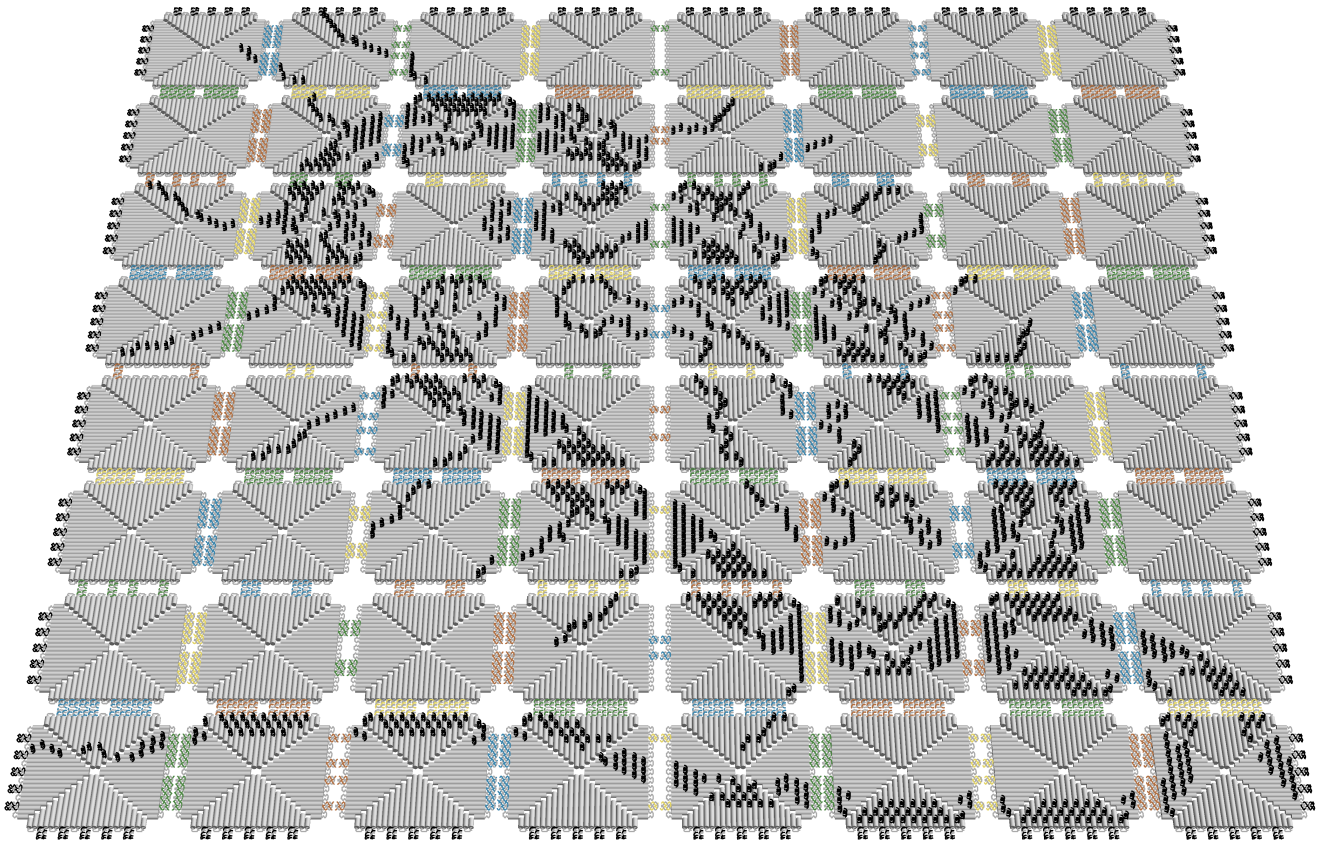
SUPPLEMENTARY INFORMATION

doi:10.1038/nature24655

Fractal assembly of micrometre-scale DNA origami arrays with arbitrary patterns

Supplementary information

Grigory Tikhomirov[1⋆], Philip Petersen[2⋆] and Lulu Qian[1,3]



Contents

1 Materials and methods

1.1 Sample preparation

Single-stranded M13mp18 DNA (scaffold strand) was purchased from Bayou Biolabs (catalog # P-107) at 1 g/L in 1× TE buffer (10 mM Tris-HCl, 1 mM EDTA, pH 8.0). The concentration of the scaffold strand was calculated on the basis of DNA ultraviolet absorbance measurements at 260 nm using NanoDrop2000 (Thermo Scientific). Staple strands (sequences listed in Supplementary Tables 1 to 5) were purchased unpurified from Integrated DNA Technologies in 1× TE buffer (pH 8.0) at 100 μM each. Negation strands (sequences listed in Supplementary Table 6) that are complementary to the edge staples were purchased at 200 μM each. The strands were diluted to 15 μM in 1× TE buffer and loaded into Echo qualified 384-well source microplate (Labcyte).

Individual DNA origami tiles were mixed from the source plate by Echo 525 liquid handler (Labcyte) to yield 10 μL total volume with 10 nM scaffold strand and 75 nM staples in 1× TE buffer with 12.5 mM Mg^{2+}, after the addition of the negation strands. (Before the negation strands were added, during annealing of the individual DNA origami tiles, the volume was slightly lower than 10 μL and the concentrations of the scaffold and staples were slightly higher than 10 nM and 75 nM, respectively.) The scaffold and staples were kept at 90 °C for 2 min and annealed from 90 °C to 20 °C at 6 sec per 0.1 °C in a twin.tec 96-well skirted PCR plate (Eppendorf, catalog # 951020401) sealed with domed cap strips (Eppendorf, catalog # 0030124839) on a Nexus Mastercycler (Eppendorf). After the anneal, a five-fold excess (relative to the concentration of the staple strands) of a full set of 44 negation strands were added to each type of DNA origami tile and quickly cooled down from 50 °C to 20 °C at 2 sec per 0.1 °C.

2 by 2 arrays were prepared by mixing equal volumes of four individual tiles and annealing from 55 °C to 45 °C at 2 min per 0.1 °C and from 45 °C to 20 °C at 6 sec per 0.1 °C. The total annealing time is roughly 3.5 hours.

4 by 4 arrays were prepared by mixing equal volumes of four 2 by 2 arrays obtained in the previous step and annealing from 45 °C to 35 °C at 8 min per 0.1 °C and from 35 °C to 20 °C at 6 sec per 0.1 °C. The total annealing time is roughly 13.5 hours.

8 by 8 arrays were prepared by mixing equal volumes of four 4 by 4 arrays obtained in the previous step and annealing from 35 °C to 25 °C at 32 min per 0.1 °C and from 25 °C to 20 °C at 6 sec per 0.1 °C. The total annealing time is roughly 53.5 hours.

For arrays with patterns, a 10-fold excess of poly-A strand was added to the arrays before AFM imaging, allowing at least ten minutes for the poly-A strand to hybridize to the poly-T staple extensions at room temperature.

Important note: it is absolutely essential that all tiles or arrays are mixed at equal concentration in all stages, otherwise the yield will decrease significantly. To achieve the best yield, a tight seal of the plate during annealing is necessary. We explored several sealing options including films, foils and caps, and the cap strips specified above produced the most reliable results. However, even with a tight seal, it is still possible that some wells in the plate will evaporate more than other wells. Taking this evaporation into consideration, it is also necessary to transfer all solution from the wells for mixing the tiles or arrays.

1.2 Echo protocol

The transfer volume in a protocol for an Echo 525 liquid handler must be multiples of 25 nL. Additionally, the volume of sample in each well of the Echo qualified 384-well source plate must be 15 to 65 μL, resulting in 15 μL of unusable sample. Because of both constraints, we diluted the edge staples to 15 μM before storing them in a source plate. This resulted in a transfer volume of 50 nL for each edge staple to have the 75 nM target concentration in a 10 μL total volume. If all staples were at 15 μM, the total volume would have exceeded 10 μL. Thus we diluted the interior staples (including those with extensions) to 30 μM, resulting in a transfer volume of 25 nL for each interior staple.

Because the bridge staples are the same in all tiles, we mixed them together and divided the mixture into five wells in a source plate. The concentration of the bridge staple mixture was at $100/38 = 2.63$ μM, for 38 distinct bridge staples. Based on the target concentration and volume, the desired transfer volume of the bridge staple mixture should be 57 nL per well. Rounding it to the multiples of 25 nL yielded the actual transfer volume of 50 nL. The smaller volume is ok because the staples are in large excess relative to the M13 scaffold. In the FracTile Compiler, we wanted to make it convenient for the users to organize their strands, by fitting all strands and buffer in a single 384-well source plate. To do that, we divided the bridge staple mixture into two wells instead of five, resulting in a transfer volume of 125 nL per well.

The concentration of the M13 scaffold varied from batch to batch, but typically the difference is no more than 10%. We used 0.343 μM of M13 divided into twelve wells in the source plate. The desired transfer volume is 24.3 nL per well, for 10 nM target concentration in 10 μL. We rounded it to 25 nL per well, which is the nearest multiples of 25 nL. Again, for the purpose of fitting all strands in one 384-well source plate, we used only three wells in the FracTile Compiler, resulting in a transfer volume of 100 nL per well.

We used eight wells of 1×TE/10×Mg^{2+}, resulting in a transfer volume of 125 nL per well for the target volume of 10 μL per tile. We used sixteen wells of 1×TE, generally resulting in a transfer volume of 175 to 200 nL per well, distributed as evenly as possible. The difference in the volume of 1×TE is due to the fact that the total number of edge staples varies in different tiles. In the FracTile Compiler, we kept the same number of wells for 1×TE/10×Mg^{2+}, but reduced it to fifteen wells for 1×TE, also for the purpose of fitting all strands in one source plate.

A full set of 44 negation strands were mixed together at 200 μM each, resulting in $200/44 = 4.545$ μM of the mixture. As the target concentration is $75 \times 5 = 375$ nM in 10 μL, the desired transfer volume is 825 nL. The negation strand mixture was added to each tile either by Echo or by manual pipetting after the tiles were individually annealed. We had sixteen wells of the negation strand mixture in a source plate, each was used to transfer 825 nL four times into a 96-well destination plate, for adding the mixture to a total of 64 tiles in an 8 by 8 array. In the FracTile Compiler, we left out the negation strand mixture from the source plate to make room for other strands.

The manual protocol generated by the FracTile Compiler uses the same concentrations and transfer volumes as in the Echo protocol.

1.3 AFM imaging

Samples for AFM imaging were prepared by diluting the annealed samples ten fold, resulting in 1 nM of the scaffold strand (i.e. the sum of all monomers) in 1× TE/Mg^{2+} buffer. After dilution, 40 μL of the sample was deposited onto freshly cleaved mica (SPI Supplies, 9.5 mm diameter, catalog # 01873-CA). After 30 sec the solution was removed by sucking up all the liquid that comes off in a single thumb-up movement while keeping the pipette attached to and almost perpendicular to the mica surface. To minimize the background during imaging, the excess of staple and negation strands was removed as follows. The mica surface was washed three times with 40 μL TE buffer containing 10 mM $MgCl_2$ and 100 mM NaCl, by performing 10 down-and-up thumb movements for each wash. After that, 80 μL of 1× TE/Mg^{2+} buffer was added onto the mica and the sample was imaged. AFM images were taken using tapping mode in fluid on a Dimension FastScan Bio (Bruker) with FastScan-D tips (Bruker). Typical scanning parameters were: scan rate = 5 Hz, lines = 1024, amplitude set point = 30–50 mV, drive amplitude = 180–240 mV, drive frequency = 110 Hz, integral gain = 1, proportional gain = 2.

2 Challenges for uniquely-addressable DNA origami arrays

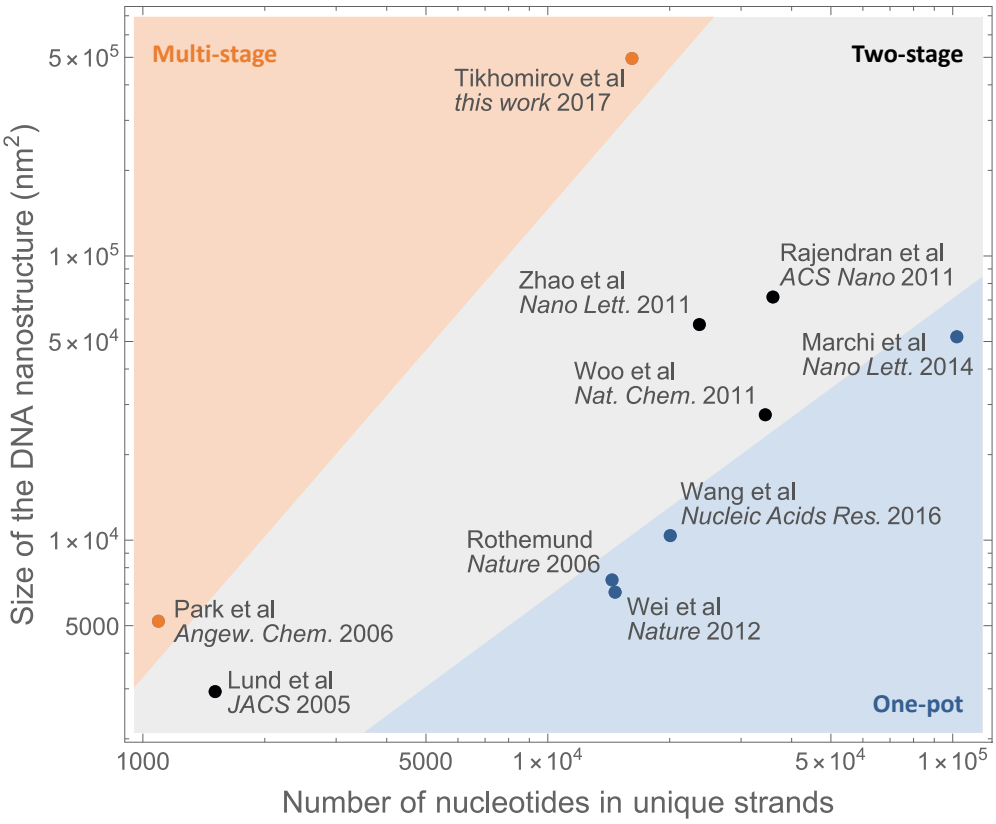


Figure 1: **Overview of uniquely-addressable two-dimensional DNA nanostructures.** Each point in the log – log plot corresponds to the size of a DNA nanostructure and the number of nucleotides in unique strands self-assembled in the nanostructure. Although Wang et al explored both one-pot and two-stage experiments, we show the work as one-pot because no interior or edge strands were reused. Similarly, Rajendran et al explored both two-stage and three-stage experiments, and we show the work as two-stage because no edge strands were reused.

There are three types of approaches for creating uniquely-addressable two-dimensional DNA nanostructures (Supplementary Fig. 1). Distinct DNA strands can be annealed together in one pot to create DNA origami[10,14] and arrays of single-stranded tiles[32] and double-crossover tiles.[33] The one-pot approaches are easy to implement, but the number of nucleotides in unique strands increases linearly with the size of the structure. The increasing number of nucleotides makes it difficult to scale up, due to the cost of the strands, the design challenges for controlling the spurious interactions among distinct strands, and the resulting decrease in yield. Using hierarchical approaches, DNA molecules can be annealed in two stages, first self-assembling into smaller structures such as cross-shaped DNA tiles[34] or DNA origami tiles,[11–13] and then the individual tiles coming together to form larger structures. In the two-stage approaches, the interior strands can be reused for different tiles, but an increasing number of unique edge strands is still required. Multi-stage hierarchical approaches make it possible to reuse both interior and edge strands, as shown in a 4 by 4 array of small DNA tiles.[17] Because of the potential for creating arbitrarily complex structures from a simple set of tiles, the strategies for multi-stage self-assembly have been explored extensively in theory.[16,35–37] However, none of these theoretical strategies has yielded a successful experimental demonstration.

In prior work, we developed a framework for creating random tilings with both unbounded and finite DNA origami arrays.[18] The finite arrays self-assembled in two stages had a four-fold rotational symmetry, which only required $n^2/4$ distinct types of tiles and $n(n-1)/2$ distinct types of edges for arrays of size n by n. For example, 4 by 4 arrays were constructed with 4 distinct tiles and 6 distinct pairs of edges (Supplementary Fig. 2, top row). The small number of distinct tiles and edges allowed us to successfully construct arrays of sizes 3 by 3 to 5 by 5. However, the rotational symmetry of these arrays does not allow unique addressability.

To create uniquely-addressable DNA origami arrays, we had to use four times the number of distinct types of tiles and edges compared to the prior work. For example, 4 by 4 arrays now require 16 distinct tiles and 24 distinct pairs of edges (Supplementary Fig. 2, middle and bottom rows). In prior work, we used two types of edge codes with a stronger and a weaker binding energy: the stronger one was composed of 11 edge staples, each has a stacking bond and a one-nucleotide sticky end; the weaker one was composed of 4 edge staples, each has a stacking bond and a two-nucleotide sticky end (Supplementary Fig. S54 of ref.[18]). The stronger edge code was used for interactions between tiles composing each of the four 2 by 2 arrays and the weaker edge code was used for interactions between the 2 by 2 arrays. Taking advantage of the M13 sequences being different on the four sides of the square origami tile, each edge code can provide a maximum of four pairs of distinct edges. Therefore, we cannot use the two types of edge codes in the arrays with rotational symmetry to create 24 distinct pairs of edges for the uniquely-addressable arrays (16 for interactions between tiles composing each of the four 2 by 2 arrays and 8 for interactions between the 2 by 2 arrays).

Keeping the edge codes palindromic as in prior work, there are a total of $\binom{4}{2} = 6$ types of edge codes, using 4 out of the 8 edge staples (positions 2 through 5 and 7 through 10) that each has a stacking bond and a two-nucleotide sticky end. Using all six edge codes, we now have $6 \times 4 = 24$ distinct edges to create the uniquely-addressable 4 by 4 arrays. However, the yield of these arrays was substantially lower than the 4 by 4 arrays with rotational symmetry (Supplementary Fig. 2, middle row), presumably due to the increased spurious interactions among the increased number of distinct tiles. More importantly, the 24 distinct pairs of edges used in the 4 by 4 arrays already reached our limit for designing orthogonal edges, and thus it is not possible to create larger uniquely-addressable arrays using the same method.

Dividing the self-assembly process into more stages could provide a natural solution for (i) reducing spurious interactions (by reducing the total number of possible reactions at any given time during self-assembly), and (ii) using fewer distinct edges (by reusing the same edge interactions for tiles that are in different test tubes at the same stage). However, multi-stage self-assembly cannot work unless the edge design is compatible with a multi-stage annealing protocol: there must exist an annealing temperature for each stage that is both low enough to keep the structures from a previous stage stable and high enough to melt the spurious interactions at the current stage. When we attempted to create the 4 by 4 arrays in three stages, we first annealed the sixteen individual tiles from 90 to 20 °C, then annealed the four 2 by 2 arrays in separate test tubes from 50 to 20 °C, and finally annealed the four 2 by 2 arrays together from 30 to 20 °C. Because the melting temperature of the 2 by 2 arrays is about 35 °C (Supplementary Fig. S23 of ref.[18]), we had to keep the annealing temperature of the third stage below that, which was not high enough to melt the spurious interactions and resulted in an extremely poor yield (Supplementary Fig. 2, bottom row).

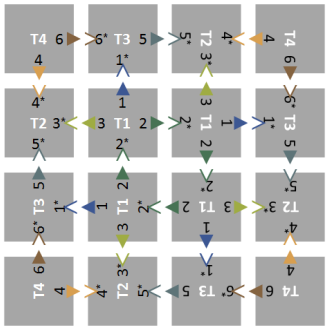
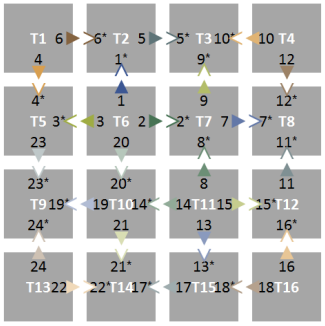
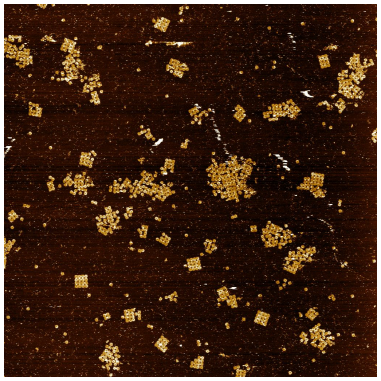
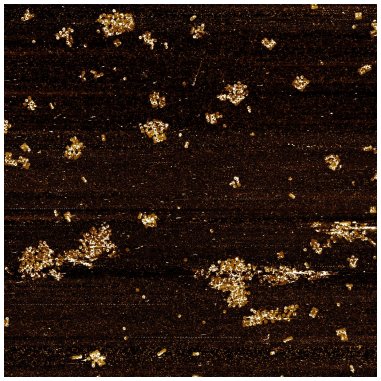
	Design diagram	Annealing protocol	AFM image	Yield
4 by 4 array with a four-fold rotational symmetry Tikomirov et. al. *Nat. Nanotechnol.* 2016		two-stage		15%
4 by 4 array with unique addressability		two-stage		4.7%
		three-stage		0%

Figure 2: **Prior work and failed attempts for creating uniquely-addressable DNA origami arrays.** All AFM images are 10 by 10 μm. The yield of each sample was estimated using AFM images of 30 by 30 μm.

3 Key definitions and abstractions

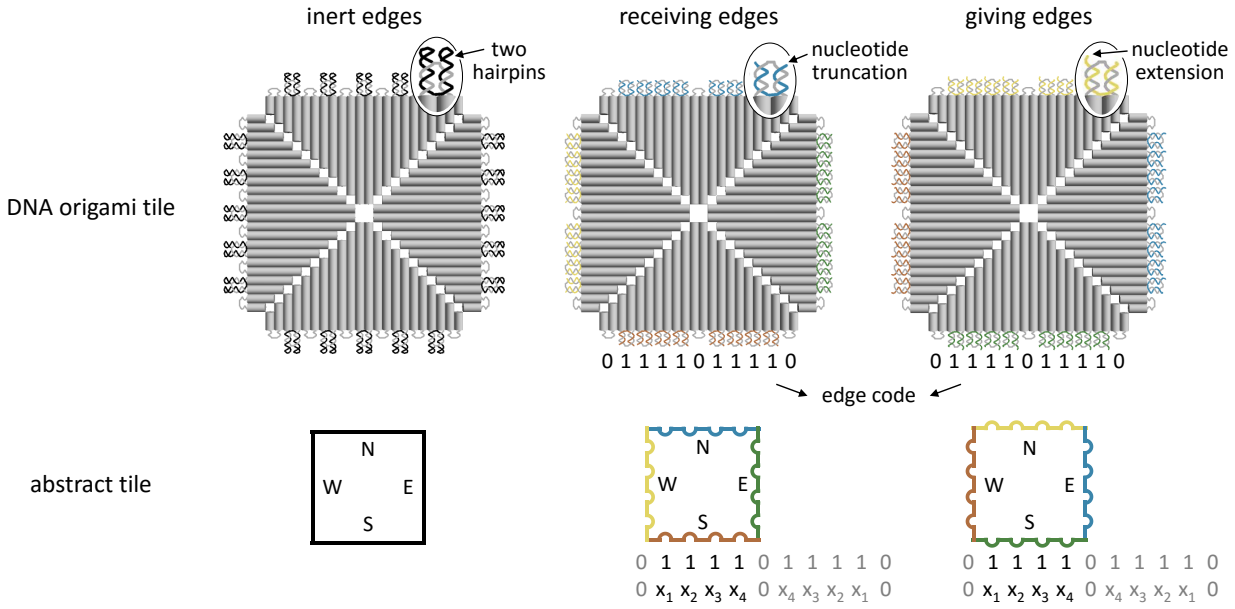


Figure 3: **Key definitions and abstractions.** A square DNA origami tile has four edges: north (N), east (E), south (S) and west (W). Each edge has a maximum of eleven staples. Because the four triangles composing the square tile are folded from different parts of the M13 scaffold, the staples on the four edges naturally have different sequences. "Inert edges" are created using five edge staples that are each capped with two hairpins to inhibit their interactions with other edge staples. Inert edge staples are colored black. Six edge staples are left out and the remaining scaffold loops are colored gray. There are two types of "active edges": "receiving edges" are created using eight or less edge staples that each has a two-nucleotide truncation on the 3' end; "giving edges" are created using eight or less edge staples that each has a two-nucleotide extension on the 5' end. An "edge code" is associated with each receiving or giving edge: the code consists of eleven 0s and 1s. Each 0 corresponds to a scaffold loop and each 1 corresponds to a staple. Receiving staples on the north, east, south and west edges are colored blue, green, orange and yellow, respectively. Giving staples are colored based on the sequence identity of the extension: extensions that are complementary to the truncations on the north, east, south and west edges are colored blue, green, orange and yellow, respectively. Although a giving edge can be complementary to any receiving edge, we only use north giving to west, east giving to north, south giving to east, and west giving to south in the design of fractal assembly. Abstract tiles are used to simplify the illustration. The edge colors in an abstract tile correspond to the staple colors in an origami tile. Each indentation corresponds to a 1 in an edge code for a receiving edge. Each bump corresponds to a 1 in an edge code for a giving edge. In fractal assembly, we use palindromic edge codes that have three 0s at fixed locations. Therefore, only the second to fifth digits are needed to infer any edge code, and only the 1s in these four digits are shown as indentations or bumps in an abstract tile.

4 Melting temperature measurement

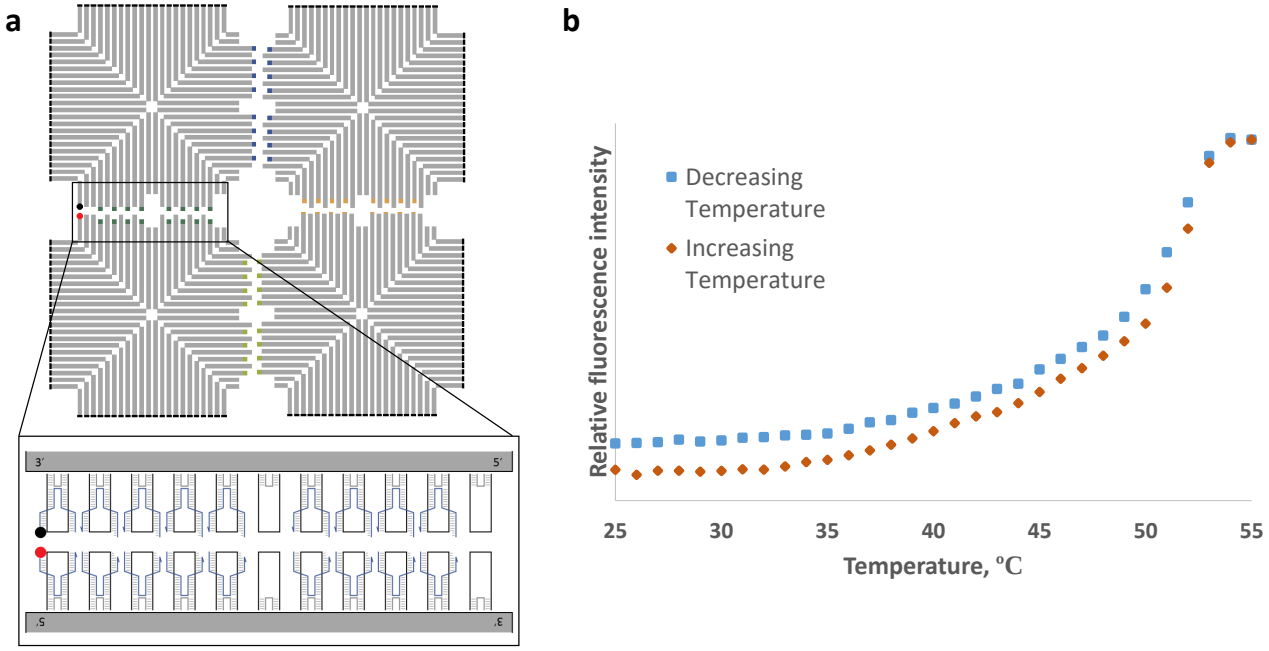


Figure 4: **Fluorescence experiments for melting temperature measurement. a,** Tile abstraction and edge design of a 2 by 2 array. A ROX fluorophore (shown as a red dot) is attached to the 5' end of an edge staple in one DNA origami tile, and a Iowa Black RQ quencher (shown as a black dot) is attached to the 3' end of an edge staple in another tile. When the tiles self-assemble into 2 by 2 arrays, the fluorophore and quencher will come into proximity and result in low fluorescence intensity. When the arrays melt, the fluorophore and quencher will become separated and result in increased fluorescence intensity. **b,** Melting graph showing relative fluorescence intensity during heating and cooling of the 2 by 2 arrays. The fluorescence intensity was measured in a Mx3005P QPCR system (Agilent Technologies). The sample was heated up from 25 to 45 °C at 5 sec/0.1°C, from 45 to 55 °C at 30 sec/0.1°C, held at 55 °C for 30 sec, cooled down from 55 to 45 °C at 30 sec/0.1°C, and then cooled down from 45 to 25 °C at 5 sec/0.1°C. Fluorescence intensity was measured with 585 nm excitation wavelength and 610 nm emission wavelength. Each data point shown in the graph was an average of all data points within the same degree. Note that the fluorophore and quencher labeled staples were added to the edge design shown in Fig. 2b. They introduced two additional stacking bonds to the interaction between tiles, with which we expect a slightly increased melting temperature.

5 Yield estimation

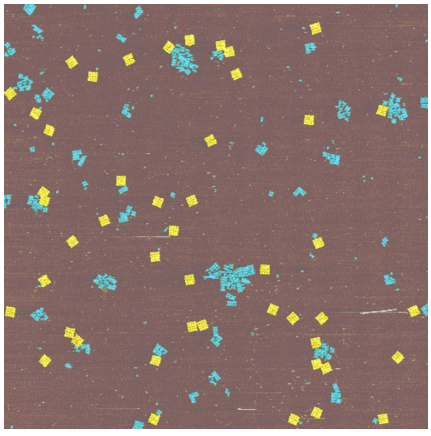
5.1 Assessing the accuracy of AFM-based yield estimation

How accurate AFM-based yield estimation is depends on if there exists a bias for the binding of structures to mica surface, specifically based on their sizes. To explore that, we mixed 4 by 4 arrays with monomers at an equal concentration of individual origami tiles (Supplementary Fig. 5a). If the monomers have the same extent of binding as the 4 by 4 arrays, we expect the estimated yield of 4 by 4 arrays will decrease to half compared to the estimated yield without monomers, while greater or smaller than half indicates less or more binding of the monomers, respectively. The AFM images showed a bias of more monomers than 4 by 4 arrays on the mica surface (Supplementary Fig. 5b and c). Our hypothesis is that monomers diffuse faster in solution and thus land faster on mica surfaces, and once landed, the binding is strong enough for them to stay on mica after washing the surface before imaging. This result suggests that the estimated yield could be lower (or higher) than the actual yield if the off-target structures are mostly incomplete structures (or aggregations). Since the off-target structures in the fractal arrays contain both incomplete and aggregated structures, the AFM-based yield estimation should be reasonably accurate, in the sense that it is not obviously lower or higher than the actual yield.

a

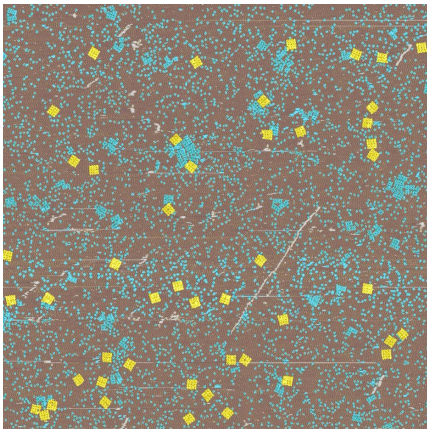
	Binding of monomers compared to that of 4 by 4 arrays to mica surface	Yield
4 by 4 array		x
4 by 4 array mixed with monomers at a 1: 1 ratio	less	$> x/2$
	same	$= x/2$
	more	$< x/2$

b 4 by 4 array



Yield: 44.70%

c 4 by 4 array + monomers



Yield: 13.88%

Figure 5: **Assessing the accuracy of AFM-based yield estimation. a,** Three cases of scenarios for estimating the yield of DNA origami arrays using AFM images. **b,** The yield of plain 4 by 4 arrays was estimated to be $x = 44.70\%$. **c,** The yield of plain 4 by 4 arrays mixed with monomers was estimated to be 13.88%, which is smaller than $x/2$.

5.2 Plain arrays

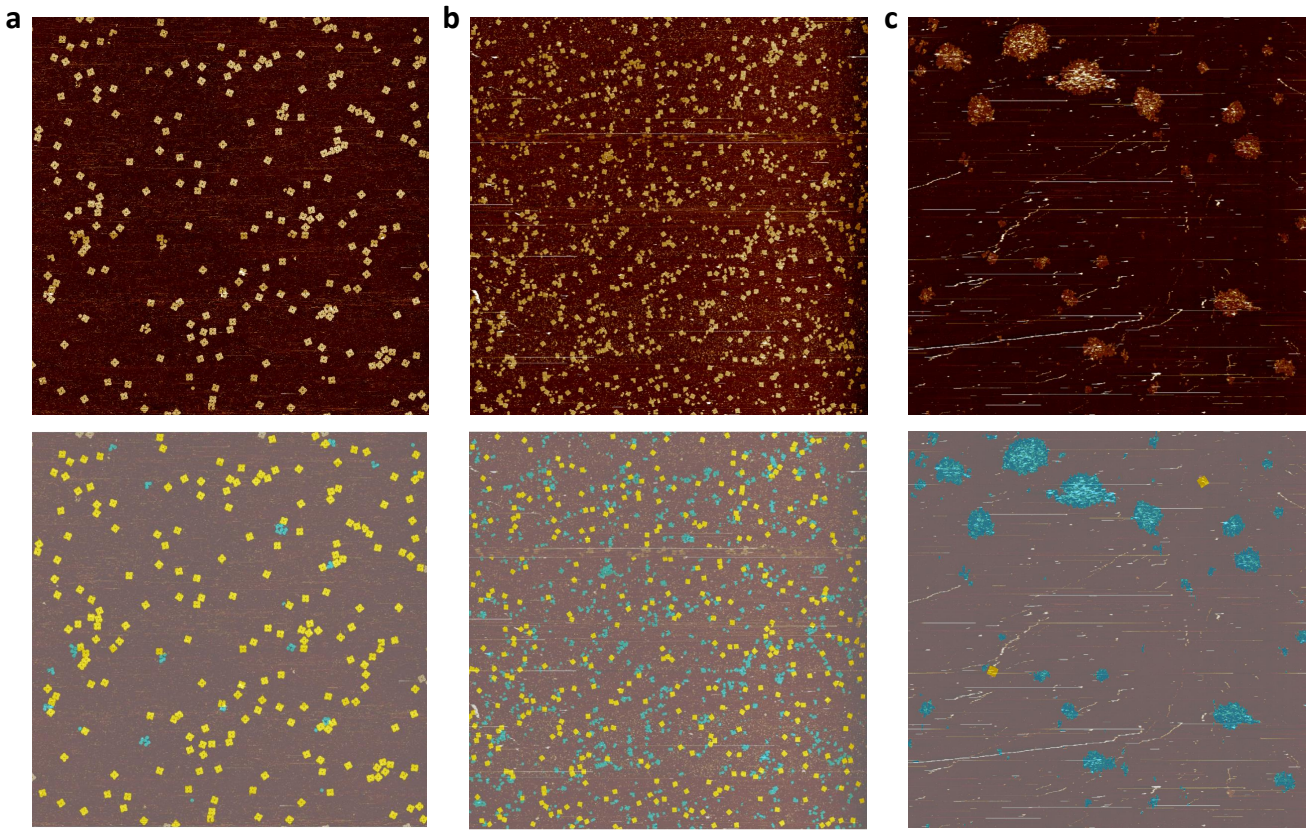


Figure 6: **AFM images of the plain arrays. a,** A 10 by 10 μm image of 2 by 2 arrays. The yield was estimated to be $92.81 \pm 1.74\%$. **b,** A 30 by 30 μm image of 4 by 4 arrays. The yield was estimated to be $47.91 \pm 1.76\%$. **c,** A 30 by 30 μm image of 8 by 8 arrays. The yield was estimated to be $1.81 \pm 1.27\%$. The yield was determined as the total pixels in complete arrays of the designed size (yellow pixels) divided by the total pixels above the threshold of background (blue pixels + yellow pixels). The standard for identifying complete arrays is explained in ref.[18] Supplementary Fig. S59. The calculation was aided by a custom software tool.[38] The error was calculated as $p\sqrt{1-p}/\sqrt{n}$, where p is the estimated yield and n is the number of complete arrays in each image, treating the yield as a Bernoulli probability.[39]

5.3 Arrays with an example pattern

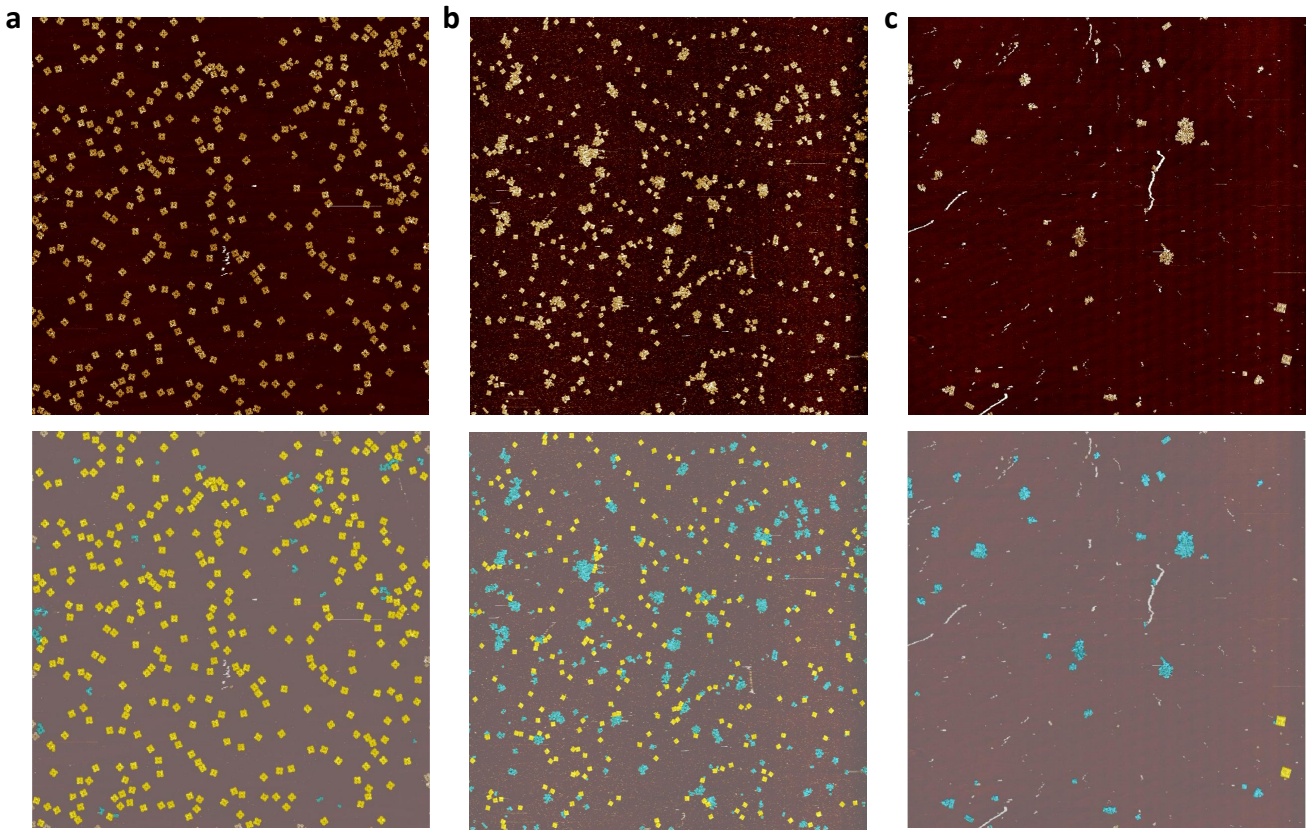


Figure 7: **AFM images of the arrays with an example pattern of Mona Lisa. a,** A 10 by 10 μm image of 2 by 2 arrays. The yield was estimated to be $94.22 \pm 1.22\%$. **b,** A 30 by 30 μm image of 4 by 4 arrays. The yield was estimated to be $41.55 \pm 1.26\%$, averaged over three images. **c,** A 30 by 30 μm image of 8 by 8 arrays. The yield was estimated to be $3.22 \pm 1.00\%$, averaged over three images. The yield was determined as the total pixels in complete arrays of the designed size (yellow pixels) divided by the total pixels above the threshold of background (blue pixels + yellow pixels). The standard for identifying complete arrays is explained in ref.[18] Supplementary Fig. S59. The calculation was aided by a custom software tool.[38] The mean was calculated as $p = \sum n_i / (\sum n_i / p_i)$, where p_i is the estimated yield and n_i is the number of complete arrays in each image. The error was calculated as $p\sqrt{1-p}/\sqrt{\sum n_i}$, treating the yield as a Bernoulli probability.[39]

6 Effect of design and experimental conditions

6.1 Rotation rule

In prior work, we have learned that surface modifications such as staple extensions can introduce extra internal curvature to individual tiles and disrupt the formation of two-dimensional arrays (Supplementary Figs. S14 and S15 in ref.[18]). We also showed that changing the orientation of tiles relative to their neighbors can prevent the curvature from propagating globally and restore the formation of two-dimensional arrays (Supplementary Fig. S19 in ref.[18]).

In this work, we specifically examined the effect of tile orientations in fractal assembly. When the tiles had the same orientation as their neighbors, we observed aggregations and no target structures (Supplementary Fig. 8a). When the tiles were rotated 90 degree compared to their neighbors, we observed a substantially increased population of incomplete but correctly-formed structures (Supplementary Fig. 8b). One example of these structures is shown in Supplementary Fig. 8c. We thus concluded that the rotation rule is crucial for balancing out the curvature of individual tiles in fractal assembly.

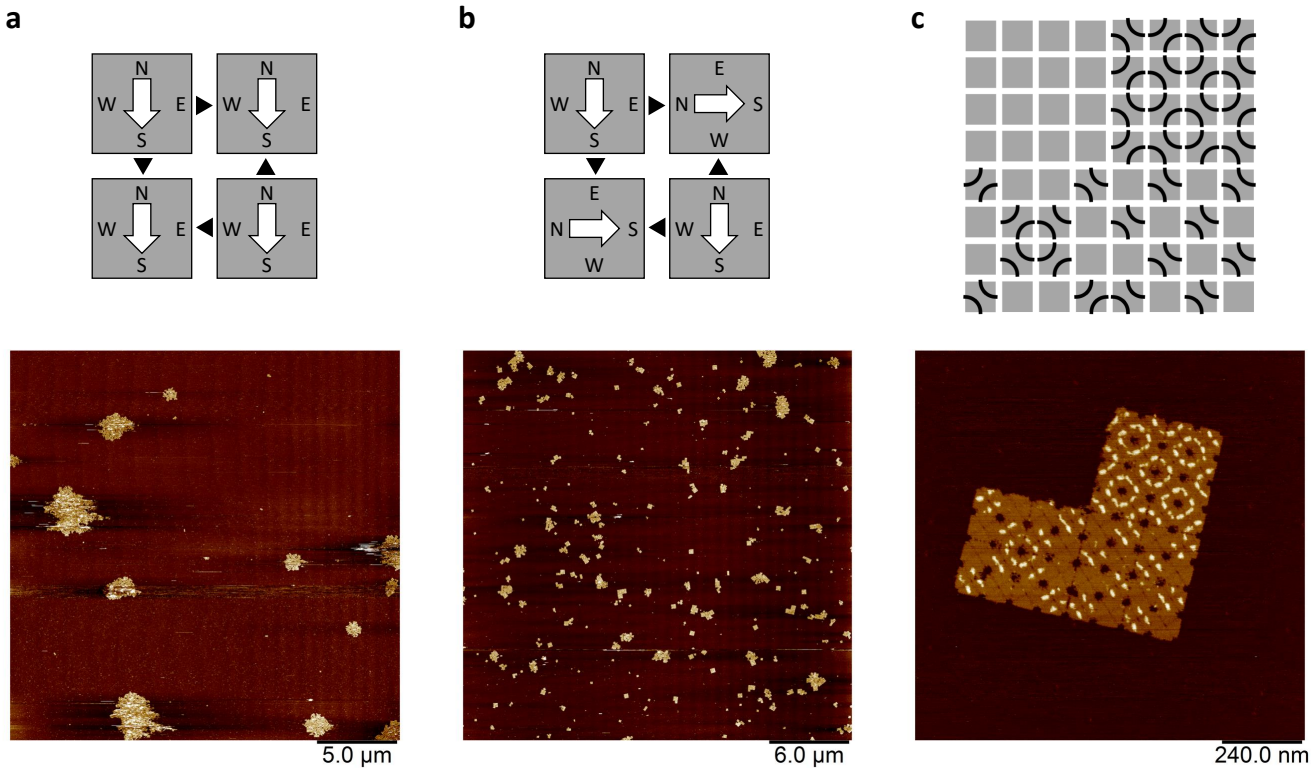


Figure 8: **Rotation rule. a,** Design diagram and AFM image of 8 by 8 fractal assembly with four tiles or arrays having the same orientation in each stage of self-assembly. **b,** Design diagram and AFM image of 8 by 8 fractal assembly with two tiles or arrays having the same orientation and the other two being rotated 90 degree in each stage of self-assembly. **c,** Design diagram and AFM image of an 8 by 8 fractal array with a specific pattern. The AFM image shows an 8 by 8 array with one missing quadrant. The image was obtained from the same sample shown in (b).

6.2 Annealing temperature

With the incomplete but correctly-formed structures, we decided to explore if an increased annealing temperature will further melt the spurious interactions, make the missing quadrant available, and promote the formation of complete structures. However, the increased annealing temperature of the final stage resulted in scrambled structures (Supplementary Fig. 9c), indicating that the temperature was too high and the 4 by 4 arrays melted into 2 by 2 arrays which then recombined into undesired structures. We thus concluded that the annealing temperatures should be kept at 55, 45 and 35 °C for the three sequential stages.

To reduce the gap of annealing temperatures and allow more sequential stages for creating larger arrays, one would have to re-design the edges of the tiles to increase the melting temperatures of the 4 by 4 and 8 by 8 arrays in fractal assembly (e.g. increase the number of edge staples from 4 and 2 to 6 and 4 for the second and third stages, respectively).

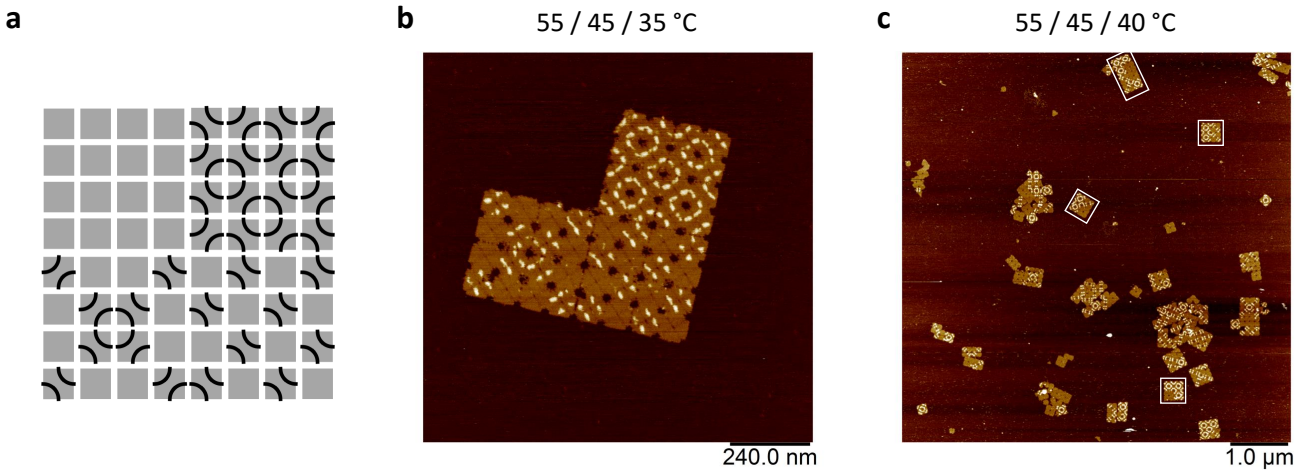


Figure 9: **Annealing temperature. a,** Design diagram of an 8 by 8 fractal array with a specific pattern. **b,** AFM image of an incomplete array with three correctly-formed quadrants. The sample was annealed from 55 to 20 °C, 45 to 20 °C and 35 to 20 °C in three sequential stages. **c,** AFM image of incomplete and scrambled arrays. The sample was annealed from 55 to 20 °C, 45 to 20 °C and 40 to 20 °C in three sequential stages. The white boxes highlight a few examples of structures that are obviously scrambled.

6.3 Non-interactive locations in the edge code

To look into the reasons for incomplete assemblies, we constructed the four quadrants of the 8 by 8 arrays separately. We discovered that three quadrants had fairly high yields (one example quadrant was shown in Supplementary Fig. 10b), but one quadrant had a low yield and some tiles near the corners and edges were deformed (Supplementary Fig. 10c). This quadrant originally had no surface modifications, so we added some patterns to verify the locations of the tiles — we found that despite being deformed, the tiles were incorporated into the desired locations for the few structures of the target size.

We hypothesized that the deformation of the tiles was caused by the interactions between the adjacent scaffold loops at the "0" locations in the edge codes. Since the edges for the interactions between the four quadrants have the largest number of "0"s, the deformation is more likely to occur at those locations. Because the interactions depend on the sequences of the scaffold loops, which are different along the edges of the four distinct quadrants, they affected one quadrant much worse than others.

To reduce the interactions between adjacent scaffold loops, we tested another implementation: instead of leaving out a staple for each "0", we used a staple with four nucleotides truncated from both 5' and 3' ends. We hoped that given the full-length staples at the "1" locations, the truncations at the "0" locations would be sufficient to reduce stacking interactions, but we were wrong. Four copies of the same quadrant by itself self-assembled into an 8 by 8 array, as a result of the stacking interactions between all "0"s (Supplementary Fig. 10d).

We thus concluded that we need a different method to reduce the deformation of tiles.

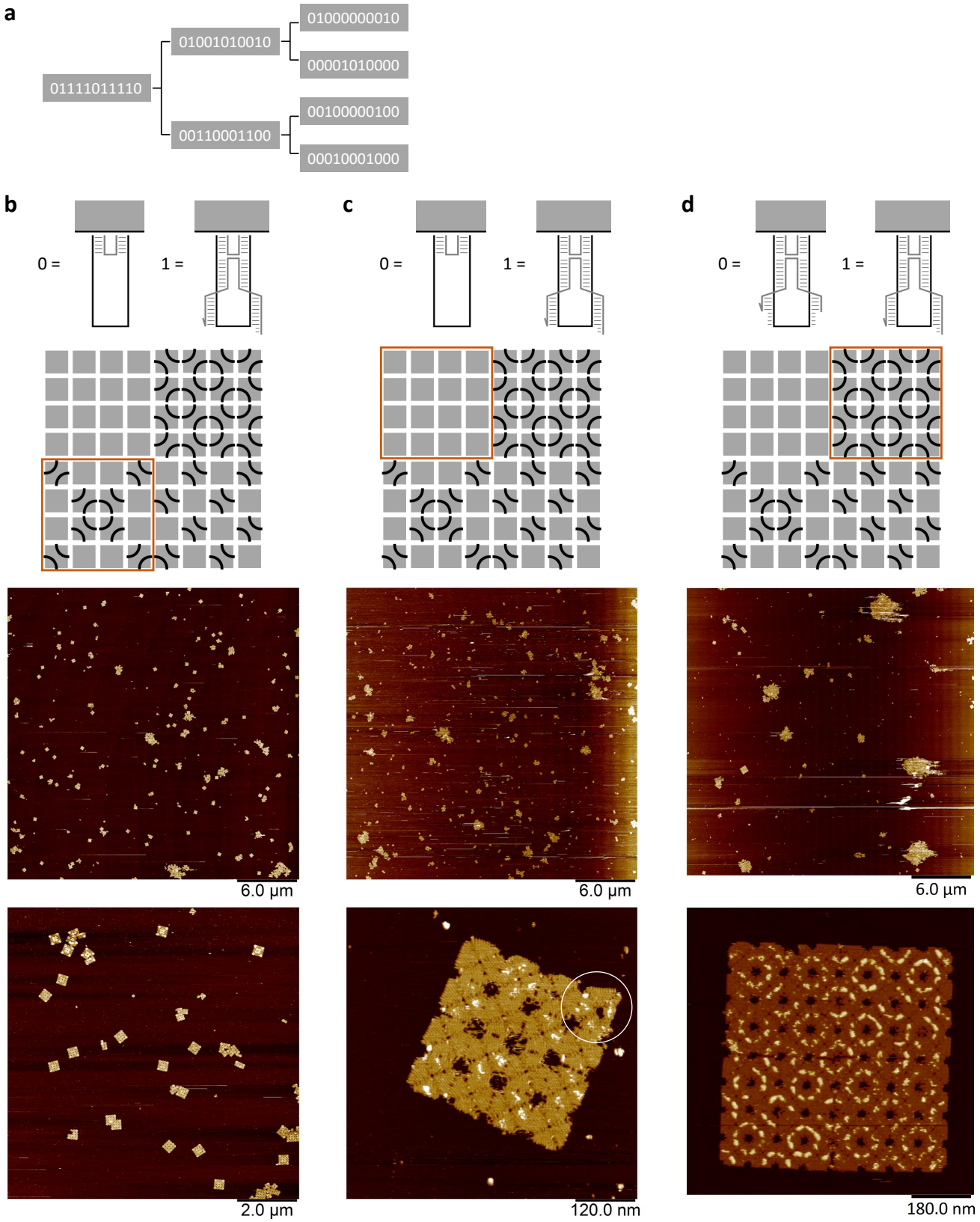


Figure 10: **Non-interactive locations in the edge code. a,** The edge codes used in 8 by 8 fractal arrays. Design diagram and AFM images of **b,** one quadrant, **c,** another quadrant (the white circle highlights an obvious deformation), and **d,** a third quadrant of the 8 by 8 array.

6.4 Inert edges

In the following five subsections, we will focus on fractal assembly of 4 by 4 arrays with an example pattern of Mona Lisa, and use it to figure out the design principles and experimental conditions needed for successful creating uniquely-addressable DNA origami arrays.

First, after we failed to improve the structural integrity of the tiles by using truncated edge staples instead of scaffold loops, as discussed in the previous subsection, we took a different approach by reducing the number of edge staples with double hairpins in the inert edges near the exterior of the array. We hypothesized that the deformation of the tiles were also facilitated by the imbalanced edges: tiles near the exterior of the array had inert edges implemented with a full set of eleven double-hairpin staples, while some of these tiles also had edges with just a few staples (four edge staples for interactions between 2 by 2 arrays and two edge staples for interactions between 4 by 4 arrays). The large difference in the number of edge staples might have created structural imbalance in these tiles, which together with the interactions between adjacent scaffold loops, led to the observed tile deformation. To test this hypothesis, we reduced the number of double-hairpin staples from eleven to five in all inert edges. Indeed, we observed that the yield of the 4 by 4 arrays increased from 2.62% to 10.21% (Supplementary Fig. 11).

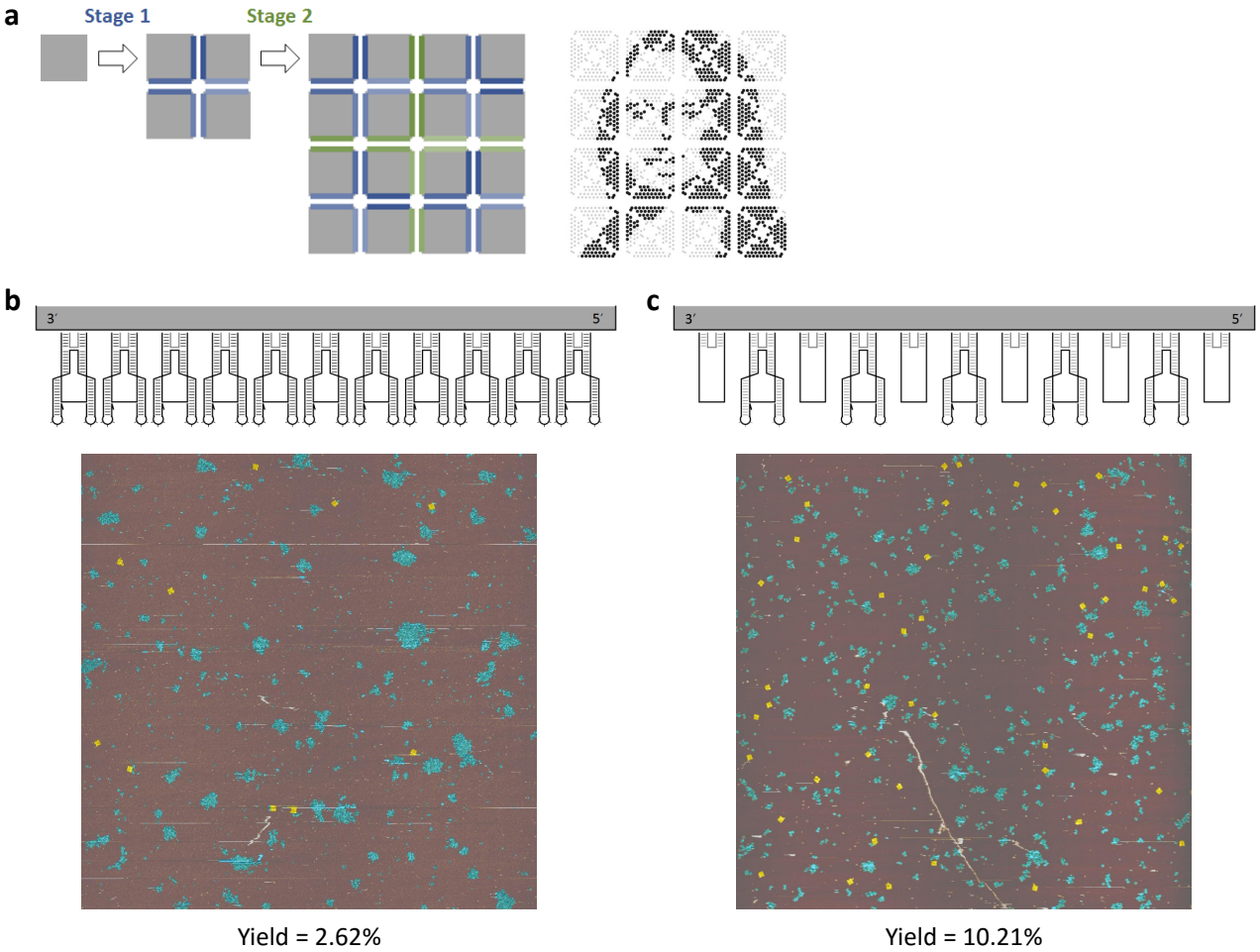


Figure 11: **Inert edges. a,** Design diagram of a 4 by 4 fractal assembly with an example pattern of Mona Lisa. Edge diagram and AFM image of the 4 by 4 array with **b,** eleven double-hairpin edge staples and **c,** five double-hairpin edge staples for each inert tile edge near the exterior of the array.

6.5 Bridge and interior staples near the seams

Starting from the following experiments, we adjusted the protocol to increase pipetting volumes and thus improve pipetting accuracy. With the adjusted protocol, the yield of 4 by 4 arrays increased from 10.21% (Supplementary Fig. 11c) to 16.11% (Supplementary Fig. 12b). To allow for more pixels, we redesigned the bridge staples and interior staples near the seams, which fortunately led to an even higher yield of 24.99%.

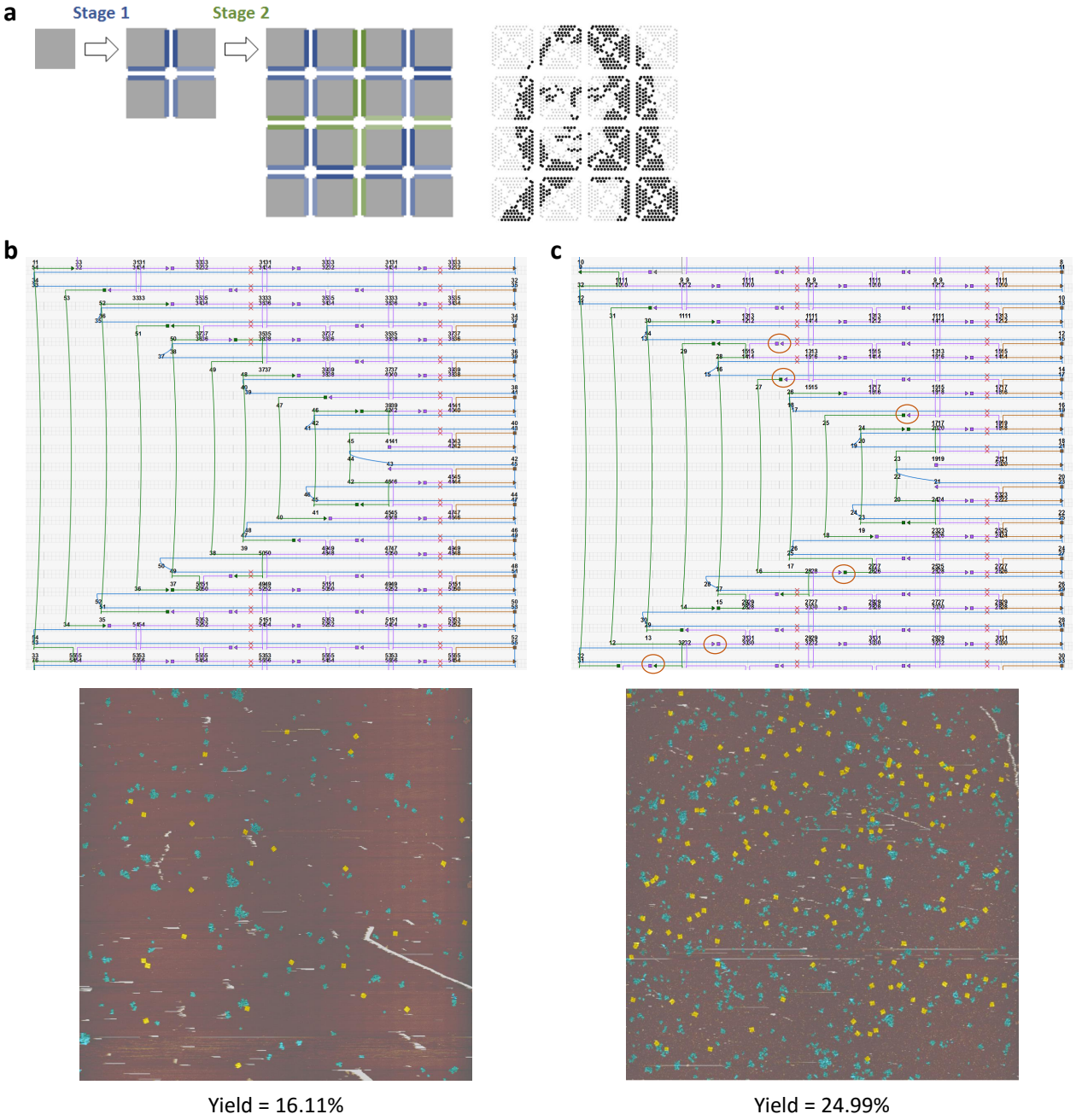


Figure 12: **Bridge and interior staples near the seams. a,** Design diagram of a 4 by 4 fractal assembly with an example pattern of Mona Lisa. Cadnano diagram and AFM image of the 4 by 4 array with **b,** 112 pixels as designed in ref.[18] and **c,** 136 pixels per tile.

6.6 Interactive locations in the edge code

Next, to further improve the yield of fractal assembly, we explored an alternative design of edge codes. In this design, we used different lengths of sticky ends for different stages, by which we hoped to reduce undesired competition caused by incomplete structures from a previous stage and promote desired self-assembly in the current stage.

We kept the second stage the same as before, but changed the edge code in the first stage to eleven edge staples that each has a stacking bond and a one-nucleotide sticky end. Because the total number of stacking bonds (consider a one-nucleotide sticky end as two stacking bonds and a two-nucleotide sticky end as three stacking bonds) are very similar (33 for eleven edge staples with one-nucleotide sticky ends and 32 for eight edge staples with two-nucleotide sticky ends), we expected that the alternative and the original edge codes should have similar binding energies.

However, the yield dropped dramatically with the alternative edge code used in the first stage (Supplementary Fig. 13). We suspect that the increased number of stacking bonds (from eight to eleven) and the decreased interference of short overhangs (from two to one nucleotide) for forming undesired stacking bonds significantly increased the spurious interactions within the first stage and resulted in a large fraction of incorrectly-formed structures, which then aggregated in the second stage. We thus concluded that reducing the spurious interactions within each stage of the fractal assembly is much more important than reducing the competition caused by incomplete structures from a previous stage.

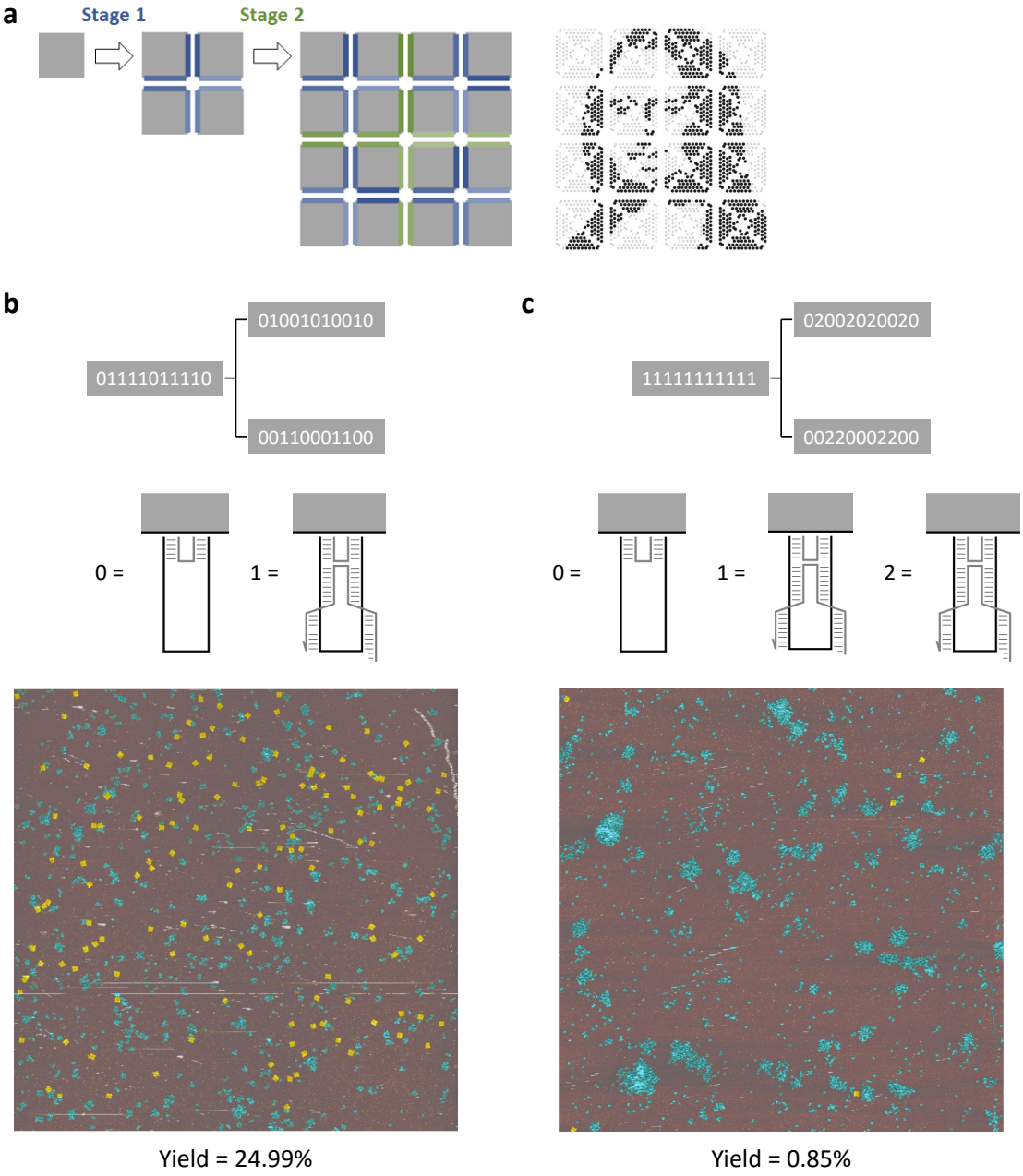


Figure 13: **Interactive locations in the edge code. a,** Design diagram of a 4 by 4 fractal assembly with an example pattern of Mona Lisa. Design diagram and AFM image of the 4 by 4 array using edge staples with **b,** the same length of sticky ends and **c,** different lengths of sticky ends in different stages.

6.7 Automatic liquid handler

Finally, we moved on to test the same fractal assembly procedure carried out by an automatic liquid handler. As there are approximately 3,000 pipetting events involved in creating a 4 by 4 fractal assembly, it is unsurprising that the procedure carried out by a human is prone to errors. In contrast, once a human user writes a program to automatically generate a protocol file that can be executed by a liquid handling robot and debugs the program, complex procedures can be reliably carried out by the robot. Indeed, the yield of 4 by 4 fractal arrays increased from 24.99% to 45.19% when all mixing steps were performed by an Echo 525 liquid handler. The time required for completing the procedure was also reduced from days to minutes.

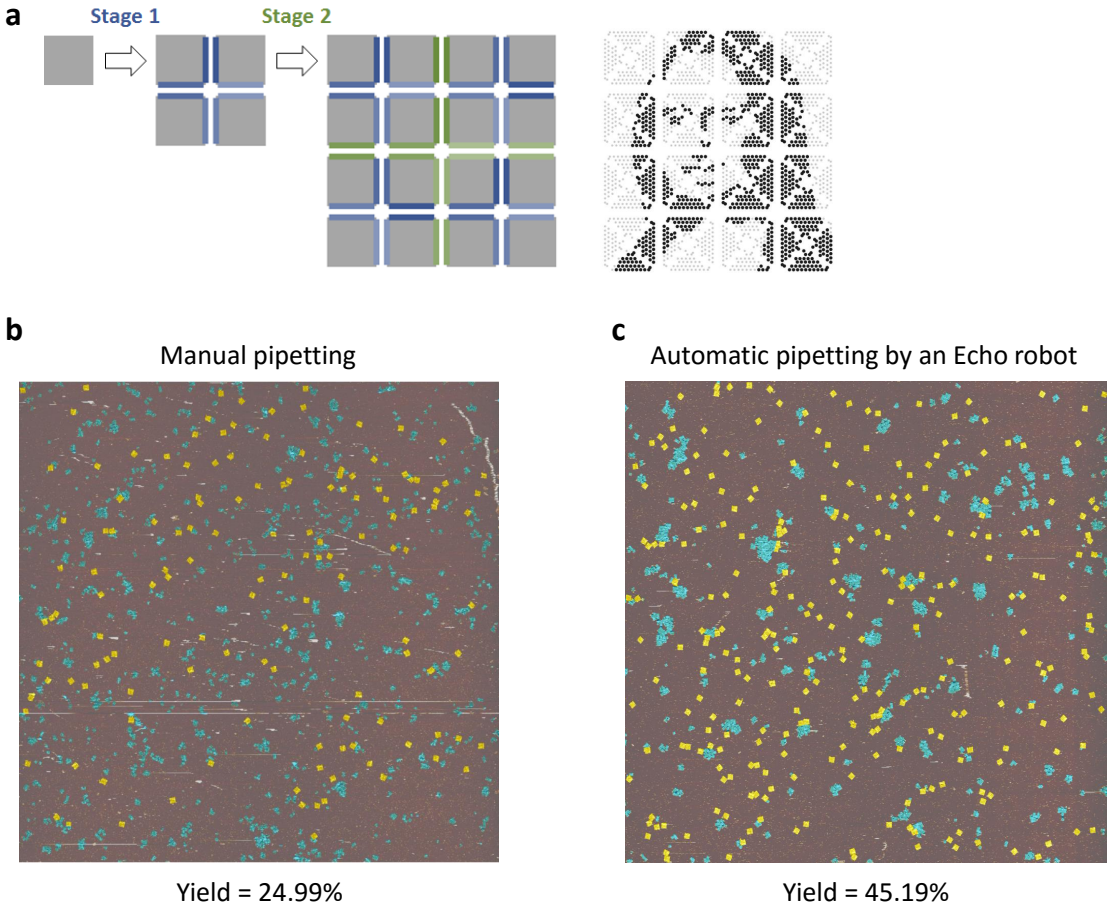


Figure 14: **Automatic liquid handler. a,** Design diagram of a 4 by 4 fractal assembly with an example pattern of Mona Lisa. AFM image of the 4 by 4 array created by **b,** manual pipetting and **c,** automatic pipetting by an Echo 525 liquid handler.

6.8 Plate sealing method

Before using a liquid handling robot, we annealed all DNA origami tiles and arrays in 0.5 mL tubes. Because the liquid handling robot transfers all solution into a 96-well destination plate, to avoid extra transferring steps, we explored a few methods for annealing the tiles and arrays in a 96-well plate. It was surprising how the plate sealing method had a substantial effect on the yield of fractal assembly: when the plate was sealed by a film with no heat applied, the yield dropped to about 0% (Supplementary Fig. 15c). We suspect that the yield suffered from severe evaporation and thus large difference in concentrations of the individual tiles and arrays annealed in separate wells. We then explored two other methods, sealing the plate with a mat (usually used for deep-well storage plates) and cap strips. The sealing mat restored the yield of 4 by 4 arrays to 30.89% and the cap strips to 41.55% (Supplementary Fig. 15d and e).

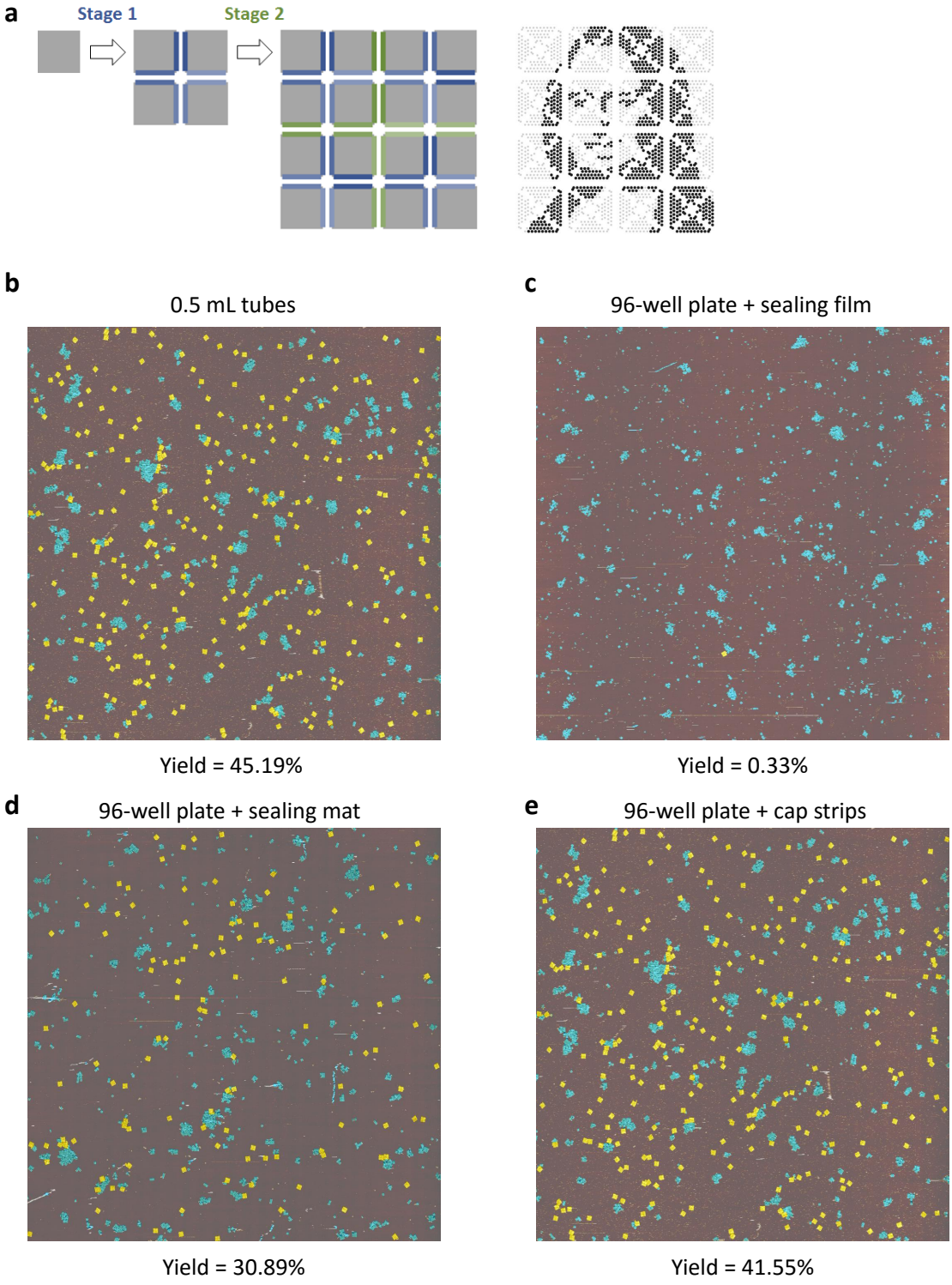


Figure 15: **Plate sealing method. a,** Design diagram of a 4 by 4 fractal assembly with an example pattern of Mona Lisa. AFM image of the 4 by 4 array annealed in **b,** 0.5 mL tubes, **c,** a 96-well plate sealed with a film (no-heat sealing), **d,** a 96-well plate sealed with a mat, **e,** a 96-well plate sealed with cap strips.

6.9 Annealing time

With all design principles and experimental conditions figured out using the 4 by 4 fractal assembly, we proceeded with the 8 by 8 fractal assembly with an example pattern of Mona Lisa. In the 4 by 4 assembly, we used four times longer annealing time for the second stage than the first stage. This was because the concentration of the self-assembly building blocks, after being combined from the previous stage, decreased by four times. If we follow the same rule, the annealing time for the third stage will be four times longer than the second stage, resulting in more than two days of annealing. To justify the time requirement, we performed an experiment with the same annealing time for the first and second stages for creating a quadrant of the 8 by 8 array, and compared the yield with the longer annealing time. Indeed, the yield decreased from 48.72% to 32.02% (Supplementary Fig. 16). We thus concluded that the longer annealing time for each sequential stage is necessary.

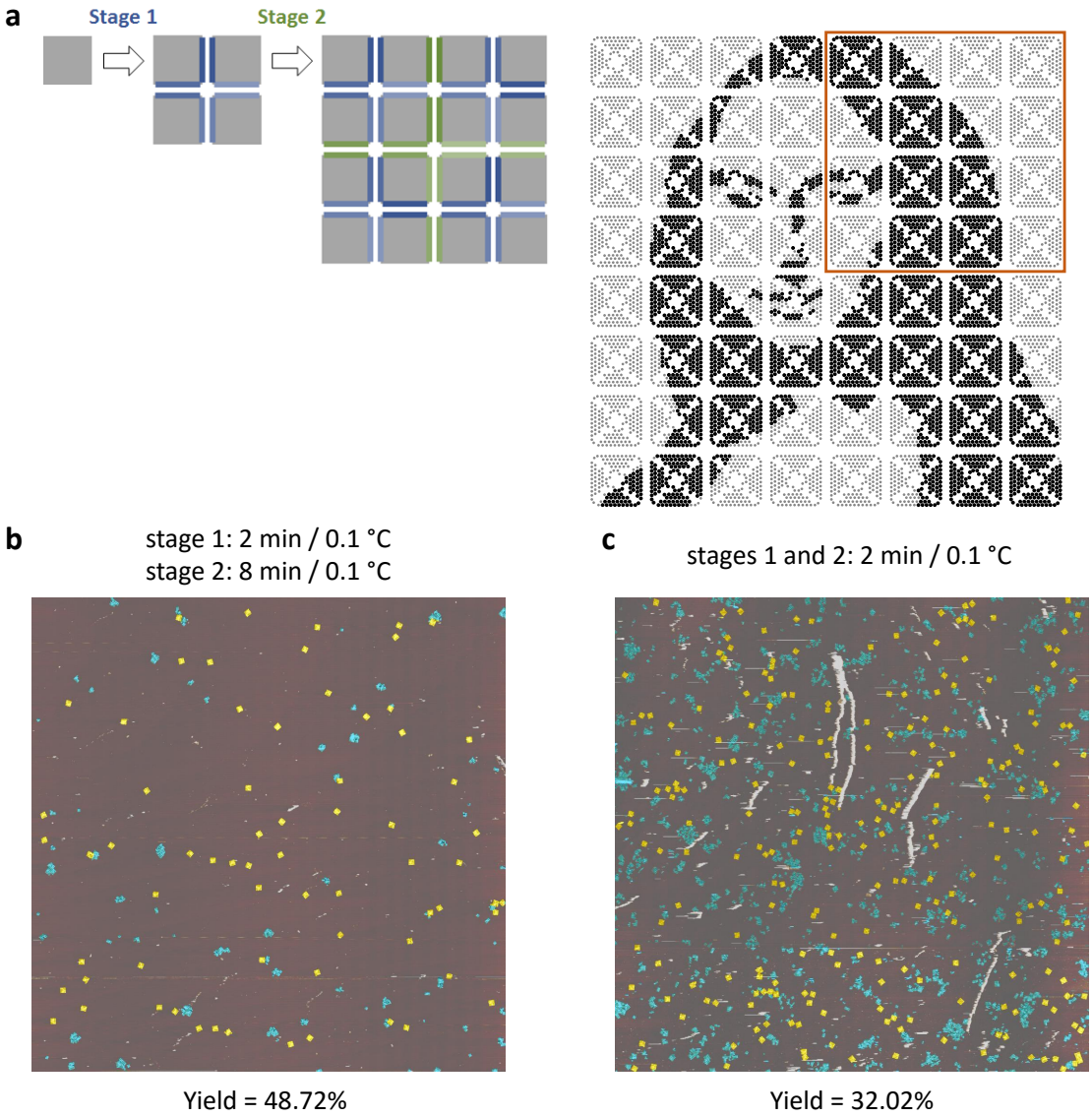


Figure 16: **Annealing time. a,** Design diagram of a quadrant of the 8 by 8 fractal assembly with an example pattern of Mona Lisa. AFM image of the quadrant annealed for **b,** 2 min per 0.1 °C in the first stage and 8 min per 0.1 °C in the second stage, and **c,** 2 min per 0.1 °C in both stages.

7 Analysis on 8 by 8 arrays with patterns

We estimated the correct tile incorporation rate for all complete 8 by 8 arrays with four distinct patterns, each using six to thirty eight AFM images that contain one complete array per image. The imaging artifacts were ignored (e.g. white blobs and smudges, streaks, and tiles damaged by AFM tip). The mean was calculated as $p = \sum n_i / \sum m_i$, where $m_i = 64$ is the total number of tiles and n_i is the number of correct tiles in each image. The error was calculated as $p\sqrt{1-p}/\sqrt{\sum n_i}$, treating the tile incorporation as a Bernoulli probability.[39]

7.1 The Mona Lisa

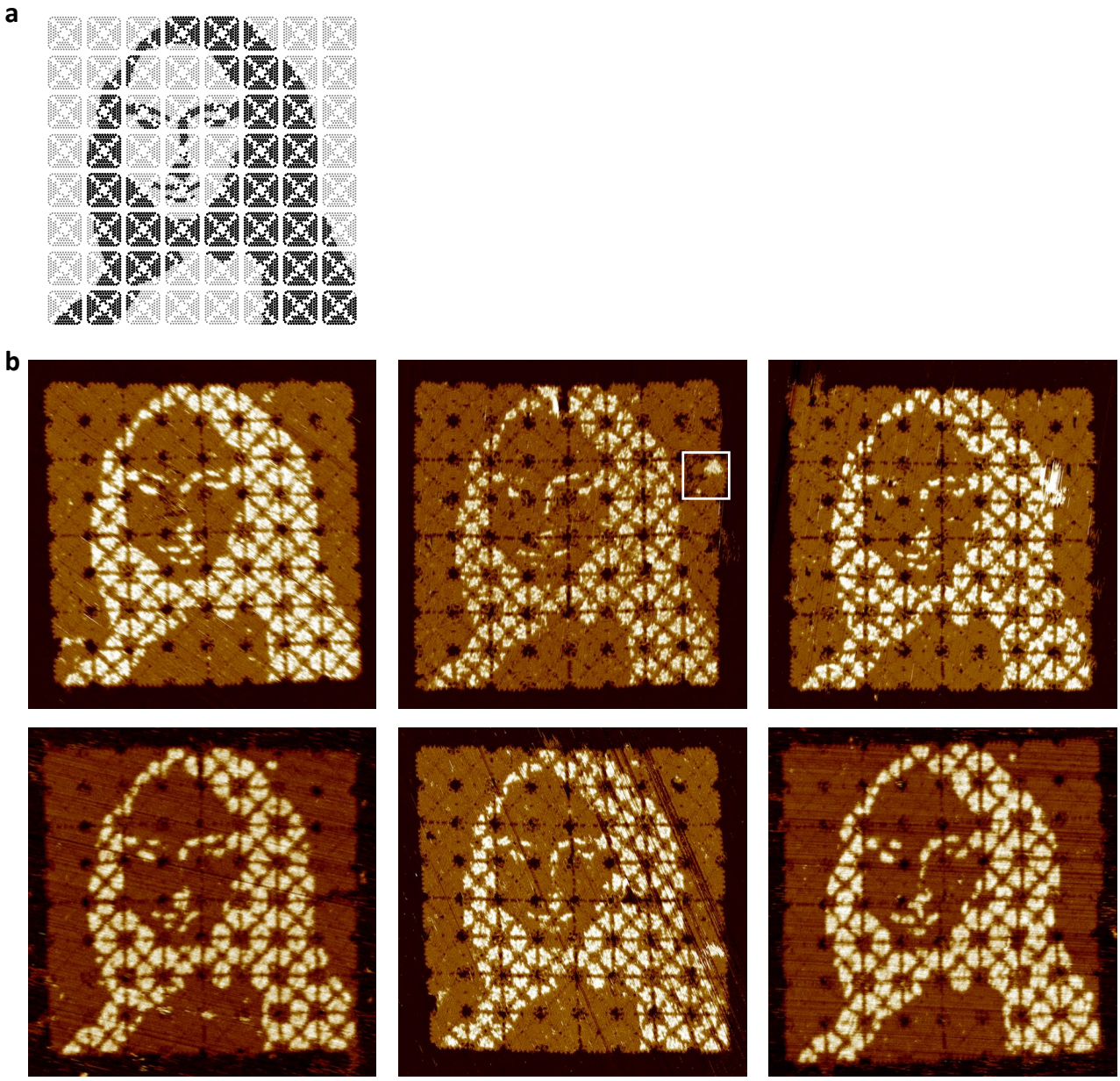


Figure 17: **8 by 8 arrays with a pattern of Mona Lisa. a,** Design diagram. **b,** AFM images. Tiles with an incorrect pattern are highlighted with a white square frame. The correct tile incorporation rate was estimated as $99.79 \pm 0.09\%$, averaged over thirty eight images including the six shown here.

7.2 A rooster

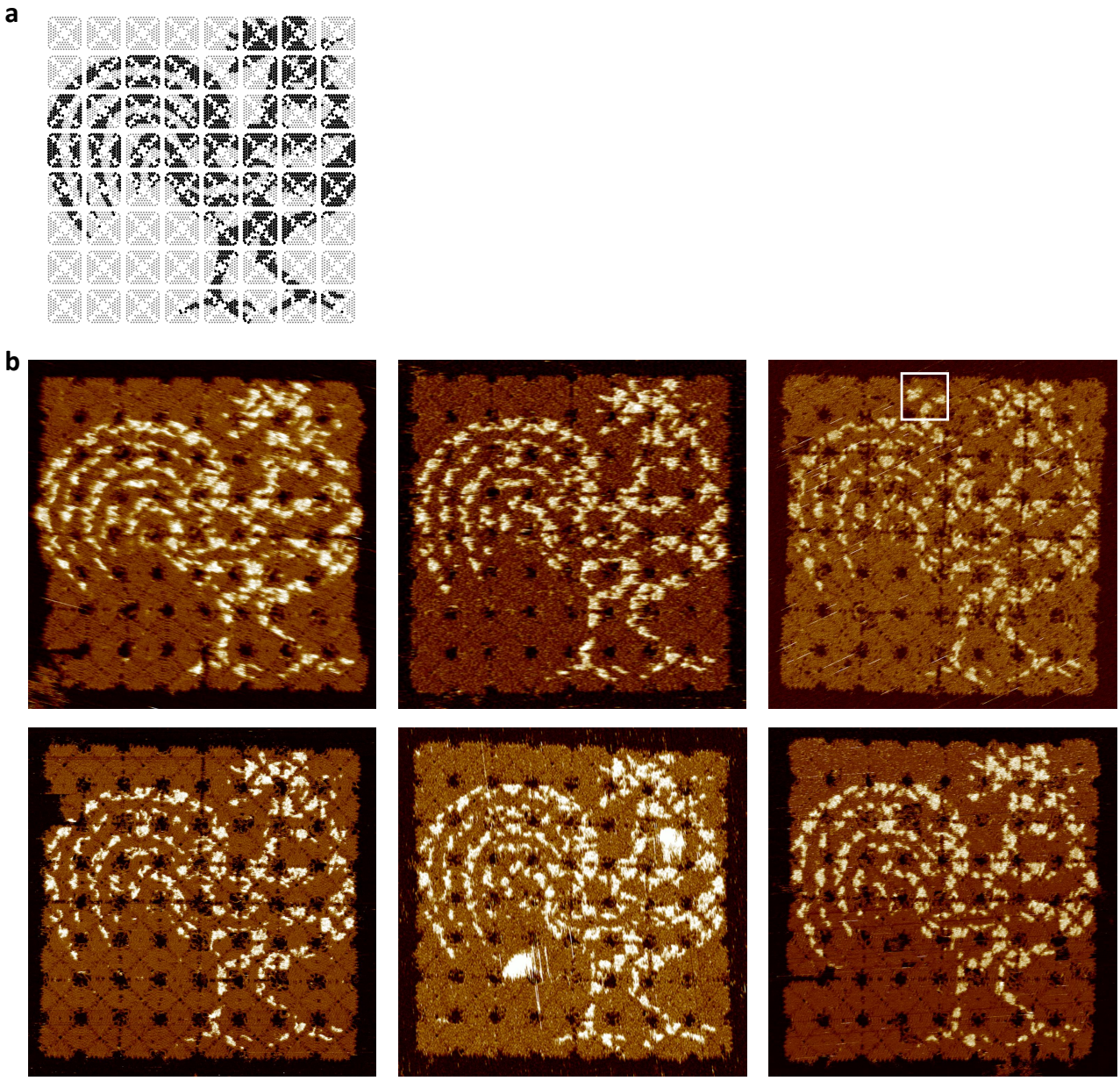


Figure 18: **8 by 8 arrays with a pattern of a rooster. a,** Design diagram. **b,** AFM images. Tiles with an incorrect pattern are highlighted with a white square frame. The correct tile incorporation rate was estimated as $99.74 \pm 0.26\%$, averaged over the six images.

7.3 A bacterium

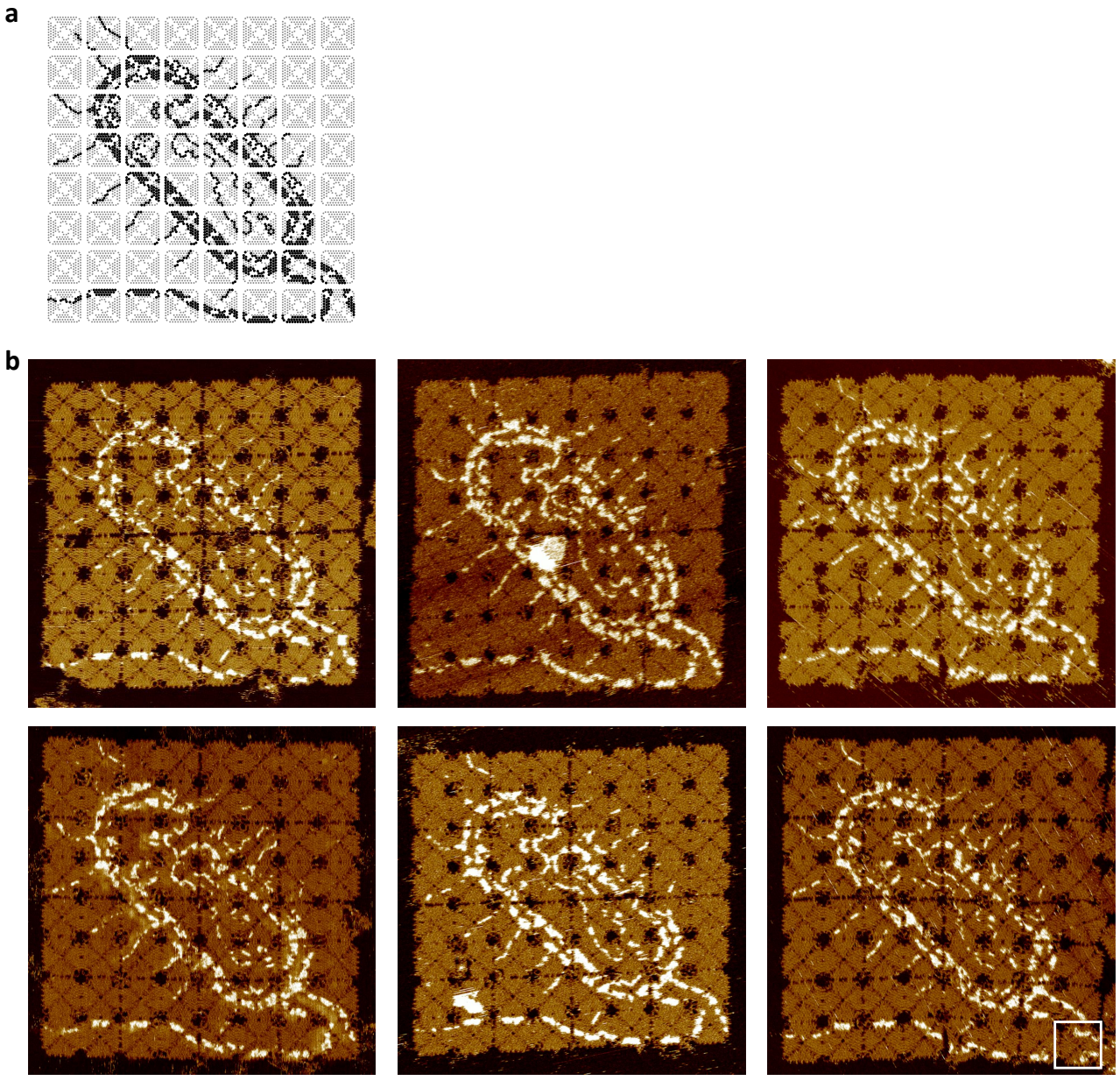


Figure 19: **8 by 8 arrays with a pattern of a bacterium. a,** Design diagram. **b,** AFM images. Tiles with an incorrect pattern are highlighted with a white square frame. The correct tile incorporation rate was estimated as $99.74 \pm 0.26\%$, averaged over the six images.

7.4 A circuit

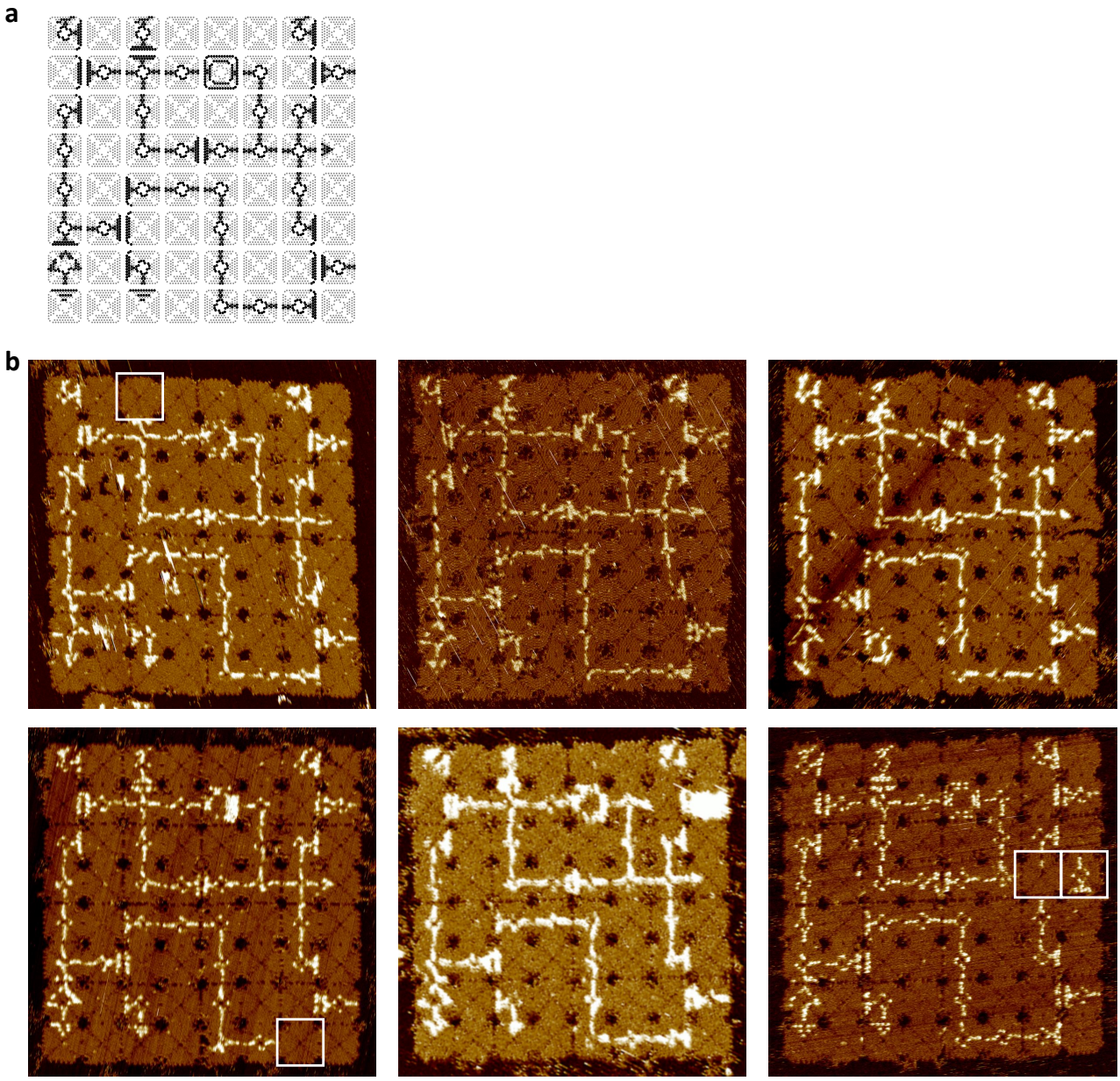
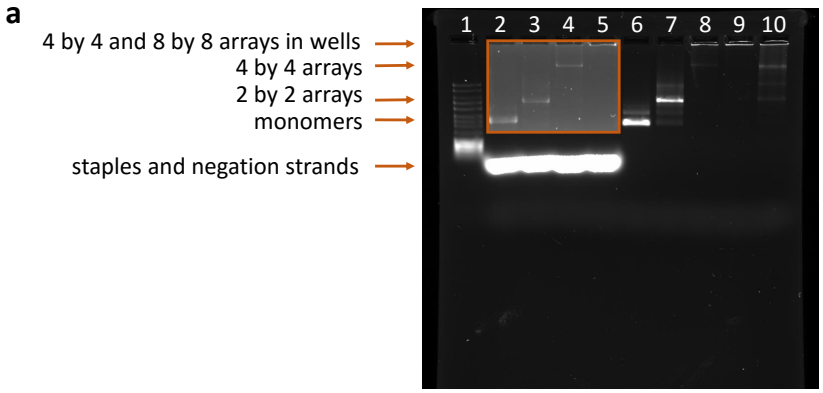


Figure 20: **8 by 8 arrays with a pattern of a photoreceptor circuit.**[31] **a,** Design diagram. **b,** AFM images. Tiles with an incorrect pattern are highlighted with a white square frame. The correct tile incorporation rate was estimated as $98.96 \pm 0.52\%$, averaged over the six images.

8 Spin-filter purification

For some applications, it would be desirable to remove excess staples and negation strands from the origami arrays after fractal assembly. To demonstrate that, we purified the 1 by 1 (monomers) to 8 by 8 arrays using 0.5 mL and 100 kDa spin filters (Amicon, #UFC510096). Each sample was filtrated six times, each time for 3 minutes at 13,000 relative centrifugal force (RCF). We first used gel electrophoresis to analyze the samples before and after purification (Supplementary Fig. 21a). It was clear, from the gel, that the excess staples and negation strands were successfully removed in all samples after purification. The monomers and 2 by 2 arrays migrated at about the same speed on the gel before and after purification, indicating that the structures remained intact after purification. The 4 by 4 and 8 by 8 arrays either partially or completely stayed in the wells, presumably because they were too large to enter the gel. Further, we used AFM to analyze the structural integrity of the larger arrays (Supplementary Fig. 21b). The yield of the 4 by 4 arrays decreased by roughly 5% after purification, but no more structural deformation was observed than in unpurified samples.



Lane #	Purification	Sample
1		1kbp ladder
2	before	1x1 plain
3		2x2 plain
4		4x4 plain
5		8x8 plain
6	after	1x1 plain
7		2x2 plain
8		4x4 plain
9		8x8 plain
10		4x4 Mona Lisa

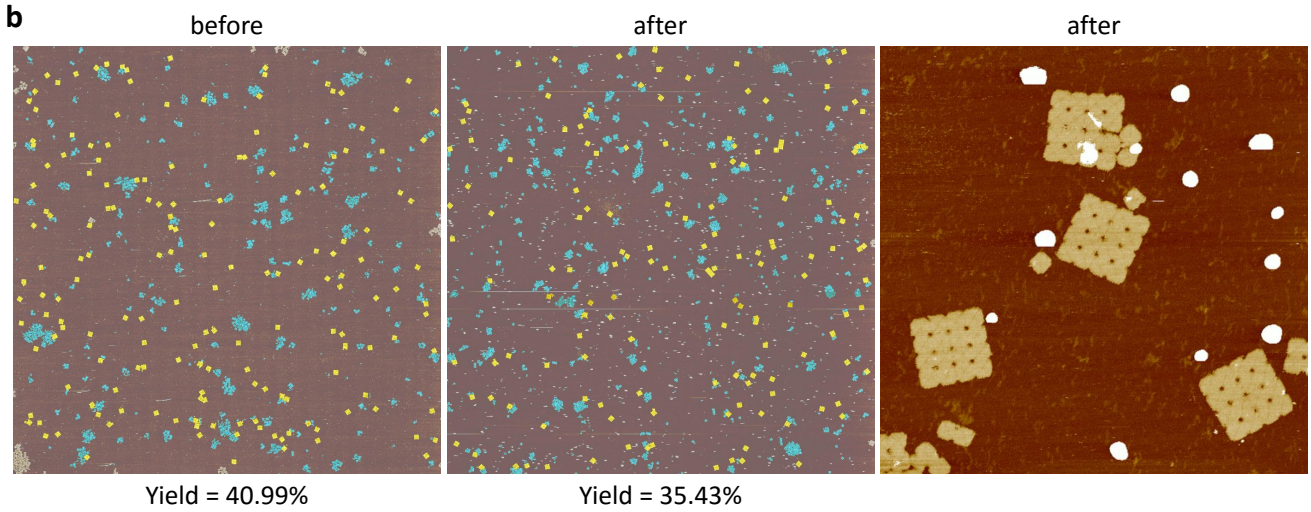


Figure 21: **Spin-filter purification. a,** 1 by 1 to 8 by 8 arrays before and after purification, analyzed on a 0.5% agarose gel, ran at 80 mV for 2 hours. The orange box highlights an area where we increased the contrast to show bands at low concentrations, which are otherwise not visible. **b,** AFM images of the 4 by 4 arrays before and after purification.

9 Strengthening the origami arrays after fractal assembly

In addition to removing excess staple and negation strands, in some cases, it would also be desirable to separate the well-formed arrays from incomplete or aggregated structures. It has been shown that multi-origami structures can be successfully separated based on their sizes, for example using glycerol-gradient centrifugation or size-exclusion chromatography.[27,40] A possible concern for applying these methods to separate fractal arrays is that the interactions between origami tiles may be too weak to survive the separation process. To demonstrate that the origami arrays can be strengthened after fractal assembly, we added a 10-fold excess of a full set of 44 edge staples, that each have two stacking bonds, to the 4 by 4 arrays shown in Supplementary Fig. 22a. After incubating the arrays with the additional edge staples at room temperature for 1 hour, we observed that the staples were integrated into the origami tiles (Supplementary Fig. 22b). With these additional edge staples, all interactions between origami tiles are increased from a total of 32 and 16 to 38 and 30 stacking bonds, respectively (each 2 nucleotide sticky end is counted as 3 stacking bonds). With the same method, the weakest interactions in the 8 by 8 arrays could also be increased from 8 to 26 stacking bonds, which we believe will be strong enough to survive the separation process.

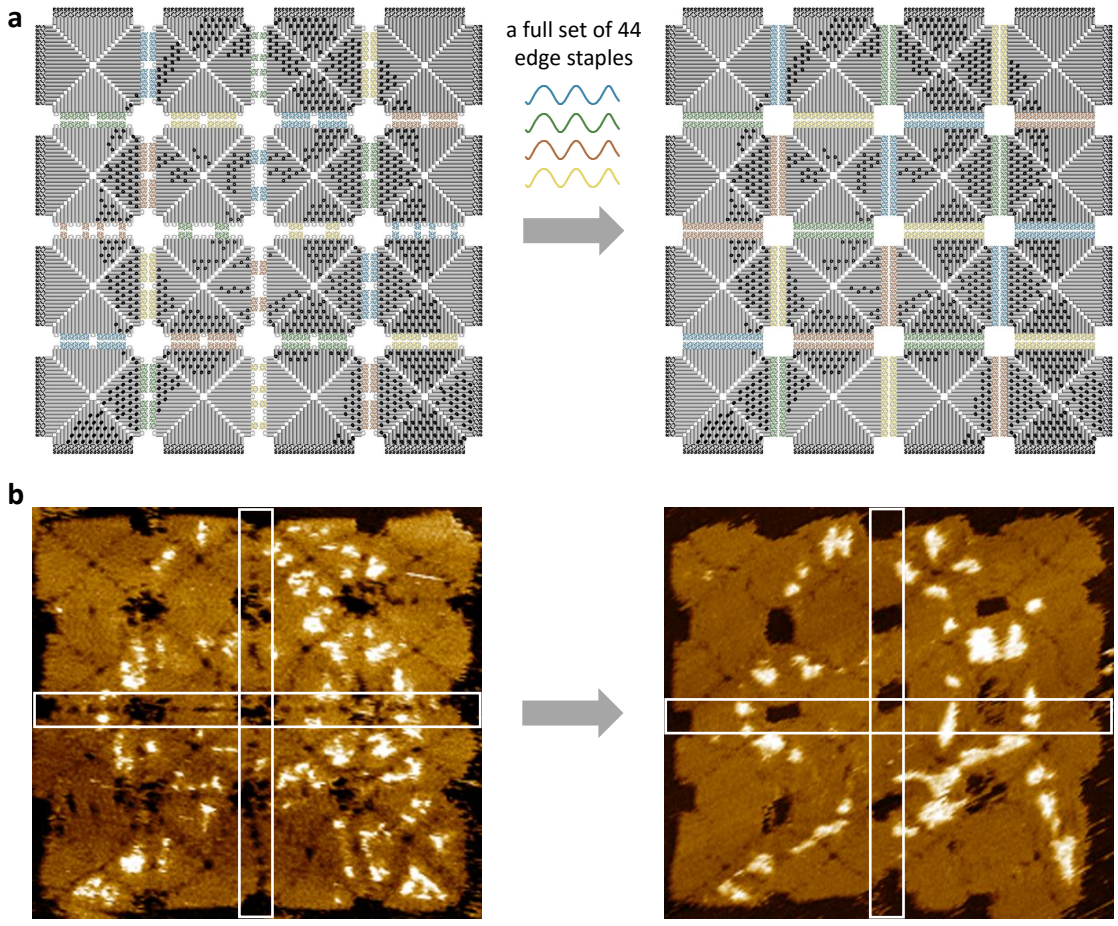


Figure 22: **Strengthening the origami arrays after fractal assembly. a,** Design diagram of a 4 by 4 array strengthened by adding a full set of 44 edge staples with stacking bonds after the fractal assembly. **b,** AFM images of the 4 by 4 array before and after strengthening. The white boxes highlight the most obvious changes in the number of edge staples.

10 Cadnano diagram

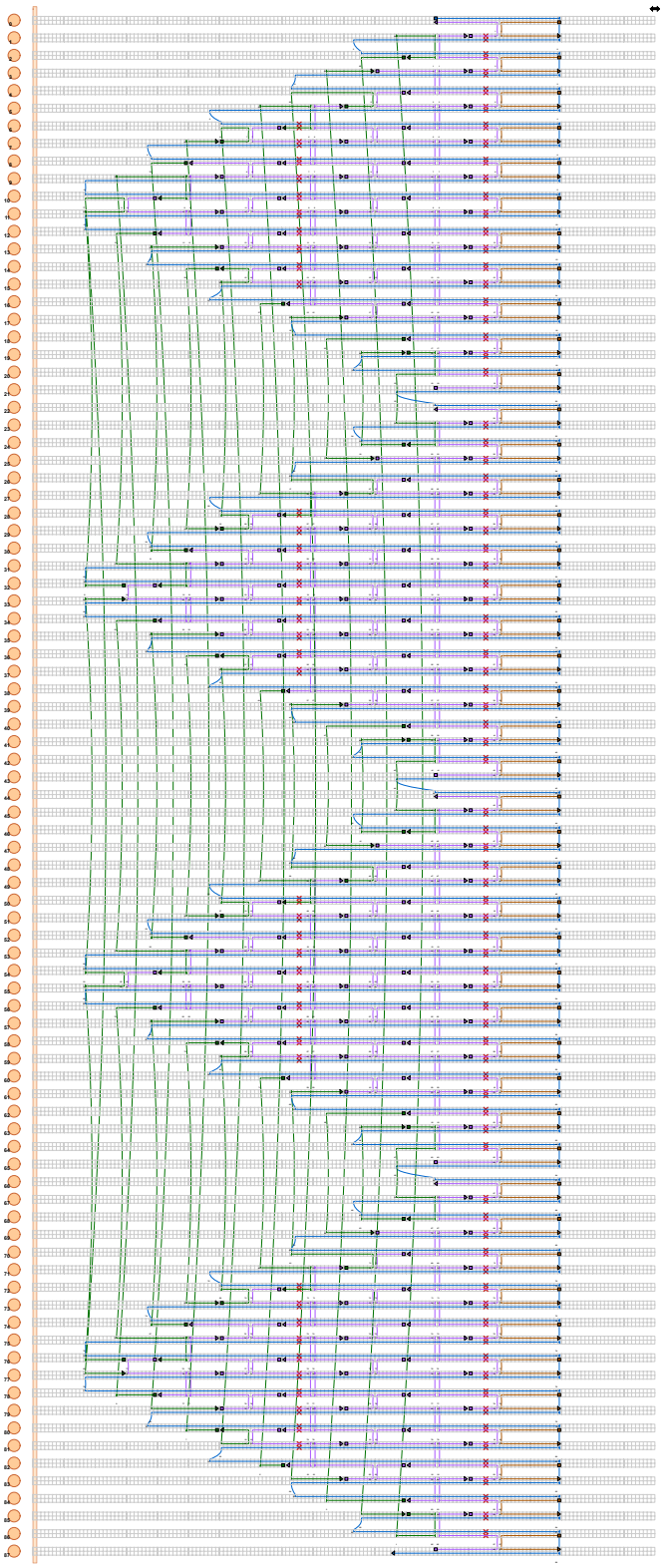


Figure 23: **Cadnano[41] diagram of the square DNA origami tile with 136 interior staples.**

11 DNA sequences

Table S1: Interior staples.

Name	Sequence
Reg-T1R01C6	TCATTTGCTAATAGTAGTAGCATT
Reg-T1R03C5	CAACTAAAGTACGGTGGGATGGCT
Reg-T1R03C6	TTTCATTGAGTAGATTTAGTTTCTATATTT
Reg-T1R04C5	TAGAGCTTCAGACCGGAAGCAAACCTATTATA
Reg-T1R05C6	GTCAGGAAGAGGTCATTTTTGCTCTGGAAG
Reg-T1R06C3	TTAAGAGGGTCCAATACTGCGGATAGCGAG
Reg-T1R06C5	GTCAGAAGATTGAATCCCCCTCAACCTCGTTT
Reg-T1R07C4	AAATATTCCAAAGCGGATTGCATCGAGCTTCA
Reg-T1R07C6	AACAGTTAGGTCTTTACCCTGATCCAACAG
Reg-T1R08C3	AGGCTTTTCAGGTAGAAAGATTCAATTACC
Reg-T1R08C5	ACCAGACGGAATACCACATTCAACGAGATGGT
Reg-T1R09C2	CATTATTAGCAAAAGAAGTTTTGC
Reg-T1R09C4	AGATTTAGACGATAAAAACCAAAAATCGTCAT
Reg-T1R09C6	ATACATACAACACTATCATAACATGCTTTA
Reg-T1R10C1	AGTCAGGACATAGGCTGGCTGACCTTTGAAAG
Reg-T1R10C3	TTATGCGATTGACAAGAACCGGAGGTCAAT
Reg-T1R10C5	TTAATTTCCAACGTAACAAAGCTGTCCATGTT
Reg-T1R11C2	GAGTAATCTTTTAAGAACTGGCTCCGGAACAA
Reg-T1R11C4	ACCCAAATAACTTTAATCATTGTGATCAGTTG
Reg-T1R11C6	GTGAATATAGTAAATTGGGCTTTAATGCAG
Reg-T1R12C3	CATAAGGGACACTAAAACACTCACATTAAA
Reg-T1R12C5	ACTTAGCCATTATACCAAGCGCGAGAGGACTA
Reg-T1R13C2	AAAAGAATAACCGAACTGACCAACTTCATCAA
Reg-T1R13C4	CCCCAGCGGGAACGAGGCGCAGACTATTCATT
Reg-T1R13C6	ACAACGGAAATCCGCGACCTGCCTCATTCA
Reg-T1R14C3	CGGGTAAAATTCGGTCGCTGAGGAATGACA
Reg-T1R14C5	AAGACTTTGGCCGCTTTTGCGGGATTAAACAG
Reg-T1R15C4	GAGTTAAATTCATGAGGAAGTTTCTCTTTGAC
Reg-T1R15C6	CTCAGCAGGCTACAGAGGCTTTAACAAAGT
Reg-T1R16C5	CTTGATACTGAAAATCTCCAAAAAAGCGGAGT
Reg-T1R17C4	TTTCACGTCGATAGTTGCGCCGACCTTGCAGG
Reg-T1R17C6	CAAAAGGTTCGAGGTGAATTTCTCGTCACC
Reg-T1R19C6	GTTAGTAACTTTCAACAGTTTCAAAGGCTC
Reg-T1R21C5	CCATGTACCGTAACACTGTAGCATTCCACAGATTCCAGAC
Reg-T2R01C6	ACCCTCATTCAGGGATAGCAAGCC
Reg-T2R03C5	TTAGGATTAGCGGGGTGGAACCTA
Reg-T2R03C6	GTACCAGGTATAGCCCGGAATAGAACCGCC
Reg-T2R04C5	TTATTCTGACTGGTAATAAGTTTTAACAAATA
Reg-T2R05C6	CAGTGCCCCCCCTGCCTATTTCTTTGCTCA
Reg-T2R06C3	GTCTCTGACACCCTCAGAGCCACATCAAAA

Name	Sequence
Reg-T2R06C5	AATCCTCAACCAGAACCACCACCAGCCCCCTT
Reg-T2R07C4	GAGCCGCCTTAAAGCCAGAATGGAGATGATAC
Reg-T2R07C6	GCCAGCAGCCTTGATATTCACAAACGGGGT
Reg-T2R08C3	TCACCGGAAACGTCACCAATGAATTATTCA
Reg-T2R08C5	ATTAGCGTCCGTAATCAGTAGCGAATTGAGGG
Reg-T2R09C2	AGGCCGGAACCAGAGCCACCACCG
Reg-T2R09C4	TAGCAGCATTGCCATCTTTTCATACACCCTCA
Reg-T2R09C6	AGTTTGCGCATTTTCGGTCATAGAGCCGCC
Reg-T2R10C1	GCCATTTGCAAACGTAGAAAATACCTGGCATG
Reg-T2R10C3	TTAAAGGTACATATAAAAGAAACAAACGCA
Reg-T2R10C5	AGGGAAGGATAAGTTTATTTTGTCAGCCGAAC
Reg-T2R11C2	AGGTGGCAGAATTATCACCGTCACCATTAGCA
Reg-T2R11C4	ACCACGGATAAATATTGACGGAAAACCATCGA
Reg-T2R11C6	TAGAAAAGGCGACATTCAACCGCAGAATCA
Reg-T2R12C3	ATAATAACTCAGAGAGATAACCCGAAGCGC
Reg-T2R12C5	AAAGTTACGCCCAATAATAAGAGCAGCCTTTA
Reg-T2R13C2	CGCTAATAGGAATACCCAAAAGAAATACATAA
Reg-T2R13C4	TGAGTTAACAGAAGGAAACCGAGGGCAAAGAC
Reg-T2R13C6	ATGAAATGAAAAGTAAGCAGATACAATCAA
Reg-T2R14C3	ATTAGACGGAGCGTCTTTCCAGAGCTACAA
Reg-T2R14C5	CAGAGAGAACAAAATAAACAGCCATTAAATCA
Reg-T2R15C4	TGCCAGTTATAACATAAAAACAGGACAAGAAT
Reg-T2R15C6	ATCCCAAAAAAATGAAAATAGCAAGAAACA
Reg-T2R16C5	AGATTAGTATATAGAAGGCTTATCCAAGCCGT
Reg-T2R17C4	CAAATCAGTGCTATTTTGCACCCAGCCTAATT
Reg-T2R17C6	TAAGAACGGAGGTTTTGAAGCCTATTATTT
Reg-T2R19C6	CTTATCACTCATCGAGAACAAGCGGTATTC
Reg-T2R21C5	AGCTAATGCAGAACGCGAGAAAAATAATATCCTGTCTTTC
Reg-T3R01C6	AGAATATCAGACGACGACAATAAA
Reg-T3R03C5	TCATATGCGTTATACAAAGGCGTT
Reg-T3R03C6	CCAGTATGAATCGCCATATTTAGTAATAAG
Reg-T3R04C5	AAATAAGAACTTTTTCAAATATATCTGAGAGA
Reg-T3R05C6	ATTTCATGACCGTGTGATAAATAATTCTTA
Reg-T3R06C3	TATATAACGTAAATCGTCGCTATATTTGAA
Reg-T3R06C5	CTACCTTTAGAATCCTTGAAAACAAGAAAACA
Reg-T3R07C4	TTTCCCTTTTAACCTCCGGCTTAGCAAAGAAC
Reg-T3R07C6	GCTTAGAATCAAAATCATAGGTTTTAGTTA
Reg-T3R08C3	TTACCTTTACAATAACGGATTCGCAAAATT
Reg-T3R08C5	AAATTAATACCAAGTTACAAAATCCTGAATAA
Reg-T3R09C2	CGGGAGAATTTAATGGAAACAGTA
Reg-T3R09C4	CTTTGAATTACATTTAACAATTTCTAATTAAT
Reg-T3R09C6	GCGAATTATGAAACAAACATCATAGCGATA
Reg-T3R10C1	GTAGATTTGTTATTAATTTTAAAAAACAATTC
Reg-T3R10C3	ATTTGCACCATTTTGCGGAACAAATTTGAG

Name	Sequence
Reg-T3R10C5	TGGAAGGGAGCGGAATTATCATCAACTAATAG
Reg-T3R11C2	AACATTATGTAAAACAGAAATAAATTTTACAT
Reg-T3R11C4	CCAGAAGGTTAGAACCTACCATATCCTGATTG
Reg-T3R11C6	ATTATCAGTTTGGATTATACTTGCGCAGAG
Reg-T3R12C3	GATTTAGATTGCTGAACCTCAAAGTATTAA
Reg-T3R12C5	ATTAGAGCAATATCTGGTCAGTTGCAGCAGAA
Reg-T3R13C2	GCATCACCAGTATTAGACTTTACAGTTTGAGT
Reg-T3R13C4	CCTCAATCCGTCAATAGATAATACAGAAACCA
Reg-T3R13C6	ACAGTTGTTAGGAGCACTAACATATTCCTG
Reg-T3R14C3	CACCGCCTGAAAGCGTAAGAATACATTCTG
Reg-T3R14C5	GATAAAACTTTTTGAATGGCTATTTTCACCAG
Reg-T3R15C4	AGACAATAAGAGGTGAGGCGGTCATATCAAAC
Reg-T3R15C6	ATGCGCGTACCGAACGAACCACGCAAATCA
Reg-T3R16C5	TCACACGATGCAACAGGAAAAACGGAAGAACT
Reg-T3R17C4	CCAGCCATCCAGTAATAAAAGGGACGTGGCAC
Reg-T3R17C6	AATACCTATTTACATTGGCAGAAGTCTTTA
Reg-T3R19C6	TTAACCGTCACTTGCCTGAGTACTCATGGA
Reg-T3R21C5	CTAAACAGGAGGCCGATAATCCTGAGAAGTGTCACGCAAA
Reg-T4R01C6	GCGCGTACTTTCCTCGTTAGAATC
Reg-T4R03C5	AAAGCCGGCGAACGTGTGCCGTAA
Reg-T4R03C6	GGAAGGGGGCAAGTGTAGCGGTGCTACAGG
Reg-T4R04C5	AGCACTAAAAAGGGCGAAAAACCGAAATCCCT
Reg-T4R05C6	GGCGATGTTTTTGGGGTCGAGGGCGAGAAA
Reg-T4R06C3	TGAGTGTTCAGCTGATTGCCCTTGCGCGGG
Reg-T4R06C5	TATAAATCGAGAGTTGCAGCAAGCGTCGTGCC
Reg-T4R07C4	GGCCCTGAAAAAGAATAGCCCGAGCGTGGACT
Reg-T4R07C6	CTGGTTTGTTCCGAAATCGGCATCTATCAG
Reg-T4R08C3	GAGAGGCGACAACATACGAGCCGCTGCAGG
Reg-T4R08C5	AGCTGCATAGCCTGGGGTGCCTAAGTAAAACG
Reg-T4R09C2	AATTCCACGTTTGCGTATTGGGCG
Reg-T4R09C4	AAGTGTAATAATGAATCGGCCAACCACCGCCT
Reg-T4R09C6	CTAACTCCCAGTCGGGAAACCTGGTCCACG
Reg-T4R10C1	GAATTCGTGCCATTCGCCATTCAGTTCCGGCA
Reg-T4R10C3	TCGACTCTGAAGGGCGATCGGTGCGGCCTC
Reg-T4R10C5	ACGGCCAGTACGCCAGCTGGCGAACATCTGCC
Reg-T4R11C2	ACTGTTGGAGAGGATCCCCGGGTACCGCTCAC
Reg-T4R11C4	TTCGCTATTGCCAAGCTTGCATGCGAAGCATA
Reg-T4R11C6	GTGCTGCCCCAGTCACGACGTTTGAGTGAG
Reg-T4R12C3	AGGAAGATCATTAAATGTGAGCGTTTTTAA
Reg-T4R12C5	AGTTTGAGATTCTCCGTGGGAACAATTCGCAT
Reg-T4R13C2	TTCATCAACGCACTCCAGCCAGCTGCTGCGCA
Reg-T4R13C4	CCCGTCGGGGGACGACGACAGTATCGGGCCTC
Reg-T4R13C6	ATTGACCCGCATCGTAACCGTGAGGGGGAT
Reg-T4R14C3	CCAATAGGAAACTAGCATGTCAAGGAGCAA

Name	Sequence
Reg-T4R14C5	TAAATTTTTGATAATCAGAAAAGCACAAAGGC
Reg-T4R15C4	ACCCCGGTTGTTAAATCAGCTCATAGTAACAA
Reg-T4R15C6	CAGGAAGTAATATTTTGTTAAAAACGGCGG
Reg-T4R16C5	TATCAGGTAAATCACCATCAATATCAATGCCT
Reg-T4R17C4	AGACAGTCCATTGCCTGAGAGTCTTCATATGT
Reg-T4R17C6	ACCGTTCATTTTTGAGAGATCTCCCAAAAA
Reg-T4R19C6	CCTTTATCATATATTTTAAATGGATATTCA
Reg-T4R21C5	AATCATACAGGCAAGGCAGAGCATAAAGCTAAGGGAGAAG

Table S2: Bridge staples.

Name	Sequence
Bri-T1R02C5	GATACATTTCGCTTTTTTGACCCTGTAAT
Bri-T1R05C4	AAGCGAACAATTGCTGAATATAATGCTGTATTTTTTTGTGAGAAAGGCCGG
Bri-T1R07C2	TGGATAGCAAGCCCGATTTTTAATCGTAAACGCCAT
Bri-T1R08C1	CAGAGGGGGTTTTGCCTTCCTGTAGCCAGCT
Bri-T1R12C1	AGGACAGATGATTTTTTCACCAGTAGCACCATTACCGACTTGA
Bri-T1R14C2	TGCCACTACTTTTTTTGCCACCCTC
Bri-T1R16C3	ACAACCATTTTTTCATACATGGCTTTTAAGCGCA
Bri-T1R18C5	GAGAATAGAAAGGAACAACTATTTTCTCAAGAGAAGGA
Bri-T1R19C5	TGTCGTCTCAGCCCTCATATTTTTTTCGCCACCCTCAGGTGTATC
Bri-T2R02C5	ACCGTACTCAGGTTTTTGATCTAAAGTTT
Bri-T2R05C4	AGGAGTGTAAACATGAAAGTATTAAGAGGCTTTTTTTGCGAATAATAATTT
Bri-T2R07C2	AGAACCGCATTTACCGTTTTACCGATATATACGTAA
Bri-T2R08C1	GAACCGCCTCTTTACCTAAAACGAAAGAGGC
Bri-T2R10C0	GGAATTAGAGCTTTTTTTTCAGACCAGGCGCGTTGGGAAGATTTTTTTTCCAGGCAAAGC
Bri-T2R12C1	ATTAAGACTCCTTTTTAATATACAGTAACAGTACCGAAATTGC
Bri-T2R14C2	AACTGAACATTTTTTTTGAATAACC
Bri-T2R16C3	TTTTATCTTTTTTATCCAATCGCAAGAGTTGGGT
Bri-T2R18C5	TTTTATTTTCATCGTAGGAATTTTTAGCCTGTTTAGTA
Bri-T2R19C5	TAATCGGCCATCCTAATTTTTTTTTTTTTTCGAGCCAACAACGCC
Bri-T3R02C5	AACATGTAATTTTTTTTGAAACCAATCAA
Bri-T3R05C4	GCGAGAAAATAAACACCGGAATCATAATTATTTTTTTCGCCCAATAGCAAG
Bri-T3R07C2	TTGCTTCTTATATGTATTTTACGCTAACGGAGAATT
Bri-T3R08C1	CATAAATCAATTTAGTCAGAGGGTAATTGAG
Bri-T3R12C1	GACAACTCGTATTTTTTCCTGTGTGAAATTGTTATCCGAGCTC
Bri-T3R14C2	GCCACGCTGTTTTTTTACCAGTGAG
Bri-T3R16C3	GCCAACATTTTTTCCACTATTAAAGAAATAGGGT
Bri-T3R18C5	CAAACTATCGGCCTTGCTGGTTTTTGAGCTTGACGGGG
Bri-T3R19C5	CTGTCCATTTTTATAATCATTTTTTTCTTAATGCGCCCACGCTGC
Bri-T4R02C5	GCGTAACCACCATTTTTGAGTAAAAGAGT
Bri-T4R05C4	CCAACGTCATCGGAACCCTAAAGGGAGCCCTTTTTTTGAACAATATTACCG
Bri-T4R07C2	ACGGGCAAGTTCCAGTTTTTTCTGACCTGCAACAGT
Bri-T4R08C1	CCAGGGTGGTTTTGCAAATGAAAAATCTAAA
Bri-T4R10C0	AATCATGGTCATTTTTTTTTTTGCCCGAACTCAGGTTTAACTTTTTTTTCAGTATGTTAG
Bri-T4R12C1	CCGCTTCTGGTTTTTTCGTTAATAAAACGAACTAAATTATACC
Bri-T4R14C2	CAAAAATAATTTTTTTTGTTTAGAC
Bri-T4R16C3	ACAAGAGTTTTTTTCGCGTTTTAATTCAAAAAGA
Bri-T4R18C5	GAGTAATGTGTAGGTAAAGATTTTTTGTTTTAAATATG
Bri-T4R19C5	ACTTTTGCATCGGTTGTACTTTTTTTAACCTGTTTAGGACCATTA

Table S3: Inert edge staples.

Name	Sequence
Edg-T1R02C7-DHP	GTGTCGTAGACACTCCCAATTCTGCGAACCCATATAACAGTTGATGTGTCGTAGACAC
Edg-T1R06C7-DHP	GTGTCGTAGACACCCATAAATCAAAAATCCAGAAAACGAGAATGAGTGTCGTAGACAC
Edg-T1R10C7-DHP	GTGTCGTAGACACGAAACACCAGAACGAGAGGCTTGCCCTGACGAGTGTCGTAGACAC
Edg-T1R14C7-DHP	GTGTCGTAGACACGAACGAGGGTAGCAACGCGAAAGACAGCATCGGTGTCGTAGACAC
Edg-T1R18C7-DHP	GTGTCGTAGACACGGGATTTTGCTAAACAAATGAATTTTCTGTATGTGTCGTAGACAC
Edg-T2R02C7-DHP	GTGTCGTAGACACGAGAGGGTTGATATAAGCGGATAAGTGCCGTCGTGTCGTAGACAC
Edg-T2R06C7-DHP	GTGTCGTAGACACGCAGGTCAGACGATTGTTGACAGGAGGTTGAGGTGTCGTAGACAC
Edg-T2R10C7-DHP	GTGTCGTAGACACGCGCCAAAGACAAAAGTTCATATGGTTTACCAGTGTCGTAGACAC
Edg-T2R14C7-DHP	GTGTCGTAGACACTTTTTTGTTTAACGTCTCCAAATAAGAAACGAGTGTCGTAGACAC
Edg-T2R18C7-DHP	GTGTCGTAGACACTAAACCAAGTACCGCATTCCAAGAACGGGTATGTGTCGTAGACAC
Edg-T3R02C7-DHP	GTGTCGTAGACACAGTAGGGCTTAATTGAAAAGCCAACGCTCAACGTGTCGTAGACAC
Edg-T3R06C7-DHP	GTGTCGTAGACACAGTCAATAGTGAATTTTTAAGACGCTGAGAAGGTGTCGTAGACAC
Edg-T3R10C7-DHP	GTGTCGTAGACACCAATATAATCCTGATTGATGATGGCAATTCATGTGTCGTAGACAC
Edg-T3R14C7-DHP	GTGTCGTAGACACACATCGCCATTAAAAAAACTGATAGCCCTAAAGTGTCGTAGACAC
Edg-T3R18C7-DHP	GTGTCGTAGACACTTGATTAGTAATAACATTGTAGCAATACTTCTGTGTCGTAGACAC
Edg-T4R02C7-DHP	GTGTCGTAGACACCGGGCGCTAGGGCGCTAAGAAAGCGAAAGGAGGTGTCGTAGACAC
Edg-T4R06C7-DHP	GTGTCGTAGACACATCCTGTTTGATGGTGGCCCCAGCAGGCGAAAGTGTCGTAGACAC
Edg-T4R10C7-DHP	GTGTCGTAGACACGTAACGCCAGGGTTTTAAGGCGATTAAGTTGGGTGTCGTAGACAC
Edg-T4R14C7-DHP	GTGTCGTAGACACTTTAAATTGTAAACGTATTGTATAAGCAAATAGTGTCGTAGACAC
Edg-T4R18C7-DHP	GTGTCGTAGACACAAATTTTTAGAACCCTTTCAACGCAAGGATAAGTGTCGTAGACAC

Table S4: Active edge staples.

Name	Sequence
Edg-2nt-Rec-T1C7R02	TCCCAATTCTGCGAACCCATATAACAGTTG
Edg-2nt-Rec-T1C7R04	ATTGCTCCTTTTGATATTAGAGAGTACCTT
Edg-2nt-Rec-T1C7R06	CCATAAATCAAAAATCCAGAAAACGAGAAT
Edg-2nt-Rec-T1C7R08	CGAGGCATAGTAAGAGACGCCAAAAGGAAT
Edg-2nt-Rec-T1C7R12	CTGATAAATTGTGTCGAGATTTGTATCATC
Edg-2nt-Rec-T1C7R14	GAACGAGGGTAGCAACGCGAAAGACAGCAT
Edg-2nt-Rec-T1C7R16	GGTTTATCAGCTTGCTAGCCTTTAATTGTA
Edg-2nt-Rec-T1C7R18	GGGATTTTGCTAAACAAATGAATTTTCTGT
Edg-2nt-Rec-T2C7R02	GAGAGGGTTGATATAAGCGGATAAGTGCCG
Edg-2nt-Rec-T2C7R04	GTATAAACAGTTAATGTTGAGTAACAGTGC
Edg-2nt-Rec-T2C7R06	GCAGGTCAGACGATTGTTGACAGGAGGTTG
Edg-2nt-Rec-T2C7R08	TAGCGCGTTTTCATCGCTTTAGCGTCAGAC
Edg-2nt-Rec-T2C7R12	CCGAAGCCCTTTTTAAAGCAATAGCTATCT
Edg-2nt-Rec-T2C7R14	TTTTTTGTTTAACGTCTCCAAATAAGAAAC
Edg-2nt-Rec-T2C7R16	AACCTCCCGACTTGCGGCGAGGCGTTTTAG
Edg-2nt-Rec-T2C7R18	TAAACCAAGTACCGCATTCCAAGAACGGGT
Edg-2nt-Rec-T3C7R02	AGTAGGGCTTAATTGAAAAGCCAACGCTCA
Edg-2nt-Rec-T3C7R04	AATGGTTTGAAATACCCTTCTGACCTAAAT
Edg-2nt-Rec-T3C7R06	AGTCAATAGTGAATTTTTAAGACGCTGAGA
Edg-2nt-Rec-T3C7R08	TGAGCAAAAGAAGATGATTCATTTCAATTA
Edg-2nt-Rec-T3C7R12	GTTATCTAAAATATCTAAAGGAATTGAGGA
Edg-2nt-Rec-T3C7R14	ACATCGCCATTAAAAAAACTGATAGCCCTA
Edg-2nt-Rec-T3C7R16	TCGTCTGAAATGGATTACATTTTGACGCTC
Edg-2nt-Rec-T3C7R18	TTGATTAGTAATAACATTGTAGCAATACTT
Edg-2nt-Rec-T4C7R02	CGGGCGCTAGGGCGCTAAGAAAGCGAAAGG
Edg-2nt-Rec-T4C7R04	ATCACCCAAATCAAGTGCCCACTACGTGAA
Edg-2nt-Rec-T4C7R06	ATCCTGTTTGATGGTGGCCCCAGCAGGCGA
Edg-2nt-Rec-T4C7R08	GCTCACTGCCCGCTTTACATTAATTGCGTT
Edg-2nt-Rec-T4C7R12	CGTTGGTGTAGATGGGGTAATGGGATAGGT
Edg-2nt-Rec-T4C7R14	TTTAAATTGTAAACGTATTGTATAAGCAAA
Edg-2nt-Rec-T4C7R16	GCCGGAGAGGGTAGCTTAGCTGATAAATTA
Edg-2nt-Rec-T4C7R18	AAATTTTTAGAACCCTTTCAACGCAAGGAT
Edg-2nt-G4-T1C7R02	AATCCCAATTCTGCGAACCCATATAACAGTTGAT
Edg-2nt-G4-T1C7R04	ATATTGCTCCTTTTGATATTAGAGAGTACCTTTA
Edg-2nt-G4-T1C7R06	TACCATAAATCAAAAATCCAGAAAACGAGAATGA
Edg-2nt-G4-T1C7R08	CACGAGGCATAGTAAGAGACGCCAAAAGGAATTA
Edg-2nt-G4-T1C7R12	GCCTGATAAATTGTGTCGAGATTTGTATCATCGC
Edg-2nt-G4-T1C7R14	AAGAACGAGGGTAGCAACGCGAAAGACAGCATCG
Edg-2nt-G4-T1C7R16	CCGGTTTATCAGCTTGCTAGCCTTTAATTGTATC
Edg-2nt-G4-T1C7R18	AGGGGATTTTGCTAAACAAATGAATTTTCTGTAT
Edg-2nt-G2-T3C7R02	ATAGTAGGGCTTAATTGAAAAGCCAACGCTCAAC
Edg-2nt-G2-T3C7R04	CGAATGGTTTGAAATACCCTTCTGACCTAAATTT

Name	Sequence
Edg-2nt-G2-T3C7R06	GAAGTCAATAGTGAATTTTTAAGACGCTGAGAAG
Edg-2nt-G2-T3C7R08	TATGAGCAAAAGAAGATGATTCATTTCAATTACC
Edg-2nt-G2-T3C7R12	TGGTTATCTAAAATATCTAAAGGAATTGAGGAAG
Edg-2nt-G2-T3C7R14	AGACATCGCCATTAAAAAAACTGATAGCCCTAAA
Edg-2nt-G2-T3C7R16	CCTCGTCTGAAATGGATTACATTTTGACGCTCAA
Edg-2nt-G2-T3C7R18	TCTTGATTAGTAATAACATTGTAGCAATACTTCT
Edg-2nt-G1-T2C7R02	ATGAGAGGGTTGATATAAGCGGATAAGTGCCGTC
Edg-2nt-G1-T2C7R04	TCGTATAAACAGTTAATGTTGAGTAACAGTGCCC
Edg-2nt-G1-T2C7R06	CGGCAGGTCAGACGATTGTTGACAGGAGGTTGAG
Edg-2nt-G1-T2C7R08	GCTAGCGCGTTTTCATCGCTTTAGCGTCAGACTG
Edg-2nt-G1-T2C7R12	TACCGAAGCCCTTTTTAAAGCAATAGCTATCTTA
Edg-2nt-G1-T2C7R14	GATTTTTTGTTTAACGTCTCCAAATAAGAAACGA
Edg-2nt-G1-T2C7R16	TAAACCTCCCGACTTGCGGCGAGGCGTTTTAGCG
Edg-2nt-G1-T2C7R18	ATTAAACCAAGTACCGCATTCCAAGAACGGGTAT
Edg-2nt-G3-T4C7R02	CTCGGGCGCTAGGGCGCTAAGAAAGCGAAAGGAG
Edg-2nt-G3-T4C7R04	AAATCACCCAAATCAAGTGCCCACTACGTGAACC
Edg-2nt-G3-T4C7R06	AAATCCTGTTTGATGGTGGCCCCAGCAGGCGAAA
Edg-2nt-G3-T4C7R08	AGGCTCACTGCCCGCTTTACATTAATTGCGTTGC
Edg-2nt-G3-T4C7R12	CCCGTTGGTGTAGATGGGGTAATGGGATAGGTCA
Edg-2nt-G3-T4C7R14	AGTTTAAATTGTAAACGTATTGTATAAGCAAATA
Edg-2nt-G3-T4C7R16	TTGCCGGAGAGGGTAGCTTAGCTGATAAATTAAT
Edg-2nt-G3-T4C7R18	ACAAATTTTTAGAACCCTTTCAACGCAAGGATAA

Table S5: Interior staples with extensions.

Name	Sequence
ssEx-Reg-T1R01C6	TCATTTGCTAATAGTAGTAGCATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R03C5	CAACTAAAGTACGGTGGGATGGCTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R03C6	TTTCATTGAGTAGATTTAGTTTCTATATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R04C5	TAGAGCTTCAGACCGGAAGCAAACCTATTATATTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R05C6	GTCAGGAAGAGGTCATTTTTGCTCTGGAAGTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R06C3	TTAAGAGGGTCCAATACTGCGGATAGCGAGTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R06C5	GTCAGAAGATTGAATCCCCCTCAACCTCGTTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R07C4	AAATATTCCAAAGCGGATTGCATCGAGCTTCATTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R07C6	AACAGTTAGGTCTTTACCCTGATCCAACAGTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R08C3	AGGCTTTTCAGGTAGAAAGATTCAATTACCTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R08C5	ACCAGACGGAATACCACATTCAACGAGATGGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R09C2	CATTATTAGCAAAAGAAGTTTTGCTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R09C4	AGATTTAGACGATAAAAACCAAAAATCGTCATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R09C6	ATACATACAACACTATCATAACATGCTTTATTTTTTTTTTTTTTTTT
ssEx-Reg-T1R10C1	AGTCAGGACATAGGCTGGCTGACCTTTGAAAGTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R10C3	TTATGCGATTGACAAGAACCGGAGGTCAATTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R10C5	TTAATTTCCAACGTAACAAAGCTGTCCATGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R11C2	GAGTAATCTTTTAAGAACTGGCTCCGGAACAATTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R11C4	ACCCAAATAACTTTAATCATTGTGATCAGTTGTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R11C6	GTGAATATAGTAAATTGGGCTTTAATGCAGTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R12C3	CATAAGGGACACTAAAACACTCACATTAAATTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R12C5	ACTTAGCCATTATACCAAGCGCGAGAGGACTATTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R13C2	AAAAGAATAACCGAACTGACCAACTTCATCAATTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R13C4	CCCCAGCGGGAACGAGGCGCAGACTATTCATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R13C6	ACAACGGAAATCCGCGACCTGCCTCATTCATTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R14C3	CGGGTAAAATTCGGTCGCTGAGGAATGACATTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R14C5	AAGACTTTGGCCGCTTTTGCGGGATTAAACAGTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R15C4	GAGTTAAATTCATGAGGAAGTTTCTCTTTGACTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R15C6	CTCAGCAGGCTACAGAGGCTTTAACAAAGTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R16C5	CTTGATACTGAAAATCTCCAAAAAAGCGGAGTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R17C4	TTTCACGTCGATAGTTGCGCCGACCTTGCAGGTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R17C6	CAAAAGGTTCGAGGTGAATTTCTCGTCACCTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R19C6	GTTAGTAACTTTCAACAGTTTCAAAGGCTCTTTTTTTTTTTTTTTTTT
ssEx-Reg-T1R21C5	CCATGTACCGTAACACTGTAGCATTCCACAGATTCCAGACTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R01C6	ACCCTCATTCAGGGATAGCAAGCCTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R03C5	TTAGGATTAGCGGGGTGGAACCTATTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R03C6	GTACCAGGTATAGCCCGGAATAGAACCGCCTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R04C5	TTATTCTGACTGGTAATAAGTTTTAACAAATATTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R05C6	CAGTGCCCCCCCTGCCTATTTCTTTGCTCATTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R06C3	GTCTCTGACACCCTCAGAGCCACATCAAAATTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R06C5	AATCCTCAACCAGAACCACCACCAGCCCCCTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R07C4	GAGCCGCCTTAAAGCCAGAATGGAGATGATACTTTTTTTTTTTTTTTTTT

Name	Sequence
ssEx-Reg-T2R07C6	GCCAGCAGCCTTGATATTCACAAACGGGGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R08C3	TCACCGGAAACGTCACCAATGAATTATTCATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R08C5	ATTAGCGTCCGTAATCAGTAGCGAATTGAGGGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R09C2	AGGCCGGAACCAGAGCCACCACCGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R09C4	TAGCAGCATTGCCATCTTTTCATACACCCTCATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R09C6	AGTTTGCGCATTTTCGGTCATAGAGCCGCCTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R10C1	GCCATTTGCAAACGTAGAAAATACCTGGCATGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R10C3	TTAAAGGTACATATAAAAGAAACAAACGCATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R10C5	AGGGAAGGATAAGTTTATTTTGTCAGCCGAACTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R11C2	AGGTGGCAGAATTATCACCGTCACCATTAGCATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R11C4	ACCACGGATAAATATTGACGGAAAACCATCGATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R11C6	TAGAAAAGGCGACATTCAACCGCAGAATCATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R12C3	ATAATAACTCAGAGAGATAACCCGAAGCGCTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R12C5	AAAGTTACGCCCAATAATAAGAGCAGCCTTTATTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R13C2	CGCTAATAGGAATACCCAAAAGAAATACATAATTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R13C4	TGAGTTAACAGAAGGAAACCGAGGGCAAAGACTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R13C6	ATGAAATGAAAAGTAAGCAGATACAATCAATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R14C3	ATTAGACGGAGCGTCTTTCCAGAGCTACAATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R14C5	CAGAGAGAACAAAATAAACAGCCATTAAATCATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R15C4	TGCCAGTTATAACATAAAAACAGGACAAGAATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R15C6	ATCCCAAAAAAATGAAAATAGCAAGAAACATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R16C5	AGATTAGTATATAGAAGGCTTATCCAAGCCGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R17C4	CAAATCAGTGCTATTTTGCACCCAGCCTAATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R17C6	TAAGAACGGAGGTTTTGAAGCCTATTATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R19C6	CTTATCACTCATCGAGAACAAGCGGTATTCTTTTTTTTTTTTTTTTTT
ssEx-Reg-T2R21C5	AGCTAATGCAGAACGCGAGAAAAATAATATCCTGTCTTTCTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R01C6	AGAATATCAGACGACGACAATAAATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R03C5	TCATATGCGTTATACAAAGGCGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R03C6	CCAGTATGAATCGCCATATTTAGTAATAAGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R04C5	AAATAAGAACTTTTTCAAATATATCTGAGAGATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R05C6	ATTTCATGACCGTGTGATAAATAATTCTTATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R06C3	TATATAACGTAAATCGTCGCTATATTTGAATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R06C5	CTACCTTTAGAATCCTTGAAAACAAGAAAACATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R07C4	TTTCCCTTTTAACCTCCGGCTTAGCAAAGAACTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R07C6	GCTTAGAATCAAAATCATAGGTTTTAGTTATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R08C3	TTACCTTTACAATAACGGATTCGCAAAATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R08C5	AAATTAATACCAAGTTACAAAATCCTGAATAATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R09C2	CGGGAGAATTTAATGGAAACAGTATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R09C4	CTTTGAATTACATTTAACAATTTCTAATTAATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R09C6	GCGAATTATGAAACAAACATCATAGCGATATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R10C1	GTAGATTTGTTATTAATTTTAAAAAACAATTCTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R10C3	ATTTGCACCATTTTGCGGAACAAATTTGAGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R10C5	TGGAAGGGAGCGGAATTATCATCAACTAATAGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R11C2	AACATTATGTAAAACAGAAATAAATTTTACATTTTTTTTTTTTTTTTTTT

Name	Sequence
ssEx-Reg-T3R11C4	CCAGAAGGTTAGAACCTACCATATCCTGATTGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R11C6	ATTATCAGTTTGGATTATACTTGCGCAGAGTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R12C3	GATTTAGATTGCTGAACCTCAAAGTATTAATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R12C5	ATTAGAGCAATATCTGGTCAGTTGCAGCAGAATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R13C2	GCATCACCAGTATTAGACTTTACAGTTTGAGTTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R13C4	CCTCAATCCGTCAATAGATAATACAGAAACCATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R13C6	ACAGTTGTTAGGAGCACTAACATATTCCTGTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R14C3	CACCGCCTGAAAGCGTAAGAATACATTCTGTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R14C5	GATAAAACTTTTTGAATGGCTATTTTCACCAGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R15C4	AGACAATAAGAGGTGAGGCGGTCATATCAAACTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R15C6	ATGCGCGTACCGAACGAACCACGCAAATCATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R16C5	TCACACGATGCAACAGGAAAAACGGAAGAACTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R17C4	CCAGCCATCCAGTAATAAAAGGGACGTGGCACTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R17C6	AATACCTATTTACATTGGCAGAAGTCTTTATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R19C6	TTAACCGTCACTTGCCTGAGTACTCATGGATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T3R21C5	CTAAACAGGAGGCCGATAATCCTGAGAAGTGTCACGCAAATTTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R01C6	GCGCGTACTTTCCTCGTTAGAATCTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R03C5	AAAGCCGGCGAACGTGTGCCGTAATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R03C6	GGAAGGGGGCAAGTGTAGCGGTGCTACAGGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R04C5	AGCACTAAAAAGGGCGAAAAACCGAAATCCCTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R05C6	GGCGATGTTTTTGGGGTCGAGGGCGAGAAATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R06C3	TGAGTGTTCAGCTGATTGCCCTTGCGCGGGTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R06C5	TATAAATCGAGAGTTGCAGCAAGCGTCGTGCCTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R07C4	GGCCCTGAAAAAGAATAGCCCGAGCGTGGACTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R07C6	CTGGTTTGTTCCGAAATCGGCATCTATCAGTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R08C3	GAGAGGCGACAACATACGAGCCGCTGCAGGTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R08C5	AGCTGCATAGCCTGGGGTGCCTAAGTAAAACGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R09C2	AATTCCACGTTTGCGTATTGGGCGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R09C4	AAGTGTAATAATGAATCGGCCAACCACCGCCTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R09C6	CTAACTCCCAGTCGGGAAACCTGGTCCACGTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R10C1	GAATTCGTGCCATTCGCCATTCAGTTCCGGCATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R10C3	TCGACTCTGAAGGGCGATCGGTGCGGCCTCTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R10C5	ACGGCCAGTACGCCAGCTGGCGAACATCTGCCTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R11C2	ACTGTTGGAGAGGATCCCCGGGTACCGCTCACTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R11C4	TTCGCTATTGCCAAGCTTGCATGCGAAGCATATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R11C6	GTGCTGCCCCAGTCACGACGTTTGAGTGAGTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R12C3	AGGAAGATCATTAAATGTGAGCGTTTTTAATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R12C5	AGTTTGAGATTCTCCGTGGGAACAATTCGCATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R13C2	TTCATCAACGCACTCCAGCCAGCTGCTGCGCATTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R13C4	CCCGTCGGGGGACGACGACAGTATCGGGCCTCTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R13C6	ATTGACCCGCATCGTAACCGTGAGGGGGATTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R14C3	CCAATAGGAAACTAGCATGTCAAGGAGCAATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R14C5	TAAATTTTTGATAATCAGAAAAGCACAAAGGCTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R15C4	ACCCCGGTTGTTAAATCAGCTCATAGTAACAATTTTTTTTTTTTTTTTTTT

Name	Sequence
ssEx-Reg-T4R15C6	CAGGAAGTAATATTTTGTTAAAAACGGCGGTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R16C5	TATCAGGTAAATCACCATCAATATCAATGCCTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R17C4	AGACAGTCCATTGCCTGAGAGTCTTCATATGTTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R17C6	ACCGTTCATTTTTGAGAGATCTCCCAAAAATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R19C6	CCTTTATCATATATTTTAAATGGATATTCATTTTTTTTTTTTTTTTTTTT
ssEx-Reg-T4R21C5	AATCATACAGGCAAGGCAGAGCATAAAGCTAAGGGAGAAGTTTTTTTTTTTTTTTTTTTT

Table S6: Negation strands.

Name	Sequence
Neg-T1R00C7	TTTTCAGCTCGCGCCCTAGAATTGATGCCACC
Neg-T1R02C7	ATCAACTGTTATATGGGTTCGCAGAATTGGGA
Neg-T1R04C7	TAAAGGTACTCTCTAATATCAAAAGGAGCAAT
Neg-T1R06C7	TCATTCTCGTTTTCTGGATTTTTGATTTATGG
Neg-T1R08C7	TAATTCCTTTTGGCGTCTCTTACTATGCCTCG
Neg-T1R10C7	TCGTCAGGGCAAGCCTCTCGTTCTGGTGTTTC
Neg-T1R12C7	GCGATGATACAAATCTCGACACAATTTATCAG
Neg-T1R14C7	CGATGCTGTCTTTCGCGTTGCTACCCTCGTTC
Neg-T1R16C7	GATACAATTAAAGGCTAGCAAGCTGATAAACC
Neg-T1R18C7	ATACAGAAAATTCATTTGTTTAGCAAAATCCC
Neg-T1R20C7	ACTGGTGACGAAACTCAGGCGTTGTAGTTTGT
Neg-T2R00C7	CTGAGGGTGGCGGTTCAATGAGGGTGGTGGCT
Neg-T2R02C7	GACGGCACTTATCCGCTTATATCAACCCTCTC
Neg-T2R04C7	GGGCACTGTTACTCAACATTAACTGTTTATAC
Neg-T2R06C7	CTCAACCTCCTGTCAACAATCGTCTGACCTGC
Neg-T2R08C7	CAGTCTGACGCTAAAGCGATGAAAACGCGCTA
Neg-T2R10C7	TGGTAAACCATATGAACTTTTGTCTTTGGCGC
Neg-T2R12C7	TAAGATAGCTATTGCTTTAAAAAGGGCTTCGG
Neg-T2R14C7	TCGTTTCTTATTTGGAGACGTTAAACAAAAAA
Neg-T2R16C7	CGCTAAAACGCCTCGCCGCAAGTCGGGAGGTT
Neg-T2R18C7	ATACCCGTTCTTGGAATGCGGTACTTGGTTTA
Neg-T2R20C7	ATTGTTGATAAACAGGTGTTCAGGACTTATCT
Neg-T3R00C7	CTTTTGTCGGTACTTTGACAGAATTACTTTAC
Neg-T3R02C7	GTTGAGCGTTGGCTTTTCAATTAAGCCCTACT
Neg-T3R04C7	AAATTTAGGTCAGAAGGGTATTTCAAACCATT
Neg-T3R06C7	CTTCTCAGCGTCTTAAAAATTCACTATTGACT
Neg-T3R08C7	GGTAATTGAAATGAATCATCTTCTTTTGCTCA
Neg-T3R10C7	ATGAATTGCCATCATCAATCAGGATTATATTG
Neg-T3R12C7	CTTCCTCAATTCCTTTAGATATTTTAGATAAC
Neg-T3R14C7	TTTAGGGCTATCAGTTTTTTTAATGGCGATGT
Neg-T3R16C7	TTGAGCGTCAAAATGTAATCCATTTCAGACGA
Neg-T3R18C7	AGAAGTATTGCTACAATGTTATTACTAATCAA
Neg-T3R20C7	GTCTAAAATCCCTTTACTGGCGTACCGTTCCT
Neg-T4R00C7	GTCAAAGCAACCATAGCACGTTATACGTGCTC
Neg-T4R02C7	CTCCTTTCGCTTTCTTAGCGCCCTAGCGCCCG
Neg-T4R04C7	GGTTCACGTAGTGGGCACTTGATTTGGGTGAT
Neg-T4R06C7	TTTCGCCTGCTGGGGCCACCATCAAACAGGAT
Neg-T4R08C7	GCAACGCAATTAATGTAAAGCGGGCAGTGAGC
Neg-T4R10C7	CCAACTTAATCGCCTTAAAACCCTGGCGTTAC
Neg-T4R12C7	TGACCTATCCCATTACCCCATCTACACCAACG
Neg-T4R14C7	TATTTGCTTATACAATACGTTTACAATTTAAA
Neg-T4R16C7	ATTAATTTATCAGCTAAGCTACCCTCTCCGGC

Name	Sequence
Neg-T4R18C7	TTATCCTTGCGTTGAAAGGGTTCTAAAAATTT
Neg-T4R20C7	ATTTTGCTAATTCTTTGAGGCTTTATTGCTTA

References

[1] Nadrian C Seeman. Nucleic acid junctions and lattices. *Journal of Theoretical Biology*, 99(2):237–247, 1982.

[2] Kyle Lund, Anthony J Manzo, Nadine Dabby, Nicole Michelotti, Alexander Johnson-Buck, Jeanette Nangreave, Steven Taylor, Renjun Pei, Milan N Stojanovic, Nils G Walter, et al. Molecular robots guided by prescriptive landscapes. *Nature*, 465(7295):206–210, 2010.

[3] Hongzhou Gu, Jie Chao, Shou-Jun Xiao, and Nadrian C Seeman. A proximity-based programmable DNA nanoscale assembly line. *Nature*, 465(7295):202–205, 2010.

[4] Shelley FJ Wickham, Jonathan Bath, Yousuke Katsuda, Masayuki Endo, Kumi Hidaka, Hiroshi Sugiyama, and Andrew J Turberfield. A DNA-based molecular motor that can navigate a network of tracks. *Nature Nanotechnology*, 7(3):169–173, 2012.

[5] Anupama J. Thubagere, Wei Li, Robert F. Johnson, Zibo Chen, Shayan Doroudi, Yae Lim Lee, Gregory Izatt, Sarah Wittman, Niranjan Srinivas, Damien Woods, Erik Winfree, and Lulu Qian. A cargo-sorting DNA robot. *Science*, 357(6356):eaan6558, 2017.

[6] Gourab Chatterjee, Neil Dalchau, Richard A Muscat, Andrew Phillips, and Georg Seelig. A spatially localized architecture for fast and modular DNA computing. *Nature Nanotechnology*, 12(9):920–927, 2017.

[7] Hareem T Maune, Si-ping Han, Robert D Barish, Marc Bockrath, William A Goddard III, Paul WK Rothemund, and Erik Winfree. Self-assembly of carbon nanotubes into two-dimensional geometries using DNA origami templates. *Nature Nanotechnology*, 5(1):61–66, 2010.

[8] Jakob Bach Knudsen, Lei Liu, Anne Louise Bank Kodal, Mikael Madsen, Qiang Li, Jie Song, Johannes B Woehrstein, Shelley FJ Wickham, Maximilian T Strauss, Florian Schueder, et al. Routing of individual polymers in designed patterns. *Nature Nanotechnology*, 10(10):892–898, 2015.

[9] Ashwin Gopinath, Evan Miyazono, Andrei Faraon, and Paul WK Rothemund. Engineering and mapping nanocavity emission via precision placement of DNA origami. *Nature*, 535(7612):401405, 2016.

[10] Paul WK Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006.

[11] Sungwook Woo and Paul WK Rothemund. Programmable molecular recognition based on the geometry of DNA nanostructures. *Nature Chemistry*, 3(8):620–627, 2011.

[12] Arivazhagan Rajendran, Masayuki Endo, Yousuke Katsuda, Kumi Hidaka, and Hiroshi Sugiyama. Programmed two-dimensional self-assembly of multiple DNA origami jigsaw pieces. *Acs Nano*, 5(1):665–671, 2011.

[13] Zhao Zhao, Yan Liu, and Hao Yan. Organizing DNA origami tiles into larger structures using preformed scaffold frames. *Nano Letters*, 11(7):2997–3002, 2011.

[14] Alexandria N Marchi, Ishtiaq Saaem, Briana N Vogen, Stanley Brown, and Thomas H LaBean. Toward larger DNA origami. *Nano Letters*, 14(10):5740–5747, 2014.

[15] Andre V Pinheiro, Dongran Han, William M Shih, and Hao Yan. Challenges and opportunities for structural DNA nanotechnology. *Nature Nanotechnology*, 6(12):763–772, 2011.

[16] David Doty. Theory of algorithmic self-assembly. *Communications of the ACM*, 55(12):78–88, 2012.

[17] Sung Ha Park, Constantin Pistol, Sang Jung Ahn, John H Reif, Alvin R Lebeck, Chris Dwyer, and Thomas H LaBean. Finite-size, fully addressable DNA tile lattices formed by hierarchical assembly procedures. *Angewandte Chemie*, 118(5):749–753, 2006.

[18] Grigory Tikhomirov, Philip Petersen, and Lulu Qian. Programmable disorder in random DNA tilings. *Nature Nanotechnology*, 12(3):251–259, 2017.

[19] Benoit B Mandelbrot. *The fractal geometry of nature*. W. H. Freeman and Company, New York, 1982.

[20] Jeanette Nangreave, Hao Yan, and Yan Liu. Studies of thermal stability of multivalent DNA hybridization in a nanostructured system. *Biophysical Journal*, 97(2):563–571, 2009.

[21] Sherri Rinker, Yonggang Ke, Yan Liu, Rahul Chhabra, and Hao Yan. Self-assembled DNA nanostructures for distance-dependent multivalent ligand-protein binding. *Nature Nanotechnology*, 3(7):418–422, 2008.

[22] Robert Schreiber, Jaekwon Do, Eva-Maria Roller, Tao Zhang, Verena J Schüller, Philipp C Nickels, Jochen Feldmann, and Tim Liedl. Hierarchical assembly of metal nanoparticles, quantum dots and organic dyes using DNA origami scaffolds. *Nature Nanotechnology*, 9(1):74–78, 2014.

[23] Carlos Ernesto Castro, Fabian Kilchherr, Do-Nyun Kim, Enrique Lin Shiao, Tobias Wauer, Philipp Wortmann, Mark Bathe, and Hendrik Dietz. A primer to scaffolded DNA origami. *Nature Methods*, 8(3):221–229, 2011.

[24] Integrated DNA Technologies. Chemical synthesis and purification of oligonucleotides. `https://www.idtdna.com/pages/docs/technical-reports/chemical-synthesis-of-oligonucleotides.pdf`, 2005.

[25] Philip Petersen. FracTile Compiler. `http://qianlab.caltech.edu/FracTileCompiler`, 2017. Currently supported web browsers: Firefox and Chrome.

[26] John Zenk, Chanon Tuntivate, and Rebecca Schulman. Kinetics and thermodynamics of Watson–Crick base pairing driven DNA origami dimerization. *Journal of the American Chemical Society*, 138(10):3346–3354, 2016.

[27] Chenxiang Lin, Steven D Perrault, Minseok Kwak, Franziska Graf, and William M Shih. Purification of DNA-origami nanostructures by rate-zonal centrifugation. *Nucleic Acids Research*, 41(2):e40–e40, 2012.

[28] Younan Xia and George M Whitesides. Soft lithography. *Annual Review of Materials Science*, 28(1):153–184, 1998.

[29] Harish Chandran, Nikhil Gopalkrishnan, Andrew Phillips, and John Reif. Localized hybridization circuits. *Lecture Notes in Computer Science*, 6937:64–83, 2011.

[30] Lulu Qian and Erik Winfree. Parallel and scalable computation and spatial dynamics with DNA-based chemical reaction networks on a surface. *Lecture Notes in Computer Science*, 8727:114–131, 2014.

[31] T Delbrück and CA Mead. Analog VLSI phototransduction by continuous-time, adaptive, logarithmic photoreceptor circuits. *Vision Chips: Implementing vision algorithms with analog VLSI circuits*, pages 139–161, 1995.

[32] Bryan Wei, Mingjie Dai, and Peng Yin. Complex shapes self-assembled from single-stranded DNA tiles. *Nature*, 485(7400):623–626, 2012.

[33] Wen Wang, Tong Lin, Suoyu Zhang, Tanxi Bai, Yongli Mi, and Bryan Wei. Self-assembly of fully addressable DNA nanostructures from double crossover tiles. *Nucleic Acids Research*, 44(16):7989–7996, 2016.

[34] Kyle Lund, Yan Liu, Stuart Lindsay, and Hao Yan. Self-assembling a molecular pegboard. *Journal of the American Chemical Society*, 127(50):17606–17607, 2005.

[35] Gagan Aggarwal, Qi Cheng, Michael H Goldwasser, Ming-Yang Kao, Pablo Moisset De Espanes, and Robert T Schweller. Complexities for generalized models of self-assembly. *SIAM Journal on Computing*, 34(6):1493–1515, 2005.

[36] Erik D Demaine, Martin L Demaine, Sándor P Fekete, Mashhood Ishaque, Eynat Rafalin, Robert T Schweller, and Diane L Souvaine. Staged self-assembly: nanomanufacture of arbitrary shapes with O(1) glues. *Natural Computing*, 7(3):347–370, 2008.

[37] Erik D Demaine, Sarah Eisenstat, Mashhood Ishaque, and Andrew Winslow. One-dimensional staged self-assembly. *Natural Computing*, 12(2):247–258, 2013.

[38] Philip Petersen. Yield Calculator. `http://qianlab.caltech.edu/YieldCalculator`, 2016.

[39] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[40] Klaus F Wagenbauer, Floris AS Engelhardt, Evi K Stahl, Vera K Hechtl, Pierre Stömmer, Fabian Seebacher, Letizia Meregalli, Philip Ketterer, Thomas Gerling, and Hendrik Dietz. How we make DNA origami. *ChemBioChem*, 2017.

[41] Shawn M Douglas, Adam H Marblestone, Surat Teerapittayanon, Alejandro Vazquez, George M Church, and William M Shih. Rapid prototyping of 3D DNA-origami shapes with caDNAno. *Nucleic Acids Research*, 37(15):5001–5006, 2009.