ARTICLE

TECHNOLOGY

Designing modular reaction-diffusion programs for complex pattern formation

Dominic Scalise[1] & Rebecca Schulman[1,2]

Cells use sophisticated, multiscale spatial patterns of chemical instructions to control cell fate and tissue growth. While some types of synthetic pattern formation have been well studied[1–6], it remains unclear how to design chemical processes that can reproducibly create similar spatial patterns. Here we describe a scalable approach for the design of processes that generate such patterns, which can be implemented using synthetic DNA reaction-diffusion networks[7,8]. In our method, black-box modules are connected together into integrated programs for arbitrarily complex pattern formation. These programs can respond to input stimuli, process information, and ultimately produce stable output patterns that differ in size and concentration from their inputs. To build these programs, we break a target pattern into a set of patterning subtasks, design modules to perform these subtasks independently, and combine the modules into networks. We demonstrate in simulation how programs designed with our methodology can generate complex patterns, including a French flag and a stick figure.

INNOVATION

Current methods for controlling spatial variation in the concentration of biomolecules, including microcontact printing[9] and composite hydrogel assembly[10], can produce either patterns of immobilized molecules whose activity is limited, or spatial patterns that are not self-sustaining, dynamic or capable of responding to environmental signals. *In vivo*, regions of high and low concentrations of diffusing molecules arise because of the controlled interplay of molecular diffusion and chemical reactions. The resulting two- and three-dimensional patterns can remain stable over long time periods or evolve in a controlled fashion, and can form in response to a diverse array of chemical and physical stimuli[11]. Here we suggest a method to design reaction-diffusion systems to produce similarly well-defined, complex and programmable concentration patterns *in vitro*. This method could generate spatial concentration patterns that cannot be produced using existing design methods, including stable patterns of diffusing molecules, three-dimensional patterns and dynamic patterns that programmably form in response to chemical signals in the environment.

INTRODUCTION

During *Drosophila* embryogenesis, initially identical segments of a fruit fly embryo autonomously obtain their identities (*i.e.* head, thoracic, abdominal and terminal) from the different local concentrations of mRNA at different locations within the embryo[11]. Patterns of chemical concentrations are also thought to play important roles in mitotic spindle formation[12] and inter-cellular signaling[13]. Synthetic chemical patterning systems that could produce and manipulate similarly complex gradients might be useful for interaction with cells[14], for exerting dynamic spatial control over chemical systems, and for optimizing reactor throughput[6].

The idea that reaction-diffusion (RD) systems can produce patterns of concentrations was originally proposed by Alan Turing, who in a mathematical study of reaction and diffusion dynamics found mechanisms by which patterns of high and low chemical concentrations could form in an environment where chemical concentrations are initially homogeneous except for occasional local fluctuations in concentration[1]. Subsequent mathematical studies of chemical pattern formation[2,4–6] have suggested a family of reaction-diffusion processes capable of producing a diverse range of patterns including spots, stripes and spiral waves, and such patterns have been observed in chemical systems in two[3,15] and three[16] dimensions. However, because the features of these patterns arise when random fluctuations are amplified, the patterns they produce differ qualitatively from the robust patterns of signaling molecules observed in biological systems. The resulting features are often transient, and their exact shape and position are difficult to control reproducibly.

An alternative approach is to transform a simple pattern of initial chemical species into a more complex output pattern. This separates the task of generating complex patterns from the complementary task of breaking symmetry. Repeatable processes that perform pattern transformations could robustly produce specified patterns with well-controlled features on multiple size scales. Pattern transformation processes have been constructed using biological transcription factors active during development[17–19], and more recently, designed reaction-diffusion networks of synthetic DNA molecules[20,21]. While these systems can generate predictable patterns, the variety and complexity of geometries they can exhibit remains severely limited, and in many cases the patterns that arise are transient, such that the patterned species become well mixed over time.

Here we describe a staged pattern formation strategy in which specified patterns are gradually developed from simple initial patterns through

[1]Chemical and Biomolecular Engineering, [2]Computer Science, Johns Hopkins University, 3400 N Charles St, Baltimore, MD 21218, USA. Correspondence should be addressed to R.S. (rschulm3@jhu.edu).

Received 3 December 2013; accepted 5 February 2014; published online 24 March 2014; doi:10.1142/S2339547814500071

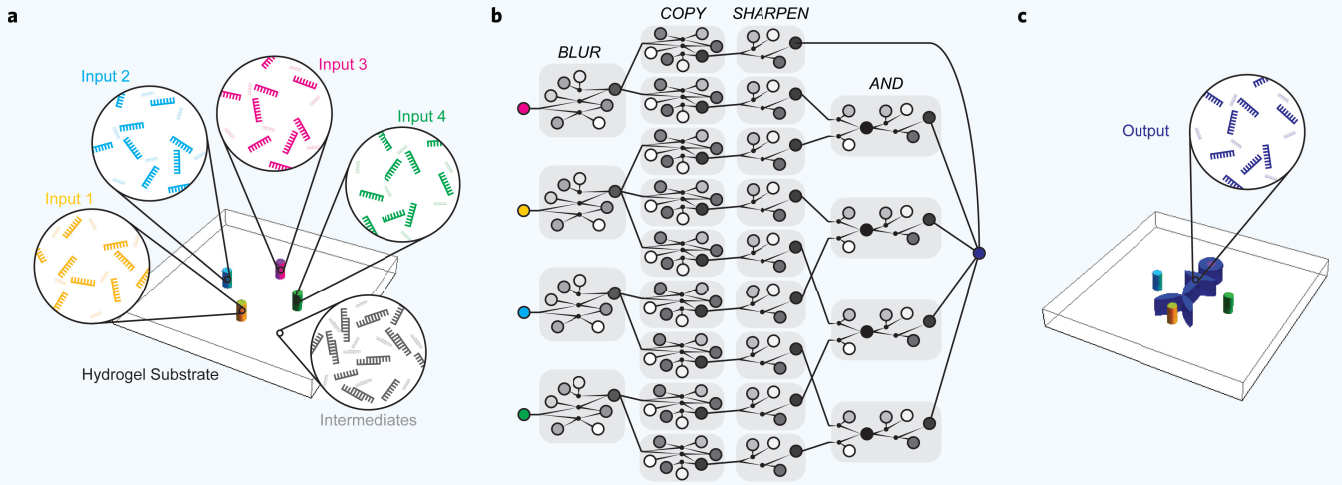


Figure 1 Reaction-diffusion programs for pattern generation (**a**) Synthetic DNA "input" molecules (yellow, cyan, magenta, and green) are fixed at particular locations inside of a hydrogel substrate. Many other species of DNA "intermediate" molecules (gray) are homogeneously distributed throughout the hydrogel. (**b**) The intermediate molecules comprise a modular reaction-diffusion "program", where each module (boxed) encodes a specific patterning instruction. The shaded circles represent different chemical species, and the lines connecting the circles represent designed reactions between these species. Intermediate molecules are designed to react only with other intermediate molecules inside of their parent module, facilitating the design of new patterns simply by rearranging the connections between modules. (**c**) The input molecules interact with the chemical reaction-diffusion network to produce a target pattern of DNA output molecules (blue).

a series of modular transformation stages. This strategy builds upon the robust repeatability of pattern transformations, expanding the complexity and diversity of attainable output patterns. Each stage performs a relatively simple transformation upon its input pattern, orchestrated by a reaction-diffusion network "module", producing an incrementally more elaborate output. Modules interact with each other only *via* their input and output patterns, enabling large composite networks to be flexibly assembled by adding modules or modified by rearranging the connections between modules. Viewing each module as a semantic command (*i.e.* *COPY*, *BLUR* or *SHARPEN*), staged networks represent algorithmic lists of patterning instructions, or *reaction-diffusion programs*, that generate complex patterns autonomously (**Fig. 1**).

A MOLECULAR LANGUAGE FOR SPATIAL PROGRAMMING

A chemical RD system consists of molecular species that can diffuse and interact through a designed set of coupled chemical reactions. The *de novo* construction of a set of molecules that interact *via* a specific set of a designed reactions can be challenging because of the difficulty in finding or designing chemical species that react as intended without crosstalk and the challenge of independently controlling the diffusion coefficients of different species. The number of technical challenges involved in such a design process increases quickly as the size of the desired network grows.

Recently, it has been proposed that the use of synthetic DNA as molecular species in RD systems could address many of these design challenges[20]. Advances in the development of DNA-based strand-displacement cascades have enabled the design of large reaction networks similar to biological signal-transduction networks[15,22,23]. The components of these cascades are short, synthetic DNA strands that are easy to synthesize and are biocompatible. DNA strand-displacement cascades have been used to implement robust logical operations and concentration amplification and thresholding. Strand-displacement cascades are programmed to perform these and other specific functions by appropriately designing their nucleotide sequences. Chemical reaction networks involving up to 130 unique species of DNA strands have been demonstrated[7,24]. While networks of this size are somewhat small compared to the size of the reaction networks within a cell, which may consist of thousands of signaling molecules and transcription factors[25], there is no fundamental obstacle to the design of larger synthetic DNA strand-displacement networks. Strand-displacement methods also offer unprecedented flexibility of design: methods summarized in **Fig. 2** have been developed with the goal of designing a set of DNA molecules that can emulate any chemical reaction network[8,26]. While quantitative control over reaction rate constants within these networks can still be a challenge[8,27], reaction networks can be designed that only depend qualitatively on the relative rates (*i.e.* fast or slow) between different reactions. The versatility of strand-displacement reactions, along with a growing number of computational tools that are available for the design of these networks[28,29], suggests that the design of large-scale DNA-based reactions networks is feasible.

The diffusion coefficients of DNA molecules are similarly programmable. In solution, single- and double-stranded DNA diffusion coefficients vary polynomially with sequence length[30] because their effective Stokes radius is a function of polymer chain length. In principle, therefore, adding additional bases to the strand or complex could slow down the diffusion coefficient of a species. More precise refinement of effective DNA diffusion rates is attainable within a porous substrate such as a hydrogel. By covalently attaching short DNA strand segments to the substrate, complementary DNA molecules diffusing nearby can transiently bind to the attached segments, slowing down their transport in a sequence-dependent manner[31,32]. The time spent bound is controlled by tuning the density of binding sites and the energy of strand interaction, which can generally be predicted *in silico* using efficient algorithms[28,33]. While these interactions cause diffusion to become anomalous, if the density of binding sites is sufficiently large, the diffusion of DNA species would be expected to be generally uniform and continuous over length scales greater than a few hundreds of nanometers, and thus would obey the standard diffusion equation. Both of these methods for slowing diffusion can be used to independently set the diffusion coefficient of different species in the network. Methods for building complex, arbitrary networks of coupled chemical reactions with synthetic DNA, and for

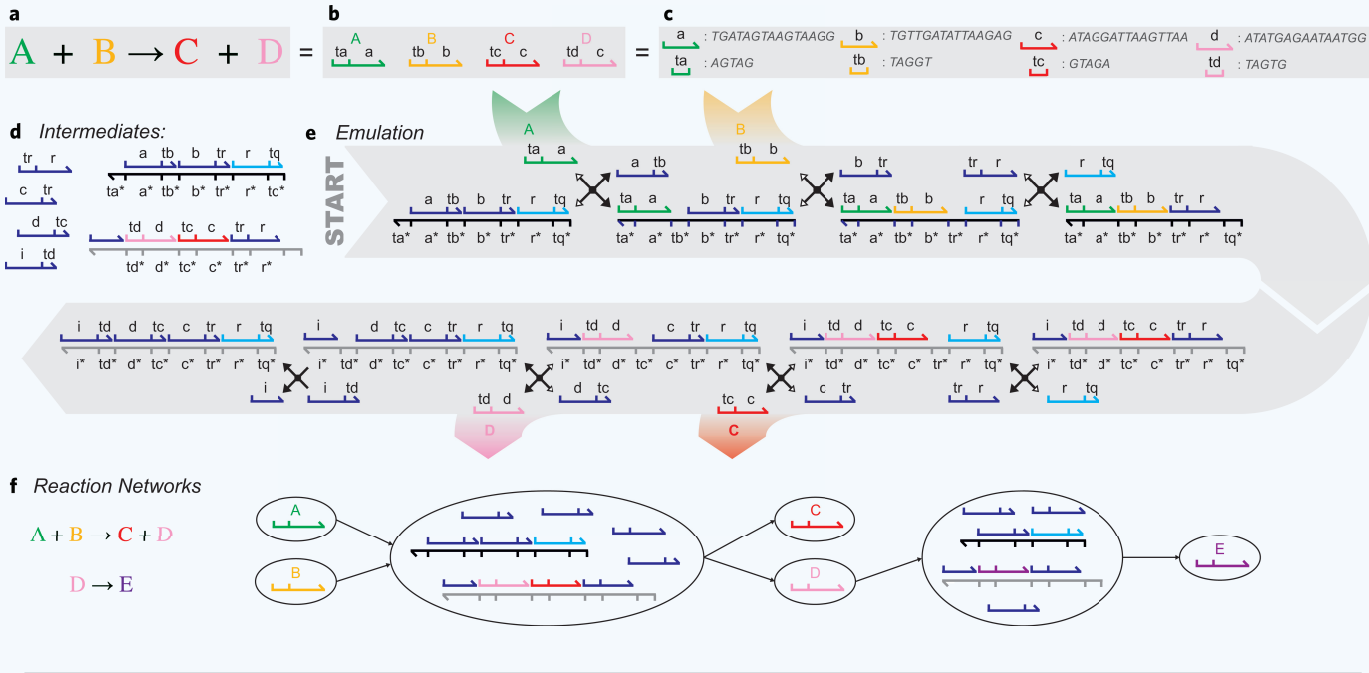


Figure 2 Networks of synthetic DNA molecules can emulate arbitrary chemical reaction dynamics. (*Architecture from* Ref. 8.) (**a**) A typical bimolecular chemical reaction. This reaction is emulated in the reaction process shown in **b–d**. (**b**) Specific single-stranded DNA molecules represent each reactant and product in the reaction. These molecules consist of a short "toehold" domain and a longer "recognition" domain. (**c**) Domains are unique sequences of base pairs (A, G, C and T). (**d**) A set of intermediate species interacts with the reactants. The collective reactions executed by the intermediates, reactants and products emulate the dynamics of the reaction in **a**. (**e**) The reactants A and B interact directly with the intermediate species, initiating a series of reactions that ultimately release the product species C and D. A reaction with fewer reactants or products can be emulated by making simple alterations to the intermediate complexes[8]. (**f**) Multiple such reactions can be chained together to form large reaction networks that emulate the dynamics of arbitrary networks of coupled chemical reactions.

independently controlling the diffusion coefficient of each DNA strand could be combined to design and build complex reaction-diffusion systems *in vitro* with synthetic DNA.

The oligonucleotides in these systems might also interact with other molecules in the environment in a sequence-specific fashion through the use of aptamers, nucleic acid sequences that bind specifically to a target ligand. Molecules for which aptamer sequences are known include many growth factors[34] and small molecules[35–37], and such connections have been generated previously as components of DNA strand-displacement networks[38].

SELF-SUSTAINING PATTERN FORMATION PROCESSES

Without a continuous supply of energy, heterogeneous distributions of chemical species will become well mixed over time, making the construction of steady-state heterogeneous patterns infeasible. In cells and tissues, proteins and other signaling molecules are constantly produced and degraded, providing a source of energy with which to maintain heterogeneous patterns of chemical species in spite of the effects of diffusion. Analogous production and decay reactions that continuously produce new species from high-energy precursors and degrade old species into inert waste could likewise produce stable patterns of concentrations *in vitro*. Here we use this technique to produce stable, static patterns of chemical concentrations.

Strand displacement systems can emulate stabilizing production and degradation reactions powered by high concentrations of precursor molecules[26]. Production can occur *via* the slow conversion of an inert precursor present at high concentration into the active species, and degradation can occur *via* the slow conversion of the active species into a lower energy waste species.

MODULAR REACTION-DIFFUSION PROGRAMS

We define a module as a coupled set of chemical reactions that perform a designed transformation of a pattern of input molecules. Modules are connected together such that "upstream" modules produce an output species that "downstream" modules accept as input species. Hierarchical modules, consisting of multiple submodules, can perform more sophisticated pattern transformations. To produce self-sustaining patterns, modules include production and degradation reactions that resupply component species, and maintain heterogeneous patterns at steady state.

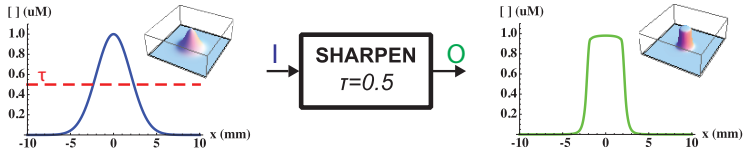
Modular engineering makes it possible to design large, complex networks by first designing simple, reusable modules, and then arranging them into networks. Each module implements a semantic patterning instruction (*e.g. COPY* or *BLUR* a pattern), which can be connected together without worrying about the details of how these instructions are implemented within each module. The modular instructions we use in this paper are detailed in **Fig. 3**.

A reaction-diffusion program is a set of connected modules that orchestrate a pattern formation process. To verify that the networks we design will form their target patterns, we use a mathematical model of each module's dynamics consisting of a set of coupled partial differential equations that governs their dynamics, assuming mass action kinetics[39]. To evaluate the function of the network, these equations can be integrated numerically using measured reaction rate constants and diffusion coefficients from the existing literature[27,40]. For instance, given an RD program consisting of the equations

$$A + B \xrightarrow{k_1} C \quad (1)$$

$$C \xrightarrow{k_2} waste \quad (2)$$

a ***SHARPEN***



Converts *I* into a discrete output of *high/low* states. *High* regions are produced where *[I]* is greater than threshold τ, and *low* regions are produced elsewhere. First, a rapid thresholding reaction depletes *[I]* by τ, so *[I]* is zero except where *[I]*>τ initially. Any remaining *I* slowly catalyzes the conversion of inert amplifier A into output *O*, at *[O]*=α. At steady state, O is *high* only where *[I]*>τ initially.

thresholding:

$$I + T \xrightarrow{k_{react,T}} waste$$

amplification:

$$I + A \xrightarrow{k_{react,A}} I + O$$

species cycling:

$$O \xrightarrow{k_{degrade,O}} waste$$

$$\xrightarrow{k_{prod,T}} T, \text{ and } T \xrightarrow{k_{degrade,T}} waste$$

$$\xrightarrow{k_{prod,A}} A, \text{ and } A \xrightarrow{k_{degrade,A}} waste$$

reaction rate constants:

$$k_{react,T} >> k_{react,A}$$

$$k_{degrade,A}=k_{degrade,T}=k_{degrade,O}$$

$$k_{prod,T} = \tau * k_{degrade,T}$$

$$k_{prod,A} = \alpha * k_{degrade,A}$$

diffusion coefficients:

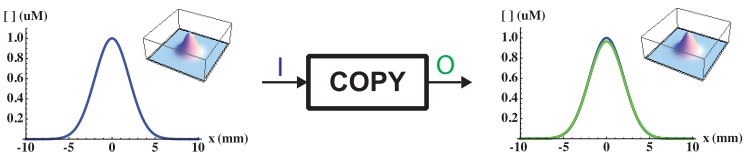
$$D_I = D_O = D_T = D_A$$

where:

$$\tau = [T]_{steadystate,noinput}$$

$$\alpha = [A]_{steadystate,noinput}$$

b ***COPY***



Takes input pattern *I* and produces a copy of this pattern in *O*, without depleting *I*. If diffusion is much slower than the chemical reactions then $\frac{\partial [O]}{\partial t} = k([I] - [O])$, which is the equation for a proportional controller. Thus a COPY module continuously restores *[O]* to the set point *[I]* everywhere. Because *I* is a catalyst, it is not consumed regardless of what happens to *O*. The COPY module buffers *I* from downstream loading, allowing modules to deplete *[O]* without affecting *[I]*. This buffer is a crucial tool for adding and rearranging modules without affecting the upstream circuit.

catalytic copy:

$$I \xrightarrow{k_{prod,O}} I + O$$

species cycling:

$$O \xrightarrow{k_{degrade,O}} waste$$

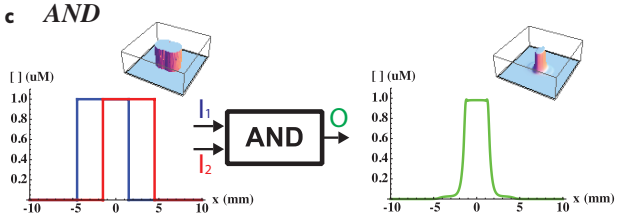
reaction rate constants:

$$k_{prod,O} = k_{degrade,O}$$

diffusion coefficients:

$$D_I = D_O$$

c ***AND***



Takes Boolean input patterns I_1 and I_2, and produces output pattern *O* that is *high* only where both inputs are *high*. First, *[I1]* and *[I2]* are summed into an intermediate pattern *N*. *[N]=2β* where both inputs are *high*, *[N]=β* where only one input is *high*, and *[N]=0* elsewhere. A SHARPEN module with τ=*1.3β* takes *N* as input, producing the described output pattern *O*.

summing of inputs:

$$I_1 + M \xrightarrow{k_{react,M}} N$$

$$I_2 + M \xrightarrow{k_{react,M}} N$$

thresholding of inputs:

$$N \rightarrow \boxed{sharpen(\tau{=}1.3\beta)} \rightarrow O$$

species cycling:

$$\xrightarrow{k_{prod,M}} M, \text{ and } M \xrightarrow{k_{degrade,M}} waste$$

$$N \xrightarrow{k_{degrade,N}} waste$$

reaction rate constants:

$$k_{degrade,N} = k_{degrade,M}$$

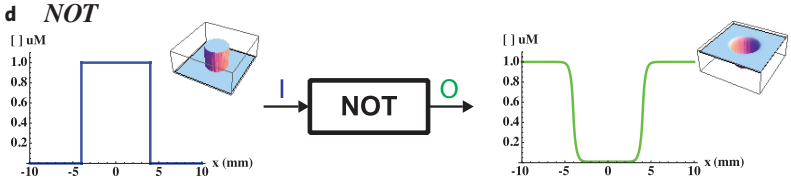
$$k_{react,M} >> k_{prod,M} = 2\beta k_{degrade,M}$$

where: $\beta = high$ concentration

diffusion coefficients:

$$D_{I_1} = D_{I_2} = D_N = D_O = D_M$$

d ***NOT***



Takes a Boolean input pattern *I*, and produces an inverted Boolean output *O*. *[O]* is *high* where *[I]* is *low*, and *low* where *[I]* is *high*. Slow cycling reactions continuously push *[O] high*, so in the absence of *I*, *[O]* is *high*. *I* and *O* rapidly annihilate each other, so *[O]* is switched to *low* in the presence of *I*.

logical inversion:

$$I + O \xrightarrow{k_{react,O}} waste$$

species cycling:

$$source \xrightarrow{k_{prod,O}} O, \text{ and } O \xrightarrow{k_{degrade,O}} waste$$

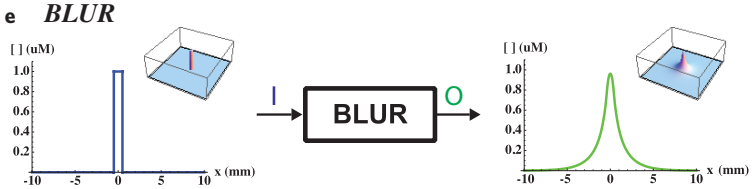
reaction rate constants/diffusion coefficients:

$$k_{react,O} >> k_{prod,O} = \alpha * k_{degrade,O}$$

$$D_I = D_O$$

where: $\alpha = [O]_{steadystate,noinput}$

e ***BLUR***



Assembles a smooth gradient *O* centered around a fixed reference point *I*. This module uses the same reactions as the COPY module, but places different constraints on the constants. The distance from the reference to any point *p* can be calculated as a function of *[O]* at *p*, provided the reaction rate constants and diffusion coefficients are known. *I* catalyzes the local production of *O* at the reference. *O* diffuses away from this point and also degrades slowly. At steady state, *[O]* at a distance *r* from the reference is: $[O]_r = [O]_0 e^{-r/\sqrt{D_O/k_{prod,O}}}$

signal generation:

$$I \xrightarrow{k_{prod,O}} I + O$$

$$O \xrightarrow{k_{degrade,O}} waste$$

reaction rate constants:

$$k_{prod,O} > k_{degrade,O}$$

diffusion coefficients:

$$D_O > D_I = 0$$

initial conditions:

$$[I]_0 = \begin{cases} 1 & \text{if } r \le r_{max}, \\ 0 & \text{otherwise .} \end{cases}$$

Figure 3 A library of reaction-diffusion modules for building pattern-formation programs.

the corresponding partial differential equation describing the concentration of C over space and time is

$$\frac{\partial C(t,x,y)}{\partial t} = D_C \nabla^2 [C](t,x,y) + k_1[A](t,x,y)[B](t,x,y) - k_2[C](t,x,y), \quad (3)$$

where D_C is the diffusion coefficient for species C. Similar equations govern the behavior of species A and B. Details concerning our modeling process, including boundary conditions, parameter values and numerical integration techniques, are provided in **Supplementary Information Section 1**.

A *DRAW FRENCH FLAG* patterning program

A central hypothesis of developmental biology is that spatial patterns of morphogens can instruct groups of cells to differentiate into specialized roles. These morphogen concentrations often encode information in the form of either high or low local concentrations. A concentration can thus be a Boolean value, *i.e.* a variable with only two possible values: *high* or *low*. In the French flag model[19], a one-dimensional linear gradient of an input morphogen is translated into three discrete regions of gene expression with the same geometry as the tri-colored French flag, such that each of three species has either a high or low concentration in each of the distinct regions. In this example, cells in these three regions could then adopt specialized identities depending on which of the Boolean *blue*, *white* or *red* regions they occupy.

While this model is broadly accepted as one explanation of pattern formation during development[25,41], a physical reaction-diffusion network capable of recapitulating this pattern *in vitro* has not been described. Here we design a modular reaction-diffusion program that generates a French-flag pattern (**Fig. 4**) from a linear input gradient.

We assume that the input to the system is a stable linear gradient of a "morphogen" I,

$$[I](x) = x\left(\frac{1\,\mu\text{M}}{10\text{ mm}}\right), \quad \text{where } x : [0\text{ mm}, 10\text{ mm}]. \quad (4)$$

The output that results is a stable French-flag pattern in which three different labeled output species, *blue* (B), *white* (W) and *red* (R) are present in the left, middle and right third regions respectively at approximately 1 μM. The concentrations of these species are close to zero elsewhere. To demonstrate our process of RD program design, we construct this patterning program in stages. We first consider how to produce a single red stripe using the reaction networks. We then add modules (**Fig. 3**) to our circuit until the circuit also produces white and blue stripes as outputs.

The challenge in generating a French flag pattern from a linear input gradient is that small differences in the concentration of the input species along the gradient must be transformed into large differences in the concentration of the output species in the final output pattern. For example (see **Fig. 4**), just to the right of the red-white boundary, $[R]$ at steady state should be high and $[W]$ should be low. An arbitrarily small distance to the left, on the other side of the boundary, $[R]$ should be low and $[W]$ should be high. Yet between these points the input concentration $[I]$ changes only linearly with distance. As shown in **Fig. 3a**, a *SHARPEN* module can produce an output species that is *high* everywhere the input species exceeds a threshold concentration and *low* everywhere else. A *SHARPEN* module with a threshold concentration of $\frac{2}{3}$ μM operating on the input gradient could thus produce an intermediate species S_1 only in the regions

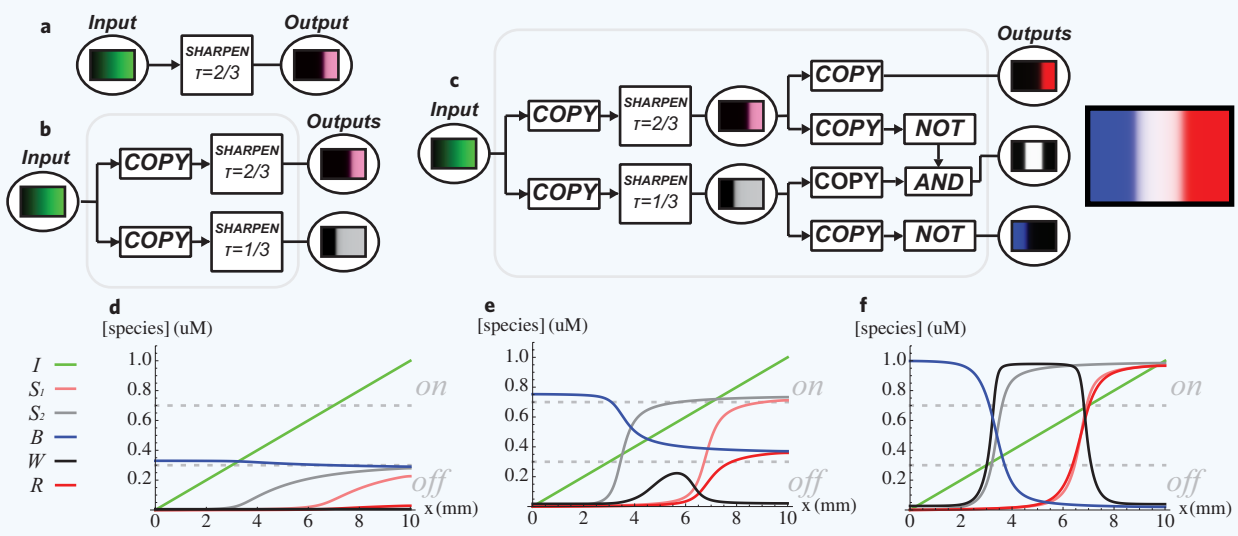


Figure 4 An RD program that produces a French flag pattern from a linear input gradient. (**a**) A simple network that generates a single stripe of intermediate species S_1 (pink) as output when the input is a linear input gradient (green). The stripe appears in regions where $[I] > \frac{2}{3}$ μM. The simulated steady-state profile of S_1 is shown in the output bubble. (**b**) A network to produce two stripes. Two copy modules produce two copies I_1 and I_2 of the input species I. I_1 and I_2 serve as inputs to two threshold amplifier modules that produce stripes of S_1 and S_2 (gray). The *COPY* modules protect the input signal from downstream loading, ensuring that the *SHARPEN* modules do not affect one another. The simulated steady state profiles of S_1 and S_2 are shown in the top and bottom output bubbles respectively. (**c**) The complete network to produce a French flag pattern from a linear gradient input. This network is an expanded version of the network in (**b**). A red stripe of R is produced where $[S_1]$ is *high*. A white stripe of W is produced where $[S_2]$ is *high* but $[S_1]$ is *low*. A blue stripe of B is produced where both $[S_1]$ and $[S_2]$ are *low*. The simulated steady state profiles of B, W and R are shown in the output bubbles, and are superimposed together in the image to the right. (**d**) Simulated concentrations of species in the French-flag RD program 200 seconds after introducing the input gradient, as they would appear if buffered against a load by a *COPY* module immediately downstream. Species are color coded to match the bubbles in (**c**). (**e**) Simulated concentrations of species after 700 seconds. (**f**) Simulated concentrations of species close to steady state. Simulation details including reaction rates, diffusion coefficients, PDEs and initial conditions are provided in the **Supplementary Information Section 2.1**.

where $[I] > \frac{2}{3}$ μM, *i.e.* in the right third of the substrate, creating a large change in $[S_1]$ across the red-white boundary (**Fig. 4a**).

A second *SHARPEN* module with a threshold value of $\frac{1}{3}$μM could produce a different intermediate species $[S_2]$ in an area that covers two-thirds of the width of the substrate, *i.e.* where both the red and white stripes should exist in the final pattern. Together, S_1, which is present only in the rightmost third of the space, and S_2 would divide the space into three discrete regions that correspond to the three desired French flag stripes: one where $[S_2]$ and $[S_1]$ are both high, one where $[S_2]$ is high and $[S_1]$ is *low*, and one where the concentrations of both species are low (**Fig. 4b**).

However, because the *SHARPEN* module depletes the concentration of its input species, if two different *SHARPEN* modules both have the same input *I*, they would compete with each other for *I*. Because both modules would deplete *I* during the course of their operation, *I* would end up being below the threshold value for amplification everywhere. As a result, neither S_1 nor S_2 would be produced anywhere. This problem would be addressed if the two *SHARPEN* modules used as their respective inputs two different species I_1 and I_2 that each have the same concentration as *I* in all locations. To produce these species, we designed a *COPY* module (**Fig. 3b**) that takes an input *I* and produces an output *O* that has the same concentration as *I* everywhere without depleting the input *I*. We include a *COPY* module that produces a copy of *I* with the same concentrations as *I* upstream of each *SHARPEN* module to prevent the two *SHARPEN* modules from competing for the same input species. The resulting circuit (**Fig. 4b**) produces a pattern where S_1 is present in the rightmost third of the space and S_2 is present everywhere but the leftmost third of the space.

To produce the French flag pattern, output species *R* should be produced where $[S_1]$ is *high*, output species *W* should be produced where $[S_2]$ is *high* but $[S_1]$ is *low*, and output species *B* should be produced where both $[S_1]$ and $[S_2]$ are *low*. *AND* and *NOT* modules (**Fig. 3c,d**) can direct the production of each of these species by producing the output molecule only where the corresponding Boolean function of the input concentrations is satisfied (**Fig. 4c**).

The French flag patterning circuit in **Fig. 4c** is a collection of connected modules, each of which contains a small RD program comprised of abstract chemical species (such as *I*, *A*, *T* and *O* in the *SHARPEN* module). This abstract chemical reaction network can be translated into a set of strand-displacement reactions that implement the same dynamics as the reaction-diffusion network we designed. There may be many different ways to translate an abstract chemical reaction network into a set of DNA strand-displacement reactions using a method such as that shown in **Fig. 2**. In the **Supplementary Information**, we propose a potential strand-displacement network for each of our modules. One measure of the complexity of the resulting network is the number of species it contains. Our network (shown in **Supplementary Information Section 2**) consists of 77 unique initial DNA strand assemblies containing a total of 108 component DNA strands. The size of the network is therefore smaller than strand-displacement networks that have been experimentally demonstrated[7].

To test that the network we designed produces the target pattern, we numerically integrated the partial differential equations for the RD system consisting of the abstract chemical species (*I*, *A*, *T*, . . .). The results (**Fig. 4c,f**) show that the network is expected to produce the designed pattern, taking on the order of an hour to reach steady state. Close to the boundaries between the blue, white and red regions, there are transition areas where either the blue and white, or the white and the red species are both produced.

These transition regions arise because of the diffusion of the output species, which acts to blur the pattern, and because of what we term "leaks" in our reactions, in which a slow reaction uses a small amount of the product that is otherwise depleted very quickly by a much faster reaction.

The French flag patterning program demonstrates that we can design RD programs that use the geometric information provided by a heterogeneously distributed input species to produce an incrementally more complex heterogeneous distribution of output species over a single spatial dimension. While microfluidic devices can already be used create a variety of gradient patterns with 1-dimensional features[42] similar to the French flag pattern, producing analogous patterns in 2- and 3-dimensions is considerably more complex.

A *DRAW STICK FIGURE* patterning program

To show that our patterning method can generate 2-dimensional patterns, we use the same step-wise modular design process to construct a program that produce a stick figure pattern that consists of a head, torso, arm and leg segments (**Fig. 5a**). In a 1-dimensional program like the French-flag generator, a *SHARPEN* module acting on a linear gradient created a linear *high/low* step function. In two dimensions, a radial gradient is generated when a species is produced at a single point and then slowly degrades as it diffuses away from its point of production. When the *BLUR* module (**Fig. 3e**) takes an input that is produced at a single point, it produces such a gradient. A *SHARPEN* module acting on a radial gradient creates a radial step function, *i.e.* a circle. The radius of this circle is specified by the threshold concentration of the *SHARPEN* module corresponding to the concentration of the radial gradient along the edge of the circle. The *SHARPEN* module will produce a *high* output signal only in regions where the radial gradient is above the threshold concentration, inside of the radius of the circle. The center of the circle is set by the location of the input point to the *BLUR* module. Together, the *BLUR* and *SHARPEN* modules encode a DRAW CIRCLE patterning instruction, which itself can be considered a module. Using the DRAW CIRCLE instruction repeatedly along with the *AND* module, it is possible to produce a complete stick figure pattern from an input consisting of four particular inputs produced only at four particular points (**Fig. 5b**).

A variety of techniques might be employed to pattern the input points for the stick figure, including micro-contact printing[9] or lithography[21]. Because the input patterns would be covalently bound to the substrate, they would remain stable over time rather than diffusing away, even though these inputs are not replenished. Alternatively, we could extend existing strand-displacement systems to actively generate and maintain the input pattern from reservoirs attached to the substrate boundaries[43], analogous to the actively generated input gradient in the French Flag program.

Since the stick figure's head is a circle, we can pattern the head with a single DRAW CIRCLE module. To build the rest of the stick figure, we need a mechanism to produce line segments for the arms, torso, and legs. One convenient way to produce an approximate line segment is to produce a species at the intersection of two large circles: If two circles are separated by slightly less than then sum of their radii, the intersection is a line-like region. We can repeatedly use the composite DRAW CIRCLE instruction to pattern pairs of circles that intersect at the desired locations for each segment, and an *AND* module to produce output species where these circles intersect. As with the DRAW FRENCH FLAG program, we include *COPY* modules where needed to prevent modules from depleting the concentrations of their input patterns and affecting upstream modules.

The combined program produces a stick figure pattern from an input pattern consisting of four points. The pattern is expected to take on the order of a day to reach steady state using the rate parameters estimated for our DNA species, provided in the **Supplementary Information**. The limiting factor preventing the speedup of formation is the diffusion rate of the DNA.

It is straightforward to rearrange components or add modules to this system to change the geometry of the output pattern. One could imagine using such a program to generate any pattern that consists of a set of line segments. While we were able to re-use some of the input signals in the DRAW STICK FIGURE program by careful use of the *COPY* module, in general each line-segment feature requires two input signals and seven modules. One interesting goal for future studies is to generate a large number of pattern features using only a fixed number of inputs and modules.

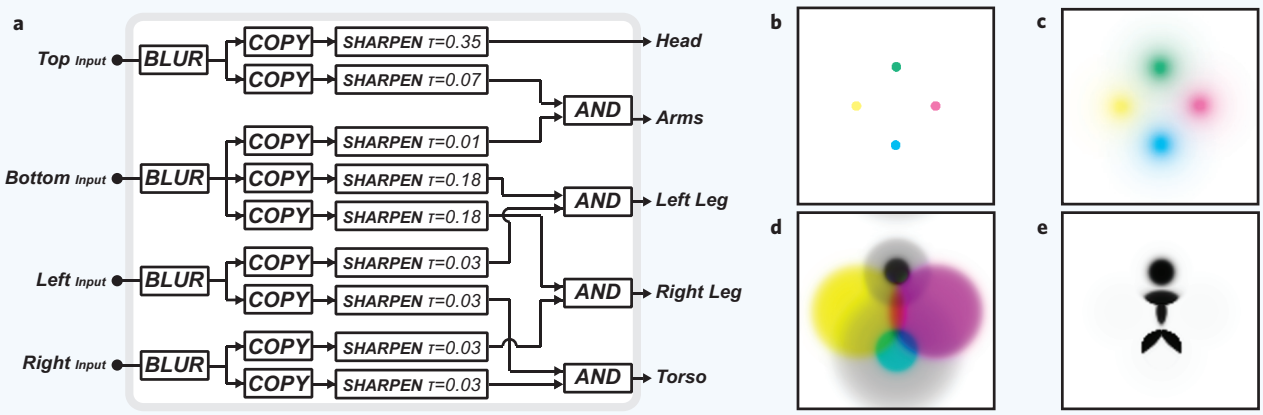


Figure 5 An RD program that produces a stick figure when the input pattern is a set of input species fixed at the illustrated locations. (**a**) The network of RD modules that interprets the signals from the input species and produces the stick figure shape as output. The magnitudes of the threshold concentrations τ for each of the *SHARPEN* modules set the respective sizes of the circles in **d**. (**b–e**) Simulated stick figure formation. (**b**) The input, four species (shown in different colors) that are each fixed at their corresponding illustrated locations. (**c**) *BLUR* modules take the four input species as inputs, producing gradients around them of controlled size. A set of *COPY* modules produces species with the same spatial extent and concentration as the original inputs. The species produced by the *COPY* modules are used as buffered inputs to the downstream modules. (**d**) A set of *SHARPEN* modules produce circles with defined radii around the input points. The intersections of these circles define the regions where the stick figure segments should appear: the magenta and yellow circles intersect at the stick-figure's torso, the yellow and cyan at the left leg, the magenta and cyan at the right leg, the two light-gray circles at the arms, and the black signal defines the stick-figure's head. (**e**) AND modules take each pair of species in (**d**) as inputs and produce the output species (black) where both inputs are present at high concentration. Simulation details, including reaction rates, diffusion coefficients, PDEs and initial conditions are provided in **Supplementary Information Section 2.2**.

DISCUSSION

In this paper we have described a method for the design of *reaction-diffusion programs*, sets of modular instructions encoded into autonomous chemical pattern generators. Each module takes simple input patterns and generates incrementally more complex output patterns. Elaborate, stable patterns of soluble molecules emerge through the combined effects of each modular stage. While the reaction-diffusion programs we describe are large, it is plausible to imagine implementing them *in vitro* using existing techniques for designing molecules with prescribed rates of reaction and diffusion: synthetic DNA strand-displacement networks of similar sizes have already been demonstrated. The continued growth in complexity and robustness of devices that can be designed with DNA strand-displacement networks suggests that the range of designed RD programs that are experimentally realizable will expand.

Further, we can imagine a "compilation" process that automatically translates arbitrary patterning programs into a network of chemical species that can be implemented using strand-displacement reactions. Specific sets of molecules can implement each module, and compilation can proceed by designing the reactions for each module such that the inputs and outputs for each module interact, but the species within different modules do not react at significant rates. Control over patterning could be extended from two into three dimensions using the existing reaction modules in 3-dimensional substrate by using an intersection of spheres instead of circles in the stick figure example, with the input points printed on, or generated from the boundary surfaces of the 3-dimensional substrate.

Eventually reaction-diffusion programs may be designed that include temporal control of network dynamics, capable of generating patterns that change shape over time, such as Belousov Zhabotinsky-type dynamics[3] or cellular automata[44]. Other potential extensions to our methods include programs that can more efficiently produce a wide variety of pattern features, and programs that produce patterns with more graded analog responses instead of only digital *high* or *low* values.

While large DNA strand-displacement networks have been implemented previously, they tend to be "one-time-use" circuits, which calculate a single output before reaching a state in which components no longer react with one another. This strategy is effective for well-mixed solutions, but reaction-diffusion patterns require a constant supply of energy to maintain a stable steady state. To provide this energy we design networks for which high-energy input species are constantly supplied and low-energy waste species are removed or degraded. Such a supply could be provided by a reservoir containing high concentrations of the necessary raw materials connected to our system directly, or if needed, through a membrane to control which molecules can pass through.

Our modules are designed around abstract chemical reactions that are not specifically constrained to any single implementation medium. In the **Supplementary Information**, we provide detailed descriptions of DNA strand-displacement species that emulate each proposed reaction, including the production and degradation reactions that supply energy to our system. However, the same abstract reactions could be compatible with a variety of other existing mechanisms[45–47]. Mechanisms that employ enzymatic activity could provide a higher energy density per molecule, consuming energy-source molecules at a lower rate compared to strand displacement networks.

An important feature of our modular design process is that patterns are produced through stages of iterative refinement. Chemical circuits found in well-studied biological development networks, such as the sea urchin network for endodermal-mesodermal differentiation[48,49] and the *Drosophila melanogaster* segment polarity network[12], exhibit many of the same basic design principles that we employ here, including the effective division of the network into modules based on their function, and the use of Boolean on/off signals to rectify concentration fluctuations, uncertainty in reaction rate constants and diffusion coefficients, or imperfectly synthesized component molecules. However, these biological circuits also appear to include feedback loops and redundant "fail-safe" layers of circuitry, in which parallel transcription pathways ensure normal operation even

when one pathway fails, making them even more robust than the networks we describe. We expect that if we were to include similar mechanisms in our synthetic systems, our RD programs could become more robust. Like networks that control body plan formation in biology, our pattern formation networks can produce multitudes of pattern variations with a limited set of components, either by changing the arrangement of modules, or by adding and removing modules. Because our networks are organized into modules with specific, well-understood functions, producing a desired new pattern is straightforward. Small portions of the pattern can be changed by changing modules that are downstream of most others in the network, while large changes to the pattern can be made by changing the modules in the network that function at the beginning of the patterning process. By modifying just a few connections or modules in a network, we could make either of these types of changes.

If reaction-diffusion programs can be readily implemented in the laboratory, they could become a flexible platform for delivering target molecules to defined portions of a substrate. We view delivered DNA as an information-carrying device; this information can translated into the form of molecules other than DNA through the use of aptamers[50]. In this context, RD programs represent spatially intelligent processors that could orchestrate complex behaviors in a growing variety of next generation biomaterials.

ACKNOWLEDGEMENTS

The authors would like to thank Paul Rothemund, Georg Seelig, Josh Fern, Abdul Mohammed, Ankur Verma, John Zenk, and the anonymous referees for insightful reading and comments. This work was supported by NSF-CCF-1161941 and a grant to the Turing Centenary Project by the John Templeton Foundation.

REFERENCES

1. Turing, A.M. The chemical basis of morphogenesis. *Philos. Trans. Roy. Soc. B* **237**, 37–72 (1952).
2. Pearson, J.E. Complex patterns in a simple system. *Science* **261**, 189–192 (1993).
3. Zhabotinsky, A.M. Periodic processes of malonic acid oxidation in a liquid phase. *Biofizika* **9**, 306–311 (1964).
4. Meinhardt, H. *The Algorithmic Beauty of Sea Shells* (Springer-Verlag, Berlin Heidelberg, 1995).
5. Epstein, I.R. & Pojman, J.R. *An Introduction to Nonlinear Chemical Dynamics* (Oxford University Press, New York, 1998).
6. Murray, J.D. *Mathematical Biology II: Spatial Models and Biomedical Applications,* 3rd Ed. (Springer, New York, 2003).
7. Qian, L. & Winfree, E. Scaling up digital circuit computation with DNA strand displacement. *Science* **332**, 1196–1201 (2011).
8. Chen, Y.-J., Daichau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D. & Seelig, D. Programmable chemical controllers made from DNA. *Nat. Nano.* **8**, 755–762 (2013).
9. Ruiza, S.A. & Chen, C.S. Microcontact printing: A tool to pattern. *Soft Matter* **3**, 168–177 (2007).
10. Du, Y., Lo, E., Ali, S. & Khademhosseini, A. Directed assembly of cell-laden microgels for fabrication of 3D tissue constructs. *Proc. Natl. Acad. Sci. USA* **105**, 9522–9527 (2008).
11. Lawrence, P.A. *The Making of a Fly: The Genetics of Animal Design* (Blackwell Scientific Publications Ltd., Oxford, UK, 1992).
12. Bastiaens, P., Caudron, M., Niethammer, P. & Karsenti, E. Gradients in the self-organization of the mitotic spindle. *Trends Cell Biol.* **16**, 125–134 (2006).
13. Lecuit, T. *et al.* Two distinct mechanisms for long-range patterning by Decapentaplegic in the *Drosophila* wing. *Nature* **381**, 387–393 (1996).
14. El-Ali, J., Sorger, P.K. & Jensen, K.F. Cells on chips. *Nature* **442**, 403–411 (2006).
15. Padirac, A., Fujii, T., Estvez-Torres, A. & Rondelez, Y. Spatial waves in synthetic biochemical networks. *J. Am. Chem. Soc.* **135**, 14586–14592 (2013).
16. Bángsági, T., Vanag, V.K. & Epstein, I.R. Tomography of reaction-diffusion microemulsions reveals three-dimensional turing patterns. *Science* **331**, 1309–1312 (2011).
17. Isalan, M., Lemerle, C. & Serrano, L. Engineering gene networks to emulate Drosophila embryonic pattern formation. *PLoS Biol.* **3**, e64 (2005).
18. Driever, W. & Nasslein-Volhard, C. A gradient of *bicoid* protein in Drosophila embryos. *Cell* **54**, 83–93 (1988).
19. Wolpert, L. Positional information and the spatial pattern of cellular differentiation. *J. Theor. Biol.* **25**, 1–47 (1969).
20. Allen, P.B., Chen, X., Simpson, Z.B. & Ellington, A.D. Modeling scalable pattern generation in DNA reaction networks. *Artif. Life* **13**, 441–448 (2012).
21. Chirieleison, S.M., Allen, P.B., Simpson, Z.B., Ellington, A.D. & Chen, X. Pattern transformation with DNA circuits. *Nat. Chem.* **5**, 1000–1005 (2013).
22. Seelig, G., Soloveichik, D., Zhang, D.Y. & Winfree, E. Enzyme-free nucleic acid logic circuits. *Science* **314**, 1585–1588 (2006).
23. Zhang, D.Y. & Seelig, G. Dynamic DNA nanotechnology using strand displacement reactions. *Nat. Chem.* **3**, 103–113 (2011).
24. L. Qian, E.W. & Bruck, J. Neural network computation with DNA strand displacement cascades. *Nature* **475**, 368–372 (2011).
25. Alon, U. *An Introduction to Systems Biology: Design Principles of Biological Circuits* (CRC Press, Boca Raton, Florida, 2006).
26. Soloveichik, D., Seelig, G. & Winfree, E. DNA as a universal substrate for chemical kinetics. *Proc. Natl. Acad. Sci. USA* **107**, 5393–5398 (2010).
27. Zhang, D.Y. & Winfree, E. Control of DNA strand displacement kinetics using toehold exchange. *J. Am. Chem. Soc.* **131**, 17303–17314 (2009).
28. Zadeh, J.N. *et al.* NUPACK: Analysis and design of nucleic acid systems. *J. Comput. Chem.* **32**, 170–173 (2011).
29. Phillips, A. & Cardelli, L. A programming language for composable DNA circuits. *J. R. Soc. Interface* **9**, S419–S436 (2009).
30. Smith, D.E., Perkins, T.T. & Chu, S. Dynamical scaling of DNA diffusion coefficients. *Macromolecules* **29**, 1372–1373 (1996).
31. Kenney, M., Ray, S. & Boles, T.C. Mutation typing using electrophoresis and gel-immobilized acrydite probes. *Biotechniques* **25**, 516–521 (1998).
32. Allen, P.B., Chen, X. & Ellington, A.D. Spatial control of DNA reaction networks by DNA sequence. *Molecules* **17**, 13390–13402 (2012).
33. SantaLucia, J., Jr. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc. Natl. Acad. Sci. USA* **95**, 1460–1465 (1998).
34. Ng, E.W. *et al.* Pegaptanib, a targeted anti-VEGF aptamer for ocular vascular disease. *Nat. Rev. Drug Disc.* **5**, 123–132 (2006).
35. Famulok, M. Oligonucleotide aptamers that recognize small molecules. *Curr. Opto. Struct. Biol.* **9**, 324–329 (1999).
36. Baker, B.R. *et al.* An electronic, aptamer-based small-molecule sensor for the rapid, label-free detection of cocaine in adulterated samples and biological fluids. *J. Am. Chem. Soc.* **128**, 3138–3139 (2006).
37. An, C.-I., Trinh, V.B. & Yokobayashi, Y. Artificial control of gene expression in mammalian cells by modulating RNA interference through aptamer-small molecule interaction. *RNA* **12**, 710–716 (2006).
38. Dirks, R.M. & Pierce, N.A. Triggered amplification by hybridization chain reaction. *Proc. Natl. Acad. Sci. USA* **101**, 15275–15278 (2004).
39. Horn, F. & Jackson, R. General mass action kinetics. *Arch. Ration. Mech. Anal.* **47**, 81–116 (1972).
40. Stellwagen, E. & Stellwagen, N.C. Determining the electrophoretic mobility and translational diffusion coefficients of DNA molecules in free solution. *Electrophoresis* **23**, 2794–2803 (2002).
41. Gierer, A. & Meinhardt, H. A theory of biological pattern formation. *Kybernetik* **12**, 3039 (1972).
42. Jeon, N.L. *et al.* Generation of solution and surface gradients using microfluidic systems. *Langmuir* **16**, 8311–8316 (2000).
43. Allen, P.B., Chen, X., Simpson, Z.B. & Ellington, A.D. Modeling scalable pattern generation in DNA reaction networks. In *Artificial Life*, Vol. 13, pp. 441–448 (2012).
44. Burks, A.W. *Von Neumann's Self-Reproducing Automata* (University of Michigan, 1969).
45. Nitta, N. & Suyama, A. Autonomous biomolecular computer modeled after retroviral replication. In *DNA Computing* (Springer, Berlin Heidelberg, 2004), pp. 203–212.
46. Kim, J., White, K.S. & Winfree, E. Construction of an *in vitro* bistable circuit from synthetic transcriptional switches. *Mol. Syst. Biol.* **2**, 68 (2006).
47. Montagne, K., Plasson, R., Sakai, Y., Fujii, T. & Rondelez, Y. Programming an *in vitro* DNA oscillator using a molecular networking strategy. *Mol. Syst. Biol.* **7**, 466 (2011).
48. Smith, J. & Davidson, E.H. Regulative recovery in the sea urchin embryo and the stabilizing role of fail-safe gene network wiring. *Proc. Natl. Acad. Sci. USA* **106**, 18291–18296 (2009).
49. Levine, M.S. & Davidson, E.H. Gene regulatory networks for development. *Proc. Natl. Acad. Sci. USA* **102**, 4936–4942 (2005).
50. Jhaveri, S., Rajendran, M. & Ellington, A.D. *In vitro* selection of signaling aptamers. *Nat. Biotechnol.* **18**, 1293–1297 (2000).

SUPPLEMENTARY INFORMATION

1 MODULE DETAILS

1.1 *SHARPEN* module

A homogeneously produced threshold species T rapidly reacts with an input I, reducing $[I]$ to zero where I is produced more slowly than T. A homogeneously produced amplification species A reacts more slowly

with excess I leftover from the thresholding reaction, producing O. The concentration of O is stabilized by a degradation reaction.

Below are the *PDEs* that govern how species are affected by this module. These are only the terms that result from the module's interaction with each species. For species on the module's interface (*i.e.* input and output species), additional external terms may affect the behavior of the species, and must be added to these equations in context. Specifically, we do not include production or degradation reactions on the input signal *PDEs*, assuming that these reactions are handled by the upstream circuit.

$$\frac{\partial[I](t,x,y)}{\partial t} = D_I\nabla^2[I](t,x,y) - k_{react,T}[I](t,x,y)[T](t,x,y)$$

$$\frac{\partial[T](t,x,y)}{\partial t} = D_T\nabla^2[T](t,x,y) + k_{prod,T} - k_{degrade,T}[T](t,x,y) - k_{react,T}[I](t,x,y)[T](t,x,y)$$

$$\frac{\partial[A](t,x,y)}{\partial t} = D_A\nabla^2[A](t,x,y) + k_{prod,A} - k_{degrade,A}[A](t,x,y) - k_{react,A}[I](t,x,y)[A](t,x,y)$$

$$\frac{\partial[O](t,x,y)}{\partial t} = D_O\nabla^2[O](t,x,y) - k_{degrade,O}[O](t,x,y) + k_{react,A}[I](t,x,y)[A](t,x,y)$$

Initial conditions:

$$[T](0,x,y) = [A](0,x,y) = [O](0,x,y) = 0\ \mu M$$

Figure 3a in the main text uses periodic boundary conditions and a 24 hour simulation time. In addition to the *PDEs* defined above, the following equation determines how we modeled the upstream behavior of the input signal I for the specific context in **Fig. 3a**. This equation must be added together with the terms that describe how the module affects the input signal in order to model the full behavior of the input signal in **Fig. 3a**.

$$\frac{\partial[I](t,x,y)}{\partial t} = k_{prod,I}[S](t,x,y) - k_{degrade,I}[I](t,x,y)$$

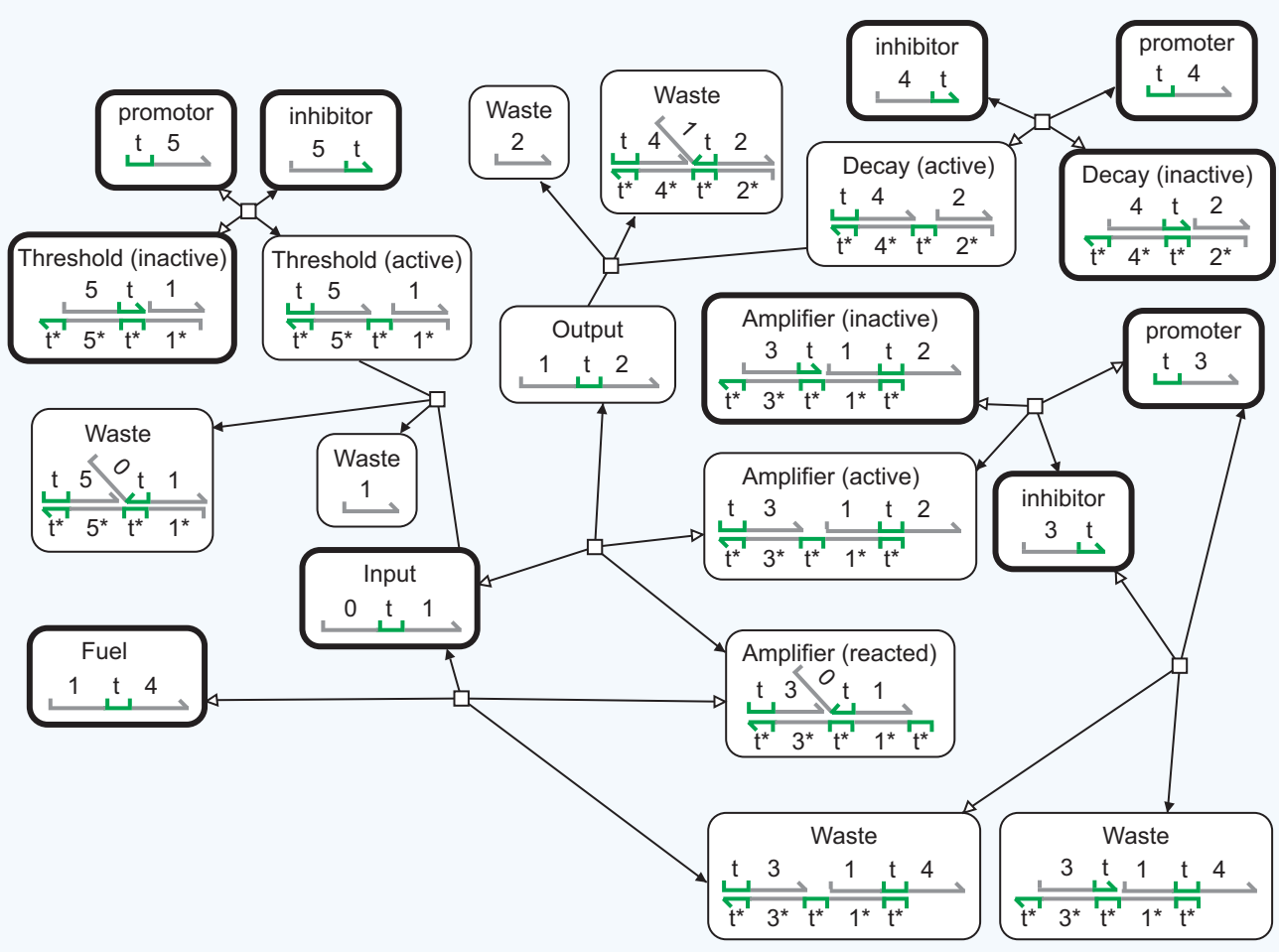
$$[I](0,x,y) = 0\ \mu M$$

with static source species S governed by

$$\frac{\partial[S](t,x,y)}{\partial t} = 0$$

$$[S](0,x,y) = e^{-\frac{(x^2+y^2)}{8}}\ \mu M$$

Also in this simulation, $k_{prod,A} = \alpha k_{degrade,A}$ and $k_{prod,T} = \tau k_{degrade,T}$, where the thresholding set point $\tau = 0.5$, and the amplification set point



SI Figure 1 *SHARPEN* module translated into strand-displacement reactions. Boxes contain domain-level definitions of each DNA species involved in the reaction network. Species that are defined in the abstract chemical reaction definition for this module have bold line weights on their boxes, while all other species are intermediate species necessary to emulate the designed abstract chemical reaction. Arrows connecting the involved species represent reactions between species. Black arrowheads designate forward reactions, and white arrowheads designate reverse reactions.

$\alpha = 1$. All other cycling (*i.e.* production and degradation) reactions use $k_{cycling} = 0.002\ s^{-1}$. Thresholding reaction rate constants are $k_{thresholding} = 20\ \mu M^{-1}s^{-1}$, and all remaining reaction rate constants are $k_{react} = 0.2\ \mu M^{-1}s^{-1}$. The diffusion coefficient for all diffusing species is $D = 0.000150\ mm^2\ s^{-1}$. We selected all of the diffusion rates used in our simulations to be the same order of magnitude as experimentally derived data from the literature[S1,S2].

The DNA strand-displacement network outlined in **SI Fig. 1** implements this module. This network uses 11 initial species complexes, consisting of 14 unique strands of DNA (including the input strand).

1.2 *COPY* module

An input species I catalyzes the production of an output species O. The input species is not produced, consumed, or significantly affected by this module. To produce a stable steady state, O also is degraded.

PDEs that govern how species are affected by this module:

$$\frac{\partial[O](t,x,y)}{\partial t} = D_O\nabla^2[O](t,x,y) + k_{prod,O}[I](t,x,y) - k_{degrade,O}[O](t,x,y)$$

Initial conditions:

$$[O](0,x,y) = 0\ \mu M$$

The production and degradation rates of O are equal, so if the time scale of diffusion is sufficiently slower than the time scale of the reaction kinetics, the equation governing $[O]$ is well approximated by

$$\frac{\partial[O](t,x,y)}{\partial t} \approx k_{cycling,O}([I](t,x,y) - [O](t,x,y)),$$

and at steady state, $[O](t, x, y) = [I](t, x, y)$.

Figure 3b in the main text uses periodic boundary conditions, a 24 hour simulation time, and

$$[I](t,x,y) = e^{-\frac{(x^2+y^2)}{8}}\ \mu M,$$

with reaction rate constants $k_{prod,O} = 0.002\ s^{-1}$ and $k_{degrade,O} = 0.002\ s^{-1}$, and diffusion coefficients $D_I = D_O = 0.000150\ mm^2\ s^{-1}$.

The DNA strand-displacement network outlined in **SI Fig. 2** implements this module. This network uses 6 initial species complexes, consisting of 8 unique strands of DNA (including the input strand).

1.2 *AND* module

PDEs that govern how species are affected by this module:

$$\frac{\partial[I_1](t,x,y)}{\partial t} = D_{I_1}\nabla^2[I_1](t,x,y) - k_{react,M}[I_1](t,x,y)[M](t,x,y)$$

$$\frac{\partial[I_2](t,x,y)}{\partial t} = D_{I_2}\nabla^2[I_2](t,x,y) - k_{react,M}[I_2](t,x,y)[M](t,x,y)$$

$$\frac{\partial[M](t,x,y)}{\partial t} = D_M\nabla^2[M](t,x,y) + k_{prod,M} - k_{degrade,M}[M](t,x,y) - k_{react,M}[I_1](t,x,y)[M](t,x,y) - k_{react,M}[I_2](t,x,y)[M](t,x,y)$$

$$\frac{\partial[N](t,x,y)}{\partial t} = D_N\nabla^2[N](t,x,y) - k_{degrade,N}[N](t,x,y) + k_{react,M}[I_1](t,x,y)[M](t,x,y) + k_{react,M}[I_2](t,x,y)[M](t,x,y)$$

Initial conditions:

$$[M](0,x,y) = [N](0,x,y) = 0\ \mu M$$

Summed species N is then passed through a *SHARPEN* module with threshold concentration $\tau = 1.3$ and amplify concentration $\alpha = 1$.

Figure 3c in the main text uses periodic boundary conditions, a 24 hour simulation time, and

$$\frac{\partial[I_1](t,x,y)}{\partial t} = k_{prod,I}[S_1](t,x,y) - k_{degrade,I}[I_1](t,x,y)$$

and

$$\frac{\partial[I_2](t,x,y)}{\partial t} = k_{prod,I}[S_2](t,x,y) - k_{degrade,I}[I_2](t,x,y)$$

starting from $[I_1](0, x, y) = [I_2](0, x, y) = 0\ \mu M$, with source species S_1 and S_2 governed by

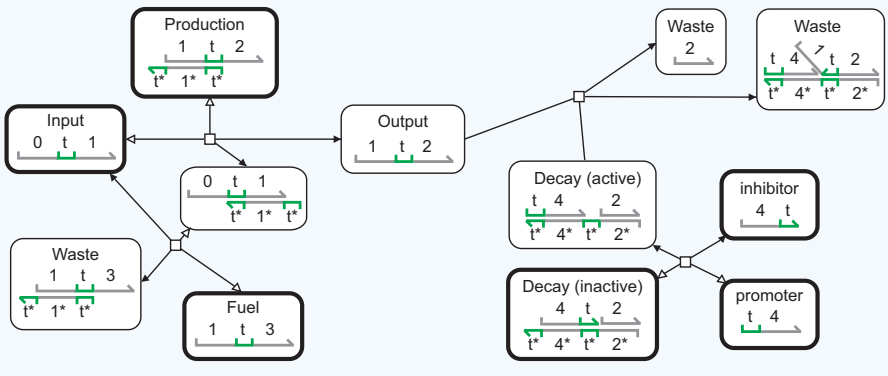
$$\frac{\partial[S_1](t,x,y)}{\partial t} = \frac{\partial[S_2](t,x,y)}{\partial t} = 0$$

$$[S_1](0,x,y) = \begin{cases} 1\ \mu M, & if\ (x+1.5)^2 + (y)^2 < 3^2 \\ 0\ \mu M, & otherwise. \end{cases}$$

$$[S_2](0,x,y) = \begin{cases} 1\ \mu M, & if\ (x-1.5)^2 + (y)^2 < 3^2 \\ 0\ \mu M, & otherwise. \end{cases}$$

Also in this simulation $k_{prod,M} = 2\beta k_{degrade,M}$, where the set point $\beta = 1$. All other cycling (*i.e.* production and degradation) reactions use $k_{cycling} = 0.002\ s^{-1}$. Thresholding reaction rate constants are $k_{thresholding} = 20\ \mu M^{-1}s^{-1}$, and all remaining reaction rate constants are $k_{react} = 0.2\ \mu M^{-1}\ s^{-1}$. The diffusion coefficient for all diffusing species is $D = 0.000150\ mm^2\ s^{-1}$.

The DNA strand-displacement network outlined in **SI Fig. 3** implements this module. This network uses 14 initial species complexes, consisting of 19 unique strands of DNA (including the input strand).



SI Figure 2 *COPY* module translated into strand-displacement reactions.

1.4 *NOT* module

PDEs that govern how species are affected by this module:

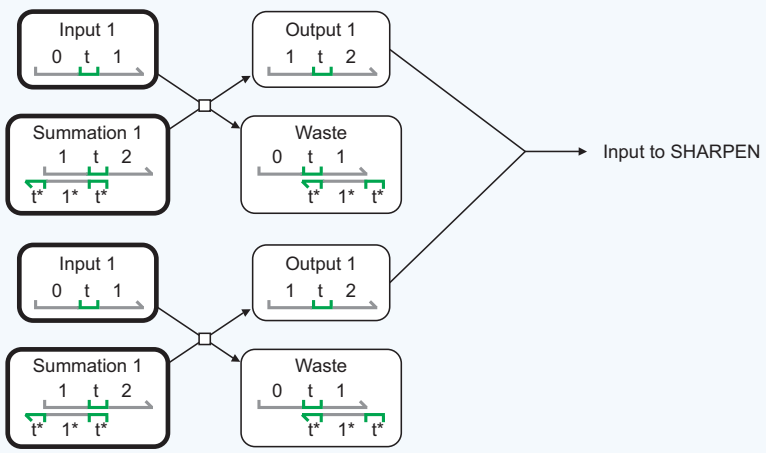
$$\frac{\partial[I](t,x,y)}{\partial t} = D_I\nabla^2[I](t,x,y) - k_{react}[I](t,x,y)[O](t,x,y)$$

$$\frac{\partial[O](t,x,y)}{\partial t} = D_O\nabla^2[O](t,x,y) + k_{prod,O} - k_{degrade,O}[O](t,x,y) - k_{react}[I](t,x,y)[O](t,x,y)$$

Initial conditions:

$$[O](0,x,y) = 0\ \mu M$$

Figure 3d in the main text uses periodic boundary conditions, a 24 hour simulation time, and



SI Figure 3 AND module translated into strand-displacement reactions.

$$\frac{\partial[I](t,x,y)}{\partial t} = k_{prod,I}[S](t,x,y) - k_{degrade,I}[I](t,x,y)$$

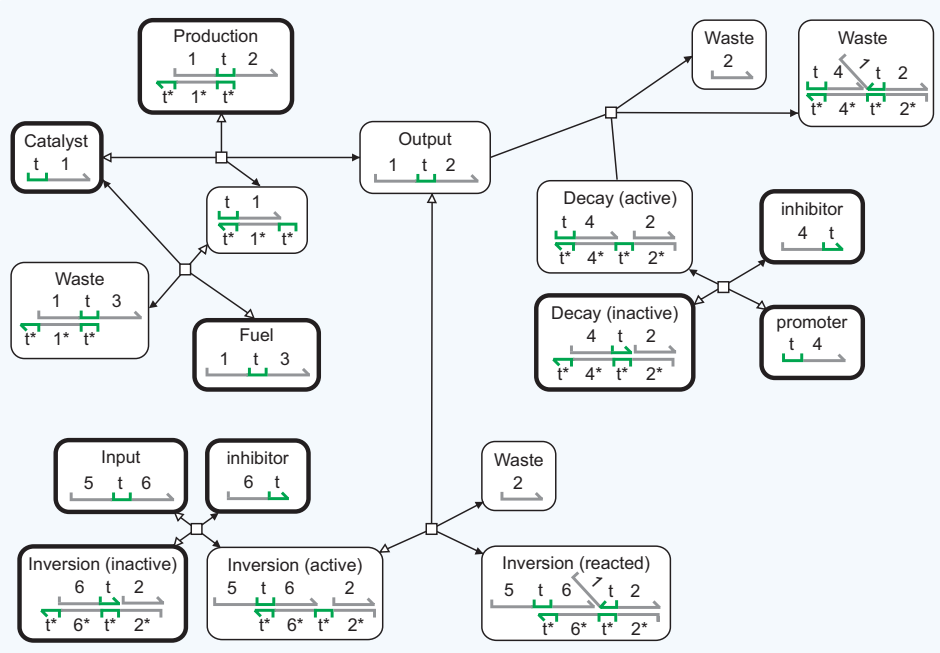
starting from $[I](0, x, y) = 0$ μM, with source species S governed by

$$\frac{\partial[S](t,x,y)}{\partial t} = 0$$

$$[S](0,x,y) = \begin{cases} 1\,\mu\text{M}, & \text{if } x^2 + y^2 < 4^2 \\ 0\,\mu\text{M}, & \text{otherwise.} \end{cases}$$

Also in this simulation $k_{prod,O} = \alpha k_{degrade,O}$, where $\alpha = 1$. All other cycling (*i.e.* production and degradation) reactions use $k_{cycling} = 0.002\ \text{s}^{-1}$. Thresholding reaction rate constants are $k_{thresholding} = 20\ \mu\text{M}^{-1}\,\text{s}^{-1}$, and the inversion reaction rate constant between I and O is $k_{react} = 0.2\ \mu\text{M}^{-1}\text{s}^{-1}$. The diffusion coefficient for all diffusing species is $D = 0.000150\ \text{mm}^2\,\text{s}^{-1}$.

The DNA strand-displacement network outlined in **SI Fig. 4** implements this module. This network uses 8 initial species complexes, consisting of 12 unique strands of DNA (including the input strand).



SI Figure 4 *NOT* module translated into strand-displacement reactions.

1.5 *BLUR* module

Input species I catalyzes the production of an output species O. The input species is not produced, consumed, or otherwise affected by this module. O diffuses away from locations where it is produced, and is degraded to produce a blurred copy of the input patter at steady-state concentration.

PDEs that govern how species are affected by this module:

$$\frac{\partial[O](t,x,y)}{\partial t} = D_O\nabla^2[O](t,x,y) + k_{prod,O}[I](t,x,y) - k_{degrade,O}[O](t,x,y)$$

Initial conditions:

$$[O](0, x, y) = 0\ \mu\text{M}$$

If the input pattern is a single reference point of high concentration I, centered at (x_0, y_0) with radius r, where

$$[I](0,x,y) = \begin{cases} 1\,\mu\text{M}, & \text{if } (x-x_0)^2 + (y-y_0)^2 < \gamma^2 \\ 0\,\mu\text{M}, & \text{otherwise.} \end{cases}$$

Then the resulting output pattern is a gradient where $[O]$ drops off monotonically with radial distance from the fixed reference point. The amplitude and width of this curve can be rationally turned, as demonstrated in **SI Fig. 5.**

Figure 3e in the main text uses periodic boundary conditions, a 24 hour simulation time, $r = 0.5$ mm, $x_0 = 0$ mm, $y_0 = 0$ mm, reaction rate constants $k_{prod,O} = 1\ \text{s}^{-1}$ and $k_{degrade,O} = 0.001\ \text{s}^{-1}$, and diffusion coefficient $D_O = 0.000100\ \text{mm}^2\,\text{s}^{-1}$.

The DNA strand-displacement network outlined in **SI Fig. 6** implements this module. This network uses 6 initial species complexes, consisting of 8 unique strands of DNA (including the input strand).

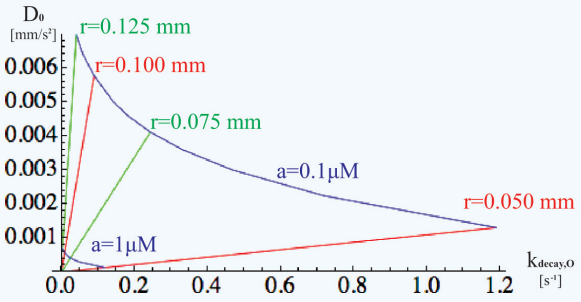
2. PROGRAM DETAILS

2.1 French flag program

The simulations for the French flag program (main text subsection) were simulated over 24 hours on x : [0, 10] mm with reflective boundary conditions. The input pattern I for this program is a linear gradient,

$$[I](x,y) = \frac{x}{10}\ \mu\text{M}.$$

Cycling (*i.e.* production and degradation) reactions use $k_{cycling} = 0.002\ \text{s}^{-1}$. Thresholding reaction rate constants are $k_{thresholding} = 20\ \mu\text{M}^{-1}\text{s}^{-1}$, and all other reaction rate constants are set to $k_{react} = 0.2\ \mu\text{M}^{-1}\,\text{s}^{-1}$. The diffusion coefficient for all diffusing species is $D = 0.000150\ \text{mm}^2\,\text{s}^{-1}$.



SI Figure 5 For the *BLUR* module simulation in **Fig. 3e** in the main text: By holding $k_{prod,O} = 1\ s^{-1}$ and tuning the values of $k_{degrad,O}$ and D_O, a diverse range of output signal amplitudes a and half-amplitude radii r are produced.

The threshold concentration $\tau = 0.667$ for the high concentration *SHARPEN* module that produces species S_1. The threshold concentration $\tau = 0.333$ for the medium concentration *SHARPEN* module that produces species S_2. All "on" reference concentration parameters (α, β) are set to 1 μM.

One way to measure the complexity of a program is to count the number of DNA strands or initial complexes it requires for implementation. The French flag program uses one input species (1 strand or 1 complex), six *COPY* modules (6 strands or 8 complexes each), two *SHARPEN* modules (11 strands or 14 complexes each), two *NOT* gates (8 strands or 12 complexes), and an *AND* gate (14 strands or 19 complexes each), however we need to subtract the input species (1 for the *COPY*, 1 for the *SHARPEN*, 1 for the *NOT*, and 2 for the *AND*) since they are already counted as the output of the upstream module. This results in a count of 77 initial complexes or 108 component DNA strands for the French flag program.

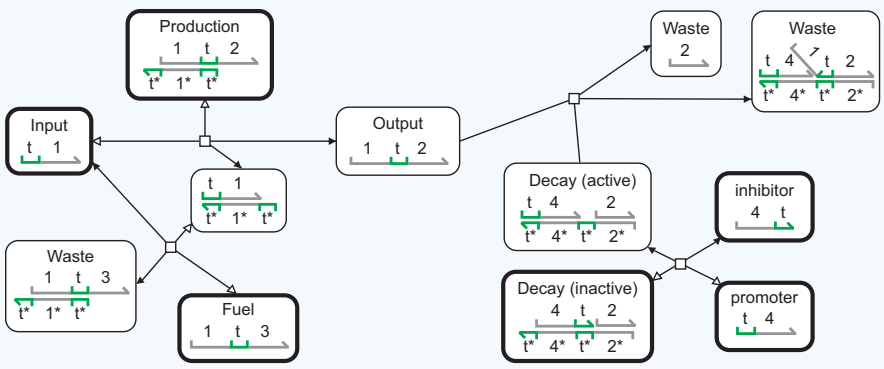
2.2 Stick figure program

The simulations for the stick figure program (main text subsection) were simulated over 24 hours on x: [–10, 10] mm, y: [–10, 10] mm with reflective boundary conditions. *SHARPEN* modules used production rate constant $k_{prod,SHARPEN} = 0.2/300\ s^{-1}$ and $k_{degrade,SHARPEN} = 0.2/5000\ s^{-1}$. All other cycling (*i.e.* production and degradation) reactions use $k_{cycling} = 0.002\ s^{-1}$. Thresholding reaction rate constants are $k_{thresholding} = 20\ \mu M^{-1}\ s^{-1}$, and all other reaction rate constants are set to $k_{react} = 0.2\ \mu M^{-1}\ s^{-1}$. The diffusion coefficient for all diffusing species is $D = 0.000150\ mm^2\ s^{-1}$.

The inputs for this simulation were four points with radius r = 0.5 mm, of top, bottom, left or right input species at [*species*] = 1 μM. These points were located 4 mm away from the origin in the positive-x, negative-x, negative-y and positive-y directions, respectively. The reference threshold concentrations for the *SHARPEN* modules are as follows (these are the result of fractional approximations and do not require this level of precision):

1. Top circle used for head patterning: τ = 0.346525
2. Top circle used for arms patterning: τ = 0.0704974
3. Bottom circle used for arms patterning: τ = 0.00746513
4. Left and right circles used for torso and legs patterning: τ = 0.0296572
5. Bottom circle used for legs patterning: τ = 0.183469

All "on" reference concentration parameters (α, β) are set to 1 μM. The stick figure program uses four input species (1 strand or 1 complex each), four *BLUR* modules (6 strands/8 complexes each), nine *COPY* modules (6 strands or 8 complexes each), nine *SHARPEN* (11 strands or 14 complexes each), and four *AND* gate (14 strands or 19 complexes each). Subtracting the inputs (2 strands/species for the *AND* module and 1 strand/species for all other modules), this results in a count of 207 initial complexes or 280 component DNA strands for the stick figure program.



SI Figure 6 *BLUR* module translated into strand-displacement reactions.

REFERENCES

S1. Lukacs, G.L., Haggie, P., Seksek, O., Lechardeur, D., Freedman, N. & Verkman, A.S. Size-dependent DNA mobility in cytoplasm and nucleus. *Journal of Biological Chemistry* **275**, 1625 (2000).

S2. Stellwagen, E., Lu, Y. & Stellwagen, N.C. Unified description of electrophoresis and diffusion for DNA and other polyions. *Biochemistry* **42**, 11745 (2003).