

Fellermann H, Lopiccolo A, Kozyra J, Krasnogor N.

[In vitro implementation of a stack data structure based on DNA strand displacement.](#)

*In: Unconventional Computation and Natural Computation: 15th International Conference, UCNC 2016.*

11-15 July 2016, Manchester, UK: Springer.

**Copyright:**

The final publication is available at Springer via <http://dx.doi.org/10.1007/978-3-319-41312-9>

**Date deposited:**

27/04/2016



This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](#)

# In vitro implementation of a stack data structure based on DNA strand displacement

Harold Fellermann,\* Annunziata Lopiccolo\*, Jerzy Kozyra, and Natalio Krasnogor†  
*Interdisciplinary Computing and Complex Biosystems Research Group*  
*School of Computing, Newcastle University*  
*Newcastle-upon-Tyne UK*

We present an implementation of an in vitro signal recorder based on DNA assembly and strand displacement. The signal recorder implements a stack data structure in which both data as well as operators are represented by single stranded DNA “bricks”. The stack grows by adding push and write bricks and shrinks in last-in-first-out manner by adding pop and read bricks. We report the design of the signal recorder and its mode of operations and give experimental results from capillary electrophoresis as well as transmission electron microscopy that demonstrate the capability of the device to store and later release several successive signals. We conclude by discussing potential future improvements of our current results.

## I. INTRODUCTION

DNA nanotechnology is now a well established method for arranging and controlling matter on the nanoscale [1]. Because of the relative ease with which molecular folding and molecular interactions can be designed by choosing appropriate nucleic acid sequences, DNA is a prominent substrate for designing artificial reaction networks with designed functionality. In particular, it has been shown that arbitrary chemical reaction networks can be translated into equivalent toehold mediated DNA strand displacement systems up to a constant scaling factor that accounts for the relatively slow speed of DNA reorganization reactions [2]. Among the broadest application areas of such designed chemistries is the area of molecular computing, where complex reaction networks consisting of dozens of molecular players with well defined interactions can be readily synthesized and tested in the laboratory.

Recent years have seen theoretical designs and molecular realizations of conventional and unconventional molecular computational circuits. The majority of this work has been concerned with implementing logic gates such as Boolean gates [3, 4], join-and-fork gates [5, 6] and seesaw gates [7, 8], and wiring these gates together to create circuits of increasing complexity, such as molecular adders [9], static lookup tables [10] and game-playing molecular automata [11].

This approach toward molecular computing, which closely mimics electrical engineering, is somewhat disconnected from algorithmic computer science, where algorithms are built by composing data and operations. Indeed, DNA computing has so far seen few designs for DNA data structures – with Qian et al.’s theoretical design of a DNA based stack machine being one noteworthy exception [12].

Here, we present the *in vitro* implementation and ex-

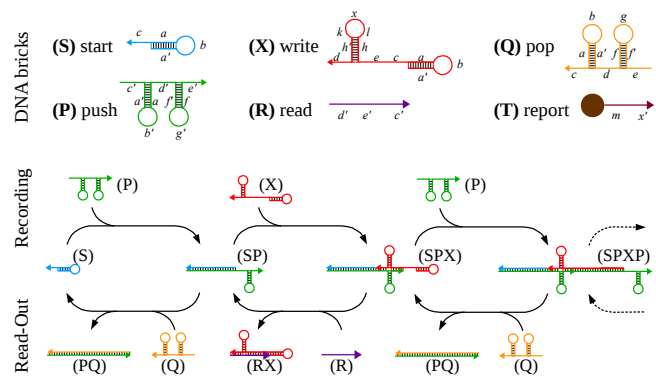


FIG. 1. Schematic of the DNA recorder. The top row shows schematics of the individual ssDNA bricks. Arrows indicate 5' → 3' direction. Below are the modes of operation to record (middle row) and read out (bottom row) signals from the stack.

perimental characterization of a DNA data structure, namely a *stack*, where data and operations form the core of the molecular interaction network. Our design shares similarities with the one presented by Qian et al. but has been optimized for maximal robustness among all molecular interactions and minimal occurrence of undesirable reactions. The stack data structure is here employed as a signal recorder and its recording and readout fidelity is characterized experimentally. We understand this contribution as a stepping stone toward *in vitro* implementations of more general data structures, as well as computationally universal stack machines. To the best of our knowledge, our work provides the first experimental results on a DNA based stack in particular, and DNA based data structures in general.

## II. A STACK DATA STRUCTURE BUILT FROM DNA

A stack is an abstract data structure that serves as a linear collection of elements, with two principal opera-

\* both HF and AL contributed equally to this work

† natalio.krasnogor@newcastle.ac.uk

tions: *push* adds an element to the stack, and *pop* removes the most recently added element that was not yet removed. Formally, this is achieved through the interface

$$\begin{aligned} \textit{push} &: \text{stack} \times \text{element} \longrightarrow \text{stack} \\ \textit{pop} &: \text{stack} \longrightarrow \text{stack} \times \text{element} \end{aligned}$$

with the invariant

$$\textit{pop}(\textit{push}(\text{stack}, \text{element})) = \text{stack}, \text{element}$$

to guarantee *last-in-first-out* operation.

Fully implementing this data type in DNA requires molecular realizations of the assembled stack, all potential elements, as well as the push and pop operations. We achieve this by associating each data element and each operation with a single stranded DNA (ssDNA) strand with partial secondary structures. We call those strands “bricks”. The stack data structure is built from bricks via hybridization of complementary DNA domains. More precisely, the stack forms a double stranded DNA (dsDNA) assembly with essentially no single stranded regions but one active toehold domain, that offers an entry for operation. Data bricks form the top strand and push bricks form the bottom strand of this dsDNA assembly.

To prevent run-away processes that might occur when adding bricks in realistic concentrations, we design the device to toggle between two states in all modes of operation. We refer to these as *data state* and *operator state*. When the stack is in data state, it will accept a single data brick. Upon binding this data element, the device toggles into the operator state in which it cannot further interact with data bricks, but instead awaits a new operator brick such as *push*. Again, only a single operator brick is accepted, and by interacting with it, the stack toggles back into the data state.

Our design differs from the one proposed by Qian et al. [12] in several important aspects:

1. We implement all data and operations as single DNA strands, whereas Qian et al. employ bricks of up to three DNA strands.
2. Our assembled DNA stack is entirely double stranded and does not feature any dangling single stranded overhangs, which are used by Qian et al. to store the actual data elements.
3. Instead, in our design data is encoded in internal secondary structure motifs in the double strand, namely in hairpin loops that form holiday junctions.
4. Our modes of operation are based on DNA interactions that are effectively irreversible at the operating temperature. Qian et al.’s design, in contrast, employs only reversible interactions and relies on detailed balance to drive the device from one configuration into another.

We have taken these design decisions, in order to minimize the amount of required distinct DNA sequences and to obtain maximally robust modes of operation, especially when envisioning ultimate *in vivo* applications.

## A. Data and Operator Brick Design

Our signal recorder operates with six distinct DNA bricks and is able to store combinations of two different signals, encoded by two types of data elements. Two further bricks are added for experimental analysis. See Fig. 1 for a schematic representation of the employed bricks and their interactions.

- *Start* (S): data brick designating the beginning of the recorder tape. It features a toehold domain for interaction with *push* and a hairpin motif at the 5’ end. This hairpin undergoes branch migration with a complementary hairpin in *push* but is otherwise not functional in the current design.
  - *Push* (P): operator brick to initiate subsequent signal recording. The brick contains the complementary toehold for interaction with *start*, a hairpin motif complementary to the one in *start*, a second hairpin for structural reasons that does not participate in branch migration, and two toehold domains, one on each side of the structural hairpin, to bind *write* bricks.
  - *Write* (X/Y): data bricks that can be stored in the recorder. These bricks contain two toehold domains complementary to the push toeholds, a structural hairpin that does not undergo branch migration, plus the same toehold domain and 5’ hairpin that form the *start* brick. Toehold domains and branch migration hairpins are identical for all types of *write* bricks. Thus, they can only differ in their structural hairpin motif. Since these hairpins do not participate in hybridization or branch migration, they can be functionalized to host any desired functionality such as recognition sites for DNA binding proteins.
- We employ two different types of *write* bricks, denoted as *write-X* and *write-Y*. *Write-Y* features a longer hairpin stem than *write-X* (twenty-five base pairs against ten base pairs) and has a different sequence in its stem loop. Although we currently employ binary data (*X* or *Y*), the approach is intrinsically *n*-ary.
- *Pop* (Q): data brick that undoes the rightmost *push* operation. This brick is the exact complement of *push*
  - *Read* (R): operator brick that removes the rightmost *write* operation. The brick is the complement of all toehold domains used in *write*’s. Notably, it

does not contain any domains that interact with the structural hairpin of *write* bricks.

- *Report* (T): non-essential bricks for experimental analysis. Report bricks do not participate directly in the operations of the stack recorder. Instead, they interact with the data domains of structural hairpins in the *write* bricks. *Report* bricks can be added to the device in any configuration since their binding sites in the data hairpins are always accessible and since they do not interfere with the operating modes of the device.

In this study, we use linear *report* strands that are 5' biotinylated via a 2.6 nm tetra-ethyleneglycol (TEG) spacer. We functionalized these *report* bricks with streptavidin coated gold nanoparticles of different diameters, which allows for easy recognition using transmission electron microscopy (TEM).

Domain sizes have been chosen with the following objectives: toeholds are long enough to span a single helical turn when hybridized with their complements (10 nt) which should promote irreversible hybridization. Hairpin loops that participate in branch migration are long enough to promote stable stems (6 base pair stems with 4-5 nt loops) but short enough to obtain quick branch migration times. The structural hairpin loop of *write* bricks together with the unpaired domain of *report* are long enough to accommodate 5 nm and 10 nm diameter nanoparticles in close vicinity to the device.

## B. Modes of Operation

DNA hybridization, branch migration and strand displacement are the three processes governing all DNA interactions involved in the system. All reactions are energetically downhill, driven by the binding energy of the closing toehold domains.

### 1. Recording

A schematic of the recording process is shown in Fig. 1 middle row. Starting from an empty stack, which is represented by the *start* brick (S), the device is toggled into its data state by providing a *push* operator (P). The *start-push* interaction begins by irreversibly binding toehold *c* and continues via branch migration among the two complementary *aba'* domains. The stack is now in its data state (SP), where a single open toehold region (*d'e'*) can recruit a *write* brick (X or Y). The *write* will partially hybridize with the *d'e'* *push* toeholds, thus toggling the stack back into its operator state (SPX). In this state, the stack exposes the same toehold-hairpin interface that characterizes the *start* brick, which allows the device to undergo subsequent rounds of recording.

Note that the assembled stack is essentially double stranded with a single exposed toehold domain. Because the structural hairpins of neither the *push* nor the *write* participate in branch migration, the stack will form holiday junctions for each recorded data element. As data specific domains are encoded in the loop regions of this holiday junction, the recording cycle is independent on the actual data written.

### 2. Read-Out

While recording proceeds from left to right in the schematic, read-out will proceed from right to left, thereby undoing any recording in the last-in first-out manner required by the stack specification. The read-out cycle is schematically presented in the bottom row of Fig. 1.

In operator state (SPX), providing a *read* brick (R) will peel the last recorded *write* brick off the stack, thereby toggling the device back into the data state (SP). This reaction proceeds in two steps: first, the *read* brick hybridizes to the stack at its unique exposed *c* domain. Secondly, the dangling *d'e'* domains of the *read* brick initiate a three-way branch migration with the *d'e'* domains of the adjacent *push* brick against the *de* domains of the *write* brick, until the *push* strand is completely displaced.

Note that the data hairpin of the *write* brick does not participate in the branch migration. This ensures that a unique *read* brick can interact with any *write* brick, ensuring that data elements can be read from the recorder without a need to know which information has been stored. The resulting *read-write* complex (RX) does not expose any single stranded domains and will not participate in further DNA interactions.

In its data state (SP), the stack can either be extended again with another data element by switching to the recording operation, or reading can be completed by toggling the stack back into its operator state. The latter is done by providing a *pop* brick (Q) that will interact with and peel off the exposed *push* brick. Analogue to the previous reaction, *pop-push* interactions are composed of their initial irreversible toehold hybridization, subsequent branch migration and eventual strand displacement. Again, the resulting *push-pop* complex (PQ) is completely double stranded and will not participate in further DNA interactions.

## III. METHODS

### A. Primary Sequence Specification

In the past we have successfully utilized evolutionary algorithms for evolving nano scale and self-assembling systems [13–15]. Thus we resorted to genetic algorithms to obtain nucleic acid sequences for all specified domains in the DNA stack specification. The fitness function of

domain	sequence
<i>a</i>	TCTCCC
<i>b</i>	GCCA
<i>c</i>	GCACACACTTC
<i>d</i>	ACACCACTTC
<i>e</i>	GGGAGACCAA
<i>f</i>	CGGCGG
<i>g</i>	CTGCC
<i>h<sub>x</sub></i>	ATTAGTAGGT
<i>h<sub>y</sub></i>	GCACGCTCGAGCTCGTATCGCAGTA
<i>k<sub>x</sub></i>	CTCTAATCAC
<i>k<sub>y</sub></i>	CATCCCTATA
<i>l<sub>x</sub></i>	AGACAAAAAA
<i>l<sub>y</sub></i>	ATTTTTTTCC
<i>m</i>	TATGACTGCAA
<i>x</i>	AGACCGCTAAA
<i>y</i>	ATACTGCTTTA

TABLE I. Sequence specification of domains in the design. Sequences are indicated in 5'→3' direction.

our algorithm (a) minimizes the total Hamming distance between the bricks target secondary structures and their folding predictions from ViennaRNA [16], and (b) maximizes the binding energies of all desired pair interactions while minimizing binding energies of all undesired pair interactions. Table I lists the nucleotide sequences of all domains, found by the highest-scoring genotype of our algorithm.

### B. Experimental Manipulation of DNA

DNA oligomers were provided by Eurogentec (Belgium) on a 100  $\mu$ M synthesis scale, with a standard desalting procedure or a required denaturing polyacrylamide gel electrophoresis (PAGE) purification for oligomers longer than 50 nucleotides and/or any 3'/5' modification. Streptavidin coated gold nanoparticles of 5 and 10 nm diameter were supplied by Life Technologies (Alexa Fluor 488 streptavidin). Samples and stock solutions were stored at -20°C.

The DNA recorder was prepared by sequentially adding 200 nM of each brick with 240 minutes waiting time between additions. DNA samples were dissolved in a total volume of 20  $\mu$ L of nuclease free water and 50 mM potassium acetate, 20 mM tris-acetate, 10 mM magnesium acetate, pH 7.9 buffer at room temperature (25 °C) and incubated for ten minutes if not otherwise specified. The mixture was shaken at 300 revolutions per minute in an Eppendorf Thermomixer Comfort set at 25°C.

Capillary electrophoresis has been performed using the Agilent Technology 2100 Bioanalyzer system with its DNA High Sensitivity Chip and adhered to manufacturer protocols.

Transmission electron microscopy (TEM) was performed with a Philips CM 100 Compustage (FEI) microscope and digital images were collected using an AMT

CCD camera (Deben). A volume of 5  $\mu$ L sample was applied on glow discharge grids preliminary washed with 0.5 mM magnesium chloride to change the hydrophilic surface charge orientation.

## IV. RESULTS

### A. Single Brick Calibration

We performed capillary electrophoresis measurements of all individual bricks in order to determine the response of the Agilent 2100 Bioanalyzer High Sensitivity DNA Assay for our non-standard DNAs. All bricks were provided in 200 nM concentration. Electropherograms always detected a single clear peak per brick. Table II summarizes for each brick its known size, the measured migration time and fluorescence area under the peak, as well as the calculated size and molarity derived by the instrument software from comparison to the reference ladder. Averages and standard deviations have been calculated from at least three independent measurements.

The measurements successfully discriminate the migration times of almost all strands (disregarding *report* strands) with significant differences. Only *start* and *read* cannot be reliably differentiated.

Striking discrepancies between the known brick sizes and the sizes derived by the software from comparison to the ladder might be attributed to two reasons: firstly, short oligomers such as *start*, *read* and *report* are well below the detection limit of the high sensitivity kit, which can resolve dsDNA fragments between 50 – 7000 base pairs in length. Secondly, the reported deviations might lie in the fact that our bricks contain extensive secondary structures that might affect their motility in the gel matrix.

A similar discrepancy is observed in the derived molarity values. This is partly due to the fact that molarity calculation is based on the base pair estimation and will thus suffer from the issues described before, partly because our bricks contain extensive ssDNA regions which interact differently with the fluorescent dye than dsDNA.

### B. Recording experiments

To probe the performance of the data recording (push) cycle, we performed experiments in which we sequentially recorded five signals (*X*, *X*, *X*, *Y*, *X*) onto the growing stack. We ran five parallel experiments and stopped them at different steps in the protocol. Gel-like images of the Bioanalyzer output are shown in Fig. 2.

For the first three recorded signals, addition of each *write-X* brick is accompanied by the appearance of a new clear peak in the spectrum: after addition of the first *write-X* brick this peak (*start-push-write-X* complex, or

brick	size[nt]	measured		derived	
		time [s]	area [FU]	size [bp]	molarity [nM]
<i>start</i> (S)	27	45.22±0.92	94.6±61.23	51±7.6	34.80±15.92
<i>push</i> (P)	64	46.81±0.76	74.4±39.2	64±6.9	8.08±0.174
<i>write-X</i> (X)	98	53.27±0.34	55.93±39.65	128±3.78	5.961±0.473
<i>write-Y</i> (Y)	128	55.35±0.06	5.27±1.15	147±0.8	0.845±0.221
<i>report-X</i> (Rx)	22	44.81±0.81	248.5±60.57	47±6.4	78.25±16.81
<i>report-Y</i> (Ry)	22	45.18±1.02	241.3±84.49	47±11.3	86.44±12.77
<i>read</i> (R)	31	44.61±0.35	73.85±15.76	46±2.82	31.67±1.21
<i>pop</i> (Q)	64	47.89±0.28	28.13±25.4	74±3.4	6.602±6.78

TABLE II. Calibration results (given as averages and standard deviation) for all individual strands provided in 200 nM concentrations.

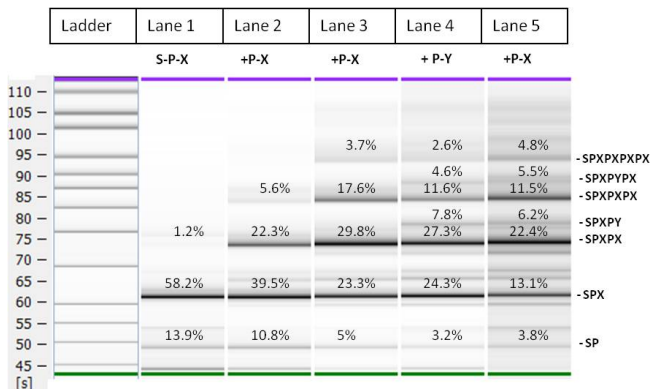


FIG. 2. Capillary electrophoresis of the recording process. Lane 1=SPX; Lane2=SPXPX; Lane 3=SPXPXPX; Lane 4=SPXPXPXPY; Lane 5=SPXPXPXPYPX. Data obtained from five parallel experiments.

SPX) accounts for more than 58% of the total fluorescence. Lane 2 shows the appearance of a second peak (SPXPX) that corresponds to the two signals. However, this second peak accounts for only about 22% of the total fluorescence, whereas almost 40% still correspond to the first signal (SPX). The situation repeats in the third lane, where the correct complex (SPXPXPX) accounts for slightly more than 17% of the fluorescence, the second signal peak (SPXPX) for about 30% and the first peak still for about 23%.

The addition of *write-Y* in lane 4 leads to the appearance of several new peaks, which we identify as SPY, SPXPY, and SPXPXPY. A very faint peak at about 98 s migration time might correspond to the desired SPXPXPXPY, but the signal is too weak to be properly identified by the analysis software. Lane 5 essentially shows the same peaks as lane 4, with peak sizes changing as expected: peaks from complexes ending in a *write-Y* brick become smaller, whereas the corresponding complexes with added *write-X* become proportionally larger.

In all lanes faint higher peaks indicate that there is a very small potential for run-away processes to create complexes with more signals than the provided ones. Yet, in all cases, the fluorescence of all these longer bands

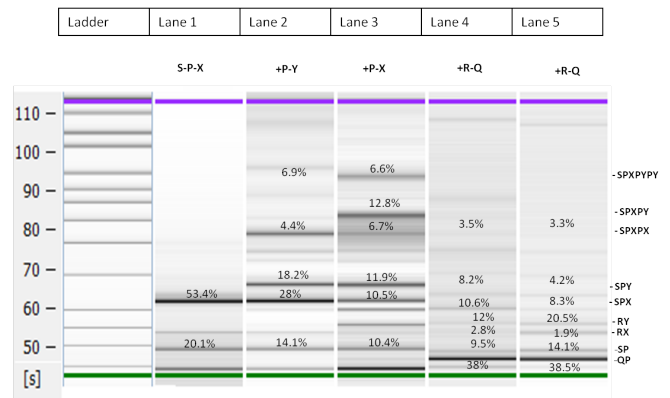


FIG. 3. Capillary electrophoresis of the recording and reading of three signals. Recording: Lane 1=SPX; Lane2=SPXPY; Lane 3=SPXPYPY. Reading: Lane 4=SPXPYPY+RQ; Lane 5=SPXPYPY+RQRQ.

combined does not exceed 10% of the total.

### C. Read Out Experiments

Next, we performed experiments to test the read-out (*pop*) cycle of the DNA stack. In this experiment, three signals (*X, Y, X*) were pushed onto the stack and subsequently removed by adding *read* (*R*) and *pop* (*Q*) bricks in molarities equal to the *start*, *push* and *write* bricks. Fig. 3 shows the gel-like images of the experiment.

Lanes 1 through 3 reconfirm the working of the recording cycle with the same observations than for the experiment of the last section: each added *write* brick generates a new peak in the spectrum with very little evidence for run-away processes and persistence of peaks that indicate intermediate complexes.

Lane 4 shows the response of the device after provision of 200 nM *read* and *pop*, which is supposed to trigger one readout cycle: newly created *push-pop* as well as *read-write* complexes result in the appearance of three new peaks at around 47.42 (QP), 52.22 (RX), and 57.39 (RY) seconds. The *push-pop* complexes account for 38% of the fluorescence, whereas *start-write-X* and *start-write-*

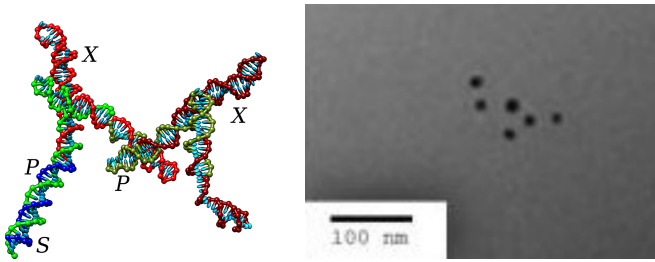


FIG. 4. **Left:** oxDNA simulation of a SPXPX complex. **Right:** Representative TEM image of a SPXPYPXPXPX complex.

$Y$  account for 2.8 and 12% respectively. Peaks associated with the different stack states SPXPYPY, SPYPY, SPXPY, and SPY decrease accordingly. The situation repeats in Lane 5 where the second readout cycle further increases *push-pop* and *read-write* peaks and simultaneously reduces intensities of the corresponding stack complexes. Noteworthy, after reading out the two recorded signals, 14.1% of the fluorescence results from the *start-push* complex whereas peaks of stacks that still contain recorded information only register with 8, 4.2, 4.8 and 3.3%.

#### D. Imaging

For additional confirmation of the recording, we imaged the assembled nanodevice using TEM. For this purpose, assembled stacks were mixed with *report* strands that, in turn, are decorated with 5 and 10 nm gold nanoparticles. *Report* bricks associate with their respective *write* bricks at any position in the assembled stack. Nanoparticles appear in TEM images as black dots that can be easily distinguished and classified.

Simple geometric considerations estimate an assembled structure where data hairpins are separated by about 15 nm with  $247^\circ$  twist. OxDNA simulations [17] (Fig. 4 left panel) indicate that the assembled stack does not necessarily extend straight forward but might instead contain a kink at each *signal-push* holiday junction. Fig. 4 (right panel) shows TEM results from an experiment where five signals ( $X, Y, X, X, X$ ) have been recorded. The image shows a stack with just one extra *write-X* on the left side of the recorder, resulting in a stack with six signals ( $X, X, Y, X, X, X$ ). The image shows a separation of 15-20 nm between the nanoparticles with a zig-zag configuration predicted by the simulations.

## V. DISCUSSION AND FUTURE WORK

We have presented a design and experimental evidence for the working of an in-vitro signal recorder based on DNA strand assembly and displacement. The recorder implements a stack data structure with push and pop operations and allows for storing two signal types.

Because we employ non-standard DNA strands, the electrophoresis analysis software does not correctly detect molecular concentrations, which prevents us to gain a precise quantitative picture of the involved processes. Nonetheless, capillary electrophoresis and TEM imaging indicate that the nanodevice is able to store at least three consecutive signals and does not suffer from problematic runaway processes.

However, after recording several signals, electrophoresis analysis indicates that the device is not only present in the desired final state, but also in several intermediate recording states. Because of the limits of experimental quantification, we can currently not offer a satisfying explanation for these intermediate peaks. This currently impacts the readout cycle, as the pop operation interacts with all present stacks and thus returns a superposition of recorded signals. While this is contrary to the intended working, we point out that such a superposition might also have advantages, as it might allow one to reverse engineer the composition *and order* of recorded information from a single electrophoresis read out.

We plan to improve experimental analysis methods using different capillary electrophoresis analysis kits (such as RNA assay kits) or molecular beacon experiments. Better experimental quantification will allow us to calibrate computational models that will in turn help us increase our understanding of the fidelity of the device.

Tantalizingly, as our design is based on ssDNA bricks, our entire data structure could – in principle – be expressed *in vivo* by a living cell as an RNA data structure and post-transcriptionally controlled. As we store data in a double-stranded fashion rather than in dangling single strands, an *in vivo* realization is likely to suffer less from enzymatic attack. Alternatively, the device could be used to programmatically and reversibly arrange matter such as liposomes [18, 19] on the nanoscale. We are currently exploring routes to implement this.

## ACKNOWLEDGMENTS

This work has been supported by EPSRC grant agreements n<sup>o</sup> EP/J004111/1, EP/J004111/2, EP/L001489/1, EP/L001489/2. We thank Chien-yi Chang, Christoph Flamm, Alessandro Ceccarelli, Omer Markovitch, and Ben Shirt-Ediss for helpful discussions.

[1] Seeman, N.C.: DNA in a material world. Nature **421**(6921) (2003) 427–431

[2] Soloveichik, D., Seelig, G., Winfree, E.: DNA as a universal substrate for chemical kinetics. Proc. Nat. Acad.



- Sci. USA **107**(12) (2010) 5393–5398
- [3] Stojanović, M.N., Stefanović, D.: Deoxyribozyme-Based Half-Adder. *J. Am. Chem. Soc.* **125**(22) (2003) 6673–6676
- [4] Seelig, G., Soloveichik, D., Zhang, D.Y., Winfree, E.: Enzyme-Free Nucleic Acid Logic Circuits. *Science* **314**(5805) (2006) 1585–1588
- [5] Cardelli, L.: Strand algebras for DNA computing. *Nat. Comput.* **10** (2011) 407–428
- [6] Chen, Y., Dalchau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D., Seelig, G.: Programmable chemical controllers made from DNA. *Nat. Nano.* **8**(10) (2013) 755–762
- [7] Qian, L., Winfree, E.: A simple DNA gate motif for synthesizing large-scale circuits. *J. R. Soc. Interface* **8**(62) (2011) 1281–97
- [8] Qian, L., Winfree, E.: Scaling up digital circuit computation with DNA strand displacement cascades. *Science* **332**(6034) (2011) 1196–201
- [9] Li, W., Zhang, F., Yan, H., Liu, Y.: DNA based arithmetic function: a half adder based on DNA strand displacement. *Nanoscale* **8**(6) (2016) 3775–3784
- [10] Liu, H., Wang, J., Song, S., Fan, C., Gothelf, K.V.: A DNA-based system for selecting and displaying the combined result of two input variables. *Nature Comm.* **6** (2015) 10089
- [11] MacDonald, J., Li, Y., Sutovic, M., Lederman, H., Pendri, K., Lu, W., Andrews, B.L., Stefanovic, D., Stojanovic, M.N.: Medium Scale Integration of Molecular Logic Gates in an Automaton. *Nano Lett.* **6**(11) (2006) 2598–2603
- [12] Qian, L., Soloveichik, D., Winfree, E.: Efficient Turing-Universal Computation with DNA Polymers. In Sakakibara, Y., Mi, Y., eds.: *DNA Computing and Molecular Programming*. Number 6518 in *Lect. Notes Comput. Sci.* Springer Berlin Heidelberg (2011) 123–140
- [13] Terrazas, G., Gheorghe, M., Kendall, G., Krasnogor, N.: Evolving tiles for automated self-assembly design. In: *IEEE Congress on Evolutionary Computation, 2007. CEC 2007.* (2007) 2001–2008
- [14] Siepmann, P., Martin, C.P., Vancea, I., Moriarty, P.J., Krasnogor, N.: A Genetic Algorithm Approach to Probing the Evolution of Self-Organized Nanostructured Systems. *Nano Letters* **7**(7) (2007) 1985–1990
- [15] Woolley, R.A.J., Stirling, J., Radocea, A., Krasnogor, N., Moriarty, P.: Automated probe microscopy via evolutionary optimization at the atomic scale. *Applied Physics Letters* **98**(25) (2011) 253104
- [16] Lorenz, R., Bernhart, S.H., Höner zu Siederdisen, C., Tafer, H., Flamm, C., Stadler, P.F., Hofacker, I.L.: ViennaRNA Package 2.0. *Algorithms Mol. Biol.* **6**(1) (2011) 26
- [17] Doye, J.P.K., Ouldridge, T.E., Louis, A.A., Romano, F., Šulc, P., Matek, C., Snodin, B.E.K., Rovigatti, L., Schreck, J.S., Harrison, R.M., Smith, W.P.J.: Coarse-graining DNA for simulations of DNA nanotechnology. *Phys. Chem. Chem. Phys.* **15**(47) (2013) 20395
- [18] Hadorn, M., Bönzli, E., Fellermann, H., Eggenberger Hotz, P., Hanczyc, M.: Specific and reversible DNA-directed self-assembly of emulsion droplets. *Proc. Nat. Acad. Sci. USA* **109**(47) (2012)
- [19] Fellermann, H., Cardelli, L.: Programmable chemistry in DNA addressable bioreactors. *R. Soc. Interface* **11**(99) (2014) 20130987