^3)$

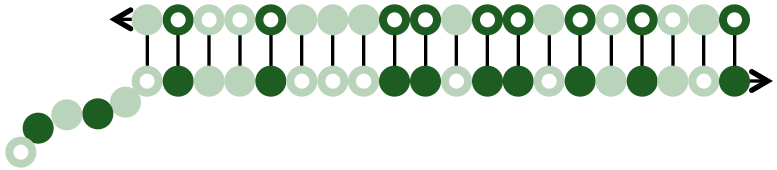
Average number of incorrectly paired nucleotides, given target s

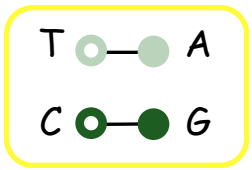
$$n(\phi, s) = N - \sum_{1 \le i \le N} P_{i,j}(\phi) S_{i,j}(s)$$

Zadeh, Steenberg, Bois, Wolfe, Pierce, Khan, Dirks, Pierce, J Comp Chem, 2011

DNA seesaw circuits: in vitro biomolecular circuitry

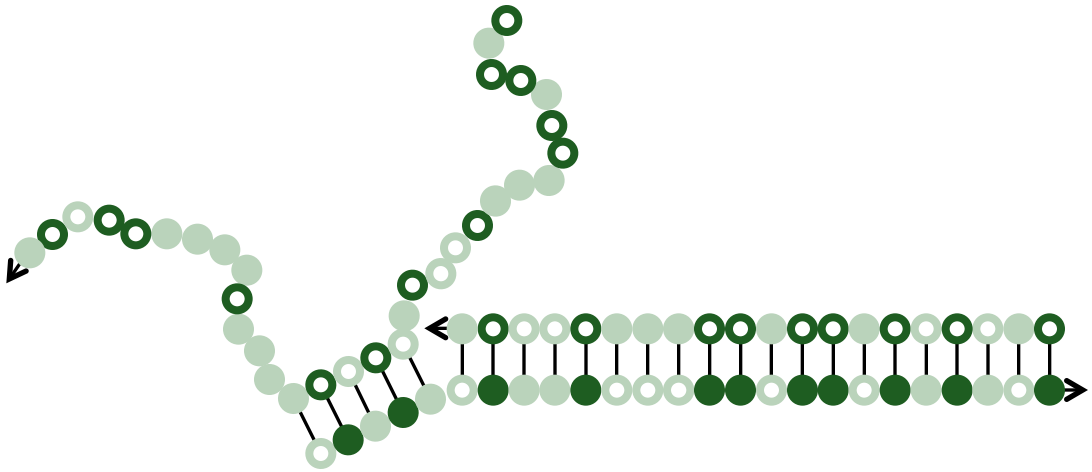


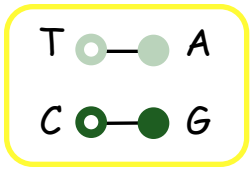




Qian, Winfree, Royal Society Interface, 2011

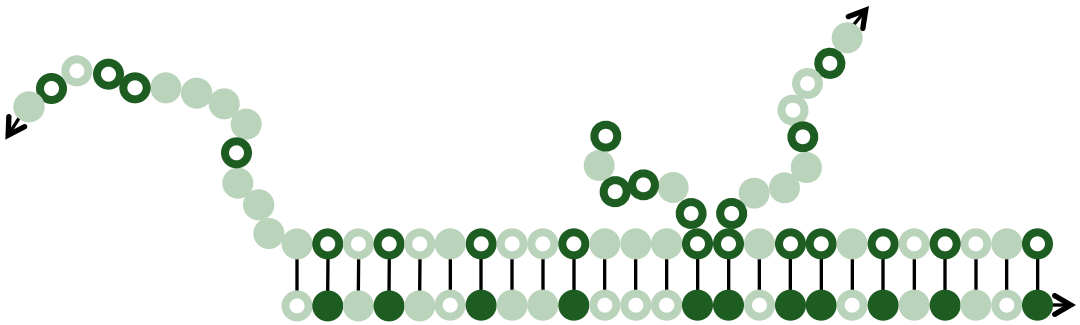
DNA seesaw circuits: in vitro biomolecular circuitry

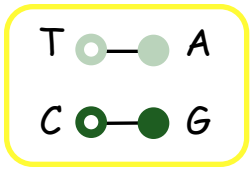




Qian, Winfree, Royal Society Interface, 2011

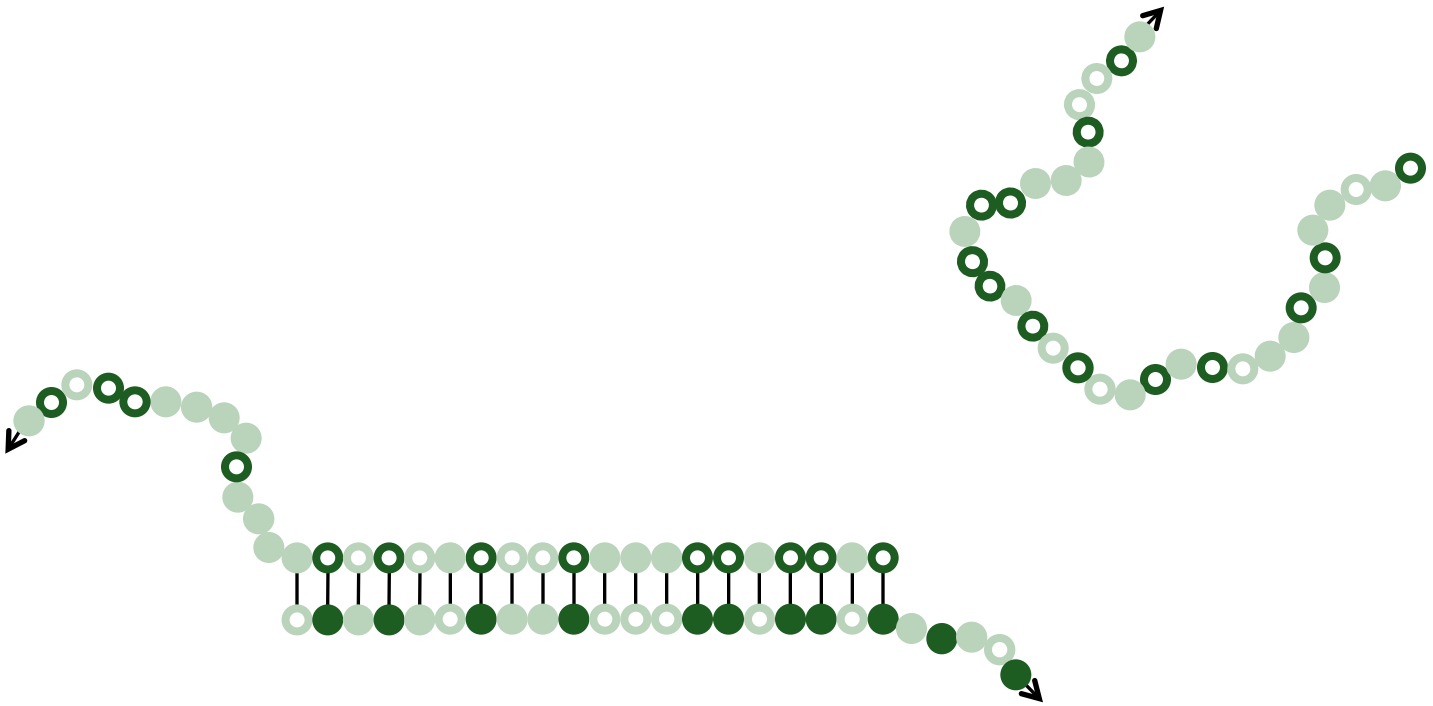
DNA seesaw circuits: in vitro biomolecular circuitry

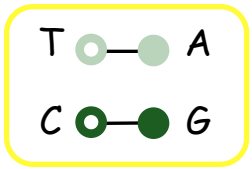




Qian, Winfree, Royal Society Interface, 2011

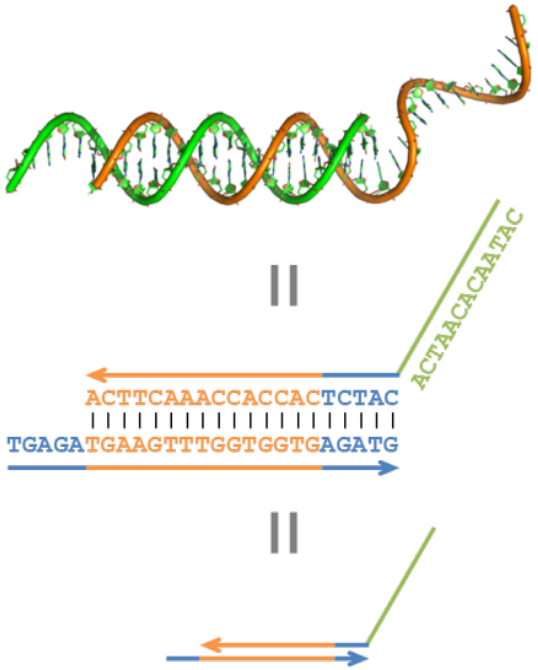
DNA seesaw circuits: in vitro biomolecular circuitry

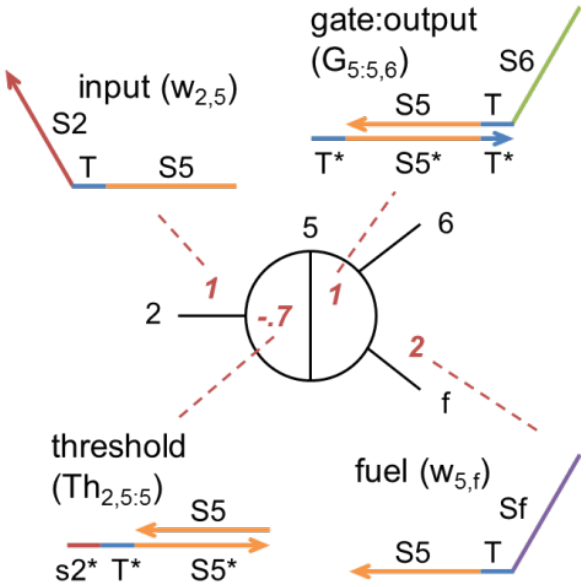


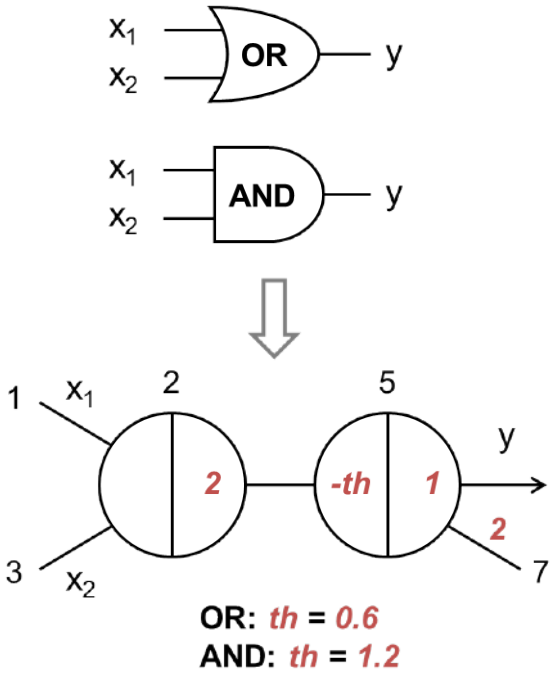


Qian, Winfree, Royal Society Interface, 2011

DNA seesaw circuits: in vitro biomolecular circuitry

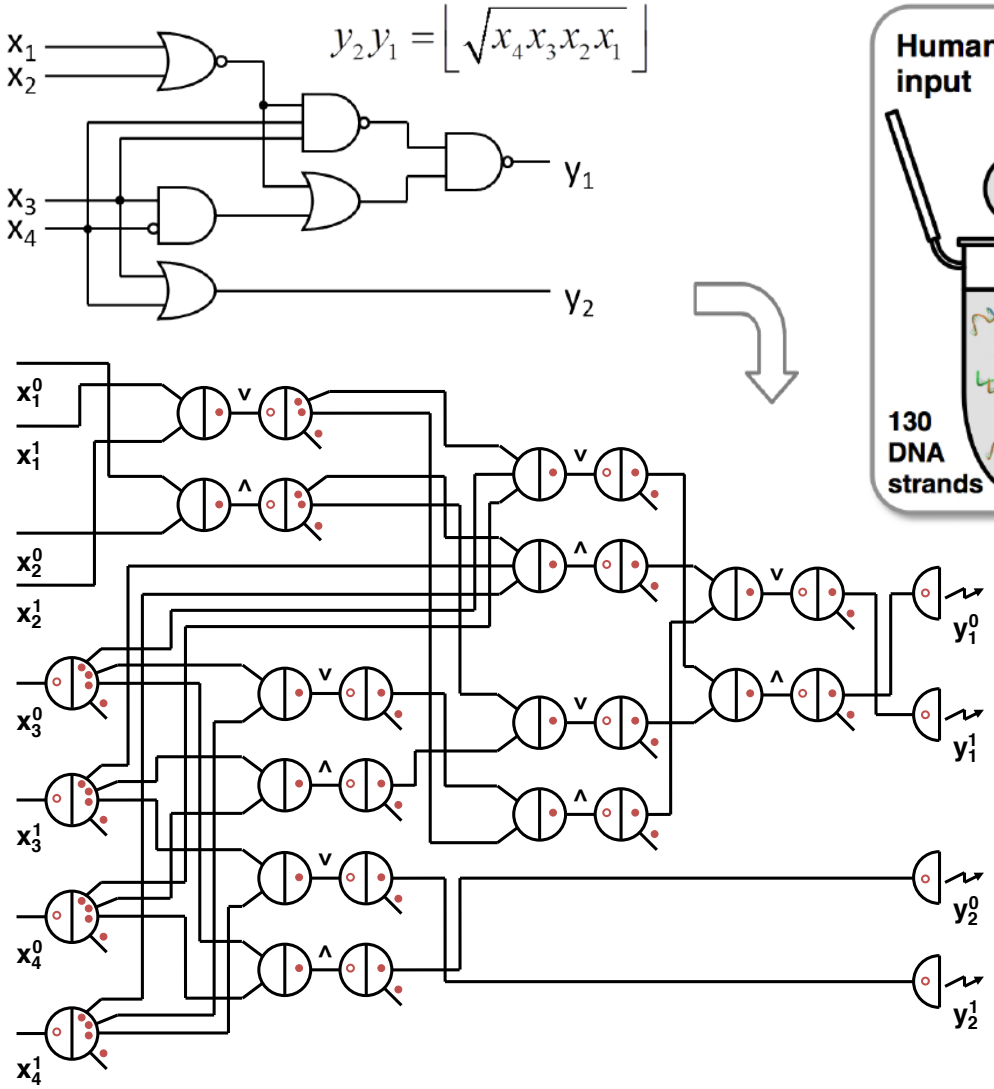






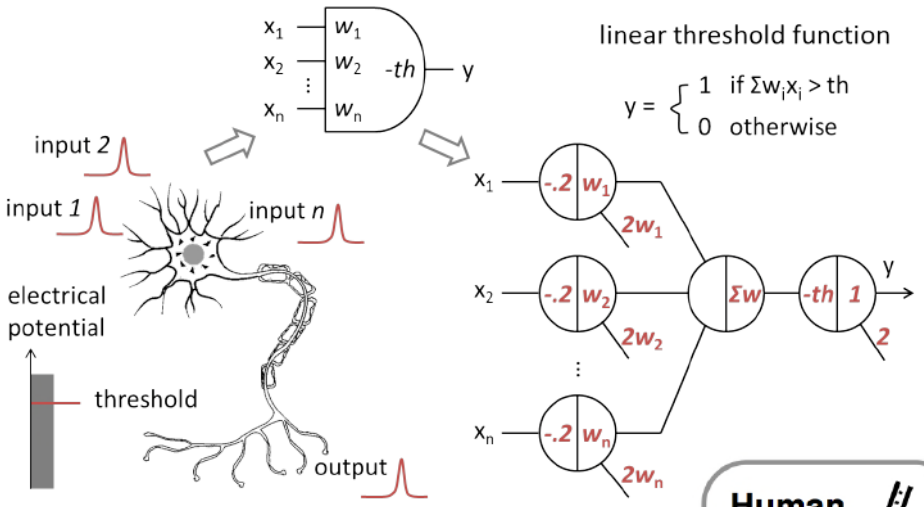
Qian, Winfree, Royal Society Interface, 2011

DNA seesaw circuits: in vitro biomolecular circuitry



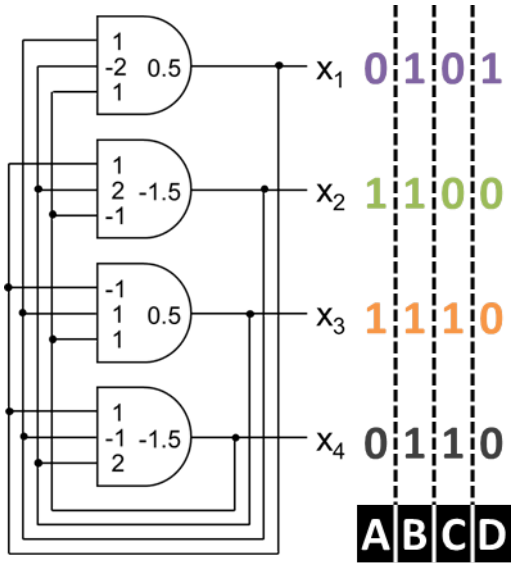
Qian, Winfree, Science, 2011

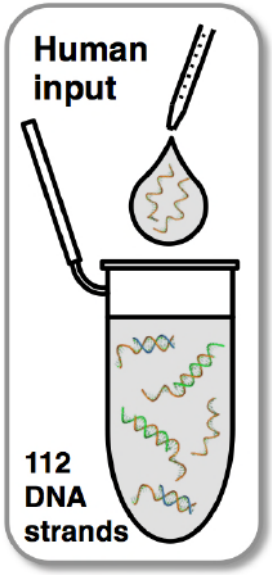
DNA seesaw circuits: in vitro biomolecular circuitry

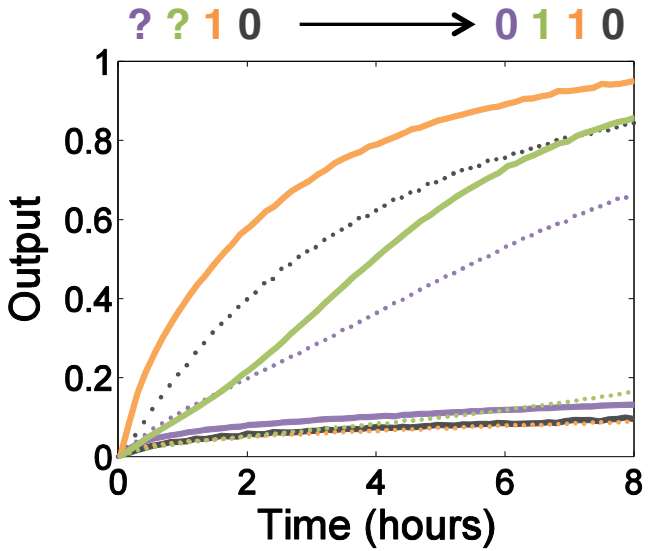


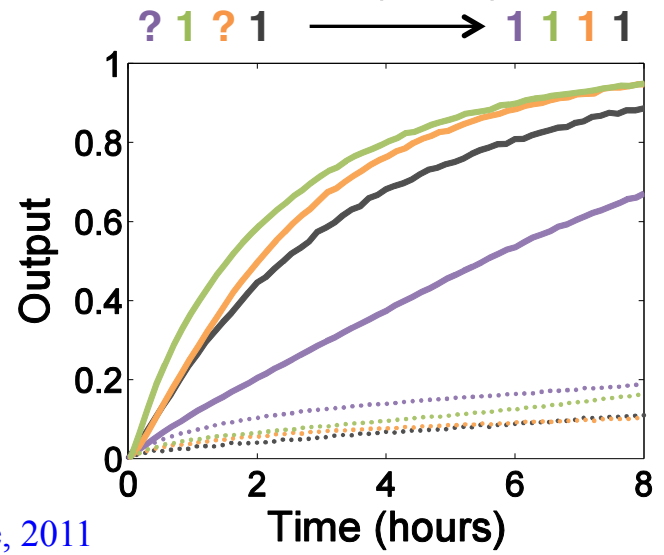
linear threshold function

$$y = \begin{cases} 1 & \text{if } \Sigma w_i x_i > th \\ 0 & \text{otherwise} \end{cases}$$



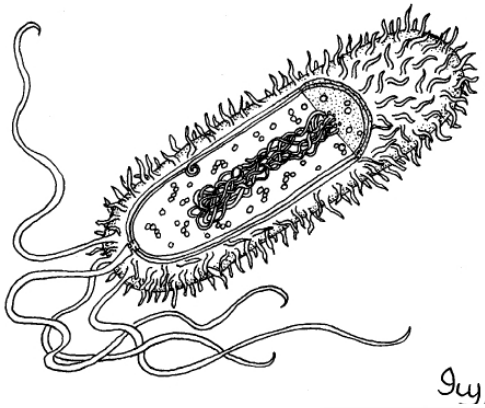


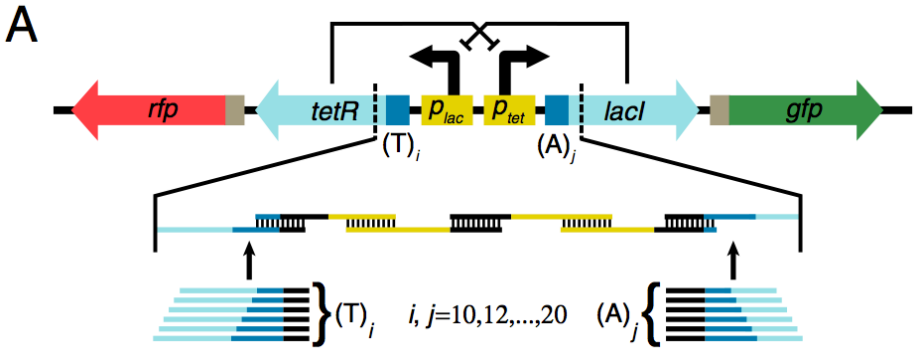


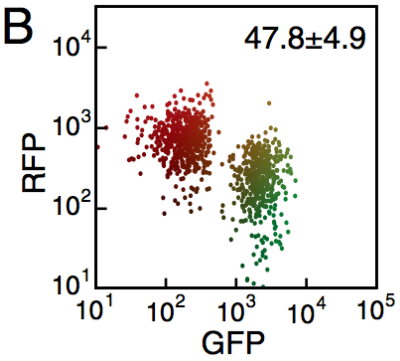


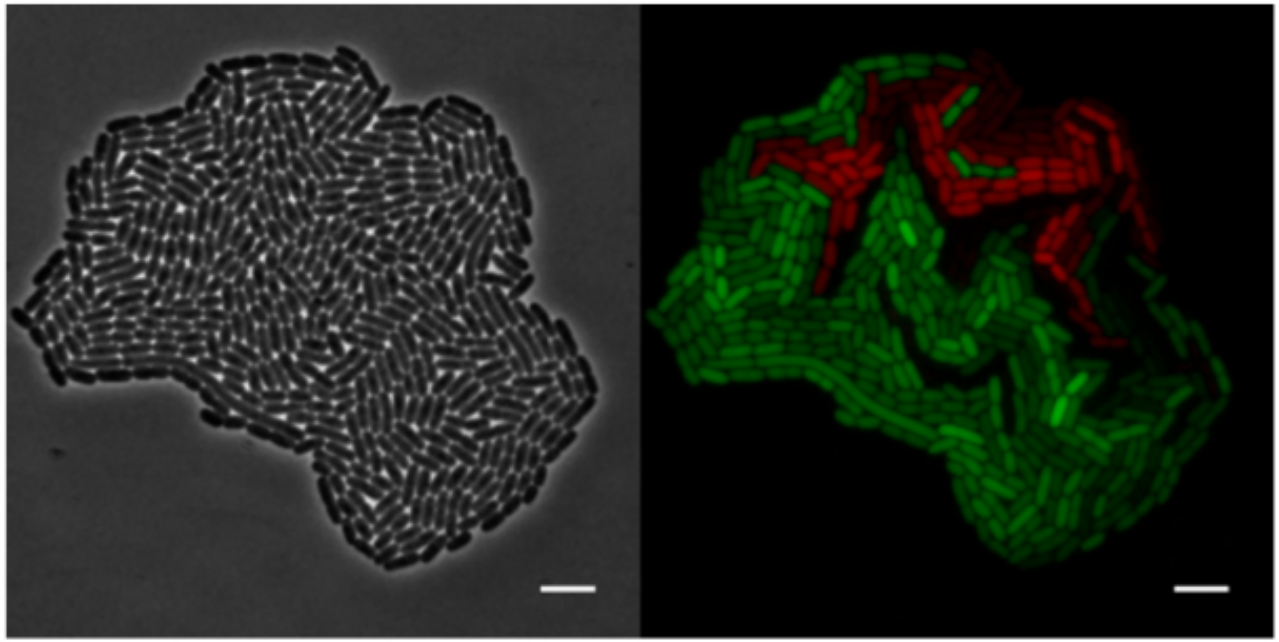
Qian, Winfree, Bruck, Nature, 2011

gro: a programming language for growing cells



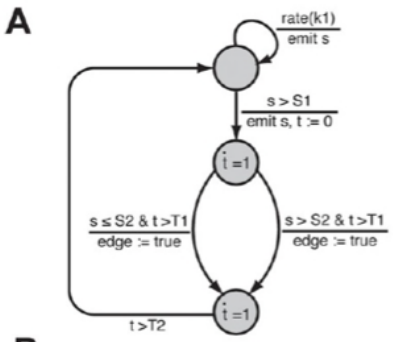






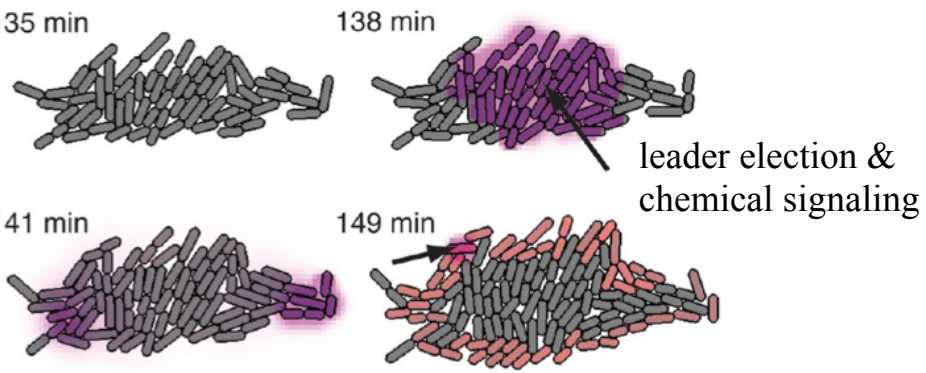
Egbert, Klavins, PNAS, 2012

gro: a programming language for growing cells

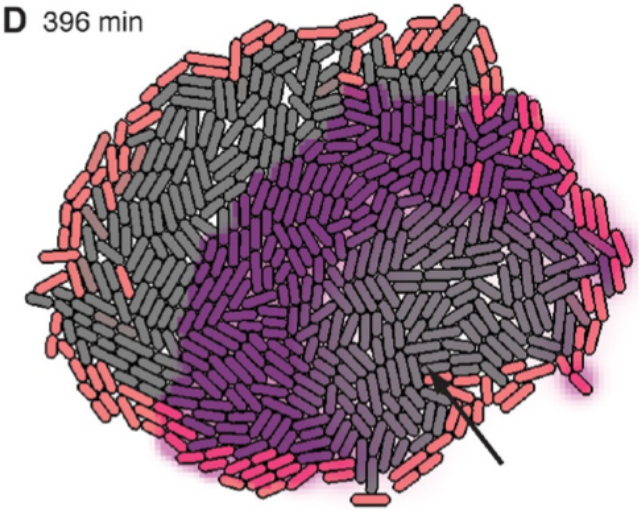


B

```
1  ahl := signal ( 1, 1 );
2
3  program edge() := {
4
5    rfp := 0;
6    edge := false;
7    r := [ t := T2];
8
9    rate ( k1 ) : {
10     emit_signal ( ahl, se )
11   }
12
13   r.t > T2 & get_signal ( ahl ) > S1: {
14     emit_signal ( ahl, se ),
15     r.t := 0
16   }
17
18   T1 - dt < r.t & r.t <= T1: {
19    edge := get_signal ( ahl ) < S2
20   }
21
22   true : {
23     r.t := r.t + dt
24   }
25
26 };
```



leader election &
chemical signaling



Jang, Oishi, Egbert, Klavins, ACS Synthetic Biology, 2012

Theoretical questions in molecular programming

Computation and construction

decisions, control, fabrication, memory

Program correctness

equivalence of high & low level specs

Abstraction hierarchy

programming languages and compilers

Complexity theory

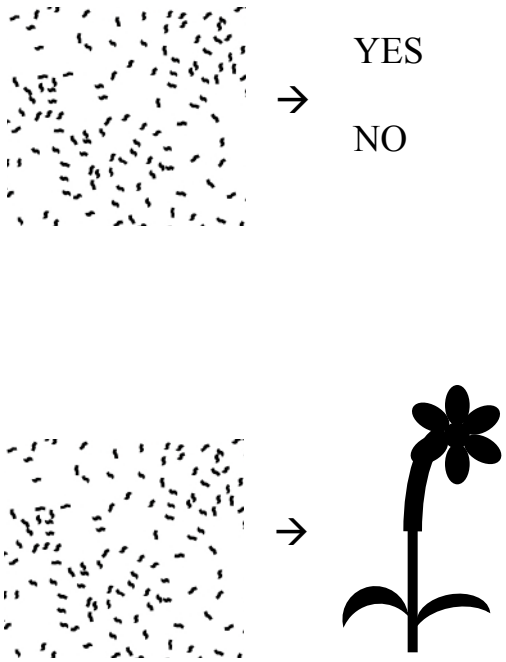
time, space, program size, energy

Exploiting randomness

designing stochastic behavior

Fault-tolerance

error correction, self-repair



Theoretical questions in molecular programming

Computation and construction

decisions, control, fabrication, memory

Program correctness

equivalence of high & low level specs

Abstraction hierarchy

programming languages and compilers

Complexity theory

time, space, program size, energy

Exploiting randomness

designing stochastic behavior

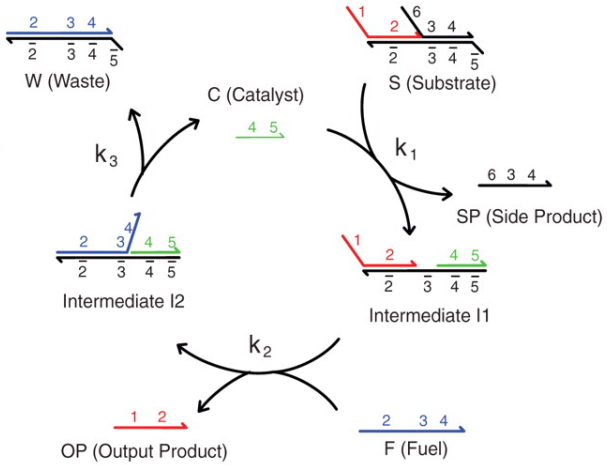
Fault-tolerance

error correction, self-repair

$S + F + C \rightarrow C + OP + SP$

?

=



Theoretical questions in molecular programming

Computation and construction
decisions, control, fabrication, memory

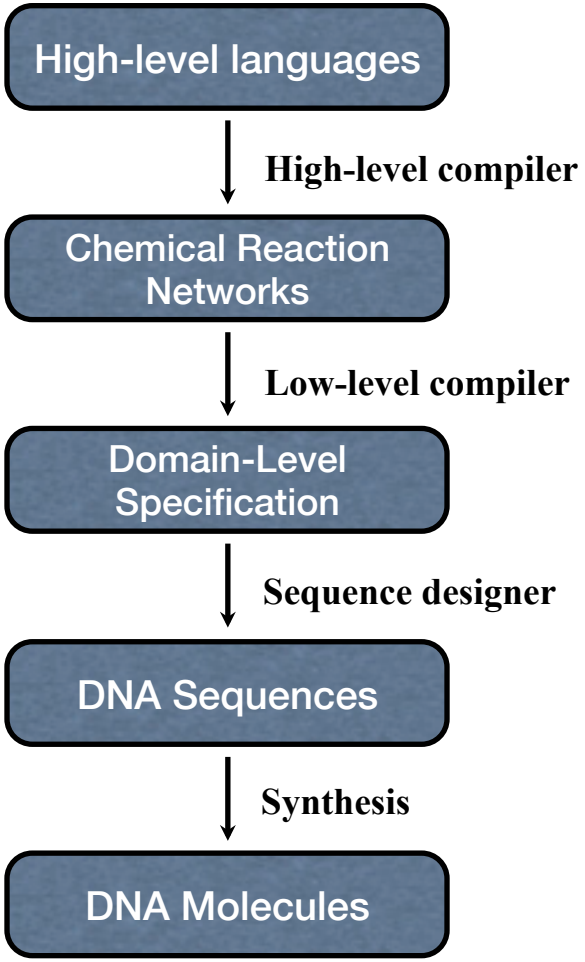
Program correctness
equivalence of high & low level specs

Abstraction hierarchy
programming languages and compilers

Complexity theory
time, space, program size, energy

Exploiting randomness
designing stochastic behavior

Fault-tolerance
error correction, self-repair



Theoretical questions in molecular programming

Computation and construction
decisions, control, fabrication, memory

Program correctness
equivalence of high & low level specs

Abstraction hierarchy
programming languages and compilers

Complexity theory
time, space, program size, energy

Exploiting randomness
designing stochastic behavior

Fault-tolerance
error correction, self-repair

Size n shape S, computed by g(n) state TM in time f(n) :

…can be self-assembled
from O(f(n)) passive tile types
in time O(n f(n))
with scale-up factor O(f(n)).

…or can be constructed by
molecular robots with
$O(g(n) + \log n)$ states, in time
$O(f(n) \log n + \log^2 n)$, with
scale-up factor 1.

Theoretical questions in molecular programming

Computation and construction

decisions, control, fabrication, memory

Program correctness

equivalence of high & low level specs

Abstraction hierarchy

programming languages and compilers

Complexity theory

time, space, program size, energy

Exploiting randomness

designing stochastic behavior

Fault-tolerance

error correction, self-repair

Theoretical questions in molecular programming

Computation and construction
decisions, control, fabrication, memory

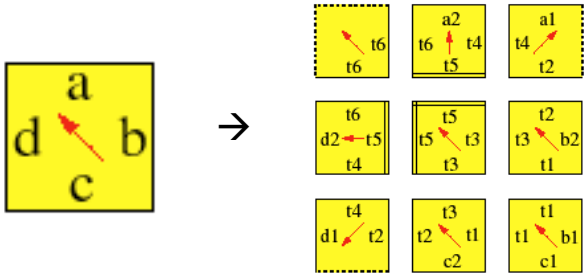
Program correctness
equivalence of high & low level specs

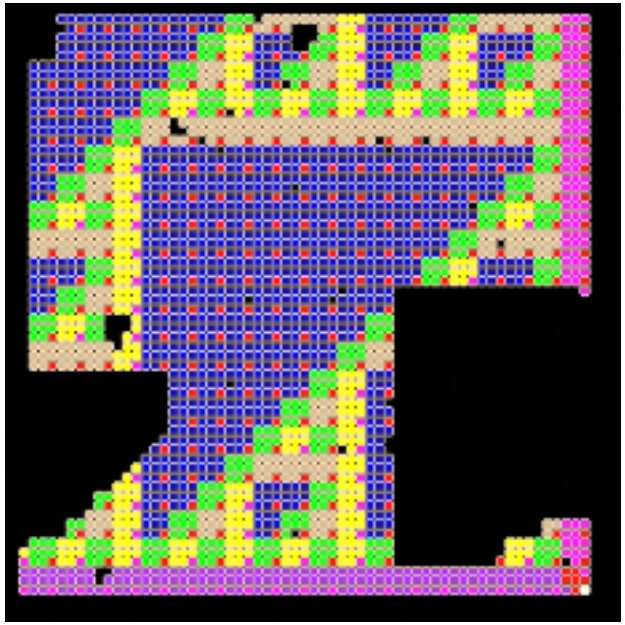
Abstraction hierarchy
programming languages and compilers

Complexity theory
time, space, program size, energy

Exploiting randomness
designing stochastic behavior

Fault-tolerance
error correction, self-repair





What if there were no Expedition, what then?

We wouldn't know if we're dreaming or not...

- **The Molecular Programming Project built a community**
  - Highly interdisciplinary – physics, chemistry, biology, computer science
  - Not yet recognized as "computer science" by many computer science departments
  - Having a strong, rich, enthusiastic ***distributed*** "department" helps recruit & educate students
  - The Expedition award provides moral support that the area is legitimate
  - Annual MPP workshops have helped coalesce a broader community

Education, Outreach, and Knowledge Transfer

Courses, books, teams

- Biomolecular Computation (Caltech)
- Biomolecular Engineering (Caltech)
- Systems Biology (Caltech)
- Information and logic (Caltech)
- Feedback Systems (Caltech)
- Mathematical Synthetic Biology (UW)
- Synthetic Biology Design Lab (UW)

Book: Biomolecular Feedback Systems
by Murray & Del Vecchio (on line)

Undergraduate competitions: BIOMOD, iGEM

People

2 high school students

33 undergraduates

25 graduate students

11 postdocs

3 programmers

8 alumni hired as faculty
MIT, Harvard, Caltech, Rice, JHU, UW, … (50% women)

Workshops, tutorials, conferences, tutorials

MPP annual workshops, 2009-2012
~40 MPP members, ~30 external

DNA Comp. & Molec. Prog., 2011

genelet wet lab tutorial, 2011

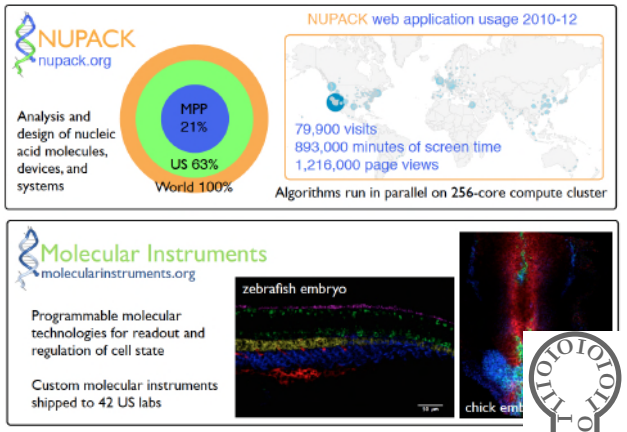
gro workshop, 2012

YouTube presentations: 68,000+ views

Local high school visiting days

Artistic collaborations

Tools and technologies



www.annerpino.com