DNA as a universal substrate for chemical kinetics

David Soloveichik[a,1], Georg Seelig[a,b,1], and Erik Winfree[c,1]

[a]Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195; [b]Department of Electrical Engineering, University of Washington, Seattle, WA 98195; and [c]Departments of Computer Science, Computation and Neural Systems, and Bioengineering, California Institute of Technology, Pasadena, CA 91125

Molecular programming aims to systematically engineer molecular and chemical systems of autonomous function and ever-increasing complexity. A key goal is to develop embedded control circuitry within a chemical system to direct molecular events. Here we show that systems of DNA molecules can be constructed that closely approximate the dynamic behavior of arbitrary systems of coupled chemical reactions. By using strand displacement reactions as a primitive, we construct reaction cascades with effectively unimolecular and bimolecular kinetics. Our construction allows individual reactions to be coupled in arbitrary ways such that reactants can participate in multiple reactions simultaneously, reproducing the desired dynamical properties. Thus arbitrary systems of chemical equations can be compiled into real chemical systems. We illustrate our method on the Lotka–Volterra oscillator, a limit-cycle oscillator, a chaotic system, and systems implementing feedback digital logic and algorithmic behavior.

molecular programming | mass-action kinetics | strand displacement cascades | chemical reaction networks | nonlinear chemical dynamics

Chemical reaction equations and mass-action kinetics provide a powerful mathematical language to describe and analyze chemical systems. For well over a century, mass-action kinetics has been used to model chemical experiments and to predict and explain their dynamical properties. Both biological and nonbiological chemical systems can exhibit complex behaviors such as oscillations, memory, logic and feedback control, chaos, and pattern formation—all of which can be explained by the corresponding systems of coupled chemical reactions (1–4). Whereas the use of mass-action kinetics to describe existing chemical systems is well established, the inverse problem of experimentally implementing a given set of chemical reactions has not been considered in full generality. Here, we ask: Given a set of formal chemical reaction equations, involving formal species $X_1, X_2, ..., X_n$, can we find a set of actual molecules $M_1, M_2, ..., M_m$ that interact in an appropriate buffer to approximate the formal system's mass-action kinetics? If this were possible, the formalism of chemical reaction networks (CRNs) could be treated as an effective programming language for the design of complex network behavior (5–9).

Unfortunately, a formally expressed system of coupled chemical equations may not have an obvious realization in known chemistry. In a formal system of chemical reactions, a species may participate in multiple reactions, both as a reactant and/or as a product, and these reactions progress at relative rates determined by the corresponding rate constants, all of which imposes formidable constraints on the chemical properties of the species participating in the reactions. For example, it is likely hard to find a physical implementation of arbitrary chemical reaction equations using small molecules, because small molecules have a limited set of reactivities.

Thus, formal CRNs may appear to be an unforgiving target for general implementation strategies. Indeed, most experimental work in chemical and biological engineering has started with particular molecular systems—genetic regulatory networks (10), RNA folding and processing (11), metabolic pathways (12), signal transduction pathways (13), cell-free enzyme systems (14, 15), and small molecules (16, 17)—and found ways to modify or re-

wire the components to achieve particular functions. Attempts to systematically understand what functional behaviors can be obtained by using such components have targeted connections to analog and digital electronic circuits (10, 18, 19), neural networks (20–22), and computing machines (15, 20, 23, 24); in each case, complex systems are theoretically constructed by composing modular chemical subsystems that carry out key functions, such as boolean logic gates, binary memories, or neural computing units. Despite its apparent difficulty, we directly targeted CRNs for three reasons. First, shoehorning the design of synthetic chemical circuits into familiar but possibly inappropriate computing models may not capture the natural potential and limitations of the chemical substrate. Second, there is a vast literature on the theory of CRNs (25, 26) and even on general methods to implement arbitrary polynomial ordinary differential equations as CRNs (27, 28). Third, as a fundamental model that captures the essential formal structure of chemistry, implementation of CRNs could provide a useful programming paradigm for molecular systems.

Here we propose a method for compiling an arbitrary CRN into nucleic-acid-based chemistry. Given a formal specification of coupled chemical kinetics, we systematically design DNA molecules implementing an approximation of the system scaled to an appropriate temporal and concentration regime. Formal species are identified with certain DNA strands, whose interactions are mediated by a set of auxiliary DNA complexes. Nonconserving CRNs can be implemented because the auxiliary species implicitly supply energy and mass.

Conveniently, the base sequence of nucleic acids can determine reactivity not only through direct hybridization of single-stranded species (29) but also through branch migration and strand displacement reaction pathways (30–32). These powerful reaction primitives have been used previously for designing nucleic-acid-based molecular machines with complex behaviors, such as motors, logic gates, and amplifiers (33–37). Here we use these reaction mechanisms as the basis for the implementation of arbitrary CRNs. Our work advances a systematic approach that aims to provide a general mechanism for implementing a well-specified class of behaviors.

Molecular Primitive: Strand Displacement Cascades

Because simple hybridization reactions cannot be cascaded, we use the more flexible strand displacement reaction as a molecular primitive. [We use "strand displacement" as a shorthand for toehold-mediated branch migration and strand displacement
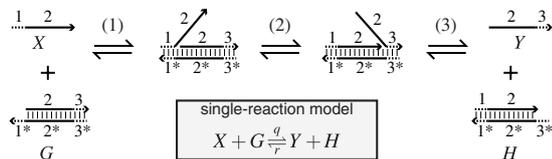
Fig. 1. Strand displacement molecular primitive. Domains are labeled by numbers, with * denoting Watson–Crick complementarity. Multiple elementary steps are indicated: (1) binding of toeholds 1 and 1*; (2) branch migration, a random walk process where domain 2 of strand 2–3 is partially displaced by domain 2 of strand 1–2; (3) the separation of toeholds 3 and 3*. (*Inset*) The single-reaction model of strand displacement.

reactions (33, 38), combined with the principles of toehold sequestering (34) and toehold exchange (35).]

Intuitively, in a strand displacement reaction (Fig. 1) a strand (X) displaces another strand (Y) from a complex (G). In our diagrams, each subsequence that acts as a single functional unit (a domain) is labeled with a unique number (e.g., the two functionally distinct domains of strand X are 1 and 2). We assume strong sequence design such that domain x^* is complementary to domain x and interacts with no other, allowing us to analyze DNA systems entirely at the domain level. Short (dashed) domains are called toeholds. A strand displacement reaction starts when two single-stranded toehold domains (e.g., 1 and 1*) bind each other. Then, a random walk process (branch migration) follows, where two domains with identical sequences compete for the same complementary domain. Domain 2 of strand 2–3 competes with and is partially displaced by domain 2 of strand 1–2. Finally, the initially bound strand 2–3 is released when toeholds 3 and 3* separate. To obtain a fast and reliable reaction, the toehold domains must be short (e.g., 6 nt) whereas the branch migration domains must be long (e.g., 20 nt). Despite the complex mechanism, a strand displacement reaction is well modeled by a single reversible reaction (38, 39) under a large range of experimental conditions where toehold binding is rate limiting (Fig. 1, *Inset*). Removing toehold 3 slows the reverse reaction enough that we can consider the forward reaction effectively irreversible (38, 40).

Strand displacement reactions are programmable by the design of sequences. A single base mismatch significantly impedes branch migration (32, 41). By varying the binding strength of the initiating toeholds, the reaction rate constant can be controlled over 6 orders of magnitude (38, 39). In turn, the length and sequence composition of the toeholds controls their binding strength. The forward and reverse rate constants q and r increase with the hybridization energy of their respective toehold domains (1 to 1* and 3 to 3*). The same strand (X) can react at different rates to different complexes, e.g., G and G'. To attain smaller $q' < q$ for G', we can use a toehold domain $1^*_{q'}$ that is not a full complement of 1. For example, $1^*_{q'}$ can be a truncation of 1* (39) (Fig. S1).

In the following, we construct systems of molecules whose interactions are mediated by strand displacement reactions. To ensure desired reactions and exclude undesired reactions, we use modular components that incorporate three key design principles. First, we design the system so that all long domains are always double-stranded, thus ensuring that toeholds and their complements are the only single-stranded domains capable of hybridizing together. Second, we require that toeholds are short enough that this interaction is fleeting unless it triggers strand displacement. Thus we need to consider only strand displacement, and not hybridization, of long domains. Third, only the desired target must have the correct combination of toehold and displacement regions, preventing any undesired strand displacement reactions. Satisfying these three design principles, which can be verified by inspection of the modules, guarantees that arbitrarily complex systems will function as intended for ideal strand displacement reactions.

Implementing Networks

We implement CRNs in DNA by using three types of modules, which specify the DNA molecules to create for each target reaction (either unimolecular or bimolecular) and for each target species. Initially, we suppose that our target CRN uses rate constants and concentrations that are realistic for aqueous-phase nucleic-acid strand displacement reactions. Additionally, the signal species must always be at much lower concentrations than the auxiliary species. Thus, the feasibility of a CRN depends upon its behavior; for example, we can implement explosive reactions like $X \rightarrow X_1 + X_1$ for only a limited time. If the target CRN operates outside of a physically realistic regime, we will need to relax the goal of implementing the exact target system and instead allow a uniform scaling down of concentrations and/or slowing down of the kinetics. We will show that such scaling can also arbitrarily increase the accuracy of our construction.

Unimolecular Reactions

DNA Implementation. Standardizing the molecular equivalents of the formal species facilitates their assignment to roles as reactants or products in multiple reactions. We implement a formal species as a single-stranded DNA molecule (signal strand) consisting of an inert history domain and a unique species identifier (a long domain flanked by two toehold domains; Fig. 2, orange boxes) that regulates its interactions with other molecules. The history domain depends on the formal reaction producing the species: If X_j is a product of multiple reactions, signal strands with differing history domains but the same species identifier would represent the same formal species X_j. In our scheme, the signal strand is active when fully single-stranded and inactive when bound to another DNA molecule. Signal strands react exclusively by strand displacement reactions initiated at the left toehold of their species identifier; thus, sequestering the left toehold into a double helix is sufficient to inactivate them. The sum of the concentrations of the active signal strands for X_j gives $[X_j]$, and the changes in $[X_j]$ are effected either by the inactivation of a signal strand when it binds to a complementary DNA molecule or by the release of a previously bound strand.

Being composed of entirely distinct domains, DNA signal strands do not interact with each other directly, but rather a set of auxiliary DNA complexes present at large concentrations mediates exactly the desired reactions. Suppose the ith formal reaction is unimolecular, as in Fig. 2. We implement reaction equation 1 as inactivation of a signal strand with species identifier X_1 coupled with activation of strands with identifiers X_2 and X_3. These chemical steps are done through two strand displacement reactions involving two auxiliary complexes G_i and T_i (for "gate" and "translator," respectively) as shown in Fig. 2. A two-step cascade ensures that arbitrary species identifiers can be connected in this reaction because there is no sequence dependence between $X_1, X_2,$ and X_3. To implement other effective net reactions—even catalytic or autocatalytic reactions such as $X_1 \rightarrow X_1 + X_2$ or $X_1 \rightarrow X_1 + X_1$—only the auxiliary complexes need to be modified. A system of multiple coupled unimolecular reactions is implemented simply by starting with the auxiliary complexes for all the desired reactions, along with appropriate initial concentrations of the signal strands. Implementing unimolecular reactions with differing numbers of products requires extending or shrinking the intermediate output O_i. Removing auxiliary complex T_i altogether results in unimolecular decay reactions like $X_1 \rightarrow \emptyset$, where $\emptyset$ indicates the absence of products.

Kinetic Analysis. To approximate ideal unimolecular mass-action kinetics, we require the dynamics of the target reaction network to be slow relative to the fastest strand displacement steps and the concentration of auxiliary DNA species to be much larger than the signal concentrations. Here and in the next section, we present an informal argument that the desired kinetics will be
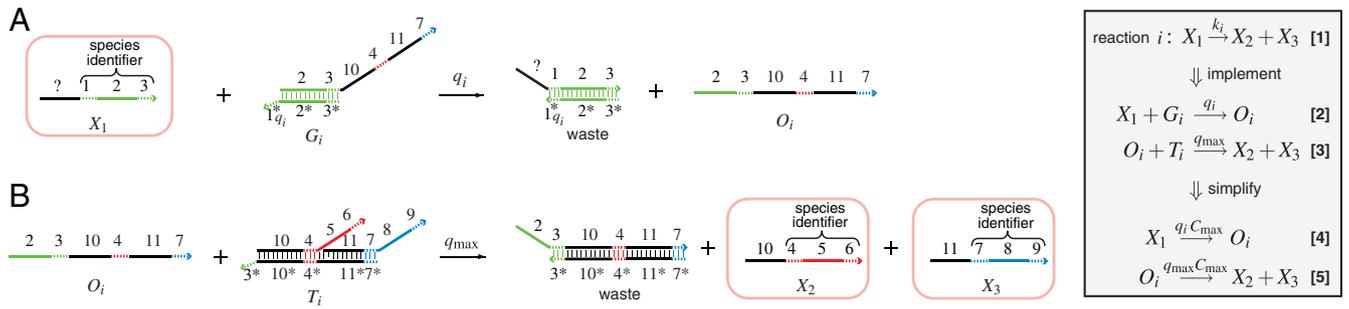
Fig. 2. Unimolecular module: DNA implementation of the formal unimolecular reaction $X_1 \rightarrow X_2 + X_3$ with reaction index i. Orange boxes highlight the DNA species that correspond to the formal species X_1 (species identifier 1-2-3), X_2 (4-5-6), and X_3 (7-8-9). Domains identical or complementary to species identifiers for X_1, X_2, and X_3 are colored red, green, and blue, respectively. Black domains (10 and 11) are unique to this formal reaction. To reduce rate constant q_i, toehold domain $1^*_{q_i}$ may not be a full complement of domain 1. (A) Complex G_i undergoes a strand displacement reaction with strand X_1, with X_1 displacing strand O_i. (B) O_i displaces X_2 and X_3 from complex T_i. Without buffer cancellation, $q_i = k_i C_{max}^{-1}$; with buffer cancellation, $q_i = \gamma^{-1} k_i C_{max}^{-1}$. Reaction equations of type **2–3** are used in simulations (Figs. 5 and 6); simplified reaction equations **4–5** are useful for analysis.

attained under these conditions. In *SI Text* we prove the convergence to the desired kinetics in the limit of high concentration of appropriate auxiliary species relative to the concentrations of the signal species.

For simplicity, we assume all full toehold domains have equal binding strength yielding a maximum strand displacement rate constant q_{max}. Let C_{max} be the starting concentration of auxiliary complexes G_i and T_i. The strand displacement cascade of Fig. 2 follows reaction equations **2** and **3**, where $q_i \leq q_{max}$ is a partial-toehold strand displacement rate constant controlled by the binding energy of domains $1^*_{q_i}$ and 1, and $1^*_{q_i}$ is chosen to obtain the desired rate constant for formal reaction i. Letting τ be the experiment's duration, we assume a regime where $\tau k_i \max([X_1]) \ll C_{max}$, and $k_i \ll q_{max} C_{max}$, and set the partial-toehold strand displacement rate constant $q_i = k_i / C_{max}$. Then over a sufficiently short duration τ, $[G_i]$ and $[T_i]$ remain effectively constant at their initial concentration C_{max}, and the cascade becomes equivalent to a pair of unimolecular reactions (Fig. 2, Eqs. **4** and **5**, where $q_i C_{max} = k_i$). Further, reaction **4** is the rate-limiting step, and we obtain the desired unimolecular kinetics for formal reaction **1**: The instantaneous rate of the consumption of X_1 and the production of X_2 and X_3 in this module is $k_i[X_1]$.

Adding Bimolecular Reactions

DNA Implementation. We now extend the construction to realize bimolecular in addition to unimolecular reactions. Suppose the ith formal reaction is bimolecular, as in Fig. 3. Implementing reaction equation **6**, X_1 must not be consumed in the absence of X_2 and vice versa, which seems difficult because X_1 and X_2 cannot react directly, and therefore one of them (say, X_1) must react with an auxiliary complex first without knowing whether the other is present. The challenge here is to minimize the X_2-independent "drain" on X_1; later we will show how we can exactly compensate for it by rescaling rate constants.

Our construction (Fig. 3) solves this problem by introducing a reversible first step. Strand X_1 reversibly displaces B_i (the "backward" strand) from auxiliary complex L_i, producing activated intermediate H_i. If no X_2 is present, B_i can reverse the reaction and release X_1 back into solution. But if X_2 is present, it can react with intermediate H_i to release intermediate output O_i. As in the unimolecular module, this in turn releases the final output X_3 and makes the overall reaction irreversible. The species identifiers of the reactants and all products can be arbitrarily specified as in the unimolecular modules. Not all B_i are necessarily unique DNA species: If reactions i and i' have the same reactants, then B_i would have the same sequence as $B_{i'}$.

Kinetic Analysis. The kinetics of the module of Fig. 3 is modeled using reaction equations **7–9**. Set the initial concentration of L_i,

B_i, and T_i to C_{max} and partial-toehold rate constant $q_i = k_i$. By assuming $\tau k_i \max([X_1]) \max([X_2]) \ll C_{max}$ and $\max([X_1]) \ll C_{max}$, the concentrations $[L_i]$, $[B_i]$, and $[T_i]$ remain effectively constant throughout the duration of the experiment, and reaction **8** is the rate-limiting step of the pair **8–9**. This system of reactions can then be approximated by reaction equations **10** and **11**. Further, (informally) if in the target reaction network $[X_1]$ varies on a slower time scale than that of the equilibration of reaction **10**, we can assume that X_1 and H_i attain instant equilibrium through reaction **10** with $[X_1]/[H_i] = q_{max}/k_i$. Then the instantaneous rate of reaction **11** is $q_{max}[X_2][H_i] = k_i[X_1][X_2]$. At first glance this appears to give the correct kinetics for the production of X_3; however, X_3 is buffered: Some fraction of it will quickly equilibrate with $H_{i'}$ of other reactions in which X_3 is the first reactant, resulting in a lower effective rate of production of X_3. X_1 and X_2 are likewise buffered. Let $\text{fr}(X_1)$ be the set of bimolecular reactions in which X_1 is the first reactant. Reactions **10** in the different reaction modules create instant equilibrium between all species in the equilibrium set of X_1, $\text{es}(X_1)$, consisting of X_1 and every H_i, $i \in \text{fr}(X_1)$. Producing or consuming a unit of any species in the equilibrium set of X_1 results in the fraction $\gamma_1 \equiv [X_1]/[\text{es}(X_1)]$ of a unit change in $[X_1]$. Thus the effective instantaneous rates of the consumption of X_1, consumption of X_2, and production of X_3 because of reaction **11** in the above module are $\gamma_1 k_i[X_1][X_2]$, $\gamma_2 k_i[X_1][X_2]$, and $\gamma_3 k_i[X_1][X_2]$, respectively.

Making the additional assumption $k_i \ll q_{max}$ shifts equilibria of reactions **10** to the left such that γ_j approaches 1. Then the instantaneous rates of variation in X_1, X_2, and X_3 because of this reaction module approach the desired bimolecular kinetics value of $k_i[X_1][X_2]$.

Canceling out the Buffering Effect. We can exactly cancel out the buffering effect, thereby easing the requirement $k_i \ll q_{max}$ for bimolecular reactions and allowing the implementation of much larger rate constants. Intuitively, because the buffering effect systematically counteracts desired concentration changes of X_j when $\gamma_j < 1$, we should be able to compensate by increasing the rate constants of all reactions involving X_j. Because a reaction between species with different γ_j will behave as if it has incorrect stoichiometry, we want all γ_j to be equal. However, even with equal γ_j, we cannot simply multiply all k_i by γ_j^{-1} because that would in turn change γ_j.

Let $\sigma_j = \Sigma_{i \in \text{fr}(X_j)} k_i$ be the sum of formal rate constants of bimolecular reactions with X_j as the first reactant, and let $\sigma = \max_j\{\sigma_j\}$. We ensure that the fraction γ_j is equal for all X_j by using a new buffering module (Fig. 4) for each X_j for which $\sigma_j < \sigma$. Each buffering module is modeled as reaction equation **12**. Initially $[LS_j] = [BS_j] = C_{max}$, and as for the bimolecular module we can reduce **12** to **13**. The equilibrium set of X_j now
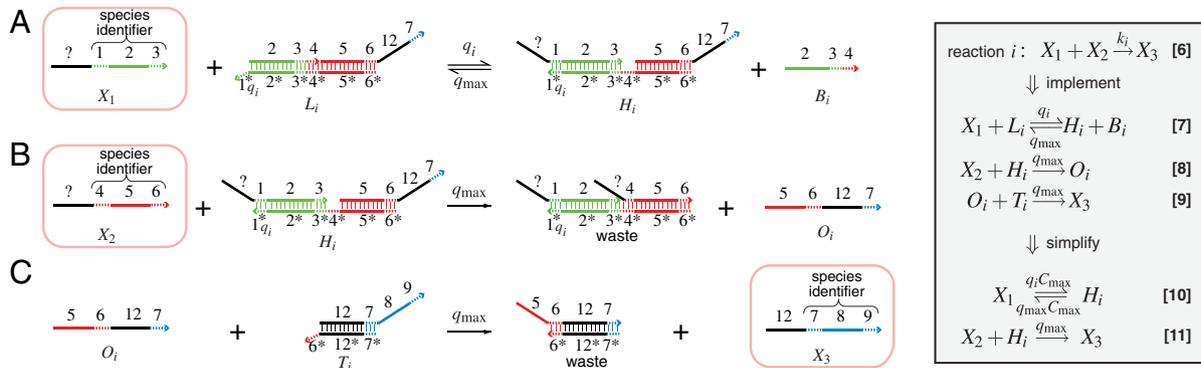
Fig. 3. Bimolecular module: DNA implementation of the formal bimolecular reaction $X_1 + X_2 \rightarrow X_3$ with reaction index i. The black domain (12) is unique to this formal reaction. (*A*) X_1 reversibly displaces B_i from complex L_i producing complex H_i. (*B*) X_2 displaces O_i from complex H_i. Occurrence of reaction *B* precludes the backward reaction of *A*. (*C*) O_i displaces X_3 from complex T_i. Without buffer cancellation, $q_i = k_i$; with buffer cancellation, $q_i = \gamma^{-1}k_i$. Reaction equations of type **7–9** are used in simulations (Figs. 5 and 6); simplified reaction equations **10–11** are useful for analysis.

Reaction boxes:

$$\text{reaction } i: \quad X_1 + X_2 \xrightarrow{k_i} X_3 \quad [6]$$
$$\Downarrow \text{ implement}$$
$$X_1 + L_i \underset{q_{max}}{\overset{q_i}{\rightleftharpoons}} H_i + B_i \quad [7]$$
$$X_2 + H_i \xrightarrow{q_{max}} O_i \quad [8]$$
$$O_i + T_i \xrightarrow{q_{max}} X_3 \quad [9]$$
$$\Downarrow \text{ simplify}$$
$$X_1 \underset{q_{max}C_{max}}{\overset{q_iC_{max}}{\rightleftharpoons}} H_i \quad [10]$$
$$X_2 + H_i \xrightarrow{q_{max}} X_3 \quad [11]$$

includes HS_j. It is not hard to show that with the buffering-scaling factor $\gamma^{-1} = q_{max}(q_{max} - \sigma)^{-1}$, setting $qs_j = \gamma^{-1}(\sigma - \sigma_j)$, and replacing k_i by $k_i' = \gamma^{-1}k_i$ for the formal rate constants when setting q_i in reactions **2** and **7**, we obtain a common buffer fraction $\gamma_j = \gamma$, which is exactly compensated for by the γ^{-1} scaling of formal rate constants. For initial concentrations c_j of formal species X_j, we start with initial concentrations $\gamma^{-1}c_j$ of DNA strands X_j. Then all species in $es(X_j)$ quickly equilibrate yielding $[X_j] = c_j$.

See *SI Text* for a summary of our algorithm for compiling an arbitrary CRN into DNA-based chemistry.

Rescaling for Feasibility and Accuracy

The above procedure will not yield a functional DNA system if the original formal CRN has infeasibly high reaction rates or concentrations. In that case, we scale the system to use lower rate constants and concentrations while maintaining the same, albeit scaled, behavior. If $[X_j](t)$ are solutions to differential equations arising from a set of unimolecular and bimolecular reactions, then $\beta \cdot [X_j](t/\alpha)$ are solutions to the same set of reactions but in which we multiply all unimolecular rate constants by $1/\alpha$ and all bimolecular rate constants by $1/(\alpha \cdot \beta)$. We introduce a mixed concentration-time-scaling parameter δ with $\alpha = \delta$ and $\beta = 1/\delta$, which scales down the concentration and slows down the dynamics by a factor of δ without increasing the largest rate constant.

We justified the accuracy of our construction by assuming the target system operates in a regime with concentrations sufficiently smaller than C_{max}, a physically determined parameter. This may not hold without rescaling, but thankfully, arbitrarily high accuracy for arbitrarily large duration of interest τ can still be attained in a regime of smaller concentrations of formal species and slowed down dynamics. We prove the convergence of the DNA-based kinetics with buffer cancellation to the target CRN in the limit $C_{max} \rightarrow \infty$ by using singular perturbation theory (27, 42) (see *SI Text*). Whereas taking the limit $C_{max} \rightarrow \infty$ is more mathematically convenient, increasing C_{max} by a factor of δ is equivalent, up to scaling in concentration of the experimental implementation, to decreasing all $[X_j]$ by a factor of δ, decreasing

all formal unimolecular rate constants by a factor of δ, and increasing τ by a factor of δ to simulate the same behavior.

Examples

We illustrate our method on the Lotka–Volterra chemical oscillator shown in Fig. 5*A*. The concentration oscillations are in the range of about 0–2. Under typical nucleic-acid experimental conditions, maximal second-order rate constants for strand displacement reactions are about 10^6/M/s, and maximum concentrations are on the order of 10^{-5} M. To fit into this regime with reasonable simulation accuracy and time span, we scale the original system by a time-scaling factor of $\alpha = 300$, and a concentration-scaling factor $\beta = 10^{-8}$ employing units of seconds and molar, and use $C_{max} = 10^{-5}$ M and $q_{max} = 10^6$/M/s. The DNA species for our implementation, including buffering modules, are shown in Fig. S2 and the possible strand displacement reactions in Fig. S3. The equations governing our DNA implementation as derived by using the transforms described in Figs. 2–4 are shown in Fig. 5*B*. Simulations shown in Fig. 5*C* confirm that the DNA implementation nicely approximates the ideal formal chemical system. Because $C_{max} < \infty$, deviations between the DNA implementation and the target system gradually develop, as the depletion of complexes L_1, G_1, and G_2 and the buildup of strands B_1 alter the effective rate constants.

We next apply our construction to more complex systems (Fig. 6). For fastest behavior, all systems are scaled so that the largest q_i or qs_j is $q_{max} = 10^6$/M/s, and $C_{max} = 10^{-5}$ M, leaving only one free scaling parameter δ, which determines both implementation accuracy and the speed of the dynamics. The Oregonator limit-cycle oscillator (Fig. 6*A*) is a simplified model of the Belousov–Zhabotinsky reaction (43). The DNA implementation of this system is relatively slow because of the wide range of rate constants. The chaotic system due to Willamowski and Rössler (44) exhibits complex concentration fluctuations and is particularly sensitive to perturbations at long time scales (Fig. 6*B*). The DNA implementation follows the ideal system relatively well for a few revolutions around the strange attractor. Fig. 6*C* demonstrates the efficacy of our construction for implementing

Fig. 4. Buffering module: DNA implementation of the buffering module used to cancel out the buffering effect. A buffering module is needed for each formal species X_j for which $\sigma_j < \sigma$. The buffering module for species X_2 is shown. X_2 reversibly displaces BS_2 from complex LS_2 to produce complex HS_2 similarly to the first reaction of the bimolecular module. The black domain (13) is unique to this buffering module for species X_2. Set $qs_j = \gamma^{-1}(\sigma - \sigma_j)$. Reaction equations **12** are used in simulations (Figs. 5 and 6); simplified reaction equations **13** are useful for analysis.

$$X_j + LS_j \underset{q_{max}}{\overset{qs_j}{\rightleftharpoons}} HS_j + BS_j \quad [12] \qquad \overset{\text{simplify}}{\Rightarrow} \qquad X_j \underset{q_{max}C_{max}}{\overset{qs_jC_{max}}{\rightleftharpoons}} HS_j \quad [13]$$

A Ideal chemical reactions

1: $X_1 + X_2 \xrightarrow{k_1} 2X_2$

2: $X_1 \xrightarrow{k_2} 2X_1$

3: $X_2 \xrightarrow{k_3} \emptyset$

	unscaled	scaled
k_1	1.5	$5\cdot10^5$ /M/s
k_2	1	1/300 /s
k_3	1	1/300 /s

B DNA reaction modules

$$1\begin{cases} X_1 + L_1 \underset{q_{max}}{\overset{q_1}{\rightleftharpoons}} H_1 + B_1 \\ X_2 + H_1 \xrightarrow{q_{max}} O_1 \\ O_1 + T_1 \xrightarrow{q_{max}} 2X_2 \end{cases}$$

buffering module

$$X_2 + LS_2 \underset{q_{max}}{\overset{qs_2}{\rightleftharpoons}} HS_2 + BS_2$$

$$2\begin{cases} X_1 + G_2 \xrightarrow{q_2} O_2 \\ O_2 + T_2 \xrightarrow{q_{max}} 2X_1 \end{cases}$$

$$3\begin{cases} X_2 + G_3 \xrightarrow{q_3} O_3 \end{cases}$$

$q_1 = \gamma^{-1} k_1$

$q_2 = \gamma^{-1} k_2 C_{max}^{-1}$

$q_3 = \gamma^{-1} k_3 C_{max}^{-1}$

$qs_2 = \gamma^{-1}(\sigma - \sigma_2)$

C Simulation of ideal and DNA reactions

Fig. 5. Lotka–Volterra chemical oscillator example. (*A*) The formal chemical reaction system to be implemented with original (unscaled) and scaled rate constants. Desired initial concentrations of X_1 and X_2 are 2 and 1 unscaled and 20 and 10 nM scaled. (*B*) Reactions modeling our DNA implementation. Each formal reaction corresponds to a set of DNA reactions as indicated. Species X_2 requires a buffering module because $\sigma_2 < \sigma$ ($\sigma = \sigma_1 = k_1$ and $\sigma_2 = 0$). Maximum strand displacement rate constant $q_{max} = 10^6$ M^{-1} s^{-1} and initial concentration of auxiliary species G_i, T_i, L_i, B_i, and LS_j, and BS_j is $C_{max} = 10$ μM. Buffering-scaling factor $\gamma^{-1} = q_{max}(q_{max} - \sigma)^{-1} = 2$. The initial concentrations of strands X_1 and X_2 introduced into the system is $\gamma^{-1}20$ nM = 40 nM and $\gamma^{-1}10$ nM = 20 nM. (*C*) Plot of the concentrations of X_1 (*Red curve*) and X_2 (*Green curve*) for the ideal system (*Dashed line*) and the corresponding DNA species (*Solid line*).

a chemical logic circuit responding to an external input signal (black trace). The 2-bit counter is a classic example of a digital circuit with feedback (45); the high or low values of the red and green output species give the binary count of the number of input pulses, 0–3. This feedback circuit contrasts with the use-once circuits of ref. 34. Fig. 6*D* shows a different style of algorithmic behavior: a state machine (5, 6). This state machine increments the number of green spikes between consecutive red spikes by 1 every time.

Experimental Considerations

The correctness of our systematic construction was predicated on several idealizations of DNA behavior, and it is worth considering the deviations that we would expect in practice. A good approximation to strong sequence design (domain x binds exclusively to x^*) should be possible for several thousand long domains by using existing techniques developed for strand displacement systems (46). There are a limited number of short toehold domain sequences available, but it is straightforward to modify our construction to reuse toehold sequences without introducing errors.

This limit also constrains choices for reaction rate constants but can be countered by adjustment of auxiliary complex concentrations. More serious issues are presented by leak reactions in which an output is produced even if no input is present. Although experimentally characterized strand displacement systems exhibit leak rate constants up to a million times slower than the fastest desired reactions (35, 39), a leak could pose a problem for some target CRNs. One way to ameliorate a leak, while also allowing for unbounded running times, would be to provide auxiliary complexes at low concentrations in a continuous-flow stirred-tank reactor (1). These issues are discussed further in *SI Text*.

Conclusions

With a rich history and extensive theoretical and software tools, formal CRNs are a powerful descriptive language for modeling chemical reaction kinetics. By providing a systematic method for compiling formal CRNs into DNA molecules, our work suggests that CRNs can also be regarded as an effective programming language and used prescriptively for the synthesis of unique molecular systems. This view is bolstered by the fact that CRNs

A Oregonator (limit cycle oscillator)

$X_2 \rightarrow X_1$
$X_2 + X_1 \rightarrow \emptyset$
$X_1 \rightarrow 2X_1 + X_3$
$2X_1 \rightarrow \emptyset$
$X_3 \rightarrow X_2$
$X_3 \rightarrow \emptyset$

B Rössler (chaotic)

$X_1 \rightarrow 2X_1$
$2X_1 \rightarrow X_1$
$X_2 + X_1 \rightarrow 2X_2$
$X_2 \rightarrow \emptyset$
$X_1 + X_3 \rightarrow \emptyset$
$X_3 \rightarrow 2X_3$
$2X_3 \rightarrow X_3$

C 2-bit pulse counter (digital circuit)

where

dual rail representation:

species	value
$x(0)$ $x(1)$	x
high low	0
low high	1

logic	thresholding
$x(0) + y(1) \rightleftharpoons on_w$	$w(1) + w(1) \rightleftharpoons on_z$
$on_w + w(0) \rightarrow on_w + w(1)$	$w(0) + w(0) \rightleftharpoons off_z$
$x(1) + w(1) \rightarrow x(1) + w(0)$	$on_z + z(0) \rightarrow on_z + z(1)$
$y(0) + w(1) \rightarrow y(0) + w(0)$	$off_z + z(1) \rightarrow off_z + z(0)$

D Incrementer state machine (algorithmic)

v:=0
w:=0

v > 0?

no   yes

w:=w+1
v:=w

v:=v-1

clock made from chemical oscillator:

clk1
clk3 ← → clk2

catalyzed by clk1	catalyzed by clk2	catalyzed by clk3
$v1 \rightarrow v2 + v3$	$r1 \rightarrow r2 + r3$	$v2 \rightarrow \emptyset$ $v3 \rightarrow v1$
$c1 \rightarrow c2 + c3$		$c2 \rightarrow \emptyset$ $c3 \rightarrow c1$
$w2 \rightarrow w1$		$i \rightarrow \emptyset$ $r3 \rightarrow r1$

anytime:

$v2 + c2 \rightarrow \emptyset$ $r2 + v2 \rightarrow d + v2$ $c2 + r2 \rightarrow c2 + i + w1$
$d + v1 \rightarrow \emptyset$ $i + w1 \rightarrow i + v1 + w2$

Fig. 6. Examples showing more complex behavior. In all the maximum strand displacement rate constant $q_{max} = 10^6$ M^{-1} s^{-1} and the initial concentration of auxiliary species $C_{max} = 10$ μM. See Figs. S4 and S5 and *SI Text* for the rate constants used in *A–D*, as well as the details of *C* and *D*. Plots show the ideal CRN (*Dashed lines*) and the DNA reactions (*Solid lines*).

can implement arbitrary ordinary differential equations, analog and digital circuits, Turing-universal behavior, and—in the limit of small signal species concentrations and finite reaction volume—stochastic CRNs (47, 48). It is perhaps surprising that a simple primitive such as sequence-specific strand displacement proves sufficient for the implementation of arbitrarily complex chemical reaction kinetics, somewhat akin to how a few basic electronic elements such as transistors and wires are sufficient for the construction of arbitrarily complex logic circuits.

Our construction has several notable features for future molecular programming efforts. First, the signal species, entirely unreactive by themselves, react only when immersed in a complex buffer of auxiliary complexes. Thus, when implementing a system, one focusses on designing the buffer rather than the signal species, which stay the same regardless of the system. Our construction also highlights the rich dynamical behavior possible in multistranded nucleic-acid systems even without enzymes and covalent bond modification. Furthermore, established techniques for interfacing DNA to other chemistries (e.g., ref. 49) can enable programmable DNA reaction networks to respond to or control non-nucleic-acid systems, such as complex chemical synthesis pathways, biomedical diagnostics in vitro, or cellular functions in vivo. Finally, our construction and variants thereof (48) rely exclusively on a very simple molecular primitive, sequence-specific strand displacement reactions, suggesting that it could be adapted for other molecular substrates, such as RNA (11) or even peptides (17).

Many important chemical behaviors, such as pattern formation and self-assembly, involve geometric aspects that are beyond the scope of well-stirred reactions considered here. However, with a slight extension of our approach—controlling the relative diffusion rates of signal species—it would be possible to implement arbitrary reaction-diffusion systems (1). Though more challenging, regulating molecular assembly and disassembly steps with DNA signal species would allow for qualitatively more powerful computational systems (50) and systems capable of movement and construction of molecular-scale objects. We hope that the methods presented here provide a framework for successfully generalizing and unifying the existing experimental achievements in this direction (36, 37, 51).

1. Epstein IR, Pojman JA (1998) *An Introduction to Nonlinear Chemical Dynamics: Oscillations, Waves, Patterns, and Chaos* (Oxford Univ Press, London).
2. Arkin A, Ross J (1994) Computational functions in biochemical reaction networks. *Biophys J* 67:560–578.
3. Tyson J, Chen K, Novak B (2003) Sniffers, buzzers, toggles and blinkers: Dynamics of regulatory and signaling pathways in the cell. *Curr Opin Cell Biol* 15:221–231.
4. Wolf D, Arkin A (2003) Motifs, modules and games in bacteria. *Curr Opin Microbiol* 6:125–134.
5. Liekens AML, Fernando CT (2007) *Turing Complete Catalytic Particle Computers*, Lecture Notes in Computer Science (Springer, Berlin), Vol 4648, pp 1202–1211.
6. Soloveichik D, Cook M, Winfree E, Bruck J (2008) Computation with finite stochastic chemical reaction networks. *Nat Comp* 7:615–633.
7. Cook M, Soloveichik D, Winfree E, Bruck J (2009) *Algorithmic Bioprocesses* (Springer, Berlin), pp 543–584.
8. Fett B, Bruck J, Riedel M (2007) *Synthesizing Stochasticity in Biochemical Systems* (Association for Computing Machinery, New York), pp 640–645.
9. Cardelli L (2008) From processes to ODEs by chemistry. *TCS 2008, IFIP International Federation for Information Processing*, (Springer, Boston), Vol 273, pp 261–281.
10. Andrianantoandro E, Basu S, Karig DK, Weiss R (2006) Synthetic biology: New engineering rules for an emerging discipline. *Mol Syst Biol* 2:2006.0028.
11. Win M, Liang J, Smolke C (2009) Frameworks for programming biological function through RNA parts and devices. *Chem Biol* 16:298–310.
12. Keasling J (2008) Synthetic biology for synthetic chemistry. *ACS Chem Biol* 3:64–76.
13. Levskaya A, et al. (2005) Synthetic biology: Engineering Escherichia coli to see light. *Nature* 438:441–442.
14. Noireaux V, Bar-Ziv R, Libchaber A (2003) Principles of cell-free genetic circuit assembly. *Proc Natl Acad Sci USA* 100:12672–12677.
15. Ran T, Kaplan S, Shapiro E (2009) Molecular implementation of simple logic programs. *Nat Nanotechnol* 4:642–648.
16. Benner S, Sismour A (2005) Synthetic biology. *Nat Rev Genet* 6:533–543.
17. Ashkenasy G, Ghadiri M (2004) Boolean logic functions of a synthetic peptide network. *J Am Chem Soc* 126:11140–11141.
18. Magnasco MO (1997) Chemical kinetics is Turing universal. *Phys Rev Lett* 78:1190–1193.
19. Simpson Z, Tsai T, Nguyen N, Chen X, Ellington A (2009) Modelling amorphous computations with transcription networks. *J R Soc Interface* 6:S523–S533.
20. Hjelmfelt A, Weinberger ED, Ross J (1991) Chemical implementation of neural networks and Turing machines. *Proc Natl Acad Sci USA* 88:10983–10987.
21. Buchler NE, Gerland U, Hwa T (2003) On schemes of combinatorial transcription logic. *Proc Natl Acad Sci USA* 100:5136–5141.
22. Kim J, Hopfield JJ, Winfree E (2004) *Neural Network Computation by in Vitro Transcriptional Circuits. Neural Information Processing Systems*, ed Saul LK (MIT Press, Cambridge, MA), Vol 17, pp 681–688.
23. Okamoto M, Sakai T, Hayashi K (1987) Switching mechanism of a cyclic enzyme system: Role as a "chemical diode." *Biosystems* 21:1–11.
24. Hjelmfelt A, Weinberger ED, Ross J (1992) Chemical implementation of finite-state machines. *Proc Natl Acad Sci USA* 89:383–387.
25. Horn F, Jackson R (1972) General mass action kinetics. *Arch Ration Mech An* 47:81–116.
26. Érdi P, Tóth J (1989) *Mathematical Models of Chemical Reactions: Theory and Applications of Deterministic and Stochastic Models* (Manchester Univ Press, Manchester).
27. Klonowski W (1983) Simplifying principles for chemical and enzyme reaction kinetics. *Biophys Chem* 18:73–87.
28. Korzuhin M (1967) *Oscillatory Processes in Biological and Chemical Systems* (Nauka, Moscow), (in Russian), pp 231–251.
29. Morrison L, Stols L (1993) Sensitive fluorescence-based thermodynamic and kinetic measurements of DNA hybridization in solution. *Biochemistry* 32:3095–3104.
30. Green C, Tibbetts C (1981) Reassociation rate limited displacement of DNA strands by branch migration. *Nucleic Acids Res* 9:1905–1918.
31. Weinstock PH, Wetmur JG (1990) Branch capture reactions: Effect of recipient structure. *Nucleic Acids Res* 18:4207–4213.
32. Panyutin I, Hsieh P (1993) Formation of a single base mismatch impedes spontaneous DNA branch migration. *J Mol Biol* 230:413–424.
33. Yurke B, Turberfield AJ, Mills AP, Jr, Simmel FC, Neumann J (2000) A DNA-fuelled molecular machine made of DNA. *Nature* 406:605–608.
34. Seelig G, Soloveichik D, Zhang DY, Winfree E (2006) Enzyme-free nucleic acid logic circuits. *Science* 314:1585–1588.
35. Zhang DY, Turberfield AJ, Yurke B, Winfree E (2007) Engineering entropy-driven reactions and networks catalyzed by DNA. *Science* 318:1121–1125.
36. Yin P, Choi H, Calvert C, Pierce N (2008) Programming biomolecular self-assembly pathways. *Nature* 451:318–322.
37. Bath J, Turberfield A (2007) DNA nanomachines. *Nat Nanotechnol* 2:275–284.
38. Yurke B, Mills AP (2003) Using DNA to power nanostructures. *Genet Program Evol M* 4:111–122.
39. Zhang DY, Winfree E (2009) Control of DNA strand displacement kinetics using toehold exchange. *J Am Chem Soc* 131:17303–17314.
40. Reynaldo LP, Vologodskii AV, Neri BP, Lyamichev VI (2000) The kinetics of oligonucleotide replacements. *J Mol Biol* 297:511–520.
41. Panyutin I, Hsieh P (1994) The kinetics of spontaneous DNA branch migration. *Proc Natl Acad Sci USA* 91:2021–2025.
42. Khalil HK (1996) *Nonlinear Systems* (Prentice Hall, Englewood Cliffs, NJ).
43. Field RJ, Noyes RM (1974) Oscillations in chemical systems. IV. Limit cycle behavior in a model of a real chemical reaction. *J Chem Phys* 60:1877–1884.
44. Willamowski KD, Rössler OE (1980) Irregular oscillations in a realistic abstract quadratic mass action system. *Z Naturforsch A* 35:317–318.
45. Marcovitz A (2004) *Introduction to Logic Design* (McGraw–Hill, New York), 2nd Ed, pp 369–441.
46. Qian L, Winfree E (2009) A simple DNA gate motif for synthesizing large-scale circuits. *DNA Computing 14*, Lecture Notes in Computer Science (Springer, Berlin), Vol 5347, pp 70–89.
47. Gillespie D, et al. (1977) Exact stochastic simulation of coupled chemical reactions. *J Phys Chem* 81:2340–2361.
48. Cardelli L (2009) Strand algebras for DNA computing. *DNA Computing 15*, Lecture Notes in Computer Science (Springer, Berlin), Vol 5877, pp 12–24.
49. Gartner Z, et al. (2004) DNA-templated organic synthesis and selection of a library of macrocycles. *Science* 305:1601–1605.
50. Bennett C (1982) The thermodynamics of computation—A review. *Int J Theor Phys* 21:905–940.
51. Rothemund P, Papadakis N, Winfree E (2004) Algorithmic self-assembly of DNA sierpinski triangles. *PLoS Biol* 2:e424.
52. Soloveichik D, Seelig G, Winfree E (2009) DNA as a Universal Substrate for Chemical Kinetics (Extended Abstract). *DNA Computing 14*, Lecture Notes in Computer Science (Springer, Berlin), Vol 5347, pp 57–69.