DNA-Based Programmable Molecular Devices

John Reif
Dept CS
Duke University

Reif’s DNA Self-Assembly Group

Current PhD Graduate Students
Hieu Bui
Reem Mokhtar
Tianqi Song
Shalin Shah
Xin Song
Dan Fu

Visiting Scholars:
Abeer Eshra

Prior Recent PhD Graduate Students
Nikhil Gopalkrishnan
Peng Yin
Harish Chandran
Harish Chandran
Urmic Majumder
Sudhanshu Garg
Feynman’s Ill-Conceived Top-Down Approach to Nanotechnology

Feynman Suggested a Top-Down Approach to Nanotechnology

Feynman (“Plenty of room at the bottom”, 1959):
• Can the doctor be swallowed? (Albert Hibbs)
• Can we build tiny factories that can arrange atoms the way we want?
• Can we write the 24 volumes of the Encyclopedia Brittanica on the head of a pin?

⇒ Not Chemically Feasible at the Molecular Scale

“This fact - that enormous amounts of information can be carried in an exceedingly small space - is, of course, well known to the biologists, and resolves the mystery which existed before we understood all this clearly, of how it could be that, in the tiniest cell, all of the information for the organization of a complex creature such as ourselves can be stored. All this information—whether we have brown eyes, or whether we think at all, or that in the embryo the jawbone should first develop with a little hole in the side so that later a nerve can grow through it - all this information is contained in a very tiny fraction of the cell in the form of long-chain DNA molecules in which approximately 50 atoms are used for one bit of information about the cell.”
Bottom-Up Self-Assembly in Nature

Spontaneous organization of components into stable superstructures due to local interactions

From microscopic living cells to gigantic galaxies

Figure 3-25. Molecular Biology of the Cell, 4th Edition.
Introduction to DNA Self-Assembly
Overview

• **Why do self-assembly using DNA?**

1. **Natural nanoscale material**

2. **Ability to carry information can be exploited in self-assembly process**

3. **Well established base-pairing model in which the stability of a base-pair depends on their identity (A-T, C-G)**
Key to DNA Self-Assembly

Hybridization

3’ T T G T T T A A C C T 5’
5’ A A C A A A T T G G A 3’

3’ T T G T T T A A C C T 5’
5’ A A C A A A T T G G A 3’
What is DNA Self-Assembly?

Programming DNA strands to organize themselves into nanoscale shapes, patterns, and devices through Watson-Crick base-pairing.
DNA Nanotechnology via Self-Assembly

Seeman 1982:
• “It is possible to generate sequences of oligomeric nucleic acids which will preferentially associate to form migrationally immobile junctions, rather than linear duplexes, as they usually do.”

Some results of DNA self-assembly

NYU 1991
The Electrophoretic Properties Of A DNA Cube And Its Substructure Catenanes : Mao And Seeman

Purdue 2005
Self-assembly Of Hexagonal DNA Two-dimensional (2D) Arrays: He, Chen, Liu, Ribbe, And Mao

Caltech 2004
Algorithmic Self-assembly Of DNA Sierpinski Triangles: Rothemund, Papadakis, Winfree

Folding DNA To Create Nanoscale Shapes And Patterns: Rothemund 2006

2009
Harvard
Self-assembly Of DNA Into Nanoscale Three-dimensional Shapes: Douglas, Dietz, Liedl, Hogberg, Graf, Shihi

Duke 2003
Unpublished Data: Majumder, Reif

2006
Finite-size, Fully-addressable DNA Tile Lattices Formed By Hierarchical Assembly Procedures : Park, Pistol, Ahn, Reif, Lebeck, Dwyer, Labe

2003
4x4 DNA Tile And Lattices: Characterization, Self-assembly And Metallization Of A Novel DNA Nanostructure Motif : Yan, Park, Finkelstein, Reif And Labe
The Big Picture

• What does this Chemistry (DNA Nanostructures & DNA Hybridization Reactions) have to do with Computer Science (CS)?

• Translating CS to the bio/nanoscale:
  
  – **Molecular Patterning:** We can design DNA to self-assemble into patterned in a programmable way.
  
  – **Molecular Computers:** We can design DNA to perform programmed Computation at the Molecular Scale.
  
  – **Molecular Motors and Robots Motion:** We can use these components to perform actions and movements.
Milestones for DNA Computation, NanoAssembly & Robotics

• 1994: Adelman solved a combinatorial optimization problem (the Hamiltonian path problem) using DNA

• 2000: Mao, LaBean, Reif and Seeman demonstrate first molecular computations via DNA tile self-assembly.

• 2000: Yurke and Turberfield demonstrate a DNA “tweezer” robotic device

• 2003: Yan and Reif give first experimental demonstration of programmable molecular-scale patterning of DNA nanostructures

• 2004: Yin, Turberfield, and Reif demonstrate the first autonomous DNA enzymatic walker

• 2004: Rothemund and Winfree (CS) demonstrate 2D algorithmic self-assembly of DNA Sierpinski Triangles

• 2006: Rothemund demonstrates a general technique “DNA Origami” for self-assembling DNA nanostructures with high yield

• 2009: William Shih extends DNA Origami to three dimensions

• 2010: Stephanovic programs a robotic DNAzyme spider that can walk and respond to instructional signals on a DNA landscape

• 2011: Qian and Winfree calculate square root using DNA circuit

• 2006: Tiaqi and Reif demonstrate DNA circuits for analog computation.
Design & Experimental Demonstration of DNA Tiles and Lattices
DNA tiles

- **Anti-parallel crossovers:**
  - cause a reversal in direction of strand propagation through the tile following exchange of strand to a new helix.

- **Pads:**
  - Tiles have sticky ends that preferentially match the sticky ends of certain other DNA tiles.
  - The sticky ends facilitate the further assembly into tiling lattices.
  - Total of 4 Pads of single stranded DNA at ends.

**Lattice Self-Assembly:**
- Driven by Watson-Crick base pairing: $A \leftrightarrow T$ & $C \leftrightarrow G$
- Leads to energy minimization of the final structure
  - Base pairing and base stacking
- Programmability:
  - AGTGC sticks to GCACT (reverse complement)

**Self-Assembly from DNA strands, to Tiles, to Lattices**
TX tiles


• TX tiles – extension of the DX tile/ Has three helices made of 4 strands
Unique Sticky Ends on DNA tiles. Input layers can be assembled via unique sticky-ends at each tile joint thereby requiring one tile type for each position in the input layer.

Tiling self-assembly: 
proceeds by the selective annealing of the pads of distinct tiles, which allows tiles to compose together to form a controlled tiling lattice.
TX lattices


**AB* Lattice.** An atomic force microscope image of DNA lattice formed by two TAO tiles one of which contains an extra loop directed out of the plane. These loops form the visible stripe features with the expected spacing of ~28 nm.
Corrugated Lattice of Cross Tiles

Computational DNA Lattices
First Experimental Demonstrations of Computation via Tiling Assembly:
1D DNA Tiling Computation:


1st Experimental Demonstrations of Molecular Computation via DNA Tiling


- First experimental demonstration of computation via molecular self-assembly: Computation of XOR using DNA triple-crossover molecules
“String Tile” Addition. Example.

\[
\begin{array}{cccc}
\text{c}_0 & \text{c}_0 & \text{c}_1 & \text{c}_0 \\
\end{array}
\]

\[
\begin{array}{cccc}
\begin{array}{cccc}
\text{A} & \text{O} & \text{R} & \text{I} \\
\text{B} & 101 + 001 & 110 \\
\end{array}
\end{array}
\]

- Anneal strands to form assembly.
- Ligate reporter strand segments.
- Purify reporter strand and read values by PCR.

• **String Tile Addition Pads:**
  – The sticky end pads on right encode:
    • carry bits coming in and \( I_{A_i} \) and \( I_{B_i} \) encode the two input bits.
  – Left-hand pads pass new carry value on to next step
  – Reporter strands indicated by arrows; \( O_i \) encodes: output bit.

<table>
<thead>
<tr>
<th>tile</th>
<th>( c_i )</th>
<th>( I_{A_i} )</th>
<th>( I_{B_i} )</th>
<th>( O_i )</th>
<th>( c_{i+1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0 0 1</td>
<td>1 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0 1 0</td>
<td>1 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0 1 1</td>
<td>0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1 0 0</td>
<td>1 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1 0 1</td>
<td>0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1 1 0</td>
<td>0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1 1 1</td>
<td>1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Pad Programming via Truth Table:**
  – Column \( c_i \) gives values for the 3 right-hand pads (\( c_{i+1} \), \( \sim c_{2i} \), \( c_{3i} \))
  – Column \( c_{i+1} \) gives value for the 3 left-hand pads (\( \sim c_{1i+1} \), \( c_{2i+1} \), \( \sim c_{3i+1} \)).
TAE Assemblies for XOR Computation

XOR via TAE Computational Complex with Visual Readout

Subsequent experimental demonstration of computation via 2D DNA self-assembly at Caltech:

Algorithmic Self-assembly Of DNA Sierpinski Triangles: Rothemund, Papadakis, Winfree
2D DNA Tiling Computation:

Patterned DNA Lattices
Programmable Patterned DNA Nanostructures

NOT Patterned:                   Patterned:
Motivation for Patterned DNA lattices

- Allows for Attachment of Nanoparticles at Specific Sites on Lattice

- Application: Molecular Electronics: Layout of molecular electronic circuit components on DNA tiling arrays.
Programmable Barcoded Lattices


Molecular Pattern Formation using Scaffold Strands for Directed Nucleation:

• Multiple tiles of an input layer can be assembled around a single, long DNA strand we refer to as a scaffold strand (shown as black lines in the figures).

Barcode lattice displays banding patterns dictated by the sequence of bit values programmed on the input layer

```plaintext
Input Strand
1 0 1 1 0 0 0 1 0 1 1 1
```

Hao Yan
Barcode Lattice for Rendering 1 D Patterns:

H Yan, T LaBean, L Feng, J. Reif, PNAS (2003).

Barcode lattice displays banding patterns dictated by the same sequence of bit values programmed on each layer.
Hierarchical cross tile
Hierarchical Assembly of cross tiles
Addressable cross tile
Sung Ha Park, Constantin Pistol, Sang Jung Ahn, John H. Reif, Alvin R. Lebeck, Chris Dwyer, and Thomas H. LaBean, Finite-Size, Fully Addressable DNA Tile Lattices Formed by Hierarchical Assembly Procedures, Angewandte Chemie [International Edition], Volume 45,
Molecular Scale Patterning using Hierarchical Assembly of cross tiles
Hierarchical Assembly of DNA Lattices with 2 D Pattern “DNA”

Assembling a 2D Pattern by Directed Nucleation:
Self Assembly of Tiles around a DNA Strand Defining a 2D Pattern

Design Idea by LaBean & Reif, early 2000s
DNA Origami
Paul W K Rothemund’s DNA Origami

Conceptually, the second step (illustrated in Fig. 1b) proceeds by folding a single long scaffold strand (900 nucleotides (nt) in Fig. 1b) back and forth in a raster fill pattern so that it comprises one of the two strands in every helix; progression of the scaffold from one helix to another creates an additional set of crossovers, the ‘scaffold crossovers’ (indicated by small red crosses in Fig. 1b). The fundamental constraint on a folding path is that the scaffold can form a crossover only at those locations where the DNA twist places it at a tangent point between helices. Thus for the scaffold to raster progressively from one helix to another and onto a third, the distance between successive scaffold crossovers must be an odd number of half turns. Conversely, where the raster reverses direction vertically and returns to a previously visited helix, the distance between scaffold crossovers must be an even number of half-turns. Note that the folding path shown in Fig. 1b is compatible with a circular scaffold and leaves a ‘seam’ (a contour which the path does not cross).

Once the geometric model and a folding path are designed, they are represented as lists of DNA lengths and offsets in units of half-turns. These lists, along with the DNA sequence of the actual scaffold to be used, are input to a computer program. Rather than assuming 10.5 base pairs (bp) per turn (which corresponds to standard B-DNA twist), the program uses an integer number of bases between periodic crossovers (for example, 16 bp for 1.5 turns). It then performs the third step, the design of a set of ‘staple strands’ (the coloured DNA strands in Fig. 1c) that provide Watson–Crick complements for the
Fig. 6. A cartoon depicts folding of DNA origami as temperature changes from 90 °C to 20 °C.
DNA Origami
Nature, 2006

Paul W K Rothemund
California Institute of Technology

Staples reverse direction at these crossovers; thus crossovers are antiparallel, a stable configuration well characterized in DNA nanostructures. Note that the crossovers in Fig. 1c are drawn somewhat misleadingly, in that single-stranded regions appear to span the inter-helix gap even though the design leaves no bases unpaired. In the assembled structures, helices are likely to bend gently to meet at crossovers so that only a single phosphate from each backbone occurs in the gap (as ref. 16 suggests for similar structures). Such small-angle bending is not expected to greatly affect the width of DNA origami (see also Supplementary Note S2).

The minimization and balancing of twist strain between crossovers is complicated by the non-integer number of base pairs per half-turn (5.25 in standard B-DNA) and the asymmetric nature of the helix (it has major and minor grooves). Therefore, to balance the strain caused by representing 1.5 turns with 16 bp, periodic crossovers are arranged with a glide symmetry, namely that the minor groove faces alternating directions in alternating columns of periodic crossovers (see Fig. 1d, especially cross-sections 1 and 2). Scaffold crossovers are not balanced in this way. Thus in the fourth step, the twist of scaffold crossovers is calculated and their position is changed (typically by a single bp) to minimize strain; staple sequences are recomputed accordingly. Along seams and some edges the minor groove angle places scaffold crossovers in tension with adjacent periodic crossovers (Fig. 1d, cross-section 2); such situations are left unchanged.

Wherever two staples meet there is a nick in the backbone. Nicks occur on the top and bottom faces of the helices, as depicted in Fig. 1d. In the final step, to give the staples larger binding domains with the scaffold (in order to achieve higher binding specificity and higher binding energy which results in higher melting temperatures), pairs of adjacent staples are merged across nicks to yield fewer, longer, staples (Fig. 1e). To strengthen a seam, an additional pattern of breaks and merges may be imposed to yield staples that cross the seam; a seam spanned by staples is termed ‘bridged’. The pattern of merges is not unique; different choices yield different final patterns of nicks and staples. All merge patterns create the same shape but, as shown later, the merge pattern dictates the type of grid underlying any pixel pattern later applied to the shape.

Folding M13mp18 genomic DNA into shapes

To test the method, circular genomic DNA from the virus M13mp18 was chosen as the scaffold. Its naturally single-stranded 7,249-nt sequence was examined for secondary structure, and a hairpin with a 20-bp stem was found. Whether staples could bind at this hairpin was unknown, so a 73-nt region containing it was avoided. When a linear scaffold was required, M13mp18 was cut (in the 73-nt region) by digestion with BsrBI restriction enzyme. While 7,176 nt remained available for folding, most designs did not fold all 7,176 nt; short (#25 nt) ‘remainder strands’ were added to complement unused sequence. In general, a 100-fold excess of 200–250 staple and remainder strands were mixed with scaffold and annealed from

Figure 2 | DNA origami shapes. Top row, folding paths. a, square; b, rectangle; c, star; d, disk with three holes; e, triangle with rectangular domains; f, sharp triangle with trapezoidal domains and bridges between them (red lines in inset). Dangling curves and loops represent unfolded sequence. Second row from top, diagrams showing the bend of helices at crossovers (where helices touch) and away from crossovers (where helices bend apart). Colour indicates the base-pair index along the folding path; red is the 1st base, purple the 7,000th. Bottom two rows, AFM images. White lines and arrows indicate blunt-end stacking. White brackets in a mark the height of an unstretched square and that of a square stretched vertically (by a factor >1.5) into an hourglass. White features in f are hairpins; the triangle is labelled as in Fig. 3k but lies face down. All images and panels without scale bars are the same size, 165 nm × 165 nm. Scale bars for lower AFM images: b, 1 μm; c–f, 100 nm.

doi:10.1038/nature08016
Autonomous DNA Walkers: DNA Devices that Walk on DNA Nanostructures
First DNA Walker Devices: Formulation & First Designs
[Reif, 2002]

Designs for the first autonomous DNA nanomechanical devices that execute cycles of motion without external environmental changes.

Walking DNA device
Use ATP consumption

Rolling DNA device
Use hybridization energy

These DNA devices translate across a circular strand of ssDNA and rotate simultaneously.
Generate random bidirectional movements that acquire after n steps an expected translational deviation of $O(n^{1/2})$. 
Unidirectional Autonomous Walker

Molecular-Scale device in which an autonomous walker moves unidirectionally along a DNA track, driven by the hydrolysis of ATP

Our work: DNA walker
First autonomous DNA robotic device

- First Experimental Demonstration of an autonomous DNA walker
- Series of stators (blue)
- One walker (red)
- Use of ligase and restriction enzymes
Demonstrated First Autonomous DNA Walker:


Restriction enzymes

Ligase

Walker

Anchorage

Track

A* B C D A

PfM I

BstAP I
Autonomous DNA Racetrack Runners:
DNA Devices that Walk on Circular DNA Nanostructures
• phi-29 strand displacing polymerase

• Pushes cargo strand around a circular track
DNA wheels setup

T → BP → BP → T → T4 Ligase → T

W → BP → BP → T → T4 Ligase → T

BQ
DNA wheels motion
DNA wheels motion

Fig. 6. (a) The fluorescence shown by the assembly in absence of the cargo containing the quencher (b) The fluorescence quenched by the assembly of cargo containing the quencher (c) The fluorescence remains quenched even after the activity of the polymerase $\phi 29$, which indicates that the cargo is not dislodged from the wheel W

Fig. 7. (a) The fluorescence is shown by the assembly in absence of the cargo containing the quencher (b) The fluorescence is quenched after the assembly of the cargo containing the quencher (c) The fluorescence reappears after the polymerase $\phi 29$ pushes the wheel containing the quencher

Fig. 8. (a) The fluorescence is shown by the assembly in absence of the cargo containing the quencher (b) The fluorescence remains after the assembly of the cargo containing the quencher, away from the fluorophore (c) The fluorescence quenches after the polymerase $\phi 29$ pushes the wheel before it stops at stopping sequence, and the sticky end of the cargo hybridizes with the track to quench the fluorescence
Autonomous DNA Devices that Compute as They Walk
Programmable Autonomous DNA Nanorobotic Devices Using DNAzymes

John H. Reif and Sudheer Sahu

- **DNAzyme calculator**: a limited ability computational device
- **DNAzyme FSA**: a finite state automata device, that executes finite state transitions using DNAzymes
  - extensions to probabilistic automata and non-deterministic automata,
- **DNAzyme router**: for programmable routing of nanostructures on a 2D DNA addressable lattice
- **DNAzyme porter**: for loading and unloading of transported nano-particles
- **DNAzyme doctor**: a medical-related application to provide transduction of nucleic acid expression.
  - can be programmed to respond to the under-expression or over-expression of various strands of RNA, with a response by release of an RNA

All Devices:
- Autonomous, programmable, and no protein enzymes.
- The basic principle involved is inspired by Mao’s DNAzyme Walker
DNAzyme FSA (inputs, transitions)

\[
\begin{align*}
    &a_2 \ x_2 \ a_1 \ x_1 \ b_2 \ x_2 \ b_1 \ x_1 \ a_2 \ x_2 \ a_1 \ x_1 \\
    &\downarrow \ 0 \ 1 \ 0 \ \\
    &a_2 \ x_2 \ a_1 \ x_1 \ b_2 \ x_2 \ b_1 \ x_1 \ a_2 \ x_2 \ a_1 \ x_1 \\
    &t_2 \ t_1 \ t_2 \ t_1 \ t_2 \ t_1 \ t_2 \ t_1
\end{align*}
\]
DNAzyme Crawler

Sudheer Sahu
DNAzyme Calculator
DNA Doctor

DNAzyme Device for DNA Doctor
(John H. Reif and Sudheer Sahu, 2006)
• John Reif  
  www.cs.duke.edu/~reif/  

• PhD Candidates:  
  5th Year  
  PhD Graduate  
  Students:  

1st and 2nd Year  
PhD Graduate  
Students:
<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sudhanshu Garg</td>
</tr>
<tr>
<td>Hieu Bui</td>
</tr>
<tr>
<td>Reem Mokhtar</td>
</tr>
<tr>
<td>Tianqi Song</td>
</tr>
<tr>
<td>Xin Song</td>
</tr>
<tr>
<td>Shalin Shah</td>
</tr>
<tr>
<td>Dan Fu</td>
</tr>
<tr>
<td>Abeer Ashra (visiting)</td>
</tr>
</tbody>
</table>
What we do: Grad Student’s Research

- Sudhanshu Garg: Exponential auto-catalytic system
- Hieu Bui: DNA-origami-based localized circuit
- Tianqi Song: Analog computer using DNA
- Reem Makhtar: Self-reconfigurable DNA origami
- Xin Song & Abeer Eshra: Renewable DNA Logic Circuits
- Shalin Shah: Super-resolution imaging via advanced DNA paint
- Dan Fu: Casting complex 3D shapes from DNA origami
Localized DNA Hybridization Reactions on Nanotracks & DNA Origami

Hieu Bui
PhD
Department of Computer Science
Duke University
What is the problem?
- At high concentration, DNA circuits using DNA hybridization often suffer a moderately high leakage rate.
- At low concentration, required to avoid unwanted interactions, DNA circuits often proceed at much slower rates.

How to solve it?
- Use DNA self-assembly to place DNA circuits using DNA hybridization on the surface of DNA nanostructures in order to speed up the reaction rates and to minimize leakages.

Techniques:
1. Cascaded DNA Hybridization Chain Reaction of DNA Hairpins
2. Localized Cascade DNA Hybridization Chain Reactions (LCD-HCR) of DNA Hairpins on a DNA Track
3. LCD-HCR of DNA Hairpins on a DNA Origami Rectangle
4. Resettable and Redundancy LCD-HCR
Our Techniques and Papers:

0. DNA Hairpin Reactions:

1. Localized DNA Hybridization Reactions:
   • Hieu Bui, Tianqi Song, and John H Reif, Localized DNA Computation, Chapter in book: “From Parallel to Emergent Computing” (Edited by Andrew Adamatzky), CRC Press, to appear (June 2017).

2. Cascaded DNA Hybridization Chain Reaction of DNA Hairpins

3. Localized Cascade DNA Hybridization Chain Reactions of DNA Hairpins on a DNA Track
   • Hieu Bui, Sudhanshu Garg, Reem Mokhtar, Harish Chandran, Vincent Miao and John Reif, Design and Analysis of Localized DNA Hybridization Chain Reactions, Small (2017), 1602983. DOI: 10.1002/smll.201602983

4. Localized DNA Hybridization Chain Reactions on a DNA Origami Rectangle

4. Resettable and Redundancy LCD-HCR: Ongoing Work
Cascade Chain Reaction

DNA nanostructures

Experimental Results

Gel confirmation

Design

Real time kinetics

Localized Cascade DNA Hybridization Chain Reactions (LCD-HCR) of DNA Hairpins on a DNA Track

Reif et al., Design and Analysis of Localized DNA Hybridization Chain Reactions. Small, 1602983 (2017)
Experimental Results:
Localized reactions 6 times faster than in solution.

Tethering DNA reactions on the same platform enhanced the reaction rates.

Reif et al., Design and Analysis of Localized DNA Hybridization Chain Reactions. Small, 1602983 (2017)
Tethering DNA circuits on a DNA origami rectangle

CAD illustration

Design structure

Experimental verification via AFM

An ideal substrate for localized circuits. Why?
LCD-HCR of DNA Hairpins on a DNA Origami Rectangle

Design

Experimental Results

Before Initiation

After Completion

After Completion with Streptavidin

Reif et al., Localized DNA Hybridization Chain Reactions on DNA Origami. (submitted)
LCD-HCR of DNA Hairpins on a DNA Origami Rectangle

Experimental Results

Single molecule imaging via TIRF

Reif et al., Localized DNA Hybridization Chain Reactions on DNA Origami. (submitted)
On-Going Experiments: Resettable & Redundancy LCD-HCR

Resettable LCD-HCR: Once the forward cascade chain reaction pathway has reached the completion, the circuit can be reset back to the original configuration with the help of a set of reset DNA sequences.

Redundancy LCD-HCR: The forward cascade reaction pathway can proceed with a higher propensity due to the double encoded outputs in each gate.
Resettable & Redundancy LCD-HCR on DNA origami: Branching structures & Dendritic trees

Using resetting and redundancy gates to increase chain reaction success and to construct dendritic trees
DNA Origami Transformers

Reem Mokhtar
- PhD Candidate
Department of Computer Science
- Duke University

Example DNA Origami Transformers:
- Folding into an open box, and closed box.
Inspiration: Protein Transformations

Prior & Related Work on Limited DNA Origami Transformations

7 George M Church, S M Douglas, and Ido Bachelet. “A logic-gated nanorobot for targeted transport of molecular payloads”. In: Science Signaling 335.6070 (2012), p. 831. URL: http://stke.sciencemag.org/cgi/content/full/sci;335/6070/831.
9 Ebbe Andersen et al. “Self-assembly of a Nanoscale DNA Box with a Controllable Lid”. In: Nature 459.7243 (2009), pp. 73–76.
10 Haorong Chen et al. “Understanding the mechanical properties of DNA origami tiles and controlling the kinetics of their folding and unfolding reconfiguration.” In: Journal of the American Chemical Society 136.19 (May 2014), pp. 6995–7005.
Prior & Related Work on Limited DNA Origami Transformations (Cont)

DNA Origami Transformers: Goals

1. Set principles for the shape transformation of DNA origami nanostructures
2. Extend the capabilities of DNA origami shape transformation
3. Establish structurally-based computation within nanostructure
A DNA Origami Transformation via Staple Strand Displacement
UNZIP = A DNA Origami Transformation using Multiple Staple Strand Displacements
A reverse DNA Origami Transformation via Staple Strand Hybridization
ZIP = A DNA Origami Transformation using Multiple Staple Strand Hybridizations
Example:
Separating Mini-Origami Segments

1: 70 nm
2: 30 nm
Example: Separating Mini-Origami Segments

12 staple strands between helices 3 and 4, each with toeholds of 8-nt length on either side of the staple strands. Adding complimentary strands separates using toehold-mediated strand displacement.
Example:
Separating Mini-Origami Segments
Example #1 of a DNA Origami Transformer:

Folding into an open box, and transforming into a closed box.
Example #2 of a DNA Origami Transformer:

Folding into a box, long rectangle, articulated robot arms, and pick-and-place device.
Example #3 of a DNA Origami Transformer:

A Reconfigurable Network of Chambers containing reactive molecules
DNA-based Analog Computing

• Tianqi Song
  • PhD Candidate
  • Dept of Computer Science
  • Duke University
Introduction

• Analog computing: signals are directly represented by physical quantity.

• Analog DNA circuits: use real inputs and outputs as specified by concentrations of specific DNA strands.

• DNA-based analog computing has several potential molecular-scale applications:
  – Analog sensing & control of drug delivery,
  – Mechanical analog control systems,
  – Artificial neuron networks.
Our Architectures for Analog DNA Circuits

• Analog DNA Architecture I: based on addition, subtraction and multiplication gates. Circuits to compute polynomials are constructed from these gates.

  Tianqi Song; Sudhanshu Garg; Reem Mokhtar; Hieu Bui; John Reif; ACS Synth. Biol. 2016, 5, 898-912. DOI: 10.1021/acssynbio.6b00144).

• Analog DNA Architecture II: algebraic computations (e.g., sqrt(x), ln(x), exp(x)) are done using multiple autocatalytic amplifiers.

  Tianqi Song; Sudhanshu Garg; Reem Mokhtar; Hieu Bui; John Reif. Design and Analysis of Compact DNA Strand Displacement Circuits for Analog Computation Using Autocatalytic Amplifiers (under revision of ACS Synth. Biol.)
Analog DNA Gates for Architecture I

Tianqi Song; Sudhanshu Garg; Reem Mokhtar; Hieu Bui; John Reif;

Abstractions of our analog DNA gates: (left) addition gate, \( p_a = a_1 + a_2 \); (middle) subtraction gate, \( p_s = s_1 - s_2 \); (right) multiplication gate, \( p_m = m_1 * m_2 \).
Addition Gate:

Input $a_1 \rightarrow [I_{a1}]_0$
Input $a_2 \rightarrow [I_{a2}]_0$

Output $p_a \rightarrow [Oa]_{\text{final}}$

For input range $(0, r_a)$: $[Ga1]_0 = [Da1]_0 = [Ga2]_0 = [Da2]_0 = r_a$ and $[Fa]_0 = 2r_a$. 
Subtraction Gate:

Input $s_1 \rightarrow [Is1]_0$
Input $s_2 \rightarrow [Is2]_0$

Output $p_s \rightarrow [Is1]_{\text{final}}$

remainder after computing process.

For input range $(0, r_s)$:
$[Gs]_0 = [Ds]_0 = r_s$

$$Is2 + S \xrightarrow{k_1} S'$$

$$Is1 + S' \xrightarrow{k_2} \emptyset$$
Multiplication Gate:

Input $m_1 \rightarrow [I_{m1}]_0$
Input $m_2 \rightarrow [I_{m2}]_0$

Output $p_m \rightarrow [O_{m}]_{\text{final}}$

For input range $(0, r_m)$:

$[G_{m1}]_0 = [D_{m1}]_0 = [F_{m1}]_0 = [G_{m2}]_0 = [F_{m2}]_0 = [G_{m3}]_0 = [D_{m3}]_0 = [G_{m4}]_0 = r_m$.

For the reaction rate constants: $k_s \ll k_f, k_1, k_2$.

$Im_1 + M_1 \xrightarrow{k_i} Im_{1a}$ \hspace{1cm} (3a)

$Im_2 + M_2 \xrightarrow{k_i} Im_{2a} + Im_{2b}$ \hspace{1cm} (3b)

$Im_{2a} + M_3 \xrightarrow{k_i} G'm_3$ \hspace{1cm} (3c)

$Im_{2b} + Gm_4 \xrightarrow{k_i} \emptyset$ \hspace{1cm} (3d)

$Im_{1a} + Gm_4 \xrightarrow{k_i} \emptyset$ \hspace{1cm} (3e)

$Im_{1a} + G'm_3 \xrightarrow{k_i} O_{m1}$ \hspace{1cm} (3f)

$O_{m1} + \text{amplifier} \xrightarrow{k_i} \frac{O_{m} + O_{m} + \ldots + O_{m}}{r_{m}O_{m}}$ \hspace{1cm} (3g)
Multiplication Gate (Contd.)

• By reactions (3c) and (3d), \( \frac{m_2}{m_2 + (r_m - m_2)} \) portion of Im1a will be consumed by reaction (3f), given that reactions (3e) and (3f) have the same rate constants.

• \([Om1]_{\text{final}} \) will be \( m_1 \times \left( \frac{m_2}{m_2 + (r_m - m_2)} \right) = \frac{m_1 \times m_2}{r_m} \) without the amplifier.

• \([Om]_{\text{final}} \) will be \( \left( \frac{m_1 \times m_2}{r_m} \right) \times r_m = m_1 \times m_2 \).
Simulation Model

- **Software:** Visual GEC and Matlab
- **Rate Constants:**
  - toehold binding $2 \times 10^{-3}$ nM$^{-1}$ s$^{-1}$
  - toehold unbinding 10 s$^{-1}$
  - branch migration $8000/x^2$ s$^{-1}$
  - branch migration with mismatch $0.01 \times 8000/x^2$ s$^{-1}$, where $x$ is the length (number of nucleotides) of branch migration domain.
Simulation Results

![Simulation Results](image)

**Figure 20.** Examples to show the execution of our gates when the input range is \((0,4)\). The vertical axes represent the concentrations of output DNA strands. The ranges between the red and green dotted lines are the valid output ranges. We do not show the curves for the whole simulated period \((7.2 \times 10^5\) seconds) for the convenience to see the shapes of the curves at the early stage.

![Simulation Results](image)

**Figure 23.** Performance of the gates when the input range is \((0,4)\). The color represents \(\log_e(t)\) where \(t\) is the time (seconds) that the output stays within the valid output range. \(\log_e(t)\) is used instead of \(t\) simply for convenience in plotting.
Design of experiments:

- **Addition gate**: test how the transducer works since the gate is just made of two transducers with same output.
- **For truncation (subtraction) gate**: test the whole gate. Reporter complex is added after the cancellation between It1 and It2 is finished.
- **For multiplication gate**: test its key mechanism -- competitive hybridization.
Experimental Results (Contd.)

(a) $[D_{a1}]_0 = [G_{a1}]_0 = 100 \text{ nM}, [F_a]_0 = [RC]_0 = 120 \text{ nM}$.  
(b) $[G_{t1}]_0 = [D_{t1}]_0 = 100 \text{ nM}, [RC]_0 = 120 \text{ nM}$.  
(c) $[I]_0 = 100 \text{ nM}$.

Relative errors ((experimental result – expected result) / expected result):
(a) < 5% for all cases,  
(b) < 5% for all cases,  
(c) <10% for all cases except the case of RC 200 nM, RCC 400 nM (14.6%).
Example Analog Circuit using Architecture I

A circuit to compute $f(x) = 1 + x + x^2/2! + x^3/3!$ for $0 < x < 1$.

(a) Execution of the circuit to compute $f(x)$ when $x = 0.7$. (b) Performance of the circuit to compute $f(x)$, where $t$ is the time (seconds) that the output stays in the valid output range.
Analog DNA Architecture II
Tianqi Song; Sudhanshu Garg; Reem Mokhtar; Hieu Bui; John Reif.
Design and Analysis of Compact DNA Strand Displacement Circuits for Analog Computation Using Autocatalytic Amplifiers (under revision of ACS Synth. Biol.)

The main idea of Architecture II:
Use several autocatalytic amplifier systems with exponential output signal (in certain period) to regulate each other to perform computation. Each autocatalytic amplifier’s response grows exponentially $\exp(ct)$ with time $t$ after being triggered (for a tunable constant $c$).

The basic motif: A tunable autocatalytic amplifier made from Seesaw Gates [Qian et al. Science 2011]

DNA system for Autocatalytic amplifier:
Leak simulation of autocatalytic seesaw amplifier:
Gate(G1 : W1,2) and Gate(G2 : W2,1) are at 100 nM.
Fuel(W1;3) and fuel(W2;4) are at 200 nM.
(a) How the concentration of output grows over time.
(b) We plot \( \ln(\text{output concentration}) \) over time. The straight line domain is where the output signal can be modeled by functions in the form of \( a e^{ct} \) where \( a, c \) are constants and \( t \) is time duration.
Example Circuit to Compute Square Root using Architecture II

Three modules in the architecture: input module $A_I$, output module $A_O$ and stopper module $A_S$. Each module is an autocatalytic amplifier.

Let $[O_I]=f(t)$, $[output]=g(t)$. To compute $\sqrt{x}$, we tune $A_I$ and $A_O$ such that $g(f^{-1}(x)) = \sqrt{x}$ for $x$ in a restricted range, where $[input]_0 = x$, and $f^{-1}(x)$ is for how long $A_O$ runs.

Uses formula:
$\sqrt{x} = \exp(\ln(x)/2)$
A Circuit to Compute Square Root Based on Architecture II (Contd.)

<table>
<thead>
<tr>
<th>Module</th>
<th>$\sqrt{x}$</th>
</tr>
</thead>
</table>
| Input      | $[G_1]_{ini} = [G_2]_{ini} = 100 \text{ nM}$  
             | $[F_1]_{ini} = [F_2]_{ini} = 200 \text{ nM}$  
             | $[H]_{ini} = [F]_{ini} = 100 \text{ nM}$       |
| Output     | $[G_1]_{ini} = [G_2]_{ini} = 100 \text{ nM}$  
             | $[F_1]_{ini} = [F_2]_{ini} = 1771 \text{ nM}$  
             | $[H]_{ini} = [F]_{ini} = 100 \text{ nM}$       |
| Stopper    | $[G_1]_{ini} = [G_2]_{ini} = 2000 \text{ nM}$ 
             | $[F_1]_{ini} = [F_2]_{ini} = 4000 \text{ nM}$  
             | $[H]_{ini} = [F]_{ini} = 2000 \text{ nM}$       |

Table 1: Left: The setup for a circuit to compute $\sqrt{x}$ where $x \in (6.83, 10.49)$ (nM). Right: The simulated performance of the circuit to compute $\sqrt{x}$ (nM), $x \in (6.83, 10.49)$ (nM): the simulation is conducted for samples $\{6.83, 6.93, \ldots, 10.43\}$. The percentage error is computed by formula $\left| \frac{(1/r)\sqrt{c(x)} - \sqrt{x}}{\sqrt{x}} \right|$, where $\sqrt{c(x)}$ is the result from simulating the circuit, $(1/r)$ is a calibration factor and $\sqrt{x}$ is the theoretical result.
DNA Hairpin Based Autocatalytic Systems

Sudhanshu Garg
PhD Graduated Spring, 2016
Duke University
DNA Hairpin based Autocatalytic Systems

Autocatalytic Chemical Reaction Network.

\[
\begin{align*}
I_a + A & \rightarrow I_a + B & \text{Catalytic} \\
B & \rightarrow I_x & \text{Transduction} \\
I_x + X & \rightarrow I_x + Y & \text{Catalytic} \\
Y & \rightarrow I_a & \text{Transduction} \\
A & \leftrightarrow X & \text{Cross-Catalytic}
\end{align*}
\]

# Self replicating DNA Nanostructures from Hairpins (Garg et. al. In preparation)

*** Leakless DNA Strand Displacement Cascades (Thachuk et. al. 2015)
Insight: Irreversible Hairpin Opening

Traditional Hairpin Opening is reversible

Forward Reaction: Toehold Mediated Strand Displacement
Backward Reaction: Remote Toehold Mediated Strand Displacement

Irreversible Hairpin Opening

# Self replicating DNA Nanostructures from Hairpins (Garg et. al. In preparation)
Experimental Results:
DNA Hairpin Based Autocatalytic Systems

- Autocatalytic Systems leak quickly.
- Multiple curves show the rate of leakage in the system.
- Reducing leaks involves getting rid of synthesis errors,
- extremely pure strands can result in very low leak.
DNA Logic Gate: Challenges

- Toehold exchange strand displacement [1]

- Scalable circuits using toehold exchange strand displacement:
- Square root circuit, seesaw DNA logic gates [2]

Problem:
Irreversible and Non-reusable, once reaches equilibrium and generates output.

Proposed solutions:
1) Use of photo-controllable molecules intercalated within DNA strands to reverse the reaction. (in collaboration with Xin Song)
2) Extraction mechanism and use of multiple universal toeholds with sequestering within hairpin loops.

Application:
- Building reversible nano-robotic devices like walkers, gears and tweezers.
- Photo-control chemical reactions for biological and medical use

References:
Renewable DNA Circuits

Sudhanshu Garg
PhD Candidate
Duke University
Current circuits can only perform 1 cycle of computation, or a single instruction.

Logical Progression: Create circuits that can perform multiple cycles of computation, or multiple instructions.

In each cycle, introduce a new set of inputs (instruction), that will be processed in that cycle.

First, pull-back the existing input. Then introduce the new input.

# Universal Computation using Time Responsive DNA Circuits (Garg et. al. In preparation)
Cycle 1: Introduce Inputs
(Eg 2-input gate: OR, AND, NOR etc.)
Cycle 2:
Pull back inputs from the circuit
Construction of a complex gate (XOR)

- **X XOR Y**
- **(X AND NOT Y) OR (Y AND NOT X)**

- **5 DNA Gates:**
  - 2 ANDs
  - 1 ORs
  - 2 NOTs
Experimental Results
using a 2-input 1-output Gate Complex

Sequence of Addition at 100nM*:

4. Cycle 1: Introduce both Inputs, X and Y
5. Pull Back Input X and Y
7. Cycle 2: Reintroduce Both Inputs X and Y.
10. Signal to consume all Reporter Complex.

(*Some numbered steps have not been detailed)
Current & Past Projects

• Renewable DNA Seesaw Logic Gates and Circuits
• Inductive Nano-Patterning and Applications
• Silica Encapsulation of FRET Devices on DNA Scaffold
Solution 1: Photo-controllable toehold-mediated strand displacement

- Azobenzene isomerization application on toeholds [1]
- Optically controlled release of ssDNA using BHQs [2]

Solution 2: Multiple universal toeholds sequestering in hairpin loop and extraction mechanism

Step 1: Forward reaction with hairpin gate:

Step 2: Extractors to get backward reaction:

References:
Self-Assembled and Renewable DNA Seesaw Logic Gates with Photonic Controls and Signals
### Self-Assembled and Renewable DNA Seesaw Logic Gates with Photonic Controls and Signals

**Species name** | **Regeneration yield (%)** | **Regeneration yield (%)** | **Regeneration yield (%)** | **Regeneration yield (%)**  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100% effective toehold sequestering</td>
<td>80% effective toehold sequestering</td>
<td>60% effective toehold sequestering</td>
<td>40% effective toehold sequestering</td>
</tr>
<tr>
<td>Input 1</td>
<td>99.7</td>
<td>99.4</td>
<td>88.4</td>
<td>33.6</td>
</tr>
<tr>
<td>Input 2</td>
<td>99.7</td>
<td>99.4</td>
<td>88.4</td>
<td>33.6</td>
</tr>
<tr>
<td>IntGate</td>
<td>99.8</td>
<td>99.6</td>
<td>89.6</td>
<td>40.2</td>
</tr>
<tr>
<td>ThrGate</td>
<td>99.8</td>
<td>99.4</td>
<td>82.8</td>
<td>0.4</td>
</tr>
<tr>
<td>AmpGate</td>
<td>99.5</td>
<td>98.7</td>
<td>75.2</td>
<td>8.2</td>
</tr>
<tr>
<td>Fuel</td>
<td>99.8</td>
<td>99.4</td>
<td>87.6</td>
<td>54.1</td>
</tr>
<tr>
<td>RepGate</td>
<td>99.9</td>
<td>99.6</td>
<td>89.8</td>
<td>45.5</td>
</tr>
</tbody>
</table>

### Table for different input scenarios

**Species name** | **Regeneration yield (%)** | **Regeneration yield (%)** | **Regeneration yield (%)** | **Regeneration yield (%)**  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input ON-ON</td>
<td>Input OFF-OFF</td>
<td>Input OFF-ON</td>
<td>Input OFF-OFF</td>
</tr>
<tr>
<td>Input 1</td>
<td>99.4</td>
<td>99.2</td>
<td>99.0</td>
<td>97.0</td>
</tr>
<tr>
<td>Input 2</td>
<td>99.4</td>
<td>99.0</td>
<td>99.2</td>
<td>97.0</td>
</tr>
<tr>
<td>IntGate</td>
<td>99.6</td>
<td>99.6</td>
<td>99.6</td>
<td>99.7</td>
</tr>
<tr>
<td>ThrGate</td>
<td>99.4</td>
<td>99.5</td>
<td>99.5</td>
<td>99.5</td>
</tr>
<tr>
<td>AmpGate</td>
<td>98.7</td>
<td>98.9</td>
<td>98.9</td>
<td>99.6</td>
</tr>
<tr>
<td>Fuel</td>
<td>99.4</td>
<td>99.5</td>
<td>99.5</td>
<td>99.8</td>
</tr>
<tr>
<td>RepGate</td>
<td>99.6</td>
<td>99.7</td>
<td>99.7</td>
<td>99.8</td>
</tr>
</tbody>
</table>

Xin Song, Abeer Eshra, Chris Dwyer and John Reif, *Renewable DNA Seesaw Logic Circuits Enabled by Photoregulation of Toehold-Mediated Strand Displacement* (In peer review for journal publication)
Renewable DNA Computing

Abeer Eshra
Visiting scholar in John Reif Lab, Duke University
PhD candidate, Department of Computer Science and Engineering
Faculty of Electronic Engineering, Menoufia University
A Renewable DNA Seesaw Motif Using Hairpins

Application: 2 Input Logic OR Gate

Results of basic motif:

Polyacrylamide gel electrophoresis analysis

In both figures, red circles show restoration after adding extractors. Green circles show computation after adding input and fuel to start a forward reaction and reuse the circuit. In Figure 7, a blue circle is showing accumulating waste from previous restoration. Intensity of adjacent band in lane 10 shows that concentration of waste is increasing.

OR gate with reversal and changing cases. Left: OFFON then ONON. Right: OFFON, ONOFF then ONON.
Reif Group Papers on the Web

Reif Papers on DNA nanoscience on the Web:

- Survey on DNA Computation:

Other Reif Papers on the Web:
Talks Downloadable from Reif’s Website

- [www.cs.duke.edu/~reif/paper/DNA-NanoscienceTalks](http://www.cs.duke.edu/~reif/paper/DNA-NanoscienceTalks)

**This Talk: DNA-Based Programmable Molecular Devices**


**Other Talks:**

DNA Computing: Theory, Experiments & Software:


Self-Assembled DNA Nanostructures:


DNA-Based Programmable Autonomous Molecular Robotic Devices: