Simplifying Chemical Reaction Network Implementations with Two-Stranded DNA Building Blocks

Robert F. Johnson
California Institute of Technology, Pasadena, CA, USA
rfjohnso@dna.caltech.edu

Lulu Qian
California Institute of Technology, Pasadena, CA, USA

Abstract

In molecular programming, the Chemical Reaction Network model is often used to describe real or hypothetical systems. Often, an interesting computational task can be done with a known hypothetical Chemical Reaction Network, but often such networks have no known physical implementation. One of the important breakthroughs in the field was that any Chemical Reaction Network can be physically implemented, approximately, using DNA strand displacement mechanisms. This allows us to treat the Chemical Reaction Network model as a programming language and the implementation schemes as its compiler. This also suggests that it would be useful to optimize the result of such a compilation, and in general to find effective ways to design better DNA strand displacement systems.

We discuss DNA strand displacement systems in terms of "motifs", short sequences of elementary DNA strand displacement reactions. We argue that describing such motifs in terms of their inputs and outputs, then building larger systems out of the abstracted motifs, can be an efficient way of designing DNA strand displacement systems. We discuss four previously studied motifs in this abstracted way, and present a new motif based on cooperative 4-way strand exchange. We then show how Chemical Reaction Network implementations can be built out of abstracted motifs, discussing existing implementations as well as presenting two new implementations based on 4-way strand exchange, one of which uses the new cooperative motif. The new implementations both have two desirable properties not found in existing implementations, namely both use only at most 2-stranded DNA complexes for signal and fuel complexes and both are physically reversible. There are reasons to believe that those properties may make them more robust and energy-efficient, but at the expense of using more fuel complexes than existing implementation schemes.

2012 ACM Subject Classification Computer systems organization → Molecular computing

Keywords and phrases Molecular programming, DNA computing, Chemical Reaction Networks, DNA strand displacement

Digital Object Identifier 10.4230/LIPIcs.DNA.2020.2

Funding *Robert F. Johnson*: NSF Graduate Research Fellowship.
Lulu Qian: NSF grant CCF-1908643.

Acknowledgements We would like to thank Chris Thachuk and Erik Winfree for helpful discussions on new DNA strand displacement motifs and optimization thereof.

1 Introduction

What does it mean to optimize a molecular system? One particular field in molecular programming is currently faced with that question. The Chemical Reaction Network (CRN) model is often used to describe systems of interacting molecules. The model can either describe real systems, to analyze their behavior and computational function, or describe hypothetical systems, with known computational function but perhaps no known physical

26th International Conference on DNA Computing and Molecular Programming (DNA 26).
Editors: Cody Geary and Matthew J. Patitz; Article No. 2; pp. 2:1–2:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

example. It was therefore a significant breakthrough when Soloveichik et al. showed that any CRN, real or hypothetical, can be approximately implemented by a system of DNA strand displacement (DSD) mechanisms [34]. This allows the Chemical Reaction Network model to be used as a programming language, where programs can be written in the abstract and compiled into physical molecules. Other CRN-to-DSD implementation schemes promptly followed [27, 4], each with their own strengths and weaknesses. Some have been implemented experimentally, with variable – but mostly good – degrees of success and robustness [7, 36]. Given a programming language and a concept of compiling it, one would naturally want to optimize the result of that compilation and ask, can we do better than the best implementation schemes so far?

So what does it mean to optimize a DSD system? We focus on DNA-only (or "enzyme-free") systems using standard toehold-mediated 3-way [45, 48] and 4-way [25, 10] strand displacement mechanisms. First, such DSD CRN implementations so far require "fuel species" (or "fuels"), DNA complexes that have to be synthesized by whatever method and added to the DSD system at the start. Fuel complexes that mediate a reaction by interacting with signal strands are often referred to as "gates", though this is not usually formally defined. When testing DSD circuits in the lab, fuels are chemically synthesized, annealed, and manually added to the test tube; in the hypothetical future where DSD is used in autonomous molecular devices, those devices would need some as-yet-undecided mechanism to synthesize or input fuels. Any property of the fuel species, such as length of strands, number of strands, or number of fuels, that makes them more costly to synthesize, or more difficult to synthesize without undesired byproducts, is thus a target for optimization. Second, no physical DSD system ever does exactly what the formal DSD model says it should. Some of this is due to improbable, but not impossible, "leak reactions" not included in the formal model, while some is due to the aforementioned undesired byproducts or other imperfect synthesis of the fuels [36].

In terms of robust DSD systems and their fuels, we can take a lesson from experiments with seesaw gates [28, 40]. For a two-reactant two-product reaction, the Soloveichik et al. translation scheme uses 3-stranded fuels [34], the Cardelli scheme 4-stranded fuels [4], and the Qian et al. scheme [27] (in the corrected version) a 5-stranded or a 7-stranded fuel. The seesaw gates compute logic gates which are less complex than chemical reactions, but they do so with only single strands and 2-stranded complexes [28]. Possibly because of this, they have been used to build larger circuits and to be robust to experimental imperfections, such as unpurified strands [40].

For this purpose, we have been investigating implementing CRNs using only 2-stranded fuels. Simple DSD systems, such as detecting a desired sequence [5] or AND gates [16], are often 2-stranded, in addition to the seesaw gates mentioned above. There is even a class of hairpin-based systems that construct larger structures from single-stranded initial complexes [44], including the Hybridization Chain Reaction often used in imaging [11], and a design for hairpin-based logic circuits [12]. However, none of these are a full Chemical Reaction Network implementation, or even an equivalently powerful dynamical system – while logic gates are universal for computing functions, CRNs have a dynamical behavior that logic gates in general do not.

We focus in this work on DSD systems using only 2-stranded fuels and where all mechanisms are physically reversible. We focus on 2-stranded fuels for the robustness concerns above, as well as the theoretical question of whether 2-stranded complexes are sufficient for complex behavior (as discussed further in [18]). We focus on physical reversiblility because it reduces the quantity of fuel consumed by reversible reactions. Many interesting computations and

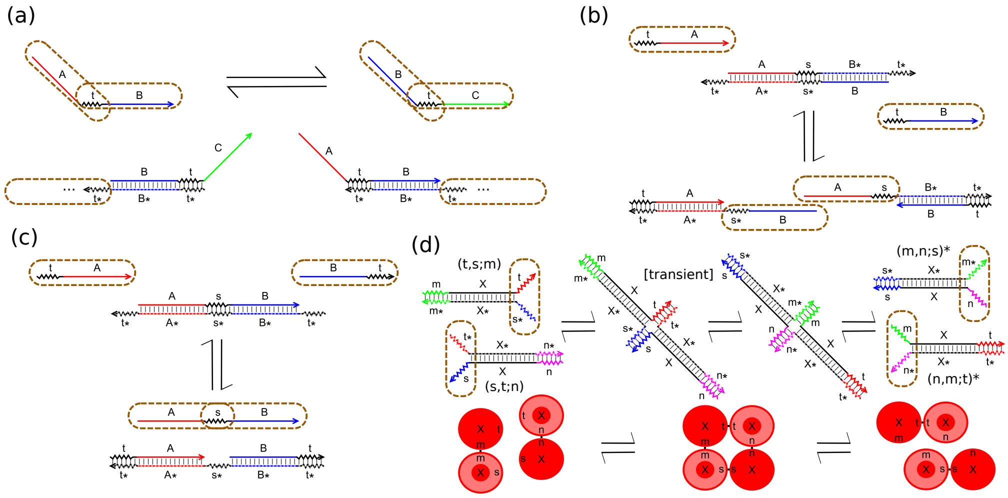

Figure 1 Four previously studied reversible 2-stranded DSD motifs, shown through common examples. (a) Toehold exchange; (b) Symmetric cooperative hybridization; (c) Asymmetric cooperative hybridization; (d) 4-way strand exchange, with a diagram and names used in the abstracted notation we will introduce.

dynamical behaviors require reversible reactions. For example, logically reversible operations allow computation with arbitrarily low energy if they are implemented with physically reversible reactions [2, 3], such as DSD implementations of stack machines [27], Gray code counters [9], and space-bounded computations [37]. DNA buffers [29] use reversible reactions to maintain stable [30] and dynamical [31] spatial patterns. DNA circuits can be reset to process new input signals when reversible reactions are used for restoring fuel molecules in response to reset signals [14, 13, 12]. (Existing implementations often are or can be made physically reversible; Qian et al. [27] demonstrate it explicitly, while simple methods to make other existing schemes [34, 4] physically reversible is an exercise for the interested reader.)

In this work, we discuss ways of implementing CRNs using only 2-stranded fuels and where all mechanisms are physically reversible. We discuss four known 2-stranded DSD motifs that can serve as building blocks for such implementations, and we present a new cooperative 4-way strand exchange motif that starts with 2-stranded complexes. We discuss two ways of implementing general CRNs with these motifs, and tradeoffs between the two schemes. Finally, we show how, using CRN bisimulation, these schemes can be proven correct assuming the assumptions of the formal DSD model reflect real DSD systems.

We believe that having abstract descriptions of simple motifs will help the design of complex DSD systems. Whatever complex behavior is desired, it may be easier to implement by combining the simple logical operations of known motifs. To demonstrate this, we first discuss the 5 motifs and their behavior on an abstract level, then show how various CRN implementations can be constructed and comprehended by combining those abstract behaviors.

2 Two-stranded motifs

We identify five “motifs”, or simple condensed reactions, out of which we build two-stranded CRN implementations. Four of these motifs have been previously studied, while one is new. We discuss the properties of each motif in itself, while in Section 3 we will discuss how those properties interact when building larger circuits. For building two-stranded CRNs, key questions about a given motif are what logical operation it represents, whether its outputs have the form of its inputs and/or the inputs of the other motifs, and whether its outputs and reverse gates are 2-stranded.

Toehold Exchange

A reversible 3-way strand displacement exchanges which of two strands is bound to a gate (Figure 1 (a)). The input strand is an unbound toehold-long domain combination, while the input gate has that long domain bound with that toehold open. The reaction has two high-level effects. First, the output strand has the same long domain (B, in the figure) in a different toehold context, and may have different long domains (A versus C) on the other side of its newly open toehold. Second, the gate now has a different toehold open, which may allow interaction with adjacent domains. See for example the first CRN implementations [34], seesaw gates [28], and various others [47].

Cooperative Hybridization (symmetric)

Two 3-way strand displacement reactions occur simultaneously on either side of a gate complex, meeting in the middle and allowing the two halves to dissociate only if both inputs are present (Figure 1 (b)). The input strands are unbound toehold-long domain combinations, while the output signals have the same long domains adjacent to different open toeholds. See for example Cherry et al.'s winner-take-all circuits [8]. This mechanism, like the two other cooperative motifs, is "cooperative" in the sense that it requires two inputs to simultaneously, "cooperatively", displace parts of the gate complexes for a productive reaction to happen.

Cooperative Hybridization (asymmetric)

Two 3-way strand displacement reactions occur simultaneously on either side of a gate complex, meeting in the middle and releasing an output strand only if both inputs are present (Figure 1 (c)). The input strands are unbound toehold-long domain combinations, while the output strand has those two long domains in combination with a different toehold; but with only one toehold, barring complex mechanisms either one but only one of them can react. However, even if both inputs are single strands the reverse gate is a 3-stranded complex, so this motif is not "reversible with 2-stranded fuels". Introduced and tested by Zhang [46].

4-way Strand Exchange

Two 2-stranded complexes bind by two toeholds and exchange strands via 4-way branch migration (Figure 1 (d)). The inputs are 2-stranded complexes sharing a common long domain, with complementary pairs of open toeholds and (if the reaction is reversible) a closed toehold on each. The outputs are 2-stranded complexes in the same form, with the formerly open toeholds now paired up and closed and the formerly closed toeholds now split and open. Experimentally tested by Dabby [10]. Various mechanisms, simple and complex, based on 4-way strand exchange have been used experimentally in a number of devices [41, 24, 5, 16].

4-way Cooperative Hybridization

Two 4-way branch migrations happen on either side of a gate, meeting in the middle and separating into two intermediate complexes (Figure 2). Observe that the "top" toeholds (t and t) on the initial X and Y complexes end up on one of the two products, while the "bottom" toeholds (s^* and v^*) end up on another. That is, each of the two products carries only half of the information of the original reactants, and products of different instances of this reaction can interact in the reverse reaction. If for example the (t, t) top half of this reaction interacted with a (v^*, s^*) bottom half from a different instance, while the (s^*, v^*) bottom half interacted with an (a, a) top half, the result would be X and Y complexes with the same form as the original reactants but different toehold combinations. The effect of such a quadruplet of reactions is strand exchange between one pair of complexes coupled to strand exchange between the other, simultaneously changing the open toehold combinations on distinct long domains. This is important because affecting distinct long domains in a coupled manner was the one thing that, under a set of additional restrictions that this mechanism satisfies, our previous work [18] showed that uncooperative 4-way strand exchange could not do.

While the other four mechanisms discussed have been experimentally demonstrated to work, cooperative 4-way branch migration has not yet been tested. In particular, the final dissociation step requires 3 toeholds separated by two 4-way junctions to dissociate. We think this is plausible, based on Dabby's observation that 2 toeholds separated by one 4-way junction can dissociate [10]; or, if this is not the case, that there is some $0 < \mathit{Length}(l) \leq 6$ for which that dissociation is possible and reversible. It is possible that $\mathit{Length}(l) = 0$ (i.e. no third toehold) will give the desired behavior, but from Dabby's results, "closed" (both toehold lengths at least 2) 4-way branch migration seems to proceed much faster than "open" 4-way branch migration. Thus we suspect that $\mathit{Length}(l) \neq 0$, and in particular $\mathit{Length}(l) \geq 2$, will give the desired fast and reversible reaction kinetics.

An abstraction for 4-way-based mechanisms

Common to both uncooperative and cooperative 4-way strand exchange is a basic signal complex: two strands, one long domain bound to its complement flanked by one bound pair of complementary toeholds and one open pair of non-complementary toeholds, as seen repeatedly in Figures 1 (d) and 2. As both types of 4-way strand exchange transform complexes of this form into complexes of the same form with different domain combinations, we find an abstract description of this type of molecule useful. For example, we write the molecule with long domain X, open 3' (end of the DNA) toehold t, open 5' toehold s^*, and bound toehold m as $X(t, s; m)$. Note that the semicolon distinguishes open toeholds t, s^* available for interaction from the closed (m, m^*) toehold pair that cannot interact with other complexes, but can be opened for interaction by a reaction. When the long domain is unimportant or universal, such as a system composed entirely of uncooperative 4-way strand exchange, we omit it and write simply $(t, s; m)$. For experimental reasons we prefer to have strands made up of only non-$*$ or only $*$ domains, and design non-$*$ and $*$ domains to have distinct sequence properties (for example, using a three-letter code [28]). Then $X(t, s; m)$ unambiguously describes the top reactant of Figure 1 (d), with s understood to mean an open s^* toehold. With that assumption, the top product in Figure 1 (d) would be $X(m, n; s)^*$, with the first toehold listed still being on the 3' end of its strand, but now understood to mean an open m^* toehold. Without that assumption, we might use a more general notation where those molecules are $X(t, s^*; m)$ and $X^*(m^*, n; s^*)$ respectively. The circle abstraction shown in said figures is also useful to illustrate strand exchange reactions. Each circle represents a strand with one long domain and two toeholds, where half-faded circles represent strands made of $*$ domains. Thin connections (both figures) represent strands bonded directly, requiring matching domains; thick connections labelled with a toehold domain (horizontal in Figure 2) represent strands connected by gate strands from a cooperative 4-way strand exchange reaction, which can be between any domains so long as the appropriate gate exists.

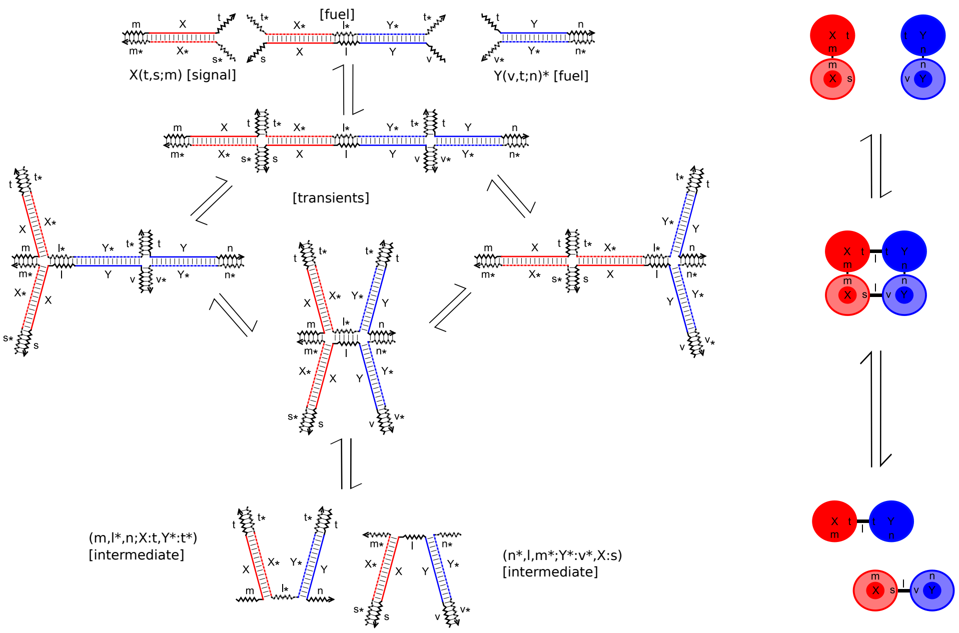

Figure 2 A cooperative 4-way branch migration mechanism. Initial X and Y complexes combine with a gate that matches their open toehold combinations, producing two 3-stranded complexes each with one of the strands of X and one of the strands of Y. These complexes can recombine with each other or with the corresponding products of a similar reaction, which in the latter case will produce X and Y complexes with different toehold combinations. On the right, this reaction is shown in abstracted form. The cooperative 4-way CRN is based on groups of four of these reactions, two in the reverse of the direction shown, where in the reverse reactions each product of one forward reaction interacts with the corresponding product of the other forward reaction. Complexes are labeled with names in the abstract notation if applicable, and their role in the cooperative 4-way CRN implementation scheme. "Signal" and "fuel" complexes have 2 strands as desired; stable "intermediate" complexes can have any number of strands; and "transient" complexes will quickly decay to one side or the other of the reaction. The marking of $X(t,s,m)$ as signal and $Y(v,t,n)^*$ as fuel is based on the CRN implementation scheme presented in Section 3, but in general the two can be any combination of signal and fuel, or could be intermediates of a more complex pathway.

In Figure 2 we introduce a similar notation for the "intermediate" products of a cooperative 4-way strand exchange reaction, in that case $(m,l^*,n;X:t,Y^*:t^*)$ and $(n^*,l,m^*;Y^*:v^*,X:s)$. Again the semicolon distinguishes the three open toeholds, listed from 5' to 3' end, from the bound long domain-toehold pairs; each of those pairs is listed as the domains that appear first in 5' to 3' order. Thus the full reaction is

$$X(t,s;m) + Y(v,t,n)^* \rightleftharpoons (m,l^*,n;X:t,Y^*:t^*) + (n^*,l,m^*;Y^*:v^*,X:s)$$

assuming the appropriate fuel (top center), which we do not give a notation to and omit from the reaction, is present.

3 Chemical Reaction Network implementations

The above motifs can be combined in various ways to construct implementations of arbitrary Chemical Reaction Networks. To implement arbitrary CRNs, the reaction $A+B \rightarrow C+D$ (or $A+B \rightarrow C$ and $A \rightarrow B+C$) is sufficient; for arbitrary reversible CRNs, the reaction $A+B \rightleftharpoons C$ (or *a fortiori*, $A+B \rightleftharpoons C+D$) is sufficient [26]. From a logical perspective, "join" and "fork" operations are sufficient; the above reactions represent those logics.

We take modular CRN bisimulation [19] as the definition of a "correct" CRN implementation scheme. Given that a scheme is correct, there are a number of other conditions that would be useful to satisfy for various reasons, theoretical and practical. CRN implementations typically have *signal* complexes that are the primary form of a given formal species, and *fuel* complexes that are assumed to be always present and drive the reactions. For a CRN to have "only 2-stranded inputs", as desired in this work, means that all signal complexes and fuel complexes are single strands or 2-stranded. We implicitly assume that we are discussing *systematic* CRN implementations, where we give a template for a generic reaction and construct larger CRNs by combining independent copies of the template with different domain identities. In such a case we can ask how the number of toehold domains scales, i.e. whether different reactions can use the same toeholds or have to create new ones; as toeholds are limited in length by thermodynamics, a system with $O(n)$ toeholds may be able to implement small CRNs but a system with $O(1)$ toeholds is better if possible. Whether a scheme requires cooperative mechanisms is worth noting. Finally, it is desirable for reversible reactions ($A + B \rightleftharpoons C + D$) to be implemented with physically reversible mechanisms, so that going forward and backward multiple times does not consume fuel; to be truly reversible, the 2-stranded fuel criterion should include the reverse fuels as well. For further discussion and formal definitions of these criteria, see [18], which also contains a proof that no CRN implementation scheme using only 4-way branch migration can satisfy all of them.

Toehold Exchange-based CRNs

Existing CRN implementations [34, 27, 4] are often based on toehold exchange mechanisms where e.g. $A + B \to C$ is implemented by a toehold exchange reaction with A opening a toehold on the gate for a reaction involving B. These schemes can be understood in light of the motifs previously discussed: the property of toehold exchange that a different toehold on the gate is opened allows join and fork logic. The property that the released strand has a different long domain/toehold combination is used to pass signals between gates. The same shared-toehold logic could also be used with 4-way branch migration instead of toehold exchange, similar to the 4-way-based AND gate [16] (although that gate itself uses a toehold hidden in a loop rather than a toehold shared between adjacent long domains, which is a line of investigation to be explored elsewhere).

Such a shared-toehold mechanism seems to require a 3-stranded complex for the gate molecule to achieve join logic, so it does not meet the goal of this paper, but is worth mentioning as the current state of the art. Another relevant mechanism using toehold exchange is the seesaw gate [28], where transduction logic combines with threshold logic to check whether the total amount of signal is more than either A or B can produce by itself. This achieves join logic for macroscopic signals but cannot satisfy criteria such as CRN bisimulation for individual molecules.

3-way Cooperative CRNs

The symmetric cooperative hybridization is $A + B \rightleftharpoons C + D$ logic, if we consider the same long domain in a different toehold context to be a different signal. Since toehold exchange reactions depend on the combination of long domain and toehold, this is valid. Thachuk et al. use a combination of symmetric cooperative hybridization and toehold exchange to implement leakless $A + B \to C + D$ reactions in exactly this manner [38, 39, 42]).

From our perspective, the only problem is that symmetric cooperative hybridization with 1-stranded inputs produces 2-stranded products, and toehold exchange with a 2-stranded input signal produces a 3-stranded reverse gate. For physically reversible reactions, this

Table 1 List of species for the 4-way $O(n)$-toeholds reaction $A + B \rightleftharpoons C + D$, in the abstracted notation. Species in columns A, B, C, and D represent the given formal species. Species in columns labeled $\emptyset$ are fuels and assumed to be always present. a_i domains are toeholds specific to species A, and similarly for B, C, and D; r_i domains are specific to the reaction $A + B \rightleftharpoons C + D$; this ensures no crosstalk with other pathways.

A	$\emptyset$	B	$\emptyset$
$(a_1, a_2; a_3)$	$(a_2, a_1; r_5)$	$(b_1, b_2; b_3)$	$(b_2, b_1; r_6)$
$(r_5, a_3; a_1)^*$	$(a_3, r_5; a_2)^*$ $(a_3, r_5; r_2)^*$	$(r_6, b_3; b_1)^*$	$(b_3, r_6; b_2)^*$ $(b_3, r_6; r_1)^*$
$(r_2, a_1; r_5)$	$(a_1, r_2; a_3)$ $(a_1, r_2; r_3)$	$(r_1, b_1; r_6)$	$(b_1, r_1; b_3)$ $(b_1, r_1; r_4)$
$(r_3, r_5; r_2)^*$	$(r_5, r_3; a_1)^*$ $(r_5, r_3; r_1)^*$	$(r_4, r_6; r_1)^*$	$(r_6, r_4; b_1)^*$ $(r_6, r_4; r_2)^*$
$(r_1, r_2; r_3)$	$(r_2, r_1; r_5)$	$(r_2, r_1; r_4)$	$(r_1, r_2; r_6)$

C	$\emptyset$	D	$\emptyset$
$(c_1, c_2; c_3)$	$(c_2, c_1; r_3)$	$(d_1, d_2; d_3)$	$(d_2, d_1; r_4)$
$(c_3, r_3; c_2)^*$	$(r_3, c_3; c_1)^*$ $(r_3, c_3; r_2)^*$	$(d_3, r_4; d_2)^*$	$(r_4, d_3; d_1)^*$ $(r_4, d_3; r_1)^*$
$(c_2, r_2; r_3)$	$(r_2, c_2; c_3)$ $(r_2, c_2; r_4)$	$(d_2, r_1; r_4)$	$(r_1, d_2; d_3)$ $(r_1, d_2; r_3)$
$(r_3, r_4; r_2)^*$	$(r_4, r_3; c_2)^*$	$(r_4, r_3; r_1)^*$	$(r_3, r_4; d_2)^*$

3-stranded gate would be considered a reverse fuel, and the system would not be made with entirely 2-stranded fuels. Thus this mechanism meets all our criteria for irreversible CRNs, but not reversible CRNs.

4-way-based CRNs with $O(n)$ toeholds

The two-toehold-mediated 4-way strand exchange mechanism effectively exchanges toeholds on a common long domain; note that while the inputs both have t and s toeholds, the outputs have one with only t and one with only s. When a signal complex goes through multiple copies of this reaction with different fuels, it can turn any combination of toeholds into any other combination. When two signals with complementary pairs of toeholds meet in this reaction, it produces two signals with different combinations in $A + B \rightleftharpoons C + D$ logic. So for example, we can turn $(a_1, a_2; a_3)$ into $(r_1, r_2; r_3)$ and $(b_1, b_2; b_3)$ into $(r_2, r_1; r_4)$, which will react and produce $(r_3, r_4; r_2)^*$ and $(r_4, r_3; r_1)^*$, which can be turned into $(c_1, c_2; c_3)$ and $(d_1, d_2; d_3)$ respectively. Thus two-toehold-mediated 4-way strand exchange alone can implement arbitrary reversible CRNs if we allow $O(n)$ toeholds.

A list of all species involved is given in Table 1. Note that fuels $(r_2, r_1; r_5)$ and $(r_1, r_2; r_6)$ can interact, but the products can do nothing but reverse the reaction, and the same is true for $(r_4, r_3; c_2)^*$ with $(r_3, r_4; d_2)^*$.

4-way Cooperative CRNs

The cooperative 4-way strand exchange motif in Figure 2, when its products recombine with products of a different instance of the reaction, *simultaneously* exchanges the toehold combinations on a complex with long domain X and a complex with long domain Y. If

$A(t,s;m)$ is the signal molecule for A, then simultaneously breaking the (t,s) combination on A and putting together a (u,v) combination on some long domain R is effectively converting $A(t,s;m) \rightleftharpoons R(v,u;n)^*$ if all other molecules involved are considered fuels. Where R is unique to the reaction $A+B \rightleftharpoons C+D$, we can convert the four signal species from their own long domains to the R domain, then use a two-toehold-mediated 4-way strand exchange reaction to implement the reaction itself. In contrast to the previous implementation scheme, that each reaction has a different long domain allows the toeholds (u, v, etc.) to be universal, using $O(1)$ toeholds at the expense of requiring cooperative hybridization. In the notation used in Figure 2, this quadruplet of reactions (with the appropriate top-center fuels assumed present but not written) is

$$\begin{aligned}
&A(t,s;m) + R(s,u;n)^* \rightleftharpoons (m,l^*,n;A:t,R^*:u^*) + (n^*,l,m^*:R^*:s^*,A:s)\\
&A(u,v;m) + R(v,t;n)^* \rightleftharpoons (m,l^*,n;A:u,R^*:t^*) + (n^*,l,m^*:R^*:v^*,A:v)\\
&(m,l^*,n;A:u,R^*:t^*) + (n^*,l,m^*:R^*:s^*,A:s) \rightleftharpoons A(u,s;m) + R(s,t;n)^*\\
&(m,l^*,n;A:t,R^*:u^*) + (n^*,l,m^*:R^*:v^*,A:v) \rightleftharpoons A(t,v;m) + R(v,u;n)^*
\end{aligned}$$

where $A(t,s;m)$ and $R(v,u;n)^*$ are the designated meaningful complexes. The other 2-stranded complexes – $A(u,v;m)$, $A(u,s;m)$, $A(t,v;m)$, $R(s,u;n)^*$, $R(v,t;n)^*$, and $R(s,t;n)^*$ are treated as fuels and assumed always present. If this motif works as hypothesized and without leak, $R(v,u;n)^*$ can only be produced by consuming $A(t,s;m)$ and vice versa.

As this scheme is based on the $O(n)$-toehold scheme, we reuse the mechanism from Table 1. Assume all complexes in that list have long domain R, unique to the reaction $A+B \rightleftharpoons C+D$. To the toeholds listed, add toeholds t,s,m,n,l, and let $a_3 = b_3 = c_3 = d_3 = n^*$, with u and v in the above quadruplet renamed appropriately. Then use cooperative 4-way strand exchange to convert $A(t,s;m) \rightleftharpoons (R^*(a_1^*,a_2^*;n))^* = R(a_1,a_2;n^*)$ (the fuel will have R^* on the "top" strand with A), $B(t,s;m) \rightleftharpoons R(b_1,b_2;n^*)$, $C(t,s;m) \rightleftharpoons R(c_1,c_2;n^*)$, and $D(t,s;m) \rightleftharpoons R(d_1,d_2;n^*)$. This gives a mechanism with one long domain per species, one long domain per reaction, and a total of 19 toeholds. Because the long domains now indicate species/reaction identity, the toeholds can be shared between all species and reactions without crosstalk.

4 Correctness of the schemes

The correctness of the schemes can be verified by CRN bisimulation, a formal definition of correctness of a CRN implementation that implies several desirable properties [19]. Below we give an intuitive explanation of why the schemes are correct that parallels the definition of CRN bisimulation; readers familiar with CRN bisimulation can fill in the details of the formal proof. Intuitively, CRN bisimulation consists of *interpreting* each DNA complex as zero or more formal species, then confirming that the behavior of the formal system and the interpreted DSD system are the same from any initial state. That is to say, any reaction of the DNA complexes should be interpreted as a reaction of formal species that is either valid or trivial ("anything that can happen, should"), and any reaction of the formal interpretation of a set of DNA complexes should be possible, perhaps after some trivial reactions, starting from that set of DNA complexes ("anything that should happen, can").

Table 1 is effectively a proof of the correctness of the $O(n)$-toehold 4-way-based scheme according to CRN bisimulation [19]. For each $A+B \rightleftharpoons C+D$ reaction, construct a copy of this mechanism with unique r_i domains, but any a_i domains in common with other reactions using the same formal species; reactions with fewer reactants or products can have one of A, B, C, or D as a fuel; reactions with more reactants or products should be broken into steps with at most 2 of each [26]. DNA complexes in columns labeled A, B, C, or D are interpreted as one copy of the corresponding species, while complexes in columns labeled $\emptyset$ are fuels. Formally, fuels are assumed always present and removed from the enumerated implementation CRN before bisimulation verification; so for example the physical pathway $(r_2, a_2; r_3) + (a_2, r_2; r_5) \rightleftharpoons (r_5, r_3; r_2)^* + (r_3, r_5; a_2)^*$ would be represented as $(r_2, a_2; r_3) \rightleftharpoons (r_5, r_3; r_2)^*$, and then interpreted as the trivial reaction $A \rightleftharpoons A$. Using the abstraction for 4-way strand exchange notation, the table is structured such that each non-fuel species can interact with the (usually two) fuel species in the same row, producing the corresponding fuel+non-fuel pair above or below it; that the final $A + B$ forms react to produce the final $C + D$ forms, while their fuels also have a spurious-but-harmless reaction with each other; and that, given the uniqueness of the domains, no other intra-module or inter-module reactions exist. In CRN bisimulation, we say that a reaction interpreted as, for example, $A \rightleftharpoons A$ is "trivial", and in this case all reactions are trivial except $(r_1, r_2; r_3) + (r_2, r_1; r_4) \rightleftharpoons (r_3, r_4; r_2)^* + (r_4, r_3; r_1)^*$ which is interpreted as the desired reaction $A + B \rightleftharpoons C + D$. With $(a_1, a_2; a_3)$ etc. as the signal species, one can see that the signal species can implement the formal reaction, and any intermediate species can turn into the common species with the same interpretation by interacting with only fuels. Intuitively this is a good argument for correctness, and readers familiar with CRN bisimulation will recognize the above as a sufficient condition for modular CRN bisimulation with respect to the signal species as common species.

For the cooperative 4-way scheme, the same bisimulation logic applies. In the notation used in Figure 2 and Section 3, in e.g. $A(t, s; m) \rightleftharpoons R(a_1, a_2, n^*)$ the signal complex $A(t, s; m)$, output complex $R(a_1, a_2, n^*)$, and intermediate $(m, l^*, n; A : t, R : a_2)$ all interpreted as A, while the other three intermediates and all the fuels will each be interpreted as nothing. From there the bisimulation proof follows the $O(n)$-toeholds case. In this case the lack of crosstalk between modules is assured by the distinct long domains; even if toehold combinations are identical, different long domains will make the reaction unproductive. The remaining caveat is with the cooperative 4-way mechanism itself. We designed the system so that the toeholds along the cooperative reaction are *always* m, l, n. Thus, we *assume* that intermediates of the cooperative pathway will all have the matching m, l, n toeholds, and all three toeholds will bind and dissociate as a unit. Whether this is actually true or not will be determined experimentally; if not, there may be problematic crosstalk between, for example, an (A, R_1) and (A, R_2) pair of long domains which leads to temporarily duplicated signals. If it is true, however, then the result of such a crosstalk will be a release of one side with the other suspended, one of which carries the signal, and the system will be correct according to bisimulation.

5 Discussion

We discussed the use of DNA Strand Displacement to implement Chemical Reaction Networks, and the desire to create larger, more robust DSD CRN implementations. We then presented 2-stranded DSD motifs which we used to build 2-stranded CRN implementations, in the hope that they would be more robust than those which rely on 3-or-more-stranded complexes. There is some indication that 2-stranded DSD systems in general are more robust (as we briefly reviewed in the introduction), but whether these particular systems are more robust than the current state-of-the-art CRN implementations is an open question.

We can compare Soloveichik et al.'s original CRN scheme [34, 36] (which is reasonably representative of other toehold exchange schemes), our $O(n)$-toehold 4-way strand exchange scheme, and our ($O(1)$-toehold) cooperative 4-way strand exchange scheme. While 3- and

4-stranded complexes may be less robust, in other aspects the toehold exchange scheme is simpler than our two schemes: it uses one long domain per formal species, one long domain per reaction, and can be done with a single, universal toehold. To go from reactant signal species to product signal species in the toehold exchange scheme (as implemented experimentally [36]) takes 4 toehold exchange steps in an $A+B \to C+D$ reaction, and generalizes naturally to $n+m$ steps in an n-reactant m-product reaction. In contrast, while the cooperative 4-way scheme also uses one long domain per formal species and reaction, as described above it uses 19 universal toeholds and takes 30 reactions for $A+B \to C+D$. (By "reaction" we mean roughly one condensed reaction as described in Peppercorn [17], generalized to include trimolecular reactions. So one toehold exchange or one 2-toehold-mediated 4-way strand exchange is one reaction, as is the cooperative 4-way strand exchange shown in Figure 2; note that using that mechanism to exchange e.g. $A(t,s;m) \rightleftharpoons R(a_1,a_2;n^*)$ takes 4 such reactions.) The $O(n)$-toeholds scheme takes only 14 reactions for $A+B \to C+D$, but with one universal long domain it takes 3 toeholds per species and 6 per reaction, which may run out of design space for large CRNs. Also, 14 reactions is still much more than 4. These pathways are not provably optimal; we suspect they can be reduced to less than 14 and 30, but still more than 4.

The increase in number of reactions to implement $A+B \to C+D$ may just be a cost of using 2-stranded complexes. The fundamental question is, given a complex of a certain size, how much information can it store? How can complexes meant to represent A, C, and an E from another reaction all present different enough open and bound domains that none can undergo a reaction meant for a different one? With 3-stranded complexes and toehold exchange, the long domain identity and open toehold does this very efficiently. With 2-stranded complexes and 4-way strand exchange, we use pairs of toehold identity to represent signal identity, which means we need extra reactions to (a) change the toehold identity one strand at a time, and (b) ensure that intermediates of different pathways don't try to pass through the same toehold combination.

This question, then, connects to another work of ours. The final result of that work was a proof that a systematic CRN implementation that satisfies certain desirable conditions, including using only 2-stranded inputs and the other conditions discussed at the beginning of Section 3, cannot be done with DSD using only 4-way branch migration [18]. The steps taken to prove that result involve questions of what sort of transformations are possible with DSD reactions, and how and whether the possibility of certain transformations can depend on the features of the strands. This "dependence" is in the sense that the release of a strand in toehold exchange "depends on" the incoming strand having the correct toehold and long domain identities, or the way we have to structure our CRN implementations so that production of the output species depends on the inputs having the correct toehold identity pairs. Thus, further exploration of that line of investigation might help answer some of the questions suggested by the mechanisms in this paper, of whether 2-stranded complex based CRN implementations inherently require longer pathways, and quantitatively how much longer. Moreover, the investigation could be expanded to include other CRN implementations involving enzymes. For example, transcriptional circuits [20, 21], PEN-DNA toolbox [23, 1], primer exchange reaction cascades [22], and strand-displacing polymerase systems [35, 32, 33] all have elementary reactions that can be abstracted as motifs and are candidates for formal analysis. In these systems, it is possible to start with fewer and simpler fuel molecules (e.g. single strands only) while more complex molecules can be generated by DNA polymerase to carry out desired reactions. In addition to 3-way and 4-way strand displacement with standard toeholds, other mechanisms could also be investigated, including remote [15], associative [6], and allosteric [43] toeholds. These mechanisms may allow further simplification of the implementations as they enrich the design space with alternative representations of signals.

It is also worth discussing how we discovered the cooperative 4-way strand exchange motif and associated CRN implementation in the process of working out the impossibility proof in [18]. We give an intuitive list of those conditions at the beginning of Section 3, but readers desiring a formal list of conditions should see [18]. Two of the conditions are using only $O(1)$ toeholds and not using cooperative mechanisms, so both the $O(n)$ toeholds uncooperative 4-way strand exchange based scheme and the $O(1)$ toeholds cooperative 4-way strand exchange based scheme satisfy all but one of the conditions, each failing to satisfy a different one. Thus in some sense this paper is the positive counterpart to the previous negative result, forming a tight upper and lower bound on the complexity of DSD implementations of CRNs. But this pair of results also has implications for design of DSD systems. The cooperative 4-way strand exchange motif and the process by which we came up with it is potentially a proof of concept that, in systematically eliminating possibilities in DSD systems, we can find new motifs in whatever remains. How exactly this can be generalized we do not know, but if it can be, it may make the process of designing DSD systems faster and more systematic.

Another aspect worth mentioning is the focus on motifs before building up CRN implementations. We argued that each of the 5 motifs has certain abstract behaviors, and that larger systems such as CRN implementations can be thought of in terms of those behaviors. When building large systems, it is much easier if one can build mid-sized building blocks out of the fundamental units, then build larger systems out of the mid-sized building blocks. Motifs take that role between fundamental DSD steps (bind, unbind, 3-way branch migration, 4-way branch migration) and systems on the scale of CRN implementations. To the extent that we were able to describe our CRN implementations in terms of the motifs rather than in terms of the underlying DSD steps, this approach should be considered for future DSD system design.

References

1 Nathanaël Aubert, Clément Mosca, Teruo Fujii, Masami Hagiya, and Yannick Rondelez. Computer-assisted design for scaling up systems based on DNA reaction networks. *Journal of The Royal Society Interface*, 11(93):20131167, 2014.

2 Charles H Bennett. Logical reversibility of computation. *IBM journal of Research and Development*, 17(6):525–532, 1973.

3 Charles H Bennett. The thermodynamics of computation—a review. *International Journal of Theoretical Physics*, 21(12):905–940, 1982.

4 Luca Cardelli. Two-domain DNA strand displacement. *Mathematical Structures in Computer Science*, 23(02):247–271, 2013.

5 Sherry Xi Chen, David Yu Zhang, and Georg Seelig. Conditionally fluorescent molecular probes for detecting single base changes in double-stranded DNA. *Nature Chemistry*, 5(9):782, 2013.

6 Xi Chen. Expanding the rule set of DNA circuitry with associative toehold activation. *Journal of the American Chemical Society*, 134(1):263–271, 2012.

7 Yuan-Jyue Chen, Neil Dalchau, Niranjan Srinivas, Andrew Phillips, Luca Cardelli, David Soloveichik, and Georg Seelig. Programmable chemical controllers made from DNA. *Nature Nanotechnology*, 8(10):755–762, 2013.

8 Kevin M Cherry and Lulu Qian. Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature*, 559(7714):370, 2018.

9 Anne Condon, Alan J Hu, Ján Maňuch, and Chris Thachuk. Less haste, less waste: on recycling and its limits in strand displacement systems. *Interface Focus*, 2(4):512–521, 2012.

10 Nadine L Dabby. *Synthetic molecular machines for active self-assembly: prototype algorithms, designs, and experimental study.* PhD thesis, California Institute of Technology, February 2013.

11 Robert M Dirks and Niles A Pierce. Triggered amplification by hybridization chain reaction. *Proceedings of the National Academy of Sciences*, 101(43):15275–15278, 2004.

12 Abeer Eshra, Shalin Shah, Tianqi Song, and John Reif. Renewable DNA hairpin-based logic circuits. *IEEE Transactions on Nanotechnology*, 18:252–259, 2019.

13 Sudhanshu Garg, Shalin Shah, Hieu Bui, Tianqi Song, Reem Mokhtar, and John Reif. Renewable time-responsive DNA circuits. *Small*, 14(33):1801470, 2018.

14 Anthony J Genot, Jonathan Bath, and Andrew J Turberfield. Reversible logic circuits made of DNA. *Journal of the American Chemical Society*, 133(50):20080–20083, 2011.

15 Anthony J Genot, David Yu Zhang, Jonathan Bath, and Andrew J Turberfield. Remote toehold: a mechanism for flexible control of DNA hybridization kinetics. *Journal of the American Chemical Society*, 133(7):2177–2182, 2011.

16 Benjamin Groves, Yuan-Jyue Chen, Chiara Zurla, Sergii Pochekailov, Jonathan L Kirschman, Philip J Santangelo, and Georg Seelig. Computing in mammalian cells with nucleic acid strand exchange. *Nature Nanotechnology*, 11(3):287, 2016.

17 Casey Grun, Karthik Sarma, Brian Wolfe, Seung Woo Shin, and Erik Winfree. A domain-level DNA strand displacement reaction enumerator allowing arbitrary non-pseudoknotted secondary structures. *CoRR*, 2015. URL: `http://arxiv.org/abs/1505.03738`, `arXiv:1505.03738`.

18 Robert F. Johnson. Impossibility of sufficiently simple chemical reaction network implementations in DNA strand displacement. In Ian McQuillan and Shinnosuke Seki, editors, *Unconventional Computation and Natural Computation*, pages 136–149. Springer International Publishing, 2019. `doi:10.1007/978-3-030-19311-9_12`.

19 Robert F Johnson, Qing Dong, and Erik Winfree. Verifying chemical reaction network implementations: A bisimulation approach. *Theoretical Computer Science*, 2018. `doi:10.1016/j.tcs.2018.01.002`.

20 Jongmin Kim, John Hopfield, and Erik Winfree. Neural network computation by in vitro transcriptional circuits. In *Advances in Neural Information Processing systems*, pages 681–688, 2005.

21 Jongmin Kim and Erik Winfree. Synthetic in vitro transcriptional oscillators. *Molecular Systems Biology*, 7(1):465, 2011.

22 Jocelyn Y Kishi, Thomas E Schaus, Nikhil Gopalkrishnan, Feng Xuan, and Peng Yin. Programmable autonomous synthesis of single-stranded DNA. *Nature Chemistry*, 10(2):155, 2018.

23 Kevin Montagne, Raphael Plasson, Yasuyuki Sakai, Teruo Fujii, and Yannick Rondelez. Programming an in vitro DNA oscillator using a molecular networking strategy. *Molecular Systems Biology*, 7(1):466, 2011.

24 Richard A Muscat, Jonathan Bath, and Andrew J Turberfield. A programmable molecular robot. *Nano letters*, 11(3):982–987, 2011.

25 Igor G Panyutin and Peggy Hsieh. The kinetics of spontaneous DNA branch migration. *Proceedings of the National Academy of Sciences*, 91(6):2021–2025, 1994.

26 Tomislav Plesa. Stochastic approximation of high-molecular by bi-molecular reactions. *arXiv preprint arXiv:1811.02766*, 2018.

27 Lulu Qian, David Soloveichik, and Erik Winfree. Efficient Turing-universal computation with DNA polymers. In Yasubumi Sakakibara and Yongli Mi, editors, *DNA Computing and Molecular Programming*, volume 6518 of Lecture Notes in Computer Science, pages 123–140. Springer, 2011.

28 Lulu Qian and Erik Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332(6034):1196–1201, 2011.

29 Dominic Scalise, Nisita Dutta, and Rebecca Schulman. DNA strand buffers. *Journal of the American Chemical Society*, 140(38):12069–12076, 2018.

30 Dominic Scalise and Rebecca Schulman. Designing modular reaction-diffusion programs for complex pattern formation. *Technology*, 2(01):55–66, 2014.

31 Dominic Scalise and Rebecca Schulman. Emulating cellular automata in chemical reaction-diffusion networks. *Natural Computing*, 15(2):197–214, 2016.

32 Shalin Shah, Tianqi Song, Xin Song, Ming Yang, and John Reif. Implementing arbitrary CRNs using strand displacing polymerase. In *International Conference on DNA Computing and Molecular Programming*, pages 21–36. Springer, 2019.

33 Shalin Shah, Jasmine Wee, Tianqi Song, Luis Ceze, Karin Strauss, Yuan-Jyue Chen, and John Reif. Using strand displacing polymerase to program chemical reaction networks. *Journal of the American Chemical Society*, 2020.

34 David Soloveichik, Georg Seelig, and Erik Winfree. DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107(12):5393–5398, 2010.

35 Tianqi Song, Abeer Eshra, Shalin Shah, Hieu Bui, Daniel Fu, Ming Yang, Reem Mokhtar, and John Reif. Fast and compact DNA logic circuits based on single-stranded gates using strand-displacing polymerase. *Nature Nanotechnology*, 14(11):1075–1081, 2019.

36 Niranjan Srinivas, James Parkin, Georg Seelig, Erik Winfree, and David Soloveichik. Enzyme-free nucleic acid dynamical systems. *Science*, 358:doi:10.1126/science.aal2052, 2017.

37 Chris Thachuk and Anne Condon. Space and energy efficient computation with DNA strand displacement systems. In *International Workshop on DNA-Based Computers*, pages 135–149. Springer, 2012.

38 Chris Thachuk and Erik Winfree. A fast, robust, and reconfigurable molecular circuit breadboard. 15th Annual Conference on Foundations of Nanoscience, invited talk, 2018. URL: `https://thachuk.com/talk/2018-fnano-invited/2018-FNANO-invited.pdf`.

39 Chris Thachuk, Erik Winfree, and David Soloveichik. Leakless DNA strand displacement systems. In Andrew Phillips and Peng Yin, editors, *DNA Computing and Molecular Programming*, volume 9211 of Lecture Notes in Computer Science, pages 133–153. Springer, 2015.

40 Anupama J Thubagere, Chris Thachuk, Joseph Berleant, Robert F Johnson, Diana A Ardelean, Kevin M Cherry, and Lulu Qian. Compiler-aided systematic construction of large-scale DNA strand displacement circuits using unpurified components. *Nature Communications*, 8:14373, 2017.

41 Suvir Venkataraman, Robert M Dirks, Paul WK Rothemund, Erik Winfree, and Niles A Pierce. An autonomous polymerization motor powered by DNA hybridization. *Nature Nanotechnology*, 2(8):490, 2007.

42 Boya Wang, Chris Thachuk, Andrew D Ellington, Erik Winfree, and David Soloveichik. Effective design principles for leakless strand displacement systems. *Proceedings of the National Academy of Sciences*, 115(52):E12182–E12191, 2018.

43 Xiaolong Yang, Yanan Tang, Sarah M Traynor, and Feng Li. Regulation of DNA strand displacement using an allosteric DNA toehold. *Journal of the American Chemical Society*, 138(42):14076–14082, 2016.

44 Peng Yin, Harry MT Choi, Colby R Calvert, and Niles A Pierce. Programming biomolecular self-assembly pathways. *Nature*, 451(7176):318–322, 2008.

45 Bernard Yurke and Allen P Mills. Using DNA to power nanostructures. *Genetic Programming and Evolvable Machines*, 4(2):111–122, 2003.

46 David Yu Zhang. Cooperative hybridization of oligonucleotides. *Journal of the American Chemical Society*, 133(4):1077–1086, 2010.

47 David Yu Zhang and Georg Seelig. Dynamic DNA nanotechnology using strand-displacement reactions. *Nature Chemistry*, 3(2):103–113, 2011.

48 David Yu Zhang and Erik Winfree. Control of DNA strand displacement kinetics using toehold exchange. *Journal of the American Chemical Society*, 131(47):17303–17314, 2009.