Design and synthesis of dynamically assembling DNA nanostructures

Design and synthesis of dynamically assembling DNA nanostructures

A dissertation presented

by

John Paul Sadowski

to

The Department of Chemistry and Chemical Biology

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Chemistry

Harvard University

Cambridge, Massachusetts

December 2013

Dissertation Advisor: Professor Peng Yin John Paul Sadowski

Design and synthesis of dynamically assembling DNA nanostructures

Abstract

Kinetically controlled isothermal growth is fundamental to biological development, but it remains challenging to rationally design molecular systems that self-assemble isothermally into complex geometries via prescribed assembly and disassembly pathways. By exploiting the programmable chemistry of base pairing, sophisticated spatial and temporal control have both been demonstrated in DNA self-assembly, but largely as separate pursuits. This dissertation extends a new approach, called developmental self-assembly, that integrates temporal with spatial control by using a prescriptive molecular program to specify the kinetic pathways by which DNA molecules isothermally self-assemble into well-defined three-dimensional geometries.

First, a new general-use computational sequence design platform, called Multisubjective, is discussed. Multisubjective identifies the specific bases that are responsible for undesired secondary structure, and redesigns only those bases using one of the supported client designers. It combines features of combinatorial and thermodynamic designers, judiciously targeting the thermodynamic analysis to specific candidate designs on an occasional basis and using the fast combinatorial algorithms to do most of the design work. This allows Multisubjective to successfully design large, complex nucleic acid systems in a reasonable amount of time.

Second, the synthesis and characterization of a wireframe DNA tetrahedron is presented as a proof of principle for developmental self-assembly. Nine DNA reactants initially co-exist in a metastable state, but upon catalysis by a DNA initiator molecule, navigate 24 individually characterizable intermediate states via prescribed assembly pathways to arrive at the tetrahedral final product. In contrast to previous work on dynamic DNA nanotechnology, this developmental program coordinates growth of ringed substructures into a three-dimensional wireframe superstructure, taking a step towards the goal of kinetically controlled isothermal growth of complex three-dimensional geometries.

Lastly, design approaches for developmental self-assembly must take into account potential structural problems specific to hairpin structures, which can interfere with the kinetics of the desired strand displacement reactions. Preliminary data is presented regarding the effect of tail structure, including toehold–tail interactions, on the hairpin opening kinetics.

Table of Contents

Appendix B

Preface

This dissertation represents the culmination not only of my graduate research, but also of a 12-year career in DNA nanotechnology. When I started work in this field as a high school student in 2001, DX tile arrays were the state of the art, a formation gel was Figure 2 of every paper, and nearly all of the major results of the field came from just three laboratories, those of Nadrian Seeman, Erik Winfree, and Bernie Yurke. Now, especially since the introduction of DNA origami in 2006 and the increasingly sophisticated development of dynamic strand displacement methods for DNA computation over the last couple of years, interest in the field has exploded, and we have begun to seriously consider using these capabilities to develop actual applications. The thought that DNA nanotechnology might soon solve a real-life problem better than any other method is an exciting one, and having experienced the field coming to this point has been a wonderful thing to live through.

The research presented here represents a convergence of ideas from the dynamic and structural subfields of DNA nanotechnology. While this is a relatively new approach, I have striven to maintain the rigor of analysis that I remember from the "old days" before DNA origami, complete with a formation gel as Figure 2 of the published version of the tetrahedron chapter. This research also integrates the full spectrum of approaches needed to design and understand these systems, containing chapters on computational design, molecular engineering, and basic science. I hope that this convergence of ideas one day bears fruit and allows us to build the kinds of functional nanomaterials that we can only dream of today.

Of course, I have had the support of many people over the years. I thank my parents, and the administration and faculty of North Shore Schools, for allowing me a progressive education where I could develop my talents at an appropriate pace, and my colleagues at the California Institute of Technology for providing the most engaging and stimulating environment I have ever been a part of.

I have to especially acknowledge my many research advisors and mentors. Nadrian Seeman and Philip Lukeman generously gave me my first research experience while I was still in high school, and at Caltech James Heath, Peter Dervan, and Justin Cohen provided years of support that led to my first publications. I like to say that, while an undergraduate program is like walking up a long flight of stairs onto a platform, graduate school is more akin to digging a hole hoping to find a trampoline with which to jump onto the platform. I credit my graduate school advisors, David Liu and Peng Yin, for their faith in my ability to both dig holes and to extricate myself from them.

If there is one lesson I have learned in graduate school, it is that it is as important to master the chemistry between people as the chemistry between molecules. I am privileged to have been in an environment where I could make use of the intellect and expertise of my fellow scientists in the various laboratories of which I have been a member. These are too numerous to name, but some are acknowledged in the individual chapters of this dissertation. I also thank the organizers and members of both the Science Policy Group and the Philosophy of Science Society at the Harvard Graduate School of Arts and Sciences for enriching my experience of science beyond the bench. Farther afield, my participation in Wikipedia not only provided me with a great avenue for informing the wider public about nanotechnology and related topics, but has also, unexpectedly, exposed me to an entire new world of interesting people and opportunities.

And lastly, science without art is meaningless, so I thank my friends in the theater groups I have been involved with, especially Theater Arts at Caltech (TACIT) and the MIT Musical Theatre Guild. The sense of camaraderie and shared accomplishment that comes with each production is something that truly can be found nowhere else.

John P. Sadowski

Cambridge, Massachusetts

December 13, 2013

Chapter 1

Introduction to DNA nanotechnology

"Serious-minded people have few ideas. People with ideas are never serious."

—Paul Valéry

Overview

DNA nanotechnology is the design and manufacture of artificial nucleic acid structures for technological uses. In this field, nucleic acids are used as non-biological engineering materials for nanotechnology rather than as the carriers of genetic information in living cells. Researchers have created static structures such as two- and three-dimensional crystal lattices, nanotubes, polyhedra, and arbitrary shapes, as well as functional devices such as molecular machines and DNA computers. The field is beginning to be used as a tool to solve basic science problems in structural biology and biophysics, including applications in crystallography and spectroscopy for protein structure determination. Potential applications in molecular scale electronics and nanomedicine are also being investigated.

The conceptual foundation for DNA nanotechnology was first laid out by Nadrian Seeman in the early 1980s, and the field began to attract widespread interest in the mid-2000s. This use of nucleic acids is enabled by their strict base pairing rules, which cause only portions of strands with complementary base sequences to bind together to form strong, rigid double helix structures. This allows for the rational design of base sequences that will selectively assemble to form complex target structures with precisely controlled nanoscale features. A number of assembly methods are used to make these structures, including tile-based structures that assemble from smaller structures, folding structures using the DNA origami method, and dynamically reconfigurable structures using strand displacement techniques. DNA is the dominant material used, but structures incorporating other nucleic acids such as RNA[1,2] and peptide nucleic acid (PNA)[3] have also been constructed, leading to the occasional use of the name nucleic acid nanotechnology to describe the field.

[1] A. Chworos, I. Severcan, A. Y. Koyfman, P. Weinkam, E. Oroudjev, H. G. Hansma, and L. Jaeger. Building programmable jigsaw puzzles with RNA. *Science*, 306: 2068–72 (2004). doi:10.1126/science.1104686

[2] P. Guo. The emerging field of RNA nanotechnology. *Nature Nanotech.*, 5:833–42 (2010). doi:10.1038/nnano.2010.231

[3] P. S. Lukeman, A. C. Mittal, and N. C. Seeman. Two dimensional PNA/DNA arrays: estimating the helicity of unusual nucleic acid polymers. *Chem. Commun.*, 2004:1694–1695, 2004. doi:10.1039/B401103A

Fundamental concepts

Properties of nucleic acids. Nanotechnology is often defined as the study of materials and devices with features on a scale below 100 nanometers. DNA nanotechnology is an example of bottom-up molecular self-assembly, in which molecular components spontaneously organize into stable structures; the particular form of these structures is induced by the physical and chemical properties of the components selected by the designers.[4] In DNA nanotechnology, the component materials are strands of nucleic acids such as DNA; these strands are often synthetic and are almost always used outside the context of a living cell. DNA is well suited to nanoscale construction because the binding between two nucleic acid strands depends on simple base pairing rules that are well understood, and form the specific nanoscale structure of the nucleic acid double helix. These qualities make the assembly of nucleic acid structures easy to control through nucleic acid design. This property is absent in other materials used in nanotechnology, including proteins, for which protein design is very difficult, and nanoparticles, which lack the capability for specific assembly on their own.[5]

The structure of a nucleic acid molecule consists of a sequence of nucleotides distinguished by which nucleobase they contain. In DNA, the four bases present are adenine (A), cytosine (C), guanine (G), and thymine (T). Nucleic acids have the property that two molecules will only bind to each other to form a double helix if the two sequences are complementary, meaning that they form matching sequences of base pairs, with A only binding to T, and C only to G.[5,6] Because the formation of correctly matched base pairs is energetically favorable, nucleic acid strands are expected in most cases to bind to each other in the conformation that maximizes the number of correctly paired bases. The sequences of bases in a

[4] J. A. Pelesko. "Self-assembly: the science of things that put themselves together." New York: Chapman & Hall/CRC, 2007. pp. 5, 7. ISBN 978-1-58488-687-7

[5] N. C. Seeman. Nanomaterials based on DNA. *Annu. Rev. Biochem.*, 79:65–87, 2010. doi:10.1146/annurev-biochem-060308-102244

[6] E. C. Long. Fundamentals of nucleic acids. In Hecht, S. M. "Bioorganic chemistry: nucleic acids." New York: Oxford University Press, 1996. pp. 4–10. ISBN 0-19-508467-5

system of strands thus determine the pattern of binding and the overall structure in an easily controllable way. In DNA nanotechnology, the base sequences of strands are rationally designed by researchers so that the base pairing interactions cause the strands to assemble in the desired conformation.[5,7]

Subfields. DNA nanotechnology is sometimes divided into two overlapping subfields: structural DNA nanotechnology and dynamic DNA nanotechnology. Structural DNA nanotechnology, sometimes abbreviated as SDN, focuses on synthesizing and characterizing nucleic acid complexes and materials that assemble into a static, equilibrium end state. On the other hand, dynamic DNA nanotechnology focuses on complexes with useful non-equilibrium behavior such as the ability to reconfigure based on a chemical or physical stimulus. Some complexes, such as nucleic acid nanomechanical devices, combine features of both the structural and dynamic subfields.[8,9]

The complexes constructed in structural DNA nanotechnology use topologically branched nucleic acid structures containing junctions. (In contrast, most biological DNA exists as an unbranched double helix.) One of the simplest branched structures is a four-arm junction that consists of four individual DNA strands, portions of which are complementary in a specific pattern (Fig. 1.1a). Unlike in natural Holliday junctions, each arm in the artificial immobile four-arm junction has a different base sequence, causing the junction point to be fixed at a certain position. Multiple junctions can be combined in the same complex, such as in the widely used double-crossover (DX) motif, which contains two parallel double helical domains with individual strands crossing between the domains at two crossover points (Fig. 1.1b). Each crossover point is itself topologically a four-arm junction, but is constrained to a single orientation, as opposed to the flexible single four-arm junction, providing a rigidity that makes the DX motif suitable as a structural building block for larger DNA complexes.[5,7]

[7] N. C. Seeman. Nanotechnology and the double helix. *Sci. Am.*, 290(6):64–75, 2004. doi:10.1038/scientificamerican0604-64

[8] D. Y. Zhang and G.Seelig. Dynamic DNA nanotechnology using strand-displacement reactions. *Nature Chem.*, 3:103–13, 2011. doi:10.1038/nchem.957

[9] N. C. Seeman. An overview of structural DNA nanotechnology. *Mol. Biotechnol.*, 37:246–57, 2007. doi:10.1007/s12033-007-0059-4

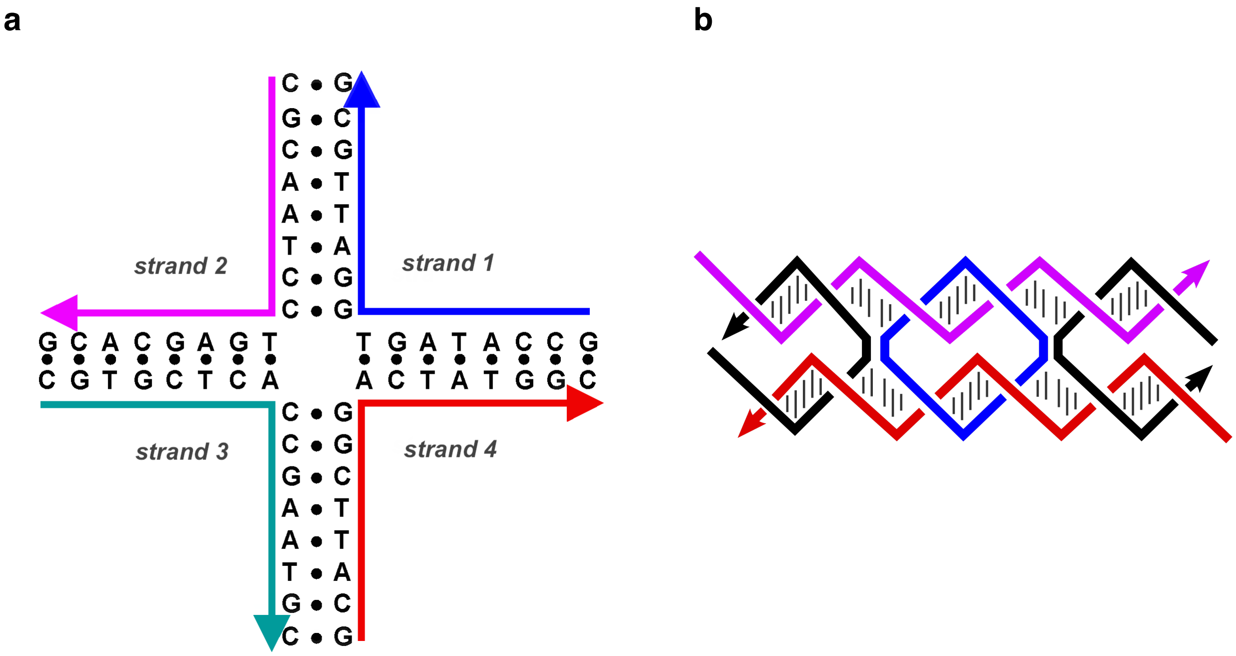


Figure 1.1: Examples of designed DNA junctions. **a**, These four strands associate into a DNA four-arm junction because this structure maximizes the number of correct base pairs, with A matched to T and C matched to G. **b**, This double-crossover (DX) supramolecular complex consists of five DNA single strands that form two double-helical domains, on the top and the bottom in this image. There are two crossover points where the strands cross from one domain into the other.[10]

Dynamic DNA nanotechnology uses a mechanism called toehold-mediated strand displacement to allow the nucleic acid complexes to reconfigure in response to the addition of a new nucleic acid strand. In this reaction, the incoming strand binds to a single-stranded toehold region of a double-stranded complex, and then displaces one of the strands bound in the original complex through a branch migration process. The overall effect is that one of the strands in the complex is replaced with another one.[8] In addition, reconfigurable structures and devices can be made using functional nucleic acids such as deoxyribozymes and ribozymes, which are capable of performing chemical reactions, and aptamers, which can bind to specific proteins or small molecules.[11]

[10] C. Mao. The emergence of complexity: lessons from DNA. *PLoS Biol.*, 2:2036–8, 2004. doi:10.1371/journal.pbio.0020431

[11] Y. Lu and J. Liu. Functional DNA nanotechnology: Emerging applications of DNAzymes and aptamers. *Curr. Opin. Biotech.*, 17: 580–8, 2006. doi:10.1016/j.copbio.2006.10.004

History

The conceptual foundation for DNA nanotechnology was first laid out by Nadrian Seeman in the early 1980s.[12] Seeman's original motivation was to create a three-dimensional DNA lattice for orienting other large molecules, which would simplify their crystallographic study by eliminating the difficult process of obtaining pure crystals. This idea had reportedly come to him in late 1980, after realizing the similarity between the woodcut *Depth* by M. C. Escher and an array of DNA six-arm junctions.[7,13] A number of natural branched DNA structures were known at the time, including the DNA replication fork and the mobile Holliday junction, but Seeman's insight was that immobile nucleic acid junctions could be created by properly designing the strand sequences to remove symmetry in the assembled molecule, and that these immobile junctions could in principle be combined into rigid crystalline lattices. The first theoretical paper proposing this scheme was published in 1982, and the first experimental demonstration of an immobile DNA junction was published the following year.[5,14]

In 1991, Seeman's laboratory published a report on the synthesis of a cube made of DNA, the first synthetic three-dimensional nucleic acid nanostructure, for which he received the 1995 Feynman Prize in Nanotechnology. This was followed by a DNA truncated octahedron. However, it soon became clear that these structures, polygonal shapes with flexible junctions as their vertices, were not rigid enough to form extended three-dimensional lattices. Seeman developed the more rigid double-crossover (DX) motif, and in 1998, in collaboration with Erik Winfree, published the creation of two-dimensional lattices of DX

[12] J. A. Pelesko. "Self-assembly: the science of things that put themselves together." New York: Chapman & Hall/CRC, 2007. pp. 201, 242, 259. ISBN 978-1-58488-687-7.

[13] See "Current crystallization protocol" (http://seemanlab4.chem.nyu.edu/nano-pro.html) for a statement of the problem, and "DNA cages containing oriented guests" (http://seemanlab4.chem.nyu.edu/nano-cage.html) for the proposed solution.

[14] P. W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440:297–302, 2006. doi:10.1038/nature04586

tiles.[7,12,15] These tile-based structures had the advantage that they provided the capability to implement DNA computing, which was demonstrated by Winfree and Paul Rothemund in their 2004 paper on the algorithmic self-assembly of a Sierpinski gasket structure, and for which they shared the 2006 Feynman Prize in Nanotechnology. Winfree's key insight was that the DX tiles could be used as Wang tiles, meaning that their assembly was capable of performing computation.[12] The synthesis of a three-dimensional lattice was finally published by Seeman in 2009, nearly thirty years after he had set out to achieve it.[16]

New capabilities continued to be discovered for designed DNA structures throughout the 2000s. The first DNA nanomachine—a motif that changes its structure in response to an input—was demonstrated in 1999 by Seeman. An improved system, which was the first nucleic acid device to make use of toehold-mediated strand displacement, was demonstrated by Bernard Yurke the following year. The next advance was to translate this into mechanical motion, and in 2004 and 2005, a number of DNA walker systems were demonstrated by the groups of Seeman, Niles Pierce, Andrew Turberfield, and Chengde Mao.[17] The idea of using DNA arrays to template the assembly of other molecules such as nanoparticles and proteins, first suggested by Bruche Robinson and Seeman in 1987,[18] was demonstrated in 2006 and 2007 by the groups of Hao Yan, Peter Dervan, and Thomas LaBean.[5,19]

In 2006, Rothemund first demonstrated the DNA origami technique for easily and robustly creating folded DNA structures of arbitrary shape. Rothemund had conceived of this method as being conceptually intermediate between Seeman's DX lattices, which used many short strands, and William

[15] P. W. K. Rothemund. Scaffolded DNA origami: from generalized multicrossovers to polygonal networks. In J. Chen, N. Jonoska, and G.Rozenberg, "Nanotechnology: science and computation." *Natural Computing Series*. New York: Springer, 2006. pp. 3–21. doi:10.1007/3-540-30296-4_1

[16] S. M. Douglas, H. Dietz, T. Liedl, B. Högberg, F. Graf, and W. M. Shih. Self-assembly of DNA into nanoscale three-dimensional shapes. *Nature*, 459:414–8, 2009. doi:10.1038/nature08016

[17] J. Bath and A. J. Turberfield. DNA nanomachines. *Nature Nanotech.*, 2:275–84, 2007. doi:10.1038/nnano.2007.104

[18] B. H. Robinson and N. C. Seeman. The design of a biochip: a self-assembling molecular-scale memory device. *Protein Eng.*, 1:295–300, 1987. doi:10.1093/protein/1.4.295

[19] M. Endo and H. Sugiyama. Chemical approaches to DNA nanotechnology. *ChemBioChem*, 10:2420–43, 2009. doi:10.1002/cbic.200900286

Shih's DNA octahedron, which consisted mostly of one very long strand. Rothemund's DNA origami contains a long strand whose folding is assisted by a number of short strands. This method allowed the creation of much larger structures than were previously possible, and which are less technically demanding to design and synthesize.[15] DNA origami was the cover story of Nature on March 15, 2006.[20] Rothemund's research demonstrating two-dimensional DNA origami structures was followed by the demonstration of solid three-dimensional DNA origami by Douglas *et al.* in 2009,[16] while the labs of Jørgen Kjems and Yan demonstrated hollow three-dimensional structures made out of two-dimensional faces.[21]

DNA nanotechnology was initially met with some skepticism due to the unusual non-biological use of nucleic acids as materials for building structures and doing computation, and the preponderance of proof-of-principle experiments that extended the capabilities of the field but were far from actual applications. Seeman's 1991 paper on the synthesis of the DNA cube was rejected by the journal *Science* after one reviewer praised its originality while another criticized it for its lack of biological relevance. By the early 2010s, however, the field was considered to have increased its capabilities to the point that applications for basic science research were beginning to be realized, and practical applications in medicine and other fields were beginning to be considered feasible.[21,22] The field had grown from very few active laboratories in 2001 to at least 60 in 2010, which increased the talent pool and thus the number of scientific advances in the field during that decade.[33]

[20] A. V. Pinheiro, D. Han, W. M. Shih, H. Yan. Challenges and opportunities for structural DNA nanotechnology. *Nature Nanotech.*, 6:763–72, 2011. doi:10.1038/nnano.2011.187

[21] R. F. Service. News focus: DNA nanotechnology grows up. *Science*, 332:1140–3, 2011. doi:10.1126/science.332.6034.1140

[22] K. Hopkin. Profile: 3-D seer. *The Scientist*, August 2011. Available at http://www.the-scientist.com/?articles.view/articleNo/30936/title/3-D-Seer/

Design

DNA nanostructures must be rationally designed so that the individual nucleic acid strands will assemble into the desired structures. This process usually begins with the specification of a desired target structure or functionality. Then, the overall secondary structure of the target complex is determined, specifying the arrangement of nucleic acid strands within the structure, and which portions of those strands should be bound to each other. The last step is the primary structure design, which is the specification of the actual base sequences of each nucleic acid strand.[23,24]

Motif design. The first step in designing a nucleic acid nanostructure is to decide how a given structure should be represented by a specific arrangement of nucleic acid strands. This design step determines the secondary structure, or the positions of the base pairs that hold the individual strands together in the desired shape.[23] Several approaches have been demonstrated:

- Tile-based structures. This approach breaks the target structure into smaller units with strong binding between the strands contained in each unit, and weaker interactions between the units. It is often used to make periodic lattices, but can also be used to implement algorithmic self-assembly, making them a platform for DNA computing. This was the dominant design strategy used from the mid-1990s until the mid-2000s, when the DNA origami methodology was developed.[23,25]
- Folding structures. An alternative to the tile-based approach, folding approaches make the nanostructure from a single long strand. This long strand can either have a designed sequence that folds due to its interactions with itself, or it can be folded into the desired shape by using shorter,

[23] U. Feldkamp and C. M. Niemeyer. Rational design of DNA nanoarchitectures. *Angew. Chem. Int. Ed.*, 45: 1856–76, 2006. doi:10.1002/anie.200502358

[24] A. Brenneman and A. Condon. Strand design for biomolecular computation. *Theor. Comput. Sci.*, 287:39–58, 2002. doi:10.1016/S0304-3975(02)00135-4

[25] C. Lin, Y. Liu, S. Rinker, and H. Yan. DNA tile based self-assembly: building complex nanoarchitectures. *ChemPhysChem*, 7:1641–47, 2006. doi:10.1002/cphc.200600260

"staple" strands. This latter method is called DNA origami, which allows the creation of nanoscale two- and three-dimensional shapes.[14,20]

- Dynamic assembly. This approach directly controls the kinetics of DNA self-assembly, specifying all of the intermediate steps in the reaction mechanism in addition to the final product. This is done using starting materials that adopt a hairpin structure; these then assemble into the final conformation in a cascade reaction, in a specific order. This approach has the advantage of proceeding isothermally, at a constant temperature. This is in contrast to the thermodynamic approaches, which require a thermal annealing step where a temperature change is required to trigger the assembly and favor proper formation of the desired structure.[20,26]

Sequence design. After any of the above approaches are used to design the secondary structure of a target complex, an actual sequence of nucleotides that will form into the desired structure must be devised. Nucleic acid design is the process of assigning a specific nucleic acid base sequence to each of a structure's constituent strands so that they will associate into a desired conformation. Most methods have the goal of designing sequences so that the target structure has the lowest energy, and is thus the most thermodynamically favorable, while incorrectly assembled structures have higher energies and are thus disfavored. This is done either through simple, faster combinatorial methods such as sequence symmetry minimization, or by using a full nearest-neighbor thermodynamic model, which is more accurate but slower and more computationally intensive. Geometric models are used to examine tertiary structure of the nanostructures and to ensure that the complexes are not overly strained.[24,27]

Nucleic acid design has similar goals to protein design. In both, the sequence of monomers is designed to favor the desired target structure and to disfavor other structures. Nucleic acid design has the advantage of being much computationally easier than protein design, because the simple base pairing

[26] P. Yin, H. M. T. Choi, C. R. Calvert and N. A. Pierce. Programming biomolecular self-assembly pathways. *Nature*, 451: 318–22, 2008. doi:10.1038/nature06451

[27] R. M. Dirks, M. Lin, E. Winfree, and N. A. Pierce. Paradigms for computational nucleic acid design. *Nucleic Acids Res.*, 32:1392–403, 2004. doi:10.1093/nar/gkh291

rules are sufficient to predict a structure's energetic favorability, and detailed information about the overall three-dimensional folding of the structure is not required. This allows the use of simple heuristic methods that yield experimentally robust designs. However, nucleic acid structures are less versatile than proteins in their functionality because of proteins' increased ability to fold into complex structures, as well as the limited chemical diversity of the four nucleotides as compared to the twenty proteinogenic amino acids.[27]

Structural DNA nanotechnology

Structural DNA nanotechnology, sometimes abbreviated as SDN, focuses on synthesizing and characterizing nucleic acid complexes and materials where the assembly has a static, equilibrium endpoint. The nucleic acid double helix has a robust, defined three-dimensional geometry that makes it possible to predict and design the structures of more complicated nucleic acid complexes. Many such structures have been created, including two- and three-dimensional structures, and periodic, aperiodic, and discrete structures.[9]

Extended lattices. Small nucleic acid complexes can be equipped with sticky ends and combined into larger two-dimensional periodic lattices containing a specific tessellated pattern of the individual molecular tiles.[9] The earliest example of this used double-crossover (DX) complexes as the basic tiles, each containing four sticky ends designed with sequences that caused the DX units to combine into periodic two-dimensional flat sheets that are essentially rigid two-dimensional crystals of DNA (Fig. 1.2).[28,29] Two-dimensional arrays have been made from other motifs as well, including the Holliday

[28] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:529–44, 1998. doi:10.1038/28998

[29] F. Liu, R. Sha, and N. C. Seeman. Modifying the surface features of two-dimensional DNA crystals. *J. Am. Chem. Soc.*, 121:917–22, 1999. doi:10.1021/ja982824a

junction rhombus lattice,[30] and various DX-based arrays making use of a double-cohesion scheme (Fig. 1.3).[31,32]

Two-dimensional arrays can be made to exhibit aperiodic structures whose assembly implements a specific algorithm, exhibiting one form of DNA computing (Fig. 1.4).[33] The DX tiles can have their sticky end sequences chosen so that they act as Wang tiles, allowing them to perform computation. A DX array whose assembly encodes an XOR operation has been demonstrated; this allows the DNA array to implement a cellular automaton that generates a fractal known as the Sierpinski gasket. The third image at right shows this type of array.[37] Another system has the function of a binary counter, displaying a representation of increasing binary numbers as it grows. These results show that computation can be incorporated into the assembly of DNA arrays.[34]

DX arrays have been made to form hollow nanotubes 4–20 nm in diameter, essentially two-dimensional lattices which curve back upon themselves.[23] These DNA nanotubes are somewhat similar in size and shape to carbon nanotubes, and while they lack the electrical conductance of carbon nanotubes, DNA nanotubes are more easily modified and connected to other structures. One of many schemes for constructing DNA nanotubes uses a lattice of curved DX tiles that curls around itself and closes into a

[30] C. Mao, W. Sun, and N. C. Seeman. Designed two-dimensional DNA Holliday junction arrays visualized by atomic force microscopy. *J. Am. Chem. Soc.*, 121:5437–43, 1999. doi:10.1021/ja9900398

[31] P. E. Constantinou, T. Wang, J. Kopatsch, L. B. Israel, X. Zhang, B. Ding, W. B. Sherman, X. Wang, J. Zheng, R. Sha, and N. C. Seeman. Double cohesion in structural DNA nanotechnology. *Org. Biomol. Chem.*, 4:3414–19, 2006. doi:10.1039/b605212f

[32] F. Mathieu, S. Liao, J. Kopatsch, T.Wang, C. Mao, and N. C. Seeman. Six-helix bundles designed from DNA. *Nano Lett.*, 5:661–5, 2005. doi:10.1021/nl050084f

[33] N. Seeman. Structural DNA nanotechnology: growing along with Nano Letters. *Nano Lett.*, 10:1971–8, 2010. doi:10.1021/nl101262u

[34] R. D. Barish, P. W. K. Rothemund, and E. Winfree. Two computational primitives for algorithmic self-assembly: copying and counting, 2005. *Nano Lett.*, 5:2586–92. doi:10.1021/nl052038l

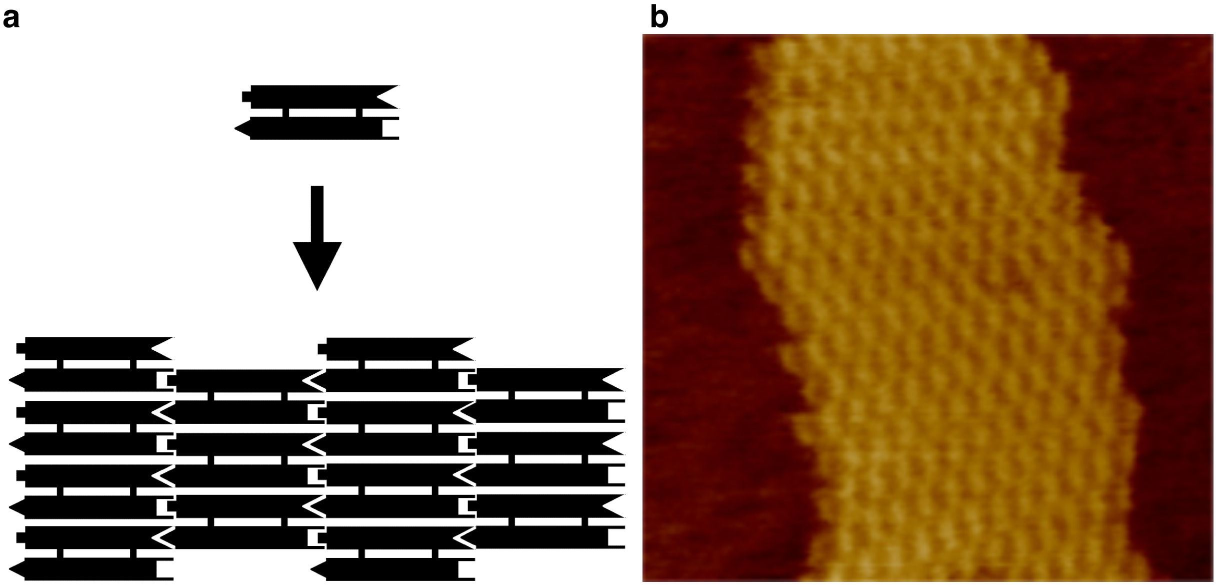


Figure 1.2: The assembly of a DX array. a, Schematic diagram. Each bar represents a double-helical domain of DNA, with the shapes representing complementary sticky ends. The DX complex at top will combine with other DX complexes into the two-dimensional array shown at bottom.[10] **b**, An atomic force microscope image of the assembled array. The individual DX tiles are clearly visible within the assembled structure. The field is 150 nm across.

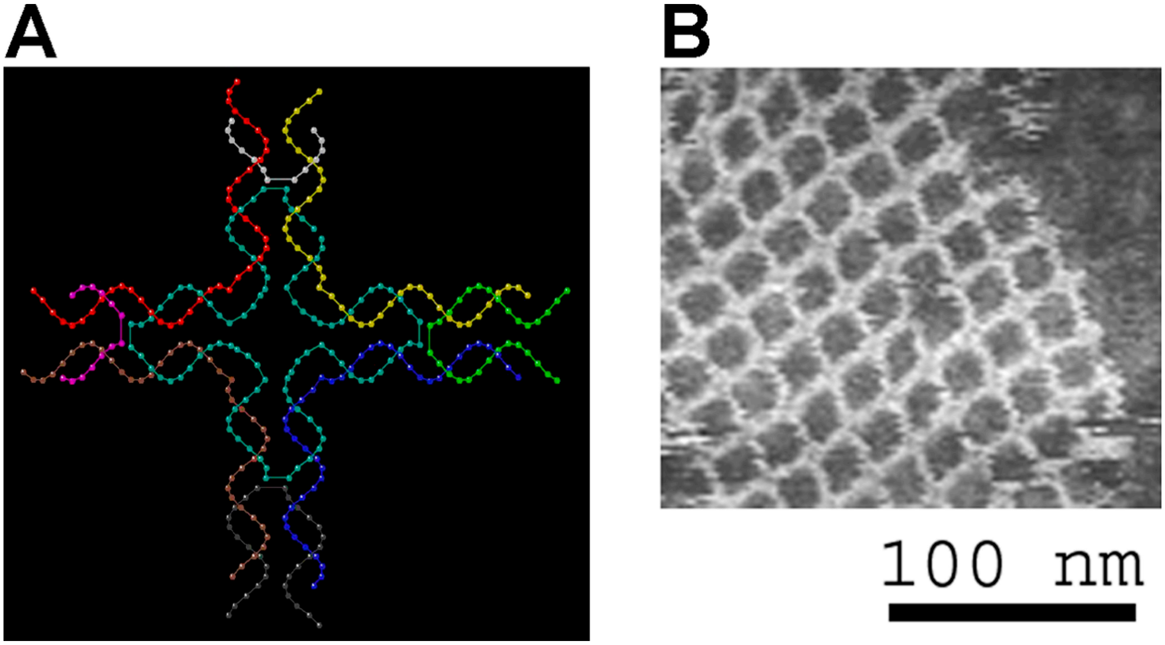


Figure 1.3: The assembly of a double-cohesion array. a, A model of a DNA tile used to make another two-dimensional periodic lattice. **b**, An atomic force micrograph of the assembled lattice.[35,36]

[35] M. Strong. Protein nanomachines. *PLoS Biol.*, 2:e73, 2004. doi:10.1371/journal.pbio.0020073

[36] H. Yan, S. H. Park, G. Finkelstein, J. H. Reif, and T. H. Labean. DNA-templated self-assembly of protein arrays and highly conductive nanowires. *Science*, 301:1882–4, 2003. doi:10.1126/science.1089389

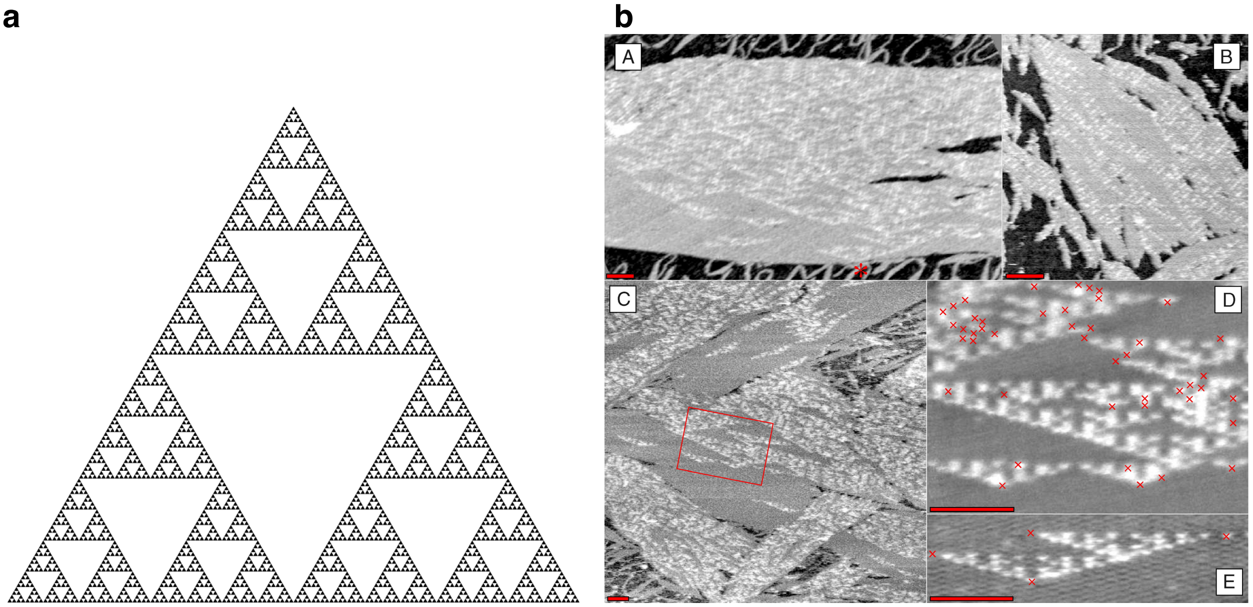


Figure 1.4: An aperiodic two-dimensional lattice that assembles into a fractal pattern. a, The Sierpinski gasket fractal. **b**, DNA arrays that display a representation of the Sierpinski gasket on their surfaces[37]

tube.[38] In an alternative method that allows the circumference to be specified in a simple, modular fashion using single-stranded tiles, the rigidity of the tube is an emergent property.[39]

The creation of three-dimensional lattices out of DNA was the earliest goal of DNA nanotechnology, but this proved to be one of the most difficult to realize. Success using a motif based on the concept of tensegrity, a balance between tension and compression forces, was finally reported in 2009.[33,40]

Discrete structures. Researchers have synthesized a number of three-dimensional DNA complexes that each have the connectivity of a polyhedron, such as a cube or octahedron, meaning that the DNA

[37] P. W. K. Rothemund, N. Papadakis, and E. Winfree. Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biol.*, 2:2041–53, 2004. doi:10.1371/journal.pbio.0020424

[38] P. W. K. Rothemund, A. Ekani-Nkodo, N. Papadakis, A. Kumar, D. K. Fygenson, and E. Winfree. Design and characterization of programmable DNA nanotubes. *J. Am. Chem. Soc.*, 126:16344–52, 2004. doi:10.1021/ja044319l

[39] P. Yin, R. F. Hariadi, S. Sahu, H. M. T. Choi, S. H. Park, T. H. Labean, and J. H. Reif. Programming DNA tube circumferences. *Science*, 321:824–6, 2008. doi:10.1126/science.1157312

[40] J. Zheng, J. J. Birktoft, Y. Chen, T. Wang, R.Sha, P. E. Constantinou, S. L. Ginell, C. Mao, and N. C. Seeman. From molecular to macroscopic via the rational design of a self-assembled 3D DNA crystal. *Nature*, 461:74–7, 2009. doi:10.1038/nature08274

duplexes trace the edges of a polyhedron with a DNA junction at each vertex.[20] The earliest demonstrations of DNA polyhedra were very work-intensive, requiring multiple ligations and solid-phase synthesis steps to create catenated polyhedra.[41] Subsequent work yielded polyhedra whose synthesis was much easier. These include a DNA octahedron made from a long single strand designed to fold into the correct conformation,[42] and a tetrahedron that can be produced from four DNA strands in a single step.[43]

Nanostructures of arbitrary, non-regular shapes are usually made using the DNA origami method. These structures consist of a long, natural virus strand as a "scaffold", which is made to fold into the desired shape by computationally designed short "staple" strands. This method has the advantages of being easy to design, as the base sequence is predetermined by the scaffold strand sequence, and not requiring high strand purity and accurate stoichiometry, as most other DNA nanotechnology methods do. DNA origami was first demonstrated for two-dimensional shapes, such as a smiley face and a coarse map of North America.[14,20] Solid three-dimensional structures can be made by using parallel DNA helices arranged in a honeycomb pattern,[16] and structures with two-dimensional faces can be made to fold into a hollow overall three-dimensional shape, akin to a cardboard box. These can be programmed to open and reveal or release a molecular cargo in response to a stimulus, making them potentially useful as programmable molecular cages.[44,45]

[41] Y. Zhang and N. C. Seeman. Construction of a DNA-truncated octahedron. *J. Am. Chem. Soc.*, 116:1661–9, 1994. doi:10.1021/ja00084a006

[42] W. M. Shih, J. D. Quispe, and G. F. Joyce. A 1.7-kilobase single-stranded DNA that folds into a nanoscale octahedron. *Nature*, 427:618–621, 2004. doi:10.1038/nature02307

[43] R. P. Goodman, I. A. T. Schaap, C. F. Tardin, C. M. Erben, R. M. Berry, C. F. Schmidt, and A. J. Turberfield. Rapid chiral assembly of rigid DNA building blocks for molecular nanofabrication. *Science*, 310:1661–5 (2005). doi:10.1126/science.1120367

[44] E. S. Andersen, M. Dong, M. M. Nielsen, K. Jahn, R. Subramani, W. Mamdouh, M. M. Golas, B. Sander, and H. Stark. Self-assembly of a nanoscale DNA box with a controllable lid. *Nature*, 459:73–6, 2009. doi:10.1038/nature07971

[45] Y. Ke, J. Sharma, M. Liu, K. Jahn, Y. Liu, and H. Yan. Scaffolded DNA origami of a DNA tetrahedron molecular container. *Nano Lett.*, 9:2445–7, 2009. doi:10.1021/nl901165f

More recently, the single-stranded tile (SST) method has been developed to make discrete shapes in a modular fashion. The SST method uses single strands with four hybridization domains as “bricks” that gain an emergent rigidity upon assembling. This method has the advantage of being modular and reusable; unlike DNA origami, making a new shape does not require a redesign of the structure, as unneeded bricks can simply be excluded from a pre-existing “canvas”. This method has been used to create both two-dimensional[46] and three-dimensional[47] shapes.

Templated assembly. Nucleic acid structures can be made to incorporate molecules other than nucleic acids, sometimes called heteroelements, including proteins, metallic nanoparticles, quantum dots, and fullerenes. This allows the construction of materials and devices with a range of functionalities much greater than is possible with nucleic acids alone. The goal is to use the self-assembly of the nucleic acid structures to template the assembly of the nanoparticles hosted on them, controlling their position and in some cases orientation.[19,20] Many of these schemes use a covalent attachment scheme, using oligonucleotides with amide or thiol functional groups as a chemical handle to bind the heteroelements. This covalent binding scheme has been used to arrange gold nanoparticles on a DX-based array,[48] and to arrange streptavidin protein molecules into specific patterns on a DX array.[49] A non-covalent hosting scheme using Dervan polyamides on a DX array was used to arrange streptavidin proteins in a specific pattern on a DX array.[50,51] Carbon nanotubes have been hosted on DNA arrays in a pattern allowing the

[46] B. Wei, M. Dai, and P. Yin. Complex shapes self-assembled from single-stranded DNA tiles. *Nature*, 485:623–6, 2012. doi:10.1038/nature11075

[47] Y. Ke, L. Ong, W. Shih, and P. Yin. Three-dimensional structures self-assembled from DNA bricks. *Science*, 338:1177–83, 2012. doi:10.1126/science.1227268

[48] J. Zheng, P. E. Constantinou, C. Micheel, A. P. Alivisatos, R. A. Kiehl, and N. C. Seeman. 2D nanoparticle arrays show the organizational power of robust DNA motifs. *Nano Lett.*, 6:1502–4, 2006. doi:10.1021/nl060994c

[49] S. H. Park, C. Pistol, S. J. Ahn, J. H. Reif, A. R. Lebeck, C. Dwyer and T. H. LaBean. Finite-size, fully addressable DNA tile lattices formed by hierarchical assembly procedures. *Angew. Chem. Int. Ed.*, 45:735–739, 2006. doi: 10.1002/anie.200503797.

[50] J. D. Cohen, J. P. Sadowski, and P. B. Dervan. Addressing single molecules on DNA nanostructures. *Angew. Chem. Int. Ed.*, 46:7956–9, 2007. doi:10.1002/anie.200702767

assembly to act as a molecular electronic device, a carbon nanotube field-effect transistor.[52] In addition, there are nucleic acid metallization methods, in which the nucleic acid is replaced by a metal which assumes the general shape of the original nucleic acid structure,[53,54] and schemes for using nucleic acid nanostructures as lithography masks, transferring their pattern into a solid surface.[55,56]

Dynamic DNA nanotechnology

Dynamic DNA nanotechnology focuses on creating nucleic acid systems with designed dynamic functionalities related to their overall structures, such as computation and mechanical motion. There is some overlap between structural and dynamic DNA nanotechnology, as structures can be formed through annealing and then reconfigured dynamically, or can be made to form dynamically in the first place.[17,20]

Nanomechanical devices. DNA complexes have been made that change their conformation upon some stimulus, making them one form of nanorobotics. These structures are initially formed in the same way as the static structures made in structural DNA nanotechnology, but are designed so that dynamic reconfiguration is possible after the initial assembly.[8,17] The earliest such device made use of the transition

[51] J. D. Cohen, J. P. Sadowski, and P. B. Dervan. Programming multiple protein patterns on a single DNA nanostructure. *J. Am Chem. Soc.*, 130:402–3, 2008. doi:10.1021/ja0772400

[52] H. T. Maune, S.-P. Han, R. D. Barish, M. Bockrath, W. A. Goddard III, P. W. K. Rothemund, and E. Winfree. Self-assembly of carbon nanotubes into two-dimensional geometries using DNA origami templates. *Nature Nanotech.*, 5:61–6, 2009. doi:10.1038/nnano.2009.311

[53] J. Liu, Y. Geng, E. Pound, S. Gyawali, J. R. Ashton, J. Hickey, A. T. Woolley, and J. N. Harb. Metallization of branched DNA origami for nanoelectronic circuit fabrication. *ACS Nano*, 5:2240–7, 2011. doi:10.1021/nn1035075

[54] S. Surwade, F. Zhou, B. Wei, W. Sun, A. Powell, C. O'Donnell, P. Yin, and H. Liu. Nanoscale growth and patterning of inorganic oxides using DNA nanostructure templates. *J. Am. Chem. Soc.*, 35:6778–81, 2013. doi:10.1021/ja401785h

[55] Z. Deng and C. Mao. Molecular lithography with DNA nanostructures. *Angew. Chem. Int. Ed.*, 43:4068–70, 2004. doi:10.1002/anie.200460257

[56] Z. Jin, W. Sun, Y. Ke, C.-J. Shih, G. L.C. Paulus, Q. H. Wang, B. Mu, P. Yin, and M. S. Strano. Metalized DNA nanolithography for encoding and transferring spatial information for graphene patterning. *Nature Comm.*, 4:1663, 2013. doi:10.1038/ncomms2690

between the B-DNA and Z-DNA forms to respond to a change in buffer conditions by undergoing a twisting motion.[57] This reliance on buffer conditions, however, caused all devices to change state at the same time. Subsequent systems could change states based upon the presence of control strands, allowing multiple devices to be independently operated in solution. Some examples of such systems are a "molecular tweezers" design that has an open and a closed state,[58] a device that could switch from a paranemic-crossover (PX) conformation to a double-junction (JX2) conformation, undergoing rotational motion in the process,[59] and a two-dimensional array that could dynamically expand and contract in response to control strands.[60] Structures have also been made that dynamically open or close, potentially acting as a molecular cage to release or reveal a functional cargo upon opening.[44,61,62]

DNA walkers are a class of nucleic acid nanomachines that exhibit directional motion along a linear track. A large number of schemes have been demonstrated.[17] One strategy is to control the motion of the walker along the track using control strands that need to be manually added in sequence.[63,64] Another approach is to make use of restriction enzymes or deoxyribozymes to cleave the strands and

[57] C. Mao, W. Sun, Z. Shen, and N. C. Seeman. A DNA nanomechanical device based on the B–Z transition. *Nature*, 397:144–6, 1999. doi:10.1038/16437

[58] B. Yurke, A. J. Turberfield, A. P. Mills Jr., F. C. Simmel, and J. L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406:605–9, 2000. doi:10.1038/35020524

[59] H. Yan, X. Zhang, Z. Shen, and N. C. Seeman. A robust DNA mechanical device controlled by hybridization topology. *Nature*, 415:62–5, 2002. doi:10.1038/415062a

[60] L. Feng, S. H. Park, J. H. Reif, and H. Yan. A two-state DNA lattice switched by DNA nanoactuator. *Angew. Chem. Int. Ed.*, 42:4342–6, 2003. doi:10.1002/anie.200351818

[61] R. P. Goodman, M. Heilemann, S. Doose, C. M. Erben, A. N. Kapanidis, and A. J. Turberfield. Reconfigurable, braced, three-dimensional DNA nanostructures. *Nature Nanotech.*, 3:93–6, 2008. doi:10.1038/nnano.2008.3

[62] S. M. Douglas, I. Bachelet, G. M. Church. A logic-gated nanorobot for targeted transport of molecular payloads. *Science* 335:831–4, 2012. doi:10.1126/science.1214081

[63] J.-S. Shin and N. A. Pierce. A synthetic DNA walker for molecular transport. *J. Am. Chem. Soc.*, 126:10834–5, 2004. doi:10.1021/ja047543j

[64] W. B. Sherman and N. C. Seeman. A precisely controlled DNA biped walking device. *Nano Lett.*, 4:1203–7, 2004. doi:10.1021/nl049527q

cause the walker to move forward, which has the advantage of running autonomously.[65,66] A later system could walk upon a two-dimensional surface rather than a linear track, and demonstrated the ability to selectively pick up and move molecular cargo.[67] Additionally, a linear walker has been demonstrated that performs DNA-templated synthesis as the walker advances along the track, allowing autonomous multistep chemical synthesis directed by the walker.[68]

Strand displacement cascades. Cascades of strand displacement reactions can be used for either computational or structural purposes. An individual strand displacement reaction involves revealing a new sequence in response to the presence of some initiator strand (Fig. 1.5). Many such reactions can be linked into a cascade where the newly revealed output sequence of one reaction can initiate another strand displacement reaction elsewhere. This in turn allows for the construction of chemical reaction networks with many components, exhibiting complex computational and information processing abilities. These cascades are made energetically favorable through the formation of new base pairs, and the entropy gain from disassembly reactions. Strand displacement cascades allow for isothermal operation of the assembly or computational process, as opposed to traditional nucleic acid assembly's requirement for a thermal annealing step, where the temperature is raised and then slowly lowered to ensure proper formation of the desired structure. They can also support catalytic functionality of the initiator species, where less than one equivalent of the initiator can cause the reaction to go to completion.[8,26]

[65] Y. Tian, Y. He, Y. Chen, P. Yin, and C. Mao. A DNAzyme that walks processively and autonomously along a one-dimensional track. *Angew. Chem. Int. Ed.*, 44:4355–8, 2005. doi:10.1002/anie.200500703

[66] J. Bath, S. J. Green, and A. J. Turberfield. A free-running DNA motor powered by a nicking enzyme. *Angew. Chem. Int. Ed.*, 44:4358–61, 2005. doi:10.1002/anie.200501262

[67] K. Lund, A. J. Manzo, N. Dabby, N. Michelotti, A. Johnson-Buck, J. Nangreave, S. Taylor, R. Pei, M. N. Stojanovic, N. G. Walter, E. Winfree, and H. Yan. Molecular robots guided by prescriptive landscapes. *Nature*, 465:206–10, 2010. doi:10.1038/nature09012

[68] Y. He and D. R. Liu. Autonomous multistep organic synthesis in a single isothermal solution mediated by a DNA walker. *Nature Nanotech.*, 5:778–82, 2010. doi:10.1038/nnano.2010.190

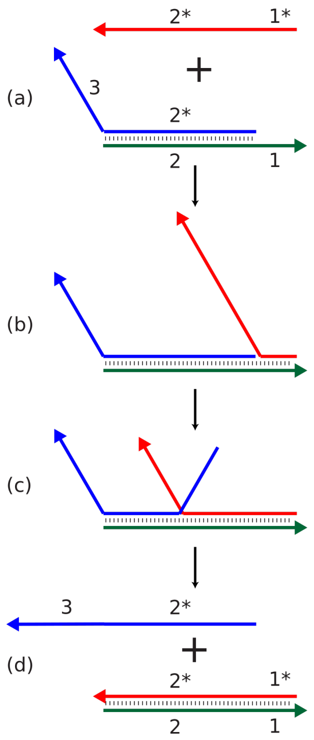


Figure 1.5: Toehold-mediated strand displacement. Dynamic DNA nanotechnology often makes use of toehold-mediated strand displacement reactions. In this example, the red strand binds to the single stranded toehold region on the green strand (region 1), and then in a branch migration process across region 2, the blue strand is displaced and freed from the complex. Reactions like these are used to dynamically reconfigure or assemble nucleic acid nanostructures. In addition, the red and blue strands can be used as signals in a molecular logic gate.

Strand displacement complexes can be used to make molecular logic gates capable of complex computation. Unlike traditional electronic computers, which use electric current as inputs and outputs, molecular computers use the concentrations of specific chemical species as signals. In the case of nucleic acid strand displacement circuits, the signal is the presence of nucleic acid strands that are released or consumed by binding and unbinding events to other strands in displacement complexes. This approach has been used to make logic gates such as AND, OR, and NOT gates.[69] More recently, a four-bit circuit was demonstrated that can compute the square root of the integers 0–15, using a system of gates containing 130 DNA strands.[70]

[69] G. Seelig, D. Soloveichik, D. Y. Zhang, and E. Winfree. Enzyme-free nucleic acid logic circuits. *Science*, 314: 1585–8, 2006. doi:10.1126/science.1132493

[70] L. Qian and E. Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332:1196–1201, 2011. doi:10.1126/science.1200520

Another use of strand displacement cascades is to make dynamically assembled structures. These use a hairpin structure for the reactants, so that when the input strand binds, the newly revealed sequence is on the same molecule rather than disassembling. This allows new opened hairpins to be added to a growing complex. This approach has been used to make simple structures such as three- and four-arm junctions and dendrimers.[26]

Typical methods

The sequences of the DNA strands making up a target structure are designed computationally, using molecular modeling and thermodynamic modeling software.[24,27] The nucleic acids themselves are then synthesized using standard oligonucleotide synthesis methods, usually automated in an oligonucleotide synthesizer, and strands of custom sequences are commercially available.[71] Strands can be purified by denaturing gel electrophoresis if needed,[72] and precise concentrations determined via any of several nucleic acid quantitation methods using ultraviolet absorbance spectroscopy.[73]

The fully formed target structures can be verified using native gel electrophoresis, which gives size and shape information for the nucleic acid complexes. An electrophoretic mobility shift assay can assess whether a structure incorporates all desired strands.[74] Fluorescent labeling and Förster resonance energy transfer (FRET) are sometimes used to characterize the structure of the complexes.[75]

[71] A. Ellington and J. D. Pollard Jr. Synthesis and purification of oligonucleotides. *Current Protocols in Molecular Biology*, 42:2.11.1–25, 2001. doi:10.1002/0471142727.mb0211s42

[72] A. Ellington and J. D. Pollard, Jr. Purification of oligonucleotides using denaturing polyacrylamide gel electrophoresis. *Current Protocols in Molecular Biology*, 42:2.12.1–7, 2001. doi:10.1002/0471142727.mb0212s42

[73] S. R. Gallagher and P. Desjardins. Quantitation of nucleic acids and proteins. *Current Protocols Essential Laboratory Techniques*, 5:2.2.1–36, 2011. doi:10.1002/9780470089941.et0202s5

[74] J. Chory and J. D. Pollard Jr. Separation of small DNA fragments by conventional gel electrophoresis. *Current Protocols in Molecular Biology*, 47:2.7.1–8, 2001. doi:10.1002/0471142727.mb0207s47

[75] N. G. Walter. Probing RNA structural dynamics and function by fluorescence resonance energy transfer (FRET). *Current Protocols in Nucleic Acid Chemistry*, 11.10.1–23, 2003. doi:10.1002/0471142700.nc1110s11

Nucleic acid structures can be directly imaged by atomic force microscopy, which is well suited to extended two-dimensional structures, but less useful for discrete three-dimensional structures because of the microscope tip's interaction with the fragile nucleic acid structure; transmission electron microscopy and cryo-electron microscopy are often used in this case. Extended three-dimensional lattices are analyzed by X-ray crystallography.[76,77]

Applications

DNA nanotechnology provides one of the few ways to form designed, complex structures with precise control over nanoscale features. The field is beginning to see application to solve basic science problems in structural biology and biophysics. The earliest such application envisaged for the field, and one still in development, is in crystallography, where molecules that are difficult to crystallize in isolation could be arranged within a three-dimensional nucleic acid lattice, allowing determination of their structure. Another application is the use of DNA origami rods to replace liquid crystals in residual dipolar coupling experiments in protein NMR spectroscopy; using DNA origami is advantageous because, unlike liquid crystals, they are tolerant of the detergents needed to suspend membrane proteins in solution. DNA walkers have been used as nanoscale assembly lines to move nanoparticles and direct chemical synthesis. Furthermore, DNA origami structures have aided in the biophysical studies of enzyme function and protein folding.[9,21] Recently, the DNA-PAINT method has been used to make fluorescent barcodes for superresolution microscopy. Unlike traditional fluorophores, which have only been used to give 11 orthogonal signals, DNA-PAINT is able to provide 216 distinguishable barcodes.[78]

[76] C. Lin, Y. Ke, R. Chhabra, J. Sharma, Y. Liu, and H. Yan. Synthesis and characterization of self-assembled DNA nanostructures. In Zuccheri, G. and Samorì, B., “DNA nanotechnology: methods and protocols.” *Methods in Molecular Biology* 749:1–11, 2011. doi:10.1007/978-1-61779-142-0_1

[77] V. A. Bloomfield, D. M. Crothers, and I. Tinoco Jr. “Nucleic acids: structures, properties, and functions.” Sausalito, Calif: University Science Books, 2000. pp. 84–86, 396–407. ISBN 0-935702-49-0.

[78] C. Lin, R. Jungmann, A.M. Leifer, C. Li, D. Levner, G.M. Church, W. Shih, and P. Yin. Sub-micrometer geometrically encoded fluorescent barcodes self-assembled from DNA. *Nature Chem.*, 4:832–9, 2012. doi:10.1038/nchem.1451

DNA nanotechnology is moving towards potential real-world applications. The ability of nucleic acid arrays to arrange other molecules indicates its potential applications in molecular scale electronics. The assembly of a nucleic acid structure could be used to template the assembly of a molecular electronic elements such as molecular wires, providing a method for nanometer-scale control of the placement and overall architecture of the device analogous to a molecular breadboard. DNA nanotechnology has been compared to the concept of programmable matter because of the coupling of computation to its material properties.[9,20]

There are potential applications for DNA nanotechnology in nanomedicine, making use of its ability to perform computation in a biocompatible format to make "smart drugs" for targeted drug delivery. One such system being investigated uses a hollow DNA box containing proteins that induce apoptosis, or cell death, that will only open when in proximity to a cancer cell.[21,79] There has additionally been interest in expressing these artificial structures in engineered living bacterial cells, most likely using the transcribed RNA for the assembly, although it is unknown whether these complex structures are able to efficiently fold or assemble in the cell's cytoplasm. If successful, this could enable directed evolution of nucleic acid nanostructures.[20]

Acknowledgements

I wish to acknowledge Wikipedia users Baffle gab1978 and Dank for copyediting this article, and the following Wikipedia users for providing useful feedback during the Featured Article review process: Graham Colm, Noleander, Dank, Iridia, EdChem, RJHall, TCO, Tony1, Adabow, Ling.Nut3, Cryptic C62, Dcrjsr, Emw, and William Avery. A complete list of collaborators is available at http://tools.wmflabs.org/xtools/articleinfo/index.php?article=DNA_nanotechnology, with J. P. S. listed under the username Antony–22.

[79] R. Jungmann, S. Renner, F. C. Simmel. From DNA nanotechnology to synthetic biology. *HFSP J.*, 2:99–109, 2008. doi:10.2976/1.2896331

Figures 1.1, 1.2a, 1.3, and 1.4b were taken or modified from images published in the Public Library of Science journals, and are used under the terms of the Creative Commons Attribution 2.5 Generic license (see http://www.plosbiology.org/static/license.action). The source article for each image is referenced in the corresponding figure caption. Figure 1.4a was created by Wikimedia Commons user PiAndWhippedCream, who has released it into the public domain (see http://en.wikipedia.org/wiki/File:SierpinskiTriangle.svg). All other figures were created by J. P. S.

Page intentionally left blank

Chapter 2

Multisubjective: better nucleic acid design through fast removal of undesired secondary structure

"All science is either physics or stamp collecting."

—Ernest Rutherford

Abstract

Multisubjective is a nucleic acid design platform that improves the quality of sequence designs for dynamic nucleic acid reaction systems, and automates several tasks in the process of sequence design. Multisubjective identifies the specific bases that are responsible for undesired secondary structure, and redesigns only those bases using one of the supported client designers. Multisubjective combines features of combinatorial and thermodynamic designers, judiciously targeting the thermodynamic analysis to specific candidate designs on an occasional basis and using the fast combinatorial algorithms to do most of the design work. Multisubjective inherits the speed of the combinatorial methods, but with increased accuracy due to its use of the thermodynamic methods. This allows Multisubjective to successfully design large, complex nucleic acid systems in a reasonable amount of time.

Introduction

Computational nucleic acid sequence design is vital to many endeavors such as DNA nanotechnology, DNA computing, DNA microarrays, and DNA-templated synthesis.[1] The goal of nucleic acid design is to generate base sequences for a set of nucleic acid strands such that the strands efficiently assemble into a desired secondary structure. It is not sufficient for a successful design to simply contain the desired base pairings, which is a computationally easy problem. The design must also minimize unintended base pairings, which often interfere with the system's assembly or functionality either by favoring misassembled side products, or by affecting the kinetics of dynamic systems.

This problem has taken on even greater importance with the rise of dynamic DNA nanotechnology, where the design process targets not only the system's final conformation but also the intermediate states. This allows capabilities such as control of assembly order, and real-time reconfiguration of complexes in response to molecular triggers.[2] Design algorithms for traditional structural DNA nanotechnology benefit from a greater tolerance for undesired interactions; the thermal annealing process causes the stronger, desired assembly interactions occur earlier as the temperature is lowered, before the weaker undesired interactions are stable. However, dynamic systems lack this annealing step because they are triggered chemically rather than thermally, making it especially important for the design algorithm to minimize any undesired interactions that might potentially outcompete the desired interactions.

Common approaches to nucleic acid design can generally be placed into one of two families. The first family contains algorithms based on simple combinatorial rules such as sequence symmetry minimization,[3,4] mismatch distance maximization,[5] and more complicated algorithms based on coding

[1] D. Tulpan, M. Andronescu, S. B. Chang, M. R. Shortreed, A. Condon, H. H. Hoos, and L. M. Smith. Thermodynamically based DNA strand design. *Nucleic Acids Res.*, 33:4951–64, 2005. doi:10.1093/nar/gki773

[2] D. Y. Zhang. Towards domain-based sequence design for DNA strand displacement reactions. *Lect. Notes Comp. Sci.*, 6518:162–75, 2011. doi:10.1007/978-3-642-18305-8_15

[3] N. C. Seeman. Nucleic acid junctions and lattices. *J. Theor. Biol.*, 99:237–47, 1982. doi:10.1016/0022-5193(82)90002-9

theory.[6] These approaches tend to be based on zeroth-order nucleic acid thermodynamics, where the favorability of each base pair is considered to be identical, or on first-order models that distinguish only between weak and strong base pairs. These are fast, but the quality of the results that they give when evaluated using criteria based on a full thermodynamic model has been questioned.[1,7] While most of these algorithms are geared toward the static assembly approach of structural DNA nanotechnology, the Domain Design (DD) software uses an algorithm was specifically designed for use with the strand displacement complexes often used in dynamic DNA nanotechnology.[2]

The second family consists of algorithms based on the full nearest-neighbor thermodynamic model, which is a second-order approach that calculates a complex's free energy by considering each set of two adjacent paired bases, among other factors. This gives a much more accurate calculation of the complex's free energy, but comes at the cost of being much more computationally intensive. The Nucleic Acid Package (NUPACK) is commonly used for this purpose in DNA nanotechnology. It includes applications for thermodynamic analysis[8] as well as the design of complexes with either a single equilibrium target structure[9] or a multi-objective design where multiple conformations can be optimized simultaneously,[10] which are all available on a web server.[11] The multi-objective designer is intended for

[4] J. Seffert and A. Huhle. A full-automatic sequence design algorithm for branched DNA structures. *J. Biomol. Struct. Dyn.*, 25:453–66, 2008. doi: 10.1080/07391102.2008.10507193

[5] A. Brenneman and A. Condon. Strand design for biomolecular computation. *Theor. Comput. Sci.*, 287:39–58, 2002. doi:10.1016/S0304-3975(02)00135-4

[6] O. Milenkovic and N. Kashyap. On the design of codes for DNA computing. *Lect. Notes Comp. Sci.*, 3969:100–19, 2006. doi:10.1007/11779360_9

[7] R. M. Dirks, M. Lin, E. Winfree, and N. A. Pierce. Paradigms for computational nucleic acid design. *Nucleic Acids Res.*, 32:1392–403, 2004. doi:10.1093/nar/gkh291

[8] R. M. Dirks, J. S. Bois, J. M. Schaeffer, E. Winfree, and N. A. Pierce. Thermodynamic analysis of interacting nucleic acid strands. *SIAM Rev.*, 49:65–88, 2007. doi:10.1137/060651100

[9] J. N. Zadeh, B. R. Wolfe, and N. A. Peirce. Nucleic acid sequence design via efficient ensemble defect optimization. *J. Comput. Chem.*, 32:439–52, 2011. doi:10.1002/jcc.21633

[10] B. R. Wolfe, J. N. Zadeh, R. M. Dirks, and N. A. Pierce. NUPACK multi-objective design tutorial. Unpublished manuscript, 2010.

the design of reconfigurable complexes for which multiple secondary structures need to be taken into account, such as those used in dynamic DNA nanotechnology.

This paper describes a new sequence design platform, called Multisubjective, that is specifically designed to deal with concerns that are important for hairpin-based strand displacement systems such as those used in developmental self-assembly. Specifically, Multisubjective is targeted towards identifying strong spurious base pairing in single-stranded regions, in a context-sensitive scheme where the user can impose more stringent requirements on toeholds than the rest of the system. In addition, Multisubjective automatically generates and analyses a representative subset of intermediate structures, the open hairpins, and allows the user to specify additional custom intermediates. Lastly, Multisubjective allows a choice of client designer to redesign the small subset of bases responsible for the spurious base pairing, and supports automatic closed-cycle interaction with two designers. Multisubjective is designed to be easy to use, using a pre-existing file format for input, and can be used in either an interactive console mode, a command-line mode that allows it to be called from other programs, and a graphical user interface available on a web server.

Conceptually, Multisubjective has the ability to combine the combinatorial and thermodynamic approaches, inheriting the benefits of each. The slow but more accurate thermodynamic model is used only to identify the problematic bases, while a fast but less accurate simple combinatorial model can be used to reassign the identities of those bases. This setup strikes a favorable balance between the two types of design, generating results that are higher quality than the underlying combinatorial-only designer, while completing design tasks in much less time than thermodynamic-only designers.

[11] J. N. Zadeh, C. D. Steenberg, J. S. Bois, B. R. Wolfe, M. B. Pierce, A. R. Khan, R. M. Dirks, and N. A. Pierce. NUPACK: Analysis and design of nucleic acid systems. *J. Comput. Chem.*, 32:170–3, 2011. doi:10.1002/jcc.21596

Considerations

The genesis of this research came in the sequence design of systems for developmental self-assembly, which uses the methods of dynamic DNA nanotechnology to create structures similar to those made in structural DNA nanotechnology. This methodology allows specification of the entire assembly pathway rather than only the final structure, with an assembly order prescribed by a defined molecular program. This also confers several practical advantages, such as isothermal operation over a wide range of temperatures, and the ability to use a molecular input rather than a temperature change to trigger the assembly, allowing the possibility of having structures that reconfigure in computationally intricate ways based upon changes in their molecular environment.

The major type of reactant used in this methodology is a nucleic acid hairpin motif, rather than the unstructured strand used in traditional DNA nanotechnology. Each hairpin can be reconfigured from a closed to an open state through the binding of an initiator domain of the appropriate sequence through a strand displacement reaction, revealing output sequences that can then participate in a cascade to bind and open further downstream hairpins.[12] Another type of reactant called a cooperative hybridization complex[13] contains two toeholds that must be activated at the same time for the strand to assemble to the growing complex; this type of reactant is used for ring forming reactions. This methodology has previously been used to make structures such as one-dimensional polymers,[12] three- and four-arm junctions, cross-catalytic circuits, dendritic trees,[14] and a wireframe tetrahedron (Chapter 3).

These strand displacement reactions present special design considerations due to their dynamic nature. Specifically, the strand displacement domains must remain unstructured when they are

[12] R. M. Dirks and N. A. Pierce. Triggered amplification by hybridization chain reaction. *Proc. Natl. Acad. Sci. USA*, 101(43):15275–8, 2004. doi:10.1073/pnas.0407024101

[13] D. Y. Zhang. Cooperative hybridization of oligonucleotides. *J. Am. Chem. Soc.*, 133:1077–86, 2011. doi:10.1021/ja109089q

[14] P. Yin, H. M. T. Choi, C. R. Calvert, and N. A. Pierce. Programming biomolecular self-assembly pathways. *Nature*, 451:318–22, 2008. doi:10.1038/nature06451

temporarily single-stranded in intermediate stages of the assembly interaction, (Fig. 2.1a) as undesired secondary structure interferes with the kinetics of the desired strand displacement reactions. Furthermore, strong spurious binding between the toehold and tail regions of the hairpins must be avoided because it inhibits the toehold association step. Indeed, preliminary results have suggested that strong spurious interactions between the toehold and the tail regions of the hairpin slow the kinetics of the hairpin opening (Chapter 4),

We decided that we wanted to remove spurious secondary structure that binds strongly, that is, with a high pair probability. We assume that these strong spurious interactions will negatively affect the system's operation, while weaker interactions can be ignored since they will form only transiently and thus can be outcompeted by the desired strand displacement reactions. We thus designed Multisubjective with the goal of identifying a relatively small set of bases that should be redesigned to interrupt any spurious secondary structure that was found. In addition, Multisubjective automatically analyzes both the closed hairpin and an automatically generated transiently single-stranded open version that is a key intermediate in the assembly. While Multisubjective was designed with hairpin systems in mind, it is also capable of designing non-hairpin-based dynamic systems, as well as structural systems, if the intermediate structures are specified manually.

Implementation

Multisubjective is a general-use tool for designing sequences for systems with user-defined secondary stuctures. Multisubjective's algorithm is based on multiple iterations of an analysis–redesign cycle (Fig. 2.1b). The analysis identifies a minimal set of bases that need to be changed because they are responsible for undesired secondary structure, or because they are part of a user-defined prevented sequence. Multisubjective then instructs a client designer to redesign only those problematic bases, holding the others constant. Multiple iterations of this cycle lead to a succession of designs, with the goal of finding a design that is stable under continued iterations, that has very few undesired base pairs.

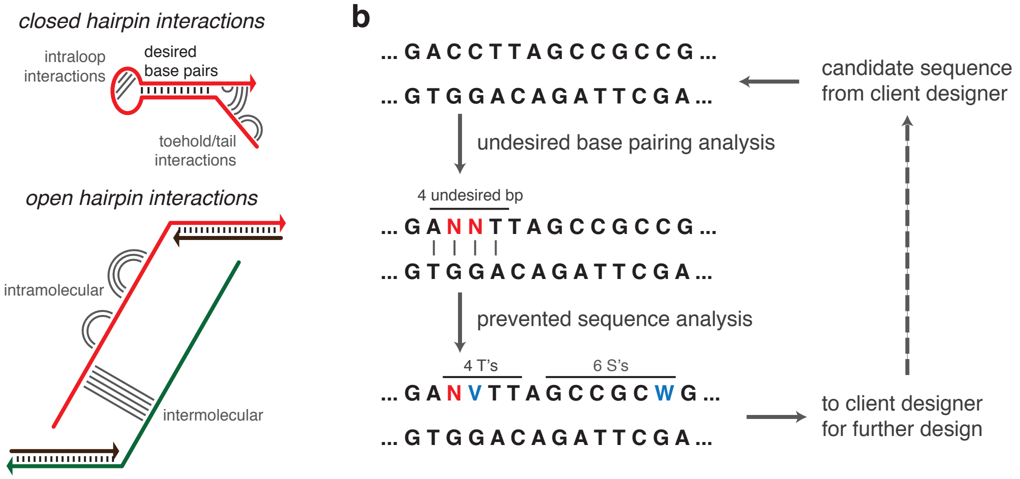

a
closed hairpin interactions
intraloop interactions
desired base pairs
toehold/tail interactions
open hairpin interactions
intramolecular
intermolecular
b
... G A C C T T A G C C G C C G ...
... G T G G A C A G A T T C G A ...
candidate sequence from client designer
undesired base pairing analysis
4 undesired bp
... G A N N T T A G C C G C C G ...
... G T G G A C A G A T T C G A ...
prevented sequence analysis
4 T's
6 S's
... G A N V T T A G C C G C W G ...
... G T G G A C A G A T T C G A ...
to client designer for further design

Figure 2.1: Design of nucleic acid hairpins. a, Some examples of undesired secondary structure. The closed hairpin (above) has a region of strong base paring between its toehold and its tail, which would inhibit hairpin opening. The single-stranded potion of the opened hairpin (bottom) should have no secondary structure, but there is again a strong region of secondary structure that would slow the strand displacement reaction. **b**, Flowchart of the Multisubjective algorithm, containing an example of its operation. A candidate sequence (upper left) is analyzed first for strong undesired base pairing (middle left) and then for prevented sequences (lower left), with certain positions being changed to degenerate bases as a result. The processed sequence is then passed to a client designer (right) that generates a new set of candidate sequences consistent with the constraints imposed by the degenerate bases and preserving all bases unchanged by Multisubjective. Multiple candidate sequences (not shown) can be analyzed in each cycle, with only the best sequence (i.e., the one with the fewest strong undesired base pairs) advancing to the next cycle.

In an effort to avoid recreating code that already exists, we used existing software packages where possible. A local copy of the NUPACK 3.0 analysis software is used to calculate pair probabilities for a candidate ensemble, which Multisubjective then analyzes to identify which bases should be changed. This is then passed to a client designer; Multisubjective interacts automatically with an implementation of the DD software, and is also capable of automatically submitting a design to the NUPACK multi-objective designer on the web server and automatically retrieving the results using the cURL[16] package. This seamless interaction with the NUPACK web server and with DD greatly decreases user turnaround time for generating new designs. In addition, Multisubjective can be run without any client designer, making random assignments for the identified bases, or the user can manually submit the output sequences to a designer of their choice.

[16] cURL is available at http://curl.haxx.se/

Structure. The reactants in developmental self-assembly are oligonucleotides that assume a hairpin structure: a double-stranded stem with a single-stranded loop on one side, and one or two single-stranded protrusions on the other. In the DU notation used by NUPACK, a hairpin is represented as Ut_1 Ds (Uℓ) Ut_2, where s is the stem length, ℓ is the loop length, and t_1 and t_2 are the toehold and tail lengths depending on the hairpin's polarity. A hairpin is considered to have positive polarity if the toehold is on the 5′ end (meaning that t_1 is the toehold length and t_2 is the tail length), and has negative polarity if the toehold is on the 3′ end (in which case the assignments are reversed). We will refer to the toehold length as t_{toe} and the tail length as t_{tail} regardless of polarity. Each hairpin has s desired base pairs at positions $(t_1 + n, t_1 + 2s + \ell + 1 - n)$ for n from 1 to s; all other base pairs are undesired.

In the course of the assembly, a strand displacement reaction converts each hairpin to an intermediate state that we term an open hairpin. In this conformation, the input region of length $t_{toe} + s$ is double stranded and bound to some other part of the growing assembly, and the output region(s) of length $\ell + s + t_{tail}$ is single stranded. Multisubjective automatically generates structures corresponding to the single-stranded portion of the open hairpins by removing the first $t_1 + s$ bases of positive polarity hairpins, or the final $t_2 + s$ bases of negative polarity hairpins. Open hairpin structures have no desired bases. Other intermediate structures or custom reactants can be input manually as "static" structures by the user; Multisubjective will automatically generate a list of desired base pairs from the user-provided structure definition.

Sequence. Each base in each structure is assigned an identity that can be any of the bases A, C, G, or T, or any of the degenerate bases representing each combination of these bases. In each iteration, Multisubjective receives a candidate sequence consisting of only non-degenerate bases. Multisubjective's analysis yields a processed sequence, usually containing degenerate bases that constrains the base identities that the client designer may assign to that position. Multisubjective uses the standard codes for degenerate bases used in molecular biology; for example, W represents an A or a T; V represents a C, G, or T; and N represents any of the four bases.

Each structure is also defined as a series of sequence domains, which are segments of specified length that can be repeated at various points within the sequence. This is necessary because the structured nature of the reactants creates sequence dependencies that constrain the identities of certain bases. These domain assignments are important because Multisubjective does analysis on strand sequences, where the multiple copies of each domain many be changed in different ways. Multisubjective uses an algorithm to reconcile the differences between copies of each domain to create a consistent set of domain sequences for the client designer to work from.

Metrics. The raw data analyzed by Multisubjective is a matrix of base pair probabilities P generated by the NUPACK analysis package. Each element $P_{i,j}$ represents the probability that the two bases with indexes i and j are paired within the thermodynamic ensemble of all possible structures. P is an N by N+1 matrix where N is the sum of the lengths of all structures considered by Multisubjective, both closed and open, with the extra column $P_{i,N+1}$ representing the probability that base i is unpaired. Multisubjective does not consider interactions between every pair of bases: all intramolecular interactions are considered, and if the user desires, the intermolecular interactions between pairs of open strands may also be considered. Any pair of bases that is not a member of these groups is assigned a zero value by NUPACK.

There are many ways to compile this data set into a single score to evaluate the quality of sequence sets generated by these designers, of which we have focused on two. The first is the normalized ensemble defect (NED), the percentage of bases expected to be mispaired at any given time within the ensemble. This is the metric used by NUPACK. We calculate the normalized ensemble defect by generalizing the equation provided in Ref. 9. We define the "correct partner" function $cp(i)$ as follows: if base i is part of a desired base pair (i,j) then $cp(i)$ evaluates to j, otherwise it evaluates to N+1, indicating an unpaired base. We then sum the probability that each base in each structure (including each open and closed hairpin structure) is paired to its correct partner:

$$NED = 1 - \frac{1}{N}\sum_{1 \le i \le N} P_{i,cp(i)}$$

The NED would evaluate to 0 for a perfect, defect-free structure where the pair probability matrix $P_{i,j}$ exactly matches the desired target structure—that is, all the elements $P_{i,cp(i)}$ representing desired base pairs evaluate to 1, while all the other elements representing undesired base pairs evaluate to 0. Lower-quality designs have higher NEDs because the pair probabilities for desired base pairs are lower, and those for undesired base pairs are higher. The theoretical maximum NED is 1, representing a structure that is never in its desired conformation. (The NED is calculated by Multisubjective and displayed for diagnostic purposes, but is not actually used in the algorithm.)

In addition, we recorded the number of "strong" undesired base pairs, N_{sup}, which we define as those having a pair probability greater than a user-supplied threshold value T. This is the metric that Multisubjective seeks to minimize. The choice of this metric reflects a tradeoff where strong undesired base pairs are removed, at the potential cost of increasing the number of weak undesired base pairs. The relative effects of strong versus weak undesired base pairs has not yet been investigated empirically, but it is possible that weaker undesired base pairs are less problematic because they are more likely to be outcompeted by the desired interactions.

$$N_{sup} = \sum_{\substack{1 \le i \le N \\ 1 \le j \le N+1 \\ j \ne cp(i) \\ P_{i,j} > T}} 1$$

The normalized ensemble defect has the advantage that it reflects the entire profile of undesired pair probabilities, as it is essentially a weighted average that is affected more by strong than weak base pairs. The number of strong undesired base pairs, on the other hand, ignores the large numbers of very weak undesired base pairs that these types of systems tend to have.

Algorithm

Multisubjective can be run either though an interactive text-based interface, can take options as part of the command line call. This latter option allows Multisubjective to be integrated into a front-end graphical user interface as part of a web server that is currently under development.[17] Multisubjective's input for the system specification is an augmented form of the NUPACK ".np" format containing additional information about the function of each strand, as well as settings specific to Multisubjective. The user-specified parameters are the pair probability threshold, an optional lower threshold for toeholds, the identities of immutable bases, and whether intermolecular interactions are to be considered.

Pseudocode for the algorithm is shown in Fig. 2.2. The initial candidate sequences themselves can be taken either from the ".np" specification file, from a ".npo" file generated by the NUPACK multi-objective designer, from a ".dd" file used natively by DD, or can be generated randomly according to sequence constraints in the ".np" specification file. Multisubjective is capable of automatically downloading a specified job from the NUPACK web server and extracting the ".npo" files for analysis.

Once the data have been loaded from the input file(s), Multisubjective uses the local installation of NUPACK to analyze the secondary structure of the strands. This is done in two batches. The first batch contains the full sequence of each hairpin, testing the closed hairpin structures, with a maximum complex size of one. The second pass contains the hairpins with their input regions removed, giving the open hairpin structures (technically, these contain only the single-stranded portion of the open hairpins, omitting the double-stranded portion for simplicity). The user may choose to only consider intramolecular interactions at this stage, or to also consider intermolecular interactions between pairs of open hairpin structures. The closed structures are explicitly defined in the specification file, and Multisubjective automatically generates the open hairpin structures from these.

[17] C. Grun, J. Werfel, D. Y. Zhang, and P. Yin. DyNAMiC Workbench: an integrated development environment for dynamic DNA nanotechnology. Manuscript in preparation.

```
Load structural specification
Load or generate sequence(s)
FOR each candidate sequence
    Run NUPACK to generate base pairing probabilities
    FOR each possible base pair in system
        IF pair is undesired, and not set as immutable
            Change central bases to N
    FOR each base in system
        IF prevented sequence detected
            Change last N or last mutable base to appropriate degenerate base
Select best candidate sequence
Construct domain sequences from strand sequences; manage collisions
If desired, send processed sequence to designer to generate new candidate sequences
```

Figure 2.2: Pseudocode for the Multisubjective algorithm

Multisubjective then tabulates the undesired secondary structure from the NUPACK analysis, and decides which bases need to be changed to disrupt this secondary structure. Multisubjective only considers undesired base pairs that have a pair probability higher than a user-provided threshold; the user may optionally set a lower threshold for interactions involving toeholds. Multisubjective groups these undesired base pairs into consecutive runs, that is, series of pairs of the form (i, j), $(i+1, j-1)$ … $(i+n, j-n)$. It targets for change each *central base pair* $(i+1, j-1)$ through $(i+n-1, j-n+1)$; for runs of two base pairs, the later one $(i+1,j-1)$ is changed, and all isolated base pairs (i, j) are changed as well. Multisubjective does not change any of the desired base pairs or the immutable bases specified by the user. Multisubjective changes one base from each of the targeted base pairs to the degenerate base N, changing the first member of the pair unless it is immutable, in which case the other member of the base pair is changed. If they are both immutable, no bases are changed and a warning is logged.

Multisubjective also checks for runs of consecutive bases of a certain identity, including both single and degenerate bases. These are called *prevented sequences*, and Multisubjective inserts the proper mixed base to disrupt this sequence, again avoiding changing immutable bases. As an example, if there are too many A's in a row, one of them will be changed to a B; if there are too many W's in a row, one will be changed to an S. The algorithm seeks to minimize the number of changed bases by changing an N to the mixed base if one is in the proper range of positions; otherwise the last base that is not immutable is changed. If all bases in the prevented sequence are immutable, a warning is logged and no change is made.

Once these changes have been made to the strand sequences, the domain sequences must be extracted for output. Each domain may appear many times throughout the system, and each instance may have been changed in different ways due to its local environment. Thus, a base collision scheme is used to resolve these differences and produce a single, unified domain sequence reflecting all the changes made to any instance of the domain sequence. The algorithm iterates over each instance of each domain, adding their changes to the unified domain sequence in turn. When the bases do not match, mixed bases other than N take precedence over N's, N's take precedence over single bases, and for two non-N bases, the intersection of the two bases is taken unless the intersection is empty, in which case no change is made to the unified sequence and a warning is logged.

Once the strand sequences have been compiled, they are written to output files in both the ".dd" format and a ".msq" format that uses the same format as the sequence specifications in the ".np" specification file. The processed sequences are then passed to the client designer. If DD is used, Multisubjective automatically runs the DD algorithm to produce ten new candidate sequences for the next round. If submission to the NUPACK web server is desired, Multisubjective generates a formatted version of the NUPACK multi-objective design file with the new block sequence assignments inserted, and submits an HTTP POST request to the NUPACK web server. Multisubjective can also be used by itself, in which case random bases are used within the mixed-base constraints it has specified.

Multisubjective loops through multiple cycles involving first the analysis of a set of candidate sequences, followed by the client designer processing the best of the analyzed candidate to generate a new set of candidate sequences. In each round, the candidate sequence with the lowest number of strong undesired base pairs is chosen to be passed to the designer. Multisubjective is considered to have reached its endpoint when the same sequence is repeatedly selected in consecutive rounds; this happens when the number of undesired base pairs becomes very low, causing the sequence space allowed to the designer to be small enough that no mutation improves the design. Thus, Multisubjective has the inherent feature that the stable points to which the sequences converge are necessarily successful designs that contain little or no undesired secondary structure. Note that as the rounds progress, the chosen sequence is not guaranteed to become monotonically better.

Results

Multisubjective version 1.0.9 was used in these studies. We used DD version 0.3 with command-line interface version 0.4.7.3. We used a local copy of NUPACK version 3.0 for the analysis, and for the NUPACK multi-objective designer we used version 3.2 on their web server. The Multisubjective settings used were a threshold of 0.6, with no lower toehold threshold, no immutable bases, and the intermolecular interactions turned on. We initiated each run using the option to select the best design out of ten independent runs of DD.

We ran trials on three developmental self-assembly systems. The first is a nanoscale wireframe tetrahedron that self-assembles from six hairpins and three cooperative hybridization complexes (Fig. 2.3a). The assembly of this structure begins with a central three-arm junction containing the six hairpins; each pair of arms is then involved in a ring forming reaction involving a cooperative hybridization complex. Each edge of the assembled tetrahedron has five turns of DNA measuring 18 nm in length. The synthesis and characterization of this structure is described in Chapter 3. The second structure is the "pizza" structure, a flat hexagonal wireframe structure with six triangular faces (Fig. 2.3b). This structure assembles similarly to the tetrahedron, except that it starts from a central six-arm junction containing twelve hairpins, leading to six ring formation reactions using six cooperative complexes. The final is a flat solid triangular structure containing 28 unique hairpins that assemble in a row-by-row fashion (Fig. 2.3c).

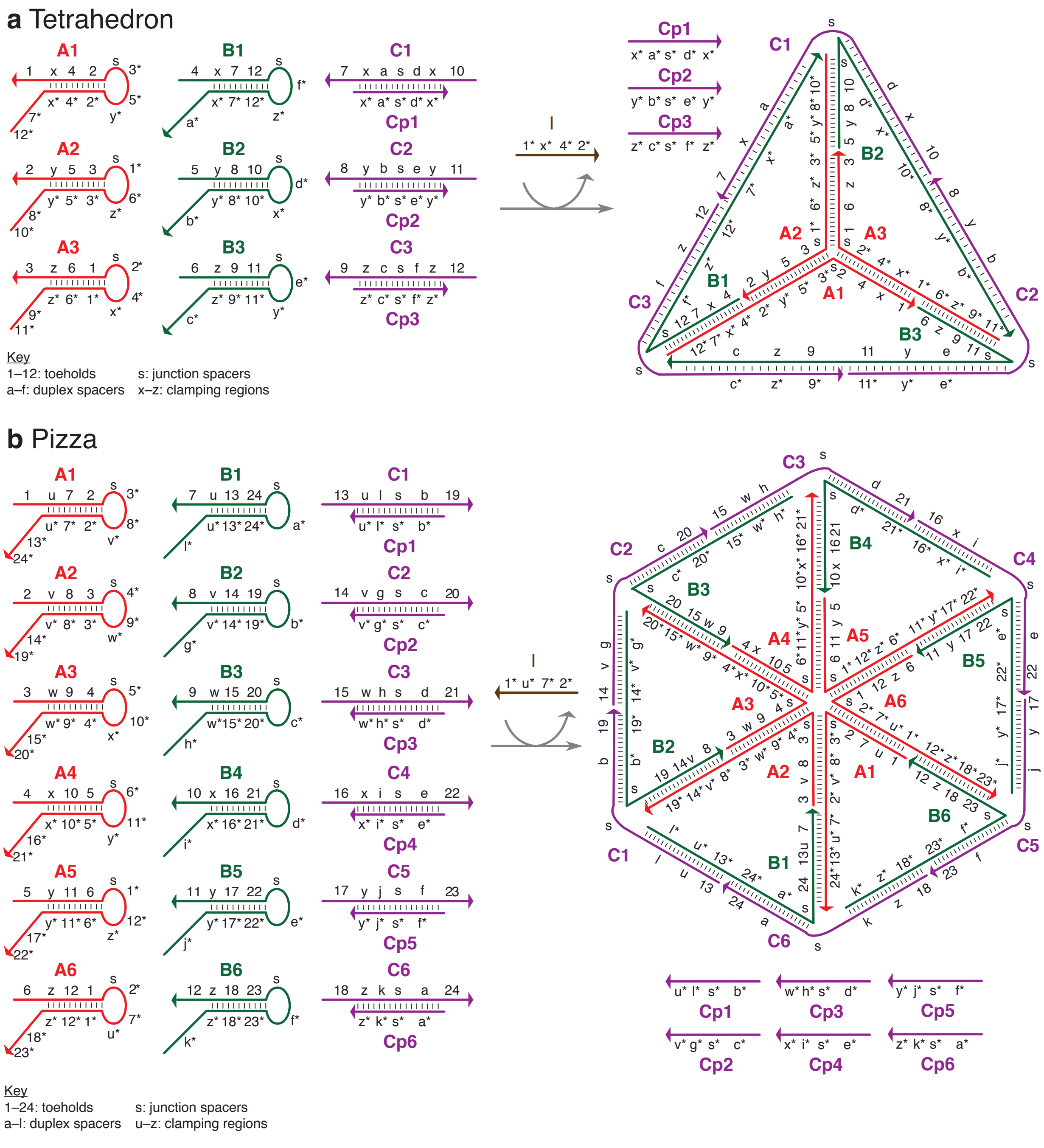


Figure 2.3: Secondary structure of systems trialed in this research. a, Wireframe tetrahedron. **b**, Wireframe pizza. **c** (next page), Solid triangle.

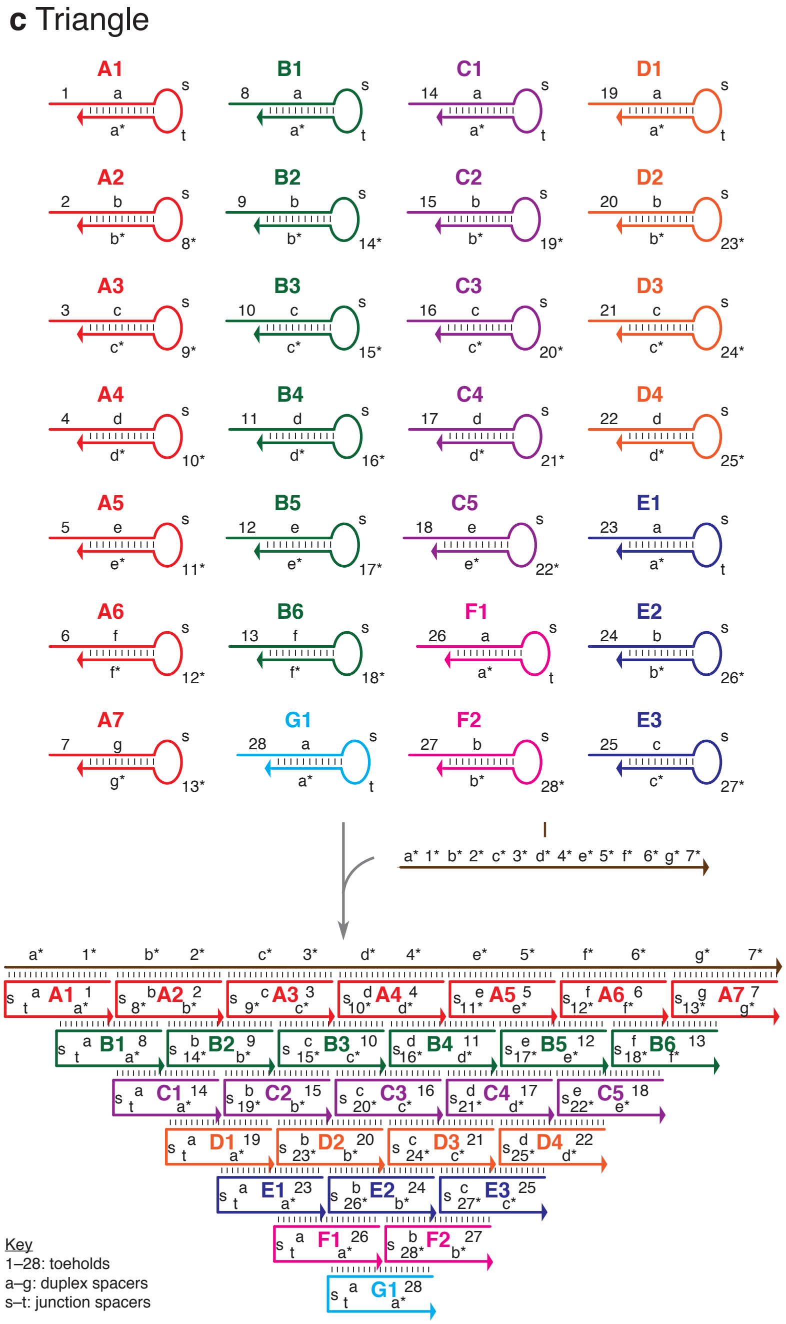


Figure 2.3 (Continued).

Multisubjective was able to successfully design sequences for all three systems, in that the resultant designs had few or zero strong undesired base pairs (Fig. 2.4). On the other hand, the NUPACK multi-objective designer was able to successfully design only the tetrahedron, the smallest of the three structures, with the other two rejected by it as being overconstrained. For all systems, we observe that Multisubjective yields sequences that have a very low number of strong undesired base pairs as compared to either random sequences or sequences designed by DD alone. While Multisubjective removes most strong undesired base pairs, it causes an increase in weaker ones, a phenomenon not observed with NUPACK multi-objective designer. Multisubjective tends to give sequences with a lower ensemble defect than random sequences, and in the same range as designs from DD alone. Note, however, that DD does not have the capability to respect the prevented sequence constraints that Multisubjective applies. This means that DD gives that have lower ensemble defects, but are in a part of the sequence space that is disallowed by Multisubjective due to prevented sequence constraints.

The trajectories connecting the best candidates in each round of one run of Multisubjective–DD shows that the designs tend to get incrementally better, but not necessarily monotonically (Fig. 2.5). The trajectories for the Multisubjective–Random sequences tend to be erratic and do not converge to a stable point, demonstrating the importance of applying DD's combinatorial rules. We observe that if the pair probability threshold is set too low, which is a more stringent condition, the number of undesired base pairs will plateau in a higher range than is needed to find a stationary point in a reasonable amount of time; therefore, the user must set a threshold that is high enough for the sequences to converge.

We also compared the running times of each designer. A single run of the NUPACK multi-objective designer takes overnight; while Multisubjective–DD takes only on the order of an hour to run for the tested structures. Having a quick running time becomes more important as the complexity of nucleic acid systems increases; the tetrahedron system, containing twelve strands in nine reactants, would

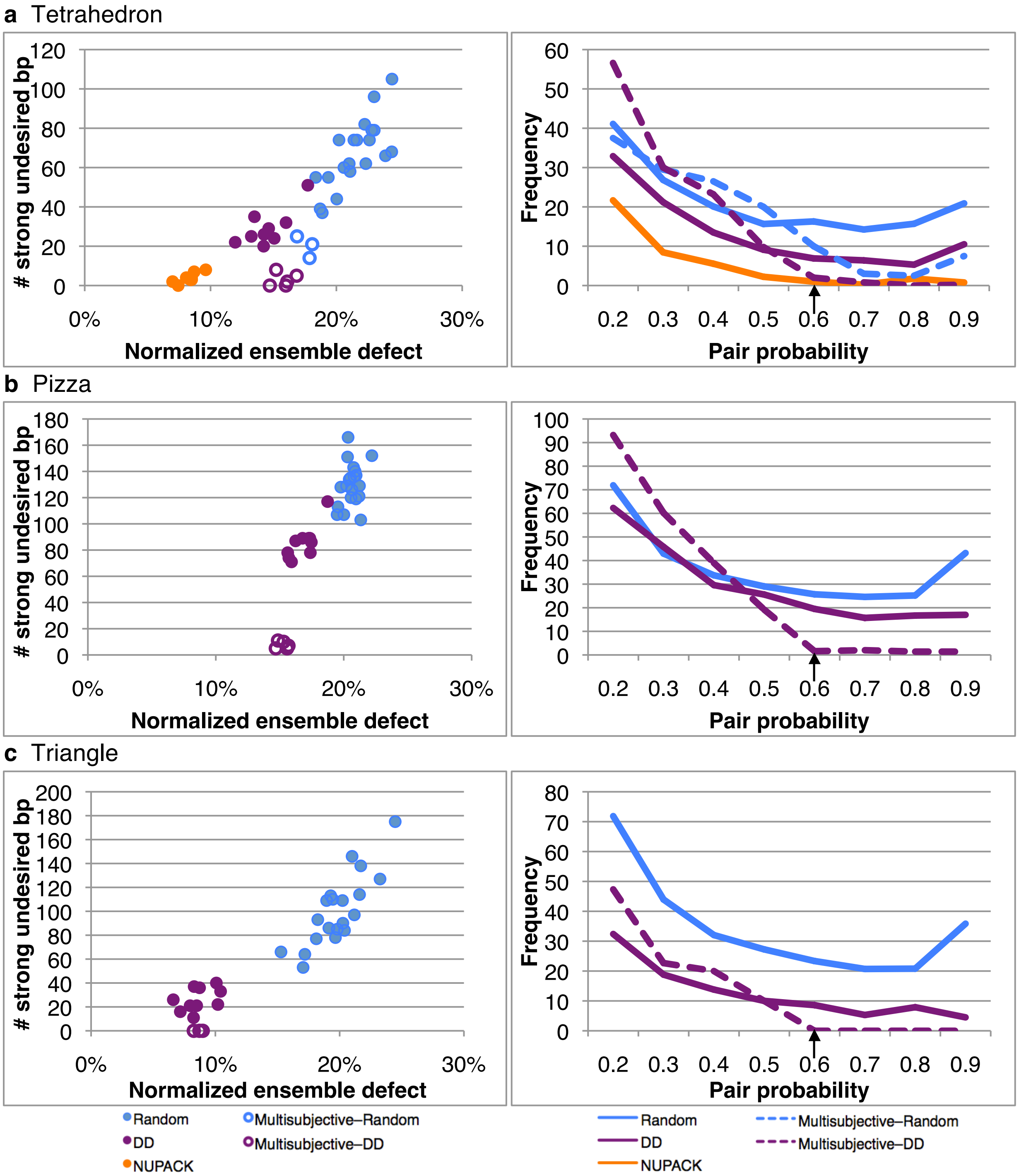


Figure 2.4: Performance of various designers. Two graphs are shown for each of the three structures. On the left, a scatter plot showing the quality of multiple designs from each designer. The vertical axis is the number of undesired base pairs in the system that have a pair probability greater than 0.6, and the horizontal axis is the normalized ensemble defect. Better designs have lower values for both metrics and are closer to the origin. Filled circles represent independent runs each designer without Multisubjective, while hollow circles represent successive iterations of Multisubjective in conjunction with the specified designers. On the right, a histogram of the pair probabilities for the undesired base pairs in each system. Each bin is 0.1 wide; the labels on the horizontal axis represent the result of truncating the pair probability. The 0.0 and 0.1 bins are omitted for clarity. The arrow represents the 0.6 pair probability threshold. **a**, Wireframe tetrahedron. **b**, Wireframe pizza. **c**, Solid triangle.

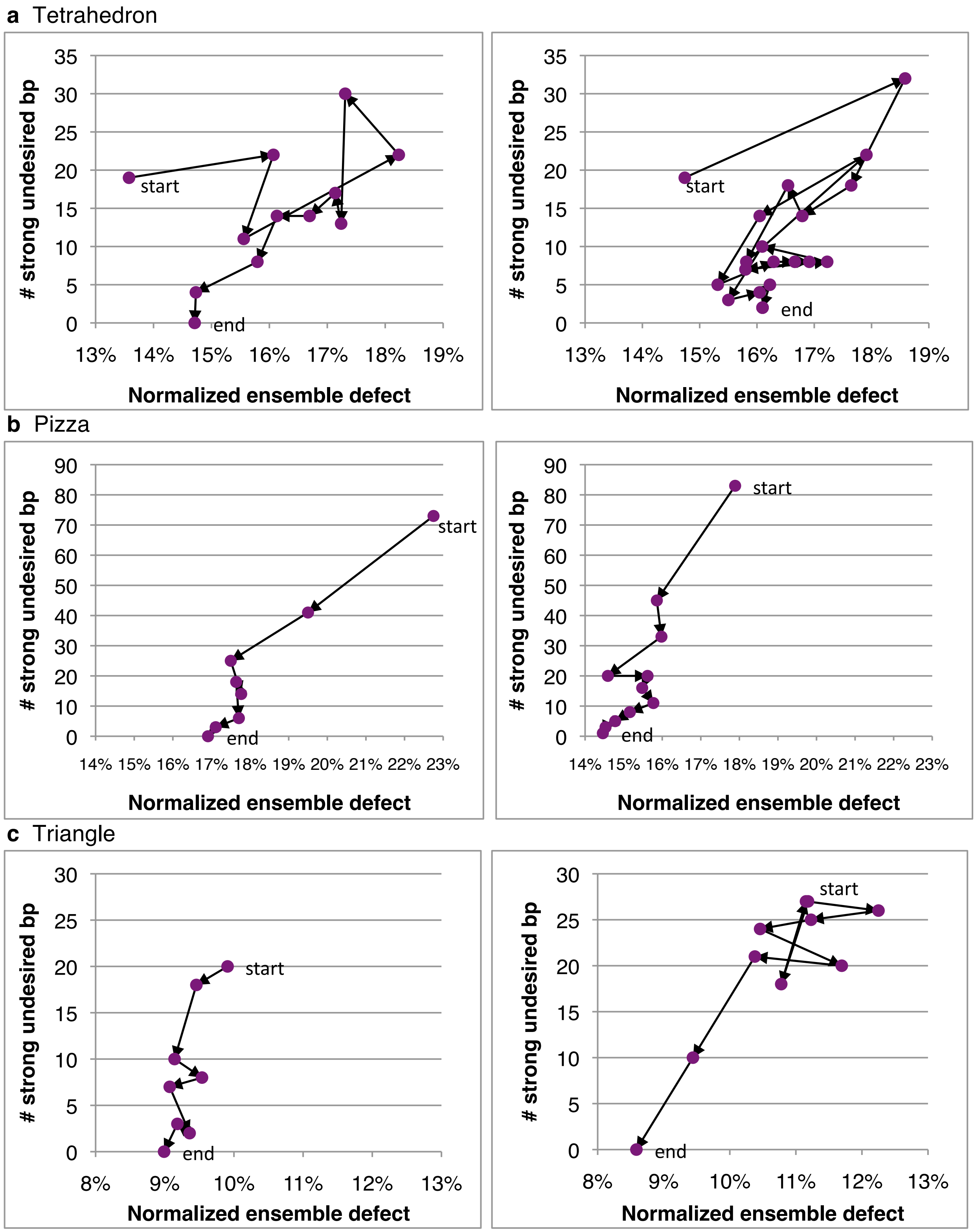


Figure 2.5: Trajectories through multiple rounds of Multisubjective. These graphs each show progression of a single design through successive rounds of Multisubjective's analyze–redesign cycle. Each dot represents the best sequence (the one with the lowest number of out of the ten candidate sequences from a single round. These are shown on a scatter plot with the same axes as the top potions of Fig. 2.4.

seem to be pushing an upper limit for thermodynamic-only designers, but Multisubjective's hybrid combinatorial–thermodynamic approach would allow the design of much more complex systems.

Discussion

Multisubjective can be seen as combining features of two classes of designers: combinatorial and thermodynamic. Simple combinatorial rules yield designs quickly, but at the cost of lower quality. Algorithms based on the full nearest-neighbor thermodynamic model give a much more accurate calculation of the complex's free energy, but this comes at the cost of being slower.[7] Multisubjective inherently incorporates both of these approaches in such a way that it minimizes use of NUPACK's thermodynamic analysis, judiciously targeting it to specific candidate designs on an occasional basis, and using the power of DD's combinatorial model to do most of the design work. This balance enables Multisubjective to inherit the speed of the combinatorial methods, as well as an improved accuracy due to its use of the thermodynamic methods. Multisubjective may also benefit from the fact that multiple bases are available for change in each cycle; this makes it less likely to become stuck in a local minimum than algorithms that attempt to change a single base at a time.

Multisubjective's complexity is limited by that of the NUPACK analysis algorithm, which has a complexity of $O(n^3)$ in the unpseudoknotted case used here.[8,18] DD contributes a lower complexity of $O(n^2)$ when used as a client designer,[2] while Multisubjective's internal algorithm is linear in the number of base pairs it needs to process, which is roughly $O(n^2)$. By comparison, while the complexity of the NUPACK multi-objective designer has not yet been reported, the single-objective designer's complexity is also $O(n^3)$.[9] We note that Multisubjective's significant reduction in observed running time implies that its constant factors are lower; this is in accordance of our strategy with minimizing use of the $O(n^3)$ thermodynamic algorithms in favor of the $O(n^2)$ combinatorial ones.

[18] R. M. Dirks and N. A. Pierce. A partition function algorithm for nucleic acid secondary structure including pseudoknots. *J. Comput. Chem.*, 24:1664–77, 2003. doi:10.1002/jcc.10296

The exact parameters of the designer, such as the pair probability threshold and the mechanism for deciding which bases to redesign, are subjective and could withstand further optimization. In addition, the designers were evaluated using theoretical criteria; an experimental comparison of sequences generated by different designers remains to be accomplished.

Acknowledgements

I thank Casey Grun for updating and maintaining the DD source code and programming the DD command-line interface, and Luvena Ong for providing the motif design for the solid triangle structure. David Zhang and Peng Yin provided useful discussions and suggestions on the manuscript.

Page intentionally left blank

Chapter 3

Developmental self-assembly of a DNA tetrahedron

The work described in this chapter was done in collaboration with Colby R. Calvert, David Yu Zhang, Niles A. Pierce, and Peng Yin.

“It’s not safe out here. It’s wondrous, with treasures to satiate desires both subtle and gross.

But it’s not for the timid.”

—Q, as written by Maurice Hurley

Abstract

Kinetically-controlled isothermal growth is fundamental to biological development, but it remains challenging to rationally design molecular systems that self-assemble isothermally into complex geometries via prescribed self-assembly and disassembly pathways. By exploiting the programmable chemistry of base pairing, sophisticated spatial and temporal control have both been demonstrated in DNA self-assembly, but largely as separate pursuits. By integrating temporal with spatial control, here we demonstrate the "developmental" self-assembly of a DNA tetrahedron, where a prescriptive molecular program orchestrates the kinetic pathways by which DNA molecules isothermally self-assemble into a well-defined three-dimensional wireframe geometry. In this reaction, nine DNA reactants initially co-exist in a metastable state, but upon catalysis by a DNA initiator molecule, navigate 24 individually characterizable intermediate states via prescribed assembly pathways, organized both in series and in parallel, to arrive at the tetrahedral final product. In contrast to previous work on dynamic DNA nanotechnology, this developmental program coordinates growth of ringed substructures into a three-dimensional wireframe superstructure, taking a step towards the goal of kinetically-controlled isothermal growth of complex three-dimensional geometries.

Introduction

Molecular self-assembly, a fundamental process underlying the development and operation of biological organisms, has emerged as an important engineering paradigm for nanotechnology. Biological development controls of molecular arrangement both spatially and temporally to produce a complex organism that robustly responds to its chemical and physical environment. In contrast, it has remained challenging to achieve sophisticated spatial and temporal control in an integrated fashion in rationally designed synthetic biomolecular systems.

There have been many recent advances in the design of sophisticated synthetic nucleic acid systems that enable either spatial or temporal control of molecular self-assembly, but these two capabilities have heretofore largely been relegated to separate realms. Past work in structural DNA nanotechnology,[1–3] including the synthesis of ribbons,[4,5] tubes,[5,6] two- and three-dimensional extended crystals,[6–11] and discrete objects,[12–21] has largely had the goal of engineering static target structures, without

[1] N. C. Seeman. Nucleic acid junctions and lattices. *J. Theor. Biol.*, 99:237–47, 1982. doi:10.1016/0022-5193(82)90002-9

[2] N. C. Seeman. DNA in a material world. *Nature*, 421:427–31, 2003. doi:10.1038/nature01406

[3] W. M. Shih and C. Lin. Knitting complex weaves with DNA origami. *Curr. Opin. Struct. Biol.*, 20:276–82, 2010. doi:10.1016/j.sbi.2010.03.009

[4] R. Schulman and Erik Winfree. Synthesis of crystals with a programmable kinetic barrier to nucleation. *Proc. Natl. Acad. Sci. USA*, 104:15236–41, 2007. doi:10.1073/pnas.0701467104

[5] P. Yin, R. F. Hariadi, S. Sahu, H. M. T. Choi, S. H. Park, T. H. LaBean, and J. H. Reif. Programming DNA tube circumferences. *Science*, 321:824–6, 2008. doi:10.1126/science.1157312

[6] H. Yan, S. H. Park, G. Finkelstein, J. H. Reif, and T. H. LaBean. DNA-templated self-assembly of protein arrays and highly conductive nanowires. *Science*, 301:1882–4, 2003. doi:10.1126/science.1089389

[7] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–44, 1998. doi:10.1038/28998

[8] H. Yan, T. H. LaBean, L. Feng, and J. H. Reif. Directed nucleation assembly of DNA tile complexes for barcode-patterned lattices. *Proc. Natl. Acad. Sci. USA*, 100:8103–8, 2003. doi:10.1073/pnas.1032954100

[9] D. Liu, M. Wang, Z. Deng, R. Walulu, and C. Mao. Tensegrity: construction of rigid DNA triangles with flexible four-arm DNA junctions. *J. Am. Chem. Soc.*, 126:2324–5, 2004. doi:10.1021/ja031754r

an explicit focus on controlling the assembly order and transient dynamics of how individual units come together to produce such a target structure. Conversely, past work in dynamic DNA nanotechnology,[22,23] including demonstrations of reconfigurable devices,[24,25] autonomous logical circuits,[26–29] dynamic self-

[10] P. W. K. Rothemund, N. Papadakis, and E. Winfree. Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biol.*, 2:2041–53, 2004. doi:10.1371/journal.pbio.0020424

[11] J. P. Zheng, J. Birktoft, Y. Chen, T. Wang, R. J. Sha, P. Constantinou, S. Ginell, C. D. Mao, and N. C. Seeman. From molecular to macroscopic via the rational design of a self-assembled 3D DNA crystal. *Nature*, 461:74–7, 2009. doi:10.1038/nature08274

[12] J. Chen and N. C. Seeman. The synthesis from DNA of a molecule with the connectivity of a cube. *Nature*, 350:631–3, 1991. doi:10.1038/350631a0

[13] P. W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440:297–302, 2006. doi:10.1038/nature04586

[14] Y. He, T. Ye, M. Su, C. Zhang, A. E. Ribbe, W. Jiang, and C. Mao. Hierarchical self-assembly of DNA into symmetric supramolecular polyhedra. *Nature*, 452:198–201, 2008. doi:10.1038/nature06597

[15] S. M. Douglas, H. Dietz, T. Liedl, B. Högberg, F. Graf, and W. M. Shih. Self-assembly of DNA into nanoscale three-dimensional shapes. *Nature*, 459:414–8, 2009. doi:10.1038/nature08016

[16] H. Dietz, S. M. Douglas, and W. M. Shih. Folding DNA into twisted and curved nanoscale shapes. *Science*, 325:725–30, 2009. doi: 10.1126/science.1174251

[17] D. Han, S. Pal, J. Nangreave, Z. Deng, Y. Liu, and H. Yan. DNA origami with complex curvatures in three-dimensional space. *Science*, 332:342–6, 2011. doi:10.1126/science.1202998

[18] B. Wei, M. Dai, and Peng Yin. Complex shapes self-assembled from single-stranded DNA tiles. *Nature*, 485:623–6, 2012. doi:10.1038/nature11075

[19] Y. Ke, L. Ong, W. Shih, and P. Yin. Three-dimensional structures self-assembled from DNA bricks. *Science*, 338:1177–83, 2012. doi:10.1126/science.1227268

[20] J. Sobczak, T. Martin, T. Gerling, and H. Dietz. Rapid folding of DNA into nanoscale shapes at constant temperature. *Science*, 338:1458–61, 2012. doi:10.1126/science.1229919

[21] D. Han, S. Pal, Y. Yang, S. Jiang, J. Nangreave, Y. Liu, and H. Yan. DNA gridiron nanostructures based on four-arm junctions. *Science*, 339:1412–5, 2013. doi:10.1126/science.1232252

[22] D. Y. Zhang and G. Seelig. Dynamic DNA nanotechnology using strand-displacement reactions. *Nature Chem.*, 3:103–13, 2011. doi:10.1038/nchem.957

[23] J. Bath and A. J. Turberfield. DNA nanomachines. *Nature Nanotech.*, 2:275–84, 2007. doi:10.1038/nnano.2007.104

[24] B. Yurke, A. J. Turberfield, A. P. Mills, F. C. Simmel, and J. L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406:605–8, 2000. doi:10.1038/35020524

assembling systems,[28–31] and walkers,[28,32–35] has typically focused on either engineering the transient interaction pattern of the individual molecular species to achieve desired computational or kinetic behavior, without explicit concern for the structural properties that the assembling and disassembling molecular species produce, or has introduced limited reconfigurability to an otherwise static structure.

To capture the complexity and robustness of molecular self-assembly demonstrated by the biological developmental process, it is necessary to design the temporal in addition to spatial order of self-assembly. This would allow direct, molecular-scale kinetic control over the entire assembly pathway, rather than being limited to specification of the final structure only. Such a system would no longer need a thermal annealing step to initiate the assembly and encourage the dominance of the desired, lowest-

[25] H. Yan, Z. Zhang, X. Shen, and N. C. Seeman. A robust DNA mechanical device controlled by hybridization topology. *Nature*, 415:62–5, 2002. doi:10.1038/415062a

[26] G. Seelig, D. Soloveichik, D. Y. Zhang, and E. Winfree. Enzyme-free nucleic acid logic circuits. *Science*, 314:1585–8, 2006. doi:10.1126/science.1132493

[27] D. Y. Zhang, A. J. Turberfield, B. Yurke, and E. Winfree. Engineering entropy-driven reactions and networks catalyzed by DNA. *Science*, 318:1121–5, 2007. doi:10.1126/science.1148532

[28] P. Yin, H. M. T. Choi, C. R. Calvert, and N. A. Pierce. Programming biomolecular self-assembly pathways. *Nature*, 451:318–22, 2008. doi:10.1038/nature06451

[29] L. Qian and E. Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332:1196–201, 2011. doi:10.1126/science.1200520

[30] R. M. Dirks and N. A. Pierce. Triggered amplification by hybridization chain reaction. *Proc. Natl. Acad. Sci. USA*, 101(43):15275–8, 2004. doi:10.1073/pnas.0407024101

[31] S. Venkataraman, R. M. Dirks, P. W. K. Rothemund, E. Winfree, and N.A. Pierce. An autonomous polymerization motor powered by DNA hybridization. *Nature Nanotech.*, 2:490–4, 2007. doi:10.1038/nnano.2007.225

[32] W. B. Sherman and N. C. Seeman. A precisely controlled DNA biped walking device. *Nano Lett.*, 4:1203–7, 2004. doi:10.1021/nl049527q

[33] J.-S. Shin and N. A. Pierce. A synthetic DNA walker for molecular transport. *J. Am. Chem. Soc.*, 126:10834–5, 2004. doi:10.1021/ja047543j

[34] P. Yin, H. Yan, X. G. Daniell, A. J. Turberfield, and J. H. Reif. A unidirectional DNA walker that moves autonomously along a track. *Angew. Chem. Int. Ed.*, 43:4906–11, 2004. doi:10.1002/anie.200460522

[35] T. Omabegho, R. Sha, and N. C. Seeman. A bipedal DNA brownian motor with coordinated legs. *Science*, 324:67–71, 2009. doi:10.1126/science.1170336

energy product; the reaction could instead proceed isothermally and *in situ* with a molecular input to trigger the assembly. This would allow the construction of a "developmental" self-assembly system where a synthetic structure "grows" following prescribed kinetic assembly pathways organized both in series and in parallel, with some reactions occurring sequentially and others occurring simultaneously, ultimately developing into the desired complex target structure.

To address this largely unexplored challenge of integrating temporal control with spatial control in synthetic molecular self-assembly, we demonstrate here the rational design and kinetically-controlled isothermal synthesis of a DNA tetrahedron with a well-defined three-dimensional wireframe structure. The assembly is the execution of a "developmental molecular program" specified using a reaction graph abstraction that describes the kinetic pathways by which the DNA reactants self-assemble and disassemble. This work builds on a previous demonstration of programming molecular self-assembly and disassembly pathways using a versatile DNA hairpin motif that was previously used to execute diverse molecular programs including catalytic formation of branched junctions, cross-catalytic circuitry, conditional assembly of dendritic structures, and autonomous locomotion.[28] Unlike these previous demonstrations, the current molecular program yields a well-defined three-dimensional wireframe structure formed from closed rings. In contrast to DNA origami and tile-based approaches to structural DNA nanotechnology, our method is initiated by a catalytic molecular trigger, works isothermally within a wide temperature range, and follows a predetermined kinetic pathway. We envision that our strategy for rationally designing developmental molecular programs can be generalized to more complex three-dimensional wireframe constructions.

Kinetic pathway design

The tetrahedron assembles from six hairpins,[28] divided into **A** and **B** groups, and three cooperative hybridization complexes[36] making up the **C** group (Fig. 3.1a). The hairpins are designed to

[36] D.Y. Zhang. Cooperative hybridization of oligonucleotides. *J. Am. Chem. Soc.*, 133:1077–86, 2011. doi:10.1021/ja109089q

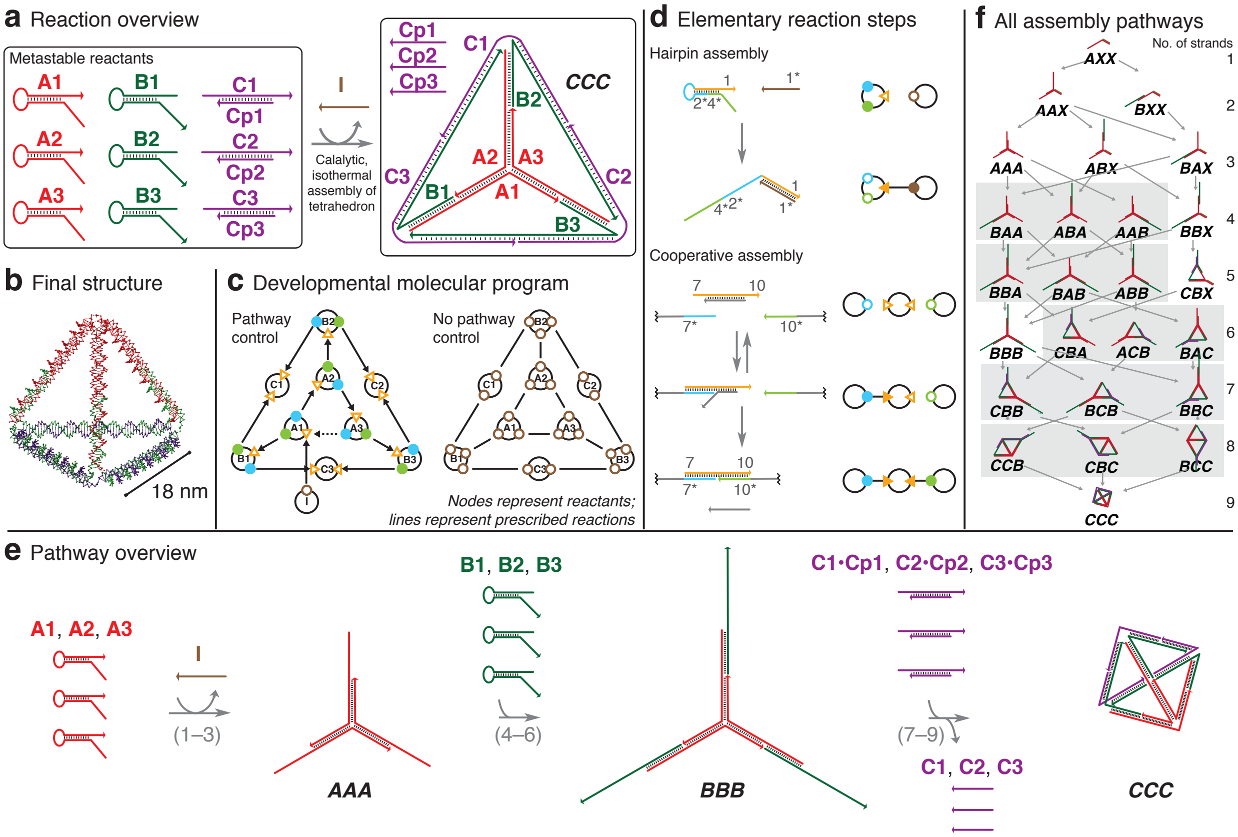


Figure 3.1: Catalytic self-assembly of a DNA tetrahedron. **a**, Overview of the reaction. **b**, A computer-rendered model (Nanoengineer-1, http://nanoengineer-1.com/) of the tertiary structure of the assembled tetrahedron. **c**, The reaction graph of the developmental molecular program specifying kinetically controlled self-assembly (left) compared to a traditional self-assembly process that lacks pathway control (right). Solid and dashed arrows depict kinetically controlled assembly and disassembly operations; line segments depict assembly operations that are not kinetically controlled. **d**, Execution schematics of two elementary reactions. Left, molecular structures; right, corresponding nodal abstractions, where lines are added to connect ports once assembly has occured. The strand regions are colored the same as the corresponding ports in the nodal representation. Top, a hairpin assembly reaction. Bottom, a cooperative assembly reaction **e**, An overview of one possible pathway. **f**, The full set of intermediates along the prescribed assembly pathways. Species that are structurally congruent are linked by gray boxes. The numbers of assembled strands (excluding the initiator) for each row are displayed.

initially keep key sequence domains, called toeholds,[24] inaccessible; each hairpin can then be opened by a specific input strand, allowing the newly accessible toeholds to participate in downstream reactions. Cooperative hybridization complexes each bind to two initiators and are used in ring formation reactions. Thus, the reactants as a group are metastable, so that no reaction appreciably proceeds in the absence of the initiator, but in the presence of the initiator the assembly reaction happens autonomously and follows prescribed kinetic assembly pathways without the need for external intervention. The final product is a three-dimensional tetrahedron with edges 18 nm long, each containing five turns of DNA (Fig. 3.1b).

The assembly of the tetrahedron represents the execution of a prescribed molecular program. The program is depicted as a reaction graph using a nodal abstraction that concisely describes the kinetic pathways by which DNA reactants self-assemble and disassemble.[28] The reaction graph emphasizes the functional relationship between reactant complexes rather than the detailed structures of each reactant (Fig. 3.1c). Each molecular species is represented by a node depicted as a black ring containing triangles representing input ports, and circles representing output ports. The initial state of a port is either 'accessible' (open symbol) or 'inaccessible' (solid symbol). The ports are functionally connected through an internal logic that toggles their states during the execution of the program, as described below. A developmental molecular program is written as a reaction graph by connecting complementary output and input ports on different nodes via either assembly operations (solid arrow) or disassembly operations (dashed arrow).

The reaction graph precisely defines which reactions must occur in series and which reactions may occur in parallel. For the tetrahedron, the reaction can only start with the assembly of **I** to **A1**, because this is the only assembly reaction where both participating ports (the output port on **I** and the input port on **A1**) are initially accessible. The **B1** and **A2** nodes cannot assemble with **A1** until **A1** has been opened (via assembly with **I**) and its output ports have become accessible, but these two assembly reactions are not dependent on each other because they are on separate branches downstream of the **A1** assembly. This is in contrast with traditional thermodynamic assemblies, where all parts of each molecule are initially accessible, with no explicit control of assembly order (Fig. 3.1c, right).

The ports each correspond to a physical region on a strand or complex that has a single function during the assembly process. For hairpins, the single input region is represented by an orange triangle, and the two output regions are colored according to their position: the blue ports overlap the loops, while the green ports overlap the tails. The nodal abstraction is capable of representing the state of the system at various points in execution: in a hairpin assembly reaction (Fig. 3.1d, top) the hairpin input port and the corresponding initiator output port are both initially open, indicating that a reaction is possible. After assembly, a line is drawn between these ports indicating that they are now bound, and, importantly, the two output ports on the hairpin change state from closed to open. This corresponds to the molecular structural change, where the 1* toehold on the single-stranded initiator hybridizes with the exposed 1 toehold on the hairpin, beginning a strand displacement that opens the hairpin and exposes the two initially hidden toeholds 2* and 4*. A disassembly reaction[28] (not shown) is represented by a dashed arrow, and represents the displacement of a previous assembly with a new one. This type of reaction is used in the tetrahedron to displace the initiator partway through the assembly, allowing it to catalyze the assembly of further tetrahedra.

Cooperative complexes,[36] rather than hairpins, were used for the ring forming reactions because they were observed to reduce the formation of aggregates and multimeric products, which are the result of intermolecular interactions outcompeting the desired intramolecular ones. Cooperative hybridization complexes contain two toeholds and are assembled into the growing complex only in the presence of both initiator domains (Fig. 3.1d, bottom). This is because the singly-bound intermediate is short-lived due to the fact that neither initiator domain can fully displace the protector strand by itself, leading to quick disassembly if the second initiator is not immediately available.[36] In molecular terms, the depicted cooperative complex has two toeholds, 7 and 10. Upon binding to the exposed 7* toehold in the blue initiator, the protector strand is partially displaced. This initial reaction is by itself reversible; in the absence of the green initiator, the blue initiator will rapidly disassemble. However, if the green initiator with toehold 10 is also present, the 10 and 10* toeholds may then bind, allowing the protector strand to be completely and irreversibly displaced. In the tetrahedron, all cooperative assembly events are intended to occur intramolecularly, involving two initiator domains on the same molecule, and thus each such assembly causes the formation of a ring.

Figs. 3.1e and 3.2 depict an example execution trajectory for the molecular program. The top panel shows the molecular structures and the bottom panel shows the corresponding state of the molecular program using the nodal abstraction. The assembly begins with the sequential opening of the **A1**, **A2**, and **A3** hairpins causing self-assembly of a three-arm junction[28] corresponding to the first vertex of the tetrahedron (reactions 1–3). The initiator **I** is released upon completion of the junction (reaction 3), allowing it to catalyze the assembly of further tetrahedra. Next, each arm of this junction then extends to incorporate the **B1**, **B2**, and **B3** reactants through another set of hairpin opening reactions (reactions 4–6). Finally, the overhanging portions of the **B** strands are involved in three ring forming reactions using a set of cooperative hybridization complexes[36] incorporating the **C1**, **C2**, and **C3** strands (reactions 7–9).

Because the assembly was designed to proceed in a branched fashion, the assembly of the three independent branches is not synchronized. Thus there are multiple potential paths leading to the desired product, involving a total of 24 on-pathway intermediates (Fig. 3.1f). We identify each intermediate of the tetrahedron formation with a three-letter code where each letter denotes the progress of one of the three branches of assembly. For example, in the intermediate ***ABA***, **A1** has been incorporated in the first branch, **A2** and **B2** are in the second branch, and **A3** is in the third branch (note again that **A2** incorporation is a prerequisite of **B2** incorporation). Letter ***X*** is used to denote the lack of assembly of the **A** strand along the corresponding branch; if the third letter is ***X***, the initiator **I** is still attached (for example, ***AAX*** contains **A1**, **A2**, and **I**). See Fig. 3.3 for a detailed description of the codes. Note that complexes that are related by cyclic permutations of their abbreviations have congruent secondary structures; these are linked by gray boxes in the figure.

Methods

*Design***.** Motif design was performed by hand using principles described previously[28] and as follows. The secondary structure design of the tetrahedron (Fig. 2.3a) involved considering the

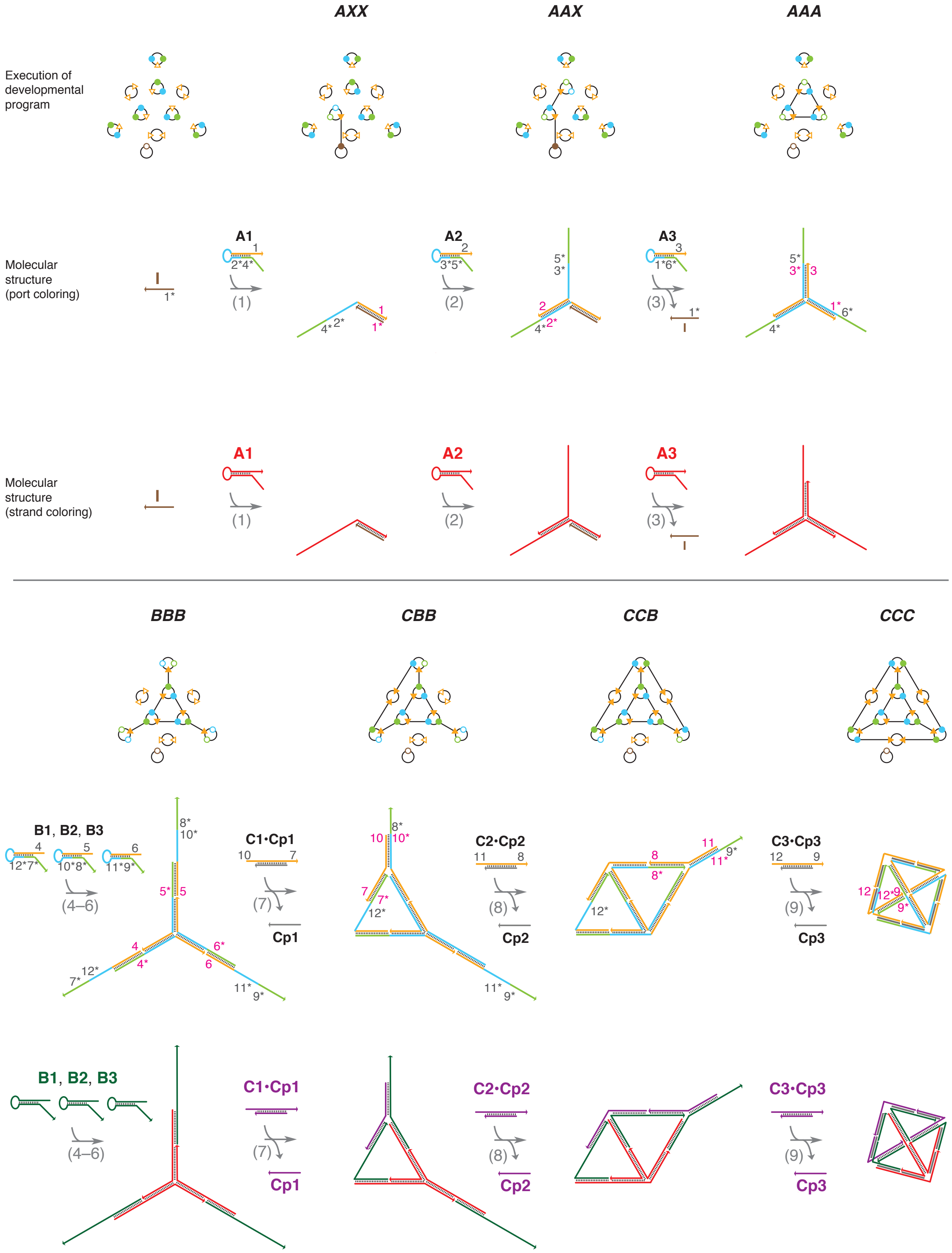


Figure 3.2: Execution of the developmental molecular program along one possible assembly pathway. Only active toehold segments are shown in this figure, with newly-hybridized toeholds shown in pink; see Fig. 2.3a for a schematic showing all segments. Top, nodal abstraction; middle and bottom, corresponding molecular structures.

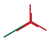
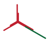
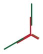
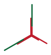
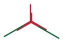
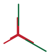
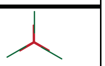
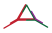
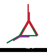
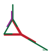
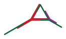
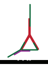
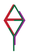
Name	A1	B1	C1	A2	B2	C2	A3	B3	C3	I	Structure
AXX	+									+	
AAX	+			+						+	
BXX	+	+								+	
AAA	+			+			+				
ABX	+			+	+					+	
BAX	+	+		+						+	
BAA	+	+		+			+				
ABA	+			+	+		+				
AAB	+			+			+	+			
BBX	+	+		+	+					+	
BBA	+	+		+	+		+				
BAB	+	+		+			+	+			
ABB	+			+	+		+	+			
CBX	+	+	+	+	+					+	
BBB	+	+		+	+		+	+			
CBA	+	+	+	+	+		+				
ACB	+			+	+	+	+	+			
BAC	+	+		+			+	+	+		
CBB	+	+	+	+	+		+	+			
BCB	+	+		+	+	+	+	+			
BBC	+	+		+	+		+	+	+		
CCB	+	+	+	+	+	+	+	+			
CBC	+	+	+	+	+		+	+	+		
BCC	+	+		+	+	+	+	+	+		
CCC	+	+	+	+	+	+	+	+	+		

Figure 3.3: Composition of the intermediates. The strands contained in each intermediate species are marked.

placement and dimensioning of the sequence domains within the tetrahedron. Because we desired each edge of the tetrahedron to have the same length, we inserted the spacer domains a–f into the tetrahedron's outer edges to lengthen them. We also added one-nucleotide junction spacers s to increase the flexibility of the junctions. We designed the tetrahedron to have an integral number of turns for each edge with the minor groove facing inwards at each junction. Two-nucleotide clamping domains x–z were placed at the outer edges of the hairpin stems to discourage leakage out of the metastable hairpin state. We avoided placing large complementary segments in the loops, which has been observed to cause leakage due to kissing-loop interactions.

Sequence design was performed with Multisubjective (version 1.0.2), a program that identifies and eliminates spurious hybridizations in the full hairpin system, in association with Domain Design (DD) (version 0.2), which uses simple heuristics to evaluate the acceptability of candidate sequence domains.[37] A local copy of NUPACK (version 3.0) was used by Multisubjective to generate base-pairing probabilities for further analysis.[38,39]

Strand synthesis. DNA strands were synthesized and purified by Integrated DNA Technologies (IDT), including strands containing phosphates and fluorophores. For the fluorescent **C** strands used in the fluorescence quenching assay only, we ordered the strands in two halves, ligated them using 27 U/μL T4 DNA ligase at a DNA concentration of 27 μM at 16°C for 2 hr, and purified the product by denaturing polyacrylamide gel electrophoresis. In all cases, we quantitated the concentrations of DNA stock solutions by measuring the ultraviolet light absorption at 260 nm with the micro-volume pedestal of a NanoDrop

[37] D. Y. Zhang. Towards domain-based sequence design for DNA strand displacement reactions. *Lect. Notes Comp. Sci.*, 6518:162–75, 2011. doi:10.1007/978-3-642-18305-8_15

[38] R. M. Dirks, J. S. Bois, J. M. Schaeffer, E. Winfree, and N. A. Pierce. Thermodynamic analysis of interacting nucleic acid strands. *SIAM Rev.*, 49:65–88, 2007. doi:10.1137/060651100

[39] J. N. Zadeh, C. D. Steenberg, J. S. Bois, B. R. Wolfe, M. B. Pierce, A. R. Khan, R. M. Dirks, and N. A. Pierce. NUPACK: Analysis and design of nucleic acid systems. *J. Comput. Chem.*, 32:170–3, 2011. doi:10.1002/jcc.21596

2000c spectrophotometer, taking the average of measurements of three samples for each strand, and using extinction coefficients provided by IDT to calculate the concentration of each strand.

Sample preparation. The hairpin strands were separately heated to 95°C for 5 min and allowed to cool to room temperature over about 20 min. Cooperative hybridization complexes were annealed in the same way, except that 1.5 equivalents of the **Cp** strands were combined with one equivalent of the **C** strand before annealing. The desired reactants were then combined to a final concentration of 100 nM (except as stated in Figs. 3.4, 3.6, 3.9, 3.10, and 3.15b) of each reactant except **A1**, for which 0.9 equivalents were used instead in order to obviate the effects of inaccuracies in stoichiometry and assist in calculation of the yield. The assembly reactions were performed at room temperature over about 20 hr in TAE/Mg^{2+} buffer containing 40 mM Tris base, 20 mM acetic acid, 12.7 mM EDTA, and 12.5 mM $MgCl_2$.

For certain assays (Figs. 3.4, 3.5, 3.7, 3.8, and 3.9), an **A1** strand incorporating a fluorescent label was used instead of the regular **A1** strand. For the fluorescence-quenching assays (Figs. 3.10 and 3.14), the **C** strands each had a different fluorophore on one end and an optional quencher on the other, as described in the figure caption. For the ***CCC*** samples in the atomic force microscopy studies (Fig. 3.11a), we used **C** strands that contained phosphates on their 5′ ends; this was not expected to affect the assembly. For the streptavidin labeling experiment (Fig. 3.11b), we used an **A2** strand containing a 5′ biotin.

For the atomic force microscope studies, the samples were purified by ultracentrifugation in a 15–45% glycerol gradient in 1× TAE/Mg^{2+} for 3 hr at 50,000 rpm. Fractions of 100 μL were manually taken from the centrifuge tube, and the fractions containing the desired product were identified by native gel electrophoresis. The glycerol was not removed after the purifications, meaning that subsequent manipulations were performed in buffer containing about 20% glycerol.

Gel electrophoresis. The gels shown are all precast 6% native polyacrylamide gels run at 100 V for 55 min at 21°C using 1× TBE running buffer, except for Figs. 3.5 and 3.7b which are 1% native agarose gels run at 100V for 90 min or 75 min, respectively, on ice. In general, in each lane one equivalent represents 0.1 pmol of material for the polyacrylamide gels, or 0.2 pmol for the agarose gels.

The gels were imaged with a Typhoon FLA 9000 gel scanner. The gels in Figs. 3.4, 3.10 (right side inset only), 3.13, 3.15, and 3.17 were stained in SYBR Gold for about 20 min before imaging; all other gels were unstained. For Fig. 3.4, we imaged the gel unstained using the fluorescent label, and then stained it and re-imaged the same gel. The two images were then manually overlaid.

We quantitated the yield by dividing the fluorescence intensity of the desired product by the intensity of the entire lane in the fluorescently labeled channel (shown as red) in Fig. 3.4, using ImageQuant TL. We used automatic band detection for the bands representing the desired product, and the reactants band if it was present. We then defined multimers as the lane area with lower mobility than the desired product band, and intermediates as the lane area between the product and reactants band. For background subtraction we used the rolling ball method with a radius of 500.

Atomic force microscopy. The samples in Fig. 3.11a were imaged directly after glycerol purification. For the streptavidin labeling experiment (Fig. 3.11b), the appropriate digestion buffer was added after assembly to a final concentration of 1×, and the structures were digested with one of the two restriction endonucleases BstZ17I or ScaI-HF at a concentration of 1 U/μL at 37°C for 2 hr. The structures were then incubated with a 10× solution of streptavidin for 30 min, and then glycerol purified.

For imaging, we added to a freshly cleaved mica surface 30 μL of filtered 5× TAE/Mg^{2+} and then 30 μL of a 10 mM solution of $NiCl_2$ to increase the strength of the DNA–mica binding. After 5 min, we added 10–30 μL of the desired fraction from the glycerol purification. AFM images were obtained using a Multimode 8 scanning probe microscope with a Digital Instruments Nanoscope V controller. Images were collected in aqueous phase using tapping mode, using the short and thin cantilevers in the SNL-10 silicon nitride cantilever chip. Each of the two panels for Fig. 3.11 contains a set of images that were generated using the same tip.

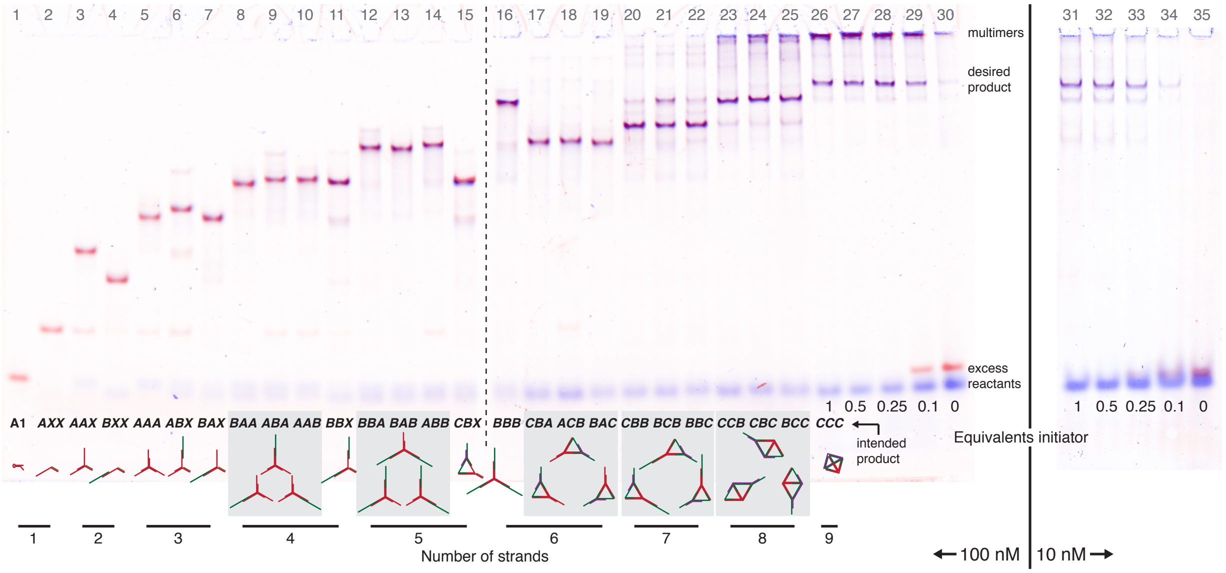


Figure 3.4: Characterization of the tetrahedron assembly pathway. This is a gel electrophoresis mobility shift assay where lanes 1–25 show all on-pathway intermediates of the tetrahedron assembly, formed by mixing different subsets of reactants with the initiator. Gray boxes mark groups of intermediates that are structurally congruent and expected to have the same mobility, and the structures of the intended products are shown, as in Fig. 3.1f. Lanes 26–35 show the analysis of catalytic turnover at two concentrations, showing reactions containing all nine reactants but varying concentrations of initiator **I**. These are 6% native polyacrylamide gels of assembly reactions containing 1 equivalent of all reactants at the specified concentration, except **A1** for which we used 0.9 equivalents of a FAM fluorophore-labeled hairpin to observe incorporation yields. The initiator was included at 1 equivalent unless otherwise specified in the figure. The assembly reactions were conducted at room temperature in TAE/Mg^{2+} buffer containing 12.5 mM $MgCl_2$ over 19 hr. The dotted line separates two gel slabs that were run simultaneously, and the solid line separates gels that were run at different times. The intensity of the FAM fluorescent label is shown in red, and SYBR Gold staining intensity is shown in blue. See Fig. 3.5 for an agarose gel of these same samples, in which the higher molecular weight side products are well-resolved.

Results

Formation and yield assays. We studied the formation of the tetrahedron using a gel electrophoresis mobility shift assay (Fig. 3.4), where each intermediate was individually synthesized by mixing the initiator with different subsets of the nine reactants of the tetrahedron assembly. We were able to observe the formation of each of the 24 possible intermediates as a distinct band in a native polyacrylamide gel (lanes 2–25). Intermediates with greater numbers of incorporated strands migrated more slowly, with the exception of the ring formation transitions, which tended to have unpredictable effects (*e.g.*, ***BBX→CBX*** and ***BBA→CBA*** showed little mobility change, and ***BBB→CBB*** showed increased mobility). This is likely because the first ring formation reduces the angle between two of the arms from its natural angle to ≈60°, making the complex more compact. Groups of intermediates with

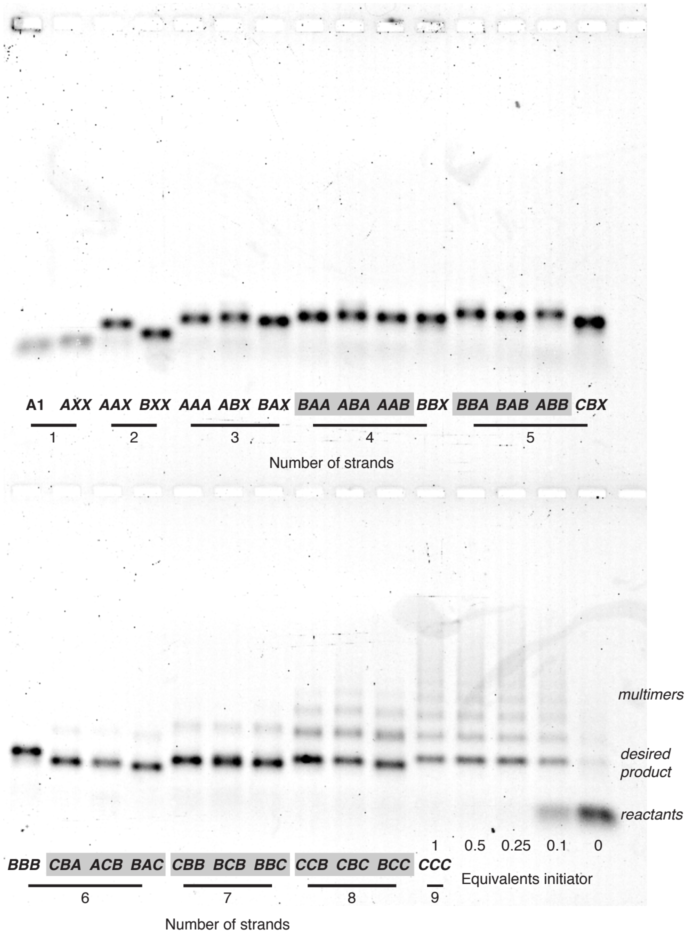


Figure 3.5: Characterization of higher molecular weight side products. While the higher molecular weight products do not exit the well in the polyacrylamide gel in Fig. 3.4, in this agarose gel they are well resolved as a series of discrete bands that might possibly correspond to well-defined multimers of the desired tetrahedron product. This gel uses samples from the same aliquots used in the formation assay in gel in Fig. 3.4. This is a 1% native agarose gel of a 100 nM assembly reaction.

congruent structures, linked by the gray boxes in the figure, were observed to have nearly identical mobility, consistent with our expectations. The band identified as the assembled tetrahedron (lane 26) had a mobility distinct from any of the intermediates. The lower-mobility bands above the tetrahedron band are hypothesized to be multimers of the tetrahedron structure, formed when assembly of the cooperative complexes cause the intermolecular joining of two different growing tetrahedra rather than the desired intramolecular ring forming reactions (Fig. 3.5).

When all reactants except **I** were incubated together (Fig. 3.4, lane 30), only a small amount of unintended "leakage" products were observed: 3.5% of the signal was in the product band and 80.% was in the reactants band. When at least 0.25 equivalents of **I** are included (lanes 26–28), the reaction proceeded to completion (that is, all **A1** hairpins were consumed), indicating a catalytic turnover of at least 4. Typical yields of the tetrahedron varied between 20–40%, depending on concentration (Fig. 3.6). At a higher reactant concentration (100 nM, Fig. 3.4 lanes 1–30), the dominant side products were the higher weight products, while at a lower concentration (10 nM, lanes 31–35), the dominant side products were lower molecular weight intermediates. Varying the toehold lengths of the cooperative complexes was not observed to qualitatively improve the reaction yield (Fig. 3.7).

We observed that the reaction yield was primarily bottlenecked by the ring forming reactions. Based on analysis of Fig. 3.4, the yield of ***BBB***, the largest intermediate with no rings, was 83% for six hairpin incorporation events. However, the average yield of the seven intermediates containing one **C** strand, and thus one formed ring, dropped to 71%. With two **C** strands, the yield was reduced to 50.%, and the yield of the full tetrahedron was only 19%. This drop in yield is likely the result of undesired intermolecular interactions outcompeting the desired intramolecular ring formation during the assembly of the cooperative complexes. This may be exacerbated by steric strain introduced in the ring formations. Although single-nucleotide spacers and nicks at each junction are meant to relieve this strain, it is not clear if this was sufficient, and the use of longer single-stranded spacers at the vertices might further improve the yield.

Gel electrophoresis studies showed that the assembly is temperature-robust, working isothermally at temperatures in roughly the range 16–41°C (Fig. 3.8). At room temperature at 100 nM, the assembly reaction was observed to be complete after 9 hr (Fig. 3.9).

Ring formation assay. While the unique mobility of the ***CCC*** band indicates that all nine strands are incorporated into a single complex, additional assays are required to demonstrate ring formation and distinguish it from, for example, a large floppy three-arm junction. To demonstrate the ring formation we conducted a fluorescence-quenching assay (Fig. 3.10), using **C** strands functionalized with a fluorophore

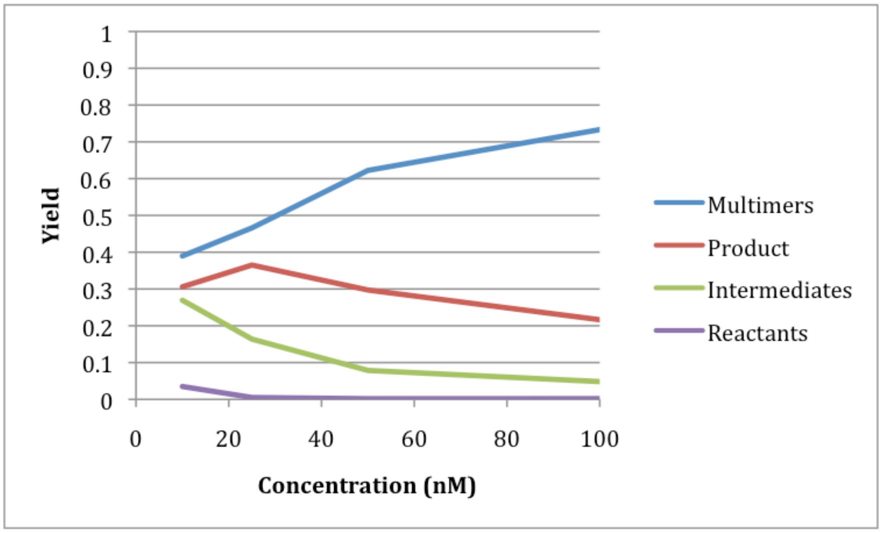


Figure 3.6: Concentration robustness of the assembly reaction. This graph shows the yield of the reaction at different concentrations of reactants, as calculated through quantitation of gel electrophoresis staining intensity. As concentration increases, the amount of intermediates decreases but the amount of higher molecular weight side products increases. Yield was calculated as the percentage of signal in the appropriate region of the lane divided by the total signal in the entire lane, using the same parameters described in the Methods section. Each value is the average of three gel lanes each containing the same sample.

a

Toehold lengths	**Total base pairs formed**	***CBB* forms?**
8:8	15	Yes
10:6	15	Yes
12:4	15	Yes
16:0	15	Unstable product
10:4	13	Yes
6:6	11	No reaction
12:0	11	No reaction
4:4	7	No reaction

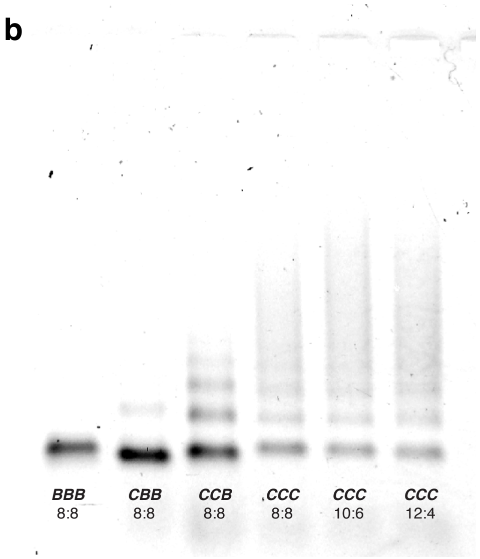


Figure 3.7: The effects of toehold lengths on the assembly of cooperative hybridization complexes. a, The results of gel electrophoresis studies on assembly of the intermediate ***CBB*** using cooperative hybridization complexes with different toehold lengths. The lengths are listed with the 5′ toehold length followed by the 3′ toehold length, separated by a colon (:). The 8:8 complex was the one used in the main body of this paper. In general, the total number of base pairs formed seems to be the main determinant of whether the assembly proceeds, rather than the lengths of the individual toeholds. Note that the junction spacer base becomes unpaired during assembly, lowering the total number of base pairs formed by one. For the failed complexes, the ***BBB*** band was observed, indicating no assembly event involving the cooperative complex had occurred. For the 16:0 complex only, the singly bound intermediate was stable enough to form a streak on the gel above the ***BBB*** band. We made these complexes by extending or truncating the protector strands as appropriate. **b**, A gel showing the full tetrahedron ***CCC*** where all three cooperative complexes have the specified toehold lengths. No qualitative difference in yield is observed. This is a 1% native agarose gel of a 100 nM assembly reaction.

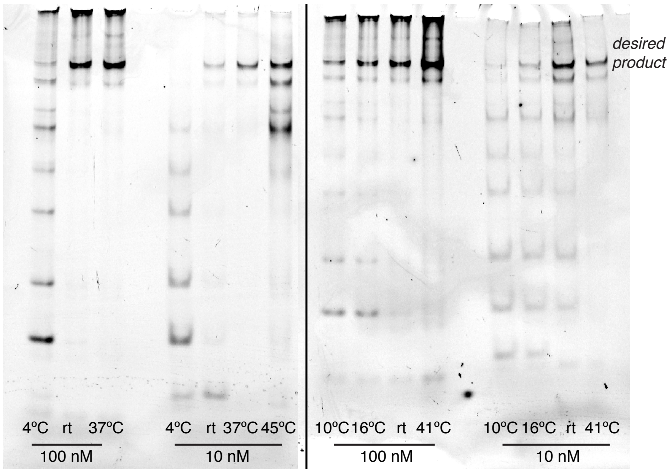


Figure 3.8: Temperature robustness of the assembly reaction. These gels show the results of assembly at different temperatures, at two concentrations after 18 hr. The reaction appears to go to completion to the same extent as the room temperature samples in roughly the range of 16–41°C. At lower or higher temperatures, the reaction does not go to completion. These are 6% native polyacrylamide gels. The solid line separates two gels that were run at different times.

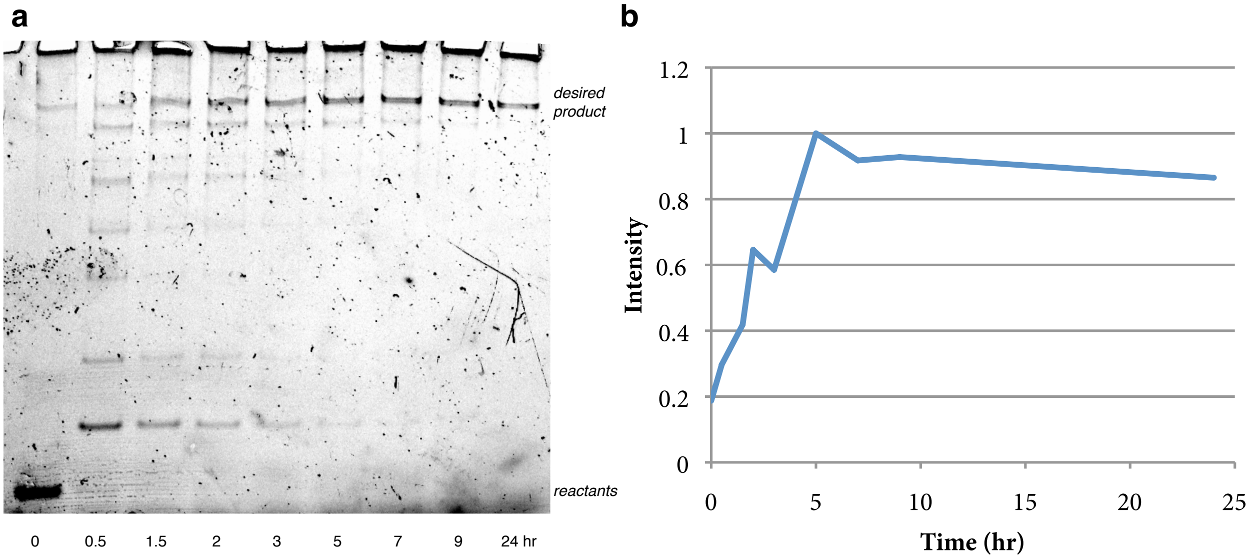


Figure 3.9: Progress of the reaction over time. a, A gel showing a timecourse of the assembly. These reactions all had the nine reactants mixed 24 hr before gel loading, and the initiator was added at the specified time before gel loading. This is a 6% native polyacrylamide gel of a 100 nM assembly reaction. **b**, A graph showing the relative intensities of the desired product bands in the gel.

and optionally a quencher. The formation of each ring caused the fluorophore from one **C** molecule to become colocalized with the quencher on the neighboring strand, quenching the fluorescent signal. For example, lane 1 contained a ***CBB*** complex where a red fluorophore was attached to the **C1** strand; as expected, a red fluorescent band was visible on the gel. Lane 2 contained a ***CCB*** complex where red and

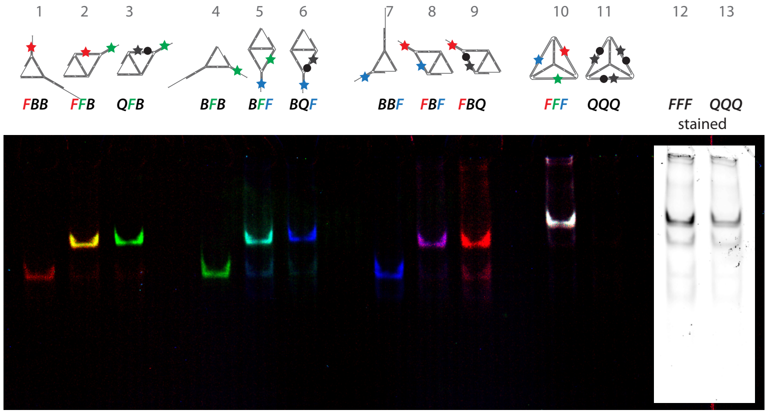


Figure 3.10: Characterization of ring forming reactions. In this fluorescence quenching assays, each of the three **C** strands was functionalized with a different fluorophore at its 5′ end (**C1**–TYE 665, red; **C2**–TAMRA, green; **C3**–FAM, blue); ***F*** represents a **C** strand with a fluorophore only, while ***Q*** represents a strand with a fluorophore expected to be quenched by a quencher on the 3′ end of a neighboring **C** strand. Lanes 1–3 show the structures ***FBB*** (***CBB*** with a red fluorophore on the **C1** strand), ***FFB*** (***CCB*** with red and green fluorophores on the **C1** and **C2** strands respectively), and ***QFB*** (***CCB*** with the same two fluorophores, plus a 3′-quencher on the **C2** strand that quenches the red 5′-fluorophore on the **C1** strand). Lanes 4–6 and 7–9 show the other two structural permutations. Lane 10 shows ***FFF*** (the full tetrahedron with all fluorophores), and lane 11 shows ***QQQ*** (the full tetrahedron with all fluorophores and all quenchers). Lanes 12–13 are the same lanes in the same gel as lanes 10–11, but after staining with SYBR Gold. This is a 6% native polyacrylamide gel of a 10 nM assembly reaction with 1 equivalent of initiator, conducted at room temperature in TAE/Mg^{2+} buffer containing 12.5 mM Mg^{2+} over 22 hr.

green fluorophores were respectively attached to the **C1** and **C2** strands; both red and green fluorescence were detected in the target band, as expected. Lane 3 showed another ***CCB*** complex which additionally had a quencher attached to the **C2** strand; here, green but not red fluorescence signal was detected, indicating that proper ring formation had resulted in the colocalization of the quencher on **C2** with the red fluorophore on **C1**, thus removing the red fluorescence from the product band. In the full tetrahedron, all three fluorophores were fully visible in the absence of quenchers (lane 10) but fully quenched when quenchers were added (lane 11), indicating that all three ring formations were simultaneously successful, and that the tetrahedron's overall structure formed as designed.

Microscopy studies. We further confirmed the correct formation of the geometric structures of the tetrahedron and its key intermediates using atomic force microscopy (AFM). ***BBB*** appeared as a three-arm junction, ***CBB*** as a single triangle, ***CCB*** as a double triangle, and ***CCC*** as a flattened tetrahedron, all consistent with their designed shapes (Fig. 3.11a). Additionally, digestion at an intentionally engineered endonuclease restriction site on a specific edge converted a full tetrahedron back to a double triangle

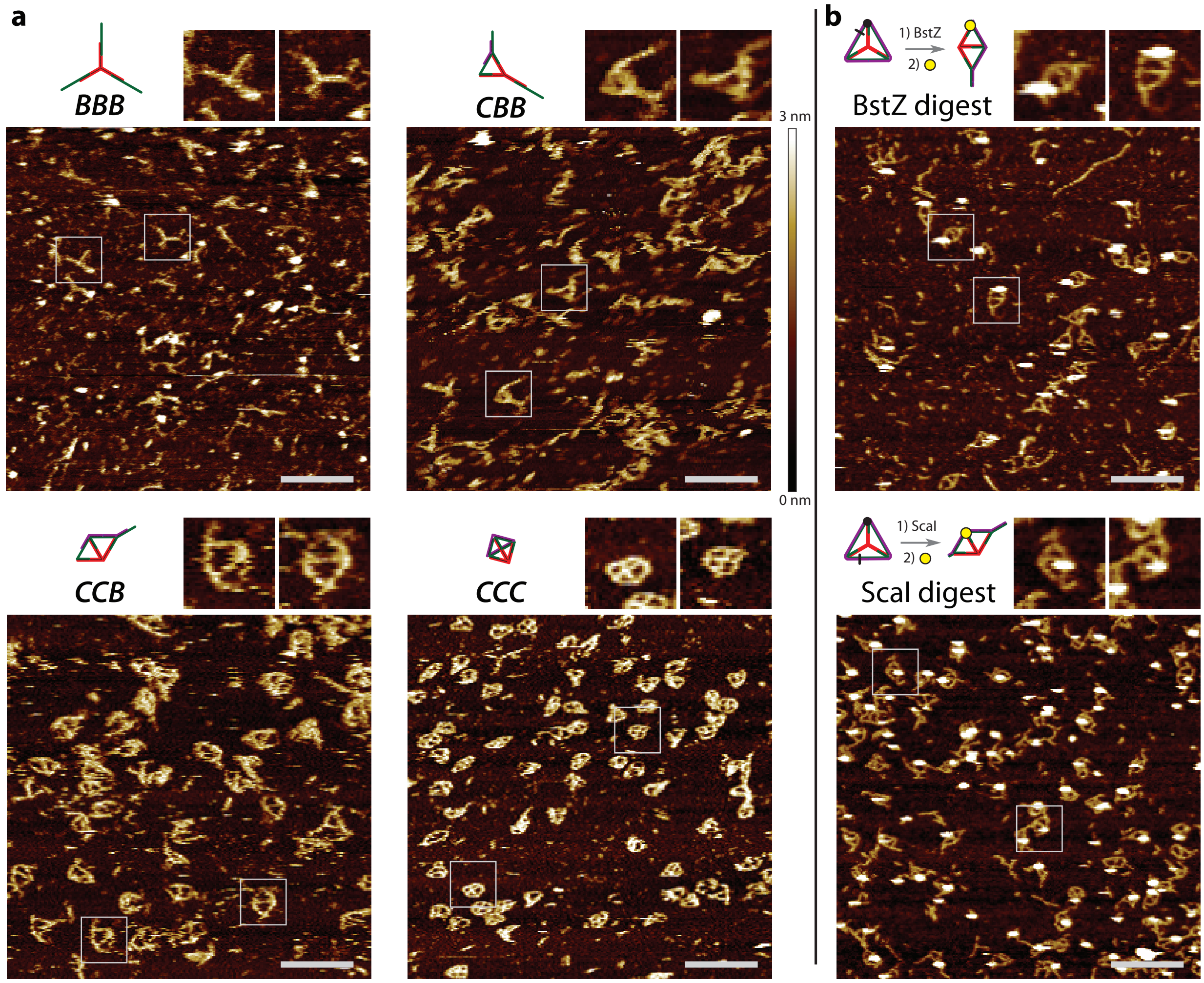


Figure 3.11: Atomic force microscopy images of tetrahedron self-assembly. a, Images of purified samples of the intermediates ***BBB***, ***CBB***, ***CCB***, and the full tetrahedron ***CCC***, which respectively appeared as a three-arm junction, a single triangle, a double triangle, and a three- or four-lobed structure corresponding to a flattened tetrahedron, each consistent with our design. In each image, the double-stranded edges of each wireframe structure are clearly resolved. **b**, Images of the full tetrahedron cut by restriction endonucleases followed by incubation with streptavidin. Digestion with either of two endonucleases targeted to different edges restored the double-triangle structures, with the streptavidin (appearing as small white circular features in the AFM image) appearing at the expected biotin-modified vertices. In the schematic diagrams, the black bar intersecting tetrahedron edge represents the designed restriction site, the black dot is the 5′-biotin, and the yellow circle is the streptavidin. The scale bars of the larger images are 100 nm long; zoomed-in images are 62.5 nm in width, at double the scale of the larger images. These samples were assembled at a concentration of 100 nM and then purified by glycerol gradient ultracentrifugation; for panel **b**, the restriction and streptavidin binding were performed before purification.

pattern under AFM, as expected. The double triangle structure contained two non-congruent types of vertices, which were distinguished by attaching a streptavidin molecule to one of them. Digestion with either of two endonucleases targeted to two different edges yielded images of structures with the streptavidin on the expected vertex, further confirming the correct formation of the full tetrahedron (Fig. 3.11b).

Alternative structures

Bridge regions for ring formation. In the course of developing the tetrahedron presented in the previous section, which we have designated TET9, a number of alternative designs were evaluated. Two of these, TET6 and TET8, used "bridge regions" for the ring formation reactions instead of cooperative hybridization complexes. The bridge binding interaction occurs between two complementary regions that are independently revealed on different hairpins; these are intended to hybridize to each other once both are present within the same complex. In the TET6 and TET8 systems, the bridge interactions are each between regions on a pair of one **B** and one **C** hairpin (Fig. 3.12). TET6 and TET8 differ in the length of the region involved in the bridge interaction: for TET6, 12 base pairs are formed, while for TET8, 20 base pairs are formed.

A partial formation assay on TET6 suggested that all nine strands did assemble into a single complex (Fig. 3.13), but a fluorescence quenching assay failed to indicate that the rings were forming to any significant degree (Fig. 3.14a). However, the quenching experiment was successful for a single ring closing event on a variant version of TET6 where one bridge region was extended to form 20 instead of 12 base pairs (Fig. 3.14b). This suggested that the shorter length was not enough to drive the ring formation to completion, and TET8 was designed to incorporate 20 nt bridge regions throughout.

However, TET8 did not form properly (Fig. 3.15). We believe this was because of an effect we call segment proliferation: because segments must be complementary both across the closed hairpin stems and within the final structure, chains of complementarity dependence can cause segments to appear multiple times in the system. In this case, the bridge segments had proliferated so that they appeared in the tails and stems of the three **B** hairpins, causing them to form an off-pathway trimer when open, leading to the formation of aggregates. Further investigation indicated that there was no "Goldilocks" length that was neither too short to form the rings, nor too long to avoid off-pathway interactions leading to aggregation. This result led to the design of TET9 using cooperative hybridization complexes instead of bridge interactions for the ring formations.

In summary, bridge interactions are capable of forming rings if the interaction forms enough base pairs, but their use was found to be incompatible with the specific tetrahedron architecture used here due to issues with segment proliferation. Nevertheless, bridge interactions are still an option for use in future assemblies with different architectures.

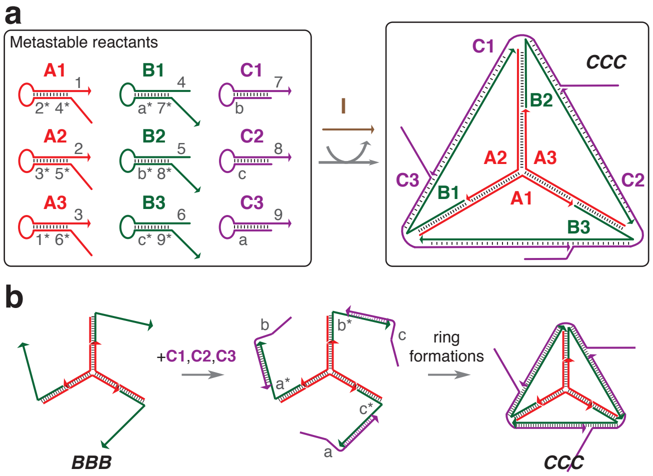


Figure 3.12: Overview of the TET6 and TET8 systems. a, Nine hairpins self-assemble in the presence of a catalytic initiator to form the tetrahedron. Portions of the **C** strands become single-stranded attachments to the final structure. Only toeholds (numbers) and bridge regions (letters) are marked in this figure. **b**, Schematic showing the bridge interactions used for the ring forming steps. The assembly exposes complementary single-stranded regions called “bridge regions” on three corresponding sets of **B** and **C** hairpins. The hybridization of each set of bridge regions is intended to drive the ring formations to occur. TET6 had 12 nt bridge regions, while TET8 had 20 nt bridge regions. Note that the ring formations do not occur in a synchronized manner, as shown here, but each may occur at any time after the corresponding set of a **B** and a **C** hairpin has assembled.

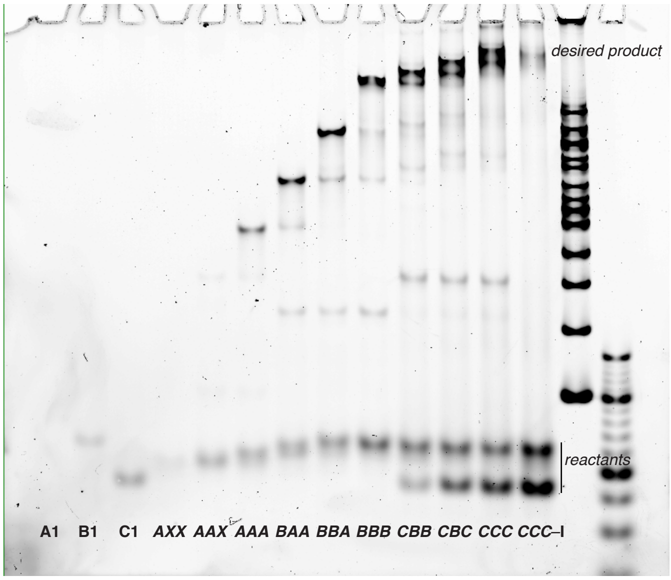


Figure 3.13: Partial formation assay of TET6. The desired product forms. The ***CCC*–I** lane contains all nine reactants without the initiator. This is a 6% native polyacrylamide gel of a 100 nM assembly reaction with 0.5× initiator.

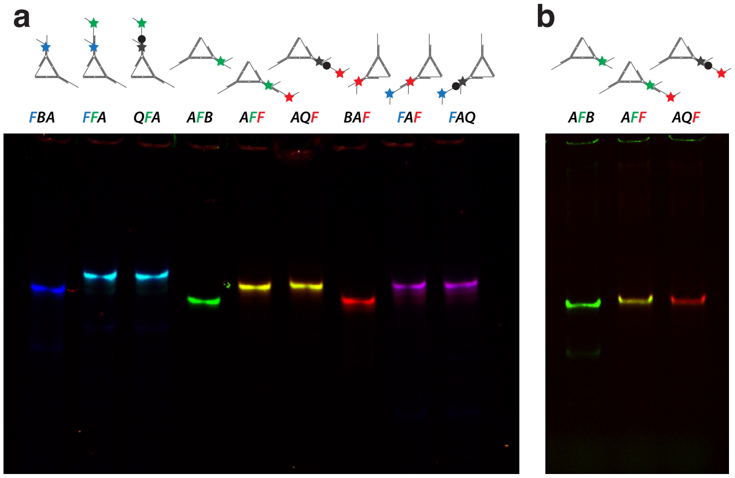


Figure 3.14: Fluorescence quenching assay of TET6. a, This quenching experiment uses the same principles as that in Fig. 3.10, but this gel shows no significant quenching in the lanes for ***QFA***, ***AQF***, and ***FAQ***. This suggests that the rings have not formed in any of these structures, meaning that the full complex is likely to mostly exist as a large three-arm junction corresponding to the center structure in Fig. 3.12b. (The diagrams above the gel represent the desired rather than the experimentally determined structures.) **b**, A quenching experiment using an altered set of TET6 strands that had a bridge region with a length of 20 nt rather than 12 nt for one of the bridge interactions. The desired quenching in the ***AQF*** lane is observed, indicating that a longer bridge region length is necessary to drive the desired interaction to completion. These are 6% native polyacrylamide gels of 100 nM assembly reactions with 0.5× initiator.

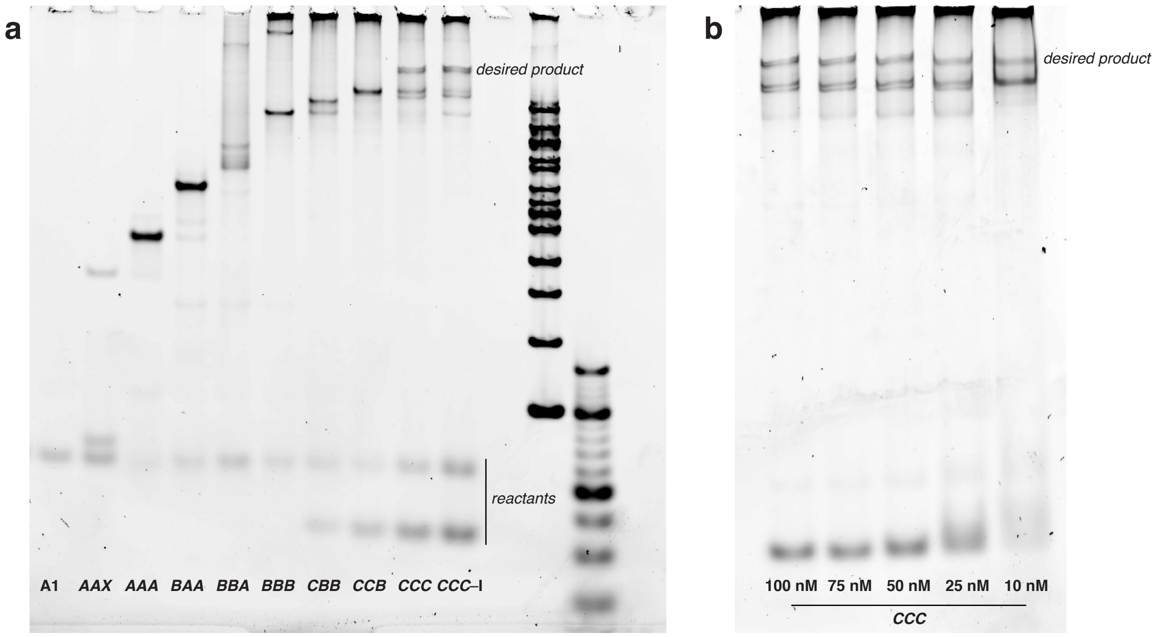


Figure 3.15: Partial formation assay of TET8. a, Undesired higher molecular weight products are seen in the ***BBA*** lane and dominate in the ***BBB*** lane and subsequent lanes. These side products are thought be aggregates resulting from undesired interactions involving open **B** hairpins; these undesired interactions are believed to be the result of segment proliferation. This is a 6% native polyacrylamide gel of a 100 nM assembly reaction with 1× initiator. **b**, A gel showing the full tetrahedron at different assembly concentrations. The aggregate remains the dominant product at concentrations as low as 10 nM. This is a 6% native polyacrylamide gel.

Three-output hairpins for ring formation. TET10 is a structure designed so that the ring-closing interactions occur within one edge rather than overlapping two edges of the tetrahedron, leading to ring formation reactions that are modular with respect to the edges (Fig. 3.16a,b). This necessitates a novel three-output hairpin motif for the three **A** hairpins, each having an extra initially-accessible output region on its tail. Because these open output regions assemble to cooperative hybridization complexes, no assembly can occur until that cooperative complex's other initiator becomes available (Fig. 3.16c). A partial formation gel suggests that this design works qualitatively as well as TET9, although some of the intermediates seem to be of lower quality (Fig. 3.17). Nevertheless, this indicates that the three-output motif is a viable option for future assemblies.

The nodal reaction graphs for each of the tetrahedra appear in Fig. 3.18.

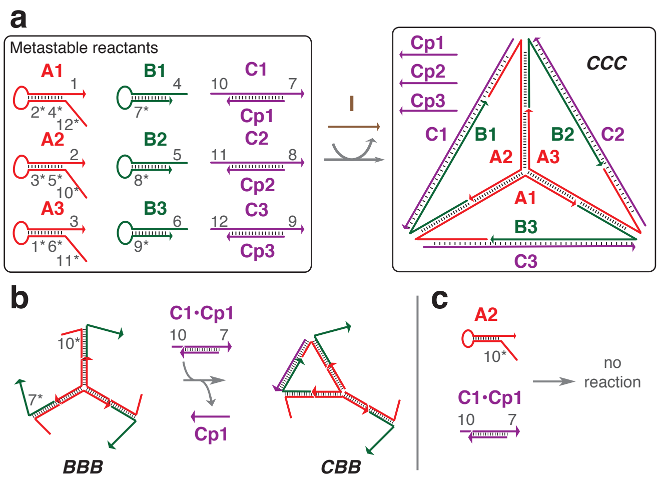


Figure 3.16: Overview of the TET10 system. **a**, Six hairpins and three cooperative hybridization complexes combine in the presence of a catalytic initiator to form the tetrahedron. Relative to TET9, the **A** hairpins are elongated and the **B** hairpins are truncated. This results in the **C** strands being in a different position, with each strand corresponding to one edge of the tetrahedron rather than overlapping two edges each. Only toeholds are marked in this figure. **b**, Schematic of a ring formation reaction. **c**, Even though the 10 toehold of the cooperative complex and 10* toehold on the tail of the hairpin are both initially accessible, no assembly occurs because the 7* toehold (in the **B1** hairpin) is not available until it is exposed by assembly events elsewhere.

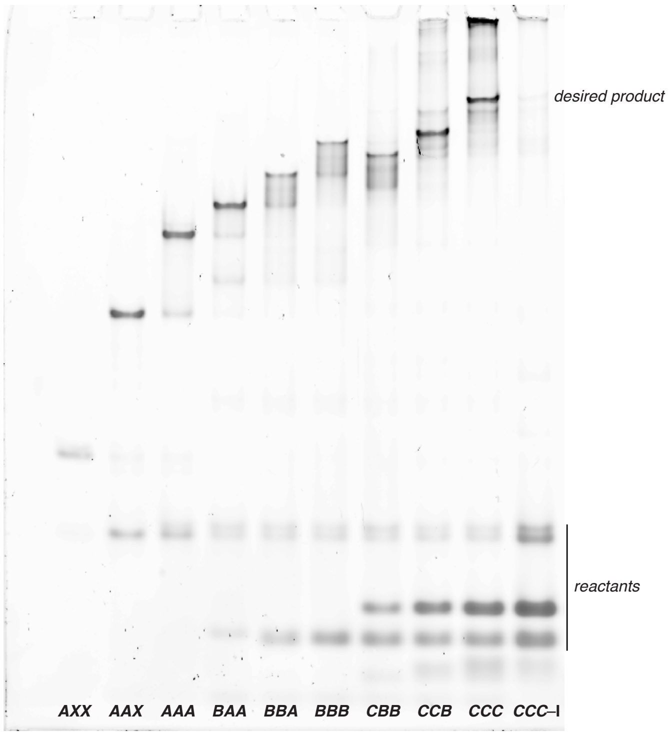
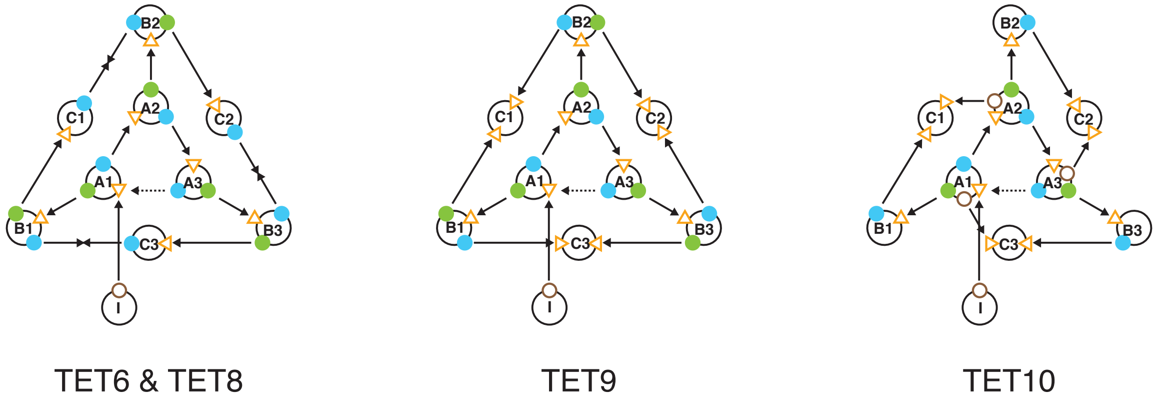


Figure 3.17: Partial formation assay of TET10. The desired product is observed to form with a qualitatively similar yield to TET9, but the quality of the ***BBA***, ***BBB***, and ***CBB*** intermediates appears to be lower than that of TET9. This is a 6% native polyacrylamide gel of a 100 nM assembly reaction with 1× initiator.



TET6 & TET8

TET9

TET10

Figure 3.18: Nodal reaction graphs of the tetrahedra. Each molecular species is represented by a node appearing as a black ring with shapes representing input and output ports. Each port corresponds to a physical region on the strand or complex, containing one or several sequence domains. Triangles represent input ports while circles represent output ports. Solid ports are inaccessible at the beginning of the assembly and become accessible only when the input ports on that reactant becomes assembled, while outlined ports are accessible at the beginning of the assembly. The color of the ports represents their position on the strand. TET6 and TET8 contain a new type of ports that we call bridge ports. Bridge ports are represented by converging double arrows; these assembly events occur only when both output ports are simultaneously accessible. Cooperative hybridization complexes are represented here as having two input ports and no output ports; assembly occurs only when both ports are activated simultaneously. Assembly reactions are represented by solid arrows and disassembly reactions by dashed arrows; these connect the nodes into a reaction graph that specifies the overall assembly order.

Discussion

We have extended the nascent pathway-controlled DNA self-assembly approach,[28] demonstrating the developmental growth of a well-defined three-dimensional wireframe DNA tetrahedron, formed from metastable reactants conditionally in the presence of a single-stranded trigger and through a kinetically programmed pathway. In particular, this is the first catalytic self-assembly design that incorporates ring forming events, a potentially useful topological primitive for the growth of larger assemblies with well-defined geometries. Ring formation is challenging because the desired intramolecular reaction that forms the ring must compete with off-pathway intermolecular reactions that lead to multimers and other aggregation side products. Using cooperative hybridization complexes for the ring forming reactions appears to avoid significant difficulties encountered with the non-cooperative bridge region designs. We believe that even greater improvements in yield could be achieved by designing the geometry of ring forming steps so that they do not require such large conformational changes in the molecule. This would favor the desired intramolecular reaction both thermodynamically and kinetically, because the reaction would induce less strain and would have an increased attempt frequency.

This work represents an advance over previous attempts to control assembly order. One previous approach imposes assembly order by designing different parts of the structure to have distinct melting or formation temperatures, usually by using binding domains of different lengths. The assembly events thus occur in a particular order as the temperature is lowered during the annealing process. This has been used a number of times,[40,41] including with some polyhedra.[42,43] However, such approaches typically require external modulation of the assembly environment in the form of an annealing ramp, and couples the assembly order to bulk environmental conditions such as temperature. Also, while other approaches have recently been demonstrated that work isothermally within narrow ranges of temperatures,[20] no explicit assembly order control is designed in these systems. In contrast, our approach allows direct, molecular-level control of assembly order through complete pathway engineering, is independent of bulk properties like temperature, and operates isothermally over a wide range of temperatures that includes biologically relevant conditions.

Another route for direct isothermal molecular control of assembly order is the algorithmic self-assembly of DNA tiles.[4,10,44] Here, in a seeded growth system, tiles are incorporated sequentially at particular positions, as directed by cooperative binding to neighboring tiles on the growth front. However,

[40] M. T. Kumara, D. Nykypanchuk, and W. B. Sherman. Assembly pathway analysis of DNA nanostructures and the construction of parallel motifs. *Nano Lett.*, 8:1971–7, 2008. doi: 10.1021/nl800907y

[41] T. M. Snyder and D. R. Liu. Ordered multistep synthesis in a single solution directed by DNA templates. *Angew. Chem. Int. Ed.*, 44:7379–82, 2005. doi: 10.1002/anie.200502879

[42] W.M. Shih, J.D. Quispe, and G.F. Joyce. A 1.7-kilobase single-stranded DNA that folds into a nanoscale octahedron. *Nature*, 427:618–21, 2004. doi:10.1038/nature02307

[43] R.P. Goodman, I.A.T. Schaap, C.F. Tardin, C.M. Erben, R.M. Berry, C.F. Schmidt, and A.J. Turberfield. Rapid chiral assembly of rigid DNA building blocks for molecular nanofabrication. *Science*, 310. doi:10.1126/science.1120367

[44] R. D. Barish, R. Schulman, P. W. K. Rothemund, and E. Winfree. An information-bearing seed for nucleating algorithmic self-assembly. *Proc. Natl. Acad. Sci. USA*, 106:6054, 2009. doi:10.1073/pnas.0808736106

the tile based approach is dependent upon the supersaturated nature of the tile species, and is limited to the specific geometry of tile-based lattices with a growth front. More broadly, the tile-based approach involves “static” monomers incorporated by simple binding to the growing structure without explicitly designed internal state changes. In contrast to the “passive” assembly using tiles, our “active” developmental self-assembly uses reconfigurable hairpins as assembling monomers that are capable of conditional configurational change and can implement both assembly and disassembly. An analytical model has found that active methods can in principle be exponentially faster than the corresponding passive methods.[45] Developmental assembly with active components provides a more versatile, expressive molecular programming language for integrating temporal and spatial control and a potentially more efficient method for the active construction of molecular structures.

Kinetically controlled developmental self-assembly mimics the well-orchestrated nature of biological reaction cascades, which operate autonomously and without the need for external intervention to maintain their function. The triggered, isothermal nature of our assembly methodology would allow the formation of complex structures to be integrated with nucleic acid computational circuits, allowing the use of structural changes as an output mode of logical computation. These could furthermore be interfaced with non-nucleic acid inputs such as proteins and small molecules through the use of aptamers,[30] which would allow these systems to interact with the larger chemical world and respond in potentially intricate ways to their molecular environment. This methodology is additionally expected to be more amenable to *in vivo* applications, where thermal annealing is not an available mode of assembly. Thus, synthetic developmental self-assembly promises to open new doors to bridge computation, chemistry, and biology.

[45] D. Woods, H.-L. Chen, S. Goodfriend, N. Dabby, E. Winfree, and P. Yin. Active self-assembly of algorithmic shapes and patterns in polylogarithmic time. In “ITCS 2013: Innovations in Theoretical Computer Science,” Berkeley, CA:Association for Computing Machinery, January 2013. pp. 353–4.

Acknowledgements

The authors thank Casey Grun and Erik Winfree for discussions, and Danielle Pastuszak for help with draft preparation. This work was funded by Office of Naval Research Grant N000141010827, Office of Naval Research Young Investigator Program Award N000141110914, NIH Director's New Innovator Award 1DP2OD007292, NSF CAREER Award CCF1054898, and Wyss Institute for Biologically Inspired Engineering Faculty Startup Fund to P.Y., and NIH P50 HG004071 and NSF-CCF-0832824 (The Molecular Programming Project) to N.P.; J.P.S. acknowledges a Graduate Research Fellowship from NSF and a J. Marshall & Jane H. Booker Graduate Scholarship from the Buttonwood Foundation, and D.Y.Z. is supported by NIH Transition to Independence Award 1K99EB015331.

Chapter 4

Steric, electrostatic, and conformational effects on DNA hairpin opening kinetics

"All models are wrong, but some are useful."

—George E. P. Box

Abstract

Design approaches for developmental self-assembly must take into account potential structural problems that are specific to hairpin structures, which can interfere with the kinetics of the desired strand displacement reactions. Here, we present preliminary data regarding the effect of tail structure, including toehold–tail interactions, on the hairpin opening rate. We observed a timecourse of hairpin opening completion, using the opening of a downstream fluorescent reporter complex for output. We find that in these experiments, under conditions typical for developmental self-assembly, the tail seems to have little discernible effect unless toehold–tail binding involves a run of at least three complementary base pairs.

Introduction

Toehold-mediated strand displacement is fast becoming a vital method for DNA nanotechnology. It is a powerful tool that allows the reconfiguration of molecular structure in a modular, controllable way, underlying many approaches to DNA computing,[1,2] reconfigurable structures,[3–5] walkers,[6,7] and developmental self-assembly.[8] Strand displacement reactions have the practical benefits that they operate isothermally and can be designed to be reversible. These features are difficult or impossible to achieve using traditional methods involving thermal anneals.

The most basic strand displacement reaction (Fig. 1.5) involves three strands: the *base strand* (green) and the *incumbent strand* (blue) are initially associated, with a single-stranded region called a *toehold* (domain 1) on the base strand, next to the double stranded region (domain 2) where the two strands are associated. A third strand, called an *initiator* or *invading strand* (red), is able to bind to the toehold and then displace the incumbent strand, which then dissociates from the complex. The products are the free incumbent strand, and a complex of the invading strand with the base strand.

[1] G. Seelig, D. Soloveichik, D. Y. Zhang, and E. Winfree. Enzyme-free nucleic acid logic circuits. *Science*, 314: 1585–8, 2006. doi:10.1126/science.1132493

[2] L. Qian and E. Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332:1196–1201, 2011. doi:10.1126/science.1200520

[3] B. Yurke, A. J. Turberfield, A. P. Mills Jr., F. C. Simmel, and J. L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406:605–9, 2000. doi:10.1038/35020524

[4] H. Yan, X. Zhang, Z. Shen, and N. C. Seeman. A robust DNA mechanical device controlled by hybridization topology. *Nature*, 415:62–5, 2002. doi:10.1038/415062a

[5] L. Feng, S. H. Park, J. H. Reif, and H. Yan. A two-state DNA lattice switched by DNA nanoactuator. *Angew. Chem. Int. Ed.*, 42:4342–6, 2003. doi:10.1002/anie.200351818

[6] J.-S. Shin and N. A. Pierce. A synthetic DNA walker for molecular transport. *J. Am. Chem. Soc.*, 126:10834–5, 2004. doi:10.1021/ja047543j

[7] W. B. Sherman and N. C. Seeman. A precisely controlled DNA biped walking device. *Nano Lett.*, 4:1203–7, 2004. doi:10.1021/nl049527q

[8] P. Yin, H. M. T. Choi, C. R. Calvert and N. A. Pierce. Programming biomolecular self-assembly pathways. *Nature*, 451: 318–22, 2008. doi:10.1038/nature06451

Strand displacement is a complex multistep process whose exact mechanism and kinetics are poorly understood. Nevertheless, it is important to understand the factors that affect strand displacement kinetics, so that design algorithms can avoid conditions that slow or disfavor the intended reactions. Strand displacement reactions can be broken down into series of three physical processes: the association of the initiator strand to the exposed toehold, followed by a branch migration process, and optionally the dissociation of an incumbent toehold. Because the migration of the branch point is nearly isoenergetic, this process is often intuitively thought of as a random walk whose basic steps consist of single-base association and dissociation steps involving the invading and incumbent domains. The reaction is driven to completion by the enthalpy gain from the formation of base pairs to the toehold, and by the fact that the reverse reaction lacks a toehold and is thus kinetically unfavorable.[9,10]

This class of reaction has proven difficult to model quantitatively. Zhang and Winfree's model for toehold exchange complexes was the first to integrate all three processes involved in toehold-mediated strand displacement—the association, branch migration, and dissociation—into a single model. The model considers each of these three processes to be a single step with associated rate constants, from which a bimolecular rate constant can be derived to more simply approximate the reaction kinetics. Empirical investigations showed that at low concentrations (~ 1 nM), the toehold exchange reaction followed bimolecular kinetics, and the bimolecular rate constant could be quantitatively predicted based upon the domain lengths and the toehold binding energies. The bimolecular approximation was not predictive at higher concentrations because under these conditions the intermediates have non-negligible steady-state concentrations. A critical concentration was analytically derived, above which the bimolecular approximation deviated appreciably from the three-step model. When the invading toehold

[9] D. Y. Zhang and E. Winfree. Control of DNA strand displacement kinetics using toehold exchange. *J. Am. Chem. Soc.*, 131:17303–14, 2009. doi:10.1021/ja906987s

[10] N. Srinivas, T. E. Ouldridge, P. Šulc, J. M. Schaeffer, B. Yurke, A. A. Louis, J. P. K. Doyle, and E. Winfree. On the biophysics and kinetics of toehold-mediated strand displacement. *Nucleic Acids Res.*, 2013:1–18, 2013. doi:10.1093/nar/gkt801

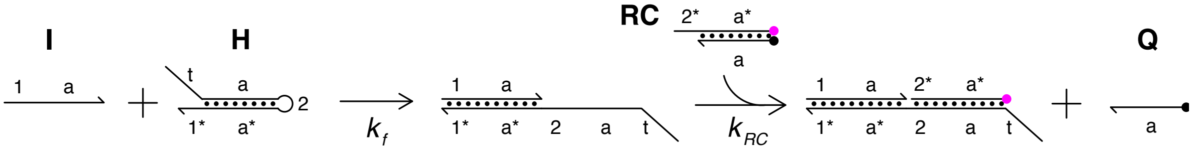


Figure 4.1: Our model system for the hairpin opening. The initiator **I** opens the hairpin **H** with rate k_f, followed by a dissociation reaction with the reporter complex **RC** at a much faster rate k_{RC} to yield free **Q** strand. This last reaction separates a fluorophore (pink dot) from a quencher (black dot), increasing the fluorescence as the reaction proceeds.

length was greater than 6 nt and the incumbent toehold length was equal to or less than 6 nt, this critical concentration was estimated to be about 30 nM.[9]

The present study is intended to be a preliminary investigation into the kinetics of hairpin opening reactions. Hairpins are structurally different than toehold exchange complexes in that the incumbent domain is part of the same strand as the base domain. In addition, we are interested in the kinetics of hairpins at the concentrations typically used for developmental self-assembly, around 10–100 nM. These are higher concentrations than the toehold exchange studies were performed at, at which the bimolecular model is unlikely to hold.

Lastly, the hairpins used in developmental self-assembly often have tails. This has a number of consequences that may affect the hairpin opening rate: the *steric* effect of the tail physically blocking the initiator strand, the *electrostatic* effect of the negatively-charged tail and initiator repelling each other, and the *conformational* effects of the tail engaging in spurious binding to the toehold. We designed a series of hairpins with constant toehold, stem, and loop sequences, changing only the tail sequence to measure the effects on the opening rate. We designed a hairpin with a poly-T tail and a toehold containing only the bases T and C, so that there would be no spurious toehold–tail binding. We then made several variants by introducing non-T bases into the tail to induce toehold–tail binding. The opening of the hairpin triggered a downstream reaction with a reporter complex that caused an increase in fluorescent signal (Fig. 4.1).

Methods

Strand synthesis. DNA strands were synthesized and purified by Integrated DNA Technologies (IDT), including strands containing fluorophores. We quantitated the concentrations of DNA stock solutions by measuring the ultraviolet light absorption at 260 nm with the micro-volume pedestal of a NanoDrop 2000c spectrophotometer, taking the average of measurements of three samples for each strand, and using extinction coefficients provided by IDT to calculate the concentration of each strand.

Sample preparation. The hairpin strands were separately heated to 95°C for 5 min and allowed to cool to room temperature over about 20 min. The reporter complex was annealed in the same way, except that 1.2 equivalents of the **Q** strands were combined with one equivalent of the **F** strand before annealing. For the complex **1**, stoichiometric quantities of strands **1a** and **1b** were mixed before annealing. We then mixed the desired reactants in a half-area, non-stick 96-well plate to final concentrations of 0.5 equivalents initiator, 1 equivalent hairpin, and 1.5 equivalent reporter complex in a final volume of 175 μL. These concentrations were used in order to obviate the effects of inaccuracies in stoichiometry by ensuring that the initiator would be the limiting reactant; we believe that the concentration of the initiator would be the most precise since all reactions would use samples from the same aliquot.

Fluorimetry. We used a Synergy Neo HTS Multi-Mode Microplate Reader to observe the fluorescence with excitation at 661 nm and emission at 691 nm. Alexa Fluor 660 was chosen because its peak excitation and emission wavelengths are far apart, in order to eliminate crosstalk between the two channels. We took measurements every 30 s over 3–5 hr. We recorded the amount of time between the initiator addition and the first measurement for each well, and factored this into our analysis.

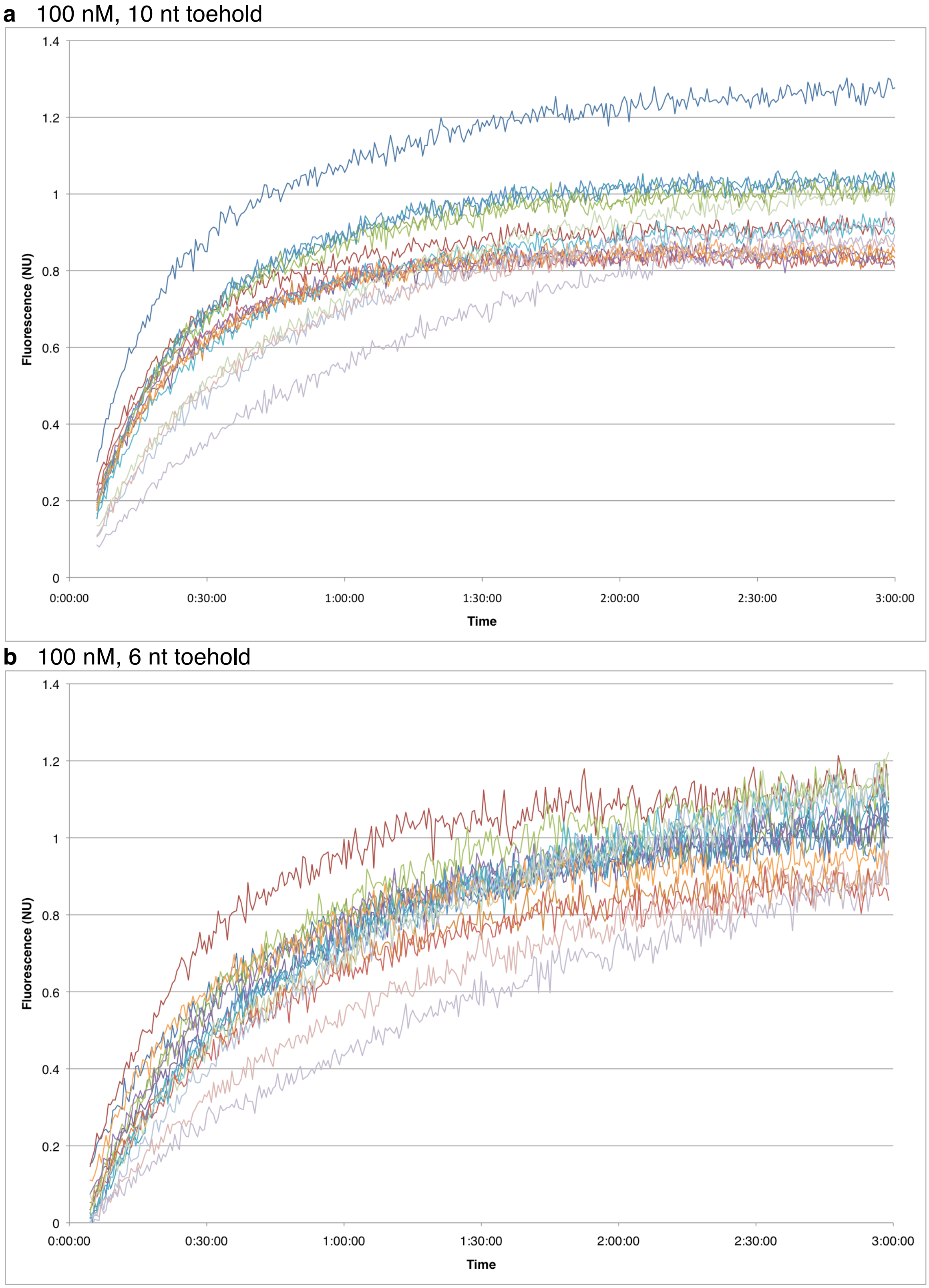
Analysis. We ran five trials of each hairpin on the same plate for each set of conditions, except for the 6 nt toehold samples at 100 nM, for which three trials were run. Values are reported as normalized units (NU), where the raw absorbance has been adjusted linearly so that the background fluorescence of the quenched reporter complex is 0 NU, while the fluorescence of the unquenched fluorophore complex at the same concentration as the limiting reagent is 1 NU. Thus, the value in NU is theoretically equal to the fraction completion of the reaction. We experimentally determined the two anchor values of the quenched and unquenched reporter complex; to reduce noise, we averaged all data points over time for the quenched anchor point, and used a 5-minute moving average for the unquenched anchor point, because it was observed to gradually change over time.

The value $[Q]_{1hr}$ is a proxy for the rate of the reaction, and represents the degree of completion after one hour. It was calculated as the average of the five samples over the ten time points from 60 min to 64.5 min, and the standard deviation of these fifty data points is also reported. We experimentally observed that the reporter complex displacement reaction was complete within about 10 min (data not shown), more than an order of magnitude slower than the timescales of the hairpin opening reactions, and was thus expected to have a minimal effect on the value of $[Q]_{1hr}$.

Results

The averaged, concentration-normalized timecourses for the hairpins under each set of conditions tested are shown in Fig. 4.2. We tested sixteen hairpins under four sets of conditions: at 100 nM or 10 nM hairpin concentrations, each with the full initiator with a 10 nt toehold, and a shortened initiator that had a 6 nt toehold domain. The extracted values for $[Q]_{1hr}$ and their standard deviations are shown in Fig. 4.3.

Comparing hairpins to displacement complexes. We ran the strand displacement complex **1**, which is composed of two strands corresponding to two halves of the hairpin **2**. We found that the hairpin has a slightly lower rate than the complex under three of the four conditions tested. This is expected because the reverse rate should be higher for the hairpin due to the increased effective concentration of the bound output region. However, the fact that the complex has greater than 1 NU for the 100 nM, 10 nt sample indicates that the stoichiometry of the two strands **1a** and **1b** may have been inexact.



a 100 nM, 10 nt toehold

b 100 nM, 6 nt toehold

Figure 4.2: Timecourses of all hairpins. The fluorescence is shown in normalized units (NU) that correspond to the fraction completion of the reactions, as explained in the Methods section. Each data point shown is the average of 3–5 independent trials.

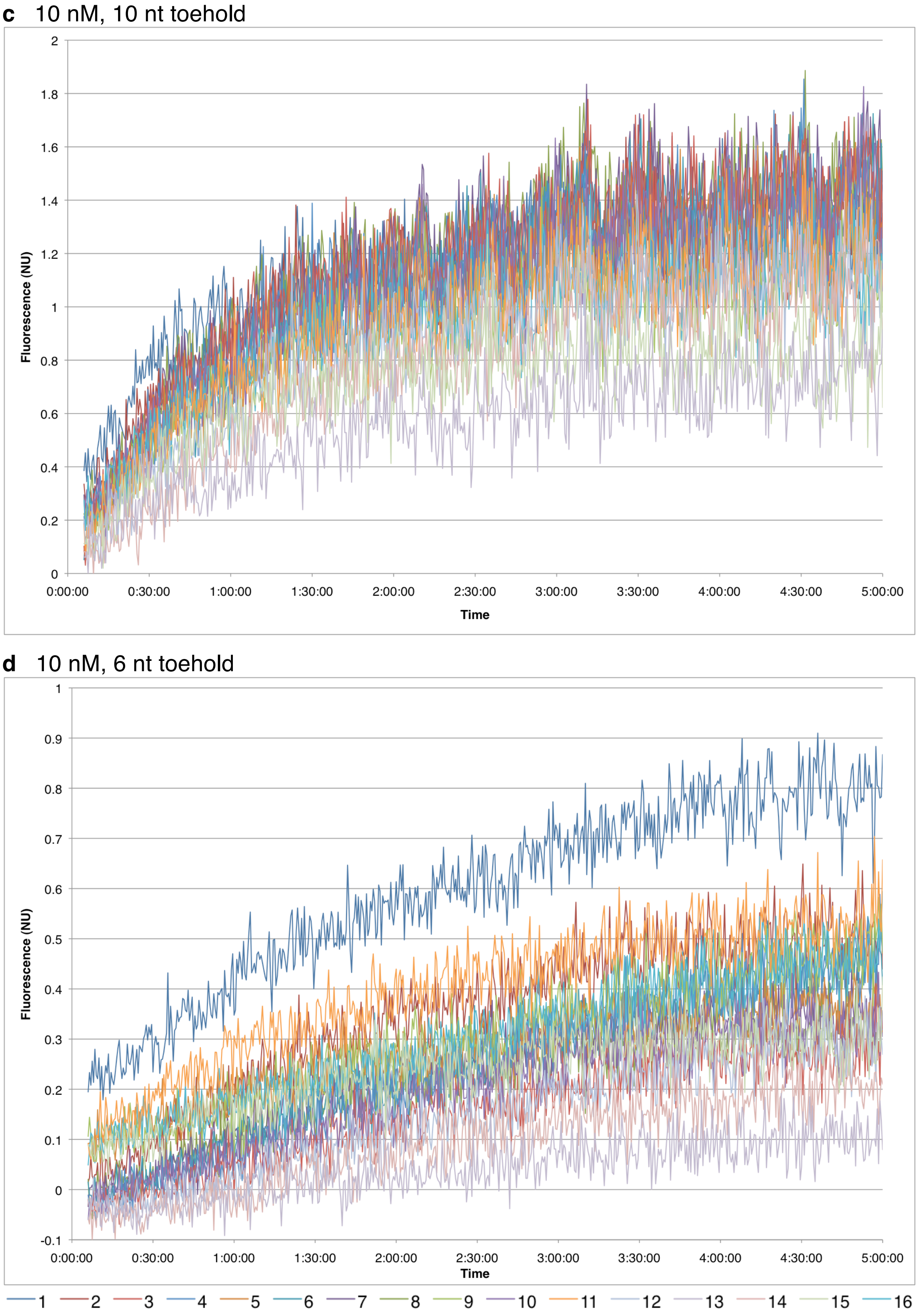


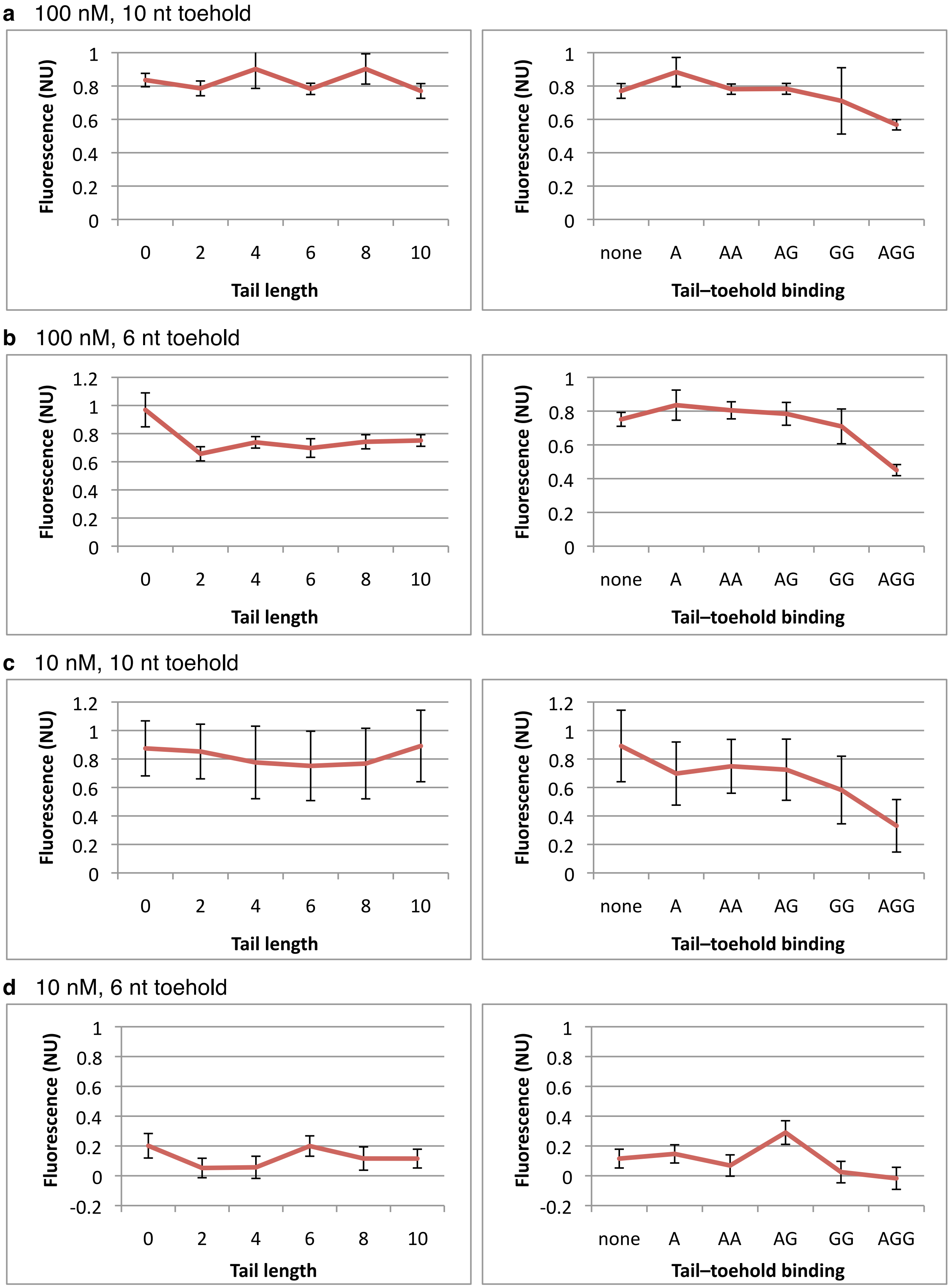
Figure 4.2 (Continued).

Molecule	Tail	100 nM, 10 nt		100 nM, 6 nt		10 nM, 10 nt		10 nM, 6 nt	
		$[Q]_{1hr}$	σ	$[Q]_{1hr}$	σ	$[Q]_{1hr}$	σ	$[Q]_{1hr}$	σ
1	*none (broken loop)*	1.087	.113	.791	.138	.962	.267	.429	.106
2	*none*	.836	.040	.969	.121	.875	.193	.201	.082
3	T2	.786	.044	.657	.050	.853	.192	.052	.065
4	T4	.902	.116	.738	.041	.776	.255	.057	.074
5	T6	.783	.034	.698	.066	.751	.244	.200	.068
6	T8	.902	.091	.742	.050	.768	.248	.116	.078
7	T10	.770	.044	.751	.041	.891	.251	.115	.063
8	*PEG*	.892	.093	.805	.061	.927	.270	.113	.083
9	T5-A-T4	.883	.088	.835	.089	.698	.222	.147	.061
10	T4-AA-T4	.781	.031	.805	.051	.749	.189	.069	.071
11	T4-AG-T4	.783	.033	.784	.068	.725	.215	.290	.079
12	T4-GG-T4	.711	.199	.710	.103	.582	.237	.025	.072
13	T3-AGG-T4	.568	.031	.451	.033	.331	.184	-.017	.074
14	T6-GG-T2	.712	.037	.554	.036	.498	.165	-.014	.058
15	T2-GG-T6	.744	.217	.710	.075	.587	.202	.151	.079
16	T1-AATAA-T4	.779	.208	.762	.060	.706	.233	.177	.070

Figure 4.3: Sequences and fitted rate data. $[Q]_{1hr}$ is the fraction completion of the reaction after 1 hr, as inferred from fluorescence data, while σ is the standard deviation of the data points averaged to calculate $[Q]_{1hr}$. The initiator sequence was TCTTCCTCTC.

Effect of tail length and charge. We ran hairpins with poly-T tail lengths of 0, 2, 4, 6, 8, and 10 nt (hairpins **2**–**7**). We found that the rate showed no discernible trend as the tail length increased from 0 nt up to 10 nt (Fig. 4.4, left panels). We also ran a hairpin with a PEG tail (hairpin **8**); this mimics the steric effects of the tail while lacking the electrostatic repulsion of a nucleotide tail. The rates for the PEG tails were generally within the range for the poly-T tails under corresponding conditions.

Effect of toehold–tail binding. We introduced non-T bases into the tail to cause complementary sequences for toe–tail binding. These included no interactions (hairpin **7**), a single base A forming no base pair–base pair stacks (hairpin **9**), three single stacks of varying strength AA, AG, and GG (hairpins **10**–**12**), and a double stack AGG (hairpin **13**). Our results indicate that the AGG interaction, and perhaps the GG interaction, are strong enough to lower the rate, but the weaker interactions do not have a clearly discernible effect (Fig. 4.4, right panels).



a 100 nM, 10 nt toehold

b 100 nM, 6 nt toehold

c 10 nM, 10 nt toehold

d 10 nM, 6 nt toehold

Figure 4.4: Graphs of selected data from Fig. 4.3. The error bars are standard deviations.

We also studied the effect of the position of the bases by placing the GG sequence closer or farther from the stem (hairpins **14**–**15**). This had no discernable effect. We also studied the effect of having two AA stacks rather than one (hairpin **16**). Again, this seemed to have little effect, as the rate with two AA stacks was not significantly lower than that with one AA stack.

Bimolecular model. We also attempted to analyze the system in terms of the bimolecular model used by Zhang and Winfree.[9] We ran hairpin **2** at different concentrations and fit the timecourses to a bimolecular rate constant model. We found that the fitted rate constant varied strongly with respect to hairpin concentration (data not shown). A fundamental feature of a valid kinetic model is that rate constant must be invariant over reactant concentrations. The lack of this feature suggests that the bimolecular model does not accurately describe the actual kinetics of our system. Indeed, this is not entirely unexpected, as our range of concentrations is much higher than the studies on toehold exchange complexes for which the bimolecular approximation was found to hold.

Discussion

These results for these experiments indicate that, under the conditions tested, the tail appears to have little discernible effect on the hairpin opening kinetics, unless there is relatively strong toehold–tail binding. This supports the importance of eliminating these strong spurious interactions while ignoring weak ones in sequence design approaches for systems of this type. Nevertheless, this is intended to be only a preliminary study to observe overall trends rather than to generate a predictive quantitative model, and these results are not to be considered conclusive. Further study with other concentrations and toehold lengths, as well as different sequences, is needed.

The low dependence of the rates on toehold configuration may indicate that the toehold association step may not be the rate-limiting step in this reaction: the intramolecular branch migration process is likely slower and thus dominates the rate. It is possible that as the concentration or the toehold association energy decrease, a critical point is reached at which the toehold association is sufficiently slowed that it becomes the rate-limiting step and begins to dominate the overall rate. More research is required to investigate this hypothesis.

Acknowledgements

I thank David Zhang for conceiving this project and performing the motif and sequence design, and Xi Chen for useful discussions.

Appendix A

Technical documentation for Multisubjective v1.0.9

"We will encourage you to develop the three great virtues of a programmer: laziness, impatience, and hubris.... Laziness, the quality that makes you go to great effort to reduce overall energy expenditure. It makes you write labor-saving programs that other people will find useful, and document what you wrote so you don't have to answer so many questions about it.... Impatience, the anger you feel when the computer is being lazy. This makes you write programs that don't just react to your needs, but actually anticipate them. Or at least pretend to.... Hubris, excessive pride, the sort of thing Zeus zaps you for. Also the quality that makes you write (and maintain) programs that other people won't want to say bad things about."

—Larry Wall

Overview

Multisubjective improves the quality of sequence designs for developmental nucleic acid systems and other systems used in DNA nanotechnology, and automates several tasks in the process of sequence design. Multisubjective can work in concert with another designer, such as the NUPACK multi-objective designer or Dave Zhang's Domain Design (DD), to minimize undesired secondary structure in these systems.

Multisubjective analyzes the secondary structure of a candidate sequence or set of candidate sequences, and identifies a minimal set of specific bases which are responsible for undesired secondary structure. It then instructs the designer to redesign only those bases, holding the others constant. Multisubjective also enforces exclusion of sequences consisting of a certain number of a single base type in a row, including mixed bases such as S or W. Iterating the analysis and redesign steps leads to sequences of better quality.

Multisubjective also streamlines the design process by interacting directly with a client designer, automatically passing the analyzed sequences to the designer and getting the redesigned candidate sequences from it. Multisubjective natively uses DD as the designer. It is also capable of automatically downloading and processing NUPACK job results, and of automatically submitting new design jobs to the web server. Multisubjective uses a local copy of NUPACK analysis to generate the base pair probabilities used in Multsubjective's analysis.

Installation. In all cases:

- Install NUPACK 3.0 (http://www.nupack.org/).
- Obtain the file *multisubjective.cpp* (v1.0.9) and compile it using Xcode, gpp, or another compliant compiler.

If automatic interaction with the NUPACK web server is to be used:

- Install cURL 7.21.6 (http://curl.haxx.se/).

If automatic interaction with DD is to be used:

- Install Node.js 0.6.12 (http://nodejs.org/).
- Create the directory *$HOME/Documents/Multisubjective* . (Another location may be used, in which case the *$CLDDPATH* environment variable should be set to specify this.)
- Run the command “npm install commander underscore format” in that directory.
- Obtain the files *dd.js* (v0.3) and *cldd.js* (v0.4.7.3) and place them in that directory. Set *cldd.js* to be executable.

Input

Multisubjective requires a specification file that is an augmented version of the NUPACK “.np” specification, and optionally a file containing the base sequences for the system. These should be placed in the Multisubjective working directory, which can be specified within the program interface. The shown filenames are the defaults, but most of them can be changed through the user interface or the command line call.

NUPACK multi-objective specification. The file *specification.np* contains an augmented from of the NUPACK multi-objective design specification for the structure. The file must also contain extra lines needed by Multisubjective that relate to the function of the strands within the developmental hairpin system. The syntax is identical to the NUPACK syntax with different keywords. The format of this file is explained in a later section in this documentation.

Strand or block sequences. Multisubjective loads the actual sequences in one of two formats. The file *candidate.npo* contains strand sequences, and can be automatically obtained from the NUPACK web server. (Note that Multisubjective will overwrite this file if a new job is requested.) The file *candidate.dd* contains block sequences, and is used if Multisubjective is told to load data in DD format. If multiple DD files are to be run, the series of ten files *candidate-1.dd* to *candidate-10.dd* are used. Multisubjective can also create random sequences, in which case no sequence input file is needed.

Output

Multisubjective outputs data into three files, with an extra two generated if autosubmission to the NUPACK web server is utilized. These are all placed into the Multisubjective working directory specified in the program interface. Note that, if Mutisubjective is set to run in a loop, the files *final.dd* and *final.msq* will be generated, containing a copy of the best candidate sequence from the final iteration of the analysis–redesign loop.

New block sequences. The file *analysis.msq* contains the sequences of all the blocks, including mixed bases as determined by the Multisubjective algorithm, in the NUPACK "domain=" format. These can be copy-pasted into a NUPACK multi-objective design specification for manual input to the web server. The file *analysis.dd* contains the same sequences in DD format, but with each mixed base set to a random base consistent with that mixed base, with all the non-mixed bases set to be immutable by DD, and with each domain's sequence constraint set to the union of all the mixed bases in that domain.

Internal data log. The file *analysis.log* contains a mirror of internal data stored after Multisubjective has finished loading and processing the three input files, as well as a record of Multisubjective's secondary structure analysis. It is useful for debugging purposes.

NUPACK web server autosubmission data. If automatic submission to the NUPACK web server is utilized, Multisubjective will generate two additional files. The file *analysis.post* contains a machine-readable version of the full NUPACK multi-objective design file including the new block sequences with mixed bases, used in the HTTP POST request to the NUPACK webserver. The file *response.html* contains the web server's response to the POST request. If successful, the file contains the URL of the newly created job, and a subsequent run of Multisubjective can use this file to autofill the job number and token, to simplify the design process.

Program operation

Upon starting, the program gives the user a choice of input modes:

- Load a DD file (d). The program loads sequence data from the file *candidate.dd* . Multisubjective automatically converts the block sequences contained in the DD file into the strand sequences used by Multisubjective.
- Load multiple DD files (m). The program will run ten times using the files *candidate-1.dd* to *candidate-10.dd* . This option is useful for quickly comparing several related DD outputs and determining which is optimal.
- Fill with random bases (f). The program creates ten sets of random block sequences consistent with the base constraints in the "domain=" blocks of the specification file, and uses these to construct ten sets of strand sequences. (Note that the constraints will not be honored in subsequent rounds.) This option should be used if sequences are to be directly loaded from the specification file, even if none of them are degenerate.
- Seed with independent DD trials (i). DD is run ten times from ten independent seeds consistent with the base constraints in the "domain=" blocks of the specification file, and the resulting ten sets of sequences are used as the initial candidate sequences. (Note that the constraints may not be honored by DD and will not be honored by Multisubjective in subsequent rounds.)
- Load a NUPACK-MO file (n). The preexisting *candidate.npo* file is used. This is useful for running the same sequence many times.
- Autofill from last MO web submission (a). The program gets the job number and token from *response.html*, which contains the information for the job most recently submitted to the NUPACK web server, in a previous run of Multisubjective. This is useful when iterating sequences between Multisubjective and NUPACK multi-objective design.
- Input by job number (j). If the user inputs a number, the program checks whether the job has already been downloaded within the current working directory. If not, the program asks for a token, uses

cURL to download the job from the NUPACK web server to the file *mo-output.zip*, and unzips the job into the folder */mo_output* . It then asks for the trial id and copies the requested sequence file to *candidate.npo*, overwriting a previous file if necessary. (The keyword "all" can be used in place of a trial id to cause Multisubjective to test trials 0–9 of in that job.) If the job has already been downloaded, it asks only for a trial id, and copies the desired file from */mo_output* to *candidate.npo* .

- Set other options (s). A submenu allows the user to change the Multisubjective working directory (where the input and output files are stored), the local NUPACK home directory, the filenames of the two specification files, and the filename prefixes of the sequence input and output files. These can be entered manually, or a filename can be provided that should contain these data each on a single line. These settings are stored in the default configuration file, and automatically reloaded on subsequent runs of Multisubjective.

The user also has a choice of iteration modes. In all cases, the sequence given to the designer has only those bases unlocked that were identified by Multisubjective as problematic. If multiple sequences were tested by Multisubjective, only the one with the fewest number of undesired bases is passed to the designer. The iteration modes are:

- Run DD once (o). The program runs one round of DD, yielding ten new designs.
- Run DD in loop (l). Ten rounds of DD are run, with ten new designs generated in each round with only the one with the fewest number of undesired bases advancing to the next round.
- Submit to NUPACK-MO web server (w). Multisubjective generates a formatted version of the NUPACK multi-objective design file, with the new block sequence assignments inserted in place of the grave character (`) delineated block in the *specification.np* input file, and replacing exclamation point characters (!) with hash characters (#) to comment out the Multisubjective-defined lines. Note that the submitted request copies the contents of *specification.np* verbatim, and it is recommended to verify that the file contains no errors beforehand. Multisubjective then uses cURL to submit an HTTP POST request to the NUPACK web server.

- Use random bases in loop (r). Ten rounds are run, with ten new random designs generated in each round (within the prevented sequence constraints) with only the one with the fewest number of undesired bases advancing to the next round.
- No designer (x). No designer is used after Multisubjective analyzes the input design for undesired secondary structure.

After data have been loaded from the input files, Multisubjective uses the local installation of NUPACK to analyze the secondary structure of the strands. This is done in two passes. The first pass contains the full sequence of each hairpin, testing the “closed” hairpin structures. The second pass includes both the hairpins with their input domains removed, giving the “open” hairpin structures, as well as the bridge complexes. All NUPACK analysis files are stored in the directory */nupack* .

Multisubjective then tabulates the undesired secondary structure from the NUPACK analysis, and decides which bases need to be changed to disrupt this secondary structure, avoiding changing the desired bases in the hairpin and bridge complexes, as well as the immutable bases calculated from user input. All such bases are changed to N. It also checks for the existence of prevented sequences, and inserts the proper mixed base to disrupt this sequence (e.g., if there are too many A’s in a row, one of them will be changed to a B; if there are too many W’s in a row, one will be changed to an S), again avoiding changing immutable bases. The algorithm seeks to minimize the number of changed bases by changing an N to the mixed base if one is in the proper range of positions. If the presence of immutable bases prevents any of these changes from being made, a warning is output to the screen.

Once these changes have been made to the strand sequences, the block sequences must be extracted for output. Since each block may appear many times throughout the system, and each instance may have been changed in different ways, a base collision scheme is used to resolve these differences. Mixed bases other than N take precedence over N’s, N’s take precedence over single bases, and for two non-N bases, the intersection of the two bases is taken unless the intersection is empty, in which case the earlier base is used and a warning is output to the screen.

Once the strand sequences have been compiled, they are written to the output files. If appropriate, the output files are then sent to the designer chosen by the user. If Multisubjective is to be run in a loop, the input mode is reset to ‘m’ and the program then loops ten times, and then prompts the user to run more loops in blocks of five. The pair probability threshold may be changed after each block of rounds. After all desired loops are complete, Multisubjective cheerfully exits.

Command line operation

If Multisubjective is called with any command line arguments, the user interface will be skipped and information will be taken from the arguments. The syntax is:

multisubjective -m mode [-j job_number] [-r trial_id] [-k token] [-c config_filename] [-d working_dir] [-i specification_infile_prefix] [-s sequence_infile_prefix] [-o outfile_prefix] [-h NUPACK_homedir] [-w]

where:

- The argument *mode* is a two-letter code specifying the input and iteration modes. The first letter is one of the letters ‘d’, ‘m’, ’f’, ‘i’, ‘n’, ‘a’, or ‘j’, representing the input mode as explained above. The second letter is one of the letters ‘o’, ‘l’, ’w’, ‘r’, or ‘x’, representing the iteration mode as explained above. This argument is required. If option ‘j’ is used, the arguments *job_number* and *trial_id* are always required, and the argument *token* is also required if the job is not available locally.
- The argumen*t config_filename* is the filename including full path of a configuration file. If this argument is used, the options *-d*, *-i*, *-s*, *-o*, and *-h* should be omitted as these data are being loaded from the configuration file instead.
- The other command line arguments correspond to items listed in “Format of the configuration file” below; the filenames should not include a path as *working_dir* is used. If any or all of these are omitted

and *-c config_filename* is not used, the omitted data will be loaded from the Multisubjective default configuration file if it exists, otherwise the default values will be used.

- The option *–w* enables Workbench mode. This generates three extra files: *analysis.mso* contains Multisubjective's analysis data in JSON format, while *nupack/ms0.ocx-mfe* and *nupack/ms2.ocx-mfe* are the NUPACK-generated minimum free energy structures for the closed and open strands, respectively.

Format of the specification file

The file *specification.np* contains the NUPACK multi-objective design specification for the structure. Multisubjective accepts both the old and new NUPACK formats. Multisubjective only uses the following lines: "structure", "strand", ".seq", and the reaction condition lines; any other lines used by NUPACK may be present and will be ignored by Multisubjective. Structure inputs in DU notation are preferred; dot-paren notation is also supported, with a slightly modified format in some specific cases. (For dot-paren structures that have two stems immediately adjacent to each other, i.e., there are no unpaired bases between them, a colon character (:) must be inserted to separate the sets of parentheses from each stem. If the colon character is omitted, an error will usually result, but there are some structures that will be translated into an incorrect but valid structure without causing an error.) Multisubjective ignores strand breaks ('+' characters) and concatenates each structure into a single strand.

If submission to the NUPACK web server is to be used, the "domain" group of lines must be bracketed with two lines each containing a single grave character as a comment (#`); alternatively, the entire group of lines may be omitted and replaced with a line containing two grave characters in a comment (#``). The grave characters tell Multisubjective where to insert the new sequences in the HTTP POST request after they have been generated. The old block of assignments, if present, is discarded by Multisubjective.

The file *specification.np* must also contain data needed by Multisubjective that is absent in the NUPACK specification. Most of these data relate to the function of the strands within the developmental hairpin system. The syntax is identical to the NUPACK syntax with different keywords, all of which begin with an exclamation point character (!). The format of this file is explained in a later section in this documentation.

In most cases there are two varieties of specification for each type of strand function. The *automatic* notation causes Multisubjective to calculate some of the needed information based upon the user specification and information in *specification.np*, while in the *explicit* notation the user directly supplies the result of these calculations. The automatic notation contains a colon (:) as its operator, while the explicit notation contains an equals sign (=). The automatic notation is easier to use, but the explicit notation is useful for non-standard structures and in troubleshooting.

Static structures

Automatic: !static [name]

Structures declared in this way are included in the "closed" analysis pass, but are omitted from the "open" analysis pass. This is useful for analyzing strand that are not dynamic, or for directly analyzing strands whose configurational changes are too complex for Multisubjective's preprogrammed strand types; the closed and open structures can be individually passed to Multisubjective using this notation.

Hairpins

Automatic: !hairpin [name] : [+/-]

Explicit: !hairpin [name] = [offset]

A hairpin is specified using the syntax "hairpin A1 : +" where A1 is the name of the strand (which must correspond to the name used in the NUPACK specification in *specification.np*), and one of the characters '+' or '-' to specify the polarity of the hairpin. A hairpin has positive polarity if its toehold is at the 5' end of the strand, and has a negative polarity if its toehold is at the 3' end.

The explicit syntax instead specifies a number called the *offset*, which tells Multisubjective what portion of the hairpin sequence to remove to obtain the open hairpin sequence. This has the same sign as the polarity, and its magnitude is either the length of the input port (if the polarity is positive), or the length of the open portion (if the polarity is negative).

Cooperative hybridization complexes

Explicit: !coop [name] = [size]

The *size* of a cooperative complex is the length of the longer strand in the complex. In the structural specification in *specification.np*, the longer strand should be listed first, i.e., it should have a form like "structure C1 = U8 D37 (U8 +)".

Specifying immutable bases

Explicit: !immutable = [off/auto/blocks [blocknames]]

Multisubjective requires the location of bases desired to be immutable. The "off" option is the default and will result in no bases being immutable. The "auto" option will cause immutable bases to be set using the "domain=" lines, where each position with an unambiguous base (A, C, G, or T) will be set as immutable. (Note that if a sequence is loaded from a file, the locations of the immutable bases will be preserved, but the base identities will be overwritten.) The "blocks" option is followed by a list of block names fro which all bases within those blocks will be set to be immutable. Specifying immutable bases is useful in the case of endonuclease restriction sites, or for pre-specifying sequences of clamping regions.

Specifying the pair probability threshold

Explicit: !threshold = [value/worst N]

Explicit: !toethreshold = [value]

Explicit: !intermolecular = [on/off]

The threshold can be specified here, as well as an optional lower threshold for base pairs involving an exposed toehold. If either of these lines is absent, the default value of 0.67 is used for that threshold. The keyword "worst" followed by an integer N causes the threshold to be automatically set to the pair probability of the Nth worst undesired base pair for each candidate sequence. The "intermolecular" line specifies whether intermolecular interactions between open hairpins are to be considered; this increases the quality of the sequences but is much slower.

Specifying the prevented sequence limits

Explicit: !prevent [base] = [value]

The specified base, which may be a mixed base, is prevented from appearing in consecutive repeats of the given value. The default values are: 4 for A, C, G, and T; 6 for S, W, R, Y, K, and M; 100 for B, D, H, and V. Note that X and N are not valid arguments for this statement.

Format of the configuration file

Configuration files contain the following information, each item on a separate line:

- The Multisubjective working directory, where input and output files are stored
- The filename prefix, without an extension, of the specification input files (the default is *spec*)
- The filename prefix, without an extension, of the sequence input file (the default is *candidate*)
- The filename prefix, without an extension, of the output files (the default is *analysis*)
- Optionally, the NUPACK home directory preset, which is used only if $NUPACKHOME is not set

The default configuration file is *$HOME/Documents/Multisubjective/multisubjective.cfg* . If it exists, it is loaded at the beginning of the program. A new copy is saved after the user inputs new data through the user interface, but not if these options are changed through the command line. Deleting *multisubjective.cfg* will cause the Multisubjective's original default values to be restored.

Exception handling, signal handling, and exit status

If an error condition occurs during the program's execution, an exception will be thrown with a unique identifying number in the range 1–139. A short descriptive message containing the exception number will be output to the screen, and the exception number will be noted at the end of the *analysis.log* file. Signals indicating a program error are caught and handled in the same way as exceptions.

Upon successful termination, Multisubjective will have exit status 0. If an exception was thrown, the exit status will be equal to the number of the exception if it is in the range 1–250, otherwise the exit status will be 255. If the program terminated because it received the signals SIGINT, SIGHUP, SIGTERM, or SIGQUIT, the exit status will be one of the numbers 251–254, respectively.

Appendix B

Strand sequences

"Est ultima mundi ut scimus, et salveo."

—Rem

TET6

A1: ATAGTCTCCACAACAATCGATCGCTCCAAATCACAAGGTCAACAAACAGCACATGTGATTTGGAGCGATCGATGAAGTA

A2: TCTTCTCTAACCCAAACAAGGTCAACAAACAGCACAATCTCACTACTTCATCAGTGCTGTTTGTTGACCTTGTGATTTG

A3: TTCTATACTCCTTCACACAATCTCACTACTTCATCGATCGCTCCAAATCACAAGATGAAGTAGTGAGATTGTGCTGTTT

B1: GAGCGATCGATTGTTGTGGAGACTATAGAAATATAGTCTCCACAACAATCAGTACTACAACAAACAAACACCGAGAAGA

B2: GTTGACCTTGTTTGGGTTAGAGAAGAACTAAATCTTCTCTAACCCAAACAGTATACCATCAGAAACGATATCTATAGAA

B3: GTGAGATTGTGTGAAGGAGTATAGAAATGGGCTTCTATACTCCTTCACACATTAATTCGCGACCAAACCAACAGACTAT

C1: ATCAGTACTACAACAAACAAACACCGAGAAGATTTAGATCTTCTCGGTGTTTGTTTGTTGTAGTACTGATTGTTGTGG

C2: ACAGTATACCATCAGAAACGATATCTATAGAAGCCCAATTCTATAGATATCGTTTCTGATGGTATACTGTTTGGGTTA

C3: CACATTAATTCGCGACCAAACCAACAGACTATATTTCAATAGTCTGTTGGTTTGGTCGCGAATTAATGTGTGAAGGAG

I: TACTTCATCGATCGCTCCAAATCACA

Sequences for quenching experiments (Fig. 3.14a)

These were ordered in two parts and then ligated.

C1aF: ATCAGTACTACAACAAACAAACACC/iCy5/GAGAAGATTTAGATCTTCTCGGTGTTTGT

C1bQ: /5Phos/TTGTTGTAGTACTGATTGTTGTGG/3IABkFQ/

C2aF: ACAGTATACCATCAGAAACGATATCTATA/i6-TAMN/GAAGCCCAATTCTATAGATATC GTT

C2bQ: /5Phos/TCTGATGGTATACTGTTTGGGTTA/3IAbRQSp/

C3aF: CACATTAATTCGCGACCAAACCAAC/i6-FAMK/AGACTATATTTCAATAGTCTGTTGGT TTG

C3bQ: /5Phos/GTCGCGAATTAATGTGTGAAGGAG/3IAbRQSp/

Sequences for alternate TET6 structure (Fig. 3.14b)

B2R: GTTGACCTTGTTTGGGTTAGAGAAGAACTAAATCTTCTCTAACCCAAACAGTATACCAT CAGAATGAAGGAGTATAGAA

C2Ra: ACAGTATACCATCAGAA/i6-TAMN/TGAAGGAGTATAGAAGCCCAATTCTATAC

C2Rb: /5Phos/TCCTTCATTCTGATGGTATACTGTTTGGGTTA

TET8

A1: TTGAAGGTCCATTTAGGCTCTGCGTGATCTGGTTCGTCTGTTACACTCTAGATAACCAG ATCACGCAGAGCCTACATCGTT

A2: GCTTGTCTTACACATTCGTCTGTTACACTCTAGATTAGGATTAACGATGTAGTTCTAGA GTGTAACAGACGAACCAGATCA

A3: AATTCCAGGTACAGGATTAGGATTAACGATGTAGGCTCTGCGTGATCTGGTTTCTACAT CGTTAATCCTAATCTAGAGTGT

B1: CGCAGAGCCTAAATGGACCTTCAATCGTATACCCAGATCTTGAAGGTCCATTTAGCA CATTTACAAGCCCTTAACTC

B2: AACAGACGAATGTGTAAGACAAGCTGTCGCGAGTTAAGGGCTTGTCTTACACATTATG CACAGGAATTCAGGGATAG

B3:AATCCTAATCCTGTACCTGGAATTTGAGTACTATCCCTGAATTCCAGGTACAGGAACACTAGCTTCAAGATCTGGGT

C1: AGCACATTTACAAGCCCTTAACTCGCGACTGAGTTAAGGGCTTGTAAATGTGCTAAATGGAC

C2: TTATGCACAGGAATTCAGGGATAGTACTCTCTATCCCTGAATTCCTGTGCATAATGTGTAAG

C3: GAACACTAGCTTCAAGATCTGGGTATACGTACCCAGATCTTGAAGCTAGTGTTCCTGTACCT

I: AACGATGTAGGCTCTGCGTGATCTGGTT

TET9

A1: AGATGAATTCTTAACTCGTGTCCGGGATATGGGTCTATAAGTGTGATCGGTGTACCCATATCCCGGACACGACGAACCA

A2: GATATGTGATTTAGAACTATAAGTGTGATCGGTGGCTATGAGGATGGTTCGTTCACCGATCACACTTATAGACCCATAT

A3: TAGAATTTGATGACTAGCTATGAGGATGGTTCGTCGTGTCCGGGATATGGGTTACGAACCATCCTCATAGCCACCGATC

B1: CCCGGACACGAGTTAAGAATTCATCTTAAGGTGACATTGTTCGGCAGATGAATTCTTAACTCGGTAGTATACGTTAGAG

B2: ACACTTATAGTTCTAAATCACATATCTCTACTGTAGACTTTGACGGATATGTGATTTAGAACTTAGTCGCGATAACTTA

B3: TCCTCATAGCTAGTCATCAAATTCTATTGTACCCTTGGATTAGCTTAGAATTTGATGACTAGCGGAAGTACTGGATCGT

C1: CACATATCCGTCAAAGTCTACAGTAGTCTCTAACGTATACTACCGAGTTAAGA

C1p: CGGTAGTATACGTTAGAGACTACTGTAGACTTTGACG

C2: AAATTCTAAGCTAATCCAAGGGTACATTAAGTTATCGCGACTAAGTTCTAAAT

C2p: CTTAGTCGCGATAACTTAATGTACCCTTGGATTAGCT

C3: ATTCATCTGCCGAACAATGTCACCTTTACGATCCAGTACTTCCGCTAGTCATC

C3p: GCGGAAGTACTGGATCGTAAAGGTGACATTGTTCGGC

I: TGGTTCGTCGTGTCCGGGATATGGGT

Sequences for quenching experiments (Fig. 3.10)

These were ordered in two parts and then ligated.

C1aF: /5TYE665/CACATATCCGTCAAAGTCTACAGTAG

C1bQ: /5Phos/TCTCTAACGTATACTACCGAGTTAAGA/3IABkFQ/

C2aF: /56-TAMN/AAATTCTAAGCTAATCCAAGGGTACA

C2bQ: /5Phos/TTAAGTTATCGCGACTAAGTTCTAAAT/3IAbRQSp/

C3aF: /56-FAMK/ATTCATCTGCCGAACAATGTCACCTT

C3bQ: /5Phos/TACGATCCAGTACTTCCGCTAGTCATC/3IAbRQSp/

TET10

A1: TGATTAAGTAAGTATGAGTATGATAGTGTAGTATTGGCGGATAGATGATGGTGACCTAGTGATGATAAGTACATGTCACCATCATCTATCCGTACCCATG

A2: GTGTCTCGGAACATGTAAGATAGTGATAGAATAGATACTAGTGATGATAAGTACAGCGACTTTGGCATGGGTATTGTACTTATCATCACTAGGTCACCAT

A3: AAATGGAACTGTTATGGTGATGTGTCTTGTCAGAATTGCGACTTTGGCATGGGTACGGATAGATGATGGTGACTTACCCATGCCAAAGTCGCTGTACTTA

B1: CATCTATCCGCCAATACTACACTATCTGATAGTGTAGTATTGGCGAAATGTATACGTCG

B2: TCATCACTAGTATCTATTCTATCACTTAGTGATAGAATAGATACTTCGTAACGGATAGT

B3: CCAAAGTCGCAATTCTGACAAGACACTGTGTCTTGTCAGAATTGCATAAGTACTAATGG

C1: TCTTACATGTTCCGAGACACCGACGTATACATTTCGCCAATACTACACTATC

C1p: AGTATTGGCGAAATGTATACGTCGGTGTCTCGGAAC

C2: TCACCATAACAGTTCCATTTACTATCCGTTACGAAGTATCTATTCTATCACT

C2p: AATAGATACTTCGTAACGGATAGTAAATGGAACTGT

C3: TACTCATACTTACTTAATCACCATTAGTACTTATGCAATTCTGACAAGACAC

C3p: TCAGAATTGCATAAGTACTAATGGTGATTAAGTAAG

I: CATGGGTACGGATAGATGATGGTGAC

Hairpin kinetics

1a: GTTGAGCTCTAACGTC CATCCAAC

1b: GACGTTAGAGCTCAAC TCTTCCTCTC

2: GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

3: TT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

4: TTTT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

5: TTTTTT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

6: TTTTTTTT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

7: TTTTTTTTTT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

8: /5Sp9//iSp9//iSp9//iSp9/GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCC TCTC

9: TTTTTATTTT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

10: TTTTAATTTT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

11: TTTTAGTTTT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

12: TTTTGGTTTT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

13: TTTAGGTTTT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

14: TTTTTTGGTT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

15: TTGGTTTTTT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

16: TAATAATTTT GTTGAGCTCTAACGTC CATCCAAC GACGTTAGAGCTCAAC TCTTCCTCTC

I10: GAGAGGAAGA GTTGAGCTCTAACGTC

I6: GGAAGA GTTGAGCTCTAACGTC

Q: /5IAbRQ /GTTGAGCTCTAACGTC

F: GTTGGATG GACGTTAGAGCTCAAC /3AlexF532N/

Colophon

This thesis was typeset in 11-point Minion Pro for the main text, and 10-point Minion Pro for captions and footnotes. According to Adobe Systems, “Minion is an Adobe Originals typeface designed by Robert Slimbach. It was inspired by classical, old style typefaces of the late Renaissance, a period of elegant, beautiful, and highly readable type designs. Minion Pro exhibits the aesthetic and functional qualities that make text type highly readable, yet is also suitable for display settings.” Legends internal to figures were set in 6- to 12-point Helvetica.

This thesis was laid out using Microsoft Office Word Mac 2008 and exported to PDF. Most figures were prepared using Adobe Illustrator CS4. Graphs were prepared with Microsoft Office Excel Mac 2008.

This copy was provided by the author to the Harvard Faculty of Arts and Sciences Office of the Registrar in digital form in January 2014.

Document information: 4 chapters and 2 appendices; 44 sections; 82 subsections; 129 pages; 28,630 words (main text: 24,951, notes: 3,679); 190,874 characters; 32 figures; 6 epigraphs; 152 notes. The original file size was 16.3 MB.