ACS SyntheticBiology

pubs.acs.org/synthbio

Research Article

Programming the Sequential Release of DNA

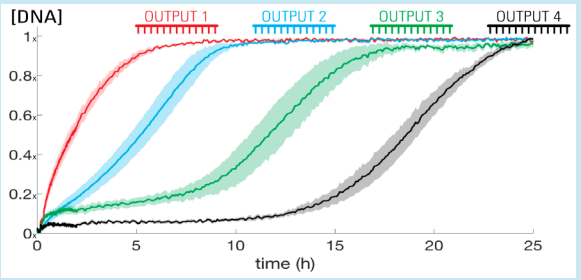
Dominic Scalise, Moshe Rubanov, Katherine Miller, Leo Potters, Madeline Noble, and Rebecca Schulman*

Cite This: *ACS Synth. Biol.* 2020, 9, 749–755

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: This study presents a mechanism for releasing a series of different short DNA sequences from sequestered complexes, one after another, using coupled biochemical reactions. The process uses stages of coupled DNA strand-displacement reactions that first release an output molecule and then trigger the initiation of the next release stage. We demonstrate the sequential release of 25 nM of four different sequences of DNA over a day, both with and without a centralized "clock" mechanism to regulate release timing. We then demonstrate how the presence of a target input molecule can determine which of several different release pathways are activated, analogous to branching conditional statements in computer programming. This sequential release circuit offers a means to schedule downstream chemical events, such as steps in the assembly of a nanostructure, or stages in a material's response to a stimulus.



KEYWORDS: *DNA nanotechnology, DNA strand-displacement, chemical computing, programmable matter*

At their most basic level, computers are machines that execute a programmed series of instructions. In many types of chemical computing, each individual instruction consists of the release of a molecular species into solution. One simple chemical program is "Release output molecule 1, and then release output molecule 2, and then release output molecule 3" (Figure 1). By designing the released molecules to activate different self-assembling building blocks,[1] this type of sequential release program could be used to coordinate the multistage assembly of a target nanostructure by making the components available one-by-one as the structure is ready to incorporate them.

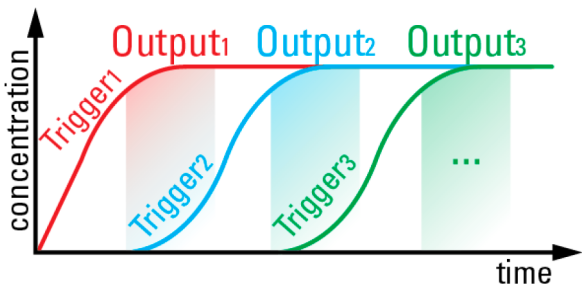


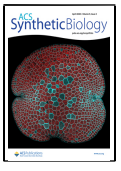
Figure 1. Sequential release. At each stage, an output molecule is released, after which the next output stage is triggered. In the schematic example, the completion of the red stage triggers the start of the cyan stage, and the completion of the cyan stage triggers the start of the green stage. Staged release could proceed over further stages.

The delayed release of molecular signals is an essential mechanism for executing sequences of events in biological systems, such as the regulation of cell cycle progression and cell differentiation,[2] can enable the automated delivery of complex drug-release patterns,[3] and can allow materials to self-disassemble after a programmable lifespan.[4] In this study we develop a DNA strand displacement circuit[5−7] that acts as a temporal scheduling device, releasing a set of single-stranded DNA output molecules in a specified order, one after another.

BACKGROUND

Molecular circuits can regulate the timing of chemical events,[8,9] such as the release of therapeutics[10,11] and the planned self-degradation of materials.[4,12] Many different classes of molecular circuits exist that can perform some information processing tasks, including intercellular circuits,[13] cell-free gene regulatory networks,[14] and enzymatic circuits primarily based on polymerase.[15] In this study we will focus on enzyme-free DNA strand-displacement (DSD) circuits, because strand displacement circuits are simple and cheap, and because they require only DNA oligonucleotides, they are particularly simple to adopt for use in a variety of contexts and

Received: October 1, 2019
Published: March 26, 2020



https://dx.doi.org/10.1021/acssynbio.9b00398
ACS Synth. Biol. 2020, 9, 749–755

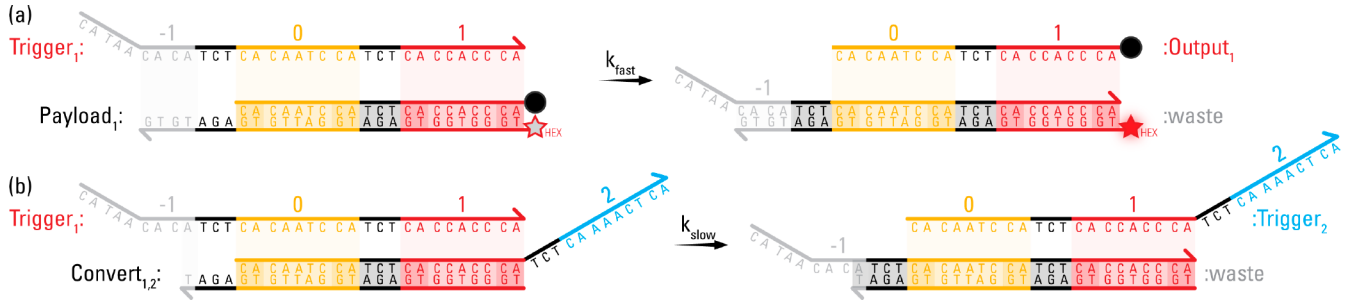


Figure 2. DNA strand-displacement (DSD) reactions for sequential release. (a) Example payload reaction in which a Trigger_1 strand releases an Output_1 strand from a Payload_1 complex. (b) Example convert reaction in which a Trigger_1 strand reacts with a $\text{Convert}_{1,2}$ complex to release a Trigger_2 strand.

environments. These circuits can also manipulate diverse libraries of other molecules, ions, light, thermal energy, and electronics.[1,16,17]

In a typical DSD reaction, an invading strand of DNA binds to a multistranded complex to displace a strand initially bound to the complex.[18] The effective reaction rate constant for a DSD reaction is primarily determined by the length of a short single-stranded "toehold" domain that initiates the reaction, with a longer toehold generally resulting in a faster reaction rate constant. By tuning the length of the toehold, the reaction rate constant of a DSD reaction can be specified across roughly 6 orders of magnitude.[18]

Various DSD architectures[7,19,20] have been developed with primitive reactions that can be composed together into larger reaction networks to execute specific tasks. DSD reactions have previously been used to implement a diverse set of information processing circuits, including Boolean logic circuits,[7,21−23] amplifiers,[24−26] and neural networks.[27] In principle, DSD circuits can be implemented either in solution, or tethered to surfaces.[28−30]

Here we describe a set of simple DSD reactions that can control the sequential release of different oligonucleotide sequences at specific prespecified times. While reaction cascades have previously been used to implement some specific tasks such as multilayer Boolean logic,[7] or in driving complex dynamics such as oscillations[31] the timing of the reactions within these systems could not be explicitly controlled and these circuits were not designed to expand the range and number of release steps and their relative timing. Further, these cascades often require multiple stages of signal amplification, which in chemical circuits can be leaky and hard to control. In this paper, we devise a simple mechanism for ordering release reactions driven by a single, robust driving reaction rather than multiple signal amplification steps.

DESIGN

The output strand for each i-th stage is initially sequestered within a multistranded DNA complex, which we call Payload_i. During stage i of the release process, a specific Trigger_i molecule first reacts very quickly with Payload_i to release Output_i (eq 1). Next, Trigger_i reacts slowly with a second complex called $\text{Convert}_{i,i+1}$ to release Trigger_{i+1}, thus triggering the next $(i + 1)$th stage (eq 2). A large difference between the rate constants for these two reactions ensures that Output_i is mostly released before Trigger_{i+1} is released. As a result, each release stage starts only after the previous release stage is mostly complete.

$$\text{Trigger}_i + \text{Payload}_i \xrightarrow{k_{\text{fast}}} \text{Output}_i \tag{1}$$

$$\text{Trigger}_i + \text{Convert}_{i,i+1} \xrightarrow{k_{\text{slow}}} \text{Trigger}_{i+1} \tag{2}$$

Given an initial set of Trigger, Payload, and Convert species that can react according to eqs 1 and 2, a multistage cascade that releases many different Output molecules can be initiated by adding Trigger_1 to a solution containing Payload_i and $\text{Convert}_{i,i+1}$ for $i \in [1, 2, 3, ..., n]$. The initial concentration of each Payload_i species determines the concentration of Output_i released at the i-th stage.

The sequential release cascade is designed so that the concentrations of Outputs released at each stage are independent from each other so that more or less of each successive payload can be released than at the previous stage by initializing each stage with a higher or lower initial concentration of Payload (see SI1). However, for the scope of this paper, we will provide all of the Payloads at the same initial concentration $[\text{Payload}]_0$, and will set the initial concentrations of each $\text{Convert}_{i,i+1}$ to be

$$[\text{Convert}_{i,i+1}]_0 = 1.5 \cdot [\text{Payload}]_0 \cdot (N - i) \tag{3}$$

to ensure that there is enough of each Convert species to fully react with all of the downstream stages, where N is the total number of stages, and the coefficient of 1.5 ensures a slight excess to keep the reaction from reaching a regime where the kinetics are limited by low Convert species concentrations.

The circuit comprising reactions 1 and 2 releases the Output molecules in a controlled order but not in regularly timed intervals. After the reaction cascade is initiated by the addition of a large pulse of Trigger_1, each successive stage releases its respective Output species at a rate slower than the rate of the previous stage. This slowdown is caused by the depletion of Trigger_1, which drives the reaction, and an increase in the total number of Convert reactions needed to produce Trigger_i from Trigger_1 as i increases.

We can better regulate the timing of the circuit by using an upstream reaction that continuously releases Trigger_1 at a slow, constant rate k_{prod} (eq 4), which we call the "clock" reaction. The constant production rate ensures that the total amount of Trigger_1 released is proportional to the time elapsed, providing a global measure of elapsed time. By coupling the clock reaction to the sequential release cascade, we can provide a steady source of fresh Trigger_1 molecules, counteracting depletion and preventing the circuit from slowing down as much over time.

https://dx.doi.org/10.1021/acssynbio.9b00398
ACS Synth. Biol. 2020, 9, 749−755

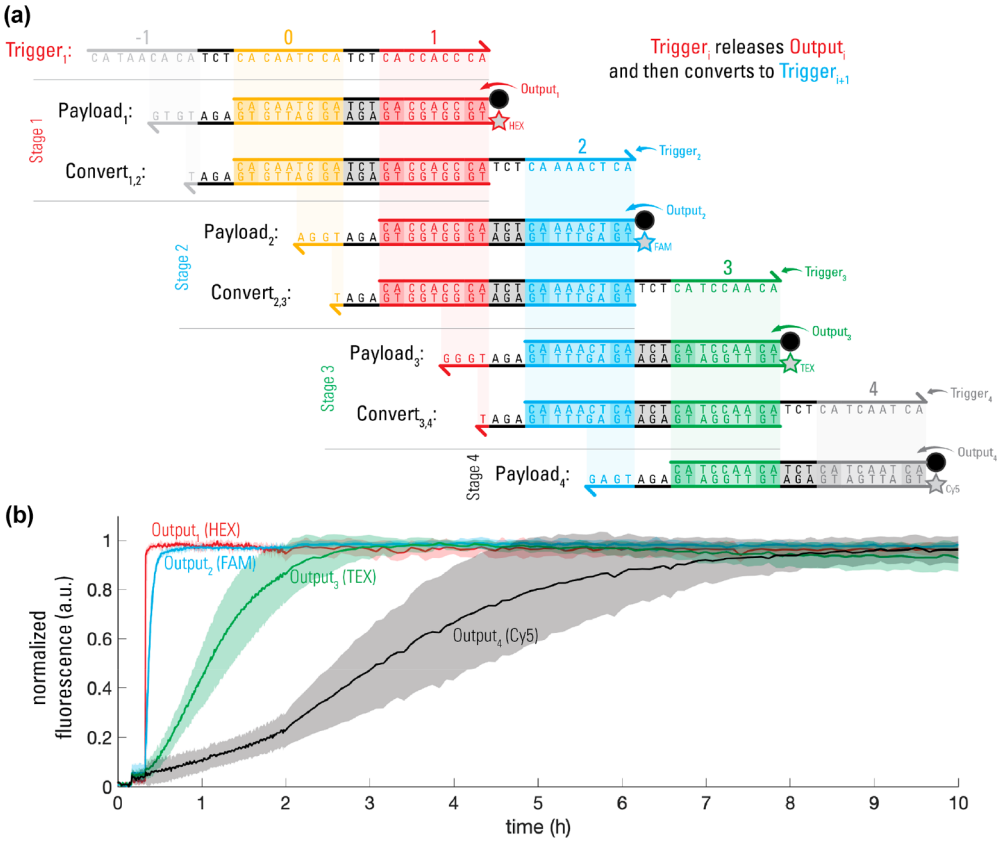


Figure 3. Unclocked sequential release. (a) The Trigger and the Payload and Convert complexes for a 4-stage sequential release reaction using the reactions shown in Figure 2. At each stage, a Trigger molecule first reacts quickly with the Payload to release an Output into solution. Any remaining Trigger then reacts slowly with the Convert complex, which converts it into the Trigger molecule for the next stage. (b) Experimental data showing the fluorescent Outputs being released in order. Initial solution contained 25 nM Payloads, $37.5\cdot(4-i)$ nM $\text{Convert}_{i,i+1}$, for $i = 1-4$, and 112.5 nM of the Trigger_1. See SI3 for methods and SI5 for data processing procedure. See SI6 for a negative control in which no Trigger_1 is added. Shaded regions, here and elsewhere for kinetic data unless otherwise stated, indicate 95% confidence intervals, calculated as $\mu \pm t\frac{\sigma}{\sqrt{n}}$ where μ is the sample mean, σ is the sample standard deviation, $n = 4$ is the sample size, and $t = 3.18$ is the coefficient for a 95% confidence interval with 3 degrees of freedom.

$$\varnothing \xrightarrow{k_{\text{prod}}} \text{Trigger}_1 \tag{4}$$

We thus designate the circuit comprising the reactions in eqs 1 and 2 as the *unclocked sequential release circuit* and the circuit comprising reactions in eqs 1, 2, and 4 as the *clocked sequential release circuit*. A more detailed theoretical model of these reactions is provided in the Supporting Information section SI1.

EXPERIMENTAL RESULTS

Unclocked Sequential Release Cascades. To implement eqs 1−2 with DSD reactions, we started with a set of Outputs, which in this implementation are sequences of DNA, to release in series. We designed the circuit so that at the start of the reaction, each of the Output strands is fully hybridized to the complementary domain of a strand contained within a Payload complex. Because the toehold of the Output strand in such a complex is in a double-stranded state, it is not available to react with other species in the circuit or in a potential downstream process. We then designed a set of Trigger strands to release each of the Outputs from a corresponding Payload *via* DSD reactions (Figure 2a). We selected a 7 nt toehold for the Trigger complex, which has the fastest average rate constant ($k_{\text{fast}} \approx 4\ \mu\text{M}^{-1}\ \text{s}^{-1}$) for DSD reactions at room temperature.[18]

Next, we designed a set of Convert species that could each react with respective Trigger_i molecules to release the next Trigger_{i+1}, thereby effectively translating a Trigger_i molecule into a Trigger_{i+1} molecule (eq 2, Figure 2b). We designed each Convert species so that this reaction is mediated by a 4 nt toehold, which results in a reaction rate constant significantly slower than a reaction with a 7 nt toehold ($k_{\text{slow}} \approx 2 \times 10^{-2}\ \mu\text{M}^{-1}\ \text{s}^{-1}$). To make it possible to track the kinetics of the different Output release processes, each of the Outputs had a conjugated quencher that suppresses the fluorescence of a fluorescent dye conjugated to the bottom strands of the Payload complexes (Figure 2b). When the Output is released from the Payload, the fluorescence increases and allows us to track the kinetics of the reaction. Our design uses the "leakless" DNA strand-displacement architecture[32] to suppress undesired (*i.e.*, leak) reactions between the Convert and Trigger complexes. This method requires that the Output sequences have some necessary sequence overlaps between the stages of the cascade. However, a cascade in which the released strands do not have overlap could be designed by including an additional translation reaction[32] for each stage that takes an Output strand as an input and produces a different sequence as a product (see SI2).

https://dx.doi.org/10.1021/acssynbio.9b00398
ACS Synth. Biol. 2020, 9, 749−755

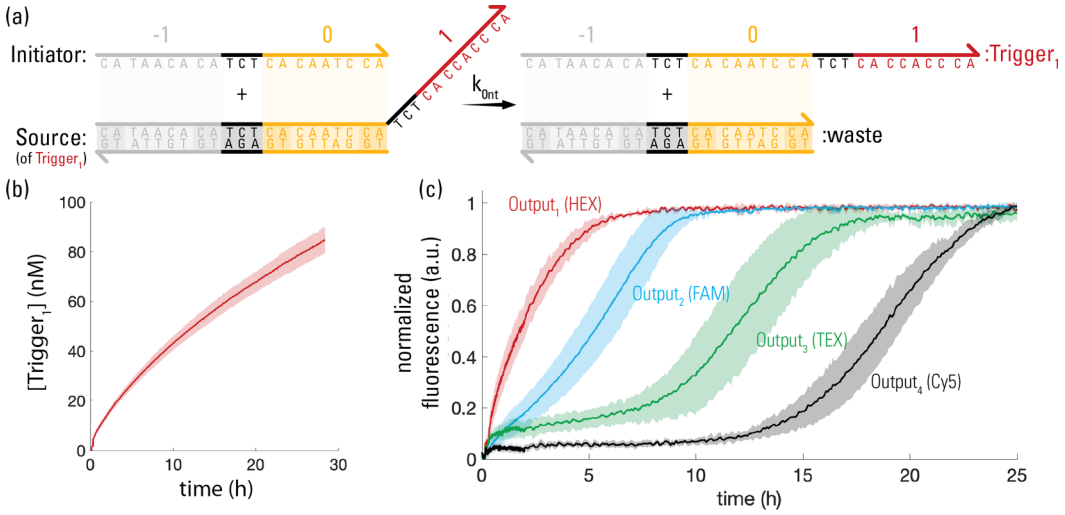


Figure 4. Clocked sequential release. (a) A DSD clock reaction, in which $Trigger_1$ strand is released from a Source complex by an Initiator strand at an approximately constant rate.[33] (b) The release of $Trigger_1$ from the clock reaction. Initial concentrations of Source and Initiator 1 μM each, with 300 nM $Payload_1$. See SI5 for fluorescence calibration procedure. (c) 1 μM Source and Initiator with 25 nM Payloads, and 37.5·(4 − i) nM $Convert_{i,i+1}$, for i = 1−4. The timing of release events is now rate limited by the rate of release of $Trigger_1$, making the delay times between stages roughly linear.

To test whether the sequential release cascade releases the Outputs in the designed order, we prepared a four-stage cascade with each $Payload_i$, for i = 1, 2, 3, and 4, and each $Convert_{i,i+1}$ complex for i = 1, 2, and 3 (Figure 3a). We then added $Trigger_1$ to initiate the reaction cascade and tracked the resulting kinetics by measuring changes in fluorescence over time (see full materials and methods in SI3−SI4). As anticipated, we observed that the Outputs were released in the designed order (Figure 3b), with the release rate of the Outputs decreasing at each stage.

Clocking. To make the timing of the release events more uniform, we used an upstream clock reaction[33,34] that continuously replenishes $Trigger_1$ at a constant rate (emulating eq 4), in lieu of adding a pulse of $Trigger_1$ to the solution to start the reaction cascade. In the clocked reaction system, an excess of $Trigger_1$ is initially sequestered in an inert state within an abundant Source complex. An Initiator strand reacts with the Source to displace $Trigger_1$ from the Source (eq 5, Figure 4a), so that it can then react with the species that implement the sequential release cascade.

$$\text{Source} + \text{Initiator} \xrightarrow{k_{0nt}} \text{Trigger}_1 + \text{waste} \tag{5}$$

The reaction between Source and Initiator can be designed to have a very small rate constant, in which case very little Source and Initiator are consumed on time scales of several days, and thus can approximate their concentrations as constant over this time duration. We use a reaction with such a small rate constant, k_{0nt}, allowing us to treat the net clock production rate as roughly a constant k_{prod} (eq 6)[33,34] within the clocked sequential release system. In the absence of a downstream load, the dynamics of production of $Trigger_1$ should evolve as

$$\left.\frac{d[\text{Trigger}_1]}{dt}\right|_{clock} = k_{release}[\text{Source}][\text{Initiator}] \approx k_{prod} \tag{6}$$

We first found conditions where the clock circuit could release the $Trigger_1$ molecule at a fairly constant rate over the desired course of our sequential release circuit. We mixed Source and Initiator together at 1 μM in the presence of 300 nM $Payload_1$ and observed the release of roughly 80 nM $Trigger_1$ over 24 h (Figure 4b).

Next, we coupled the clock reaction to the sequential release cascade to release the Outputs at more even intervals. In the unclocked version of the circuit, an excess of $Trigger_1$ is provided over the total amount of all Payload and Convert species. It immediately begins reacting with all of the $Payload_1$ and $Convert_1$ species. Increasing the reaction rate constants of the Payload and Convert species will increase the rate at which all of the downstream Outputs are released. In contrast, the clock reaction releases $Trigger_1$ slowly. Increasing the reaction rate constants of the Payload and Convert species cannot ever allow the rate at which Outputs are released to be faster than k_P, because one $Trigger_1$ molecule is ultimately required to release each Output molecule, and only a limited quantity of $Trigger_1$ has been produced per unit time by the clock reaction. We call this the $Trigger_1$-limited regime. If the Payload and Convert reaction rate constants are sufficiently high, then the rate at which Outputs will be released at each stage saturates at k_P.

When we mixed Source and Initiator with the Payload and Convert complexes, we observed the release of the Output strands still occurred in the designed order, but the rates of these release events were now slower and occurred at more evenly timed intervals (Figure 4c).

Branched Pathways and Conditional Statements. Every time one of the sequential release circuits developed in Section 2 runs, it should release the same molecules in the same order: these circuit are hard-coded to direct the release of a specific set of Outputs at a specific set of times. These circuits thus execute linear release programs analogous to a computer program that does not have any conditional "if" statements. The ability to include conditional statements in release programs would make it possible for sequential release processes encoded in chemical reactions to make decisions about which specific Output strands to release during their execution. Such decisions might be based on whether external stimuli molecules are present in solution.

https://dx.doi.org/10.1021/acssynbio.9b00398
ACS Synth. Biol. 2020, 9, 749−755

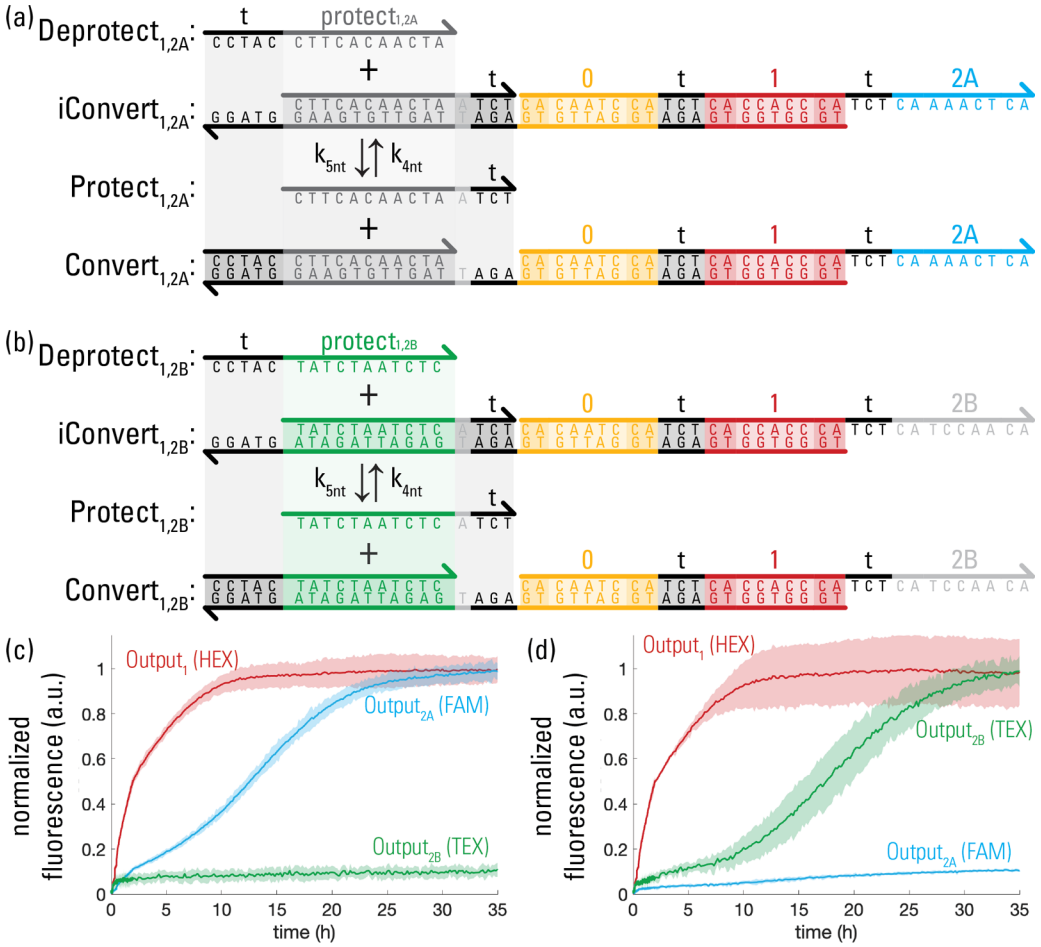


Figure 5. Branching during sequential release. (a) DSD reaction schematic for a conditional $Convert_{1,2A}$ complex, which is only active in the presence of an associated $Deprotect_{1,2A}$ strand (see sequences for parallel $Convert_{1,2B}$ system in SI3). (b) DSD reaction schematic for a $Convert_{1,2B}$ complex, activated by a $Deprotect_{1,2B}$ strand. (c) Kinetics of a branched, two-stage sequential release program with the conditional branch 2A activated by the addition of $Deprotect_{1,2A}$. Fluorescence for the FAM channel increases, indicating that $Output_{2A}$ is released as expected. (d) Kinetics of a two-stage sequential release program in which branch 2B is activated by the addition of $Deprotect_{1,2B}$. Reactions in both (c) and (d) included 1 μM Source and Initiator, 25 nM Payloads, 37.5 nM Convert complexes, and 50 nM of the stated Deprotect strands. In both (c) and (d) there is a small degree of leak fluorescence observed in the inactive pathway, likely due to a small fraction of deletion errors during synthesis of the Protect strands, which can partially expose the protected toehold and allow Trigger stands to react with the impure iConvert species.

We next updated the design of the Sequential Release circuit to incorporate the capacity for decision-making during the release process. To do so, we modified the Convert complex to create a "conditional" Convert complex that is initially inactive, which we called iConvert. An iConvert complex is activated when it reacts with its target Deprotect strand (eq 7).

$$\text{Deprotect}_{i,j} + \text{iConvert}_{i,j} \xrightarrow{k_{\text{deprotect}}} \text{Convert}_{i,j} \qquad (7)$$

The toehold that initiates the reaction between the iConvert complex and the Trigger strands is initially covered to prevent an iConvert complex from reacting with its upstream Trigger complex (Figure 5a,b). Each iConvert complex has a corresponding Deprotect strand that is designed to react with the iConvert complex to expose this toehold, thus allowing it to react with the trigger. For simplicity we approximate the deprotection of the branching iConvert complexes as irreversible, because the product activated Convert complex will be irreversibly consumed by the downstream release cascade. The branching reaction serves as a conditional statement of the form:

if(**Deprotect**$_{i,j}$ is present){
    convert(**Trigger_i** to **Trigger_j**);
}

Using the design in Figure 5a,b, multiple inactive Convert complexes that are sensitive to different Deprotect strands could be combined in the same solution to create conditional statements on a single release stage with more than one branching case. Multiple sets of inactive convert complexes could also create branch points at multiple stages of the release process.

We tested the ability to use iConvert complexes to create conditional release circuits by preparing a two-stage sequential release circuit with a single Payload for the first stage and two different Payloads for the second stages. We refer to these species as $Payload_1$, $Payload_{2A}$ and $Payload_{2B}$. Two inactive Convert complexes $iConvert_{1,2A}$ and $iConvert_{1,2B}$ form separate branches that allow the release of $Output_{2A}$ and $Output_{2B}$ from their respective Payloads. $iConvert_{1,2A}$ and $iConvert_{1,2B}$ can be activated by $Deprotect_{1,2A}$ and $Deprotect_{1,2B}$ respectively (see SI Table S2 for sequences). We made two solutions, each containing the clock circuit, $iConvert_{1,2A}$, $iConvert_{1,2B}$, $Payload_{2A}$ and $Payload_{2B}$. To one solution we added

https://dx.doi.org/10.1021/acssynbio.9b00398
ACS Synth. Biol. 2020, 9, 749−755

Deprotect$_{1,2A}$ (Figure 5c), and to the other solution we added Deprotect$_{1,2B}$ (Figure 5d). In each case, the Output for the activated branch was released, while the Output for the branch that should have remained inactive was not released to any significant extent.

DISCUSSION

In this paper we developed a DNA strand-displacement circuit that can release a series of different Output strands of DNA, one after another. This type of circuit could serve as a simple scheduling program to trigger different molecular events at different times. We demonstrated the operation of this circuit by constructing a four-stage sequential release circuit, which released 25 nM of Output at each stage. We showed that the circuit can run in an asynchronous or clocked configuration. In the unclocked mode, the time delay between the completions of different release stages is nonuniform, so that the reaction slows down dramatically between stages, while in the clocked mode, there is more uniform temporal spacing between stages. Lastly, we demonstrated that a sequential circuit can be modified to enable conditional logic, in which different branches of the release program can be activated depending on the presence of activating signal strands in the solution.

The four-stage release circuit demonstrated here can only coordinate four events. Building larger circuits with additional stages using the designed principles we developed, or looped circuits in which feedback from the final stage stimulates an additional round of release from the initial stage could significantly increase the complexity of the events coordinated by sequential release circuits. Further, the conditional branching mechanism could be reimagined as a "pause" feature, in which the release program halts until receiving a "continue" signal from the environment. In principle, similar sequential release circuits could be used to regulate timing in downstream DNA circuits, although the output strands would likely need to be modified (see SI2), and small amounts of output leaks may need to be suppressed by threshold species[7] particularly in sensitive circuits such as catalytic or autocatalytic amplifiers.[24−26,35] Finally, coupling between the Output strands and other systems besides DNA computing circuits,[16] such as aptamers[36,37] or nanostructures,[1] could enable these sequential release circuits to control the growth and actuation of downstream materials.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acssynbio.9b00398.

(SI1) Detailed models; (SI2) Eliminating sequence overlap between output strands; (SI3) Methods; (SI4) Sequences; (SI5) Reporter calibrations; (SI6) Untriggered cascade, negative control (PDF)

AUTHOR INFORMATION

Corresponding Author

Rebecca Schulman − *Chemical and Biomolecular Engineering and Computer Science, Johns Hopkins University, Baltimore, Maryland 21218, United States;* orcid.org/0000-0003-4555-3162; Email: rschulm3@jhu.edu

Authors

Dominic Scalise − *Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, Maryland 21218, United States;* orcid.org/0000-0001-8218-1797

Moshe Rubanov − *Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, Maryland 21218, United States;* orcid.org/0000-0001-8050-7407

Katherine Miller − *Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, Maryland 21218, United States*

Leo Potters − *Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, Maryland 21218, United States*

Madeline Noble − *Chemical and Biomolecular Engineering and Computer Science, Johns Hopkins University, Baltimore, Maryland 21218, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acssynbio.9b00398

Funding

This work was funded by NSF award SHF-1527377 to R.S.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors wish to thank Philip Dorsey and Chris Thachuk for helpful advice and discussions.

REFERENCES

(1) Zhang, D. Y., Hariadi, R. F., Choi, H. M., and Winfree, E. (2013) Integrating DNA strand-displacement circuitry with DNA tile self-assembly. *Nat. Commun. 4*, 1965.

(2) Rappaport, N., Winter, S., and Barkai, N. (2005) The ups and downs of biological timers. *Theor. Biol. Med. Modell. 2* (1), 22.

(3) Santini, J. T., Jr., Cima, M. J., and Langer, R. (1999) A controlled-release microchip. *Nature 397* (6717), 335.

(4) Heinen, L., and Walther, A. (2017) Temporal control of i-motif switch lifetimes for autonomous operation of transient DNA nanostructures. *Chem. Sci. 8* (5), 4100−4107.

(5) Zhang, D. Y., and Seelig, G. (2011) Dynamic DNA nanotechnology using strand-displacement reactions. *Nat. Chem. 3*, 103−113.

(6) Seelig, G., Soloveichik, D., Zhang, D. Y., and Winfree, E. (2006) Enzyme-free nucleic acid logic circuits. *Science 314*, 1585−1588.

(7) Qian, L., and Winfree, W. (2011) Scaling up digital circuit computation with DNA strand displacement cascades. *Science 332*, 1196−1201.

(8) Johnson-Buck, A., and Shih, W. M. (2017) Single-molecule clocks controlled by serial chemical reactions. *Nano Lett. 17* (12), 7940−7944.

(9) Garg, S., Shah, S., Bui, H., Song, T., Mokhtar, R., and Reif, J. (2018) Renewable Time-Responsive DNA Circuits. *Small 14* (33), 1801470.

(10) Lilienthal, S., Shpilt, Z., Wang, F., Orbach, R., and Willner, I. (2015) Programmed DNAzyme-triggered dissolution of DNA-based hydrogels: means for controlled release of biocatalysts and for the activation of enzyme cascades. *ACS Appl. Mater. Interfaces 7* (16), 8923−8931.

(11) Douglas, S. M., Bachelet, I., and Church, G. M. (2012) A logic-gated nanorobot for targeted transport of molecular payloads. *Science 335* (6070), 831−834.

(12) Heinen, L., and Walther, A. (2019) Programmable dynamic steady states in ATP-driven nonequilibrium DNA systems. *Sci. Adv. 5*, No. eaaw0590.

(13) Oishi, K., and Klavins, E. (2014) Framework for engineering finite state machines in gene regulatory networks. *ACS Synth. Biol. 3* (9), 652−665.

(14) Franco, E., Friedrichs, E., Kim, J., Jungmann, R., Murray, R., Winfree, E., and Simmel, F. C. (2011) Timing molecular motion and production with a synthetic transcriptional clock. *Proc. Natl. Acad. Sci. U. S. A. 108* (40), E784−E793.
(15) Zadorin, A. S., Rondelez, Y., Galas, J. C., and Estevez-Torres, A. (2015) Synthesis of programmable reaction-diffusion fronts using DNA catalyzers. *Phys. Rev. Lett. 114* (6), 068301.
(16) Scalise, D., and Schulman, R. (2019) Controlling matter at the molecular scale with DNA circuits. *Annu. Rev. Biomed. Eng. 21*, 469−493.
(17) Bujold, K. E., Lacroix, A., and Sleiman, H. F. (2018) DNA nanostructures at the interface with biology. *Chem 4*, 495−521.
(18) Zhang, D. Y., and Winfree, E. (2009) Control of DNA strand displacement kinetics using toehold exchange. *J. Am. Chem. Soc. 131*, 17303−17314.
(19) Soloveichik, D., Seelig, G., and Winfree, E. (2010) DNA as a universal substrate for chemical kinetics. *Proc. Natl. Acad. Sci. U. S. A. 107* (12), 5393−5398.
(20) Chen, Y. J., Dalchau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D., and Seelig, G. (2013) Programmable chemical controllers made from DNA. *Nat. Nanotechnol. 8* (10), 755.
(21) Genot, A. J., Bath, J., and Turberfield, A. J. (2011) Reversible logic circuits made of DNA. *J. Am. Chem. Soc. 133* (50), 20080−20083.
(22) Garg, S., Shah, S., Bui, H., Song, T., Mokhtar, R., and Reif, J. (2018) Renewable Time-Responsive DNA Circuits. *Small 14* (33), 1801470.
(23) Eshra, A., Shah, S., Song, T., and Reif, J. (2019) Renewable DNA hairpin-based logic circuits. *IEEE Trans. Nanotechnol. 18*, 252−259.
(24) Zhang, D. Y., and Seelig, G. (2010). DNA-based fixed gain amplifiers and linear classifier circuits. In *International Workshop on DNA-Based Computers*, pp 176−186, Springer, Berlin, Heidelberg.
(25) Chen, X., Briggs, N., McLain, J. R., and Ellington, A. D. (2013) Stacking nonenzymatic circuits for high signal gain. *Proc. Natl. Acad. Sci. U. S. A. 110* (14), 5386−5391.
(26) Dirks, R. M., and Pierce, N. A. (2004) Triggered amplification by hybridization chain reaction. *Proc. Natl. Acad. Sci. U. S. A. 101* (43), 15275−15278.
(27) Cherry, K. M., and Qian, L. (2018) Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature 559* (7714), 370−376.
(28) Chatterjee, G., Dalchau, N., Muscat, R. A., Phillips, A., and Seelig, G. (2017) A spatially localized architecture for fast and modular DNA computing. *Nat. Nanotechnol. 12* (9), 920.
(29) Qian, L., and Winfree, E. (2014). Parallel and scalable computation and spatial dynamics with DNA-based chemical reaction networks on a surface. In *International Workshop on DNA-Based Computers*, pp 114−131, Springer, Cham.
(30) Bui, H., Shah, S., Mokhtar, R., Song, T., Garg, S., and Reif, J. (2018) Localized DNA hybridization chain reactions on DNA origami. *ACS Nano 12* (2), 1146−1155.
(31) Srinivas, N., Parkin, J., Seelig, G., Winfree, E., and Soloveichik, D. (2017) Enzyme-free nucleic acid dynamical systems. *Science 358* (6369), No. eaal2052.
(32) Thachuk, C., Winfree, E., and Soloveichik, D. (2015). Leakless DNA strand displacement systems. *International Workshop on DNA-Based Computers*, pp 133−153, Springer, Cham.
(33) Fern, J., Scalise, D., Cangialosi, A., Howie, D., Potters, L., and Schulman, R. (2017) DNA Strand-Displacement Timer Circuits. *ACS Synth. Biol. 6* (2), 190−193.
(34) Scalise, D., Dutta, N., and Schulman, R. (2018) DNA Strand Buffers. *J. Am. Chem. Soc. 140*, 12069−12076.
(35) Baccouche, A., Montagne, K., Padirac, A., Fujii, T., and Rondelez, Y. (2014) Dynamic DNA-toolbox reaction circuits: A walkthrough. *Methods 67* (2), 234−249.
(36) Gatto, B., Palumbo, M., and Sissi, C. (2009) Nucleic acid aptamers based on the G-quadruplex structure: therapeutic and diagnostic potential. *Curr. Med. Chem. 16*, 1248−1265.
(37) Huizenga, D. E., and Szostak, J. W. (1995) A DNA aptamer that binds adenosine and ATP. *Biochemistry 34*, 656−665.

https://dx.doi.org/10.1021/acssynbio.9b00398
ACS Synth. Biol. 2020, 9, 749−755