Molecular Programming: Chemistry as a New Information Technology

Erik Winfree

Computer Science & CNS & Bioengineering,
California Institute of Technology
The DNA and Natural Algorithms (DNA) Group

National Science Foundation. Gordon and Betty Moore Foundation.
Biology is an information technology

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
How can we engineer molecular systems of comparable sophistication? Molecular programming!
Chemistry as an information technology

- Chemistry
  - Information-based chemistry
    - Life on Earth (biology)
  - Molecular programming
Information technologies

computers

computational biology

biology

synthetic biology

DNA
RNA
protein

computational chemistry

chemistry

molecular programming

A + B → X + Y
Y + Z ↔ YZ
Embedded computation in electro-mechanical systems

- Computers
- Appliances
- Vehicles
Embedded computation in chemical and biological systems

A → 0.03 → 2A
2A → 5×10^4 → A
B + A → 10^5 → 2B
B → 0.01 →
A + C → 10^5 →
C → 0.0165 → 2C
2C → 5×10^4 → C

Medical diagnostics & therapeutics

Controlling complex chemical synthesis

Synthesis of complex materials
Natural and artificial technology
Biomimetic technologies

Electro-mechanical technologies make it possible to create macroscopic robots with life-like autonomous behaviors.

Boston Dynamics Big Dog

Molecular technologies will make it possible to create molecular robotic systems with life-like autonomous behaviors.

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?

```plaintext
#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() );
```
Nucleic acid nanotechnology as a first step toward chemical information technology

- Information-based
- Programmable
- Algorithmic
- Computation and construction
The Brewery
(by Ann Erpino)
DNA as an Engineering Material

Ned Seeman
(New York University)

Wood.
Glue.
Wool.
Silk.
Leather.
DNA.

single-stranded double-stranded

AGCTTGCAATGCTATTCGAAGACT
AGCGCAATTAGCATTCGAAGACT

DNA as an Engineering Material

Ned Seeman
(New York University)

Wood.
Glue.
Wool.
Silk.
Leather.
DNA.

single-stranded double-stranded

AGCTTGCAATGCTATTCGAAGACT
AGCGCAATTAGCATTCGAAGACT

DNA as an Engineering Material

Ned Seeman
(New York University)

Wood.
Glue.
Wool.
Silk.
Leather.
DNA.
DNA as an Engineering Material

Ned Seeman
(New York University)

Wood.
Glue.
Wool.
Silk.
Leather.
DNA.

Seeman, J. Theor. Biol. 1982
DNA as a Computing Substrate

Len Adleman (USC)

Gears.
Springs.
Fluids.
Vacuum tubes.
Transistors.
DNA.

Growth of design complexity in DNA nanotechnology and DNA computing

- **Folding**
- **Self-Assembly**
- **Circuits**
- **Machines**

DNA 4-arm junctions (Seeman, 1982)

- Doubles every 3 years
Programmable information-based chemistry

RNA biology

DNA nanotechnology
Architectures for structural molecular programs

a) DNA origami
   - Rothemund, 2006
   - Shih & Lin, 2010

b) Single-strand tiles (SST)
   - Wei et al, 2012
   - Ke et al, 2012

c) Algorithmic self-assembly
   - Barish et al, 2010

Constantine Evans, 2014
What kinds of dynamical behaviors are nucleic acid systems capable of?

- goes to completion…?
- oscillates…?
- does something complex…?
DNA strand displacement circuits

Lulu Qian, Erik Winfree, Shuki Bruck, *Nature*, 2011
DNA strand displacement circuits

Lulu Qian, Erik Winfree, *Science*, 2011
DNA strand displacement circuits

1x = 50 nM = $3 \cdot 10^{13}$ copies per milliliter

Lulu Qian, Erik Winfree, *Science*, 2011
A seesaw catalyst with threshold

Gate: \( G_{5:56} \)

Threshold: \( T_{25:5} \)

Fuel: \( w_{57} \)

Output reporter

\[ |T| = 5 \quad |Si| = 15 \]
OR / AND gates

\[ \text{Output} = \begin{cases} 1 & \text{if } x_1 = 1 \text{ or } x_2 = 1 \\ 0 & \text{otherwise} \end{cases} \]

\[ \text{Output} = \begin{cases} 1 & \text{if } x_1 \land x_2 = 1 \\ 0 & \text{otherwise} \end{cases} \]
A square root circuit

\[ y_2y_1 = \sqrt{x_4x_3x_2x_1} \]
A square root circuit

\[ y_2 y_1 = \left[ \sqrt{x_4 x_3 x_2 x_1} \right] \]
A square root circuit

\[ y_2y_1 = \left[ \sqrt{x_4x_3x_2x_1} \right] \]

130 DNA strands (15-33 bases each)  74 initial DNA species
A square root circuit

\[x_4x_3x_2x_1=0000 \quad y_2y_1=00\]

\[x_4x_3x_2x_1=0001 \quad y_2y_1=01\]

\[x_4x_3x_2x_1=0100 \quad y_2y_1=10\]

\[x_4x_3x_2x_1=1001 \quad y_2y_1=11\]
A square root circuit
Linear Threshold (LT) gate

\[ x_i \in \{0, 1\} \]

\[ F(x) = -th + \sum_{i=1}^{n} w_i x_i \]

\[ f(x) = \begin{cases} 
1 & \text{if } F(x) \geq 0 \\
0 & \text{if } F(x) < 0 
\end{cases} \]
Artificial neural network

Hopfield associative memory

Hopfield associative memory

A “read your mind” game

Q1: Did the scientist study neural networks?
Q2: Was the scientist British?
Q3: Was the scientist born in the 20th century?
Q4: Was the scientist a mathematician?

Answers: Yes (1), No (0) or I don’t know (?)

0 1 1 0  Rosalind Franklin
1 1 1 1  Alan Turing
0 0 1 1  Claude Shannon
1 0 0 0  Santiago Ramon y Cajal

Lulu Qian, Erik Winfree, Jehoshua Bruck, Nature, 2011
Hopfield associative memory

112 DNA strands (15-33 bases each)
72 initial DNA species
Human: The scientist I am thinking of was born in the 20th century (1) but was not a mathematician (0).

DNA associative memory: The scientist you are thinking of didn’t study neural networks (0) and was British (1), so I guess she is Rosalind Franklin (0 1 1 0).
Hopfield associative memory

A “read your mind” game

Q1: Did the scientist study neural networks?
Q2: Was the scientist British?
Q3: Was the scientist born in the 20th century?
Q4: Was the scientist a mathematician?

Answers: Yes (1), No (0) or I don’t know (?)

<table>
<thead>
<tr>
<th>Answer</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 0</td>
<td>Rosalind Franklin</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>Alan Turing</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>Claude Shannon</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>Santiago Ramon y Cajal</td>
</tr>
</tbody>
</table>

Human: The scientist I am thinking of was British (1) and a mathematician (1).

DNA associative memory: The scientist you are thinking of studied neural networks (1) and was born in the 20th century (1), so I guess he is Alan Turing (1 1 1 1).
Hopfield associative memory

A “read your mind” game

Q1: Did the scientist study neural networks?
Q2: Was the scientist British?
Q3: Was the scientist born in the 20th century?
Q4: Was the scientist a mathematician?

Answers: Yes (1), No (0) or I don’t know (?)

0 1 1 0 Rosalind Franklin
1 1 1 1 Alan Turing
0 0 1 1 Claude Shannon
1 0 0 0 Santiago Ramon y Cajal

Human: The scientist I am thinking of was British (1) but was not born in the 20th century (0).

DNA associative memory: There’s wrong information that you provided, I cannot recognize this scientist (x x x x).
Toss a mixture of DNA in a test tube... What can it do? What can’t it do?
What kinds of dynamical behaviors are nucleic acid systems capable of?

- goes to completion…?
- oscillates…?
- does something complex…?
Some behaviors of simple chemical systems
(mass action, well-mixed)

- **stabilization**
- **oscillation**
- **chaos**

open systems: no conservation of mass/energy
Some behaviors of simple chemical systems (mass action, well-mixed)

Chemical reaction networks (CRNs) as a programming language…
…use the formalism prescriptively, not descriptively…

- Mathematically equivalent to Petri Nets, Vector Addition Systems, etc.
  - e.g. Petri (1966); Karp (1969); Goss & Peccaud (1998)
- Can implement arbitrary sequential digital logic circuits
  - e.g. Magnasco (1997)
- Can implement (nearly) arbitrary dynamical system behaviors
  - e.g. Korzuhin (1967); Klonowski (1983)
- Can simulate Turing machine computations with arbitrarily small error
  - e.g. Soloveichik, Cook, Winfree, Bruck (2008); Cummings, Doty, Soloviechik (2014)
- Etc. etc…

But does every formal chemical reaction network exist?
Compiling CRNs into DNA

CRN program:

\[ A \xrightarrow{30} 2A \]
\[ 2A \xrightarrow{0.5} A \]
\[ B + A \xrightarrow{1} 2B \]
\[ B \xrightarrow{10} \]
\[ A + C \xrightarrow{1} \]
\[ C \xrightarrow{16.5} 2C \]
\[ 2C \xrightarrow{0.5} C \]

Initial conditions:

5 A's
3 B's
10 C's

Soloveichik, Seelig, Winfree, PNAS, 2010
Some behaviors of simple chemical systems (mass action)
A compiler & verification hierarchy

High-level languages
- Digital circuitry, analog differential equations, Turing machines, finite state machines, ...

Chemical reaction networks
- low-level compiler
- COPASI, ...

Domain-level specification
- reaction enumerator
- Visual DSD Peppercorn

DNA sequences
- sequence design
- NUPACK, ...

DNA molecules
- synthesis

Chemical reaction network behavior
- proof/disproof of correctness
- Nuskell

Domain-level reaction networks
- probabilistic assessment
- KinDA

Sequence-level markov chains
- secondary structure simulator
- Kinefold Multistrand

Experimental data
- laboratory experiments
High-level languages

Digital circuitry, analog differential equations, Turing machines, finite state machines, ...

Chemical reaction networks

semantics

COPASI, ...

Chemical reaction network behavior

proof/disproof of correctness

Domain-level specification

reaction enumerator

Visual DSD
Peppercorn

Domain-level reaction networks

probabilistic assessment

Kinefold
Multistrand

Sequence-level markov chains

behavioral assessment

DNA sequences

sequence design

NUPACK, ...

DNA molecules

synthesis

Experimental data

laboratory experiments

A compiler & verification hierarchy
Abstractions for molecular programs

NUPACK: rigorous thermodynamic and kinetic sequence design and analysis

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

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

Equilibrium probability

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

Partition function

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

Average number of incorrectly paired nucleotides, given target \( s \)

\[
n(\phi, s) = N - \sum_{1 \leq i \leq N} P_{i,j}(\phi)S_{i,j}(s)
\]

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

Wolfe, Pierce, ACS Synthetic Biology, 2015
Nuskell: A verifying compiler for domain-level DNA implementations of chemical reaction networks

"Compiling and verifying DNA-based chemical reaction network implementations"
Seung Woo Shin, Master's Thesis (2011)

"A bisimulation approach to verification of molecular implementations of formal chemical reaction networks"
Qing Dong, Master's Thesis (2012)

"Verifying Chemical Reaction Network Implementations: A Pathway Decomposition Approach"
Verification pipeline

Target CRN

\[ X_3 + X_4 \xrightarrow{k_1} X_5 \]
\[ X_5 \xrightarrow{k_2} X_1 \]
\[ X_1 + X_2 \xrightarrow{k_3} X_3 \]

Claimed implementation

Implementation CRN

\[ X_3 + g_1 \xrightarrow{k_4} i + g_2 \]
\[ i + g_2 \xrightarrow{k_5} X_3 + g_1 \]
\[ i + x_4 \xrightarrow{k_6} j + w_1 \]
\[ j + g_3 \xrightarrow{k_7} X_5 + w_2 \]
\[ X_5 + g_4 \xrightarrow{k_8} k + w_3 \]
\[ k + g_5 \xrightarrow{k_9} X_1 + w_4 \]
\[ X_1 + g_6 \xrightarrow{k_{10}} l + g_7 \]
\[ l + g_7 \xrightarrow{k_{11}} X_1 + g_6 \]
\[ l + X_2 \xrightarrow{k_{12}} m + w_5 \]
\[ m + g_8 \xrightarrow{k_{13}} X_3 + w_6 \]
Multistrand: stochastic simulation of the kinetics of multiple interacting nucleic acid strands

"Stochastic Simulation of the Kinetics of Multiple Interacting Nucleic Acid Strands"

"Stochastic Simulation of the Kinetics of Multiple Interacting Nucleic Acid Strands"
Joseph M. Schaeffer, Chris Thachuk, and Erik Winfree, LNCS 9211 (2015)

Hands-on live demo: http://www.multistrand.org
\[
\Delta G_{\text{box}}(s) = \sum_{c \in s} \Delta G^*(c)
\]

\[
\Delta G^*(c) = \left( \sum_{\text{loop} \in c} \Delta G(\text{loop}) \right) + (L_c - 1) \ast (\Delta G_{\text{assoc}} + \Delta G_{\text{volume}})
\]

* same energy model as NUPACK 3.0
Kinetics: continuous-time markov chain representing a random walk on the energy landscape for The Box.

Elementary steps: single base pairs form and break.

Move selection:

\[
Pr(m) = \frac{k_{im}}{\sum_j k_{ij}}
\]

\[
Pr(\Delta t) = \sum_j k_{ij} * e^{-\sum_j k_{ij} \Delta t}
\]

Detailed balance:

\[
\frac{k_{ij}}{k_{ji}} = e^{-\frac{\Delta G(j) - \Delta G(i)}{RT}}
\]

\[
Pr(i) = \frac{e^{-\Delta G_{box}(i)/RT}}{Q_{kin}}
\]

\[
Q_{kin} = \sum_i e^{-\Delta G_{box}(i)/RT}
\]
Kinetics:
continuous-time markov chain representing a random walk on the energy landscape for The Box.

Elementary steps:
single base pairs form and break.

Move selection:

Detailed balance:
Simulating a single strand
**KinDA**: design and analysis of kinetics for domain-level DNA strand displacement systems

"Automated Sequence Analysis for Domain-level DNA Strand Displacement Systems"
Joseph Berleant, Chris Berlind, Joseph Schaeffer, Niranjan Srinivas, Chris Thachuk, Erik Winfree (in preparation)
Case Study: Entropy-driven catalyst

**Domain** | **Published** | **Random**
---|---|---
1 | CTTTCCTACA | AAACGCGAAA
2 | CCTACGTCTCCAACCTAATACG | CTTTAATATGATTGAGCTCAGT
3 | CCCT | TTCC
4 | CATTCATACCCTACG | AGTCTTAATTGACCTACA
5 | TCTCCA | CAAAGA
6 | CCACATAACAATCATATT | ACAAGAGCCTTGATTA
Case Study: Entropy-driven catalyst

PUBLISHED

1.2x10^7 M^{-1}s^{-1}

4.4x10^6 M^{-1}s^{-1}

RANDOM

WASTE

18 s^{-1}

73 s^{-1}

INPUT

OUTPUT

CCTACGTCCTCCACTAACCTACGGCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).......................

CTTTAATATGGATTTTGCACCAGTTTCCAGTCTTAATTGACCCA

........(((........))).......(((......)))...

CCTACGTCTCCAACTAACTTACGGCCCTCATTCAATACCCTACG

........(((.............))).....................
Programming Dynamical Behaviors in Chemical Systems using DNA Strand Displacement Cascades

Niranjan Srinivas, James Parkin, Georg Seelig, Erik Winfree, David Soloveichik (in preparation)
A compiler & verification hierarchy

High-level languages
- Digital circuitry, analog differential equations, Turing machines, finite state machines, ...

Chemical reaction networks
- Try it for: $B + A \rightarrow 2B$
- $C + B \rightarrow 2C$
- $A + C \rightarrow 2A$

Domain-level specification
- low-level compiler
  - Nuskell
- sequence design
  - NUPACK, ...

DNA sequences
- synthesis

DNA molecules

Chemical reaction network behavior

Domain-level reaction networks

Sequence-level markov chains

Experimental data
- proof/disproof of correctness
  - Nuskell
- probabilistic assessment
  - KinDA
- behavioral assessment

Try it for:
- $B + A \rightarrow 2B$
- $C + B \rightarrow 2C$
- $A + C \rightarrow 2A$
A test tube of synthetic DNA molecules

<table>
<thead>
<tr>
<th>Concentration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 nM</td>
<td>Ap</td>
</tr>
<tr>
<td>10 nM</td>
<td>Thr_A</td>
</tr>
<tr>
<td>10 nM</td>
<td>Br</td>
</tr>
<tr>
<td>10 nM</td>
<td>Thr_B</td>
</tr>
<tr>
<td>13 nM</td>
<td>Cj</td>
</tr>
<tr>
<td>10 nM</td>
<td>Thr_C</td>
</tr>
<tr>
<td>100 nM</td>
<td>BACK_BA</td>
</tr>
<tr>
<td>100 nM</td>
<td>REACT_BABr</td>
</tr>
<tr>
<td>75 nM</td>
<td>HELPER_BB_s</td>
</tr>
<tr>
<td>25 nM</td>
<td>HELPER_CAT_BB_s</td>
</tr>
<tr>
<td>100 nM</td>
<td>PROD_ABrBs</td>
</tr>
<tr>
<td>100 nM</td>
<td>BACK_AC</td>
</tr>
<tr>
<td>100 nM</td>
<td>REACT_AC_Ap</td>
</tr>
<tr>
<td>75 nM</td>
<td>HELPER_AA_q</td>
</tr>
<tr>
<td>25 nM</td>
<td>HELPER_CAT_AA_q</td>
</tr>
<tr>
<td>100 nM</td>
<td>PROD_BCAq</td>
</tr>
<tr>
<td>100 nM</td>
<td>PROD_CAq</td>
</tr>
</tbody>
</table>
An enzyme-free oscillator: the Displacillator

B + A → 2B
C + B → 2C
A + C → 2A

Rock-Paper-Scissors CRN

Niranjan Srinivas, James Parkin, Georg Seelig, Erik Winfree, David Soloveichik (in preparation)
Toss a mixture of DNA in a test tube… What can it do? What can’t it do?
What can information-based chemistry do?

- chemistry
- information-based chemistry
- molecular programming
- Life on Earth (biology)
Why Molecular Programming?

Chemistry will be the new information technology of the 21\textsuperscript{st} C

– Information encoded in synthetic molecules can direct processes such as folding, self-assembly, circuitry, and machinery, thereby providing programmable control of a wide range of chemical systems.

It will transform industry, much as electronics did in the 20\textsuperscript{th} C

– Potential applications synthesizing programmable materials, devices, diagnostics, therapeutics… anything that chemistry can do… that will be empowered by pervasive embedded information processing and programmable behaviors.