nature chemistry

SUPPLEMENTARY INFORMATION

DOI: 10.1038/NCHEM.1070

Programmable molecular recognition based on the geometry of DNA nanostructures

Sungwook Woo[1*] and Paul W. K. Rothemund[1,2,3*]

Departments of [1]Bioengineering, [2]Computer Science, and [3]Computation & Neural Systems, California Institute of Technology, 1200 E. California Blvd. Pasadena, CA 91125, USA

* Correspondence: woo@dna.caltech.edu and pwkr@dna.caltech.edu

Table of Contents

Supplementary Note S1: Materials and methods 3
S1.1. Sample preparation 3
S1.2. Atomic force microscopy 3
Supplementary Note S2: Design details 4
S2.1. Design of binary codes for stacking bonds 4
S2.1.1. Design criteria for binary sequences 4
S2.1.2. Example binary sequences 5
S2.1.3. Why use 7 active patches with a mismatch constraint of 4? 6
S2.1.4. Error rates for binary sequences investigated in this study 8
S2.2. Design of shape codes for stacking bonds. 9
S2.2.1. Design criteria for shape sequences 9
S2.2.2. Full list of candidate shape sequences for the (4,3,2) system 10
S2.2.3. Orthogonality graph for the (4,3,2) candidate shape sequences 11
S2.2.4. Size of shape sequence spaces with other parameters 12
S2.3. Finding codes: searching for large orthogonal sets of sequences. 13
S2.4. Design of the origami structures 15
S2.5. Edge structure 16
S2.6. Quencher strands 18
S2.7. Warnings 19
S2.7.1. Length and width of a patch in shape design 19
S2.7.2. Potential interference from the remainder staples 21
S2.7.3. Possible collisions between edge staples 22
Supplementary Note S3: Thermodynamic measurements 23
S3.1. First energy model: assuming loop-loop interactions are neutral 25
S3.2. Second energy model: fitting with non-zero loop-loop interactions 28
Supplementary Note S4: Additional AFM Data 30
S4.1. Stacking of rectangles 30
S4.2. 5-origami chains with orthogonal binary-coded bonds 32
S4.3. Origami dimers and chains with orthogonal shape-coded bonds 33
References 34
Supplementary Note S5: Sequence lists and diagrams 35

Supplementary Note S1:Materials and methods

S1.1. Sample preparation

Individual origami structures that were not destined to be mixed with other structures were prepared by a protocol similar to that presented in earlier work. Single-stranded M13mp18 DNA (scaffold strand) was purchased from New England Biolabs (Catalog # N4040S) and staple strands were obtained unpurified from Integrated DNA Technologies in water at 150 μM each. Scaffold strand and staple strands for each design were mixed together to target concentrations of ~2 nM and ~75 nM, respectively, in 1× Tris-Acetate-EDTA (TAE) buffer with 12.5 mM magnesium acetate (TAE/Mg^{2+}). The mixtures were kept at 90°C for 5 min and annealed from 90°C to 20°C with a constant rate of -1°C/min.

To create origami chains with multiple bonds based on binary sequences (as shown in Fig. 2c of the main text), constituent origami were first annealed separately from 90°C to 20°C. Next, corresponding quencher strand mixtures for each origami (those that matched the edge staples used, see Section S2.6) were added (at 10× the edge staple concentration) to each origami mixture. Each of the solutions was kept at room temperature for 1 hr to ensure complete hybridization, and then they were mixed together, heated to 50°C, kept for 12 hr at 50°C, and then cooled to 20°C at a rate of -5°C/hr.

For the origami chain (**A-B-C-D**) and dimers (**A-B**, **B-C**, **C-D**) with shape complementarity (as shown in Fig. 3c,d), each origami mixture (scaffold + corresponding staples) was annealed separately from 90°C to 50°C (with a rate of -1°C/min), mixed together at 50°C, and kept at 50°C for 12 hr, then cooled to 20°C at a rate of -5°C/hr. The mixing operation was performed inside a temperature-controlled chamber (Coy Laboratory Products Inc.), to maintain the temperature at 50°C while the samples were transferred between test tubes.

S1.2. Atomic force microscopy

Samples for AFM imaging were prepared by depositing 5 μl of the origami solution with 20 μl of TAE/Mg^{2+} buffer onto freshly-cleaved mica (Ted Pella). In most cases, clean buffer solution was deposited first and the origami solution was added on top of it. (For concentrated samples we felt this procedure minimized spatial variation in the density of origami on the mica.) In cases wherein we were concerned that this procedure might distort data (i.e. for thermodynamic data, section S3) we pre-diluted the origami solution by 5-fold, and then deposit 25 μl onto mica. AFM images were taken under TAE/Mg^{2+} buffer in Tapping Mode with a Nanoscope III Multimode AFM (Veeco Metrology Group, now Bruker AXS). Typically, we used silicon nitride cantilevers with 2 nm radius silicon tips as AFM probes (the "short, fat" A cantilever on SNL probes from Veeco, now Brucker AFM Probes).

Supplementary Note S2: Design details

S2.1. Design of binary codes for stacking bonds

S2.1.1. Design criteria for binary sequences

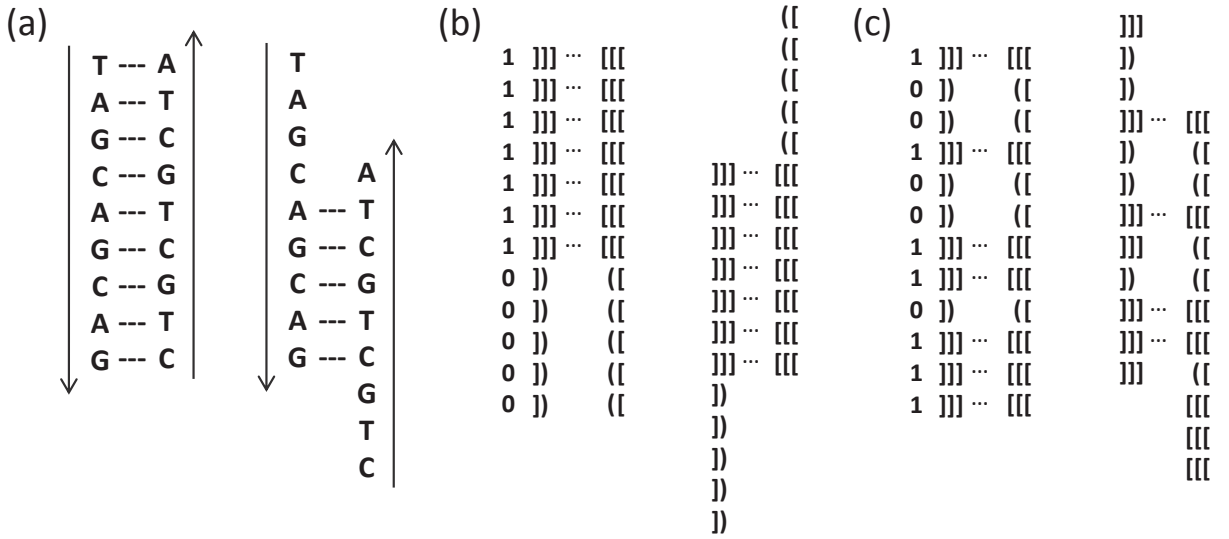


Figure S1. Issues in binary sequence design. (a) DNA sequence design must deal with the problem of undesired partial complementarity. A desired bond is at left, an undesired partial bond at right. Binary sequence design is analogous, as explained in the text. (b) A simple binary sequence that allows a full-strength, self-complementary incorrect bond; this sequence, while nonpalindromic, is not uniquely-orienting. (c) A binary sequence whose strongest partial bonds are only of strength 4; an example is shown at right.

The basic design criteria for binary sequences can be understood by analogy to criteria used for DNA sequence design (Fig. S1a). Consider a DNA strand with the sequence 5'-TAGCAGCAG-3'; it is fully complementary to (and hence would bind most strongly with) a strand bearing the sequence 5'-CTGCTGCTA-3' (Fig. S1a-left). However, the two strands also have a partially complementary subsequence of length five, and could bind (albeit more weakly) via this partial interaction (Fig. S1a-right). In general, when DNA sequences are designed, they are designed to *minimize* such undesired interactions—with themselves, with their complements, and with any other strands that will be present in solution at the same time. Simple algorithms for designing sequences use discrete criteria based on the maximal number of base pairs that occur in any partially complementary species. For example, an algorithm might be designed to find sequences that minimize this number. Partial bonds having the same number of base pairs but different sequences are not equal in strength, and so more sophisticated algorithms minimize the sequence-dependent binding energy of undesired interactions. Still more sophisticated algorithms use such binding energies to maximize the probability that the desired interactions form by considering the thermodynamic partition function.

Here, because we do not yet have a complete energy model for stacking bonds, we take a simple approach based on counting (and minimizing) the number of active patches involved in the strongest partial bonds. For example, consider an origami with the binary sequence '111111100000'; with its complementary partner it would form a stacking bond of strength 7 (Fig. S1b-left), but when rotated it can

also form a self-complementary, undesired interaction of strength 7 (Fig. S1b-right). In contrast, the sequence '100100110111' binds its complement (Fig. S1c, left) with a strength-7 bond, but the strongest possible partial bond that it can form has only strength 4 (Fig. S1c, right).

As for DNA, we are interested in minimizing such undesired interactions. For binary sequences of length l and number of active patches p, we wrote a program that enumerates sequences which have a maximum strength i for incorrect partial bonds (the mismatch constraint) with themselves and with their complements. Conceptually, the program compares each sequence to itself (and its complementary sequence) at all possible alignments, by "sliding" the sequences relative to each other; the number of matches for each alignment is simply counted and the sequence is discarded if the number of matches exceeds i for any alignment.

The set of sequences enumerated for a given (p,i) constituted a *candidate set* from which we later attempted to construct maximal orthogonal subsets for use in making origami chains (see Section S2.3). It turns out that for p=7, and l=12 or l=16 (the length of the sequence applicable to the regular and tall rectangles used in our study, respectively), the candidate sets are empty for mismatch constraints i<3. That is, *however* we design a binary sequence with 7 active patches (for l=12 or l=16), such a sequence will have an undesired partial bond (with itself or its complement) involving at least 4 active patches. More generally, for all p there exists at least i for which candidate sequences can be found. As i is made larger, the size of the candidate set increases; this holds true for the size of the maximum orthogonal subsets as well. Thus there is a tradeoff between the mismatch constraint i (our heuristic surrogate for the experimental specificity) and the number of distinct sequences available as bond types. This can be seen in Table S1 of section 2.1.3. Note that the minimum possible i is 2, since any pair of active patches in a binary sequence belongs to a partially self-complementary subsequence with at least two active patches.

S2.1.2. Example binary sequences

For the 12-patch system with 7 active patches, a total of 98 different binary sequences were found to satisfy the mismatch constraint $i=4$; for the 16-patch system with (p,i) = (7,4), a total of 4614 sequences were obtained. We give some examples from each candidate set below. Full candidate sets are available upon request (woo@dna.caltech.edu); alternatively, one can generate the sets easily using the program code (attached as a separate Supplementary file).

12-patch system (10 examples shown, out of a total of 98):

0 1 0 0 1 0 1 1 0 1 1 1
0 1 0 0 1 1 0 0 1 1 1 1
0 1 0 1 0 1 0 0 1 1 1 1
1 1 0 0 0 1 0 1 1 0 1 1
1 1 0 0 0 1 1 1 1 0 1 0
1 1 0 0 1 0 0 0 1 1 1 1
1 1 0 0 0 0 1 1 1 1 0 1
1 1 0 0 0 1 1 0 1 1 0 1
1 1 0 0 1 0 0 0 1 1 1 1
1 1 0 0 1 0 1 1 0 1 0 1

16-patch system (10 examples shown, out of a total of 4614):

0 0 0 0 1 0 0 0 1 1 1 0 1 1 0 1
0 0 0 0 1 0 0 1 0 0 1 0 1 1 1 1
0 0 0 0 1 0 0 1 0 1 0 0 1 1 1 1
0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1
0 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0
1 0 0 1 1 0 0 0 1 1 0 1 0 0 0 1
1 0 0 1 1 0 0 0 1 1 0 1 0 1 0 0
1 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1
1 0 0 1 1 0 0 1 0 0 0 0 1 1 0 1
1 0 0 1 1 0 0 1 0 0 0 0 1 1 1 0

S2.1.3. Why use 7 active patches with a mismatch constraint of 4?

Our goal was to create the largest binary code that we could, with the largest number of distinct bond types, subject to the constraint that the bonds would have high specificity (that is, the rate of incorrect partial bond formation would be low.) We wrote a program to enumerate candidate sets for two different sequence lengths, a variety of different numbers of active patches, and mismatch constraints. We further used randomly seeded greedy search (see Section 2.3) to find the largest orthogonal subsets that we could for each candidate set. Table S1 summarizes our results. We found that choosing the parameters (p, i) to be (7,4), (8,5), or (9,6) with l=16 yielded orthogonal subsets with more than ten sequences, while still maintaining a reasonably large energetic difference between full-strength correct bonds and partial incorrect bonds.

		Total # of available patches = 12 (regular rectangle)					Total # of available patches = 16 (tall rectangle)					
		# of active patches, *p*					# of active patches, *p*					
		5	**6**	**7**	**8**	**9**	**5**	**6**	**7**	**8**	**9**	**10**
# patches *i* in partial bonds	**2**	4 (1)					320 (3)	0				
	3	214 (2)	0	0			1866 (27)	236 (6)	0	0		
	4		420 (15)	**98 (2)**	0			6520 (68)	4614 (12)	462 (2)	0	
	5			384	8 (1)	0			8322	2730 (13)	36 (2)	0
	6				328	14 (1)				6400	5870 (15)	496 (3)

Table S1. Size of candidate sets and the largest orthogonal subsets found as a function of sequence length, number of active patches, and mismatch constraint. Numbers in parentheses indicate the size of the largest orthogonal subset found (See Note S2.3). Shaded areas indicate the systems with 3-patch difference between full-strength and partial bonds (corresponding to an equilibrium ratio of $e^{-3\Delta Gp/kT}$, where ΔG_p is the free energy of a bound active patch and is equal to 2 times ΔG_{st}, the free energy of a stacked helix). Blank spaces indicate that the search process was not performed for the corresponding parameters (because the result would either be meaningless [$i \geq p$] or not useful, since either no candidate sequences would be found, or i was too close to p for bonds to be specific).

If one assumes the simplest model of binding energy for binary sequences (namely that the binding energy is linear in the number of active patches involved in a bond) then the energy of a full correct bond is $p \cdot \Delta G_p$ (where ΔG_p is the free energy of a bound active patch and is equal to 2 times ΔG_{st}, the free energy of a stacked helix), the energy of the strongest partial bond is $i \cdot \Delta G_p$ and the equilibrium ratio between the full correct bond and the strongest partial bond is: $e^{-(p-i)\Delta Gp/kT}$. A full treatment of the total error rate associated with a particular binary sequence would take into account not only the energy of the strongest partial bond, but also the number (multiplicity) of the different partial bonds having this energy, as well as the energies and multiplicities of all weaker partial bonds; such a treatment would calculate the full partition function for the system. Instead, here we simply assume that the multiplicity of the strongest partial bonds for different sequences is roughly the same. Given these assumptions then the equilibrium error rates for sequences from the three different systems—(7,4), (8,5), and (9,6)—should be the same. However, because the fraction of correct bonds versus unbound origami should increase with increasing p it would make sense to choose sequences from the system with full bonds of higher strength, *i.e.* a (9,6) system.

To check our assumptions about error rates, we measured the error rates for sample sequences from the (7,4), (8,5), and (9,6) candidate sets for length 12 sequences. Experiments analogous to those shown in Fig. 2a in the main text were conducted; Fig. S2 shows representative AFM images for each sequence tested. 'L'-shaped labels on the origami made scoring correct head-to-tail bonds (L-L) easy; incorrect bonds included both bonds with rotated orientation and bonds with head-to-tail orientation that were misaligned. Surprisingly, the (7,4) sequence gave the best error rate, with the highest fraction of correct bonds out of total bonds—96.8% (N=344, for the sequence occurring in the bottom of Fig. S2a). The other systems performed considerably less well, with the (8,5) sequence having 77.7% correct bonds (N=358, Fig. S2b) and the (9,6) sequence having 52.7% correct bonds (N=277, Fig. S2c).

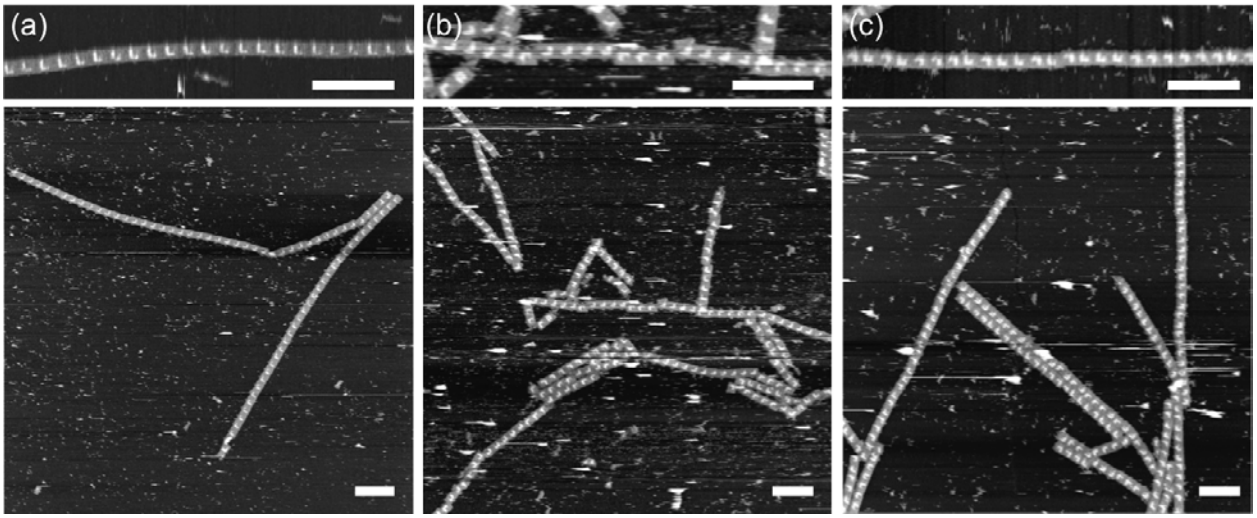


Figure S2. Comparison of sequence performance as a function of the number of active patches. A binary sequence and its complement are placed on opposite edges of an origami such that it should form long chains; each origami carries the label 'L'. Full-strength correct bonds are measured by counting the bonds with head-to-tail orientation (L-L). Partial bonds of all types are also counted; they usually involve origami bound in the rotated orientation. (a) One (7,4) system, '100100110111' (top) and another '010111100011' (bottom). Error rate data were taken for the bottom system; the top system is included to show a high-res image of a system of qualitatively similar error rate. (b) An (8,5) system, '100101011111'. (c) A (9,6) system, '110111001111'. Scale bars: 500 nm.

This surprising trend might not be a general phenomenon, since just a few sequences were examined, or it could be the case that our assumption about the multiplicity of partial bonds is wrong and that, for example, the (9,6) sequence observed just had many more partial bonds than the other systems, all of them having the strongest possible strength (i). However, given our thermodynamic experiments (Section S3) another possibility suggests itself: that ΔG_p is not constant as the number of active patches p increases and thus the total stacking bond energy is not linear in the number of active patches. In particular, if ΔG_p decreases with increasing p then our results make sense. Then the energy difference between a full correct bond and the strongest partial bond in the (9,6) system is not as large as the analogous energy difference for the (8,5) system, which in turn is not as large as that for the (7,4) system. Such a sublinearity in stacking bond energy might be explained by steric interference or electrostatic repulsion between active patches, or it might be explained by a nonlinear bending energy term that increases as the use of more active patches results in them being more spread out and requires them to overcome a large-scale deformation of the origami. Because of the trends we observed in our experiments using sequences with constant p and i (Section S3, ΔG_p seems to decrease as active patches are more spread-out along the

origami edge) we suspect the latter hypothesis is a more likely explanation. Clearly performance measurements for many more sequences should be made, but based on these preliminary experiments, we chose to explore (7,4) sequences in the context of a longer, 16-patch system.

S2.1.4. Error rates for binary sequences investigated in this study

For edges of the tall rectangle system (which has 16 total available patches), there are 4614 different binary sequences with (p, i) of (7,4), as shown in Table S1. Within the set of those binary sequences, subsets (codes) can be found for which every pair of sequences from the subset is the mutually orthogonal with the same matching criterion (no partial match between any pair of sequences involves more than 4 active patches). Our computer-aided search generated several codes of size 11 and 12; one code of size 12 and another code of size 11 were chosen for more detailed study. Each binary sequence from these codes was tested in the same way as in Fig. 2a, i.e., the binary sequence and its complement were placed on opposite edges of the tall rectangle, such that the rectangles form bonds in h2t orientation when the bonds are full-strength and correct. For each case, AFM data were analyzed and bond orientations were measured to obtain the ratio between the correct (h2t) bond orientation and the total number of bonds. The error rate measured for each binary sequence and an example of annotated AFM data from which such error rates are derived is shown in Table S2 below.

set	sequence #	sequence	% correct bonds	N
set1	1	0 0 0 0 0 1 0 1 1 1 1 0 0 0 1 1	90.07%	423
	2	1 1 1 0 0 0 0 0 1 1 0 0 0 1 0 1	84.39%	538
	3	0 0 1 0 0 0 0 0 1 0 1 1 0 1 1 1	79.73%	301
	4	1 1 0 0 0 0 0 0 1 0 1 1 1 0 0 1	86.50%	941
	5	0 1 1 0 1 0 1 0 0 0 1 0 0 1 0 1	87.96%	191
	6	0 1 0 0 0 1 0 1 0 0 0 1 1 0 1 1	97.55%	245
	7	1 0 0 0 1 0 0 0 1 1 0 1 0 0 1 1	82.25%	524
	8	0 1 1 1 1 0 0 1 0 0 0 0 1 0 1 0	96.53%	346
	9	1 0 0 1 0 0 1 0 0 1 1 0 0 1 0 1	72.54%	142
	10	1 1 1 0 1 0 1 0 0 1 0 0 0 0 1 0	87.64%	259
	11	1 0 0 0 0 1 0 1 1 0 0 1 0 1 0 1	83.68%	337
	12	1 1 0 1 0 0 0 1 0 0 0 0 1 1 0 1	83.36%	559
set2	1	0 0 0 1 0 1 1 1 1 0 0 0 1 1 0 0	94.39%	659
	2	0 0 1 0 0 1 0 0 1 1 0 1 1 1 0 0	78.07%	456
	3	1 0 1 0 0 1 0 0 0 0 1 1 0 1 1 0	95.24%	210
	4	1 0 0 0 1 0 0 0 1 0 1 0 1 1 0 1	74.29%	210
	5	0 1 1 1 0 0 0 0 1 0 1 1 0 0 1 0	95.41%	827
	6	1 1 0 0 0 0 0 0 1 0 0 1 1 1 1 0	95.54%	112
	7	1 1 0 1 0 1 0 0 1 0 0 0 0 0 1 1	94.20%	448
	8	1 0 0 1 1 0 0 1 0 1 0 0 0 1 0 1	91.67%	168
	9	1 1 0 1 0 0 0 0 0 0 1 1 0 1 0 1	81.68%	475
	10	0 1 1 0 0 1 0 1 1 0 0 0 1 0 0 1	78.33%	300
	11	1 0 0 1 1 0 1 0 0 0 1 0 0 0 1 1	93.81%	113

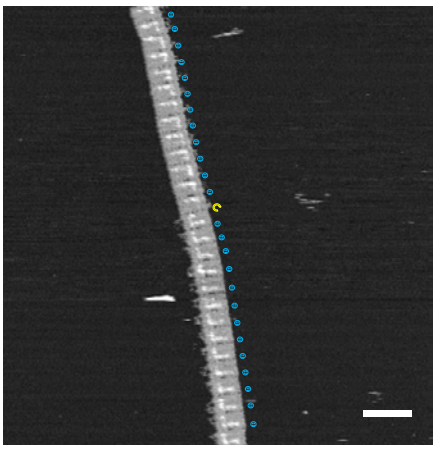


Table S2. Error rates for stacking bonds made using binary sequences from two different sets (sequences within each set are orthogonal) and an example of AFM image analysis. Each bond was labeled and counted based on its orientation. The blue circles over the AFM image indicate bonds with the correct head-to-tail orientation and the yellow circular arrow indicates a single bond with an incorrect rotated orientation. The percentage of correct bonds (with h2t orientation) out of the total number of bonds analyzed (N) was recorded for each binary sequence. The AFM image is for binary sequence #3 in set2. Scale bar: 200 nm.

S2.2. Design of shape codes for stacking bonds.

S2.2.1. Design criteria for shape sequences

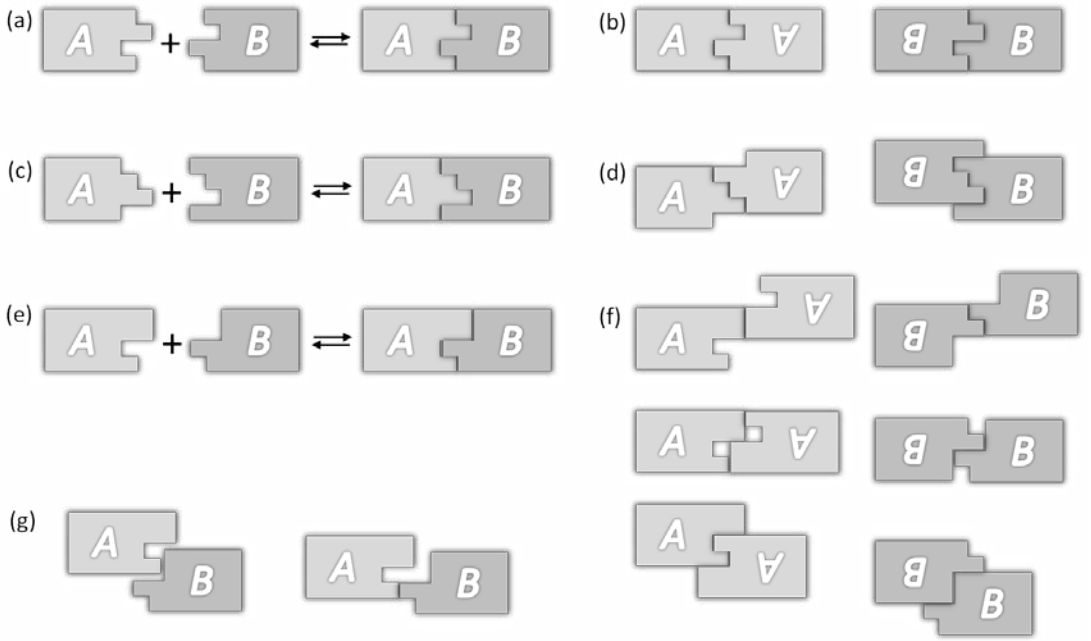


Figure S3. Examples of shape sequences and their partial bonds. (a) A 4-patch shape sequence could form a 4-patch bond between distinct origami, but because it is fully self-complementary (b) it also allows full-strength undesired bonds. (c) Another shape sequence and its complement could form a 4-patch bond, but (d) it also allows partially self-complementary 3-patch bonds (3/4 the strength of a full bond). (e) A shape sequence and its complement that we used between the **A** and **B** origami. (f) Examples of partially self-complementary (homogeneous) bonds of strength 2 for the sequences in (e). (g) Examples of partial bonds of strength 1 between the two distinct origami (heterogeneous) for the sequences in (e).

As in the design of binary sequences, the goal of minimizing undesired bonds dictates design criteria for shape sequences. As before, fully self-complementary sequences (Fig. S3ab) or partially self-complementary sequences (Fig. 3cd) must be avoided (unless a homodimer of origami is desired.) Computer enumeration of candidate sets of sequences for shape codes is essentially similar to that for binary codes, with two important differences. First, unlike the case for DNA or binary sequences, *not all shape sequences encode physically distinct bonds*; we discuss this in the next section. Second, unlike the case for DNA or binary sequences, *the program must make an extra check for the self-complementarity of the shape sequence's complement*. For DNA or binary codes, to evaluate whether a sequence should be a candidate sequence, it is sufficient to check that sequences' self-complementarity, and to check for any partial complementarity that it might have with its complement. This is because for a DNA or binary sequence, any self-complementary subsequence that occurs in the sequence implies the existence of a corresponding self-complementary subsequence in the sequence's complement, and vice versa. This is not the case for shape sequences: the shape sequence '100001' has a strongest self-complementary partial bond of strength 2, but its complement '011110' has a strongest self-complementary partial bond of strength 4. We note also that the minimum possible mismatch constraint i for shape sequences is 2 patches (as it is for binary codes), but this limit holds for a different reason in the case of shape sequences than for binary sequences. The reason is that, for an arbitrary shape sequence, the first two or last two

patches both form self-complementary subsequences that can bind to themselves without steric hindrance from any of the other patches (if the two origami carrying them are in a rotated orientation). The top two and bottom two examples in Fig. S3f demonstrate this phenomenon.

Early on in the project we thought we might achieve a large number of specific bonds through the use of a long (l=6 or l=9) shape sequences. Fitting these high complexity sequences into the relatively small area of an origami necessitated using patches that were just 2 helices wide. These proved too flexible to prevent bent-patch bonds (see S.2.7.1) so we decided to use 4-helix wide patches to implement shape codes. This restricted the length of the shape sequences we could use to just 4 patches; similarly we restricted ourselves to just three depths to avoid long, flexible patches. Fortunately, even with these restrictions, the candidate set for the mismatch constrain i=2 had 16 elements, which we discuss next.

S2.2.2. Full list of candidate shape sequences for the (4,3,2) system

Given the number of patches (p) and number of depths (d), our program searches the entire sequence space and examines the possible partial bonds for each shape sequence with itself, each sequence with its complement, and each complement with itself to see if they exceed the mismatch constraint (i). For (p,d,i) = (4,3,2) the program generated a candidate set of 16 unique shape sequences, listed at left below. Note first, that if a sequence appears, its complement does not appear: our desire is to make a set of candidate sequences for distinct bond types and a sequence and its complement are *equivalent* with respect to the physical bond type that they encode. Similarly, sequences that are related by a simple *shift* in depth, such as '0010' and '1121' (Fig. S4ab), encode the same bond type and are thus equivalent. Only a single sequence from an equivalence class is included in the candidate set. Recall that due to stacking polarity, a shape sequence (e.g. Fig. S4a) and its reverse (Fig. S4e) are inequivalent, unless they are palindromic. (Shape sequences and their reverses are expected to be similar for some properties, e.g. flexibility.)

List of the 16 shape sequences:

1 '0010' (= '1121' = '1011' = '2122')
2 '0020'
3 '0021'
4 '0100'
5 '0102'
6 '0110' (B-C bond)
7 '0200'
8 '0201'
9 '0211'
10 '0220'
11 '0221'
12 '1020' (C-D bond)
13 '1120'
14 '1200' (= '2201', A-B bond)
15 '1220'
16 '2010'

(Equivalences are not exhaustively listed.)

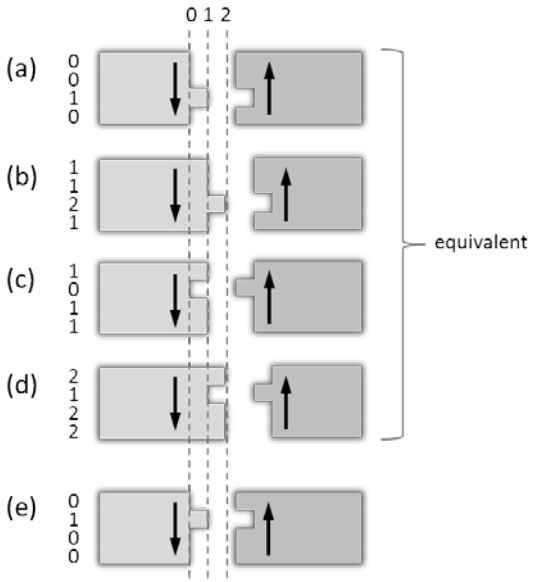


Figure S4. Equivalences between shape sequences. (a), (b), (c), and (d) are all equivalent with respect to the bond type they encode. Vertical arrows denote stacking polarity. (a) and (b) are related by a simple shift in depth, (b) and (c) are complementary, (c) and (d) are related by a depth shift, and (a) and (d) are complementary. (e) is, however, distinct from the others because it is the reverse of (a), and has opposite stacking polarity.

S2.2.3. Orthogonality graph for the (4,3,2) candidate shape sequences

Using our computer program, one can check the orthogonality between any two sequences in the set of 16 shapes listed in the previous section. By applying the same mismatch constraint for the strongest partial bonds (i = 2) between different sequences, the orthogonality relations can be determined for each pair of sequences (a total of 120 combinations). Fig. S5a shows the full orthogonality graph for all 16 shape sequences in the candidate set. Line segments between numbered circles indicate that the two shape sequences corresponding to the numbers are orthogonal to each other. (For the identity of each shape sequence, see list in the previous section.)

Sets of mutually orthogonal sequences correspond to *complete subgraphs* or *cliques* of the orthogonality graph. That is, a set of vertices for which *every* pair of vertices is connected by a line segment corresponds to a set of mutually orthogonal sequences. For example, one complete subgraph is the red triangle in Fig. S5b which corresponds to an orthogonal subset with three sequences, {6,12,14}. An exhaustive search confirmed that the size of the largest orthogonal subsets for the given system is 4 — for example the subset indicated by the red subgraph in Fig. S5c: {3,6,7,16}. We attempted to construct origami chains based on a subset of size 4. Unfortunately, the set we chose included sequence 5, which, along with its reverse sequence 16, turned out to be susceptible to the formation of bent-patch bonds. In the interest of saving time and money, and because we had demonstrated that sequence 6 worked well, we ended up choosing the 3-sequence subset {6,12,14} to explore as a shape code. The four sequence orthogonal subset {1,11,12,14} looks promising, but was not explored.

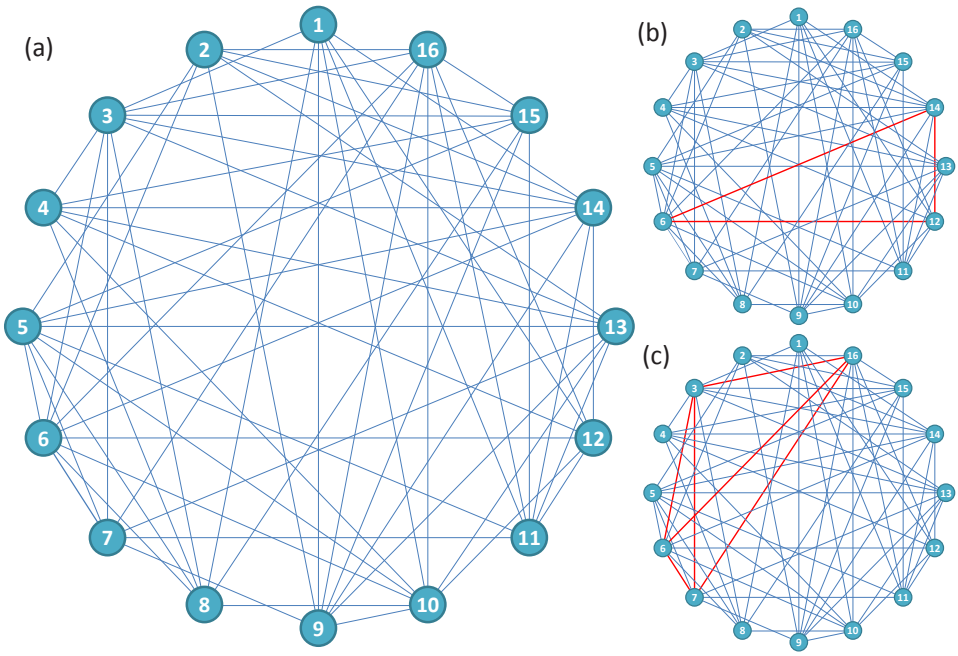


Figure S5. Orthogonality graph for the candidate set of 16 shape sequences. (a) The full graph. Each line connecting two numbered circles indicates that the two shape sequences are orthogonal. (b) Red triangle highlights an orthogonal set of size three with sequences 6, 12, and 14. (c) Red subgraph highlights an orthogonal set of size four with sequences 3, 6, 7, and 16.

S2.2.4. Size of shape sequence spaces with other parameters

In addition to the 4-patch system with d=3 and i=2, we explored other shape sequence spaces with different parameters, a couple of which were tested experimentally (some results are shown in Note S2.6.1). In the table below, we summarize the sizes of shape sequence spaces with various parameters (different numbers of patches, different numbers of depths, different numbers of patches allowed in incorrect bonds). In the right-hand column we report the size of the largest orthogonal subset discovered over the course of multiple random searches, as described in Note S2.3.

For parameters not included in this table, one can easily obtain the shape sequence space – not only the size but the entire list of candidate shape sequences – using the program code provided as a separate Supplementary file.

# patches (p)	# depths (d)	# patches allowed in incorrect bonds (i)	% patches in incorrect bonds	Size of shape sequence space	Size of largest orthogonal subsets found
4	2	2	50%	3	1
	3	2	50%	16	4
	4	2	50%	61	8
6	3	2	33%	5	1
		3	50%	49	7
	4	2	33%	145	6
		3	50%	423	22
9	3	2	22%	2	1
		3	33%	14	5
	4	2	22%	233	6
		3	33%	1113	25
	5	2	22%	6510	15
		3	33%	24791	60

Table S3. Sizes of shape sequence spaces and orthogonal subsets for different systems. The purple-shaded parameters are those for the shape sequences most explored by this study.

S2.3. Finding codes: searching for large orthogonal sets of sequences.

So far we have largely discussed, for both binary sequences and shape sequences, the symmetry and mismatch constraints that are required of a sequence for it to be useful as an individual stacking bond in isolation — constraints such that, with high probability, the sequence will bind its complement by a full correct bond rather than binding to itself or binding its complement by a partial bond. Given a set of parameters including the length of the sequence, number of active patches, the number of depths (if appropriate), and a mismatch constraint, we have shown that it is straightforward to enumerate all sequences that individually satisfy the constraints. Our two most studied examples are the 4614 binary sequences for (p,i) = (7,4) and the 16 shape sequences for (p,d,i) = (4,3,2). In order for such candidate sequences to be used in a multiple-bond system, one must find a subset that is orthogonal — that is, all pairs of sequences must satisfy the mismatch constraint. A diagram of the orthogonality relation between all 16 sequences in the (4,3,2) shape sequence candidate set, and example orthogonal subsets are discussed and diagrammed in Section S2.2.3.

An important goal is to find the *maximal* orthogonal subset of a candidate set, to find a code of sequences that can support the largest diversity of stacking bond types. For candidate sets containing relatively few sequences, such as the (4,3,2) shape sequences, an exhaustive search through all possible orthogonal subsets is possible. But for bigger candidate sets, the combinatorially large number of subsets makes finding the maximal orthogonal subset by exhaustive search a computationally intractable task.

More specifically, the problem of finding the maximal orthogonal subset is a trivial rephrasing of the well-known *Max Clique* problem in computer science. Max Clique is known to be NP-hard, and here two facts about NP-hardness are relevant: (1) Most computer scientists believe that NP-hard problems can only be solved exactly using an amount of time that is an *exponential* function of the size of the problem — this is what is meant by "computationally intractable". (2) NP-hard problems can be approximated — thus while it might be computationally intractable to find the maximal orthogonal subset, it may be possible to find large subsets, which are close in size to the maximal orthogonal subset, quickly.

There is a large literature on approximating NP-hard problems, but we did not take advantage of such approximation techniques here. Instead, to quickly get large orthogonal subsets that we could use for multiple stacking bonds, we implemented a simple, randomly seeded, greedy search procedure. For many of the smaller candidate sets, the orthogonal sets we obtain are probably maximal; in the case of the (4,3,2) shape code system we verified that this was the case. For larger candidate sets it is highly likely that the orthogonal sets we have obtained are not maximal; it might be possible that there exist 13-sequence orthogonal sets for the (7,4) binary code system. For the moment, the orthogonal sets that we have found are satisfactory since we obtained orthogonal sets whose size roughly matches the maximum number of origami that we can handle easily, or can afford in the lab. However, if the need should arise, e.g. for much larger codes, many relatively fast algorithms for finding large cliques (and hence finding large orthogonal subsets) are available. One example is *Cliquer,* a set of C routines that are available for download from: http://users.tkk.fi/pat/cliquer.html

Our program for discovering large orthogonal subsets constructs them in a "greedy" fashion starting from single sequences from the candidate set. Let the size of the candidate set be N. Each run of our

program constructs *N* different orthogonal subsets, by sequentially using each one of the elements of the candidate set as a seed for a different orthogonal subset. The details of our program are as follows:

(1) The program first picks one candidate sequence as a seed; that candidate sequence becomes the first element of the orthogonal set under construction.

(2) The program randomly picks another sequence from the candidate set and checks its orthogonality with the existing element(s), with respect to the mismatch constraint *i*.

(3) If the newly picked sequence is orthogonal to the existing element(s), it is added to the set; otherwise it is discarded.

(4) The program repeats steps (2) and (3) until all candidate sequences have been tested for the orthogonality with the growing set. After all candidates have been tested, the orthogonal set is output.

(5) The program repeats steps (1) through (4) until all candidate sequences have been used as a seed for an orthogonal set.

Since the construction of an orthogonal set is sensitive to the order of addition of candidate sequences (a different order results in a different set), each run of the program results in *N* potentially distinct orthogonal sets. Typically, we ran the program multiple times; we did not keep track of the number of runs performed. The largest orthogonal subsets found are recorded in Table S1 (for binary sequences) and Table S3 (for shape sequences).

S2.4. Design of the origami structures

Design of the rectangle systems (regular and tall) was performed based on the procedure described in the original DNA origami paper, using MATLAB code. Origami with edge shapes (**A**, **B**, **C**, and **D** origami) were designed using a modified version of the "square lattice" version of caDNAno. caDNAno software was customized to allow (1) the creation of single stranded loopouts in the scaffold strand and (2) the highlighting of the scaffold strand with a user-specified sequence (e.g. 'GC'). Both features were added to facilitate the process of shifting the scaffold strand and aligning 'GC' on the edges. In our modified version of caDNAno, the 'loop tool' which is normally used to generate double-stranded loops (loops involving both the scaffold and staple strands) has been changed to a tool that creates single-stranded loops only in the scaffold strand. The highlighting feature has been integrated into caDNAno's existing 'add sequence' function, with which a user can select the scaffold sequence to use; now a user can specify a sequence to highlight using the same dialog box (Fig. S6c). Color is fixed for each highlighted base: G-Green, C-Cyan (light blue), A-Amber (roughly orange), T-Tomato (red). The length limit for a highlighted sequence is set to be the same as caDNAno's existing length limit for the scaffold sequence (20,000), so a user can highlight the entire scaffold sequence if desired. A few other minor modifications include: (1) skipping the step of waiting for the user to click on the 5' end of the scaffold during an 'add sequence' operation if there is only one 5' scaffold end in the design, and (2) updating the scaffold and staple sequences automatically after creating a new loopout (this automatic update works only after the scaffold sequence has been defined). The modified version caDNAnoSQ_SW is available as a separate Supplementary file (or by request to woo@dna.caltech.edu for latest version). For general instructions and the original version of caDNAno, visit http://www.cadnano.org.

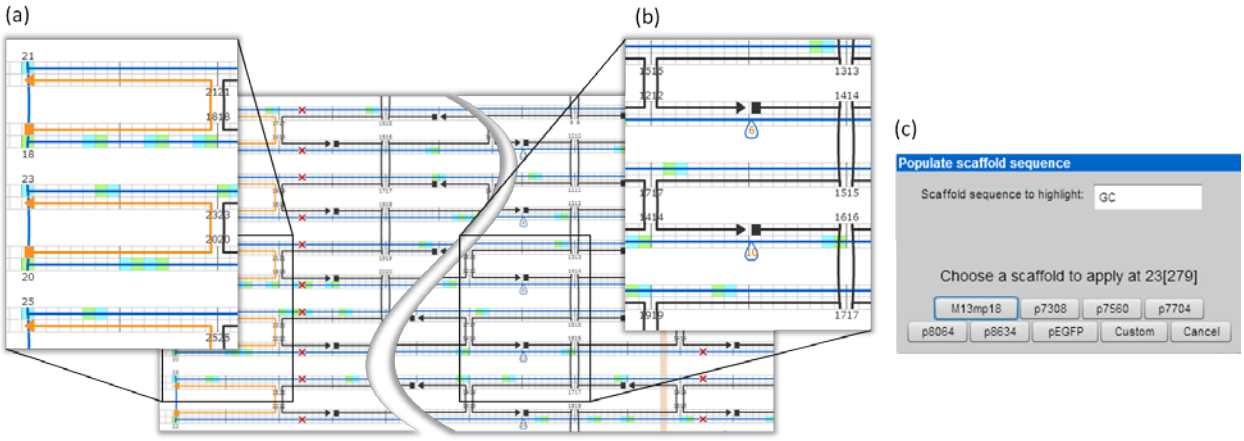


Figure S6. Screenshots of a version of caDNAno modified to allow placement of 'GC' on origami edges. All occurrences of the sequence 'GC' are highlighted along the scaffold strand in the diagram by green and cyan for G and C, respectively. (a) One can count the number of bases from a blunt-end on the edge to the next occurrence of 'GC' and (b) make a single-stranded loopout inside the structure to *shift* the scaffold sequences. In the example design shown, the scaffold strand has been already aligned to have 'GC' along the edge [as shown in the zoom-in at (a)]. (c) Here we have shown highlighting of the sequence 'GC'. One can enter any sequence to highlight at the step where user selects the scaffold strand.

S2.5. Edge structure

Because accurate models (backed by high resolution structural data) of origami edges do not exist, it is difficult to predict the exact structure and stacking configurations of the blunt-ends on the edges of origami. Here we provide gross predictions based on the distance of the blunt-ends from the nearest internal crossovers and the pattern of crossovers along the edge. We predict structures for three different edge models: (1) a *crossover-free edge* (Fig. S7a), (2) a *relaxed edge* with only scaffold crossovers (Fig. S7b), and (3) a *stressed edge* with both scaffold and staple crossovers (Fig. S7c). These predicted structures in turn make predictions about the expected strength and behavior of the stacking bonds.

For all three models we are interested in the helical twist of the base pairs on the blunt-ends at the edge, and for all three models we posit an internal crossover 16 base pairs interior to the edge. Here, we draw bars, separated by the major/minor groove angles, on the face of the blunt ends to indicate the helical twist of the base pair. To derive the orientation of these bars, we begin at the interior crossover and consider the strand that is "edgeward" of the crossover (e.g. the orange 3D strand in Fig. S7a). We model the two base pairs next to the crossover point as staggered up and down with respect to the midpoint of the crossover, having a helical twist angle that is rotated from the midpoint by ½ of the characteristic rotation/bp of B-DNA (~34.6° given 10.4 bp/turn) — that is approximately 17°. (Similar modeling is performed in Ref. 1.) From these "first edgeward base pairs" the base pairs at the blunt end are 15 base-pairs away. Thus the blunt end base pairs have a helical twist angle that is rotated ~519° (15 × 34.6°) relative to the bases of the edgeward strand in the crossover (in a clockwise direction when viewed from the blunt-ends towards the crossover) for a total of ~537° from the crossover midpoint. Given such a model, which is crossover-free at the edge, the base pairs at the blunt-end would be oriented like those depicted in Fig. S7a. We note that while we do not make such a structure in this work, origami with very similar crossover-free edges have been made before (Ref. 10 of the main text) using "tile adapters", and so such structures can be experimentally synthesized.

Now consider a second origami with the same crossover-free edge structure, but with 15 base pairs between the edge and the crossover. When such an origami binds via a stacking bond to the origami described above, then the total number of base pairs between the first internal crossover points of the two origami will be 15+16 = 31, or roughly 3 helical turns. This means that for such crossover-free origami, the blunt-ends on opposite sides of the stacking bond are oriented with a relative twist angle of ~34.6° (as depicted in Fig. S7d). Thus we would expect stacking of blunt ends between crossover free edges to be native B-form stacking, and that it should be relatively strong.

Next consider our second edge structure, the relaxed edges (Fig. S7b), for which scaffold crossovers connect every other pair of helices. This is the edge structure that we use in all our work on stacking bonds, (except for structures pictured in Fig. 1e of the main text and Fig. S14j-o). Because the scaffold crossovers act to pull the base pairs away from the helical twist angle that they would assume in a crossover-free edge, whatever structure forms at relaxed edges cannot be B-form DNA. However, because DNA can tolerate small deviations from B-form twist, we propose that the helices assume an amount of twist strain (roughly 34.6°, which is averaged over the 16 base pairs up to the crossover) and maintain native major/minor groove angles between the bases at the blunt end (as depicted in Fig. S7b).

Given our model for relaxed edges, when two origami with relaxed edges bind via a stacking bond, their blunt ends will not be able to stack via B-form stacking; rather, they should bind with slightly

different relative twist angles that are within approximately ±34.6° of the natural twist angle in B-DNA (Fig. S7e). We call such stacking between relaxed edges *near-B-form stacking,* which we predict would be roughly as strong as B-form stacking. Since relaxed edges have a top-bottom asymmetry that is defined by the major and minor grooves, near-B-form stacking can only occur when two origami bind in either the head-to-tail or rotated orientations. This prediction agrees well with the distribution of observed bond orientations, as discussed in the main text.

Finally, we consider the case of stressed edges. When staple crossovers are placed in opposition to scaffold crossovers along an edge, we propose that the balancing of the stresses they induce results in a near-flattening of the major and minor grooves (Fig. S7c). The resulting decrease in distinction between the major and minor grooves should decrease the distinction between the top and bottom of the origami. Therefore, we would predict that blunt-end stacking between such stressed edges (Fig. S7f) should allow flipped bond orientations; this is indeed what is observed in experiments involving origami with stressed edges, such as those shown in Fig. 1e of the main text and Fig. S14j-o.

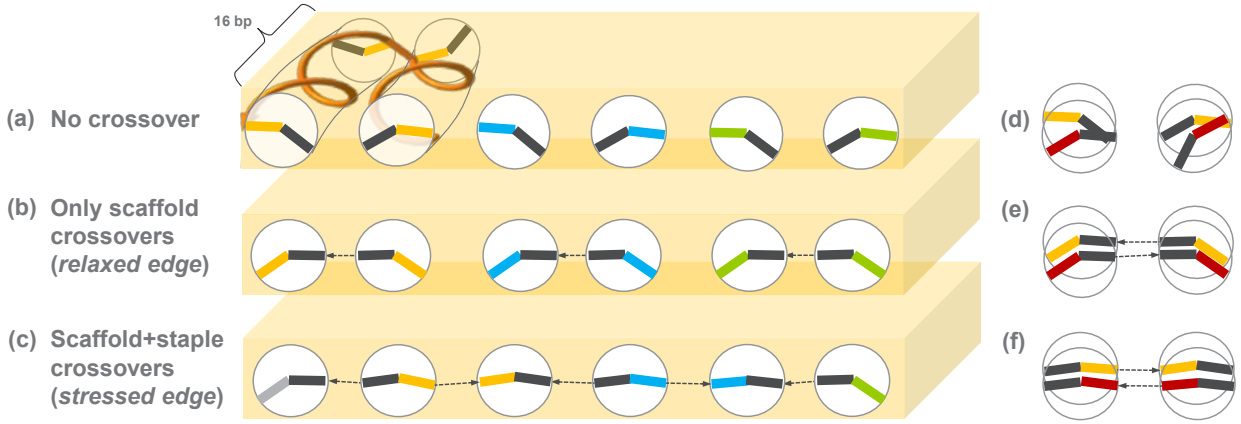


Figure S7. Comparative modeling of three different origami edge structures. (a) *Crossover-free* edges. (b) *Relaxed edges* with only scaffold crossovers. (c) *Stressed edges* with both scaffold and staple crossovers. Each circle indicates a blunt-end. In (a), both the black and colored bars (inside the circles) indicate the helical twist of bases belonging to *tile adapter strands*. In general, tile adapter strands are strands that extend from the edge of an origami to give it a geometry that is not possible using the canonical scaffold/staple geometry. Here, our intent is that the tile adapter strands create a crossover-free edge; we do not show the details and did not use them in our stacking experiments. He we use them as part of a "thought experiment" concerning the geometry of stacking bonds, but we note that tile adapter strands have been used to create origami with a very similar crossover-free edges (Ref. 10 in main text). In (b) and (c), black bars indicate the helical twist of bases from the scaffold strand, and colored bars indicate the helical twist of bases from the staple strands. Bars of the same color indicate the same strand, e.g., the orange staple in (a) runs for 1.5 helical turns in one helix, switches between helices at a 16-bp-deep internal crossover, and runs back for a length of 1.5 helical turns in the adjacent helix, as depicted by the 3D drawing. Black dotted arrows indicate crossovers at the edge. In all three models colored strands are intended to make 16-bp-deep internal crossovers. The models in (a-c), predict that the blunt ends on the edge are either B-form (a), near-B-form (b), or have disrupted base pairs that are incompatible with B-form geometry (c). (d-f) show models of the juxtaposition that occurs when two different origami edges form a stacking bond; these bonding models correspond to the edge structure models in (a), (b), and (c), respectively. Models (d) and (e) make predictions for the relative helical twist between blunt-ends across the bond. Model (f) suggests that the disturbance of the base pairs at the edge of the origami may decrease the distinction between the major and minor grooves enough to create a top-bottom pseudosymmetry. This pseudosymmetry could allow bonding between origami in one of the flipped orientations (not shown).

S2.6. Quencher strands

In a binary coded system in which multiple origami containing different binary sequences on their edges are mixed together, interference can arise between the edge staples used to set the sequences on one origami and the edges representing different sequences on another origami. This occurs because all origami in the binary coded system share the same basic design, and their edges share the same staple binding sites. For example, if one origami bears a sequence on its right edge that has a '1' in a particular position, then the staple that creates that active patch can bind to the *same* location on a different origami for which that location was intended to remain inactive, effectively "flipping" a '0' to a '1'. In the worst case, all of the origami would end up with exactly the same sequences, with the right edge of each origami encoding a sequence that represents the bitwise OR of all the sequences on the right edges of the original origami, and the left edge encoding an analogous bitwise OR of all the left-edge sequences. Prevention of such interference could be achieved by purifying the origami to remove excess staples *before* mixing the origami. But purification steps are usually accompanied by significant loss of the origami themselves and may incompletely remove staples. In particular, simple and fast methods such as spin filtration reduce excess staples only by a factor of 5- to 10- fold; more complete removal requires more stringent methods such as gel purification. Complete removal is important because, as we observed in tests of spin purification, relatively small and sporadic changes to edge sequences can significantly increase error rate. As an alternative approach, we introduced strands complementary to the edge staples, which we term *quenchers*.

Quenchers bind to the excess free edge staples in solution and effectively prevent them from binding to the scaffold strand. Quenchers were designed so that they have complementary sequences to the corresponding edge staples (thus quenchers have sequences derived from scaffold strand subsequences), and extra two thymine bases were added to both the 5'-end and the 3'-end (so that the quencher sequence becomes 5'-TT-staple complement-TT-3'). The thymine addition was done to minimize the potential influence of (1) stacking interference from blunt ends that would be generated if simple complements were used and (2) breathing of the resulting quencher-staple duplex that might allow the edge staple strand to bind to a '0' location anyway, via a branch migration process.

The efficiency of the quenchers at blocking the free edge staples was not explicitly measured. However, the high molar excess of the quenchers used (10× the concentration of edge staples) and the high free energy of binding between the quenchers and the edge staples (on the order of ~40 kcal/mol, calculated using Oligo Calc, http://www.basic.northwestern.edu/biotools/oligocalc.html) predicts the concentration of free edge staples, in the presence of the quenchers, to be extremely small — on the order of 10^{-21} nM. The experimental protocol for using quenchers is described in Supplementary Note S1, and the detailed sequences are given in Supplementary Note S5.

S2.7. Warnings

In case one wants to repeat or adapt some of our experiments, we give warnings that describe some difficulties which we have encountered and suggest some potential problems that we did not discuss in the main text.

S2.7.1. Length and width of a patch in shape design

Besides the 4-patch design in the shape code system, we have tried other designs with higher complexity (6-patch and 9-patch systems) that we expected to give higher specificity. But as the number of patches increased, we had to design each patch with less material, yielding patches with a smaller number of helices. The flexibility of DNA, coupled with the strength of the stacking interactions, caused these "narrow-patch" systems to be more vulnerable to bent-patch bonds. Fig. S8 briefly summarizes the two systems.

In the 6-patch design (Fig. S8a) we introduced physical gaps between each adjacent pair of 2-helix patches, to minimize any effect of the electrostatic repulsion. (It seemed possible that electrostatic repulsion between adjacent patches might decrease binding energy. This hypothesis has yet to be adequately tested.) The introduction of physical gaps made the patches narrower and longer, allowing various kinds of bent-patch bonds (~ 30% of all bonds) as shown in Fig. S8d,e. Because we used 3 helical turns for each depth increment, the length of the longest protruding patch was 9 helical turns, which is about 30% of the persistence length of double-crossover DNA tiles (~30 helical turns, ~100 nm) — the most similar structure to the 2-helix patch structure for which the persistence length is known[2]. We chose to use a (6,4,3) shape code so that, in principle, the maximum-strength partial bond would have three active patches (1/2 of the full strength). To our dismay, 5-patch bent-patch bonds formed; if the bending energy were small, these bonds would have a binding energy comparable to that of full 6-patch correct bond.

In the 9-patch system, (without physical gaps and with much shorter patches), a significant fraction of the bonds (~20 %) were still bent-patch partial bonds (Fig. S8i,j). We had chosen to use a (9,5,2) shape code so we had expected high binding specificity – the strongest expected incorrect bonds would have a binding energy 2/9 of a full correct bond. To decrease the flexibility of patches, we used a single helical turn for each depth increment, so that the longest protruding patch was just 4 helical turns in length. Thus it was to our further dismay that 8-patch bent-patch bonds formed, which were again potentially very close in energy to full-strength bonds. As a point of interest we note that the 6- and 9-patch systems were not twist-corrected, so the chains in Fig. S8g show the characteristic breaking pattern that is similar to that shown in Fig. 1b of the main text. The global twist might be playing a role in encouraging bent-patch bonds in these systems, but we have not done any experiments to test this possibility.

To decrease flexibility, our final "successful" shape code system employed only four of patches that were 4-helices wide and protruded at most 6 helical turns. Many questions remain: How many patches are optimal for this kind of study? How wide (in terms of number of helices) should each patch be? How long can they be? What is the bending energy of the patches under the buffer condition used? We do not yet have answers to these questions, but it is certain that there is a trade-off between the complexity (and hence the potential specificity in ideal case without helix bending) and the bond reliability. Of course, this problem is limited to "soft" systems like DNA, thus might be avoided in a system with sufficient rigidity.

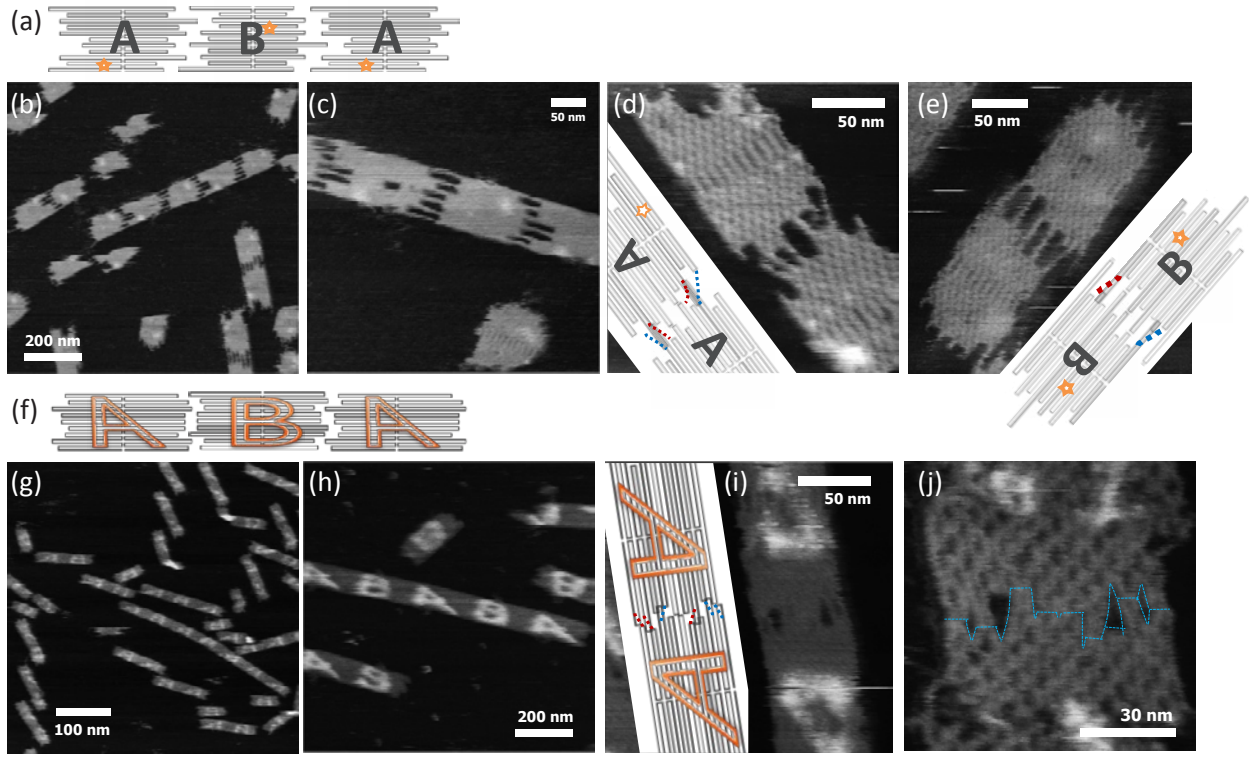


Figure S8. Performance of 6-patch and 9-patch shape-coded systems. (a) Models of the 6-patch system. The edge shapes of the **A** origami and **B** origami were designed such that the origami form continuous alternating **AB** chains. Shape sequences were '132120' for **A**-**B** bonds and '011310' for **B**-**A** bonds. Stars indicate the locations of dumbbell hairpins, which serve as topographic labels for AFM. (b) & (c) Typical AFM images of the system that show full-strength correct bonds. (d) & (e) Typical bent-patch bonds which manage to bind via 5 active patches. The red dotted lines on the models depict bent patches coming from the origami on the top, and the blue dotted lines depict bent patches coming from the origami on the bottom. (f) Models of the 9-patch system. Shape sequences were '034222043' for **A**-**B** bonds and '340224301' for **B**-**A** bonds. (g) & (h) Typical AFM images. Note that the chains in (g) show the characteristic breaking pattern of the twisted origami chains described in Fig. 1 of the main text. (i) & (j) Typical bent-patch bonds with 8-patch bond strength. The red and blue dotted lines in (i) are used in the same way as in (d) or (e). The narrow blue dotted line in (j) roughly follows the blunt-ends and helical sidewalls of the edge structures.

S2.7.2. Potential interference from the remainder staples

Since the length of the scaffold strand is fixed to be 7249 bases, a DNA origami design that uses fewer than 7249 bases will leave a *remainder* in the form of unfolded single-stranded scaffold – in most designs the remainder takes the form of a single loop. To avoid potential interactions of such a single-stranded remainder on one origami with the remainder of another origami, it is usual to add a set of *remainder staple strands* which have the function of hybridizing to the remainder and turning it into an unreactive double-stranded loop.

When multiple origami which do not share the same underlying design are mixed together, e.g. as in our **A-B-C-D** chains with shape complementarity, there is the possibility of interference between the remainder staple strands of one origami and single-stranded regions of the other origami. In general, a subset of staple strands from one origami may bind single-stranded loopouts on other origami via partial complementarity. (Such loopouts are common in our system because they are used to enforce the 'GC' sequence constraint at the blunt ends of helices.) Binding of staple strands to loopouts does not, in general, seem to affect the origami, but in certain cases remainder strands may have complementarity to surrounding scaffold sequence outside of the loopout. In such cases the remainder strands can begin to displace nearby staples. Because the remainder staples are designed to be "continuous" complements to the remainder loop, each successive remainder staple that displaces a regular staple potentially opens up a site for another remainder to bind. Remainder staples may thus sequentially unfold the local structure of another origami. This process may be energetically favorable because the remainder strands make continuous duplex which likely has a lower energy than origami structure (because of its crossovers and twist strain). In some of our initial experiments on shape complementary origami, we experienced this problem: individual origami folded well but when mixed together remainder staples from one origami caused large structural disruptions in other origami. We do not show this data since none of our final designs exhibit the problem.

Two potential solutions exist: (1) One can avoid the use of remainder staples – in most cases single-stranded remainder sections of the scaffold will cause no further problem. (2) One can design the remainder loops of different origami to coincide (have almost the same sequence), so that the remainder staples of one origami will not bind and invade loopouts of another origami. The latter approach was used successfully in our **A-B-C-D** chain system.

S2.7.3. Possible collisions between edge staples

When designing an origami system with uniform edge sequences (e.g. 'GC') as in our system, if one takes the same approach as ours – generating loopouts to shift the scaffold sequences – one should note that doing so limits the number of possible edge staple strands. In the 7249-base sequence of the M13mp18 scaffold strand, there are 393 occurrences of 'GC' (occurring on average every 18.4 bases, see schematic Fig. S9 below). Hence, ideally, there are 393 different positions at which edge staples can be located. Given a particular geometric design for an origami, one has some choice in terms of which edge staple positions to use; one can change the edge staple sequence at a particular geometric position in the origami by changing the length of the loopouts and/or changing the position at which the scaffold sequence starts in the design. However, when designing multiple origami that are large and use up all the sequence, or further when the origami designs share a similar "start position" for the scaffold sequence (as occurs in our shape-coded **A**, **B**, **C**, **D** system) there is a high probability that some of edge staples from different designs will share subsequences or have identical sequence. For the shape code system we explored this does not cause any difficulties since all edge staple positions are occupied by design. However, in some potential systems (say a hybrid shape code/binary code system) it might be possible for an edge staple present in one origami to fill in an empty edge staple position in another origami and give unexpected results. For example, in some of our initial experiments (not shown) edge staple collisions resulted in unintended aggregation. We note that taking an adapter strand approach to controlling edge sequences (as suggested in the main text) would obviate this problem.

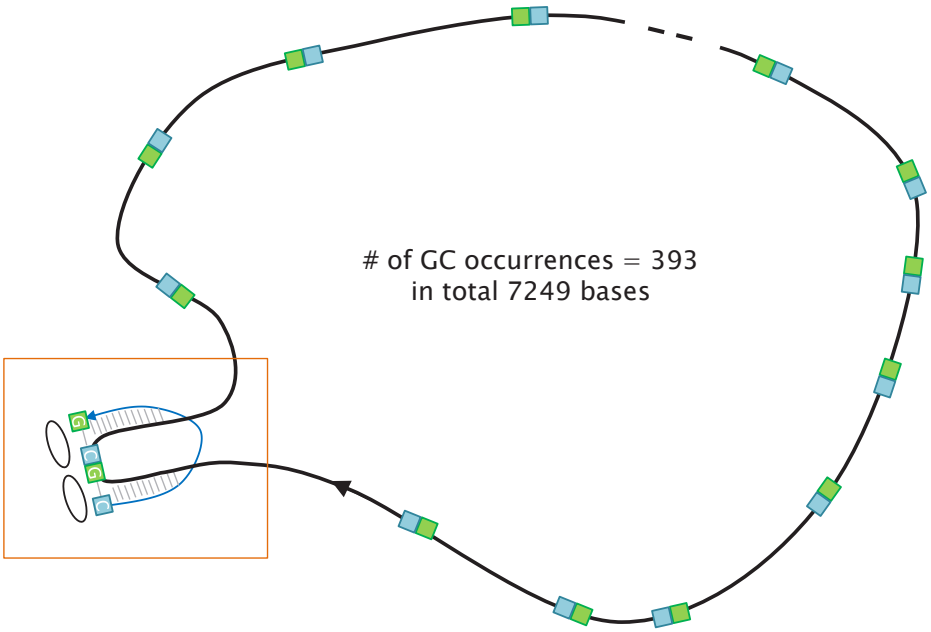


Figure S9. The limited number of 'GC' occurrences in the scaffold strand constrains the number of usable edge staple strands. In case of the M13mp18 scaffold strand, with 7249 bases in total, there are 393 occurrences of 'GC'. The black circular strand represents the scaffold. The boxed area shows how an edge staple strand (blue) binds to the scaffold and forms two 'GC' blunt-ends (depicted by ellipses).

Supplementary Note S3: Thermodynamic measurements

The free energy of the stacking bonds was measured by assuming that monomers and dimers of 'one-sided' rectangle origami (origami with edge staples on only one side, Fig. S10a) were at equilibrium. The initial monomer concentration equals the total origami concentration, which was assumed to be the initial scaffold concentration (assuming the yield of origami formation was ~100%). The equilibrium concentrations of monomers and dimers were measured by depositing the samples on mica and counting the numbers of each in AFM images (e.g. Fig. S10c). Here the relative ratio of monomers and dimers on surface was assumed to be representative of the ratio in solution. At least two processes could invalidate this assumption: (1) origami dimers might break upon deposition, artifactually elevating the monomers count or (2) origami monomers might land so close to each other that they would be scored as a dimer, artifactually elevating the dimer count. We did not try to estimate the frequency of these processes but we did dilute the origami 5-fold from their formation concentration before depositing them; this decreased the probability of a mismeasurement due to (2). In other experiments dilution was performed on the mica surface, by pipetting a sample onto a 5× larger volume of buffer on the surface. Because origami stick so quickly to mica, this protocol would run the risk of depositing dimers and monomers before they had the chance to equilibrate at the new concentration. To decrease the potential for this effect, sample solutions were pre-diluted, left to equilibrate for ~5 hours (a longer equilibration time, e.g. 10 hours, was tested for the p=6 system and did not show a statistically significant difference, so we assumed that a 5 hour equilibration time was long enough for the p=6 and weaker bonds), and then deposited without further dilution. (To be completely free from the effects of surface deposition and dilution and to obtain more detailed thermodynamic parameters, e.g. T_m, ΔH, and ΔS, one could alternatively adopt solution-based measurement techniques such as real-time FRET analysis.[3])

The free energy of the stacking bonds was calculated as follows. From the counts of monomers M, correct dimers, D, and incorrect dimers (misalignments or other orientations, *other*), and the concentration of origami, [*origami*], we calculated the monomer concentration, [M], and dimer concentration, [D]:

$$[M] = \frac{M\,[origami]}{M + 2D + 2other} \text{ and } [D] = \frac{D\,[origami]}{M + 2D + 2other}$$

From [M] and [D] we calculated the equilibrium constant and the free energy of the bond

$$K = \frac{[D]}{[M]^2} \text{ and } \Delta G = -R\,T\ln K$$

where R is the gas constant (8.314 J/mol·K) and T is the temperature 295 K (22°C).

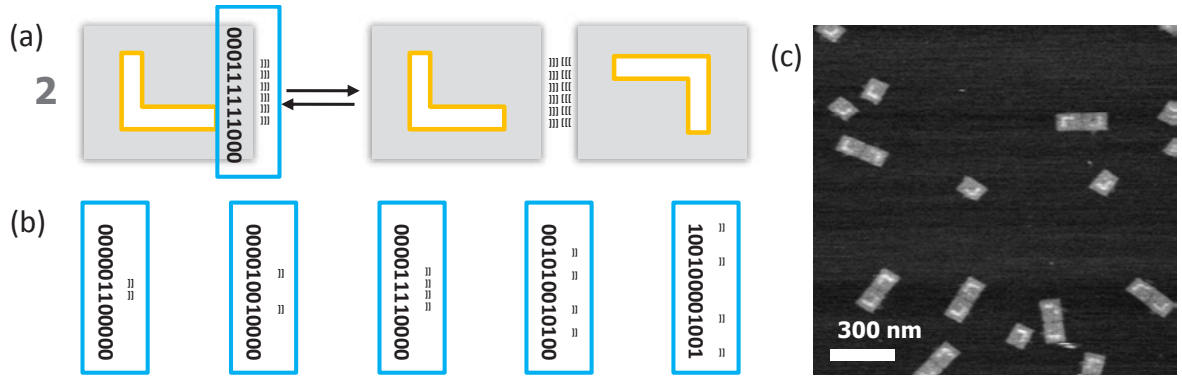


Figure S10. Thermodynamic measurements. (a) Schematic of the monomer/dimer equilibrium for 'one-sided' origami with six active stacking patches in the middle (binary sequence '000111111000'). (b) The number of active stacking patches and their locations were varied as shown and the free energy was measured in each case. (c) A representative AFM image showing the distribution of the monomers and dimers as well as the distribution of bond orientations.

S3.1. First energy model: assuming loop-loop interactions are neutral

We measured the free energy of stacking bonds for various systems with different numbers and sequences of active patches (Fig. S10ab). Assuming that loop-loop interactions have a neutral effect on the free energy of stacking bonds we calculated the free energy per helix for each system (Table S4 and Fig. S11). The total binding energy was expected to be linear in the number of active patches which would imply a constant free energy per helix, yet we observed free energies that varied between -2.67 kcal/mol and -1.42 kcal/mol depending on the system.

A few trends were observed. First, we observed that the magnitude of the binding energy per helix decreased as the number of helices increases. We hypothesize that the resulting sublinearity of binding energy is due to residual large-scale twist (or other deformation) of the origami structure; our picture is that as the number of stacking patches increases and the patches become more spread out, the bending (or twisting) penalty per patch increases. Such residual deformations seem likely to occur because, although twist correction yields a better *average* twist and *decreases* global twist, local twist still differs greatly from 10.4 bp/turn. This hypothesis is compatible with a second observed trend, one within the 4-patch systems. The three 4-patch systems were designed such that one has all its active patches compactly arranged in the middle ('0000**1111**0000'), another has its active patches highly spread-out ('**1**00**1**0000**1**00**1**') and the third has active patches with an intermediate spacing ('00**1**0**1**00**1**0**1**00'). Dimers in the system with compactly arranged active patches were most stable, while dimers in the system with highly spread-out active patches were least stable; this again suggests a bending or twisting penalty that increases as active patches spread out. Yet, the binding energies for the 2-patch systems did not show a statistically significant difference between the system with most compactly arranged active patches ('000001**1**00000') and the system with more spread-out active patches ('0000**1**00**1**0000'). One can imagine that 2-patch systems would be less affected by structural deformations because the bonds are formed by 2-point connections – no matter how much the edge is bent, 2-point connections could be made by rotation of one origami with respect to the other.

Because we hypothesize that non-stacking factors are all destabilizing, we suggest that the average energy obtained for the 2-patch systems, -2.63 kcal/mol (1× TAE with 12.5 mM Mg^{2+}, 22°C), is most reflective of a pure stacking interaction. One literature value (Ref. 25 of the main text) for the energy of GC/CG stacking is -2.17 kcal/mol (1M Na^{+} solution at 37°C). While buffer conditions between the two experiments differ, we did our best to make the measurements comparable by correcting the literature value using temperature-dependent data given in Ref. 25 of the main text. Fig. S12 shows a plot that we reproduced based on experimental data given in Fig. 3a and Supplementary Table 2 of Ref. 25 of the main text. Data were taken for five different temperatures (32°C, 37°C, 42°C, 47°C, and 52°C). Assuming that the temperature dependence of the enthalpy and the entropy of blunt-end stacking is negligible for the given temperature range, it is appropriate to make a linear fit to ΔG_{st} as a function of temperature. A regression line and its equation ($R^2 = 0.8943$) are shown in Fig. S12. Linear extrapolation to the *y*-axis (T=22°C) gives an energy of -2.42 kcal/mol at 22°C, which is a very close value to the value we obtained.

System[&]	binary code	[origami] (nM)	ΔG_{st} (kcal/mol hx)	N (origami count)
2patch-(6,7)	000001**1**00000	0.424	-2.5889	362
2patch-(5,8)	000**1**00**1**0000	0.848*	-2.6738	276
4patch-(5,6,7,8)	0000**1111**0000	0.424	-1.7644	178
4patch-(3,5,8,10)	00**1**0**1**00**1**0**1**00	0.424	-1.6593	566
4patch-(1,4,9,12)	**1**00**1**0000**1**00**1**	0.424	-1.5578	360
6patch-(4,5,6,7,8,9)	000**111111**000	0.212[#]	-1.4223	442

Table S4. Free energy of the stacking bond per helix for various systems.
[&] Numbers in parentheses indicate the locations of active stacking patches (the 1's in the binary sequences).
*A higher concentration was used because it was hard to find dimers for this system.
[#] A lower concentration was used because it was hard to find monomers for this system.

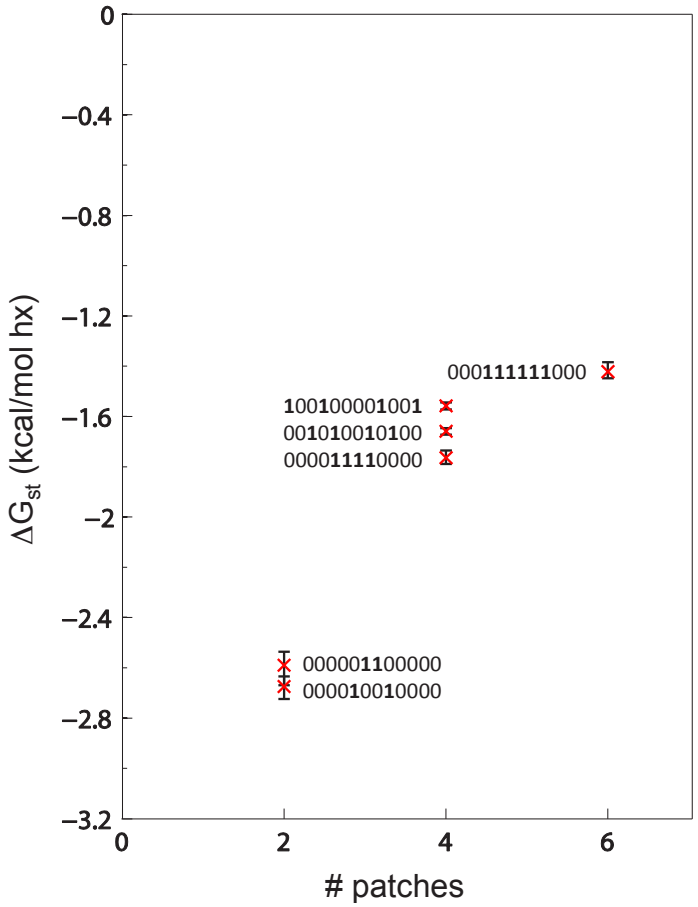


Figure S11. Free energy of the stacking bond per helix for various systems. Energy values per helix vary depending on the number of patches, indicating a nonlinear relationship between the stacking energy and the number of helices. The overall trend (decreasing $|\Delta G_{st}|$ as the number of patches increases) suggests that patches farther away from the middle of the edge must bend more (to counter some remnant global deformation) to bind; this hypothesis is consistent with the trend within the 4-patch systems. The binding energies for the 2-patch systems did not show a statistically significant difference (the error bars partially overlap.) Error bars indicate standard error, obtained by bootstrapping the count data and propagating errors through the equations.

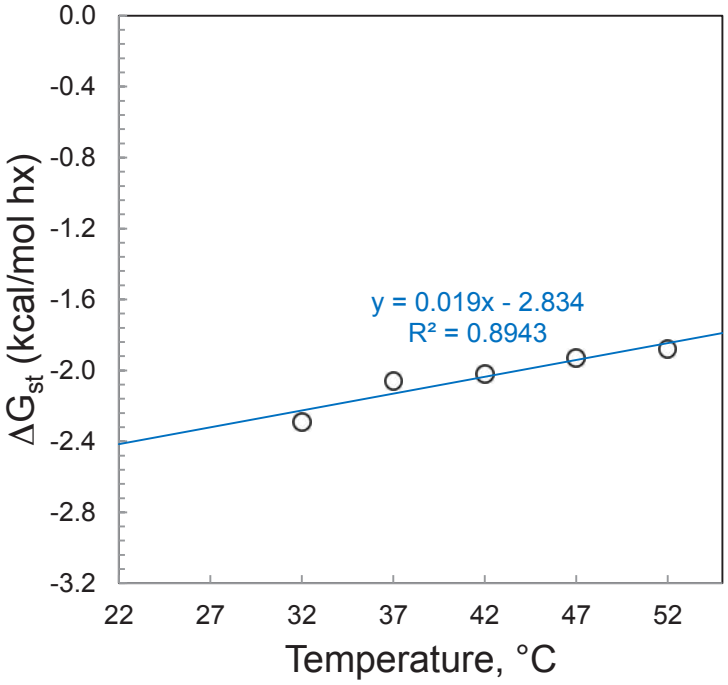


Figure S12. Temperature dependence of the stacking free energy (data taken from Ref. 25 of the main text). A linear fit and its extrapolation gives a stacking free energy of -2.42 kcal/mol at 22°C, which is very close to the value we obtained, -2.63 kcal/mol, at the same temperature.

S3.2. Second energy model: fitting with non-zero loop-loop interactions

In the previous energy model, we assumed that loop-loop interactions are negligible. Here we take an alternate approach and assume that loop-loop interactions are not necessarily neutral and further that they may have some constant free energy value, ΔG_{ll}, either positive or negative. This assumption is somewhat simplistic: if loop-loop interactions are attractive due to unintended base-pairing or base-stacking then the attraction is likely to be highly sequence-dependent, and hence different between different pairs of loops. We also assume that the stacking free energy is a linear function of both the number of active patches (each contributing $\Delta G_p = 2\Delta G_{st}$) and the number of inactive patches (each contributing ΔG_{ll}). For simplicity, for a given p, we averaged the measured free energies for different arrangements of active patches. This resulted in a single free energy for each of three values of p (2, 4, and 6) which allowed us to write the following three equations:

(1) $2\Delta G_p + 10\Delta G_{ll} = -10.53$
(2) $4\Delta G_p + 8\Delta G_{ll} = -13.28$
(3) $6\Delta G_p + 6\Delta G_{ll} = -17.07$
(all in kcal/mol)

Here we have more equations than unknowns. In principle, for this system of linear equations to be consistent, the intersection of each pair of equations should coincide exactly. In practice, because of experimental error and potential sequence-dependent effects, we expect that the intersections should lie in close proximity to each other. Fig. S13 shows a plot of the three equations. The intersections occur at $(\Delta G_p, \Delta G_{ll}) = (-2.03, -0.65)$, $(-2.24, -0.60)$, and $(-2.37, -0.48)$, for equations (1) and (2), equations (1) and (3), and equations (2) and (3), respectively. Least squares analysis gives a solution of $(\Delta G_p, \Delta G_{ll}) = (-2.23, -0.59)$ with a root mean square error of 0.24 (which can be roughly interpreted as the average distance of the solution from each intersection in the plot of ΔG_{ll} vs. ΔG_p.

Thus we find that the average free energy of loop-loop interactions (ΔG_{ll}) is negative (suggesting that loop-loop interactions contribute favorably to the binding) but small—less than half the average free energy of a single base pair, -1.41 kcal/mol (nearest neighbor model – Ref. 28 of the main text). It would be interesting to ask whether the average loop-loop interaction is typical, or whether most loop-loop interactions are neutral and just a few inactive patches contribute most of the binding energy. Answering this question will require more experiments, in particular measurements of the binding energy for stacking bonds that have the same stacking sequence, but loops of different base sequence. Another observation is that ΔG_{ll}, the binding energy of a pair of inactive patches, is one-fourth the free energy for an active patch ΔG_p. Its effect on stacking bond strength will depend not just on this ratio, but the number of inactive patches used. For a 16-patch stacking sequence with 7 active patches, the 9 inactive patches will make a contribution to the binding energy that is roughly equivalent to two active patches, and about one-fourth (coincidentally) of the total free energy of the bond. With respect to predicting the ratios of correct vs. incorrect bonds, the contribution of loop-blunt end interactions (which occur frequently in mismatch incorrect bonds) will likely have to be included; so far we have no quantitative data that address such interactions. Finally, we observe that if this model is correct then we must reconcile the relatively small $|\Delta G_{st}|$ observed (1.12 kcal/mol) with much higher literature values (2.42 kcal/mol). It may be partially due to the difference in measurement methods, or it might suggest that the near-B-form stacks

(Note S2.5) which occur in 'relaxed' edges do, in fact, have a somewhat smaller free energy. Future experiments using 'crossover-free' edges may address this question.

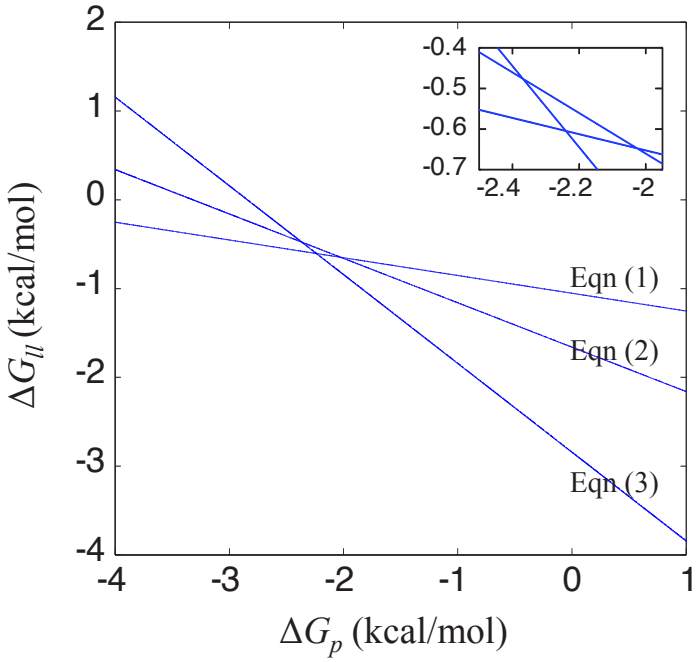


Figure S13. Plots of the three equations given for ΔG_p and ΔG_{ll} in Note S3.2. Least squares analysis gives a solution of (ΔG_p , ΔG_{ll}) = (-2.23, -0.59) with a root mean square error of 0.24. Inset shows a zoom-in view of the plot near the intersections of the three lines.

Supplementary Note S4: Additional AFM Data

S4.1. Stacking of rectangles (figure caption on next page)

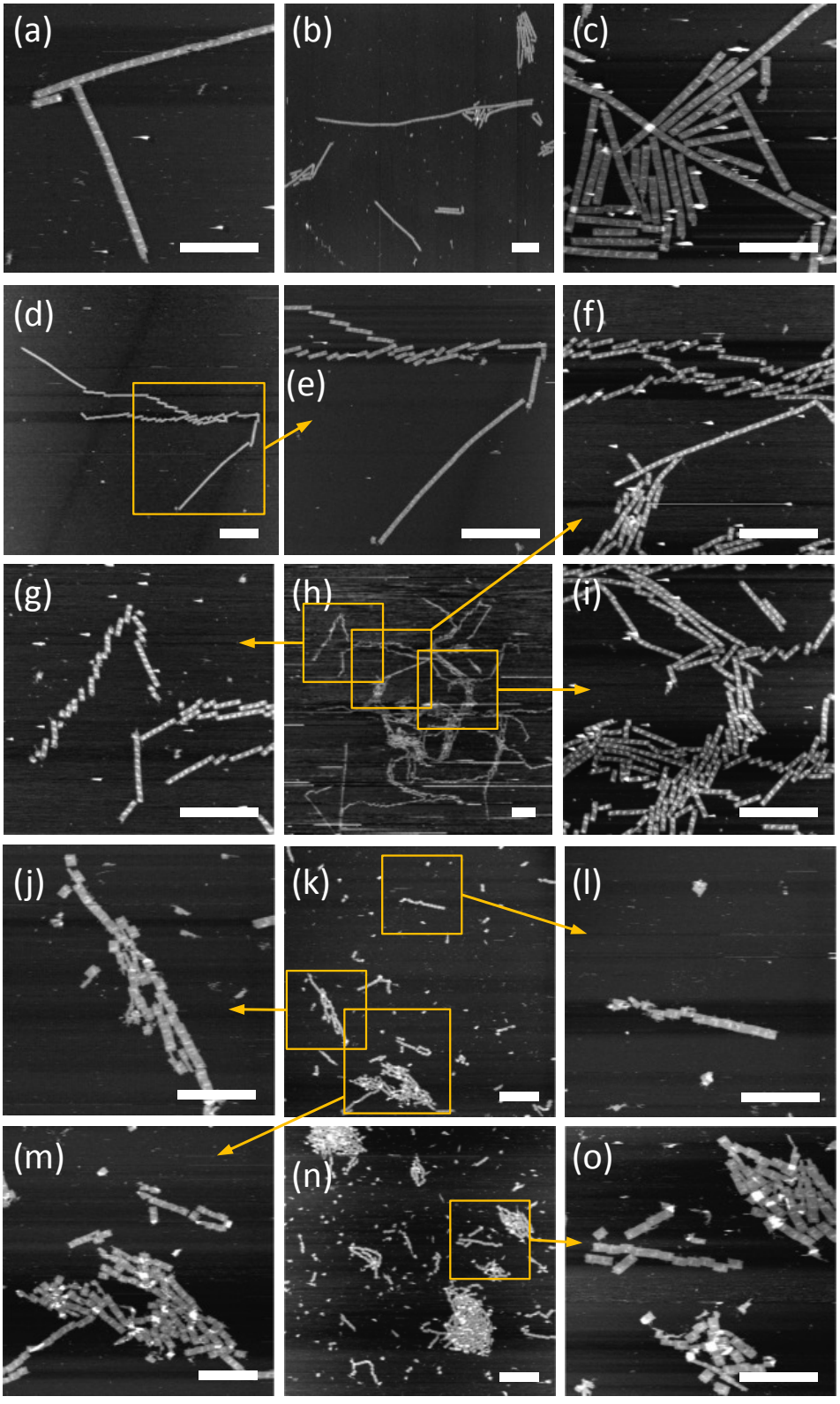


Figure S14. Wide-field AFM images of rectangle origami systems. (a-c) Twist-corrected origami with relaxed edges. Note that they form chains with lengths on the order of ~10 um. Chains formed by these origami break in a way that suggests that the breakage occurs upon deposition since pieces lie close to each other. However, in contrast to the twisted origami shown in (d-i), twist-corrected origami break into long pieces and show no preferred direction for the shift between neighboring pieces. Note also that twist-corrected relaxed chains are straight with very rare dislocations, as opposed to the twist-corrected origami with stressed edges shown in (j-o). (d-i) Twisted origami (with relaxed edges). Chains break with a characteristic periodicity (2-6 origami) and directional offset. Note that some parts of the chains seem to unwind while depositing, especially near the ends (as suggested by the straight sections near the ends of twisted chains). (j-o) Origami with stressed edges (with twist-correction). Bonds are promiscuous: many dislocations occur and the bond orientations are random. Orange boxes and arrows show zoom-in areas. Scale bars in (a), (c), (j), (l), (m), (o) are 600 nm, and scale bars in (b), (d-i), (k), (n) are 1 μm.

S4.2. 5-origami chains with orthogonal binary-coded bonds

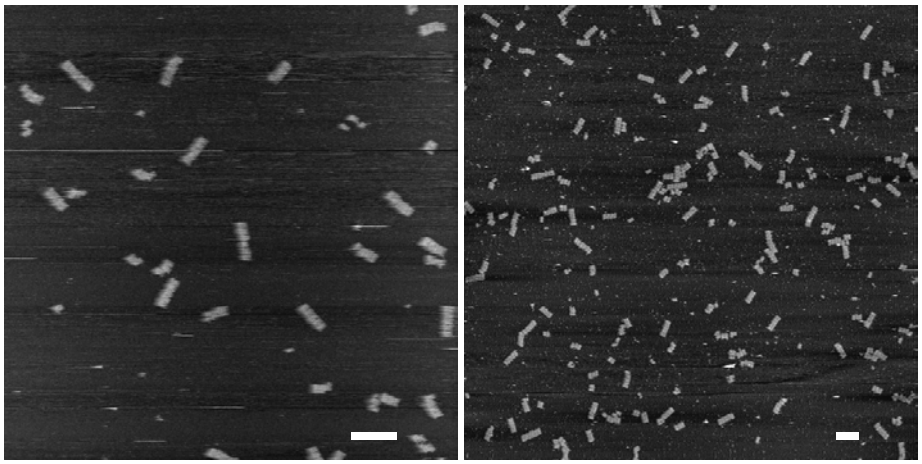


Figure S15. Wide-field AFM images of 5-origami chains with orthogonal bonds based on a binary code. Full correct bonds between each pair of origami resulted in chains with five distinct origami. Due to mismatched bonds and small stoichiometric discrepancies, shorter chains, longer chains, and 5-origami chains containing incorrect bonds were also found. 88% of total bonds analyzed (N=66) were correct bonds, and the fraction of origami found in 5-origami chains was 31% (N=192). Scale bars, 500 nm.

S4.3. Origami dimers and chains with orthogonal shape-coded bonds

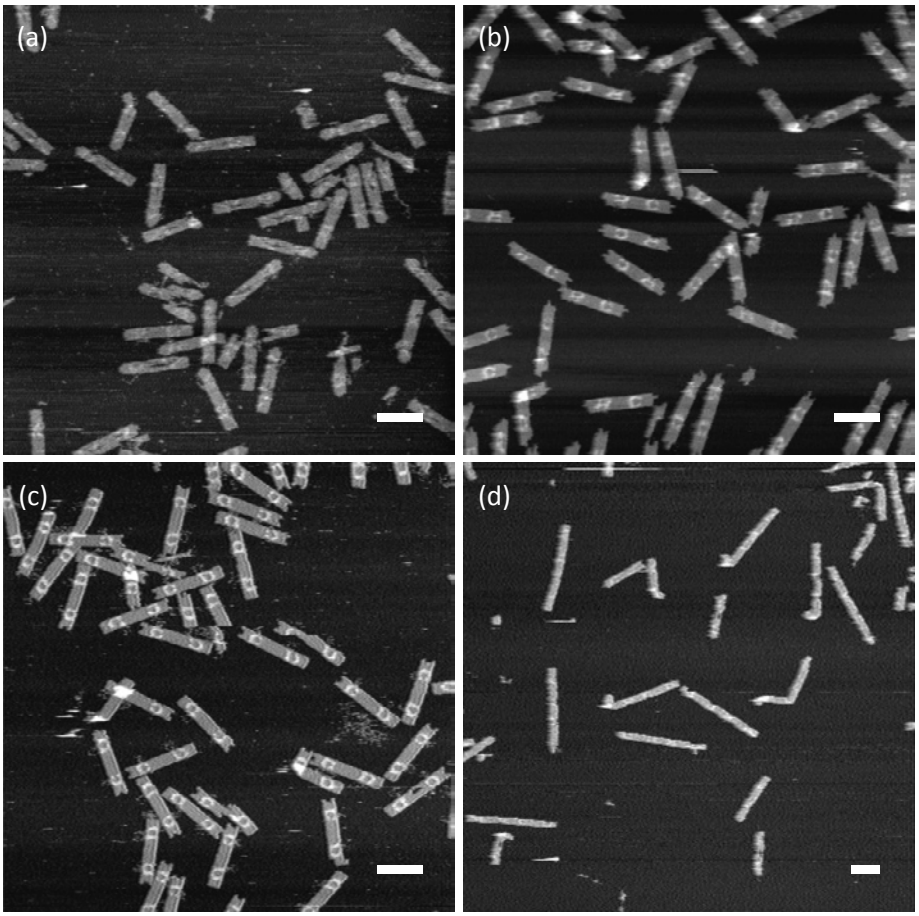


Figure S16. Wide-field AFM images of dimers and 4-origami chains with orthogonal bonds based on a shape code. (a-c) AFM of dimers of (a) $\mathbf{A}_r$ + $_l\mathbf{B}$, (b) $\mathbf{B}_r$ + $_l\mathbf{C}$, and (c) $\mathbf{C}_r$ + $_l\mathbf{D}$, respectively, show high yields of correct, full bonds. The fractions of correct bonds out of total bonds analyzed were 95% (N=191), 98% (N=203), and 97% (N=179), respectively, and the fractions of origami found in correct dimers were 91% (N=397), 90% (N=442), and 91% (N=384), respectively. (d) AFM of the **A-B-C-D** system show 4-origami chains with full, correct bonds (some chains shown are folded), and some shorter chains that may result from stoichiometric discrepancies or mismatched bonds. 81% of total bonds analyzed (N=279) were correct bonds, and the fraction of origami found in 4-origami chains was 44% (N=430). Scale bars, 200 nm.

References

1. Sherman, W. B. & Seeman, N. C. Design of minimally strained nucleic acid nanotubes. *Biophys. J.* **90**, 4546-4557, doi:10.1529/biophysj.105.080390 (2006).

2. Sa-Ardyen, P., Vologodskii, A. V. & Seeman, N. C. The flexibility of DNA double crossover molecules. *Biophys. J.* **84**, 3829-3837 (2003).

3. Saccà, B., Meyer, R., Feldkamp, U., Schroeder, H. & Niemeyer, C. M. High-throughput, real-time monitoring of the self-assembly of DNA nanostructures by FRET spectroscopy. *Angew. Chem., Int. Ed.* **47**, 2135-2137 (2008).

Supplementary Note S5: Sequence lists and diagrams

S5.1. Rectangle with 10.44 bp/turn

Core

Seq name	Sequence
r0t11ml1	TTAGATACTATTTTCATTTGGGGAATGCCT
r0t11mr1	TAAGAACGGAGGTTTTGAAGCCTAGTCAGA
r0t11mr2	TAATGCAGTTCGAGCCAGTAATAACTGACCTA
r0t11mr3	AAATCAGAGCTATTTTGCACCCAGAGAATAAC
r0t11mr_fr	ATAAGTCCATATTTAACAACGCCGTGTGAT
r0t11seam_l	AATATCGCTAAGAGGAAGCCCGAAACCTCCCG
r0t11seam_r	CCAGACGACGACAAAAGGTAAAGTATAACCTG
r0t13ml1	GAGTAATGCGGAGACAGTCAAATAACGTTA
r0t13mr1	TAAAGTACCGACAATAAACAACAGGTATTC
r0t13mr2	AATTTAATAAATGCTGATGCAAATTTTTAATG
r0t13mr3	GAGGCATTAACGCGCCTGTTTATCTTCATCGT
r0t13mr_fr	AAATAAGACCTTTTTAACCTCCGTGAGTGA
r0t13seam_l	TTTAGCTAATTTCGCAAATGGTCAAATTCTGT
r0t13seam_r	TATATTTTAGAACGCGAGAAAACTAAAGGGTG
r0t15mr1	CAAGACAAAGTTAATTTCATCTTGAGAATA
r0t15mr3	CTATATGTGGTTTGAAATACCGACCAACATGT
r0t15mr_fr	ATAACCTACAATAACGGATTCGTTATACTT
r0t15seam_l	AGAAAGGCTGTAGGTAAAGATTCATTTTCAAA
r0t17mr3	TCAATTACACATAAATCAATATATGGCTTAGG
r0t17mr_fr	CTGAATACGTATTAAATCCTTTGGCAAATC
r0t19ml1	CCACACAAGGGGTGCCTAATGAGAGCAGGC
r0t19ml3	TCCTGTGTAATTGCGTTGCGCTCAAGAGAGTT
r0t19mr2	CCCTCAATTAACACCGCCTGCAACATTCACCA
r0t19mr_fr	AACAGTTCCACCAGCAGAAGATACATTCTG
r0t19seam_r	AGCATCACGCCAGCAGCAAATGAAATAAAGTG
r0t1ml1	TCCAAAAGTTTCGAGGTGAATTTGTAATGC
r0t1ml3	TTTTCACGCCGATAGTTGCGCCGAACTTTTTC
r0t1mr2	TTTAACGGGAATGGAAAGCGCAGTCCATCTTT
r0t1mr_fr	CCGTATATGGCCTTGATATTCAGAGCCACC
r0t1seam_r	TTGATGATTCCAGTAAGCGTCATACGGTTTAT
r0t1t_seam	CGCCACCCTCAGAACCGCCACCCTCAGAACCG
r0t1tl1	CCACCCTCAGAGCCACCACCCTCAAAAGGC
r0t1tl2	GGATAGCAAGCCCAATAGGAACCCCAACAGTT
r0t1tl3	TAACACTGAGTTTCGTCACCAGTTTTTCTGT
r0t1tr1	GGTGTATCACCGTACTCAGGAGGTTAATAAGT
r0t21ml1	GAAAATCCCCTTATAAATCAAAACGGCGAA
r0t21ml3	AAGCGGTCGGTTGAGTGTTGTTCCGAGCCCCC
r0t21ml_fl	TTTCCAGTCGTAATCATGGTCACGAAAGGG
r0t21mr2	GTCACACGTTGCAACAGGAAAAACTAAAGGGA
r0t21mr3	GTCAGTATCAATATCTGGTCAGTTGCCCGAAC
r0t21mr_fr	GCCAACATGCTGGTAATATCCAAATCCTGA
r0t21seam_r	AAATGGATTACATTTTGACGCTCACGAAATCG
r0t23b_seam	CGTGGCGAGAAAGGAAGGGAAGAAAATCAGAG
r0t23bl1	GATTTAGAGCTTGACGGGGAAAGCGAATAGCC
r0t23br1	CGGGAGCTAAACAGGAGGCCGATGCTCATG
r0t23br2	TTTTAGACAGGAACGGTACGCCAGGAACAATA
r0t23br3	GAAGTGTTTTTATAATCAGTGAGCTCAAACT
r0t23ml2	CGAGATAGCACGCTGGTTTGCCCCTGAGCTAA
r0t23ml_fl	ACAAGAGCACCGCCTGGCCCTGCTGCCCGC
r0t23mr1	GAAATACCTATTTACATTGGCAGAGTGCCA
r0t23mr3	GCCAGCCAACCAGTAATAAAAGGGAAAACAGA
r0t23seam_l	GCAAAATCTGTTTGATGGTGGTTCATCGTCTG
r0t3ml1	CACTACGAATACACTAAAACACTATCTTGA
r0t3ml2	GCTTGATATTGAAAATCTCCAAAAATTTTCAG
r0t3ml3	GTTTCCATCGATTATACCAAGCGCGACCAGGC
r0t3ml_fl	ACAACCATTGCTAAACAACTTTATGTACCG
r0t3mr1	TTTACCGTACAGGAGTGTACTGGTTAGTAC
r0t3mr2	TCATAATCAATCAAGTTTGCCTTTCAAAAGGG
r0t3mr3	TAAAGCCAGGTCAGTGCCTTGAGTGATATAAG
r0t3mr_fr	ACCGGAATCGATAGCAGCACCGGAAGGTAA
r0t3seam_l	CAGCTTGCGAGCCTTTAATTGTATCATGGCTT
r0t3seam_r	CATAGCCCCGCGTTTTCATCGGCACGAAAGAG
r0t5ml1	CAAGAACCCTGCTCATTCAGTGAAATGCAG
r0t5ml2	ACCCCCAGTAAACGGGTAAAATACCTTAAACA
r0t5ml_fl	GTACAACCTTTGAGGACTAAAGCAATGACA
r0t5mr1	GACTGTAGCCTTATTAGCGTTTGCTCTGAA
r0t5mr2	CGACATTCGAAACGCAAAGACACCATAATAAG
r0t5mr3	AGCGACAGAAAATCACCGGAACCACAAACAAA
r0t5mr_fr	ATATTGAGCAAACGTAGAAAATAGCTATCT
r0t5seam_l	GCAAAAGAAGGCACCAACCTAAAATTTTCGGT
r0t5seam_r	ATATGGTTTTTGTCACAATCAATAAATCAACG
r0t7ml1	ATACATAAAACACTATCATAACCTTGCATC
r0t7mr1	AAGTTTATTACCAGCGCCAAAGAAGCGTCA
r0t7mr2	AGCAAGAATGAACACCCTGAACAATAAATCAA
r0t7mr3	ATATAAAAAACCGATTGAGGGAGGTAATCAGT
r0t7mr_fr	TACCGAAGCAGCCTTTACAGAGCTACAATT
r0t7seam_l	TAACAAAGGGATATTCATTACCCAGAAAATTC
r0t7seam_r	CCACAAGAGAGCGCTAATATCAGAGAGGCATA
r0t9ml1	AAAAAGATGTTTTAATTCGAGCTTTGACCA
r0t9mr1	GGGTAATTATTGAGTTAAGCCCAACGGAAT
r0t9mr2	GATTAGTTTATAGAAGGCTTATCCTGTTCAGC
r0t9mr3	GAATTAACACAATGAAATAGCAATACATACAT
r0t9mr_fr	TTATCCTCAAGCCGTTTTTATTAACAATAG
r0t9seam_l	GTAAGAGCCGCCAAAAGGAATTACGAGATAAC
r0t9seam_r	ACTTGCGGCGAGGCGTTTTAGCGAAGACTTCA
r1t0tr2	GGATAAGTGCCGTCGAGAGGGTTAACAGTGC
r1t0tr3	TAGGATTAGCGGGGTTTTGCTCAGTGCCTATT
r1t10fr1	AGAACAAGGAATCTTACCAACGCTGTCAAAAA
r1t10fr2	TCTTTCCATTAAACCAAGTACCGCATATCCCA
r1t10fr3	AACGGGTAGAGCCTAATTTGCCAAATCCAA
r1t12fr1	AGAATCGCCTGAACAAGAAAAATAACTCATCG
r1t12fr2	TCCTAATTACGCTCAACAGTAGGGAACACCGG
r1t12fr3	TAAAGCCATACGAGCATGTAGAATTCCAAG
r1t14fr1	GAGAGACTGCGTTAAATAAGAATACTTAATTG
r1t14fr2	AATCATAATGAATTTATCAAAATCCGCTATTA
r1t14fr3	GTCAATAGTTACTAGAAAAAGCCACCAGTA
r1t16fr1	CGGGAGAATGCTTCTGTAAATCGTATAGGTCT
r1t16fr2	ATTAATTTATACAGTAACAGTACCCTACCATA
r1t16fr3	AGATGAATTCCCTTAGAATCCTTGAGAAGA
r1t18fr1	CGACAACTATGGAAGGGTTAGAACTTTTACAT
r1t18fr2	TCAAAATTAAGTATTAGACTTTACGGTTATCT
r1t18fr3	GGATTTAGATTTGCACGTAAAACTAACGTC
r1t20fr1	CGAACGAAGAAAGGAATTGAGGAAAAACAATT
r1t20fr2	AAAATATCCTAAAACATCGCCATTGACCTGAA
r1t20fr3	TGATAGCCTTTAGGAGCACTAACCATTTGA
r1t22fr1	ATCGGCCTGAGATAGAACCCTTCTAAAAATAC
r1t22fr2	AGCGTAAGTGCCTGAGTAGAAGAAGCCACCGA
r1t22fr3	ACATCACTAATACGTGGCACAGACGCGAAC
r1t24br4	GTAAAAGAGTCTGTCCATCACGCAGTAATA
r1t2fr1	CAGACGATAACAGTTAATGCCCCCTACCAGGC
r1t2fr2	TCGGAACCCATTGACAGGAGGTTGCGCCACCC
r1t2fr3	GCCGCCAGTATTATTCTGAAACAGAAGGAT
r1t4fr1	TGAAACCACCGCCTCCCTCAGAGCAGGCAGGT
r1t4fr2	TCAGAACCTAGCAAGGCCGGAAACAAGGTGAA
r1t4fr3	ATTACCATGCCACCCTCAGAGCCCAGAGCC
r1t6fr1	GTATGTTACGGAAATTATTCATTAGTCACCAA
r1t6fr2	TTATCACCATGATTAAGACTCCTTGTAAGCAG
r1t6fr3	GAACTGGCGTCACCGACTTGAGCTAGCACC
r1t8fr1	TGAAAATAGCCCTTTTTAAGAAAAATTACGCA
r1t8fr2	ATAGCCGACGATTTTTTGTTTAACAACGAGCG
r1t8fr3	ATAAGAAAACAAAGTTACCAGAACCCAAAA
r-1t0tl4	CAACGCCTGTAGCATTCCACAGATTTGTCG
r-1t10fl1	AAACGAGAGAGTACCTTTAATTGCTACGGTGT
r-1t10fl2	ATAAGAGGCTCAAATGCTTTAAACAGAGGGGG
r-1t10fl3	GAATCCCCTCATTTTTGCGGATGAGCTCAA
r-1t12fl1	CTGGAAGTACATCCAATAAATCATTTTTGCGG
r-1t12fl2	GGCAAAGAAAATATGCAACTAAAGTCCTTTTG
r-1t12fl3	CATGTTTTATTAGCAAAATTAAGTTGTACC
r-1t14fl1	GAGAAGCCGAGAGGGTAGCTATTTCATATGTA
r-1t14fl2	TCTACAAATATGACCCTGTAATACACAGGCAA
r-1t14fl3	AAAAACATGGCTATCAGGTCATTTGAACGG
r-1t16fl1	CCCCGGTTGCTTTCATCAACATTACGTAACCG
r-1t16fl2	CGAGTAACAAACTAGCATGTCAATTTGAGAGA
r-1t16fl3	TAATCGTAAACCCGTCGGATTCTGGATAGG
r-1t18fl1	TGCATCTGCTGCAAGGCGATTAAGGGTACCGA
r-1t18fl2	CGCCAGGGGTGTAGATGGGCGCATAATGTGAG
r-1t18fl3	TCACGTTGTTTTCCCAGTCACGAATGCCTG
r-1t20fl1	GCTCGAATTCGGGAAACCTGTCGTACAGCTGA
r-1t20fl2	CATTAATGCTCTAGAGGATCCCCGTTGGGTAA
r-1t20fl3	CAGGTCGAAATCGGCCAACGCGCGGTGGTTT
r-1t22fl1	TTGCCCTTTCCACTATTAAAGAACGCCGTAAA
r-1t22fl2	CAACGTCAACCAGTGAGACGGGCAGCCAGCTG
r-1t22fl3	TTCTTTTCAAGGGCGAAAAACCGTCACCCA
r-1t24bl2	GCACTAAATCGGAACCCTAAAGGAGTTTGGA
r-1t24bl3	AATCAAGTTTTTTGGGGTCGAGGTGTGGACTC
r-1t2fl1	ATGGGATTTCGCCCACGCATAACCGCAACGGC
r-1t2fl2	CGGTCGCTGACGTTAGTAAATGAAACAAACTA
r-1t2fl3	TCTTTCCAGAGGCTTGCAGGGAGAGCAGCG
r-1t4fl1	TACAGAGGGGAGATTTGTATCATCACTTTGAA
r-1t4fl2	AATTGTGTCATCGGAACGAGGGTAGATATATT
r-1t4fl3	AAAGACAGCGAAATCCGCGACCTACGGTCA
r-1t6fl1	AGAGGACACTTGAGATGGTTTAATAACGAACT
r-1t6fl2	TAATCATTGGAACCGAACTGACCAGCCTGATA
r-1t6fl3	ATCATAAGGTGAATTACCTTATGGGACGTT
r-1t8fl1	AACGGAACCAAAAGAAGTTTTGCCAGTTCAGA
r-1t8fl2	TAATAGTAAAATCTACGTTAATAATTCAACTT
r-1t8fl3	GGGAAGAAAAATGTTTAGACTGGATTCATT
rt-rem1	AGCACGTATAACGTGCTTTCCTCGTTAG
rt-rem2	ACAGGGCGCGTACTATGGTTGCTTTGACG
rt-rem3	ACCACACCCGCCGCGCTTAATGCGCCGCT
rt-rem4	CAAGTGTAGCGGTCACGCTGCGCGTAACC
rt-rem5	AGCGAAAGGAGCGGGCGCTAGGGCGCTGG

Edge staples that make *relaxed edges*

Right edge (from top to bottom)

r1t0_edge_r_2	CTGAGACTCCTCAAGATGAAAGTATTAAGAGG
r1t2_edge_r_2	CCACCAGAACCACCACACCACCCTCAGAGCCG
r1t4_edge_r_2	CCAGCAAAATCACCAGCATTTGGGAATTAGAG
r1t6_edge_r_2	CAATAATAACGGAATAGGAAACCGAGGAAACG
r1t8_edge_r_2	CCATATTATTTATCCCGTTACAAAATAAACAG
r1t10_edge_r_2	CTGTCTTTCCTTATCAACCAATCAATAATCGG
r1t12_edge_r_2	CGTTATACAAATTCTTTGTTTAGTATCATATG
r1t14_edge_r_2	CTTAGATTAAGACGCTGAAAACATAGCGATAG
r1t16_edge_r_2	CGTAGATTTTCAGGTTAGAAATAAAGAAATTG
r1t18_edge_r_2	CCGTCAATAGATAATAAACTAATAGATTAGAG
r1t20_edge_r_2	CTATTAGTCTTTAATGCAATATTTTTGAATGG
r1t22_edge_r_2	CAATACTTCTTTGATTAAATTAACCGTTGTAG

Left edge (from top to bottom)

r-1t2_edge_l_2	CGTAACGATCTAAAGTCAGCCCTCATAGTTAG
r-1t4_edge_l_2	CGGGATCGTCACCCTCTTAAAGGCCGCTTTTG
r-1t6_edge_l_2	CCGGAACGAGGCGCAGGCTCCATGTTACTTAG
r-1t8_edge_l_2	CTCATTATACCAGTCACGATTTTAAGAACTGG
r-1t10_edge_l_2	CGGAATCGTCATAAATATAGCGTCCAATACTG
r-1t12_edge_l_2	CTGAATATAATGCTGTGCTTAGAGCTTAATTG
r-1t14_edge_l_2	CATAAAGCTAAATCGGCAATAAAGCCTCAGAG
r-1t16_edge_l_2	CAAACAAGAGAATCGAGCCTGAGAGTCTGGAG
r-1t18_edge_l_2	CGGATTGACCGTAATGCCGTGGGAACAAACGG
r-1t20_edge_l_2	CCAGTGCCAAGCTTGCCGTTGTAAAACGACGG
r-1t22_edge_l_2	CGTATTGGGCGCCAGGGGGGAGAGGCGGTTTG
r-1t24_edge_l_2	CCCACTACGTGAACCATCTATCAGGGCGATGG

Edge staples that make *stressed edges*

Right edge (from top to bottom)

r1t0_edge_r	CTGAGACTCCTCAAGATGAAAGTA
r1t2_edge_r	TTAAGAGGCCACCAGAACCACCACACCACCCT
r1t4_edge_r	CAGAGCCGCCAGCAAAATCACCAGCATTTGGG
r1t6_edge_r	AATTAGAGCAATAATAACGGAATAGGAAACCG
r1t8_edge_r	AGGAAACGCCATATTATTTATCCCGTTACAAA
r1t10_edge_r	ATAAACAGCTGTCTTTCCTTATCAACCAATCA
r1t12_edge_r	ATAATCGGCGTTATACAAATTCTTTGTTTAGT
r1t14_edge_r	ATCATATGCTTAGATTAAGACGCTGAAAACAT
r1t16_edge_r	AGCGATAGCGTAGATTTTCAGGTTAGAAATAA
r1t18_edge_r	AGAAATTGCCGTCAATAGATAATAAACTAATA
r1t20_edge_r	GATTAGAGCTATTAGTCTTTAATGCAATATTT
r1t22_edge_brc	TTGAATGGCAATACTTCTTTGATTAAATTAACCGTTGTAG

Left edge (from top to bottom)

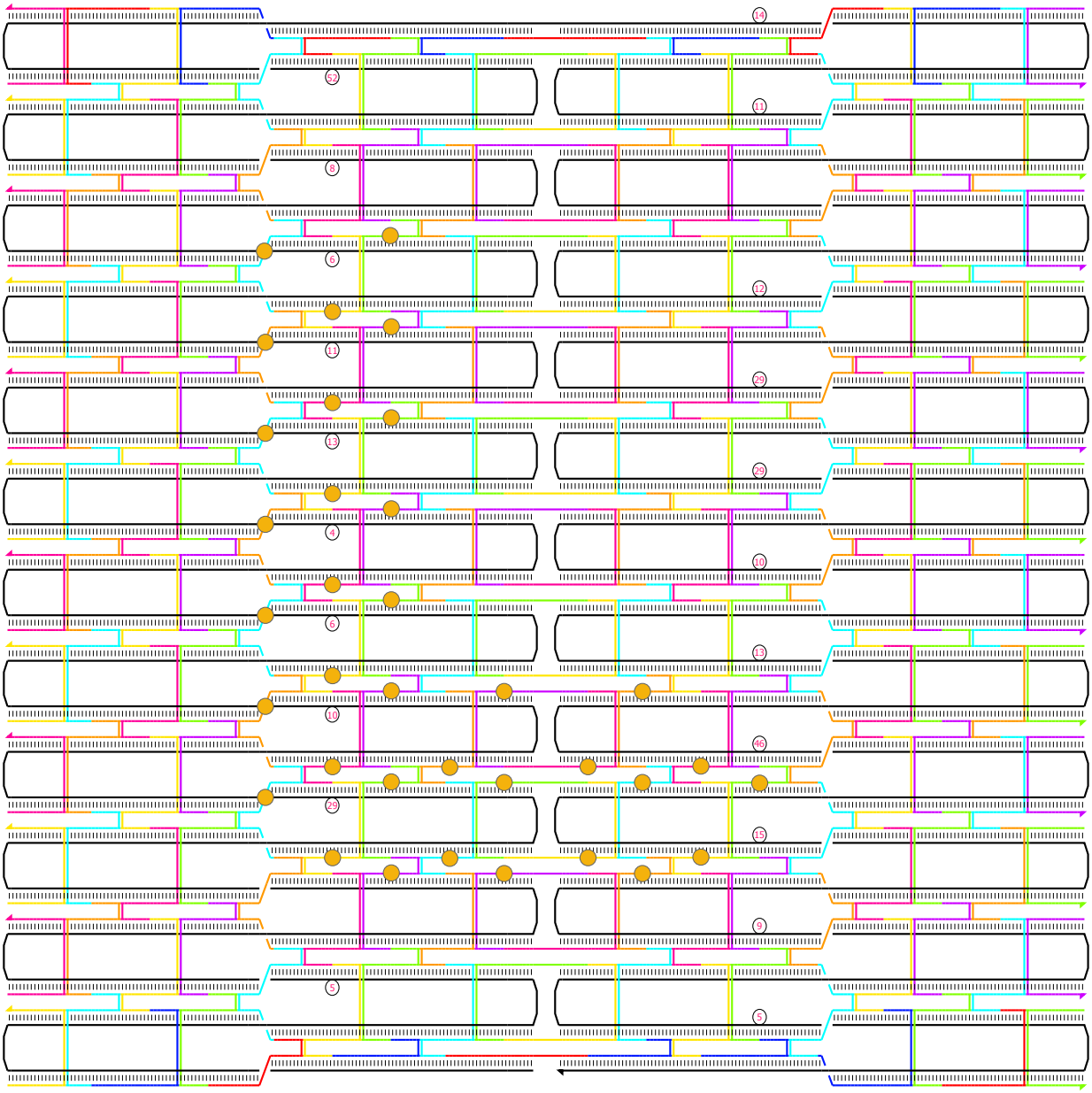
r-1t2_edge_tlc	CGCTTTTGCGTAACGATCTAAAGTCAGCCCTCATAGTTAG
r-1t4_edge_l	TTACTTAGCGGGATCGTCACCCTCTTAAAGGC
r-1t6_edge_l	AGAACTGGCCGGAACGAGGCGCAGGCTCCATG
r-1t8_edge_l	CAATACTGCTCATTATCACGATCACGATTTTA
r-1t10_edge_l	CTTAATTGCGGAATCGTCATAAATATAGCGTC
r-1t12_edge_l	CCTCAGAGCTGAATATAATGCTGTGCTTAGAG
r-1t14_edge_l	GTCTGGAGCATAAAGCTAAATCGGCAATAAAG
r-1t16_edge_l	ACAAACGGCAAACAAGAGAATCGAGCCTGAGA
r-1t18_edge_l	AACGACGGCGGATTGACCGTAATGCCGTGGGA
r-1t20_edge_l	GCGGTTTGCCAGTGCCAAGCTTGCCGTTGTAA
r-1t22_edge_l	GGCGATGGCGTATTGGGCGCCAGGGGGGAGAG
r-1t24_edge_l	CCCACTACGTGAACCATCTATCAG

Hairpin-labeled staples (hairpin sequence in lowercase)

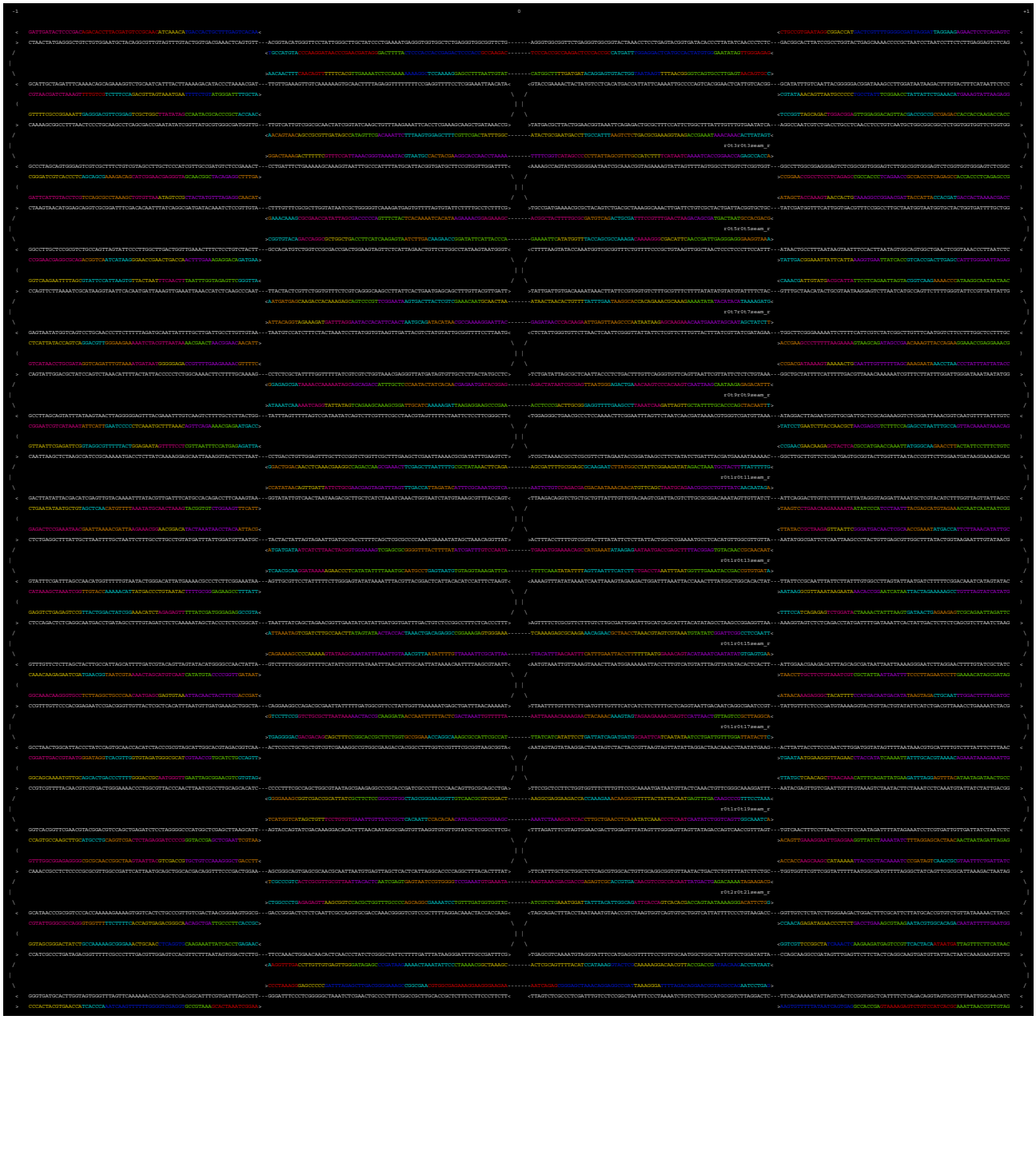
r0t5ml3_hp	GCTGGCTGCGAGAAACtcctcttttgaggaacaagttttctgtACCAGAACAGAAAGAT
r0t7ml2_hp	TGCCCTGAACCTTCATtcctcttttgaggaacaagttttctgtCAAGAGTACATCTTTG
r0t7ml3_hp	GATTTAGGCGATAAAAtcctcttttgaggaacaagttttctgtACCAAAATAAATCAGG
r0t9ml2_hp	CCAGACGAAATACCACtcctcttttgaggaacaagttttcttgtATTCAACTATAAGGCT
r0t9ml3_hp	TATTATAGCGGAAGCAtcctcttttgaggaacaagttttctgtAACTCCAAAGTTGATT
r0t11ml2_hp	AACCAGACTCAGAAGCtcctcttttgaggaacaagttttctgtAAAGCGGACTCGTTTA
r0t11ml3_hp	ATTCTGCGGGCATCAAtcctcttttgaggaacaagttttctgtTTCTACTAGGATAAAA
r0t13ml2_hp	GAAAAGGTAACGAGTAtcctcttttgaggaacaagttttctgtGATTTAGTTCAAAGCG
r0t13ml3_hp	AGAACCCTTTCAACCGtcctcttttgaggaacaagttttctgtTTCTAGCTCCCAAAAA
r0t15ml2_hp	ATATGATACATATATTtcctcttttgaggaacaagttttcttgtTTAAATGCCGCGAGCT
r0t15ml3_hp	GTATAAGCAAATAATTtcctcttttgaggaacaagttttcttgtCGCGTCTGGACGACAG
r0t15mr2_hp	GAAACAGTCTGAGCAAtcctcttttgaggaacaagttttctgtAAGAAGATGCAATTCA
r0t15seam_r_hp	AACAATTTAAGAAAACtcctcttttgaggaacaagttttctgtAAAATTAAATTTTTGT
r0t17ml2_hp	GCCATCAAAAATATTTtcctcttttgaggaacaagttttctgtGTGTACAGAACAAA
r0t17ml3_hp	CAGCTTTCCTATTACGtcctcttttgaggaacaagttttctgtCCAGCTGGTAGCTGTT
r0t17mr2_hp	TCAATATATAACATTAtcctcttttgaggaacaagttttctgtTCATTTTGATATCAAA
r0t17seam_l_hp	TAAATCAGTTAAAATTtcctcttttgaggaacaagttttctgtCGCATTAATTACATTT
r0t17seam_r_hp	ATATTCCTACCAGAAGtcctcttttgaggaacaagttttctgtGAGCGGAATCAGGCTG
r0t19ml2_hp	CCTCTTCGCGGCACCGtcctcttttgaggaacaagttttctgtCTTCTGGTATAGGAAC
r0t19mr3_hp	AGTTTGAGATCCTGATtcctcttttgaggaacaagttttctgtTGTTTGGACCTGATTG
r0t19seam_l_hp	CGCAACTGAAGCGCCAtcctcttttgaggaacaagttttctgtTTCGCCATTTATCATC
r0t21ml2_hp	CTCACATTGAAATTGTtcctcttttgaggaacaagttttcttgtTATCCGCTGGTGCGGG
r0t21seam_l_hp	TAAAGCCTCATACGAGtcctcttttgaggaacaagttttcttgtCCGGAAGCAAATCTAA
r0t7ml_fl_hp	AATTGGGGATGAACGtcctcttttgaggaacaagttttcttgtGTGTACAGAAACAAA
r0t9ml_fl_hp	GCTTTTGAACATTATtcctcttttgaggaacaagttttctgtTACAGGTGAGTAGTA
r0t11ml_fl_hp	GATTAGAATGACCATtcctcttttgaggaacaagttttctgtAAATCAAAGCGAGAG
r0t13ml_fl_hp	AGCATTATTCATTCCtcctcttttgaggaacaagttttctgtATATAACCAGGTCAG
r0t15ml1_hp	ATATTTTGCTCATTTtcctcttttgaggaacaagttttctgtTTTAACCAGCCGGAA
r0t15ml_fl_hp	AATGCCGTTTATTTCtcctcttttgaggaacaagttttctgtAACGCAAATAGTAGT
r0t17ml1_hp	ACCAGGCATTGGGAAtcctcttttgaggaacaagttttctgtGGGCGATCCACAATT
r0t17ml_fl_hp	GTAGCCAGATAATCAtcctcttttgaggaacaagttttctgtGAAAAGCGATAAATT
r0t17mr1_hp	CAAACATCCATTTGAtcctcttttgaggaacaagttttctgtATTACCTTCCAATCG
r0t19ml_fl_hp	GGATGTGCCAGTTTGtcctcttttgaggaacaagttttctgtAGGGGACGCCTTCCT
r0t19mr1_hp	AAGAAACCGATTATCtcctcttttgaggaacaagttttctgtAGATGATGGATGAAA
r0t21mr1_hp	CGCTGAGACTTGCTGtcctcttttgaggaacaagttttctgtAACCTCAACGGAACA

Diagram for rectangle with 10.44 bp/turn

with positions of dumbbells (orange circles) and positions and lengths of loopouts (black circles with numbers) indicated



Sequence diagram for rectangle with 10.44 bp/turn – (1) with edge staples that create *relaxed edges*



Sequence diagram for rectangle with 10.44 bp/turn – (2) with edge staples that create *stressed edges* (Only the edge staples are different.)

S5.2. Rectangle with 10.67 bp/turn

Core

Seq name	Sequence
r1t0g	AAACAAACATCAAGAAAACAAAATTAATTACA
r1t10e	TGTTGTTCGTGGACTCCAACGTCAGCGCGTAC
r1t10f	TAAAGAACCAGTTTGGAACAAGAGTTATTTTG
r1t12e	TAATTGCGCCTGTCGTGCCAGCTGGGGTTGAG
r1t12f	TCGGGAAATTGCGCTCACTGCCCGGCATTTTC
r1t14e	TCGGTGCGGGGGATGTGCTGCAAGACTCACAT
r1t14f	GGCGAAAGGGCCTCTTCGCTATTAATACATGG
r1t20f	GCTATATTCGCAAATGGTCAATAAAAACGAAA
r1t22e	ATCAAAAACGCGTTTTAATTCGAGACCATTAG
r1t22f	TCAAATATGATTAAGAGGAAGCCCCCAAATCA
r1t24h	ATAGTAAGAGCAACACTATCATAACGGATTGC
r1t2e	ACAAAGAACCTGATTATCAGATGAAGATGATG
r1t2f	ATCATATTACCACCAGAAGGAGCGACTTTTTC
r1t4e	CCACGCTGCACCTTGCTGAACCTCTTTGCGGA
r1t4f	TAAAGCATAGAGCCAGCAGCAAATAGTAATTC
r1t6e	ATGGAAATGATTATTTACATTGGCCAACAGTG
r1t6f	CTGAAATGACCTACATTTTGACGCAGAACGCG
r1t8e	TATGGTTGTTAGAATCAGAGCGGGAAACGCTC
r1t8f	TTTCCTCGCTTTGACGAGCACGTATGAACAAA
r3t0g	CATTTCAATTACCTGAGCAAAAGATGGCAATT
r3t10e	AAATCAAACTATCAGGGCGATGGCGGGCGCTG
r3t10f	AAAACCGTAGAATAGCCCGAGATACATTAATG
r3t12e	CTGGGGTGAACGCGCGGGGAGAGGGGCAAAAT
r3t12f	AATCGGCCCCTAATGAGTGAGCTAGCGATTAA
r3t14e	CATTCAGGACGCCAGGGTTTTCCCGAAGCATA
r3t14f	GTTGGGTACTGCGCAACTGTTGGGTAAATGTG
r3t16e	TTGTTAAAACAACCCGTCGGATTCGAAACCAG
r3t20e	CTGCGAACCAATTCTACTAATAGTTTTTTAGA
r3t20f	GGTGGCATGAGTAGATTTAGTTTGCTTCAAAG
r3t22e	TGACTATTACCGGAAGCAAACTCCTTGATTCC
r3t22f	CGAACCAGATAGTCAGAAGCAAAGCCCTCGTT
r3t24h	TACCAGACGACGATAAAAACCAAAATCAGGTC
r3t2e	TAAAAGTTTAATCCTGATTGTTTGTGATTGCT
r3t2f	CATCAATATGAGTAACATTATCATAAATATCA
r3t4e	GGCGGTCAATCAATATCTGGTCAGCCGAACGT
r3t4f	AACCCTCAGTATTAACACCGCCTGAGATTCAC
r3t6e	TACCGCCACGACCAGTAATAAAAGAACAGAGG
r3t6f	CAGTCACAGCCATTGCAACAGGAAAGCTAAAC
r3t8e	CCGCGCTTGATTAAAGGGATTTTATGCTGGTA
r3t8f	AGGAGGCCAATGCGCCGCTACAGGAAGGGCGA
r5t0g	GGGAGAAACAATAACGGATTCGCCGATTATAC
r5t10e	CTGTTTGACACCCAAATCAAGTTTGGAAGGGA
r5t10f	TGAACCATTGGTGGTTCCGAAATCCGGTTTGC
r5t12e	CAATTCCACGCCAGGGTGGTTTTTCGAAAATC
r5t12f	GTATTGGGCACAACATACGAGCCGAGTCACGA
r5t14e	TTTCCGGCAACGACGGCCAGTGCCTCCGCTCA
r5t14f	CGTTGTAAACCGCTTCTGGTGCCGTCCGTGGG
r5t16e	TTGTATAAGCGGATTGACCGTAATCAGCCAGC
r5t16f	AACAAACGGCAAATATTTAAATTGAGATCTAC
r5t18e	TTTATTTCTCAGGTCATTGCCTGACAGGAAGA
r5t18f	AAAGGCTAAACGCAAGGATAAAAAAGTAGCAT
r5t20e	TGGAAGTTAATAAATCATACAGGCGAGAAGCC
r5t20f	TAACATCCTCATTCCATATAACAGAACAGGTC
r5t22e	AACGAGAAAGAGTACCTTTAATTGACGGTGTC
r5t22f	AGGATTAGTGACCATAAATCAAAAATAGCGAG
r5t24h	AGGCTTTTGCAAAAGAAGTTTTGCGTTCAGAA
r5t2e	GACAACTCAATGGAAGGGTTAGAATTTACATC
r5t2f	TTCTGAATGTATTAAATCCTTTGCTTGGCAAA
r5t4e	GAACGAACTGAAAGGAATTGAGGAAACAATTC
r5t4f	TCAACAGTCACCAGCAGAAGATAAGGACATTC
r5t6e	TAGAAGAAAGAGATAGAACCCTTCAAAATACC
r5t6f	TGGCCAACCTCAAACTATCGGCCTGACAGGAA
r5t8e	AGAAAGCGCAGAATCCTGAGAAGTTGCCTGAG
r5t8f	CGGTACGCAAAGGAGCGGGCGCTACCACTACG
r7t0g	GATGAATATACAGTAACAGTACCTCCTACCAT
r7t10e	TCCACGCTCGTAAAGCACTAAATCTGACGGGG
r7t10f	CGAGGTGCGGTTTGCCCCAGCAGGCTTTTCAC
r7t12e	GCTGTTTCCGGGCAACAGCTGATTGCAAGCGG
r7t12f	CAGTGAGACTGTGTGAAATTGTTAAAGCTTGC
r7t14e	ATCGGCCTAGGTCGACTCTAGAGGTGGTCATA
r7t14f	ATGCCTGCCAGGAAGATCGCACTCGGGATAGG
r7t16e	TGATAATCGTGTAGATGGGCGCATACGACAGT
r7t16f	TCACGTTGAGAAAAGCCCCAAAAAGAGTCTGG
r7t18e	TATGACCCAGAGAATCGATGAACGACCCCGGT
r7t18f	AGCAAACATGTAATACTTTTGCGGAAGGCAAA
r7t20e	ATGTTTTAAAAATTAAGCAATAAAAAAAACAT
r7t20f	GAATTAGCAATATGCAACTAAAGTCTCCTTTT
r7t22e	AATCCCCCGTCATTTTTGCGGATGAGCTCAAC
r7t22f	GATAAGAGTCAAATGCTTTAAACACAGAGGGG
r7t24h	GTAATAGTAAAATGTTTAGACTGGATTCATTG
r7t2e	GATTTAGATATTTGCACGTAAAACTAACGTCA
r7t2f	ATCAAAATAGTATTAGACTTTACAAGGTTATC
r7t4e	GATAGCCCCTTTAGGAGCACTAACCATTTGAG
r7t4f	TAAAATATTAAAACATCGCCATTATGACCTGA
r7t6e	TCTTTGATGAATACGTGGCACAGACGCGAACT
r7t6f	AAGCGTAATAGTAATAACATCACTGTTTTTAT
r7t8e	AAAGCCGGAGGCCACCGAGTAAAAGCAATACT
r7t8f	AATCAGTGCGAACGTGGCGAGAAATTTGGGGT
r-7t20f	AAGACAGCTCGAAATCCGCGACCTACGGTCAA
r-7t22e	TTAATCATGAACCGAACTGACCAACGCCTGAT
r-7t22f	TCATAAGGTGTGAATTACCTTATGGGACGTTG
r-7t24h	GGAAGAAAAATCTACGTTAATAAATTTCAACT
r-7t2e	GAATCATAGAATTTATCAAAATCATCGCTATT
r-7t2f	TCAATAGTATTACTAGAAAAAGCCACCAGTAT
r-7t4e	ATCCTAATCGCTCAACAGTAGGGCAAACACCG
r-7t4f	AAAGCCAATTACGAGCATGTAGAATTCCAAGA
r-7t6e	CTTACCAATAAACCAAGTACCGCAAATATCCC
r-7t6f	ACGGGTATCGCTAACGAGCGTCTTCAGCCATA
r-7t8e	TATCTTACCCCAATCCAAATAAGATCCTGAAT
r-7t8f	TTATTTATCGAAGCCCTTTTTAAGGAAGGAAA
r-1t0g	TTTAACAATTTCATTTGAATTACCAAATCCAA
r-1t10e	TCACAATCACACCACGGAATAAGTTCCACTAT
r-1t10f	ACGCAAAGAATAGAAAATTCATATCTTTAGCG
r-1t12e	GGTCATAGTAGCGCGTTTTCATCGCTTTCCAG
r-1t12f	TCAGACTGCCCCCTTATTAGCGTTCAGTCTCT
r-1t14e	CTTTTGATCGTTCCAGTAAGCGTCCGCCAGCT
r-1t14f	GAATTTACGATACAGGAGTGTACTAGGTTTAG
r-1t20f	TGCCACTAAGAATACACTAAAACAAGTAATCT
r-1t22e	ACGTAACAACCGGATATTCATTACGAAAGACT
r-1t22f	TGACAAGAAAGCTGCTCATTCAGTAACTAATG
r-1t24h	CAGATACATAACGCCAAAAGGAATTACGAGGC
r-1t2e	AAATATATCAAAGAACGCGAGAAAGAATTATC
r-1t2f	TCGCAAGATTTAGTTAATTTCATCATAAGAGA
r-1t4e	TGTCCAGATACCGACAAAAGGTAAGAAAAATC
r-1t4f	ATATAAAGCGACGACAATAAACAAAATCAGAT
r-1t6e	AGGCGTTTCTTATCCGGTATTCTATCAATCGT
r-1t6f	ATAGAAGGTAGCGAACCTCCCGACGCGCATTA
r-1t8e	GTCAGAGGAATTAACTGAACACCCTAACGTGC
r-1t8f	GACGGGAGGTAATTGAGCGCTAATTAAAAGAA
r-3t0g	TGGAAACAGTACATAAATCAATATCTTAGGTT
r-3t20f	GGAAGTTTAGCGATTATACCAAGCCCAGGCGC
r-3t22e	CTTGCCCTGCTGACCTTCATCAAGCTCATCTT
r-3t22f	ATAGGCTGGACGAGAAACACCAGAAAAGATTC
r-3t24h	TTGAGATTTAGGAATACCACATTCGAATAAGG
r-3t2e	TAAATTTAATGTAAATGCTGATGCTTTTTTAA
r-3t2f	ATAACTATATGGTTTGAAATACCGACATGTAA
r-3t4e	GCTAATGCCATTTTCGAGCCAGTATTCTGACC
r-3t4f	GGCAGAGGAGAACGCGCCTGTTTACATCGTAG
r-5t0g	GAATAACCTTGCTTCTGTAAATCGTAGGTCTG
r-5t10f	AAGAACTGAGGGAAGGTAAATATTTAGCAAGG
r-5t12f	CCGGAAACACCGCCTCCCTCAGAGGGCAGGTC
r-5t14f	AGACGATTAAACAGTTAATGCCCCACCAGGCG
r-5t16f	GATAAGTGTGAGTTTCGTCACCAGTTTCTGTA
r-5t18f	TGGGATTTATCGCCCACGCATAACCAACGGCT
r-5t20e	AAGTACAATTTGAGGACTAAAGACGACAATGA
r-5t20f	ACAGAGGCCGGAGATTTGTATCATCTTTGAAA
r-5t22e	TAAATTGGATGAACGGTGTACAGAGCGAAACA
r-5t22f	GAGGACAGGCTTGAGATGGTTTAAACGAACTA
r-5t24h	ACGGAACAACATTATTACAGGTAGACGAGTAG
r-5t2e	ATAAATAACCTTTTAACCTCCGGATGTGAGT
r-5t2f	AGAGACTAGGCGTTAAATAAGAATTTAATTGA
r-5t4e	AGATAAGTATATTTAACAACGCCAACCGTGTG
r-5t4f	GAATCGCCCCTGAACAAGAAAAATCTCATCGA
r-5t6f	GAACAAGCCCCAGCTACAATTTTAAACGATTT
r-5t8f	TTTGTTTAAAGAAACAATGAAATAATACCCAA
r-7t0g	AATTAATTTTCCCTTAGAATCCTTGAGAAGAG
r-7t10e	TTATTCATACGCAATAATAACGGAGCAATAGC
r-7t10f	CCGAGGAATAAAGGTGAATTATCAGCCAGCAA
r-7t12e	CTCAGAACGTAGCACCATTACCATGACGGAAA
r-7t12f	AATCACCACGCCACCCTCAGAGCCCAGAGCCG
r-7t14e	TTCGGAACATTGACAGGAGGTTGACCGCCACC
r-7t14f	CCGCCAGCCTATTATTCTGAAACGAAGGATT
r-7t16e	ACAACGCCCGGGGTTTTGCTCAGTCTGCCTAT
r-7t16f	AGGATTAGTGTAGCATTCCACAGATTTGTCGT
r-7t18e	TCGGTCGCACGTTAGTAAATGAATTACAAACT
r-7t18f	CTTTCCAGTGAGGCTTGCAGGGAGAGCAGCGA
r-7t20e	AAATTGTGATCGGAACGAGGGTAGCGATATAT

Right edge (from top to bottom)

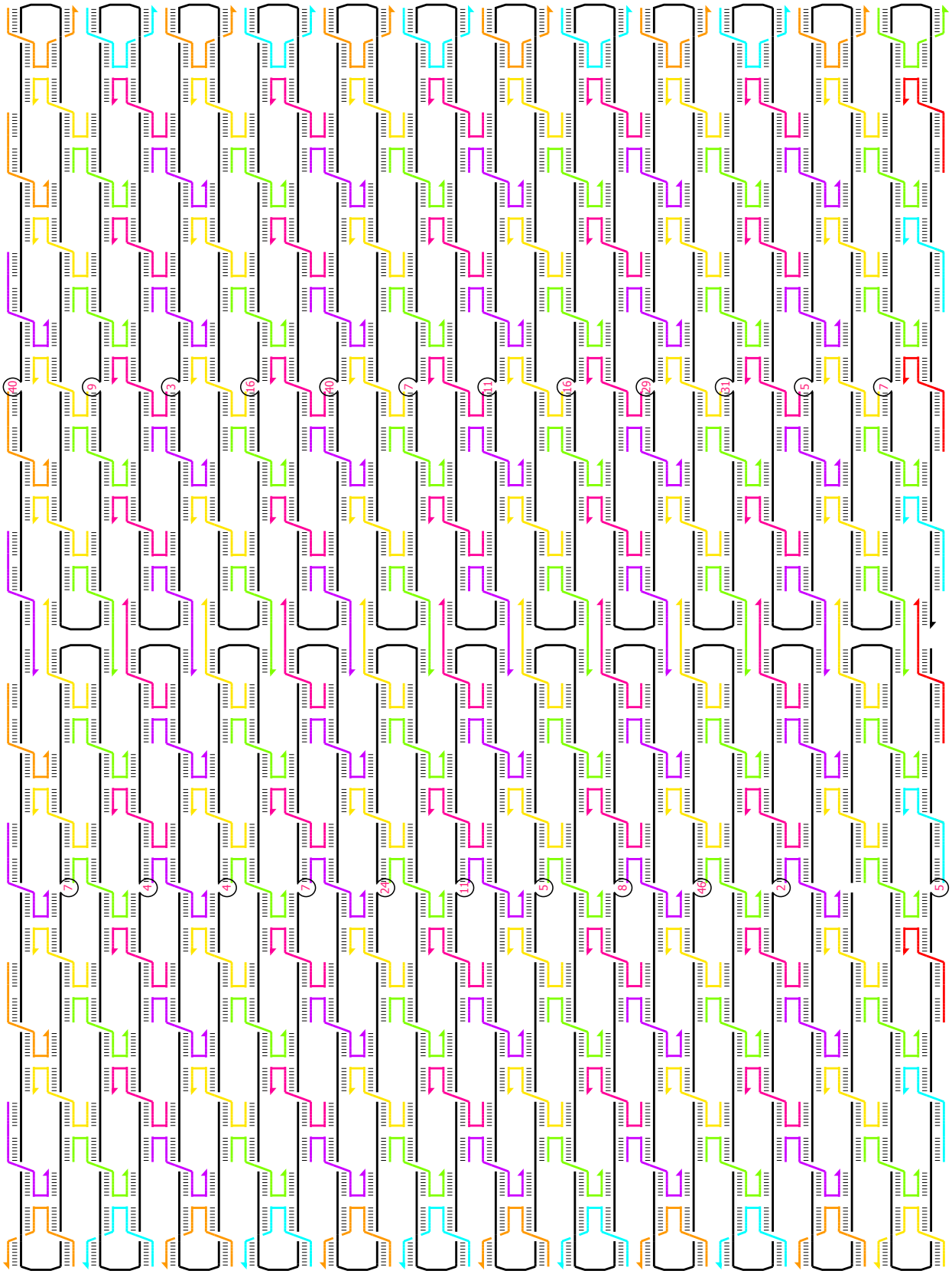
r9t0y	CGTAGATTTTCAGGTTAGAAATAAAGAAATTG
r9t2y	CCGTCAATAGATAATAAACTAATAGATTAGAG
r9t4y	CTATTAGTCTTTAATGCAATATTTTTGAATGG
r9t6y	CAAATTAACCGTTGTAGAGTCTGTCCATCACG
r9t8y	CCCCCGATTTAGAGCTGGAACCCTAAAGGGAG
r9t10y	CCCTGAGAGAGTTGCAGCCCTTCACCGCCTGG
r9t12y	CTCGAATTCGTAATCAATCCCCGGGTACCGAG
r9t14y	CCAGTTTGAGGGGACGCGTAACCGTGCATCTG
r9t16y	CATGTCAATCATATGTGTAATCGTAAAACTAG
r9t18y	CTAAATCGGTTGTACCGCCTCAGAGCATAAAG
r9t20y	CTGAATATAATGCTGTGCTTAGAGCTTAATTG
r9t22x	CGGAATCGTCATAAATATAGCGTCCAATACTG

Left edge (from top to bottom)

r-9t2t	CTTAGATTAAGACGCTGAAAACATAGCGATAG
r-9t4t	CGTTATACAAATTCTTTGTTTAGTATCATATG
r-9t6t	CTGTCTTTCCTTATCAACCAATCAATAATCGG
r-9t8t	CCAGTTACAAAATAAATCCAGAGCCTAATTTG
r-9t10t	CCGAACAAAGTTACCAAAAAGTAAGCAGATAG
r-9t12t	CCATTTGGGAATTAGACCGTCACCGACTTGAG
r-9t14t	CCACCAGAACCACCACACCACCCTCAGAGCCG
r-9t16t	CTGAGACTCCTCAAGATGAAAGTATTAAGAGG
r-9t18t	CGTAACGATCTAAAGTCAGCCCTCATAGTTAG
r-9t20t	CGGGATCGTCACCCTCTTAAAGGCCGCTTTTG
r-9t22t	CCGGAACGAGGCGCAGGCTCCATGTTACTTAG
r-9t24s	CTCATTATACCAGTCACGATTTTAAGAACTGG

Diagram for rectangle with 10.67 bp/turn

with positions and lengths of loopouts (black circles with numbers) indicated



Sequence diagram for rectangle with 10.67 bp/turn

S5.3. Tall rectangle

Core

Seq name	Sequence
t0r11ml1	AGATTTAGCGCCAAAAGGAATTACCCCCCTCA
t0r11mr1	GTAATTGAATTGAGTTAAGCCCAAGAAACCGA
t0r11mr2	AAACGATTAATCTTACCAACGCTACCTCCCGA
t0r11mr_fr	AAACAGGGCAATGAAATAGCAATTAAGCAG
t0r11seam_l	TGCCCTGAGCTGCTCATTCAGTGAAGATAACC
t0r11seam_r	CAGCCATTAATTTGCCAGTTACTAATGCAG
t0r13ml1	AATGCTTTCAAAAATCAGGTCTTTCCTTTTGA
t0r13mr1	CCAGAGCCATTATTTATCCCAATCAGAATTAA
t0r13mr2	CTTGCGGGTTTCATCGTAGGAATCGCTGTCTT
t0r13mr_fr	TTATCCTGTTTTGTTTAACGTCAAACATAA
t0r13seam_l	ATACATAAGAATACCACATTCAACAAAATAAA
t0r13seam_r	AGAACGCTAGAAGGCTTATCCGGAGAATGA
t0r15ml1	TAAGAGGTGAATATAATGCTGTAGCTAATAGT
t0r15mr1	ATCAGATAGAGGCGTTTTAGCGAAACGAGCGT
t0r15mr2	TCCTTATCAAGGTAAAGTAATTCTCATATTTA
t0r15mr_fr	GTTTTTATAGGTTTTGAAGCCTTTACAATT
t0r15seam_l	CCATAAATAAACAGTTCAGAAAACGTATTCTA
t0r15seam_r	ATGTAGAATATCCCATCCTAATTTAGAGCT
t0r17ml1	AGTAGCATGAATTAGCAAAATTAAAAAGGGT
t0r17mr1	GAAAAATAAACCAATCAATAATCGATTACCGC
t0r17mr2	ACAACGCCTTACTAGAAAAAGCCTTGACCTAA
t0r17mr_fr	ACCGACAAATTCCAAGAACGGGTGCAAGCC
t0r17seam_l	TAATTGCTCATTTTTGCGGATGGCTTACGAGC
t0r17seam_r	AGTAGGGCCAGTATAAAGCCAAATACAGGC
t0r19ml1	GAGAAAGGTTCAACCGTTCTAGCTTAAATTGT
t0r19mr1	AATTCTTACTTAATTGAGAATCGCGTCCAGAC
t0r19mr2	ATTTAATGTCATAGGTCTGAGAGATTTCCCTT
t0r19mr_fr	AATCATAAAACATGTAATTTAGGATAAAGT
t0r19seam_l	AAGGCAAATAACATCCAATAAATCCGCTCAAC
t0r19seam_r	ATATTTTAACGCGAGAAAACTTCACCATCA
t0r1ml1	AGAGGGTTCGTACTCAGGAGGTTTTTCCAGAC
t0r1ml_fl	TTGCTCAGAGAACCGCCACCCTCAAAACTACA
t0r1mr2	ACCACCAGAGAGCCACCACCGGAACGCGTTTT
t0r1seam_r	AGCCACCACCCTCAGAACCGCCGGAATAGG
t0r1t_seam	CGCAGTCTCTGAATTTACCGTTCCAGTAAGC
t0r1tl1	GTCATACATGGCTTTTGATGATACTGCCGTCG
t0r1tl2	ACTGGTAATAAGTTTTAACGGGGGAGACTCC
t0r1tr1	ATTCACAAACAAATAAATCCTCATCCAGAACC
t0r21mr1	AGACAAAGAGTTAATTTCATCTTCGTTTAGTA
t0r21mr_fr	TATCAAAAGTTTGAAATACCGACACACCGG
t0r21seam_l	ATATGATACCGGAGACAGTCAAATTTTCAAAT
t0r23mr_fr	ATCAAGAATGAAAACATAGCGATGTGAATT
t0r25ml1	CCAGTCACGCATGCCTGCAGGTCGCGCTCACT
t0r25ml_fl	GGCGATTACGGGTACCGAGCTCGACTGGGGTG
t0r25mr2	TAATGGAAAACATTATCATTTTGCAGTATTAG
t0r25mr_fr	TTGCGTAGAGAGGCGAATTATTCAACAAAC
t0r25seam_r	TAATCCTCAGATGATGGCAATTGGCCAGTG
t0r27ml1	GCCCGCTTTAATGAATCGGCCAACCCGAAATC
t0r27ml_fl	GCTAACTCGTTTGCGTATTGGGCGGGTTTGCC
t0r27mr2	ACTTTACAATCTGGTCAGTTGGCACCACGCTG
t0r27mr_fr	GTTTGAGTGGGTTAGAACCTACCAAAGAAA
t0r27seam_r	GATAATACTAATAGATTAGAGCGTCGTGCC
t0r29ml1	GGCAAAATGGTTGAGTGTTGTTCCAAGCCGGC
t0r29ml2	AGAGGCGACATTAATTGCGTTGACTCTAGA
t0r29mr1	ACTAACAACATTTGAGGATTTAGAGGAACAAA
t0r29mr2	AGAGCCAGCTAAAACATCGCCATTGACCTGAA
t0r29mr_fr	CAATCAATAACAATTCGACAACTTTTAAAA
t0r29seam_l	AGCTGCATTCCAGTCGGGAAACCTCGTCAATA
t0r29seam_r	TATTAACTAAAACAGAGGTGAGGAATAGCC
t0r31ml2	ACAAGAGTGTTTGATGGTGGTTGCGCGGGG
t0r31mr1	GCAGAAGAACCGCCTGCAACAGTGAATCAACA
t0r31mr_fr	TGATAGCCCAGCAAATGAAAAATAAACCCT
t0r31br2	AGCGTAAGAATACGTGGCACAGACGCGAAC
t0r31seam_l	CGAGATAGCCCTTATAAATCAAAAGCGGTCAG
t0r31b_seam	GAACGTGGCGAGAAAGGAAGGGAAACATTCTG
t0r31bl1	CCCGATTTAGAGCTTGACGGGGAAGTTTGGA
t0r31br1	GCCAACAGAGATAGAACCCTTCTAAAAATAC
t0r3ml1	GTTAGTAACTTTCAACAGTTTCAGACAATGAC
t0r3ml2	CACCCTCTACCAGGCGGATAAGAGGAGTGT
t0r3ml_fl	AACGATCTAAGGAACAACTAAAGGGGTGAATT
t0r3mr1	GAGCCGCCACCCTCAGAGCCGCCATAAAGCCA
t0r3mr2	CATCGGCAAAGGCCGGAAACGTCATGAATTAT
t0r3mr_fr	CCGGAACCAGCCGCCGCCAGCATCCTTGAT
t0r3seam_l	TGTATCACGATATAAGTATAGCCCACCCTCAG
t0r3seam_r	TTGCCTTAATCAGTAGCGACAGGGATTTTG
t0r5ml1	AACAACCAGAGGCTTGCAGGGAGTAAGAATAC
t0r5ml2	GAATAGAAAAGTTTTGTCGTCTAGTACCGC
t0r5mr1	AGCACCGTTAGCGTCAGACTGTAGCCGCCTCC
t0r5mr2	CACCGTCAACAATCAATAGAAAATAAACGTAG
t0r5mr_fr	CCATTAGCTTTTCGGTCATAGCCAAAATCA
t0r5seam_l	CTAAACAAATGAATTTTTCTGTATGAATCAAGT
t0r5seam_r	TGACGGAATTGAGGGAGGGAAGGATATATT
t0r7ml1	ACTAAAACCGCGAAACAAAGTACAAAATCAAC
t0r7ml2	GCTTTTGCCGATAGTTGCGCCGCGGAGTGA
t0r7mr1	TTCAACCGAATTATTCATTAAAGGCCAATGAA
t0r7mr2	AAAATACAACAAAGTTACCAGAAGTAATAAGA
t0r7mr_fr	ATTTTGTCCCGACTTGAGCCATTACCATTA
t0r7seam_l	CGGTCGCTTCGCCCACGCATAACCGTAAATAT
t0r7seam_r	TCCTTATAAAAGAACTGGCATGCAGCGATT
t0r9ml1	GTAACAAACGAGAAACACCAGAACATCAGTTG
t0r9mr1	GAATACCCTACGCAGTATGTTAGCTCATATGG
t0r9mr2	GCAAGAAAAAGCGCATTAGACGGGCAAATAAG
t0r9mr_fr	ATAGCCGATACATAAAAGGTGGCATAAGTTT
t0r9seam_l	ATACCAAGACTCATCTTTGACCCCATTAAGAC
t0r9seam_r	CACAAGAGCGCTAATATCAGAGATAAGGCT
t-1r0tl4	TTGAGTAACAGTGCCCGTATAAATTATTCT
t-1r10fl2	CGAACTGAGTGAATTACCTTATGGGACGTT
t-1r12fl2	GGGAAGAACGATAAAAACCAAAAAGAGGGG
t-1r14fl2	GTAATAGTTAAGAGGAAGCCCGATCAAAGC
t-1r16fl2	GAACCAGACATTCCATATAACAGGTTTGAC
t-1r18fl2	CATTAGATTTATGACCCTGTAATAGGATAA
t-1r20fl2	AAATTTTTGGCTATCAGGTCATTTGAACGG
t-1r22fl2	TAATCGTAGTAGCCAGCTTTCATGGATTCT
t-1r24fl2	CCGTGGGAGCCATTCAGGCTGCGCTTCGCT
t-1r26fl1	TCATGGTCAGCTGGCGAAAGGGGGAGGCAAAG
t-1r26fl2	ATTACGCCATAGCTGTTTCCTGTACATACG
t-1r28fl1	GTTTTTCTGCATAAAGTGTAAAGCATTCGTAA
t-1r28fl2	AGCCGGAATTTCACCAGTGAGACCCTGAGA
t-1r2fl1	ACCCTCAGAAAGTATTAAGAGGCTTCAGTGCC
t-1r2fl2	GAAACATGAGCCACCACCCTCATGTACCGT
t-1r30fl1	CAACGTCAGCAAGCGGTCCACGCTCCAGGGTG
t-1r30fl2	GAGTTGCAAAGGGCGAAAAACCGTCACCCA
t-1r32bl2	AATCAAGTTTTTTGGGGTCGAGGTGTGGACTC
t-1r4fl1	ATAATAATGTTTCGTCACCAGTACGAACCGCC
t-1r4fl2	AACACTGATTTTTCACGTTGAAATAATTGT
t-1r6fl1	AAGACAGCATCAGCTTGCTTTCGAAATTGCGA
t-1r6fl2	ATCGGTTTATCGGAACGAGGGTACTTTTTC
t-1r8fl2	ATGAGGAACTGCTCCATGTTACTAGGGAAC
tr-rem1	GTAAAAGAGTCTGTCCATCACGCAAATT
tr-rem2	AGTGTTTTTATAATCAGTGAGGCCACCGA
t0r29special_l1	GAAAATCCTCCACTATTAAAGAACGCCGTAAACCCGATTT

Right edge (from top to bottom)

t1r0_edge_r_2	CAGGTCAGACGATTGGTGACAGGAGGTTGAGG
t1r2_edge_r_2	CCATCTTTTCATAATCCCCTTATTAGCGTTTG
t1r4_edge_r_2	CAAAATCACCAGTAGCTGGGAATTAGAGCCAG
t1r6_edge_r_2	CAAAGACACCACGGAAACATATAAAAGAAACG
t1r8_edge_r_2	CCCTTTTTAAGAAAAGAGCTATCTTACCGAAG
t1r10_edge_r_2	CCTTTACAGAGAGAATAAAATGAAAATAGCAG
t1r12_edge_r_2	CTATTTTGCACCCAGCAAATCAAGATTAGTTG
t1r14_edge_r_2	CACTCATCGAGAACAAATTAAACCAAGTACCG
t1r16_edge_r_2	CCAGTAATAAGAGAATCAGAGGCATTTTCGAG
t1r18_edge_r_2	CGTTAAATAAGAATAACGTGTGATAAATAAGG
t1r20_edge_r_2	CTGAGAAGAGTCAATAAGCTTAGATTAAGACG
t1r22_edge_r_2	CAAAAGAAGATGATGAATTTCAATTACCTGAG
t1r24_edge_r_2	CACGTAAAACAGAAATATATCAAAATTATTTG
t1r26_edge_r_2	CCCGAACGTTATTAATCGTATTAAATCCTTTG
t1r28_edge_r_2	CTGAACCTCAAATATCCTAAAGCATCACCTTG
t1r30_edge_r_2	CTATTAGTCTTTAATGCAATATTTTTGAATGG

Left edge (from top to bottom)

t-1r2_edge_l_2	CCTATTTCGGAACCTACAGTTAATGCCCCCTG
t-1r4_edge_l_2	CCCAATAGGAACCCATTTTCAGGGATAGCAAG
t-1r6_edge_l_2	CTCCAAAAGGAGCCTTATCTCCAAAAAAAAGG
t-1r8_edge_l_2	CTTTGAGGACTAAAGAGCAACGGCTACAGAGG
t-1r10_edge_l_2	CAGACGGTCAATCATATAGCCGGAACGAGGCG
t-1r12_edge_l_2	CTCATTATACCAGTCACGATTTTAAGAACTGG
t-1r14_edge_l_2	CAAAAGAAGTTTTGCCTAGCGAGAGGCTTTTG
t-1r16_edge_l_2	CGTTTTAATTCGAGCTAAGACTTCAAATATCG
t-1r18_edge_l_2	CGAACGAGTAGATTTATTGATTCCCAATTCTG
t-1r20_edge_l_2	CCTTTATTTCAACGCAACTTTTGCGGGAGAAG
t-1r22_edge_l_2	CAAACAAGAGAATCGAGCCTGAGAGTCTGGAG
t-1r24_edge_l_2	CGAGTAACAACCCGTCCAACATTAAATGTGAG
t-1r26_edge_l_2	CGATCGGTGCGGGCCTCAACTGTTGGGAAGGG
t-1r28_edge_l_2	CTCACAATTCCACACAGTGAAATTGTTATCCG
t-1r30_edge_l_2	CCCTTCACCGCCTGGCGGGCAACAGCTGATTG
t-1r32_edge_l_2	CCCACTACGTGAACCATCTATCAGGGCGATGG

Hairpin-labeled staples (hairpin sequence in lowercase)

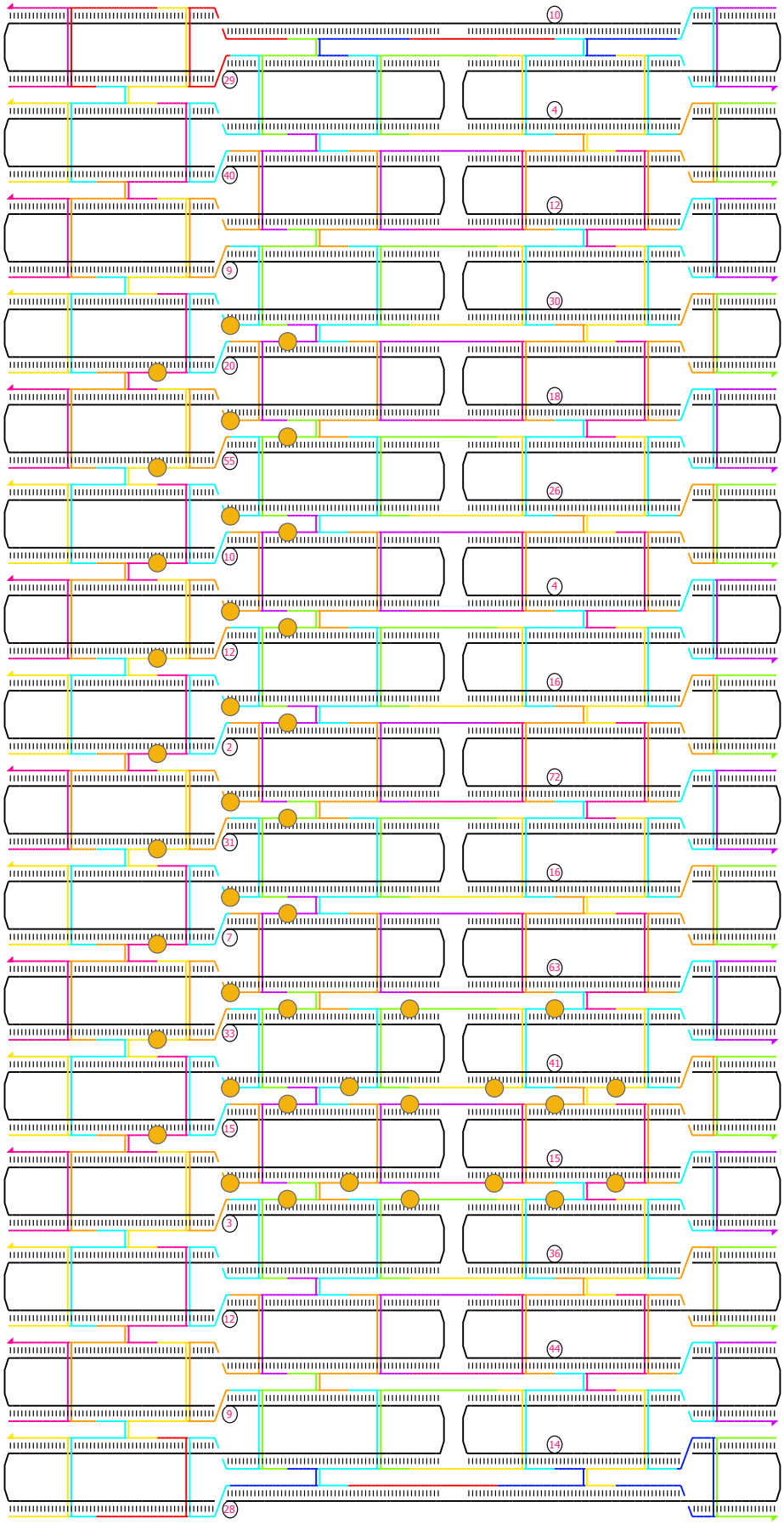
t-1r8fl1_hp	TCCGCGACGTTTCCATtcctcttttgaggaacaagttttcttgtTAAACGGGAGCAGCGA
t-1r10fl1_hp	TAATCATTCCAACTTTtcctcttttgaggaacaagttttcttgtGAAAGAGGTGTCGAAA
t-1r12fl1_hp	CCAGACGAAAATCTACtcctcttttgaggaacaagttttcttgtGTTAATAATTCAACTT
t-1r14fl1_hp	AAAAAGATAAAATGTTtcctcttttgaggaacaagttttcttgtTAGACTGGCTCGTTTA
t0r5ml_fl_hp	GCTTGATACGGGATCGtcctcttttgaggaacaagttttcttgtTCACCCTCTAAAATAC
t0r7ml_fl_hp	CCAACCTAATCGCCTGtcctcttttgaggaacaagttttcttgtATAAATTGACAGATGA
t0r9ml_fl_hp	CAAGAACCCTTGAGATtcctcttttgaggaacaagttttcttgtGGTTTAATAACGAACT
t0r11ml_fl_hp	CATTATTACAACACTAtcctcttttgaggaacaagttttcttgtTCATAACCATAGCGTC
t0r13ml_fl_hp	ATCGTCATGTCAGAAGtcctcttttgaggaacaagttttcttgtCAAAGCGGACAGGTCA
t0r9ml2_hp	TTGTATCAAACGAAAtcctcttttgaggaacaagttttcttgtGAGGCAATAAAGGCC
t0r11ml2_hp	AATTGGGGGATATTCtcctcttttgaggaacaagttttcttgtATTACCCACGGAGAT
t0r13ml2_hp	GTAAGAGCAGGTAGAtcctcttttgaggaacaagttttcttgtAAGATTCGAGTAGTA
t0r15ml2_hp	TATTATAAAATATTCtcctcttttgaggaacaagttttcttgtATTGAATGAGGCATA
t-1r16fl1_hp	GGAAGTTTCCGGAAGCtcctcttttgaggaacaagttttcttgtAAACTCCAATTGCATC
t-1r18fl1_hp	CAAAAACAACATTTCGtcctcttttgaggaacaagttttcttgtCAAATGGTCGGTGTCT
t-1r20fl1_hp	TCTACAAAAGAACCCTtcctcttttgaggaacaagttttcttgtCATATATTGGTTGTAC
t-1r22fl1_hp	GCCTTCCTAAACTAGCtcctcttttgaggaacaagttttcttgtATGTCAATTTGAGAGA
t-1r24fl1_hp	CGCCATTCACAAACGGtcctcttttgaggaacaagttttcttgtCGGATTGACGCGTCTG
t0r15ml_fl_hp	ATTAGAGAATATGCAAtcctcttttgaggaacaagttttcttgtCTAAAGTACAATAACC
t0r17ml_fl_hp	CTGAAAAGAGCATAAAtcctcttttgaggaacaagttttcttgtGCTAAATCTTAAATGC
t0r19ml_fl_hp	GAGTAATGGAGAGGGTtcctcttttgaggaacaagttttcttgtAGCTATTTCATATGTA
t0r21ml_fl_hp	GGAAGATTGCCATCAAtcctcttttgaggaacaagttttcttgtAAATAATTCCGTAATG
t0r17ml2_hp	GTTTTAAGTACCTTTtcctcttttgaggaacaagttttcttgtAATTGCTACCCTGAC
t0r19ml2_hp	GCCTCAGGTGGCATCtcctcttttgaggaacaagttttcttgtAATTCTACTCAACAT
t0r21ml2_hp	AATGCCGTGTAGGTAtcctcttttgaggaacaagttttcttgtAAGATTCGCAATAAA
t0r23ml2_hp	TAGGAACGTATAAGCtcctcttttgaggaacaagttttcttgtAAATATTGATAAATT
t0r23ml_fl_hp	GTGTAGATTCTGGTGCtcctcttttgaggaacaagttttcttgtCGGAAACCATGTGCTG
t0r25ml2_hp	CACCGCTGGGCGCATtcctcttttgaggaacaagttttcttgtCGTAACCTTAACCAA
t0r27ml2_hp	GGATCCCAGTTGGGTtcctcttttgaggaacaagttttcttgtAACGCCACTTTCCGG
t0r21ml1_hp	AAACGTTATAAATCAGtcctcttttgaggaacaagttttcttgtCTCATTTTGTGCATCT
t0r23ml1_hp	GCCAGTTTATCGCACTtcctcttttgaggaacaagttttcttgtCCAGCCAGGGGTTTTC
t0r23seam_l_hp	ATTTTTGTATATTTTGtcctcttttgaggaacaagttttcttgtTTAAAATTTTGCTTCT
t0r25seam_l_hp	TCAGGAAGGAGGGGACtcctcttttgaggaacaagttttcttgtGACGACAGTAACGGAT
t0r27seam_l_hp	CCAAGCTTGACGTTGTtcctcttttgaggaacaagttttcttgtAAAACGACCATCAATA
t0r21seam_r_hp	GTAAATCTGTGAGTGtcctcttttgaggaacaagttttcttgtAATAACCCGCATTAA
t0r23seam_r_hp	TCGCCTGCATCGGGAtcctcttttgaggaacaagttttcttgtGAAACAATATCGGCC
t0r23mr1_hp	TCAATATAGTCGCTATtcctcttttgaggaacaagttttcttgtTAATTAATCTACCTTT
t0r25mr1_hp	ACCTTTTAATTGCTTTtcctcttttgaggaacaagttttcttgtGAATACCATTTAACAA
t0r27mr1_hp	CTGATTATGATTGTTTtcctcttttgaggaacaagttttcttgtGGATTATATCAGATGA
t0r21mr2_hp	AGAATCCTAACAAAATtcctcttttgaggaacaagttttcttgtTAATTACAAGTTACAA
t0r23mr2_hp	AATCGCGCATTTTCAGtcctcttttgaggaacaagttttcttgtGTTTAACGCTTCTGAA

Hairpin-less staples corresponding to hairpin-labeled ones (needed when generating hairpin patterns other than 'L')

t-1r8fl1_hp_org	TCCGCGACGTTTCCATTAAACGGGAGCAGCGA
t-1r10fl1_hp_org	TAATCATTCCAACTTTGAAAGAGGTGTCGAAA
t-1r12fl1_hp_org	CCAGACGAAAATCTACGTTAATAATTCAACTT
t-1r14fl1_hp_org	AAAAAGATAAAATGTTTAGACTGGCTCGTTTA
t0r5ml_fl_hp_org	GCTTGATACGGGATCGTCACCCTCTAAAATAC
t0r7ml_fl_hp_org	CCAACCTAATCGCCTGATAAATTGACAGATGA
t0r9ml_fl_hp_org	CAAGAACCCTTGAGATTAGTTTAATAACGAACT
t0r11ml_fl_hp_org	CATTATTACAACACTATCATAACCATAGCGTC
t0r13ml_fl_hp_org	ATCGTCATGTCAGAAGCAAAGCGGACAGGTCA
t0r9ml2_hp_org	TTGTATCAAACGAAAGAGGCAATAAAGGCC
t0r11ml2_hp_org	AATTGGGGGATATTCATTACCCACGGAGAT
t0r13ml2_hp_org	GTAAGAGCAGGTAGAAGATTCGAGTAGTA
t0r15ml2_hp_org	TATTATAAAATATTCATTGAATGAGGCATA
t-1r16fl1_hp_org	GGAAGTTTCCGGAAGCAAACTCCAATTGCATC
t-1r18fl1_hp_org	CAAAAACAACATTTCGCAAATGGTCGGTGTCT
t-1r20fl1_hp_org	TCTACAAAAGAACCCTCATATATTGGTTGTAC
t-1r22fl1_hp_org	GCCTTCCTAAACTAGCATGTCAATTTGAGAGA
t-1r24fl1_hp_org	CGCCATTCACAAACGGCGGATTGACGCGTCTG
t0r15ml_fl_hp_org	ATTAGAGAATATGCAACTAAAGTACAATAACC
t0r17ml_fl_hp_org	CTGAAAAGAGCATAAAGCTAAATCTTAAATGC
t0r19ml_fl_hp_org	GAGTAATGGAGAGGGTAGCTATTTCATATGTA
t0r21ml_fl_hp_org	GGAAGATTGCCATCAAATTAATTCCGTAATG
t0r17ml2_hp_org	GTTTTAAGTACCTTTAATTGCTACCCTGAC
t0r19ml2_hp_org	GCCTCAGGTGGCATCAATTCTACTCAACAT
t0r21ml2_hp_org	AATGCCGTGTAGGTAAAGATTCGCAATAAA
t0r23ml2_hp_org	TAGGAACGTATAAGCAAATATTGATAAATT
t0r23ml_fl_hp_org	GTGTAGATTCTGGTGCCGGAAACCATGTGCTG
t0r25ml2_hp_org	CACCGCTGGGCGCATCGTAACCTTAACCAA
t0r27ml2_hp_org	GGATCCCAGTTGGGTAACGCCACTTTCCGG
t0r21ml1_hp_org	AAACGTTATAAATCAGTCGCTATTTGTGCATCT
t0r23ml1_hp_org	GCCAGTTTATCGCACTTCAGCCAGGGGTTTTC
t0r23seam_l_hp_org	ATTTTTGTATATTTTGTTAAAATTTTGCTTCT
t0r25seam_l_hp_org	TCAGGAAGGAGGGGACGACGACAGTAACGGAT
t0r27seam_l_hp_org	CCAAGCTTGACGTTGTAAAACGACCATCAATA
t0r21seam_r_hp_org	GTAAATCTGTGAGTGAATAACCCGCATTAA
t0r23seam_r_hp_org	TCGCCTGCATCGGGAGAAACAATATCGGCC
t0r23mr1_hp_org	TCAATATAGTCGCTATTAATTAATCTACCTTT
t0r25mr1_hp_org	ACCTTTTAATTGCTTTCTGAATACCATTTAACAA
t0r27mr1_hp_org	CTGATTATGATTGTTTCGGATTATATCAGATGA
t0r21mr2_hp_org	AGAATCCTAACAAAATTAATTACAAGTTACAA
t0r23mr2_hp_org	AATCGCGCATTTTCAGGTTTAACGCTTCTGAA

Diagram for tall rectangle

with positions of dumbbells (orange circles) and positions and lengths of loopouts (black circles with numbers) indicated



Sequence diagram for tall rectangle

S5.4. "A" origami

Core

Seq name	Sequence
10[143]-8[144]	AGTACAACCCCACGCATAACCGATACAACTTT
10[175]-8[176]	ACCCCCAGGGCTTGCAGGGAGTTATAAAGTAA
10[431]-8[432]	GCCAGTTAAGATAGCCGAACAAAGGTAGAAAA
10[463]-8[464]	CGCTAACGCGAGGAAACGCAATAAAGACTCCT
11[128]-13[127]	AACCATCGGGAGATTTGTATCATCACTTTGAA
11[416]-13[415]	AAGTAAGCCAAAATAAACAGCCATGGCGTTTT
11[64]-8[80]	GCTTGCTTTCGAGGTGAATTTCTTTAATTTTT
11[96]-13[95]	TGATACCGTCGAAATCCGCGACCTACGGTCAA
12[143]-10[144]	TAACGGAAGATGAACGGTGTACAGCGAAACAA
12[175]-10[176]	GGGAAGAATGGCTGACCTTCATCACATCTTTG
12[431]-10[432]	AACAACATTCCCGACTTGCGGGAGCCTAATTT
12[463]-10[464]	GTAAAGTAATCAAGATTAGTTGCTCTTACCAA
13[128]-15[127]	AGAGGACACAACATTATTACAGGTCTATCATA
13[288]-12[272]	GTACCGCATTCCAAGAACGGGTATCGAGTAGT
13[384]-15[383]	TATCCGGTAACAATAGATAAGTCCATTCTTAC
13[416]-15[415]	AGCGAACCGTTCAGCTAATGCAGAGTAGGGCT
13[96]-15[95]	TCATAAGGTTGAGATTTAGGAATAAAAGGAAT
14[143]-12[144]	GAGCTTCATTACCAGACGACGATAAAACGAAC
14[175]-12[176]	CCGAAAGAAGAGGCTTTTGCAAAAAGGACGTT
14[239]-12[240]	TCTTTACCCGGAATCGTCATAAATTTTCAACT
14[271]-15[287]	AAACGAGACTCAAATGCTTTAAACACCGACCG
14[367]-12[368]	AATCGCAACATATGCGTTATACAATGAACAAG
14[431]-12[432]	TACCTTTTAATCGCCATATTTAACCGACAATA
14[463]-12[464]	TGAATTTATTTAGGCAGAGGCATTGACAAAAG
14[47]-17[31]	GGATGGCTTAGAGCTTAATTGCTGATATGCAA
15[128]-17[127]	ACCCTCGTAAGCGAACCAGACCGGACATTTCG
15[384]-17[383]	CAGTATAAATATGTAAATGCTGATGGAAACAG
15[416]-17[415]	TAATTGAGTAACCTCCGGCTTAGGGAATAACC
15[96]-17[95]	TACGAGGCGGTCAGGATTAGAGAGCGAACGAG
16[143]-14[144]	AGATTCAACAATAACCTGTTTAGCTTTAATTC
16[175]-14[176]	TAAATGCAGGCGCGAGCTGAAAAGGAGGAAGC
16[239]-14[240]	CTTTTGCGAGGCAAAGAATTAGCAAAATCAGG
16[335]-14[336]	CGCCTGATCAAAATTAATTACATTAAAACTTT
16[367]-14[368]	CCTTTTACAATTACCTTTTTTAATGCAAATCC
16[431]-14[432]	ACAGAAATGTAAATCGTCGCTATTTGAGAGAC
16[463]-14[464]	ACCTACCAAGAATCCTTGAAAACAGTCAATAG
16[47]-14[48]	GCTATTTTCGGTGTCTGGAAGTTTATTTTTGC
16[79]-14[80]	TCTAGCTGTTGATTCCCAATTCTGTACCTTTA
17[128]-19[127]	CAAATGGTAAGGGTGAGAAAGGCCTTGTATAA
17[32]-19[31]	CTAAAGTATGAGAGATCTACAAAGCAAACAAG
17[320]-19[319]	CAAGAAAATGCTTTGAATACCAAGAATTCGAC
17[384]-19[383]	TACATAAATCAGATGAATATACAGTTATCATT
17[416]-19[415]	TTGCTTCTAAAGAAATTGCGTAGAAAGGAGCG
17[96]-19[95]	TAGATTTAACCATCAATATGATATGATAATCA
18[143]-16[144]	AGATCGCATTAAATTGTAAACGTTGTAGGTAA
18[175]-16[176]	TGAGGGGATTCGCATTAAATTTTTATATATTT
18[335]-16[336]	TATCTTTATTAAATCCTTTGCCCGAACGGATT
18[367]-16[368]	GTTGAAAGAAAAGTTTGAGTAACATAACAGTA
18[431]-16[432]	TCACCTTGATCATATTCCTGATTACACGTAAA
18[463]-16[464]	AGAGCCAGTCATCAATATAATCCTGGGTTAGA
18[47]-16[48]	GGCGATCGTGAACGGTAATCGTAAAGAGGGTA
18[79]-16[80]	TCGCCATTCATATGTACCCCGGTTTCAACCGT
19[128]-21[127]	GCAAATATCTCCAGCCAGCTTTCCCTCTAGAG
19[288]-18[272]	TAGAAGTACAATAGATAATACATTACAACCCG
19[32]-21[31]	AGAATCGAGTGCGGGCCTCTTCGCCTGCAAGG
19[320]-21[319]	AACTCGTAGGAGCACTAACAACTAACAGAGAT
19[384]-21[383]	TTGCGGAAATCTGGTCAGTTGGCATGGCTATT
19[416]-21[415]	GAATTATCCTGAACCTCAAATATCGCCCTAAA
19[96]-21[95]	GAAAAGCCGCCGGAAACCAGGCAACCAGTGCC
20[143]-18[144]	GCTGATTGGGTACCGAGCTCGAATCCTCAGGA
20[175]-18[176]	GTTTTTCTTAGCTGTTTCCTGTGTTGCCAGTT
20[271]-21[287]	CCCGCTTTACATTAATTGCGTTGCACGACCAG
20[335]-18[336]	TTTGACGCTCTGACCTGAAAGCGTATCTAAAA
20[367]-18[368]	ACAGGAAAAGACAATATTTTTGAAAATCAACA
20[431]-18[432]	GAAGAACTATTAAAAATACCGAACCTAAAGCA
20[463]-18[464]	TTGATTAGGATAAAACAGAGGTGACCACGCTG
20[47]-18[48]	TCCGAAATTTGGGTAACGCCAGGGTTGGGAAG
20[79]-18[80]	CCAGCAGGCGTTGTAAAACGACGGAGCGCCAT
21[128]-23[127]	GATCCCCGCCCTTCACCGCCTGGCTACGTGAA
21[288]-20[272]	TAATAAAACAGATTCACCAGTCACGCTCACTG
21[32]-23[31]	CGATTAAGCGGCAAAATCCCTTATAGTGTTGT
21[320]-23[319]	AGAACCCTTCAATCGTCTGAAATGACTATGGT
21[384]-23[383]	AGTCTTTAACAATATTACCGCCAGAACAGGAG
21[416]-23[415]	ACATCGCCCAAACTATCGGCCTTGGGAACGGT
21[96]-23[95]	AAGCTTGCGCAAGCGGTCCACGCTCGAAAAAC
23[128]-20[144]	CCATCACCCAAATCAAGTTTTTTGGGGCAACA
23[160]-20[176]	GGTGCCGTAAAGCACTAAATCGGACCAGGGTG
23[224]-20[240]	CGAACGTGGCGAGAAAGGAAGGGAGTCGTGCC
23[256]-23[287]	GAAAGGAGCGGGCGCTAGGGCGCTGCCGCGCT
23[32]-20[48]	TCCAGTTTGGAACAAGAGTCCACTATGGTGGT
23[320]-20[336]	TGCTTTGACGAGCACGTATAACGTACCTACAT
23[416]-20[432]	ACGCCAGAATCCTGAGAAGTGTTTCCTGAGTA
23[64]-20[80]	AACGTGGACTCCAACGTCAAAGGGGGTTTGCC
6[143]-9[127]	TTTAACGGGGTCAGTGCCTTGAGTGATATAAG
6[303]-6[272]	TGACAGGAGGTTGAGGCAGGTCAGACGATTGG
6[335]-9[319]	CCAGAACCACCACCAGAGCCGCCGATCACCGT
6[399]-9[383]	CCTCCCTCAGAGCCGCCACCCTCAACCATTAG
8[143]-6[144]	CAACAGTTGGAATAGGTGTATCACTAATAAGT
8[175]-6[176]	TTTTCTGTTAGTACCGCCACCCTCCATGGCTT
8[239]-6[240]	CAGCCCTCGGATAGCAAGCCCAATATCCTCAT
8[271]-9[287]	ACAAACTAGTAACACTGAGTTTCGATTGACGG
8[335]-6[336]	CCAGCGCCTGAGCCATTTGGGAATAGCCGCCA
8[367]-6[368]	GTCACAATCACCAGTAGCACCATTGAACCGCC
8[463]-6[464]	TATTACGCGCCTTTAGCGTCAGACAGCCCCCT
8[79]-6[80]	TCACGTTGGCGGGGTTTTGCTCAGTGCCTATT
9[128]-11[127]	TATAGCCCTCAGCGGAGTGAGAATATGACAAC
9[288]-8[272]	AAATTATTGGGAGGAAGGTAAATTCACCAGT
9[320]-11[319]	CACCGACTAAAGACAAAAGGGCGATAATATCA
9[96]-11[95]	GGATAAGTAAGGAATTGCGAATAAAAACAGCT
23[352]-20[368]	CGTTAGAATCAGAGCGGGAGCTACCATTGCA
23[448]-20[464]	AGTGAGGCCACCGAGTAAAAGAGATACTTCT
23[96]-20[112]	CGTCTATCAGGGCGATGGCCCACCCTGAGAG
6[111]-9[95]	CCGTATAAACAGTTAATGCCCCCTACCAGGC
6[367]-9[351]	ACCCTCAGAGCCACCACCCTCAGTAGAGCCA
10[111]-8[112]	AATTGTGATAGTTGCGCCGACAAGAAAGGA
11[160]-13[159]	GTCGCTGACGATTATACCAAGCGACCAGGC
11[448]-13[447]	AGGAAACAGCGTCTTTCCAGAGGTTTTGAA
12[111]-10[112]	TCATCAGGAACCGAACTGACCAGCCTGATA
12[399]-10[400]	TGTTTATCATTCTAAGAACGCGAATTATTT
13[160]-15[159]	GCATAGGCAAATCTACGTTAATAAAAACCA
13[256]-15[255]	TTGCCCTGCTTGAGATGGTTTAAATTCATT
13[448]-15[447]	GCCTTAAATTCTGTCCAGACGAAACGCCAA
14[111]-12[112]	TCCAACAATAGTAAGAGCAACAAGAAAGAT

Core (continued)

Seq name	Sequence
14[399]-12[400]	ATATAACTAGCCAACGCTCAACAACGCGCC
14[79]-17[63]	ATTGCTCCTTTTGATAAGAGGTCCATTCCA
15[160]-17[159]	AAATAGCGCTTCAAATATCGCGTTATATTT
15[256]-17[255]	GAATCCCCATGACCATAAATCAAAAATTAA
15[352]-17[351]	TTAGTATGACAAAGAACGCGAGTAACAATT
15[448]-17[447]	CATGTAATCAAAATCATAGGTCAATTAATT
16[111]-14[112]	TCAAATCGTTTGACCATTAGATAAGCAAAC
16[399]-14[400]	GTTTAACGTCAATATATGTGAGTTTGGGTT
17[160]-19[159]	TCATTTGGATGCCTGAGTAATGTAATATTT
17[352]-19[351]	TCATTTGATCGGGAGAAACAATAACGTTAT
17[448]-19[447]	TTCCCTTTATCAAAATTATTTGTCAGATGA
17[64]-19[63]	TATAACAGATAAATTAATGCCGGAACTAGC
18[111]-16[112]	TTCTGGTCCAAAAACAGGAAGAGGAGACAG
18[303]-16[304]	AGAGCCGTTTAGACTTTACAAACTTACAAA
18[399]-16[400]	CAATCAATCAAAGAAACCACCAGTTTTCAG
19[160]-21[159]	TGTTAAAACGACGACAGTATCGGTCGTAAT
19[352]-21[351]	TAATTTTGAATTGAGGAAGGTTAAGAATAC
19[448]-21[447]	TGGCAATCAGCAAATGAAAAATGAACCACC
19[64]-21[63]	ATGTCAATCAGGCTGCGCAACTGTTTTCCC
20[111]-18[112]	AGTTGCAATGCCTGCAGGTCGAGGCACCGC
20[303]-18[304]	TACATTGGGGGACATTCTGGCCAATAGATT
20[399]-18[400]	TATCCAGAATGCGCGAACTGATAAAACCCT
21[160]-23[159]	CATGGTCATTTCACCAGTGAGACGGGTCGA
21[256]-23[255]	GCTAACTCCCAGTCGGGGAAACCTAGAAAGC
21[352]-23[351]	GTGGCACAACGCTCATGGAAATGCTTTCCT
21[448]-23[447]	AGCAGAATAATAACATCACTTGTTATAATC
21[64]-23[63]	AGTCACGACGAAAATCCTGTTTGATTAAAG
23[288]-20[304]	TAATGCGCCGCTACAGGGCGCGTGATTATT
23[384]-20[400]	GCCGATTAAAGGGATTTTAGACACTGGTAA
6[175]-9[159]	TTGATGATACAGGAGTGTACTGGCGTACTC
6[271]-9[255]	CCTTGATATTCACAAACAAATAAAGGAACC
6[79]-9[63]	TCGGAACCTATTATTCTGAAACAGAAGGAT
8[111]-6[112]	ACAACTAGCCGTCGAGAGGGTTAACAGTGC
8[303]-6[304]	CCGATTGACATTAAAGGTGAATTCCAGCAT
9[160]-11[159]	AGGAGGTTATGGGATTTTGCTAAATATTCG
9[256]-11[255]	CATGTACCCAACGCCTGTAGCATTACAGAG
9[352]-11[351]	GCAAAATCAATAGAAAATTCATTTAAGCCC
9[448]-11[447]	CAAGTTTAGTATGTTAGCAAACTTACCAGA
9[64]-11[63]	TAGGATTAAAAATCTCCAAAAAATTTATCA
22[279]-25[267]	GGCAAGTGTAGCGGTCACGC
25[268]-22[280]	TGCGCGTAACCACCACACCC

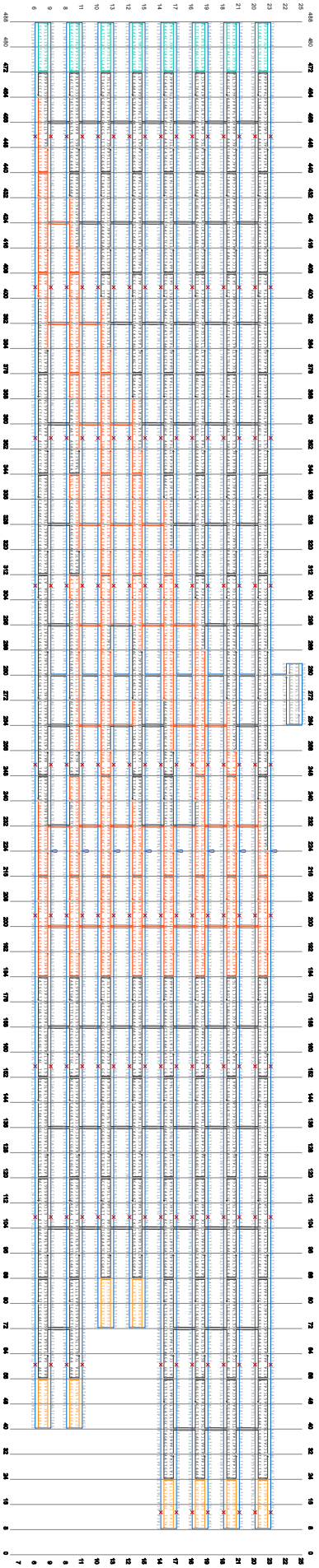
Left edge

9[40]-6[40]	CTGAGACTCCTCAAGATGAAAGTATTAAGAGG
11[40]-8[40]	CCTTTAATTGTATCGGAAGGCTCCAAAAGGAG
13[72]-10[72]	CCGGAACGAGGCGCAGGCTCCATGTTACTTAG
15[72]-12[72]	CAGATACATAACGCCACCACATTCAACTAATG
17[8]-14[8]	CTCAACATGTTTTAAAATATAATGCTGTAG
19[8]-16[8]	CCTGAGAGTCTGGAGGCTATCAGGTCATTG
21[8]-18[8]	CGAAAGGGGGATGTGTATTACGCCAGCTGG
23[8]-20[8]	CCCGAGATAGGGTTGAAATCAAAAGAATAG

Hairpin-labeled staples (hairpin sequence in lowercase)

10[239]-8[240]	CACTACGACGAGGGTAtcctcttttgaggaacaagttttcttgtGCAACGGCTCCACAGA
10[271]-11[287]	GTTTCCATGACTAAAGtcctcttttgaggaacaagttttcttgtACTTTTTCCCCTGAAC
10[335]-8[336]	AGAGAGAAACCCACAAtcctcttttgaggaacaagttttcttgtGAATTGAGATGGTTTA
10[367]-8[368]	TAACGTCAAGAGCAAGtcctcttttgaggaacaagttttcttgtTAAACAATGTTTATTTT
11[192]-13[191]	TTTTGCGGAATACACTtcctcttttgaggaacaagttttcttgtAAAACACTAGAGTAAT
11[224]-13[223]	CATCGGAAAGGCACCAtcctcttttgaggaacaagttttcttgtACCTAAAATACCCAAA
11[288]-10[272]	AAAGTCAGGAGAATTAtcctcttttgaggaacaagttttcttgtACTGAACAATGAGGAA
11[320]-13[319]	GAGAGATATAACATAAtcctcttttgaggaacaagttttcttgtAAACAGGGCAAGCCGT
11[384]-13[383]	ATAGCTATCCAAATAAtcctcttttgaggaacaagttttcttgtGAAACGATAGAAGGCT
12[239]-10[240]	TTAATCATGCTGCTCAtcctcttttgaggaacaagttttcttgtTTCAGTGACGTAATGC
12[271]-13[287]	AAATTGGGACGAGAAAtcctcttttgaggaacaagttttcttgtCACCAGAATAAACCAA
12[335]-10[336]	TGTAGAAATCATCGTAtcctcttttgaggaacaagttttcttgtGGAATCATGCCTTTAC
12[367]-10[368]	AAAAATAACAAGCAAAtcctcttttgaggaacaagttttcttgtTCAGATATTTTTTGTT
13[192]-15[191]	CTTGACAAGCTCATTAtcctcttttgaggaacaagttttcttgtTACCAGTCGAAGTTTT
13[224]-15[223]	GTAACAAATGTGAATTtcctcttttgaggaacaagttttcttgtACCTTATGGTTTAGAC
13[320]-15[319]	TTTTATTTCCAATCAAtcctcttttgaggaacaagttttcttgtTAATCGGCAGAATAAA
14[335]-12[336]	TTCAAATATCATAATTtcctcttttgaggaacaagttttcttgtACTAGAAATACGAGCA
15[192]-17[191]	GCCAGAGGGCATCAAAtcctcttttgaggaacaagttttcttgtAAGATTAAGTGGCATC
15[224]-17[223]	CAATACTGCTGACTATtcctcttttgaggaacaagttttcttgtTATAGTCATAACATCC
15[288]-14[272]	TGTGATAATTTAATGGtcctcttttgaggaacaagttttcttgtTTTGAAATAGTTCAGA
15[320]-17[319]	CACCGGAATATTTTAGtcctcttttgaggaacaagttttcttgtTTAATTTCACAAACAT
16[271]-17[287]	GTTGTACCGCCTCAGAtcctcttttgaggaacaagttttcttgtGCATAAAGTTCAATTA
17[192]-19[191]	AATTCTACAATTTTTAtcctcttttgaggaacaagttttcttgtGTGCCTAAGGCGGATT
17[224]-19[223]	TACAGGCAGGAGAAGCtcctcttttgaggaacaagttttcttgtCTTTATTTACGCCATC
17[288]-16[272]	CCTGAGCAGAGGCGAAtcctcttttgaggaacaagttttcttgtTTATTCATCTAAATCG
18[239]-16[240]	GACCGTAATGGCCTTCtcctcttttgaggaacaagttttcttgtCTGTAGCCCTGTAATA
18[271]-19[287]	TCGGATTCTAAATGTGtcctcttttgaggaacaagttttcttgtAGCGAGTATGAGGATT
19[192]-21[191]	AGCTCATTATCGTAACtcctcttttgaggaacaagttttcttgtCGTGCATCGAAATTGT
19[224]-21[223]	TTCGCGTCTGGGATAGtcctcttttgaggaacaagttttcttgtGTCACGTTCATACGAG
20[239]-18[240]	AGCTGCATAGCCTGGGtcctcttttgaggaacaagttttcttgtGTGCCTAAGGCGGATT
21[192]-23[191]	TATCCGCTGTTTGCGTtcctcttttgaggaacaagttttcttgtATTGGGCGACCCTAAA
21[224]-23[223]	AAGTGTAATAATGAATtcctcttttgaggaacaagttttcttgtCGGCCAACACGGGGAA
6[239]-9[223]	TAAAGCCAGAATGGAAtcctcttttgaggaacaagttttcttgtAGCGCAGTGAGCCACC
6[431]-9[415]	ATCACCGGAACCAGAGtcctcttttgaggaacaagttttcttgtCCACCACCAAACCATC
8[431]-6[432]	TACATACACACCGTAAtcctcttttgaggaacaagttttcttgtCAGTAGCTAATCAAA
9[192]-11[191]	CACCCTCATCTTTCCAtcctcttttgaggaacaagttttcttgtGACGTTAGAAGGCCGC
9[224]-11[223]	ATTTTCAGATAGTTAGtcctcttttgaggaacaagttttcttgtCGTAACGACAGCGAAA
9[384]-11[383]	CAAGGCCGGACACCACtcctcttttgaggaacaagttttcttgtGGAATAAGAAATAGCA
9[416]-11[415]	GATAGCAGTAAAGGTGtcctcttttgaggaacaagttttcttgtGCAACATATTTAAGAA
23[192]-20[208]	GGGAGCCCCCGATTTtcctcttttgaggaacaagttttcttgtAGAGCTTGGCGCGGGG
6[207]-9[191]	TTACCGTTCCAGTAAtcctcttttgaggaacaagttttcttgtGCGTCATAAGAACCGC
6[463]-9[447]	TATTAGCGTTTGCCAtcctcttttgaggaacaagttttcttgtTCTTTTCAGACAGAAT
10[207]-8[208]	GCAAAAGGATCGTCAtcctcttttgaggaacaagttttcttgtCCCTCAGTCTAAAGT
10[303]-8[304]	TTAGACGGAGGGTAAtcctcttttgaggaacaagttttcttgtTTGAGCGCCATTCAA
10[399]-8[400]	ATCCCAATCTTACCGtcctcttttgaggaacaagttttcttgtAAGCCCTTTAAAAGA
11[256]-13[255]	GCTTTGAGTAAACGGtcctcttttgaggaacaagttttcttgtGTAAAATAATAAGGC
11[352]-13[351]	AATAATAAAAATGAAtcctcttttgaggaacaagttttcttgtAATAGCATACCGCGC
12[207]-10[208]	AGAACTGGAACCGGAtcctcttttgaggaacaagttttcttgtTATTCATCGAAAGAG
12[303]-10[304]	CCTTATCACTCATCGtcctcttttgaggaacaagttttcttgtAGAACAAGAAGCGCA
13[352]-15[351]	CCAATAGTATCCCATtcctcttttgaggaacaagttttcttgtCCTAATTAAGCCTGT
14[207]-12[208]	GCGGATTGGGTAATAtcctcttttgaggaacaagttttcttgtGTAAAATCGATTTTA
14[303]-12[304]	GACCTAAAATAAGGCtcctcttttgaggaacaagttttcttgtGTTAAATATGTCTTT
16[207]-14[208]	GGATAAATAATAGTAtcctcttttgaggaacaagttttcttgtGTAGCATGAAGCAAA
16[303]-14[304]	ATCGCGCAAAAGAAGtcctcttttgaggaacaagttttcttgtATGATGAAATCTTCT
17[256]-19[255]	GCAATAAAAAAAACAtcctcttttgaggaacaagttttcttgtTTATGACCAGCTTTC
18[207]-16[208]	TGGGCGCTTTTAACCtcctcttttgaggaacaagttttcttgtAATAGGACAACGCAA
19[256]-21[255]	ATCAACATTCCGTGGtcctcttttgaggaacaagttttcttgtGAACAAACTGAGTGA
20[207]-18[208]	AGAGGCGCACAATTCtcctcttttgaggaacaagttttcttgtCACACAAGGTGTAGA
8[207]-6[208]	TTGTCGGAACCGCCtcctcttttgaggaacaagttttcttgtACCCTCACTCTGAAT
8[399]-6[400]	AACGCAAAGAAACGTtcctcttttgaggaacaagttttcttgtCACCAATGGGAACCG

Positions of dumbbells are indicated by the red-colored staple strands (28-nt dumbbell hairpin sequences are inserted in the middle of each staple strand. See sequence list.)



S5.5. "B" origami

Core

Seq name	Sequence
10[143]-8[144]	TTTGTATCTCGCCCACGCATAACCATAATAAT
10[175]-8[176]	ATACCAAGTGAGGCTTGCAGGGAGGAATAGAA
10[271]-11[287]	GGGTAAAATTCATGAGGAAGTTTCAAAGAACT
10[335]-8[336]	CCGAACAAACGTAGAAAATACATAAATCACCA
10[367]-8[368]	TGAAATAGATATAAAAGAAACGCAACTTGAGC
10[431]-8[432]	GCTAATATATATGGTTTACCAGCGTGAGGGAG
11[128]-13[127]	AACAACCAATCGCCTGATAAATTGACAGATGA
11[192]-13[191]	CGCTTTTGACTCATCTTTGACCCCCAAGAACC
11[288]-10[272]	GGCATGATAATAACGGAATACCCACATTAAAC
11[384]-13[383]	ACGGAATACCCAATAATAAGAGCATAAACAGC
11[416]-13[415]	GAAAATTCCAGAGAGATAACCCACTAAGAAAC
11[96]-13[95]	GCTTGATACCTGCTCCATGTTACTAGGGAACC
12[143]-10[144]	TTCATCAGCAGACCAGGCGCATAGAACGGAGA
12[175]-10[176]	TAACGGAATCAAGAGTAATCTTGACAGCGATT
12[271]-13[287]	TTAATCATGCTTGAGATGGTTTAAATTAGTTG
12[335]-10[336]	GAGGCGTTCAACGCTAACGAGCGTCGAAGCCC
12[367]-10[368]	ATTACCGCTTTGCCAGTTACAAAAAGAAACAA
12[431]-10[432]	ATTAAACCGTTTAACGTCAAAAATAATTGAGC
12[463]-10[464]	GCTGTCTTTACAGAGAGAATAACATAACTGAA
13[128]-15[127]	ACGGTGTATTGAGATTTAGGAATAAAAAACCA
13[192]-15[191]	GGATATTCAAAATCTACGTTAATAGTTTAGAC
13[384]-15[383]	CATATTATGTTTTTATTTTCATCGGTCCAGAC
13[416]-15[415]	GATTTTTTAAGTACCGCACTCATCGCTAATGC
13[96]-15[95]	GAACTGACGCAGATACATAACGCCCTATCATA
14[143]-12[144]	CCGGAAGCAGAGGCTTTTGCAAAATAGAAAGA
14[175]-12[176]	CGTTTTAAGGGGTAATAGTAAAATAAACGAAC
14[367]-12[368]	AATTACTAAAGGTAAAGTAATTCTTAGGAATC
14[431]-12[432]	ATGGTTTGCCTGTTTATCAACAATGAACGGGT
14[463]-12[464]	TTAGTTAAAAGAAAAATAATATCCAATAATCG
14[47]-17[31]	GCTGAATATAATGCTGTAGCTCAAGTCTGGAA
15[128]-17[127]	AAATAGCGAAACTCCAACAGGTCAAACCTGTT
15[192]-17[191]	TGGATAGCAAGCCCGAAAGACTTCAGTAGTAG
15[384]-17[383]	GACGACAAAATAAGAATAAACACCATTTATCA
15[416]-17[415]	AGAACGCGAAATACCGACCGTGTGCTTTTTAA
15[96]-17[95]	ACCCTCGTTTTAATTGCTCCTTTTGACCATTA
16[143]-14[144]	AGAAAGGCTTTTCATTTGGGGCGCGAACCAGA
16[175]-14[176]	AGTAATGTCATCAATTCTACTAATAAATATCG
16[271]-17[287]	TTATGACCGCTAAATCGGTTGTACTTCTGTAA
16[431]-14[432]	CGGGAGAATTAGGTTGGGTTATATTAAATTTA
16[463]-14[464]	GATGAATACTGATGCAAATCCAATAATATATT
16[47]-14[48]	TCTACAAACCATATAACAGTTGATGCTTAATT
16[79]-14[80]	AATGCCGGCGAGTAGATTTAGTTTGATAAGAG
17[128]-19[127]	TAGCTATACGGAGACAGTCAAATCTTTAAATT
17[192]-19[191]	CATTAACACATATATTTTAAATGCTTTTTAAC
17[288]-16[272]	ATCGTCGCGAGTGAATAACCTTGCCAAAAACA
17[32]-19[31]	GTTTCATTGGCTATCAGGTCATTGATGAACGG
17[320]-19[319]	TCCTTGAATAATGGAAACAGTACAGCGGAACA
17[384]-19[383]	AAATCATAACCAAGTTACAAAATCGCAATTCA
17[416]-19[415]	CCTCCGGCACAATAACGGATTCGCATTATACT
17[96]-19[95]	GATACATTTTCAACCGTTCTAGCTCCCAAAAA
18[143]-16[144]	CTTCTGGTTAATATTTTGTTAAAAAAAGGGTG
18[175]-16[176]	AGATCGCATGTTAAATCAGCTCATAATGCCTG
18[271]-19[287]	GACCGTAACTCCGTGGGAACAAACATTTTAAA
18[431]-16[432]	CAATCAATATGGAAGGGTTAGAACTTTTACAT
18[79]-16[80]	GGCGATCGTGATAATCAGAAAAGCGATAAATT
19[128]-21[127]	GTAAACGTGCCGGAAACCAGGCAACCAGTGCC
19[192]-21[191]	CAATAGGAACGACGACAGTATCGGTCGTAATC
19[288]-18[272]	AGTTTGAGTGCCCGAACGTTATTAGGCGGATT
19[32]-16[48]	TAATCGTAAAACTAGCATGTCAATTTGAGAGA
19[320]-21[319]	AAGAAACCCAAACAATTCGACAACCCCTAAAA
19[384]-21[383]	TCAATATAGGAATTGAGGAAGGTTGCGGTCAG
19[416]-21[415]	TCTGAATAATCTGGTCAGTTGGCACCACGCTG
19[96]-21[95]	CAGGAAGATCAGGCTGCGCAACTGTTTTCCCA
20[143]-18[144]	GTTTTTCTATGCCTGCAGGTCGACCGGCACCG
20[175]-18[176]	AGAGGCGGGGTACCGAGCTCGAATCCTCAGGA
20[431]-18[432]	ATACCTACCAGCAAATGAAAAATCCAAACCCT
20[79]-18[80]	AGTTGCAGTTGGGTAACGCCAGGGTTGGGAAG
21[128]-23[127]	AAGCTTGCTTTCACCAGTGAGACGGTGTTGTT
21[192]-23[191]	ATGGTCATTAATGAATCGGCCAACGAAAAACC
21[384]-23[383]	TATTAACATTTACATTGGCAGATTCATCACTT
21[416]-23[415]	AGAGCCAGATTTTGACGCTCAATCATCGGCCT
21[96]-23[95]	GTCACGACCCCTTCACCGCCTGGCAAATCAAA
23[128]-20[144]	CCAGTTTGGAACAAGAGTCCACTACCAGGGTG
23[160]-20[176]	ACGTGGACTCCAACGTCAAAGGGCGCGCGGGG
23[416]-20[432]	TGCTGGTAATATCCAGAACAATATTCATGGAA
23[64]-20[80]	TCCGAAATCGGCAAAATCCCTTATCCTGAGAG
6[143]-9[127]	GATTAGGATTAGCGGGGTTTTGCTCTCAGAAC
6[335]-9[319]	GAATTTACCGTTCCAGTAAGCGTCTTTTCGGT
6[399]-9[383]	TCAGACGATTGGCCTTGATATTCAGAGCCACC
6[431]-9[415]	GCCGCCAGCATTGACAGGAGGTTGCGCCACCC
8[143]-6[144]	TTTTTCACTCAGAACCGCCACCCTAAGAGAAG
8[175]-6[176]	AGGAACAACATTTTCAGGGATAGCACATGAAA
8[335]-6[336]	TGAAACCACCTTATTAGCGTTTGCGGAAAGCG
8[367]-6[368]	CATTTGGGAAAATCACCGGAACCACAAACAAA
8[431]-6[432]	GGAAGGTAGCCACCCTCAGAGCCACCAGAGCC
9[128]-11[127]	CGCCACCCGTTGAAAATCTCCAAAACAATGAC
9[192]-11[191]	TAGGAACCAGTTTCAGCGGAGTGATTAAAGGC
9[384]-11[383]	ACCGGAACATTATCACCGTCACCGAAGACACC
9[416]-11[415]	TCAGAACCAATATTGACGGAAATTAATCAATA
9[96]-11[95]	TCAGGAGGGGAGCCTTTAATTGTACTTAAACA
10[463]-13[447]	CACCCTGAACAAAGTCAGAGGGTGAAAATAG
19[448]-16[464]	TCAAAATTATTTGCACGTAAAACTAACGTCA
23[352]-20[368]	AATACTTCTTTGATTAGTAATAACACCAGTC
23[96]-20[112]	AGAATAGCCCGAGATAGGGTTGAGGCAACAG
6[111]-9[95]	GGCGGATAAGTGCCGTCGAGAGGCACCGTAC
6[207]-9[191]	ATTTCGGAACCTATTATTCTGAAAAGCCCAA
6[367]-9[351]	TAAATCCTCATTAAAGCCAGAATCATCTTTT
10[111]-8[112]	TCCGCGACCGATAGTTGCGCCGAAAAAGGC
10[399]-8[400]	GAGTTAAGAGTTTATTTTGTCACATTCATT
11[160]-13[159]	TCGGTCGCCGCGAAACAAAGTACGCTGGCT
11[256]-13[255]	AAGACTTTTACGTAATGCCACTACGAGTAG
11[352]-13[351]	TGGCAACCAATAGCTATCTTACCTTTCCAG
12[111]-10[112]	AACTAATCAACTTTGAAAGAGGTGTCGAAA
12[399]-10[400]	AGCAAGCCTTATCCCAATCCAAAAAGAATT
13[160]-15[159]	GACCTTCACAACATTATTACAGGGAAGTTT
13[352]-15[351]	AGCCTAAGCCCAATAGCAAGCAATAAAGTA
13[448]-15[447]	CAGCCTTTCCTTATCATTCCAAAGATAAGT
14[111]-12[112]	GAGTACCTTACCAGACGACGATCCACATTC
14[399]-12[400]	AGGCGTTATAAACAACATGTTCAGAGAACA
14[79]-17[63]	GTCATTTTTGCGGATGGCTTAGATCCCAAT
15[160]-17[159]	TGCCAGAGTTCGAGCTTCAAAGCGAGCTGA
15[448]-17[447]	CCTGAACTTTCATCTTCTGACCAACTATAT
16[111]-14[112]	TATGATATCGCAAATGGTCAATGGATTAGA
16[303]-14[304]	ATATATGTTATTAATTAATTTTCCGCTCAA
16[399]-14[400]	CTTTGAATGGTCTGAGAGACTACATAAATA
17[160]-19[159]	AAAGGTGGGTAGGTAAAGATTCATTCGCAT
17[256]-19[255]	AGCATAAACTGTAATACTTTTGCACAACCC
17[448]-19[447]	GTAAATGTACAGTAACAGTACCCTACCATA
17[64]-19[63]	TCTGCGAAAGAGGGTAGCTATTTCATATGT
18[111]-16[112]	TCGCCATTTGTATAAGCAAATAACCATCAA
18[303]-16[304]	AAATCCTTTAACATTATCATTTTTAAATCA
18[399]-16[400]	AGTTGAAAATCTGATTGTTTGGCTGATTG
19[160]-21[159]	TAAATTTTCTCCAGCCAGCTTTCTCTAGAG
19[256]-21[255]	GTCGGATTTGGGATAGGTCACGTAAGCATA
19[64]-21[63]	ACCCCGGTGTGCGGGCCTCTTCGTGCAAGG
20[111]-18[112]	CTGATTGGTTGTAAAACGACGGAGCGCCAT
20[399]-18[400]	ATGGATTACCGCCTGCAACAGTGAATCAAC
21[160]-23[159]	GATCCCCGTTTGCGTATTGGGCGTTAAAGA
21[64]-23[63]	CGATTAAGCAAGCGGTCCACGCTTGGTGGT
23[384]-20[400]	GCCTGAGTAGAAGAACTCAAACTGTCTGAA
6[175]-9[159]	GTATTAAGAGGCTGAGACTCCTCCAGAGCC
8[111]-6[112]	TCCAAAATTTAGTACCGCCACCCAGTACCA
8[399]-6[400]	AAAGGTGACGCCTCCCTCAGAGCAGGCAGG
9[160]-11[159]	ACCACCCTCTAAAGGAATTGCGAGATATAT
9[352]-11[351]	CATAATCAATTAGAGCCAGCAACATAAAGG
22[239]-25[235]	AAGCGAAAGGAGCGGGCGCTAGGGCGCT
22[307]-22[280]	AGACAGGAACGGTACGCCAGAATCCTGA
22[335]-22[308]	TAAACAGGAGGCCGATTAAAGGGATTTT
25[284]-25[311]	TAATGCGCCGCTACAGGGCGCGTACTAT
25[312]-25[339]	GGTTGCTTTGACGAGCACGTATAACGTG
25[340]-22[336]	CTTTCCTCGTTAGAATCAGAGCGGGAGC
25[236]-25[259]	GGCAAGTGTAGCGGTCACGCTGCG
25[260]-25[283]	CGTAACCACCACACCCGCCGCGCT
22[259]-22[240]	GGCGAGAAAGGAAGGGAAGA
22[279]-22[260]	CGGGGAAAGCCGGCGAACGT

Right Edge

6[455]-9[455]	CCACCAGAACCACCACCACCCTCAGAGCCG
8[455]-11[455]	CGACATTCAACCGATCCAAAGACAAAAGGG
10[487]-13[487]	CATTAGACGGGAGAATTAAAAACAGGGAAGCG
12[487]-15[487]	CATGTAGAAACCAATCCATCCTAATTTACGAG
14[487]-17[487]	CGAGAAAACTTTTTCACGCAAGACAAAGAACG
16[487]-19[487]	CGTAGATTTTCAGGTTAGAAATAAAGAAATTG
18[455]-21[455]	CTGAACCTCAAATATTAAAGCATCACCTTG
20[455]-23[455]	CAACAGGAAAAACGCTACCGCCAGCCATTG

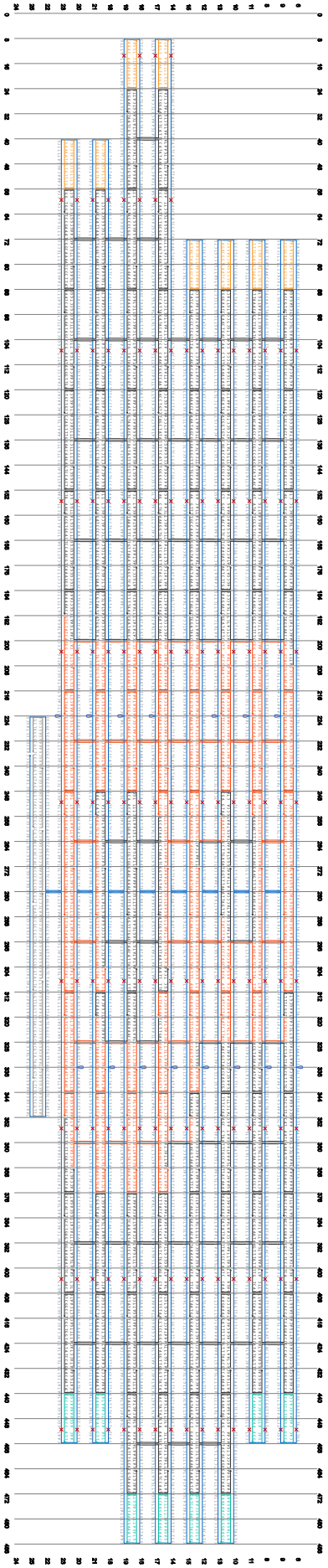
Left Edge

9[72]-6[72]	CCCGGAATAGGTGTATGTTGATATAAGTATAG
11[72]-8[72]	CTTTCGAGGTGAATTTTCGGTTTATCAGCTTG
13[72]-10[72]	CAGACGGTCAATCATATAGCCGGAACGAGGCG
15[72]-12[72]	CATAGTAAGAGCAACAAAAAGGAATTACGAGG
17[8]-14[8]	CAACTAAAGTACGGTCATGTTTTAAATATG
19[8]-16[8]	CAAACAAGAGAATCGCCTGAGAGTCTGGAG
21[40]-18[40]	CGAAAGGGGGATGTGCCTATTACGCCAGCTGG
23[40]-20[40]	CGAAAATCCTGTTTGAGGTTTGCCCCAGCAGG

Hairpin-labeled staples (hairpin sequence in lowercase)

10[239]-8[240]	CCAACCTAGGCTACAGtcctcttttgaggaacaagttttcttgtAGGCTTTGTTAGTAAA
11[224]-13[223]	GTAGCAACAAACGAAAtcctcttttgaggaacaagttttcttgtGAGGCAAAGTAACAAA
11[320]-13[319]	GTTAGCAAAGTTACCAtcctcttttgaggaacaagttttcttgtGAAGGAAATTATCCTG
12[239]-10[240]	AAGAACTGACGAGAAAtcctcttttgaggaacaagttttcttgtCACCAGAACGAAGGCA
13[224]-15[223]	TTGCCCTGGCTCATTAtcctcttttgaggaacaagttttcttgtTACCAGTCATCGTCAT
13[288]-12[272]	CTATTTTGGAAGCCTTtcctcttttgaggaacaagttttcttgtAAATCAAGTTTCAACT
13[320]-15[319]	AATCTTACTTAGCGAAtcctcttttgaggaacaagttttcttgtCCTCCCGACAGAGGCA
14[239]-12[240]	GTCAGAAGATTGAATCtcctcttttgaggaacaagttttcttgtCCCCTCAAGCGATTTT
14[271]-15[287]	CAAAAATCCAGAAAACtcctcttttgaggaacaagttttcttgtGAGAATGACATATTTA
14[335]-12[336]	AAATTCTTCCAGTAATtcctcttttgaggaacaagttttcttgtAAGAGAATAATCAGAT
15[224]-17[223]	AAATATTCCAAAGCGGtcctcttttgaggaacaagttttcttgtATTGCATCAGGCAAGG
15[288]-14[272]	ACAACGCCCTTAATTGtcctcttttgaggaacaagttttcttgtAGAATCGCCCATAAAT
15[320]-17[319]	TTTTCGAGACCAGTATtcctcttttgaggaacaagttttcttgtAAAGCCAACCTTAGAA
16[239]-14[240]	CCTTTATTAAAATTAAtcctcttttgaggaacaagttttcttgtGCAATAAACTATTATA
16[335]-14[336]	ACCTTTTTAACATAGCtcctcttttgaggaacaagttttcttgtGATAGCTTGTATCATA
16[367]-14[368]	GCGAATTAGAAGAGTCtcctcttttgaggaacaagttttcttgtAATAGTGAGGAATCAT
17[224]-19[223]	GAATTAGCTCAACGCAtcctcttttgaggaacaagttttcttgtAGGATAAAATTCGCGT
18[239]-16[240]	ATGGGCGCTAAATGTGtcctcttttgaggaacaagttttcttgtAGCGAGTAGGGAGAAG
18[335]-16[336]	AGACTTTAACCAGAAGtcctcttttgaggaacaagttttcttgtGAGCGGAATGAGCAAA
18[367]-16[368]	TATCTTTATGATTATCtcctcttttgaggaacaagttttcttgtAGATGATGGCGCAGAG
19[224]-21[223]	ATCAACATATCGTAACtcctcttttgaggaacaagttttcttgtCGTGCATCGAAATTGT
20[239]-18[240]	CCCGCTTTACAACATAtcctcttttgaggaacaagttttcttgtCGAGCCGGTGGTGTAG
20[271]-21[287]	CTAACTCAAGCCTGGGtcctcttttgaggaacaagttttcttgtGTGCCTAAGGCTATTA
20[335]-18[336]	GAACCCTTTTAAAAATtcctcttttgaggaacaagttttcttgtAATTAACCATTCTGGC
20[367]-18[368]	ACACGACCATAAAACAtcctcttttgaggaacaagttttcttgtGAGGTGAGATCTAAAA
21[224]-23[223]	AATTCCACCCAGTCGGtcctcttttgaggaacaagttttcttgtGAAACCTGACGTGAAC
21[288]-20[272]	GTCTTTAAGACAATATtcctcttttgaggaacaagttttcttgtTTTTGAATTGAGTGAG
21[320]-23[319]	CATCGCCACTGACCTGtcctcttttgaggaacaagttttcttgtAAAGCGTAGTAAAAGA
23[224]-20[240]	GTGCCGTAAAGCACTAtcctcttttgaggaacaagttttcttgtAATCGGAAGCTCACTG
23[256]-23[287]	GGGAGCCCCCGATTTAtcctcttttgaggaacaagttttcttgtGAGCTTGAGAAGTGTT
23[320]-20[336]	GTCTGTCCATCACGCAtcctcttttgaggaacaagttttcttgtAATTAACCATTCTGGC
6[239]-9[223]	GTGCCCGTATAAACAGtcctcttttgaggaacaagttttcttgtTTAATGCCTGAGTTTC
6[303]-6[272]	GCTTTTGATGATACAGtcctcttttgaggaacaagttttcttgtGAGTGTACTGGTAATA
8[239]-6[240]	TGAATTTTACTACAACtcctcttttgaggaacaagttttcttgtGCCTGTAGGAGTAACA
8[271]-9[287]	AAGTTTTGCCTCATAGtcctcttttgaggaacaagttttcttgtTTAGCGTAAGCGTCAG
9[224]-11[223]	CAGTACAACTGTATGGtcctcttttgaggaacaagttttcttgtGATTTTGCCAGCAGCG
9[288]-8[272]	ACTGTAGCAATCAAGTtcctcttttgaggaacaagttttcttgtTTGCCTTTACGATCTA
9[320]-11[319]	CATAGCCCTCGATAGCtcctcttttgaggaacaagttttcttgtAGCACCGTCGCAGTAT
23[192]-20[208]	GTCTATCAGGGCGATtcctcttttgaggaacaagttttcttgtGGCCCACTTCGTGCCA
10[207]-8[208]	CTAAAACCGGGATCGtcctcttttgaggaacaagttttcttgtTCACCCTTAAACAAC
10[303]-8[304]	AACGCAATTAAGACTtcctcttttgaggaacaagttttcttgtCCTTATTAAATCAGT
12[207]-10[208]	GGGAAGAATTACCCAtcctcttttgaggaacaagttttcttgtAATCAACAGAATACA
12[303]-10[304]	GAGGTTTTCACCCAGtcctcttttgaggaacaagttttcttgtCTACAATTCCGAGGA
13[256]-15[255]	TAAATTGGTGTGAATtcctcttttgaggaacaagttttcttgtTACCTTATATGCTTT
14[207]-12[208]	TAAGAGGGTCCAATAtcctcttttgaggaacaagttttcttgtCTGCGGAAGGACGTT
14[303]-12[304]	CAGTAGGGAACATGTtcctcttttgaggaacaagttttcttgtAATTTAGGCTTGCGG
15[256]-17[255]	AAACAGTTAGGTCTTtcctcttttgaggaacaagttttcttgtTACCCTGAGCCTCAG
15[352]-17[351]	CCGACAAGAAAAAGCtcctcttttgaggaacaagttttcttgtCTGTTTAAGATTAAG
16[207]-14[208]	GAACCCTTCCAATAAtcctcttttgaggaacaagttttcttgtATCATACAAAAAGAT
17[352]-19[351]	ACGCTGATTCATTTCtcctcttttgaggaacaagttttcttgtAATTACCTTATCATC
18[207]-16[208]	TGAGGGGACGCCATCtcctcttttgaggaacaagttttcttgtAAAAATAAATTTTTA
19[352]-21[351]	ATATTCCGGAGCACTtcctcttttgaggaacaagttttcttgtAACAACTAACCACCA
20[207]-18[208]	GCTGCATAGCTGTTTtcctcttttgaggaacaagttttcttgtCTGTGTTGCCAGTT
20[303]-18[304]	GTGGCACATGCGCGAtcctcttttgaggaacaagttttcttgtACTGATAGTCGTATT
21[256]-23[255]	AAGTGTAACATTAAtcctcttttgaggaacaagttttcttgtTGCGTTGCCCCTAAA
21[352]-23[351]	GCAGAAGAGTAATAAtcctcttttgaggaacaagttttcttgtAAGGGACGTTGTAGC
23[288]-20[304]	TTTATAATCAGTGAGtcctcttttgaggaacaagttttcttgtGCCACCGAAGAATAC
6[271]-9[255]	AGTTTTAACGGGGTCtcctcttttgaggaacaagttttcttgtAGTGCCTTCATTCCA
8[207]-6[208]	TTTCAACCATGTACCtcctcttttgaggaacaagttttcttgtGTAACACCCCTGCCT
8[303]-6[304]	AGCGACAGGCGTTTTtcctcttttgaggaacaagttttcttgtCATCGGCAATACATG
9[256]-11[255]	CAGACAGCTCGTCTTtcctcttttgaggaacaagttttcttgtTCCAGACGAGGACTA

Positions of dumbbells are indicated by the red-colored staple strands (28-nt dumbbell hairpin sequences are inserted in the middle of each staple strand. See sequence list.)



S5.6. "C" origami

Core

Seq name	Sequence
6[111]-9[95]	AAGACAGCATCGGAACGAGGGTACTTTTTCA
6[143]-9[127]	CGCTTTTGCGGGATCGTCACCCTCTAAAATAC
6[175]-9[159]	CGGTCGCTGAGGCTTGCAGGGAGCCTAAAA
6[207]-9[191]	ACAACCATCGCCCACGCATAACCAAACACTC
6[239]-9[223]	GCTTGATACCGATAGTTGCGCCGACCAAGCGC
6[271]-9[255]	CAGCTTGCTTTCGAGGTGAATTTTGTTACT
6[303]-6[272]	TCCAAAAGGAGCCTTTAATTGTATCGGTTTAT
6[335]-9[319]	TTTTCACGTTGAAAATCTCCAAAACTGAGACT
6[367]-9[351]	GGGATTTTGCTAAACAACTTTCACGGGGTTT
6[399]-9[383]	TTTCCAGACGTTAGTAAATGAATTGCCGTCGA
6[431]-9[415]	AGTTAGCGTAACGATCTAAAGTTTGGAATAGG
6[463]-9[447]	ACGCCTGTAGCATTCCACAGACAAGTACCGC
6[495]-9[479]	AACACTGAGTTTCGTCACCAGTACGAACCGCC
9[96]-11[95]	TGAGGAAGACGAGAAACACCAGAATTCAACTT
9[128]-11[127]	GTAATGCCGCTGCTCATTCAGTGACGATTTTA
9[160]-11[159]	CGAAAGAGGGATATTCATTACCCGGACGTT
9[192]-11[191]	ATCTTTGAGACCTTCATCAAGAGTAACGAACT
9[352]-11[351]	TGCTCAGAATAAGTTTTAACGGGGCATTTT
9[384]-11[383]	GAGGGTTGCATGGCTTTTGATGATTTTGCCAT
9[416]-11[415]	TGTATCACCTCTGAATTTACCGTTAACCAGAG
9[448]-11[447]	CACCCTCTCCTCATTAAAGCCAGAGCCGCC
9[480]-11[479]	ACCCTCAGACGATTGGCCTTGATAAGCCACCA
8[111]-6[112]	TGCCCTGTTTCCATTAAACGGGAGCAGCGA
8[143]-6[144]	GTAACAAAACTACGAAGGCACCAATTAAAGGC
8[175]-6[176]	CAAGAACCGCAAAAGAATACACTAGATATATT
8[207]-6[208]	GCTGGCTCCCCCAGCGATTATACAATGACA
8[335]-6[336]	CCTTGAGTGAAGGATTAGGATTAGACAGTTTC
8[367]-6[368]	GTACTGGTTACCAGGCGGATAAGTTTCTGTAT
8[399]-6[400]	GCGTCATAATATAAGTATAGCCCTGTCGTC
8[431]-6[432]	AGCGCAGTCGTACTCAGGAGGTTTGCCCTCAT
8[463]-6[464]	CAAATAAAAGAACCGCCACCCTCAAAACTACA
8[495]-6[496]	CAGGTCAGAGCCACCACCCTCATGTACCGT
11[96]-8[112]	TAATCATTGTGAATTACCTTATGATAAGGCT
11[128]-13[127]	AGAACTGGTCATAAATATTCATTGAAAACGAG
11[160]-13[159]	GGGAAGAAAGACTGGATAGCGTCTCTTTAC
11[192]-13[191]	AACGGAACGTTTTGCCAGAGGGGGAGCGGATT
11[256]-13[255]	CGCCAAAACATAACCCTCGTTTAAAGCGAA
11[288]-10[272]	CAGTAGCGAACCATCGATAGCAGCGTAAGAGC
11[352]-13[351]	CGGTCATTATTCATTAAAGGTGAAAGTTAC
11[384]-13[383]	CTTTTCATTTGAGGGAGGGAAGGTATAATAAC
11[416]-13[415]	CCACCACCCGCCAAAGACAAAAGGGATTAAGA
11[448]-13[447]	ACCCTCACAATCAATAGAAAATAAACGTAG
11[480]-8[496]	CCCTCAGAGCCGCCACCAGAACCGTTGAGG
10[143]-8[144]	CGGAATCGCTCATTATACCAGTCAAAATCAAC
10[175]-8[176]	AAATGTTTAAATCTACGTTAATAAAATCTTGA
10[207]-8[208]	AAAAGAAAACATTATTACAGGTGCGCATAG
10[367]-8[368]	ACGGAAATAGCCCCCTTATTAGCGACAGGAGT
10[399]-8[400]	TCAACCGAAATCAAAATCACCGGCCAGTAA
10[431]-8[432]	TTTACCAGGGAACCGCCTCCCTCAGAATGGAA
10[463]-8[464]	TTTTGTCAGAACCGCCACCCTCAGTTCACAAA
13[128]-15[127]	AATGACCACATTCCATATAACAGTAGTTTGAC
13[160]-15[159]	CCTGACTATATGCAACTAAAGTACAATAAC
13[192]-15[191]	GCATCAAAGAATATAATGCTGTAGGGCGCGAG
13[256]-15[255]	CCAGACCGTACCTTTAATTGCTCAAGGCAA
13[288]-12[272]	AGCAATAGAATAAGAGCAAGAAACGGTCAGGA
13[320]-15[319]	AGAAAAGTGATAACCCACAAGAATCGGTATTC
13[352]-15[351]	CAGAAGGACACCCTGAACAAAGACCTCCCG
13[384]-15[383]	GGAATACCAGGGAAGCGCATTAGATAAATCAA
13[416]-15[415]	CTCCTTATGCAGCCTTTACAGAGAGCTACAAT
13[448]-15[447]	AAAATACGATTTTTTGTTTAACAACGAGCG
12[143]-10[144]	GGAAGTTTTAAATCAAAAATCAGGCAATACTG
12[175]-10[176]	GTTTTAAATTATAGTCAGAAGCAATAATAGTA
12[271]-13[287]	TTAGAGAGGAAGCAAACTCCAACAAATGAAAT
12[335]-10[336]	ATCAGAGAAAGCAGATAGCCGAACAATTATCA
12[367]-10[368]	TTAACTGAAAACCGAGGAAACGCAAAATATTG
12[399]-10[400]	ATAAAAACCAAAAGAACTGGCATGCGACAT
12[431]-10[432]	TGAAAATATACGCAGTATGTTAGCTCATATGG
12[463]-10[464]	TAAGAAACATACATAAAGGTGGCATAAGTTTA
15[128]-17[127]	CATTAGATTGTAGGTAAAGATTCATCACCATC
15[160]-17[159]	CTGTTTAGCATATATTTTAAATGTGATAAA
15[192]-17[191]	CTGAAAAGTTCAACGCAAGGATAATTTCGAGAG
15[256]-17[255]	AGAATTAGGCTAAATCGGTTGTAATGTACC
15[288]-14[272]	AAATCAGATCATTACCGCGCCCAAAGCCTCAG
15[320]-17[319]	TAAGAACGAAGCAAGCCGTTTTTAAGTTAATT
15[352]-17[351]	ACTTGCGTTCCTTATCATTCCAGTTTGAAA
15[384]-17[383]	GATTAGTTGCATGTAGAAACCAATCGTTAAAT
15[416]-17[415]	TTTATCCTAAGAAAAATAATATCCTTACTAGA
15[448]-17[447]	TCTTTCCCCTGTTTATCAACAAGTTATACA
14[143]-12[144]	GAGTAATGACATTTCGCAAATGGTCGGTGTCT
14[175]-12[176]	AGAACCCTCTATATTTTCATTTGGCTCAACAT
14[271]-15[287]	AGCATAAACAAAATTAAGCAATAATAGCAAGC
14[303]-12[304]	CGTAGGAATATAGAAGGCTTATCTGAGTTA
14[335]-12[336]	TCGAGAACCGAGGCGTTTTAGCGATCAGAGGG
14[367]-12[368]	GGCTGTCTGGAGGTTTTGAAGCCTCGGGAGAA
14[399]-12[400]	ATTTACGAGCTATTTTGCACCCAGAATAAC
14[431]-12[432]	TCCTGAACGAATCTTACCAACGCTGTCAAAAA
14[463]-12[464]	AGAACGCGAGAGCCTAATTTGCCAAATCCAAA
14[495]-17[479]	CGACGACAATAAACAACATGTTCACGCTCAAC
14[527]-17[511]	ACCGACAAAAGGTAAAGTAATTCTCATATTTA
17[128]-19[127]	AATATGATGTAGCCAGCTTTCATCCGGATTCT
17[160]-19[159]	TTAATGCCCCATCAAAAATAATTCCGTAAT
17[192]-19[191]	ATCTACAATAAATCAGCTCATTTTGGGCGCAT
17[256]-19[255]	CCGGTTGATATAAGCAAATATTTTTCTGGT
17[288]-16[272]	GAAAACTTCCAATCGCAAGACAAAAAAAACAG
17[352]-19[351]	TACCGACATTAAGACGCTGAGAAGGGTTAG
17[384]-19[383]	AAGAATAATTAGAATCCTTGAAAAGCACGTAA
17[416]-19[415]	AAAAGCCTCTGTAAATCGTCGCTAGATTTTCA
17[448]-19[447]	AATTCTTAATCAATATATGTGAGTAACAGT
17[480]-19[479]	AGTAGGGCTGAATTACCTTTTTTATAACGGAT
17[512]-19[511]	ACAACGCCAACAAAATTAATTACAAGTTACAA
16[143]-14[144]	GCCTTCCTATTCAACCGTTCTAGCCAATGCCT
16[175]-14[176]	TAGGAACGGGAGAGGGTAGCTATTAAATTTTT
16[271]-17[287]	GAAGATTGTAATCAGAAAAGCCCCGAACGCGA
16[303]-14[304]	ATGCAAATTTTCAAATATATTTTTTTTCAT
16[335]-14[336]	GGTTGGGTTGACCTAAATTTAATGAGAACGGG
16[367]-14[368]	TAGCTTAGCGTGTGATAAATAAGGCAATAATC
16[399]-14[400]	ATTTTCCCACACCGGAATCATAACATCCTA
16[431]-14[432]	CCTTGCTTGTTTAGTATCATATGCTAGATAAG
16[463]-14[464]	AGTACATAACCAGTATAAAGCCAAGCTAATGC
16[495]-14[496]	ATTTCATTTTAATTGAGAATCGCGTCCAGA
16[527]-14[528]	ATCAAGAAAACATGTAATTTAGGCTATAAAGT
19[128]-21[127]	CCGTGGGAACGACGGCCAGTGCCAATCCGCTC
19[160]-21[159]	GGGATAGGGCCAGGGTTTTCCCAGGAAGCA
19[192]-21[191]	CGTAACCGGGATGTGCTGCAAGGCAATGAGTG
19[352]-21[351]	AACCTACCAATTCGACAACTCGAGATAAAA
19[384]-21[383]	AACAGAAATTTGAGGATTTAGAAGACACCGCC
19[416]-21[415]	GGTTTAACTAATAGATTAGAGCCGCAGCAGCA
19[448]-21[447]	ACCTTTTTATCTAAAATATCTTTGCTGAAC
19[480]-16[496]	TCGCCTGATTGCTTTGAATACCATTTAACA
19[512]-16[528]	AATCGCGCAGAGGCGAATTATTCAAAACAAAC
18[111]-21[95]	TGCCTGCAGGTCGACTCTAGAGGTGGTCATA
18[143]-16[144]	GTTGTAAAACAAACGGCGGATTGACGCGTCTG
18[175]-16[176]	TGGGTAACTCACGTTGGTGTAGATTTAACCAA
18[271]-19[287]	CAGGCTGCCCAGGCAAAGCGCCATATCAGATG
18[367]-16[368]	TTTACAAACATATCAAAATTATTTCATAGCGA
18[399]-16[400]	ATAATACATAAAGAAATTGCGTATTAATTA
18[431]-16[432]	CTAACAACGTCAGATGAATATACAGTGAATAA
18[463]-16[464]	AGGAAGGTACATCGGGAGAAACAAATGGAAAC
21[96]-23[95]	GCTGTTTCTCGGCAAAATCCCTTAGTGTTGTT
21[128]-23[127]	ACAATTCCGCGAAAATCCTGTTTGATTAAAGA
21[160]-23[159]	TAAAGTGTGCAAGCGGTCCACGCGAAAAAC
21[192]-23[191]	AGCTAACTGCCCTTCACCGCCTGGACGTGAAC
21[224]-23[223]	GCCCGCTTTTTTCACCAGTGAGACGGGTCGAG
21[320]-23[319]	AATACCGAGTAAGAATACGTGGCACAGAATCC
21[352]-23[351]	CAGAGGTCCAACAGAGATAGAAAGGCCACC
21[384]-23[383]	TGCAACAGTCACACGACCAGTAATGCAAATTA
21[416]-23[415]	AATGAAAAAATGGATTATTTACATATTAGTAA
21[448]-23[447]	CTCAAATAATACCTACATTTTGAACTCAAA
20[111]-18[112]	TCCGAAACTGTGTGAAATTGTTAGCTTGCA
20[143]-18[144]	CCCAGCAGACACAACATACGAGCCGTCACGAC
20[175]-18[176]	GAGTTGCAAAAGCCTGGGGTGCCTGATTAAGT
20[207]-18[208]	GCTGATTCACATTAATTGCGTTCAGCTGGC
20[335]-18[336]	CTGAAAGCACGAACCACCAGCAGATATTAAAT
20[367]-18[368]	CATTCTGGGAGGCGGTCAGTATTATATTAGAC
20[399]-18[400]	TTCACCAGTGCCACGCTGAGAGCTCAATAG
20[431]-18[432]	TCGTCTGAATCTAAAGCATCACCTTAGGAGCA
20[463]-18[464]	CTCATGGAATCAAACCCTCAATCAAGGAATTG
23[96]-20[112]	CCAGTTTGGAACAAGAGTCCACTATGGTGGT
23[128]-20[144]	ACGTGGACTCCAACGTCAAAGGGCTGGTTTGC
23[160]-20[176]	CGTCTATCAGGGCGATGGCCCACTCCCTGAGA
23[192]-20[208]	CATCACCCAAATCAAGTTTTTTGGGGCAACA
23[224]-20[240]	GTGCCGTAAAGCACTAAATCGGAAGCCAGGGT
23[256]-23[287]	GGGAGCCCCCGATTTAGAGCTTGAATTAAAGG
23[288]-20[304]	GATTTTAGACAGGAACGGTACGCCAGACAA
23[320]-20[336]	TGAGAAGTGTTTTTATAATCAGTGCCCTTCTG
23[352]-20[368]	GAGTAAAAGAGTCTGTCCATCACAAAAGGGA
23[384]-20[400]	ACCGTTGTAGCAATACTTCTTTGTGGCAGA
23[416]-20[432]	TAACATCACTTGCCTGAGTAGAAGACGCTCAA
23[448]-20[464]	CTATCGGCCTTGCTGGTAATATCGAAAAACG
22[231]-25[239]	GGAGCGGGCGCTAGGGCGCTGGCA
22[255]-22[232]	AGAAAGGAAGGGAAGAAAGCGAAA
22[279]-22[256]	CGGGGAAAGCCGGCGAACGTGGCG
22[303]-22[280]	GAGCGGGAGCTAAACAGGAGGCCG
22[327]-22[304]	AACGTGCTTTCCTCGTTAGAATCA
25[240]-25[263]	AGTGTAGCGGTCACGCTGCGCGTA
25[264]-25[287]	ACCACCACACCCGCCGCGCTTAAT
25[288]-25[311]	GCGCCGCTACAGGGCGCGTACTAT
25[312]-22[328]	GGTTGCTTTGACGAGCACGTAT

Right Edge

6[519]-9[519]	CCCAATAGGAACCCATTTTCAGGGATAGCAAG
8[519]-11[519]	CCAGCATTGACAGGAGACCACCAGAGCCGCCG
10[487]-13[487]	CAAAGACACCACGGAAACATATAAAAGAAACG
12[487]-15[487]	CCATATTATTTATCCCGTTACAAAATAAACAG
14[551]-17[551]	CCAGTAATAAGAGAAAGAGGCATTTTCGAG
16[551]-19[551]	CAAAAGAAGATGATGTTTCAATTACCTGAG
18[487]-21[487]	CAAATCAACAGTTGAAATATCTGGTCAGTTGG
20[487]-23[487]	CCAGCCATTGCAACAGCAGAACAATATTACCG

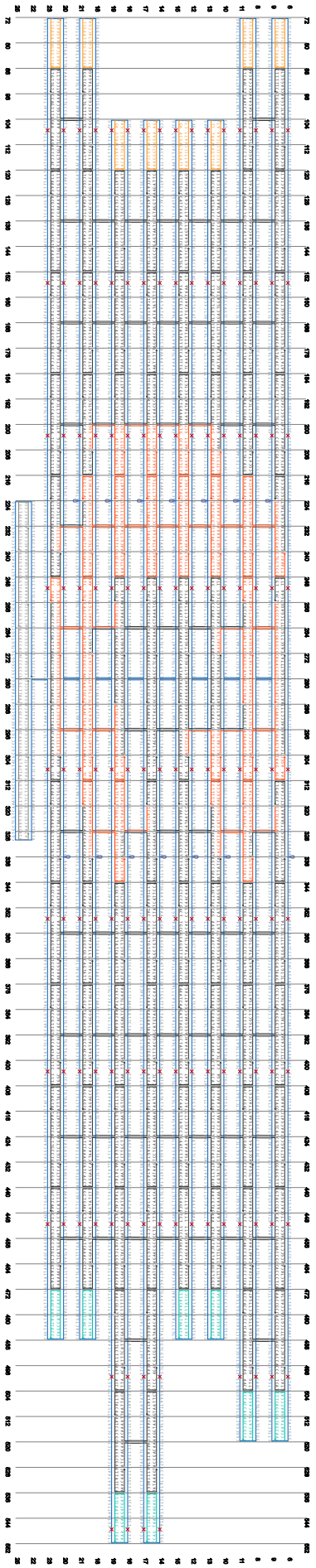
Left Edge

9[72]-6[72]	CTTTGAGGACTAAAGAGCAACGGCTACAGAGG
11[72]-8[72]	CTTGAGATGGTTTAATCGAGTAGTAAATTGGG
13[104]-10[104]	CTTTAAACAGTTCAGAATCCCCTCAAATG
15[104]-12[104]	CGAACGAGTAGATTTTGATTCCCAATTCTG
17[104]-14[104]	CCGGAGACAGTCAAAAAAGGGTGAGAAAGG
19[104]-16[104]	CGAGTAACAACCCGTAACATTAAATGTGAG
21[72]-18[72]	CTCGAATTCGTAATCAATCCCCGGGTACCGAG
23[72]-20[72]	CCCGAGATAGGGTTGATAAATCAAAGAATAG

Hairpin-labeled staples (hairpin sequence in lowercase)

9[256]-11[255]	TAGCCGGAGAACTGAtcctctttttgaggaacaagttttcttgtCCAACTTTTACATAA
8[303]-6[304]	ATGCCCCCGAAAGTAtcctctttttgaggaacaagttttcttgtTTAAGAGGAAAAGGC
10[303]-8[304]	ACCAATGAACAGAATtcctctttttgaggaacaagttttcttgtCAAGTTTGACAGTTA
12[207]-10[208]	AATTGCTAAGATTAAtcctctttttgaggaacaagttttcttgtGAGGAAGGCTTTTGC
12[303]-10[304]	AGCCCAATCTATCTTtcctctttttgaggaacaagttttcttgtACCGAAGCAAACGTC
14[207]-12[208]	CCTTTATGTGGCATCtcctctttttgaggaacaagttttcttgtAATTCTATAGAGCTT
16[207]-14[208]	TTTTTGTAGGCTATCtcctctttttgaggaacaagttttcttgtAGGTCATGGGAGAAG
19[256]-21[255]	GCCGGAAAGCAACTGtcctctttttgaggaacaagttttcttgtTTGGGAAGGTCGTGC
18[207]-16[208]	GAAAGGGTGCATCTGtcctctttttgaggaacaagttttcttgtCCAGTTTGCATTAAA
18[303]-16[304]	GGAATTATTCATCAAtcctctttttgaggaacaagttttcttgtTATAATCCAATGCTG
21[256]-23[255]	CAGCTGCAGTTTGCGtcctctttttgaggaacaagttttcttgtTATTGGGCCCCTAAA
20[303]-18[304]	TATTTTTGAGCCCTAtcctctttttgaggaacaagttttcttgtAAACATCGAAGGAGC
9[224]-11[223]	TGTCGAAAACGGTGTAtcctctttttgaggaacaagttttcttgtCAGACCAGAGAAAGAT
9[288]-8[272]	TGAAACATTGCCTATTtcctctttttgaggaacaagttttcttgtTCGGAACCCAATCATA
9[320]-11[319]	CCTCAAGAAACAGTGCtcctctttttgaggaacaagttttcttgtCCGTATAACCTTTAGC
8[239]-6[240]	ACAGATGATCCGCGACtcctctttttgaggaacaagttttcttgtCTGCTCCACTTAAACA
8[271]-9[287]	AGGGAACCACGAGGCGtcctctttttgaggaacaagttttcttgtCAGACGGTTATTATTC
11[224]-13[223]	AATACCACACCAAAATtcctctttttgaggaacaagttttcttgtAGCGAGAGCCCGAAAG
11[320]-13[319]	GTCAGACTCCATTAGCtcctctttttgaggaacaagttttcttgtAAGGCCGGCCTTTTTA
10[239]-8[240]	CGATAAAAATTCAACTtcctctttttgaggaacaagttttcttgtAATGCAGAGAAAGAGG
10[271]-11[287]	AACACTATGGAATTACtcctctttttgaggaacaagttttcttgtGAGGCATAACCGTAAT
10[335]-8[336]	CACCATTAGTAGCGCGtcctctttttgaggaacaagttttcttgtTTTTCATCGGTCAGTG
13[224]-15[223]	TATCGCGTATTTTTGCtcctctttttgaggaacaagttttcttgtGGATGGCTCTAATAGT
12[239]-10[240]	AAGAGGTCTTTAATTCtcctctttttgaggaacaagttttcttgtGAGCTTCACCAGACGA
15[224]-17[223]	TAACATCCCCTGTAATtcctctttttgaggaacaagttttcttgtACTTTTGCTGCCTGAG
14[239]-12[240]	ATTATGACAATAAATCtcctctttttgaggaacaagttttcttgtATACAGGCCTTTTGAT
17[224]-19[223]	TCGTAAAATATTTTGTtcctctttttgaggaacaagttttcttgtTAAAATTCGAGGGGAC
17[320]-19[319]	TCATCTTCTATATAACtcctctttttgaggaacaagttttcttgtTATATGTATGATTGTT
16[239]-14[240]	AACGTTAACTAGCATGtcctctttttgaggaacaagttttcttgtTCAATCATCCAAAAAC
19[224]-21[223]	CTCCAGCCCTCTTCGCtcctctttttgaggaacaagttttcttgtTATTACGCGCGCTCAC
19[288]-18[272]	ATGGCAATCATCATATtcctctttttgaggaacaagttttcttgtTCCTGATTTCGCCATT
19[320]-21[319]	TGGATTATACAAAGAAtcctctttttgaggaacaagttttcttgtACCACCAGCCATTAAA
18[239]-16[240]	GTGCGGGCAGCTTTCCtcctctttttgaggaacaagttttcttgtGGCACCGCAAATTGTA
18[335]-16[336]	TTTGCGGAACTTCTGAtcctctttttgaggaacaagttttcttgtATAATGGAAGAGTCAA
21[288]-20[272]	GAACTGATAATGGCTAtcctctttttgaggaacaagttttcttgtTTAGTCTTCGCGCGGG
20[239]-18[240]	GGTTTTTCTCCAGTCGtcctctttttgaggaacaagttttcttgtGGAAACCTGGCGATCG
20[271]-21[287]	GAGAGGCGTTAATGAAtcctctttttgaggaacaagttttcttgtTCGGCCAATAATGCGC

Positions of dumbbells are indicated by the red-colored staple strands (28-nt dumbbell hairpin sequences are inserted in the middle of each staple strand. See sequence list.)



S5.7. "D" origami

Core

Seq name	Sequence
6[111]-9[95]	CGGAACCTATTATTCTGAAACATATATAAGT
6[143]-9[127]	CCGTATAAACAGTTAATGCCCCCTCGTACTCA
6[175]-9[159]	TTAACGGGGTCAGTGCCTTGAGTGAACCGC
6[207]-9[191]	TGATGATACAGGAGTGTACTGGTAGCCACCA
6[239]-9[223]	TTACCGTTCCAGTAAGCGTCATACCCCAATAG
6[271]-9[255]	AAAGCCAGAATGGAAAGCGCAGTTCGTCAC
6[303]-6[272]	CCTTGATATTCACAAACAAATAAATCCTCATT
6[335]-9[319]	GACAGGAGGTTGAGGCAGGTCAGATAGCAAGG
6[367]-9[351]	TCAGAGCCACCACCCTCAGAGCCTCGATAGC
6[399]-9[383]	CCTCAGAGCCGCCACCCTCAGAACAATCAAGT
6[431]-9[415]	CCGGAACCAGAGCCACCACCGGAACGCGTTTT
6[79]-9[63]	TAAGAGGCTGAGACTCCTCAAGAACCAGGC
8[111]-6[112]	TTTTCACGAATAGGTGTATCACGCCTATTT
8[143]-6[144]	AGGAACAAAGTACCGCCACCCTCAAACAGTGC
8[175]-6[176]	TTTCAACAGAACCGCCACCCTCAGAATAAGTT
8[335]-6[336]	ATTCATTAGTCACCAATGAAACCAGCCACCAG
8[367]-6[368]	TCAACCGATAATCAGTAGCGACAGCGCCACCC
8[399]-6[400]	GTTTACCAAGCGTCAGACTGTAGCCGCCTC
8[431]-6[432]	ATTTTGTCTTTTCGGTCATAGCCCCAAAATCA
8[79]-6[80]	TCCAAAAGGCCGTCGAGAGGGTTGGAAAGTAT
9[128]-11[127]	GGAGGTTTCTAAAGGAATTGCGAACGATATAT
9[160]-11[159]	CACCCTCAGTTTCAGCGGAGTGATTAAAGG
9[352]-11[351]	AGCACCGTTGAGGGAGGGAAGGTAATAACG
9[384]-11[383]	TTGCCTTTGCGCCAAAGACAAAAGATTAAGAC
9[416]-11[415]	CATCGGCAACAATCAATAGAAAATAAACGTAG
9[64]-11[63]	GGATAAGTGAGCCTTTAATTGTACTTAAAC
9[96]-11[95]	ATAGCCCGGTTGAAAATCTCCAAAACAATGAC
10[111]-8[112]	TCCATGTTCGCCCACGCATAACTAATAATT
10[143]-8[144]	CCTGATAATGAGGCTTGCAGGGAGGAATAGAA
10[175]-8[176]	AAACAAAGGCGGGATCGTCACCCTTAAACAAC
10[367]-8[368]	TGAGCGCTCAAAAGAACTGGCATGGGCGACAT
10[399]-8[400]	ACTGAACAACGCAGTATGTTAGCTCATATG
10[431]-8[432]	AAACAGGGTACATAAAGGTGGCAAATAAGTTT
10[47]-13[31]	CTTTGAAAGAGGACAGATGAACGGCTTCATCA
10[79]-8[80]	CGGTCAATACCGATAGTTGCGCCGAAAAAGGC
11[128]-13[127]	TCGGTCGCATTGTGTCGAAATCCGAGAAACAC
11[160]-13[159]	CCGCTTTTTACAACGGAGATTTGGAGATGG
11[192]-13[191]	AAAGACAGCCCCCAGCGATTATACGAATTACC
11[256]-13[255]	TGAGGAAGCTACGAAGGCACCAACTACGTT
11[288]-10[272]	GAAAAGTACTATCTTACCGAAGCCAAAATACG
11[384]-13[383]	TCCTTATTCCCTGAACAAAGTCAGAAAATAAA
11[416]-13[415]	AAAATACAAAGCGCATTAGACGGGCAAATAAG
11[64]-13[63]	AGCTTGATCATAAGGGAACCGAATATTCAT
11[96]-13[95]	AACAACCATACTTAGCCGGAACGAGCTCATTC
12[111]-10[112]	AAAAGAAAGGCTTGCCCTGACGCGACCTGC
12[143]-10[144]	CGATAAAAGTAGTAAATTGGGCTTTATCATCG
12[175]-10[176]	AACACTATCAACTTTAATCATTGTCAAGCGCG
12[239]-10[240]	AATACCACCGTTGGGAAGAAAAATCCTAAAAC
12[271]-13[287]	AGGTAGAAGAACTAACGGAACAACCAAGATTA
12[303]-10[304]	GCGGGAGGTTTGCACCCAGCTACATGAAAT
12[367]-10[368]	AGGAATCACTAATTTGCCAGTTACAGGGTAAT
12[399]-10[400]	GAGAACAATTATTTATCCCAATCAGAATTA
12[431]-10[432]	AACGGGTATTTTGTTTAACGTCAATAACATAA
12[47]-10[48]	CGGAATCGCTTGACAAGAACCGGACTGACCAA
12[79]-10[80]	AAATGTTTTCAACGTAACAAAGCTGGCGCAGA
13[128]-15[127]	CAGAACGAACCAAAATAGCGAGAGCCCGAAAG
13[160]-15[159]	TTTAATTTCATAACCCTCGTTTAGAGCTTC
13[192]-15[191]	TTATGCGAGGAATTACGAGGCATACTCCAACA
13[256]-15[255]	AATAAAACAGATTCATCAGTTGATGTTTTA
13[288]-12[272]	GTTGCTATTTTTGAAGCCTTAAATATTATTAC
13[32]-15[31]	AGAGTAATTCATAAATATTCATTGAAAACGAG
13[384]-15[383]	CAGCCATAGCAAGCCGTTTTTATTAACAATAG
13[416]-15[415]	AAACGATTTTAAACCAAGTACCGCATATCCCA
13[64]-15[63]	TACCCAAAAGACTGGATAGCGTCTCTTTAC
13[96]-15[95]	AGTGAATAGTTTTGCCAGAGGGGGAGCGGATT
14[111]-12[112]	TAACATCAAGATTAAGAGGAAGGCTTTTGC
14[143]-12[144]	GGTGGCATTATCGCGTTTTAATTCCCAGACGA
14[175]-12[176]	CTATATTTCCAGACCGGAAGCAAAGTAAGAGC
14[239]-12[240]	GCGAACGAATGCTGTAGCTCAACAGATTTAGG
14[271]-15[287]	TCATTCCAACTAAAGTACGGTGTCGAGAATAT
14[303]-12[304]	GAGGCATTGACAAAAGGTAAAGTCCGACTT
14[367]-12[368]	AAAGCCAAAACGCGCCTGTTTATCTTCATCGT
14[399]-12[400]	ATCATATGTGAACAAGAAAAATAACTCATC
14[431]-12[432]	AATCATAATACGAGCATGTAGAAAATTCCAAG
15[128]-17[127]	ACTTCAAACAATTCTACTAATAGTCCAAAAAC
15[160]-17[159]	AAAGCGAATCATTTGGGGCGCGAGGGAGAA
15[256]-17[255]	AATATGCATATAACAGTTGATTCGAAAGGC
15[288]-14[272]	AAAGTACCTTCGAGCCAGTAATAATGGAAGTT
15[32]-12[48]	AATGACCATAAATCAAAAATCAGGCAATACTG
15[384]-17[383]	ATAAGTCCCGTTATACAAATTCTTTTTCAAAT
15[416]-17[415]	TCCTAATTTTACTAGAAAAAGCCTTGACCTAA
15[64]-12[80]	CCTGACTATTATAGTCAGAAGCAATAATAGTA
15[96]-17[95]	GCATCAAACAATAAATCATACAGGAGCCTCAG
16[111]-14[112]	ATAATCAGCTAAATCGGTTGTAAGTAGCAT
16[143]-14[144]	AACTAGCACCTGTAATACTTTTGCGCTGAAAA
16[175]-14[176]	AAACAAGATTCAACGCAAGGATAACTGTTTAG
16[239]-14[240]	AGAGGGTAAGATTCAAAAGGGTGACCAATTCT
16[271]-17[287]	TCAACCGTGTCAAATCACCATCAACTACCTTT
16[303]-14[304]	TATCAAAACGGCTTAGGTTGGGTTTAGGCA
16[367]-14[368]	AGTACATAGAACGCGAGAAAACTTACCAGTAT
16[399]-14[400]	ATTTCATTGTTAATTTCATCTTCGTTTAGT
16[431]-14[432]	ATCAAGAAGTTTGAAATACCGACCAACACCGG
17[128]-19[127]	ATTATGACTGTCAATCATATGTACAATTTTTG
17[160]-19[159]	GCCTTTATGAATCGATGAACGGTATAGGAA
17[224]-19[223]	GTAGGTAAGCTATTTTTGAGAGATAAATGTGA
17[256]-19[255]	CGGAGACATCTAGCTGATAAATTTGGGATA
17[288]-16[272]	TTAACCTCTCATAGGTCTGAGAGATATGATAT
17[352]-19[351]	AGACAAAAATCAATATATGTGAGTAACAGT
17[384]-19[383]	ATATTTTATGAATTACCTTTTTTATAACGGAT
17[416]-19[415]	ATTTAATGAACAAAATTAATTACAAGTTACAA
17[96]-19[95]	AGCATAAAGAAAAGCCCCAAAAACAAACGTTA
18[111]-16[112]	CTCTTCGTTAAAATTCGCATTACCCGGTTG
18[143]-16[144]	CGCAACTGGCTCATTTTTTAACCAAATCGTAA
18[175]-16[176]	CCAGGCAAAAAATAATTCGCGTCTTCTGGAGC
18[239]-16[240]	AGTATCGGGGCGGATTGACCGTAAAATGCCGG
18[271]-19[287]	CGTGCATCTGGTGTAGATGGGCGCGCACGTAA
18[367]-16[368]	TATTCCTGACATCGGGAGAAACAAATGGAAAC
18[399]-16[400]	AGAAACCATTGCTTTGAATACCATTTAACA
18[431]-16[432]	GTTTGAGTAGAGGCGAATTATTCAAAACAAAC
18[47]-21[31]	CGCCAGGGTTTTCCCAGTCACGACCATGCCTG
18[79]-21[63]	GGATGTGCTGCAAGGCGATTAAGGGTACCG
19[128]-21[127]	TTAAATCATTGGGAAGGGCGATCGTCACAATT
19[160]-21[159]	CGCCATCAAGCGCCATTCGCCATATAAAGT
19[352]-21[351]	ACCTTTTATTATCAGATGATGGTTAGGAGC
19[384]-21[383]	TCGCCTGACCAGAAGGAGCGGAATCGTCAATA
19[416]-21[415]	AATCGCGCAACATTATCATTTTGCAGTATTAG
19[96]-21[95]	ATATTTTGCTATTACGCCAGCTGGTAGCTGTT
20[111]-18[112]	TCCCTTAGAAATTGTTATCCGCGTGCGGGC
20[143]-18[144]	CCTGTTTGCATACGAGCCGGAAGCTCAGGCTG
20[335]-18[336]	AAATGAAAGTTATCTAAAATATCTCAATTCAT
20[367]-18[368]	AACACCGCACTAATAGATTAGAGCTATCATCA
20[399]-18[400]	GAAGATAAATTTGAGGATTTAGAGGAACAA
20[431]-18[432]	GCCATTAAAACAATTCGACAACTCTTTTAAAA
20[47]-18[48]	AGTCCACTCTCTAGAGGATCCCCGTTGGGTAA
20[79]-18[80]	AGGGTTGATTCGTAATCATGGTCACGAAAGGG
21[128]-23[127]	CCACACAAATGGTGGTTCCGAAATCCCGATTT
21[32]-23[31]	CAGGTCGAATTAAAGAACGTGGACCAGGGCGA
21[320]-23[319]	TGAGGAAGAATCTAAAGCATCACCTTTACATT
21[352]-23[351]	ACTAACACTGCAACAGTGCCACAGTAATAA
21[384]-23[383]	GATAATACAACAGAGGTGAGGCGGATAGAACC
21[416]-23[415]	ACTTTACAAAATACCGAACGAACCTACGTGGC
21[64]-23[63]	AGCTCGAAGTGTTGTTCCAGTTTCAAATCA
21[96]-23[95]	TCCTGTGTTAAATCAAAAGAATAGAAAGCACT
23[128]-20[144]	AGAGCTTGACGGGGAAAGCCGGCGGCGAAAAT
23[160]-20[176]	CGAGAAAGGAAGGGAAGAAAGCGAGCAAGCGG
23[192]-20[208]	GGCGCTAGGGCGGCCTTGCTGGTCCCTTCAC
23[224]-20[240]	GAACAATATTACCGCCAGCCATTGTTTTCACC
23[256]-23[287]	AAAAACGCTCATGGAAATACCTACATTTTGAC
23[288]-20[304]	GCTCAATCGTCTGAAATGGATTATTGCTGA
23[32]-20[48]	TGGCCCACTACGTGAACCATCACCGGAACAAG
23[320]-20[336]	GGCAGATTCACCAGTCACACGACCGCTGAGAG
23[352]-20[368]	AAGGGACATTCTGGCCAACAGAGTCAGTATT
23[384]-20[400]	CTTCTGACCTGAAAGCGTAAGAAACCAGCA
23[416]-20[432]	ACAGACAATATTTTTGAATGGCTAAAAACATC
23[64]-20[80]	AGTTTTTTGGGGTCGAGGTGCCGTCCCGAGAT
23[96]-20[112]	AAATCGGAACCCTAAAGGGAGCCCGGCAAAA

Right Edge

6[455]-9[455]	CCATCTTTTCATAATCCTTATTAGCGTTTG
8[455]-11[455]	CAAAGACACCACGGACATATAAAAGAAACG
10[455]-13[455]	CCTTTACAGAGAGAAAAATGAAAATAGCAG
12[455]-15[455]	CTGTCTTTCCTTATCCCAATCAATAATCGG
14[455]-17[455]	CGTTAAATAAGAATAGTGTGATAAATAAGG
16[455]-19[455]	CAAAAGAAGATGATGTTTCAATTACCTGAG
18[455]-21[455]	CCCGAACGTTATTAAGTATTAAATCCTTTG
20[455]-23[455]	CGAACTGATAGCCCTTTAGTCTTTAATGCG

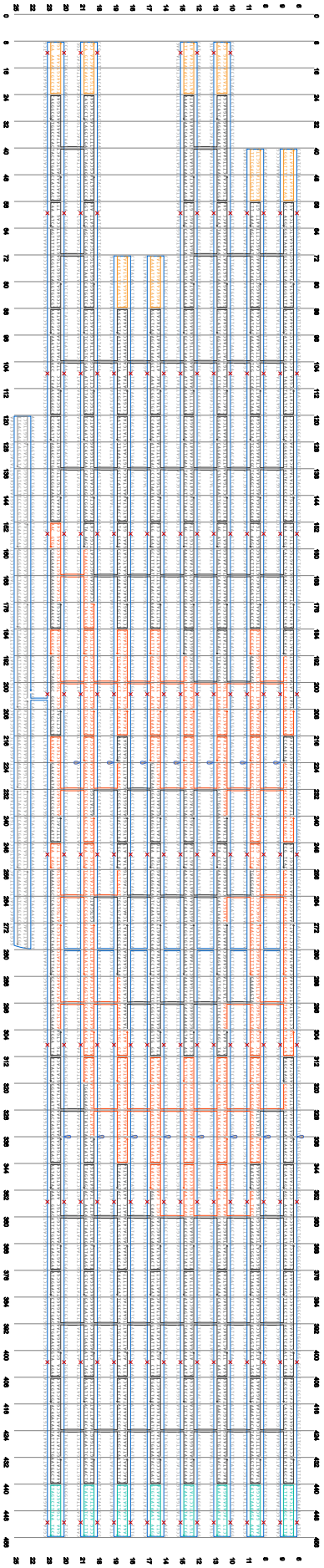
Left Edge

9[40]-6[40]	CGGGGTTTTGCTCAGTGAAGGATTAGGATTAG
11[40]-8[40]	CTTTCGAGGTGAATTTTCGGTTTATCAGCTTG
13[8]-10[8]	CATAGGCTGGCTGACTGTACAGACCAGGCG
15[8]-12[8]	CTTTAAACAGTTCAGAATCCCCCTCAAATG
17[72]-14[72]	CAAAATTAAGCAATAACAAGGCAAAGAATTAG
19[72]-16[72]	CAAATATTTAAATTGTAGGAAGATTGTATAAG
21[8]-18[8]	CCAGTGCCAAGCTTGGTTGTAAAACGACGG
23[8]-20[8]	CGAAAAACCGTCTATTCCAACGTCAAAGGG

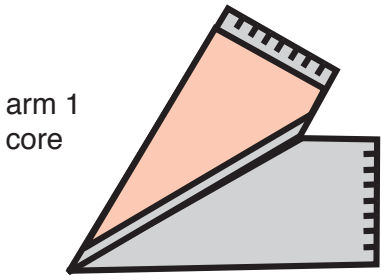
Hairpin-labeled staples (hairpin sequence in lowercase)

10[207]-8[208]	TCTTTGACATCGGAAtcctcttttgaggaacaagttttcttgtCGAGGGTTAGTAAAT
10[239]-8[240]	GAAAGAGGTTTGAGGAtcctcttttgaggaacaagttttcttgtCTAAAGACACGATCTA
10[271]-11[287]	TAATGCCATTTCCATTtcctcttttgaggaacaagttttcttgtAAACGGGTCTTTTTAA
10[303]-8[304]	AGCAATAGAGCAGATtcctcttttgaggaacaagttttcttgtAGCCGAACGTCACCG
10[335]-8[336]	GCCCAATAAACCGAGGtcctcttttgaggaacaagttttcttgtAAACGCAATAAATATT
11[224]-13[223]	ACAGAGGCCAAAAGAAtcctcttttgaggaacaagttttcttgtTACACTAACATTATAC
11[320]-13[319]	CAGAAGGAATAAGAGCtcctcttttgaggaacaagttttcttgtAAGAAACAAATTTTAT
11[352]-13[351]	GAATACCAATATCAGtcctcttttgaggaacaagttttcttgtAGAGATAGCGTCTTT
12[207]-10[208]	GCCAAAATTTTAAGAtcctcttttgaggaacaagttttcttgtACTGGCTAACACTCA
12[335]-10[336]	ACGCGAGGTTACCAACtcctcttttgaggaacaagttttcttgtGCTAACGAACCCACAA
13[224]-15[223]	AGTCAGGAATTCAACTtcctcttttgaggaacaagttttcttgtAATGCAGAAATTGCTC
13[320]-15[319]	CCTGAATCCGTTTTAGtcctcttttgaggaacaagttttcttgtCGAACCTCAATTCTGT
13[352]-15[351]	CCAGAGCTTACCGCGtcctcttttgaggaacaagttttcttgtCCCAATAGTTCAGCT
14[207]-12[208]	ACATTTCTTAGAGAGtcctcttttgaggaacaagttttcttgtTACCTTTTACATAAC
14[335]-12[336]	TATTTAACCGACAATAtcctcttttgaggaacaagttttcttgtAACAACATGCAAGCAA
15[192]-17[191]	GGTCAGGAGCAAATGGtcctcttttgaggaacaagttttcttgtTCAATAACAAATTTTT
15[224]-17[223]	TGAATATAGTAGATTTtcctcttttgaggaacaagttttcttgtAGTTTGACCAATGCCT
15[320]-17[319]	CCAGACGAAACGCCAAtcctcttttgaggaacaagttttcttgtCATGTAATTATATAAC
15[352]-17[351]	AATGCAGCGCTCAACtcctcttttgaggaacaagttttcttgtAGTAGGGCAATCGCA
16[207]-14[208]	GCTATCACATATATTtcctcttttgaggaacaagttttcttgtTTAAATGCATTAGAT
16[335]-14[336]	TTAAGACGAATGCTGAtcctcttttgaggaacaagttttcttgtTGCAAATCCTTAATTG
17[192]-19[191]	AGAACCCTGGTCATTGtcctcttttgaggaacaagttttcttgtCCTGAGAGGGCCTTCC
17[320]-19[319]	TATATGTACTGAGAAGtcctcttttgaggaacaagttttcttgtAGTCAATAGATTTTCA
18[207]-16[208]	AGCTTTCGCTTTCATtcctcttttgaggaacaagttttcttgtCAACATTCTACAAAG
18[303]-16[304]	GAACCTACTAAAGAAtcctcttttgaggaacaagttttcttgtATTGCGTAGTGAATT
18[335]-16[336]	TGGATTATGTCAGATGtcctcttttgaggaacaagttttcttgtAATATACAGTGAATAA
19[192]-21[191]	TGTAGCCACGGCACCGtcctcttttgaggaacaagttttcttgtCTTCTGGTTGAGCTAA
19[224]-21[223]	GAACAAACCCTCAGGAtcctcttttgaggaacaagttttcttgtAGATCGCAACTGCCCG
19[256]-21[255]	GGTCACGTTGCCAGTtcctcttttgaggaacaagttttcttgtTTGAGGGGTAATGAA
19[288]-18[272]	AACAGAAACATATCAAtcctcttttgaggaacaagttttcttgtAATTATTTATCGTAAC
19[320]-21[319]	GGTTTAACACTTCTGAtcctcttttgaggaacaagttttcttgtATAATGGAAAAGGAAT
20[175]-18[176]	TCCACGCTTGGGGTGCtcctcttttgaggaacaagttttcttgtCTAATGAGGCCGGAAA
20[207]-18[208]	CGCCTGGAATTGCGTtcctcttttgaggaacaagttttcttgtTGCGCTCCTCCAGCC
20[239]-18[240]	AGTGAGACGTCGTGCCtcctcttttgaggaacaagttttcttgtAGCTGCATACGACGAC
20[271]-21[287]	ATTGGGCGCGCGCGGGtcctcttttgaggaacaagttttcttgtGAGAGGCGAATATCTG
20[303]-18[304]	ACCTCAAAGCAAATCtcctcttttgaggaacaagttttcttgtAACAGTTGAGGGTTA
21[160]-23[159]	GTAAAGCCGGTTTGCtcctcttttgaggaacaagttttcttgtCCCAGCAGAACGTGG
21[192]-23[191]	CTCACATTCCCTGAGAtcctcttttgaggaacaagttttcttgtGAGTTGCAAAGGAGCG
21[224]-23[223]	GGAAACCTGGGCAACAtcctcttttgaggaacaagttttcttgtGCTGATTGAATATCCA
21[256]-23[255]	TCGGCCAACCAGGGTtcctcttttgaggaacaagttttcttgtGGTTTTTCCAACAGG
21[288]-20[272]	GTCAGTTGTATCAAACtcctcttttgaggaacaagttttcttgtCCTCAATCGTTTGCGT
8[207]-6[208]	GAATTTTTTCAGGGAtcctcttttgaggaacaagttttcttgtTAGCAAGATGGCTTT
8[239]-6[240]	AAGTTTTGACCGTAACtcctcttttgaggaacaagttttcttgtACTGAGTTCTCTGAAT
8[271]-9[287]	AGACAGCCACTACAACtcctcttttgaggaacaagttttcttgtGCCTGTAGCCAGCAAA
8[303]-6[304]	ACTTGAGCTAGCACCtcctcttttgaggaacaagttttcttgtATTACCATCGATTGG
9[192]-11[191]	CCCTCATTCTGTATGGtcctcttttgaggaacaagttttcttgtGATTTTGCCAGCAGCG
9[224]-11[223]	ACCCATGTTCGTCTTTtcctcttttgaggaacaagttttcttgtCCAGACGTAGCAACGG
9[256]-11[255]	CAGTACAACTCATAGtcctcttttgaggaacaagttttcttgtTTAGCGTATTTTTCA
9[288]-8[272]	ATCACCAGCATTTGGGtcctcttttgaggaacaagttttcttgtAATTAGAGCATTCCAC
9[320]-11[319]	CCGGAAACAAGGTGAAtcctcttttgaggaacaagttttcttgtTTATCACCAAAGTTAC

Positions of dumbbells are indicated by the red-colored staple strands (28-nt dumbbell hairpin sequences are inserted in the middle of each staple strand. See sequence list.)



S5.8. 60° corner origami with straight edges



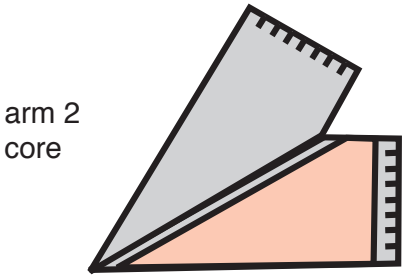
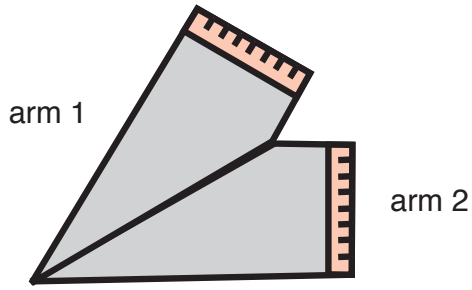
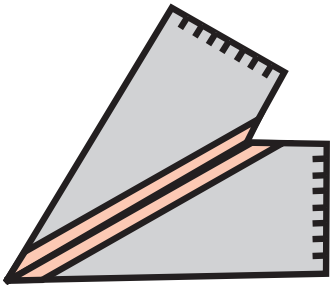


plate flat-v-top-core (arm1)
A1,flatv-h1-b32,AATATTTTATGGGATAGGTCACGTTCTGCCAG
B1,flatv-h3-b32,TTTGAGGGAACGACGGCCAGTGCCGGATCCCC
C1,flatv-h5-b32,GGGTACCGGCAGCAAGCGGTCCACTTTGATGG
D1,flatv-h7-b32,TGGTTCCGTCAATCGTCTGAAATGCACGACCA
E1,flatv-h9-b32,GTAATAAAAACAATTCGACAACTCTTTTAAAA
F1,flatv-h11-b32,GTTTGAGTTGAAAACATAGCGATAAGTGAATT
G1,flatv-h13-b32,TATCAAAACCAAGTACCGCACTCATCGTAGGAATCATTAC
A2,flatv-h1-b48,TCGCATTACCCGGTTGATAATCAGAAAAGCCCCAAAAACATAAACGTT
B2,flatv-h3-b48,CAGTATCGCGGCGGATTGACCGTAGTTAAAAT
C2,flatv-h5-b48,TTCGTAATAGTCACGACGTTGTAAGACGACGA
D2,flatv-h7-b48,AAAATCCCTGGCCCTGAGAGAGTTAGCTCGAA
E2,flatv-h9-b48,TCTGGCCAACCTACATTTTGACGCAAATCGGC
F2,flatv-h11-b48,CATTTTGCAGTATTAGACTTTACAAGGGACAT
G2,flatv-h13-b48,CTGAGAGATTTCCCTTAGAATCCTAACATTAT
H2,flatv-h15-b40,CGCGCCCAATAGCAAGAAGAACGGGTATTAAATCATAGGT
A3,flatv-h1-b64,GTTAAATCCTCCGTGGGAACAAAGCCTCAG
B3,flatv-h3-b64,GAAGATCGCGCCAGGGTTTTCCCCATGGTC
C3,flatv-h5-b64,ATAGCTGTTTGCCCTTCACCGCCTTATAAA
D3,flatv-h7-b64,TCAAAAGAAACGCTCATGGAAATACAGAGA
E3,flatv-h9-b64,TAGAACCCATTTGAGGATTTAGAGGAACAA
F3,flatv-h11-b64,AGAAACCATCGCTATTAATTAATCTACCTT
G3,flatv-h13-b64,TTTAACCTTTTCCTTATCATTCCCAAATCAGATATAGA
A4,flatv-h1-b80,TTTTAACCATCGTAAAACTAGCATGTCAATCATATGTACAATTTTT
B4,flatv-h3-b80,CCAGCTTTTAAATGTGGTCGGATTAGCTCATT
C4,flatv-h5-b80,TGAAATTGCTGCAAGGTTGGGTAACACTCCAG
D4,flatv-h7-b80,AGATAGGGGGCGCCAGACAGCTGATTCCTGTG
E4,flatv-h9-b80,TGAAAGCGCTAGGGCGACAGGAAAATAGCCCG
F4,flatv-h11-b80,AGCGGAATAATAGATTGATAATACTTCTGACC
G4,flatv-h13-b80,GGTTGGGTATAAATCAGTAAATCGCCAGAAGG
H4,flatv-h15-b72,AGGCTTATCGGTATTCAAGAAAAACGGCTGTCCCGGCTTA
A5,flatv-h3-b96,GCTTCTGGGCGAAAGGGGGATGTGTTATCCGC
B5,flatv-h5-b96,TCACAATTGGCGGTTTGCGTATTGTTGAGTGT
C5,flatv-h7-b96,TGTTCCAGGCGAAAGGAGCGGGCGTAAGAATA
D5,flatv-h9-b96,CGTGGCACAGGAGCACTAACAACTTATCATCA
E5,flatv-h11-b96,TATTCCTGTTAATGGAAACAGTACTATATAAC
F5,flatv-h13-b96,TATATGTATAGATAAGTCCTGAACTAAGAACGCGAGGCGT
A6,flatv-h5-b112,ACATACGACTATTACGCCAGCTGTGCCGGA
B6,flatv-h7-b112,CAAGAGTCCAACGCGCGGGGAGACCACACA
C6,flatv-h9-b112,ATTTTTGAAAGGAAGGGAAGAAATTTGGAA
D6,flatv-h11-b112,GATGATGGATCTAAAATATCTTTAGACAAT
E6,flatv-h13-b112,ATGCAAATTTTGAATTACCTTTTATTATCA
F6,flatv-h15-b104,TTTAGCGAACCTCCCCCTGTTTATCAACAAAATGCTG
A7,flatv-h3-b120,AAAGCGCCATTCGCCAGGTGCGGGCCTCTTCGGCCGGAAG
B7,flatv-h5-b128,CATAAAGTGCATTAATGAATCGGCCACTATTA
C7,flatv-h7-b128,AAGAACGTGGCGAACGTGGCGAGAATGGCTAT
D7,flatv-h9-b128,TAGTCTTTGGAATTGAGGAAGGTTCAATTCAT
E7,flatv-h11-b128,CAATATAAACATTTAACAATTTCACCAATCGC
F7,flatv-h13-b128,AAGACAAAGCTAATGCAGAACGCGGACTTGCGGGAGGTTT
A8,flatv-h7-b144,ACGTCAAAACCTGTCGTGCCAGCTGTAAAGCC
B8,flatv-h9-b144,AACTGATACTTGACGGGGAAAGCCGGACTCCA
C8,flatv-h11-b144,GTTTGGATAAATCAACAGTTGAAAAATGCGCG
D8,flatv-h13-b144,GAAAACTTGAAAACAAAATTAATTTCCTGATT
E8,flatv-h15-b136,TGAAGCCTTAAATCAAATAAACAACATGTTCAGAACGCGA
A9,flatv-h5-b152,CTAATGAGTGAGCTACTTTCCAGTCGGGAAGGGCGAA
B9,flatv-h7-b160,AAACCGTCGCCCCCGATTTAGAGGCCCTAA
C9,flatv-h9-b160,AACATCGCATCTGGTCAGTTGGCTATACTT
D9,flatv-h11-b160,CTGAATAATGAAACAAACATCAATTTCAAA
E9,flatv-h13-b160,TATATTTTGTCCAGACGACGACAGATTAGTTGCTATTT
A10,flatv-h9-b176,ATACCGAACGGAACCCTAAAGGGATATCAGGG
B10,flatv-h11-b176,TTAGAACCCAAACCCTCAATCAATCATTAAAA
C10,flatv-h13-b176,TCATCTTCGAGCAAAAGAAGATGATGGAAGGG
D10,flatv-h15-b168,TGCACCCAGCTACAATAAGGTAAAGTAATTCTAGTTAATT
E10,flatv-h7-b184,CACTACGTGAACCATCCCGTAAAGCACTAAATCGAACCAC
F10,flatv-h9-b192,CAGCAGAAGCTGAACCTCAAATATTACCATAT
G10,flatv-h11-b192,CAAAATTATTCATTTCAATTACCTTGACCTAA
H10,flatv-h13-b192,ATTTAATGTATAAAGTACCGACAATTTATCCTGAATCTTA
A11,flatv-h11-b208,GTAAAACACTAAAGCATCACCTTGATAAAA
B11,flatv-h13-b208,ATACCGACCGCAGAGGCGAATTATTTGCAC
C11,flatv-h15-b200,CCAACGCTAACGAGCCCAGTAATAAGAGAAGTTTGAA
D11,flatv-h9-b216,AGGCGGTCAGTATTAAGCAGCAAATGAAAAATGAAATAAA
E11,flatv-h11-b224,GAAATTGCCCAAGTTACAAAATCGCGTGTGAT
F11,flatv-h13-b224,AAATAAGGCAGAGGCATTTTCGAGGTCTTTCCAGAGCCTA
G11,flatv-h13-b240,AAGAATAACTGATTGCTTTGAATAGTAGATTT
H11,flatv-h15-b232,ATTTGCCAGTTACAAACAACATGTAATTTAGGCGTTAAAT
A12,flatv-h11-b248,AACGTCAGATGAATACAATAACGGATTCGCACACCGG
B12,flatv-h13-b256,AATCATAACATATTTAACAACGCATAAACAGCCATATT
C12,flatv-h15-b264,ATTTATCCCAATCCAACTTAATTGAGAATCGCTTACTAGA
D12,flatv-h13-b280,GTTTAGTATCATATGCACGCTCAACAGTAGGGATAAGAAA

plate flat-v-bottom-core (arm 2)
A1,flatv-h16-b39,ATACATAAAGGTGGCATAAGTTTATTTTGTCAGAATTTAC
B1,flatv-h18-b31,CGTTCCAGAGTGTACTGGTAATAAGTGAGAAT
C1,flatv-h20-b31,AGAAAGGACACGTTGAAAATCTCCCCTGCTCC
D1,flatv-h22-b31,ATGTTACTAGGGAACCGAACTGACATACCACA
E1,flatv-h24-b31,TTCAACTAGGCATAGTAAGAGCAAAATTCGAG
F1,flatv-h26-b31,CTTCAAAGAGGATTAGAGAGTACCAAAGGTGG
G1,flatv-h28-b31,CATCAATTATCATACAGGCAAGGCGTGTAGGT
H1,flatv-h30-b31,AAAGATTCCACCATCAATATGATATTCAACCGTTCTAGCCAATGCCT
A2,flatv-h18-b47,AGTCTCTCAATCAATAGAAAATAAACGTAGAAAATAC
B2,flatv-h20-b47,CAGCGGAGTTTTAACGGGGTCAGAAAGCGC
C2,flatv-h22-b47,TCCGCGAAAAAAAAAGGCTCCAAACAGTTT
D2,flatv-h24-b47,TTTAGGACAACTTTGAAAGAGGTGTCGAAA
E2,flatv-h26-b47,GCGTTTTCACTATCATAACCCTAGTTGAGA
F2,flatv-h28-b47,GAGCTGATTTAATTGCTCCTTTCAAATATC
G2,flatv-h30-b47,GAGTAATAAAGAATTAGCAAAATGGGGCGC
A3,flatv-h16-b71,TACGCAGTATGTTAGCTCATATGGTTTACCAGCATTAAAG
B3,flatv-h18-b63,CCAGAATGGTGCCTTGAGTAACAGTTGCTAAA
C3,flatv-h20-b63,CAACTTTCAAAGGAGCCTTTAATTATCGCCTG
D3,flatv-h22-b63,ATAAATTGACAGATGAACGGTGTAGGTAGAAA
E3,flatv-h24-b63,GATTCATCCGTTTACCAGACGACGGAAGCCCG
F3,flatv-h26-b63,AAAGACTTTGATAAGAGGTCATTTTAGCTATA
G3,flatv-h28-b63,TTTTCATTTTAAGCAATAAAGCCTTCATATAT
H3,flatv-h30-b63,TTTAAATGTGATAAATTAATGCCGGAGAGGGTGTCATTGCAAAATTTT
A4,flatv-h18-b79,TAAATCCTCGCCAAAGGAGGGAGGGATTAAGACTCCTTAT
B4,flatv-h20-b79,ATGGGATTTGCCCGTATTTCGGAACAAACAAA
C4,flatv-h22-b79,TTTGTATCGTATCGGTTACCGATATTTTCTGT
D4,flatv-h24-b79,TTATTACACAGACCAGTAGGCTGGAACGGAGA
E4,flatv-h26-b79,ATTAAGAGATAAAAACTTTTGCAAGAACAACA
F4,flatv-h28-b79,AACCTGTTTTGCGGATTAATGCTGTCAAAAAG
G4,flatv-h30-b79,TAGAACCCCAGAGCATCGGTTGTATGGTCAAT
A5,flatv-h16-b103,CAAAAGAACTGGCATGAAGGTAAATATTGATGGCCTTG
B5,flatv-h18-b95,ATATTCACCTATTATTCTGAAAACGTTAGT
C5,flatv-h20-b95,AAATGAAGTTGCGCCGACAATGCGCGAAAC
D5,flatv-h22-b95,AAAGTACCTGACCTTCATCAAGTAAAACGA
E5,flatv-h24-b95,ACTAACGAAGAAGTTTTGCCAGGCAAAGCG
F5,flatv-h26-b95,GATTGCATAGCTCAACATGTTTGATACATT
A6,flatv-h18-b111,CAGACGATCGGAAATTATTCATTAATAATAACGGAATACC
B6,flatv-h20-b111,CTTTCCAGCATGAAAGTATTAAGAAGGCAGGT
C6,flatv-h22-b111,ATACCAAGACAACAACCATCGCCCTTTGTCGT
D6,flatv-h24-b111,TACGTTAAAGTAATCTTGACAAGACAGCGATT
E6,flatv-h26-b111,AGTCAGAAAGGGGGTAATAGTAAAGAAAATC
F6,flatv-h28-b111,GACCATTATAAATATGCAACTAAAACTATTAT
A7,flatv-h16-b135,AAACCGAGGAAACGCAAAGGTGAATTATCACCCATTGACA
B7,flatv-h18-b127,GGAGGTTGGGCTGAGACTCCTCAACGTAACGA
C7,flatv-h20-b127,TCTAAAGTACGCATAACCGATATAACTCATCT
D7,flatv-h22-b127,TTGACCCCACCGGATATTCATTACGTCAGGAC
E7,flatv-h24-b127,GTTGGGAAATGTTTAGACTGGATATCAGGTCT
F7,flatv-h26-b127,TTACCCTGGTACGGTGTCTGGAAGTCTGCGAACGAGTAGA
A8,flatv-h18-b143,CCGCCAGGTCACCGACTTGAGCAAAGTTACCAGAAGG
B8,flatv-h20-b143,TAGTTAGGAGAAGGATTAGGATCAGAGCCG
C8,flatv-h22-b143,CTAAAACTTCGGTCGCTGAGGCAGCCCTCA
D8,flatv-h24-b143,TATACCACCAAATCAACGTAACAGAATACA
E8,flatv-h26-b143,TCAAAAAGCGTCCAATACTGCGTGGCTCAT
A9,flatv-h16-b167,AAGCAGATAGCCGAACCATTTGGGAATTAGAGCCACCAGA
B9,flatv-h18-b159,ACCACCACTAGCGGGGTTTTGCTCGTAGCATT
C9,flatv-h20-b159,CCACAGACTTGCAGGGAGTTAAAGAAACGAAA
D9,flatv-h22-b159,GAGGCAAAAAAGCTGCTCATTCAGATGCGATT
E9,flatv-h24-b159,TTAAGAACGAATCGTCATAAATATTCAGAAAACGAGAATG
A10,flatv-h18-b175,CAGAGCCGCCAGCAAAATCACCAGCCTTTTTAAGAAAAGT
B10,flatv-h20-b175,CAACGCCTAGTACCAGGCGGATAAACCACCCT
C10,flatv-h22-b175,CCAACCTAGCCGCTTTTGCGGGATACAAACTA
D10,flatv-h24-b175,ATTACCTTTGAATAAGGCTTGCCCCGAAGGCA
E10,flatv-h16-b199,CTATCTTACCGAAGCTAGCACCATTACCATGCCACCCT
F10,flatv-h18-b191,CAGAGCCGTGCCGTCGAGAGGGAGTTTCGT
G10,flatv-h20-b191,CACCAGTCGTCACCCTCAGCAGTACGTAAT
H10,flatv-h22-b191,GCCACTATGACGAGAAACACCAAATTTCAACTTTAAT
A11,flatv-h18-b207,TCAGAACCTAGCAAGGCCGGAAACAATGAAATAGCAATAG
B11,flatv-h20-b207,TAACACTGTTGATATAAGTATAGCCGCCACCC
C11,flatv-h22-b207,GGGTAAAACGAAAGACAGCATCGGATGTACCG
D11,flatv-h16-b231,AATAAGAGCAAGAAACGTCACCAATGAAACCACCGCCTCC
E11,flatv-h18-b223,CTCAGAGCCCGGAATAGGTGTATCAGCCCAAT
F11,flatv-h20-b223,AGGAACCCAACGAGGGTAGCAACGTTCATGAGGAAGTTTC
G11,flatv-h18-b239,ACCGGAATCGATAGCAGCACCGTGAGTTAAGCCCAAT
H11,flatv-h20-b239,GATAGCAACCGTACTCAGGAGGGAGCCACC
A12,flatv-h16-b263,GATAACCCACAAGAATTAATCAGTAGCGACAGAAAATCAC
B12,flatv-h18-b255,CGGAACCATTTAGTACCGCCACCCAGAGCCACCACCCTCA
C12,flatv-h18-b271,TCATAATCAATCAAGTTTGCCTTTGCGCTAATATCAGAGA
D12,flatv-h16-b287,TAATTGAAGCGTCAGACTGTAGCCTTATTAGCGTTTG





edge staples

arm 1

```
plate flat-v-edges-bridges
A1,flatv-h1-b8-II,CAAATATTTAAATTGGGAAGATTGTATAAG
B1,flatv-h3-b8-II,CATCGTAACCGTGCATGGTGTAGATGGGCG
C1,flatv-h5-b8-II,CAGGTCGACTCTAGAAAGCTTGCATGCCTG
D1,flatv-h7-b8-II,CAGGCGAAAATCCTGGCTGGTTTGCCCCAG
E1,flatv-h9-b8-II,CAGATTCACCAGTCAGATTATTTACATTGG
F1,flatv-h11-b8-II,CCCGAACGTTATTAAGTATTAAATCCTTTG
G1,flatv-h13-b8-II,CTGAGAAGAGTCAATGCTTAGATTAAGACG
H1,flatv-h15-b8-II,CCGTTTTTATTTTCATCGAGAACAAGCAAG
```

arm 2

```
A2,flatv-h17-b8-II,CAAAGACACCACGGAAACATATAAAAGAAACG
B2,flatv-h19-b8-II,CTTTTGATGATACAGGTAAGCGTCATACATGG
C2,flatv-h21-b8-II,CGAATAATAATTTTTTACAACTAAAGGAATTG
D2,flatv-h23-b8-II,CAGACGGTCAATCATATAGCCGGAACGAGGCG
E2,Flatv-h25-b8-II,CCAAAAGGAATTACGAATGCAGATACATAACG
F2,flatv-h27-b8-II,CAAACTCCAACAGGTCCGAACCAGACCGGAAG
G2,flatv-h29-b8-II,CATTAACATCCAATAACTACTAATAGTAGTAG
H2,flatv-h31-b8-II,CCGGAGACAGTCAAATAAAAGGGTGAGAAAGG
```

bridge staples

strands adjacent to bridges

```
plate flat-v-edges-bridges
A5,flatv-h0-b92,CGGTAAATAGGAACGCCATCACAGCTTTCATCAACATCCGGCACC
B5,flatv-h3-b112,AACCAGGCTGGCCTTCCTGTAGCAAAAT
C5,flatv-h5-b144,TGGGGTGCGTTGGGAAGGGCGATCTTCAG
D5,flatv-h7-b176,CGATGGCCTTGCGCTCACTGCCCGACTCA
E5,flatv-h9-b208,CAGAGGTGTTTGGGGTCGAGGTGACCCA
F5,flatv-h11-b240,TCAGGTTTGCCACGCTGAGAGCCACACCG
G5,flatv-h13-b272,AAAAGCCTTTTACATCGGGAGAAATACAG
H5,flatv-h15-b288,CGATTTTTTGTTTAACGTCAAAAACCAGTATAAAGCCAGTTAT
A6,flatv-h18-b300,AGCCCCGCGTTTTCATCGGCAAACACCCTGAACAAAGTCAGAGGG
B6,flatv-h20-b268,CCCTCTCAGAACCGCCACCCTCCATCTTT
C6,flatv-h22-b236,ACTTTGCTACAGAGGCTTTGTTTTCAGG
D6,flatv-h24-b204,GGTTTGAACGAGTAGTAAATTCATTAAAC
E6,flatv-h26-b172,ACAGTTCATTGAATCCCCCTCCATTGTGA
F6,flatv-h28-b140,CCAATTTTCATTCCATATAAACCATAAA
G6,flatv-h30-b108,TTTATCCCTGTAATACTTTTGTTTAGTTT
H6,flatv-h28-b95,TCGCAAACCAAAAACATTATGATTCAACGCAAGGATACTGAG
```

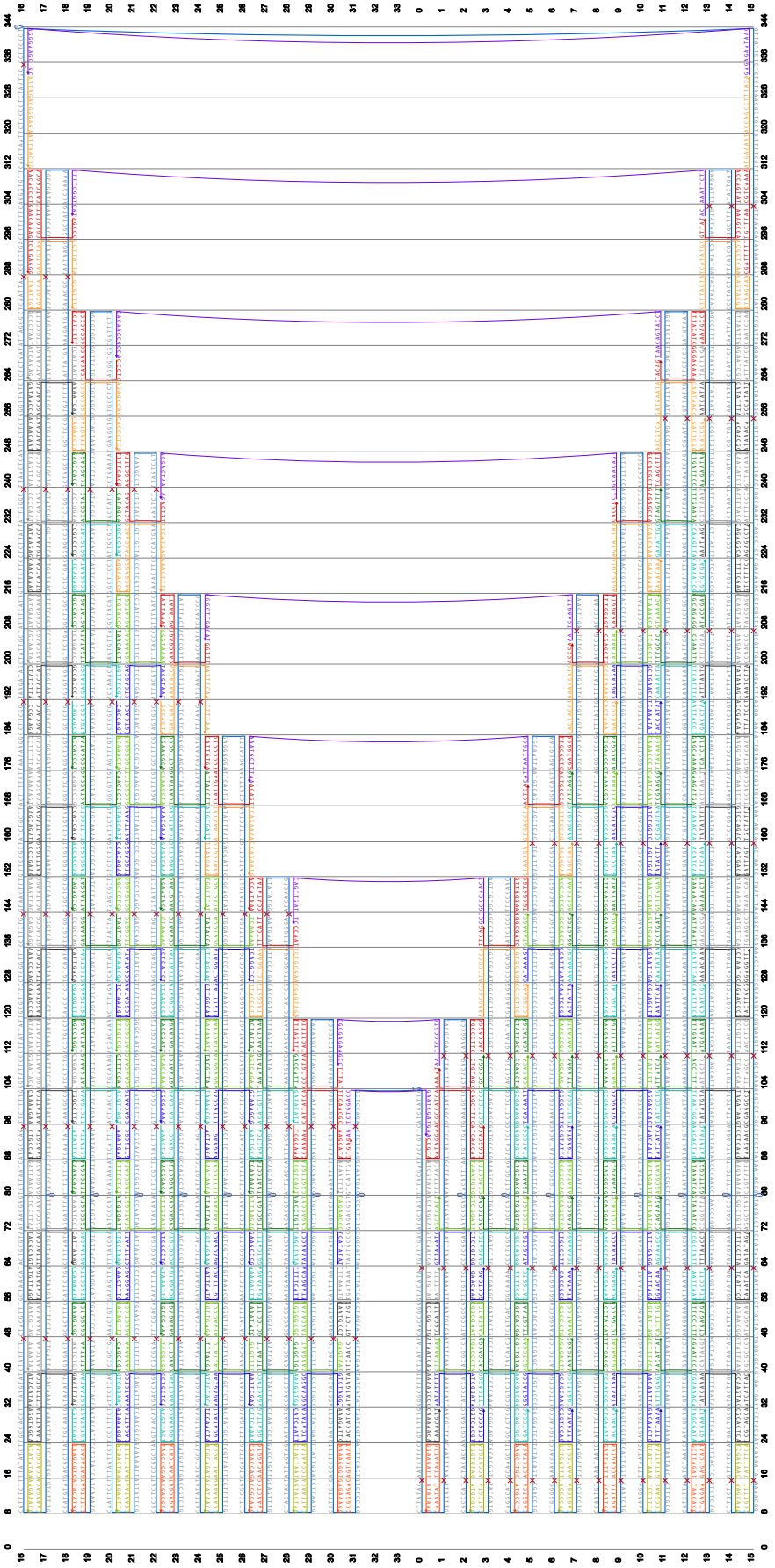
bridges

```
A7,flatv-h31-b93,AGTCTGGAGCGAATCGATGAA
B7,flatv-h1-b109,AATTCGCGTCCGGGAGAAGCC
C7,flatv-h3-b141,GCTGCGCAACTCAGTTGATTC
D7,flatv-h5-b173,CATTAATTGCGAAATGCTTTAA
E7,flatv-h7-b205,AATCAAGTTTGGGCTTGAGAT
F7,flatv-h9-b237,CCTGCAACAGTAGGACTAAAG
G7,flatv-h11-b269,TAACAGTACCTCAGAACCGCCA
H7,flatv-h13-b301,ACAAATTCTTTTTTCGGTCAT
A8,flatv-h15-b333,GAGAGAATAACAGGGAAGCGC
```

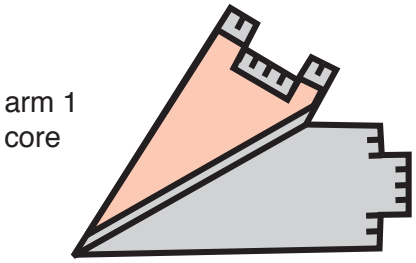
strands at corner

```
B8,flatv-h15-b312,ATGAAAATAGCAGCCTTTACA
C8,flatv-h16-b332,ATTAGACGGGAGAATTAACTG
```

Sequence diagram for 60° corner origami with straight edges



S5.9. 60° corner origami with edge shapes



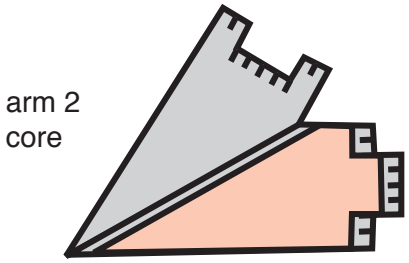
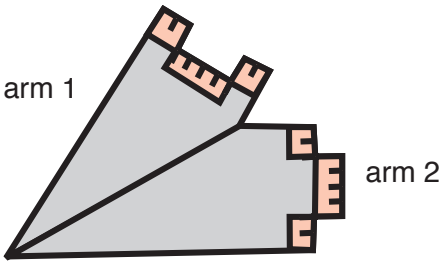
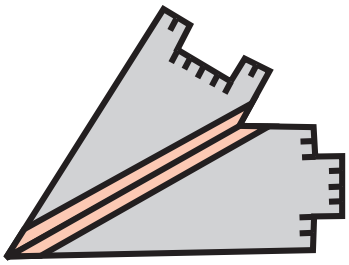


plate shape-U-top-core (arm 1)
A1, **shapeU-**h1-b64,CGAACTGAGTGAATTACCTTATGCAGGACGTTGGGAAGAA
B1, **shapeU-**h13-b64,TGCCAGCTCTTGCCTGAGTAGAAGCCAGAACAATATTACC
C1, **shapeU-**h1-b80,GAAAGAGGTGTCGAAATCCGCGACCTGCTCCATGTTACTTAAGGGAAC
D1, **shapeU-**h15-b72,GCCAGCCATTGCAACAATTAGTAATAACATCAGCATTAAT
A2, **shapeU-**h1-b96,AACGGTGTTTGAGATGGTTTAATAACGAAC
B2, **shapeU-**h3-b96,TAACGGAATCATAAATATTCATTAAACGAG
C2, **shapeU-**h5-b96,AATGACCACATTCCATATAACAGGTTTGAC
D2, **shapeU-**h7-b96,CATTAGATTGTAGGTAAAGATTCCACCATC
E2, **shapeU-**h9-b96,AATATGATAACCCGTCGGATTCTGGATAGG
F2, **shapeU-**h11-b96,TCACGTTGTAAAGCCTGGGGTGCCAACGCG
G2, **shapeU-**h13-b96,CGGGGAGAAGCAATACTTCTTTGGGAAAAACGCTCATG
A3, **shapeU-**h1-b112,GGCGCATAACGGAGATTTGTATCATCGCCTGATAAATTGACAGATG
B3, **shapeU-**h3-b112,TTACAGGTCGTAACAAAATTGGGCACAGACCA
C3, **shapeU-**h5-b112,AAATCAGGAAAACCAACGGAATCGCAACATTA
D3, **shapeU-**h7-b112,CAAATGGTGGATGGCTGGAAGTTTTAAATCAA
E3, **shapeU-**h9-b112,GTTCTAGCTTTAAATGGAGTAATGACATTTCG
F3, **shapeU-**h11-b112,GGGCGCATATCAACATCGAGTAACATTCAACC
G3, **shapeU-**h13-b112,GCGTATTGCAATTCCAATAAAGTGGTAGAT
H3, **shapeU-**h15-b104,GAAATACCATTTTGACCTAGGGCGACCGTTGTGGCGGTTT
A4, **shapeU-**h3-b128,TCATCAGTTTACCAGACGACGATATCTTTACC
B4, **shapeU-**h5-b128,CTGACTATAAGAGGTCATTTTTGCCAATAACC
C4, **shapeU-**h7-b128,TGTTTAGCTAGAACCCTCATATATTGATAAAT
D4, **shapeU-**h9-b128,TAATGCCGCCTGTAGCCAGCTTTCCGTAACCG
E4, **shapeU-**h11-b128,TGCATCTGAATTGTTATCCGCTCAGGCGCCAG
F4, **shapeU-**h13-b128,GGTGGTTTGCGAAAGGAGCGGGCGGCTCAATCGTCTGAAA
A5, **shapeU-**h5-b144,AGAAGCAATATCATAACCCTCGTTGAGATT
B5, **shapeU-**h7-b144,TCATTTGGATTGCTCCTTTTGATTATAGTC
C5, **shapeU-**h9-b144,TAGCTATTAAGGATAAAAATTTTATATTT
D5, **shapeU-**h11-b144,GAGGGGACTTCGCGTCTGGCCTTGAGAGGG
E5, **shapeU-**h13-b144,CACCAGTGCTGTTTCCTGTGTGACCAGTTT
F5, **shapeU-**h15-b136,TGGATTATTTACATTAAGGAAGGGAAGAAATTCTTTT
A6, **shapeU-**h3-b152,CCACATTCAACTAATGATAGTAAGAGCAACACAGCGGATT
B6, **shapeU-**h5-b160,GCATCAAATTAGAGAGTACCTTTAGGCGCGAG
C6, **shapeU-**h7-b160,CTGAAAAGGCCTTTATTTCAACGCTTTGAGAG
D6, **shapeU-**h9-b160,ATCTACAAACGCCATCAAAAATAAGACGACAG
E6, **shapeU-**h11-b160,TATCGGCCGTAATCATGGTCATAGAGACGGGC
F6, **shapeU-**h13-b160,AACAGCTGGGCGAACGTGGCGAGAGGCAGATTCACCAGTC
A7, **shapeU-**h7-b176,AATTCTACCTCCAACAGGTCAGGAAAGATTAA
B7, **shapeU-**h9-b176,AGGTCATTTACTTTTGCGGGAGAAGTGGCATC
C7, **shapeU-**h11-b176,ATCGCACTTTTTTTAACCAATAGGAAGGCTATC
D7, **shapeU-**h13-b176,TCACCGCCTACCGAGCTCGAATTCTCAGGAAG
E7, **shapeU-**h15-b168,ACACGACCAGTAATAACTTGACGGGGAAAGCCATTGCCCT
A8, **shapeU-**h5-b184,CCGAAAGACTTCAAACAGACCGGAAGCAAATAATAGT
B8, **shapeU-**h7-b192,AGTAGCATATTATGACCCTGTAAGCCTGAG
C8, **shapeU-**h9-b192,AGTCTGGAGTTAAATCAGCTCATCCAGCCA
D8, **shapeU-**h11-b192,GCTTTCCGTAGAGGATCCCCGGGTGGCCCT
E8, **shapeU-**h13-b192,GAGAGAGTGCCCCCGATTTAGAGAAGGGACATTCTGGC
A9, **shapeU-**h9-b208,GAGAATCGCGGTTGTACCAAAAACTAACATCC
B9, **shapeU-**h11-b208,TCTGGTGCTTCGCATTAAATTTTTGCAAACAA
C9, **shapeU-**h13-b208,GCGGTCCAGCCTGCAGGTCGACTCGCACCGCT
D9, **shapeU-**h15-b200,CAACAGAGATAGAACCCGGAACCCTAAAGGGATGCAGCAA
E9, **shapeU-**h7-b216,ATACAGGCAAGGCAAAGAGCATAAAGCTAAATATGAACGG
F9, **shapeU-**h9-b224,TAATCGTATAATATTTTGTTAAAACGGAAACC
G9, **shapeU-**h11-b224,AGGCAAAGAGTGCCAAGCTTGCATCGCTGGTT
H9, **shapeU-**h13-b224,TGCCCCAGCCGTAAAGCACTAAATCTTCTGACCTGAAAGC
A10, **shapeU-**h11-b240,CGCCATTCTTAAATTGTAAACGTAAACTAG
B10, **shapeU-**h13-b240,AAATCCTGTGTAAAACGACGGCCCGCCATT
C10, **shapeU-**h15-b232,GTAAGAATACGTGGCTTTGGGGTCGAGGTGCAGGCGA
D10, **shapeU-**h9-b248,TCATATGTACCCCGGTTTGTATAAGCAAATATAGGCTGCG
E10, **shapeU-**h11-b256,CAACTGTTTTCCCAGTCACGACGTTTTGATGG
F10, **shapeU-**h13-b256,TGGTTCCGCACCCAAATCAAGTTTACAGACAATATTTTTG
G10, **shapeU-**h13-b272,AAAATCCCGGGTAACGCCAGGGTTGGGAAGGG
H10, **shapeU-**h15-b264,AATGGCTATTAGTCTTCCACTACGTGAACCATAAATCGGC
A11, **shapeU-**h11-b280,GCGGGCCTCTTCGCTCAAGGCGATTAAGTTTTATAAA
B11, **shapeU-**h13-b288,TCAAAAGATATCAGGGCGATGGCTAATGCGCGAACTGA
C11, **shapeU-**h15-b296,TAGCCCTAAAACATCGAGGGCGAAAAACCGTCATAGCCCG
D11, **shapeU-**h13-b312,TTGAGTGTTGTTCCAGTGGACTCCAACGTCAACCATTAAA

plate **shapeU-**bottom-core (arm 2)
A1, **shapeU-**h20-b39,CGAACCTCCCGACTTGTTGCTATTTTGCACTACATAAA
B1, **shapeU-**h22-b31,GGTGGCATAAGTTTATTTTGTCCCACCAGA
C1, **shapeU-**h24-b31,ACCACCAAGGCAGGTCAGACGAAGGAGGTT
D1, **shapeU-**h24-b47,CAGAGCCGACAATCAATAGAAAATAAACGTAG
E1, **shapeU-**h26-b47,CCGTACTCTTGGCCTTGATATTCAACCACCCT
A2, **shapeU-**h16-b71,ATTTTGCGGAACAAAGTTCCTGATTATCAGATCTGATGCA
B2, **shapeU-**h18-b63,AATCCAATAATATATTTTAGTTAAATCCGGTA
C2, **shapeU-**h20-b63,TTCTAAGACCTGAATCTTACCAACTACGCAGT
D2, **shapeU-**h22-b63,ATGTTAGCTCATATGGTTTACCAGCGCCACCC
E2, **shapeU-**h24-b63,TCAGAGCCCAAACAAATAAATCCTGTATAGCC
F2, **shapeU-**h26-b63,CGGAATAGGATAGCAAGCCCAATATTTTTTCA
G2, **shapeU-**h28-b63,CGTTGAAATAATTGTATCGGTTTACATCGGAA
H2, **shapeU-**h30-b63,CGAGGGTACTTTTTCATGAGGAAGTTTCCATTAAACGGGAGCAGCGA
A3, **shapeU-**h18-b79,GTAAATGGATGGCAATTCATCATTGAGTAACATTATC
B3, **shapeU-**h20-b79,AAGGCTTTTTCATCTTCTGACCAACTATAT
C3, **shapeU-**h22-b79,TCCTTATGCTAACGAGCGTCTTAGATATAG
D3, **shapeU-**h24-b79,TCAGAACCGCCAAAGACAAAAGATTAAGAC
E3, **shapeU-**h26-b79,GATATAACATTAAAGCCAGAATCGCCACCC
F3, **shapeU-**h28-b79,ATAATAAGGAACCCATGTACCGAGAGGGTT
G3, **shapeU-**h30-b79,AAGACAGTCAGCTTGCTTTCGAAATTGCGA
A4, **shapeU-**h16-b103,TATTAATTTTAAAAGTATATAATCCTGATTGTTTAGGTTG
B4, **shapeU-**h18-b95,GGTTATATTAAATTTAATGGTTTGCAATAGCA
C4, **shapeU-**h20-b95,AGCAAATCTCCAGAGCCTAATTTGCAAAAGAA
D4, **shapeU-**h22-b95,CTGGCATGGGCGACATTCAACCGACCGCCTCC
E4, **shapeU-**h24-b95,CTCAGAGCGGAAAGCGCAGTCTCTCGGATAAG
F4, **shapeU-**h26-b95,TGCCGTCGTAACACTGAGTTTCGTAAGGAACA
G4, **shapeU-**h28-b95,ACTAAAGGGGTGAATTTCTTAAACCGGGATCG
H4, **shapeU-**h30-b95,TCACCCTCTAAAATACGTAATGCCACTACGAACTCATCTTTTAAAGGC
A5, **shapeU-**h18-b111,CCTCCGGCTTGGATTATAGAACCTTCCTTTGCCCGAACGT
B5, **shapeU-**h20-b111,ACCGCGCCAAATACCGTATGCGTTCTTTTTAA
C5, **shapeU-**h22-b111,GGAATACCCCAGTTACTAAACAGCGAATCATT
D5, **shapeU-**h24-b111,CACCGGAATTGAGGGAATTCATTAATAATAAC
E5, **shapeU-**h26-b111,GTACCAGGGAATTTACTTCCAGTAAGAGCCAC
F5, **shapeU-**h28-b111,AGAATAGACACCAGTAACTACAACTTTGCTCA
G5, **shapeU-**h30-b111,CGCTTTTGAGCTTGATTGCGCCGAGCGGAGTG
A6, **shapeU-**h16-b135,ACAACTCGTATTAAAACCATATCAAAATTAGGTCTGAG
B6, **shapeU-**h18-b127,AGACTACATACAAATTCTTACCTTATTTTC
C6, **shapeU-**h20-b127,ATCGTAGCATATTATTTATCCCAACCGAGG
D6, **shapeU-**h22-b127,AAACGCAAAGGTGAATTATCACAAAATCAC
E6, **shapeU-**h24-b127,CGGAACCAGCGTCATACATGGCTAGGATTA
F6, **shapeU-**h26-b127,GCGGGGTGCCTGTAGCATTCCACTTTCAAC
A7, **shapeU-**h18-b143,AAATCATATTTGCACGTAAAACAGCTTTACAAACAATTCG
B7, **shapeU-**h20-b143,AGCCGTTTAGTATAAAGCCAACGCATTTATCA
C7, **shapeU-**h22-b143,CAGAAGGAAATCCAAATAAGAAACAACAAGCA
D7, **shapeU-**h24-b143,TCATAATCCGTCACCGACTTGAGCAAAGTTAC
E7, **shapeU-**h26-b143,AGAAGGATTTTTGATGATACAGGACCATCTTT
F7, **shapeU-**h28-b143,CTAAACAACAGACAGCCCTCATAGTCCTCAAG
A8, **shapeU-**h16-b167,ATTTAGAAGTATTAGAAAATAAAGAAATTGCGGAAGAGTC
B8, **shapeU-**h18-b159,AATAGTGATCAACAGTAGGGCTTATACCGCAC
C8, **shapeU-**h20-b159,TCATCGAGGATTTTTTGTTTAACGAAGCAGAT
D8, **shapeU-**h22-b159,AGCCGAACCATTTGGGAATTAGAGCCCTTATT
E8, **shapeU-**h24-b159,AGCGTTTGGTGTACTGGTAATAAGATTAAGAG
F8, **shapeU-**h26-b159,GCTGAGACTTAGCGTAACGATCTAATGAATTTTCTGTATG
A9, **shapeU-**h18-b175,ACGCTGATAGATTTTCAGGTTTATAATACATTTGAGG
B9, **shapeU-**h20-b175,AACCAAGATTGAGAATCGCCATAGATTAAG
C9, **shapeU-**h22-b175,GAAAAGTTCAAAAATGAAAATAGGGTATTA
D9, **shapeU-**h24-b175,CATAGCCCCAGCAAAATCACCACTTTTTAA
E9, **shapeU-**h26-b175,TGAAAGTTTTTAACGGGGTCAGTTTTCGGT
G9, **shapeU-**h16-b295,CTCAATCAATATCTGGCGCAGAGGCGAATTATTAATGGAA
H9, **shapeU-**h18-b287,ACAGTACACAGACGACGACAATAATTTATCAACAATAGAT
A10, **shapeU-**h16-b199,TTAGAGCCGTCAATAGAACGTCAGATGAATATAACATAGC
B10, **shapeU-**h18-b191,GATAGCTTATTTAACAACGCCAACTTATCATT
C10, **shapeU-**h20-b191,CCAAGAACGCAGCCTTTACAGAGACTATCTTA
D10, **shapeU-**h22-b191,CCGAAGCCGTAGCACCATTACCATCGCGTTTT
E10, **shapeU-**h24-b191,CATCGGCATGCCTTGAGTAACAGTTTCGGAACCTATTATT
G10, **shapeU-**h18-b303,ACCTTTTTTCATTTCAATTACCTGCCTCAAATATCAAACC
H10, **shapeU-**h16-b319,TGCTGAAAGCAAAAGAAGATGACATTTAACAATTTCA
A11, **shapeU-**h18-b207,TCCTTGAAACAGTAACAGTACCTTACTAACAACTAATAGA
B11, **shapeU-**h20-b207,GTCTTTCCATGTAATTTAGGCAGACCTTAGAA
C11, **shapeU-**h22-b207,AGCAATAGGAATAACATAAAAACAAATCGGCT
D11, **shapeU-**h24-b207,GACTGTAGTAGCAAGGCCGGAAACAATGAAAT
E11, **shapeU-**h16-b231,AATATCTTTAGGAGCTTACATCGGGAGAAATATTAATT
F11, **shapeU-**h18-b223,AATTTTCGGCATTTTCGAGCCATAGAAACC
G11, **shapeU-**h20-b223,AATCAATGGGAAGCGCATTAGAATAAGAGC
H11, **shapeU-**h22-b223,AAGAAACGTCACCAATGAAACCAATCAAGTTTGCCTT
A12, **shapeU-**h18-b239,ATCGTCGCCAATAACGGATTCGCCGAGGAAGGTTATCTAA
B12, **shapeU-**h20-b239,CGAGCATGGTAATAAGAGAATATATTCTGTAA
C12, **shapeU-**h22-b239,GCCCAATACGGGAGAATTAACTGACTAATTTA
D12, **shapeU-**h16-b263,ACAGTTGAAAGGAATTTGATTGCTTTGAATACGAGTGAAT
E12, **shapeU-**h18-b255,AACCTTGCAAGTACCGACAAAAGGAAAATAAT
F12, **shapeU-**h20-b255,ATCCCATCACACCCTGAACAAAGTGATAACCCACAAGAAT
G12, **shapeU-**h18-b271,TATATGTCAAGTTACAAAATCGTCAGTTGGCAAATCA
H12, **shapeU-**h20-b271,AACAAGATAAAGTAATTCTGTCTAAATCAA





edge staples

arm 1

```
plate shape-U-edges
A1,shapeU-h1-b40-II,CAGACGGTCAATCATAGCCGGAACGAGGCG
B1,shapeU-h3-b40-II,CTCATTATACCAGTCGATTTTAAGAACTGG
C1,shapeU-h3-b72-II,AAATCTACGTTAATAATTCAACTTTAATCATTCCAACTTT
D1,shapeU-h5-b72-II,CTTTAAACAGTTCAGAGAATCCCCCTCAAATG
E1,shapeU-h7-b72-II,CGAACGAGTAGATTTATTGATTCCCAATTCTG
F1,shapeU-h9-b72-II,CCGGAGACAGTCAAATAAAAGGGTGAGAAAGG
G1,shapeU-h11-b72-II,CGGATTGACCGTAATGCCGTGGGAACAAACGG
H1,shapeU-h13-b80-II,GAATCGGCCTAATGAGTGAGCTAACTCACATTAATTGCGTACCTGTCG
A2,shapeU-h13-b40-II,CTTTCCAGTCGGGAATGCGCTCACTGCCCG
B2,shapeU-h15-b40-II,CCTTGCTGGTAATATAACTCAAACTATCGG
```

arm 2

```
A3,shapeU-h17-b40-II,CGGAATTATCATCATAAAACCACCAGAAGGAG
B3,shapeU-h19-b40-II,CGAGAAAACTTTTTCACGCAAGACAAAGAACG
C3,shapeU-h22-b47-II,AAAATACACCAGCTACAATTTTATACGCGAGGCGTTTTAG
D3,shapeU-h21-b8-II,CCTTAAATCAAGATTAGCGGGAGGTTTTGAAG
E3,shapeU-h23-b8-II,CAAAGACACCACGGAAACATATAAAAGAAACG
F3,shapeU-h25-b8-II,CATTGACAGGAGGTTGCCAGAGCCGCCGCCAG
G3,shapeU-h27-b8-II,CCACCCTCAGAACCGCGCCACCCTCAGAACCG
H3,shapeU-h26-b31-II,TAGTACCCACCCTCAGAGCCACCACCCTCATTTTCAGGGTGTATCA
A4,shapeU-h29-b40-II,CTCCAAAAGGAGCCTTATCTCCAAAAAAAAGG
B4,shapeU-h31-b40-II,CTTTGAGGACTAAAGAGCAACGGCTACAGAGG
```

bridge staples

strands adjacent to bridges

```
plate shape-U-bridges
A1,shapeU-h0-b124,GTACAGGCTGGCTGACCTTCATCATTACCCAAATCAAAGAAAGAT
B1,shapeU-h3-b144,TAGGAATAACAAGAACCGGATATTCAAG
C1,shapeU-h5-b176,GAGGAAGCAAAGGAATTACGAGGCCAGAT
D1,shapeU-h7-b208,AATAAATCGAGCTTCAAAGCGAACTATCG
E1,shapeU-h9-b240,CATGTCAAAGCAATAAAGCCTCAGAATT
F1,shapeU-h11-b272,CGATCGGTCCCAAAAACAGGAAGATGATA
G1,shapeU-h13-b304,AGATAGGGAAAGGGGGATGTGCTGATTAC
H1,shapeU-h15-b320,AATACCGAACGAACCACCAGCAGCACTATTAAAGAACGTTTGG
A2,shapeU-h18-b332,AATTATGAAACAAACATCAAGAATGAAAAATCTAAAGCATCACCT
B2,shapeU-h20-b300,GCCTGACAACATGTTCAGCTATTTGAATT
C2,shapeU-h22-b268,AGAGACAGAGGGTAATTGAGAAGTCCTG
D2,shapeU-h24-b236,GACAGATCGATAGCAGCACCGTGAGTTAA
E2,shapeU-h26-b204,CCTATGCCCGTATAAACAGTTTAGCGTCA
F2,shapeU-h28-b172,AGTAAAAGTTTTGTCGTCTTCTGAAACA
G2,shapeU-h30-b140,TCGCTTCGCCCACGCATAACCGGATTTTG
H2,shapeU-h28-b127,AGTTTCACAATGACAACAACCAGAGGCTTGCAGGGAGTGACC
```

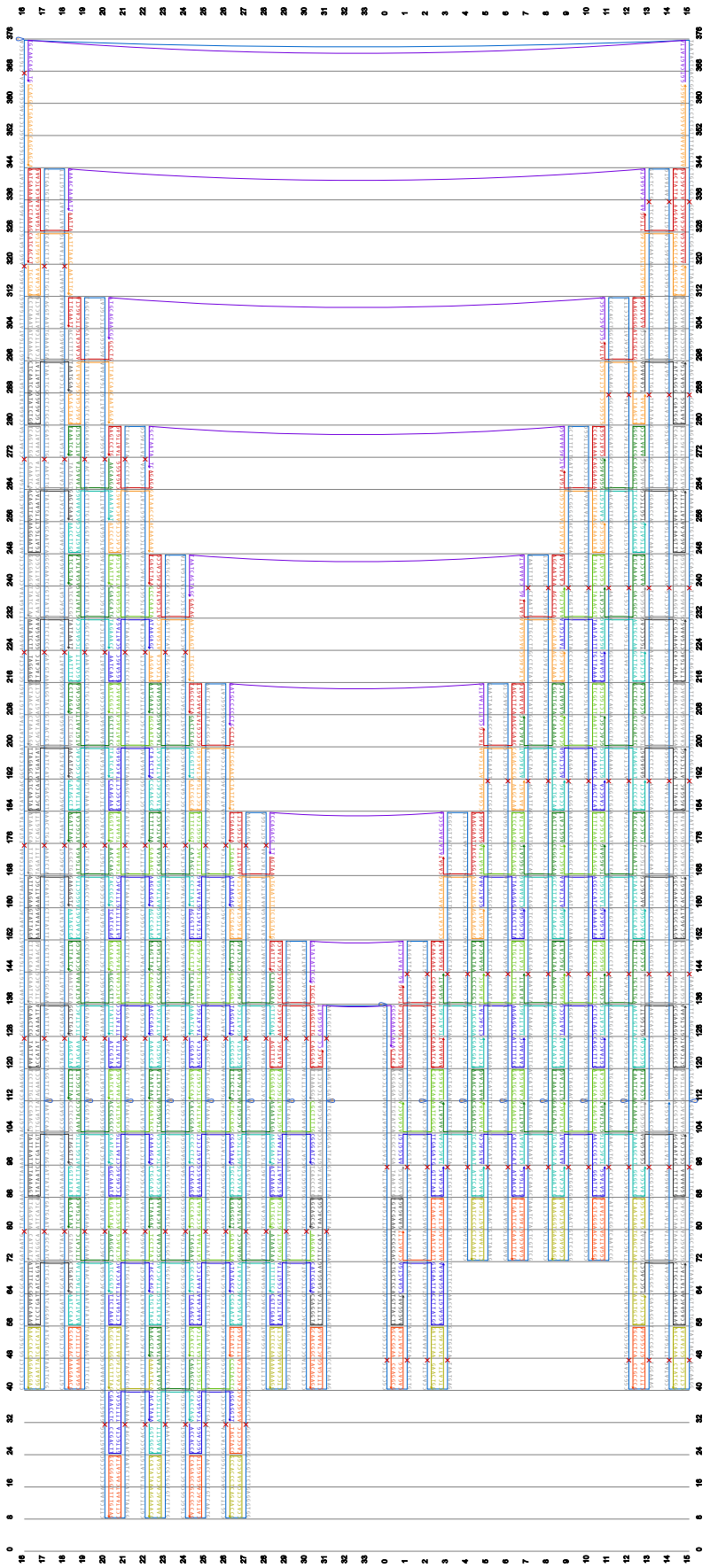
bridges

```
A3,shapeU-h31-b125,CCCAGCGATTCGCGAAACAAA
B3,shapeU-h1-b141,AGTAATCTTGGATATATTCGG
C3,shapeU-h3-b173,ACATAACGCCATCCAGACGTT
D3,shapeU-h5-b205,CGTTTTAATTCAATGCCCCCTG
E3,shapeU-h7-b237,AGCAAAATTATAATCAGTAGC
F3,shapeU-h9-b269,ATCAGAAAAGCCGCTAATATC
G3,shapeU-h11-b301,GCCAGCTGGCGATGCAGAACGC
H3,shapeU-h13-b333,AACAAGAGTCAAAACAAAATT
A4,shapeU-h15-b365,GGTCAGTATTATGCAACAGTG
```

strands at corner

```
B4,shapeU-h15-b344,AAGATAAAACAGAGGTGAGGC
C4,shapeU-h16-b364,CCACGCTGAGAGCCAGCAGCA
```

Sequence diagram for 60° corner origami with edge shapes



S5.10. 60° corner origami with edge shapes and reversed stacking polarity

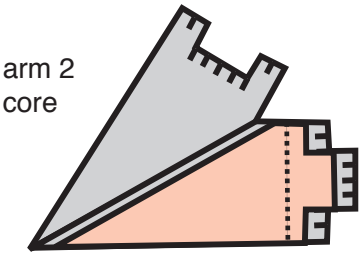


plate shape-flip-bottom-core (arm2)
A1,shflip-h22-b47,ATTTAGGATAAAGTACCGACAAAGAAAAAT
B1,shflip-h24-b47,TACCGACACACCGGAATCATAAAACATGTA
C1,shflip-h26-b55,TGAAAACATAGCGATGTGAATTTATCAAAAGTTTGAAA
D1,shflip-h20-b63,CCTGAACAAAGGTAAAGTAATTCTCATATTTA
E1,shflip-h22-b63,ACAACGCCTTACTAGAAAAAGCCTTGACCTAA
A2,shflip-h16-b79,GAAGCCTTTACAATTTTATCCTGAATCTTAC
B2,shflip-h18-b79,ATCATTACGGCTTATCAGAACGCGGAGGTTTT
C2,shflip-h20-b79,TCAACAATGAACGGGTCGCACTCATCGTAGGA
D2,shflip-h22-b79,AGAATCGCGTCCAGACACGACAATCCTGTTTA
E2,shflip-h24-b79,TCATCTTCGTTTAGTAAATTCTTACTTAATTG
F2,shflip-h26-b79,TAATTAATCTACCTTTAGACAAAGAGTTAATT
G2,shflip-h28-b79,AATTATTCAACAAACATCAATATAGTCGCTAT
H2,shflip-h30-b87,TATTTGCACGTAAAACTAACGTCATCGCCTGACAGAGGCG
A3,shflip-h33-b96,CAACGCTAACGAGCGTCTTTCCAGAGCCTAATTTG
B3,shflip-h16-b95,CTTGCGGAGGCGTTTTAGCGAACACCTTGC
C3,shflip-h18-b95,ATTTTCATCGAGAACAAGCAAGAATCAACA
D3,shflip-h20-b95,GAACGCGAAACAACATGTTCAGTAACAACT
E3,shflip-h22-b95,AGTAGGGCCAGTATAAAGCCAATATTAGAC
F3,shflip-h24-b95,ATATTTTAACGCGAGAAAACTTTATTAATT
G3,shflip-h26-b95,GTAAATCTGTGAGTGAATAACCCAGAAGGA
H3,shflip-h28-b95,AATCGCGTTGCTTTGAATACCACAATATAA
A4,shflip-h17-b112,TGAACCTCGAAAAATCTAAAGCATCCTCCCGA
B4,shflip-h19-b112,GTTGAAAGATCTGGTCAGTTGGCACCGTTTTT
C4,shflip-h21-b112,AATAGATTATATCTTTAGGAGCACCTAATGCA
D4,shflip-h23-b112,TTTACAAATTTGAGGATTTAGAAGCGCTCAAC
E4,shflip-h25-b112,TTAAAAGTTCCTTTGCCCGAACGTTTTCAAAT
F4,shflip-h27-b112,GCGGAATTGGAACAAAGAAACCACTTGCTTCT
G4,shflip-h29-b112,TCCTGATTGATGATGGCAATTCATAGTTACAA
H4,shflip-h30-b119,ATGGAAGGGTTAGAACCTACCATATCAAAAT
A5,shflip-h16-b151,AACAGTGCCACGCTGACAGCAGCAAAT
B5,shflip-h17-b120,AAATATCAAACAAAATAAACAGCCATA
C5,shflip-h18-b151,TTTATTTTGTCACAATCCTCAATCAAT
D5,shflip-h19-b120,GAATTGAGGAACAATAGAAAATTCATA
E5,shflip-h20-b151,GAACCGCCACCCTCAGGGTTATCTAAA
F5,shflip-h21-b120,AGAGCCGTCAAAGCCACCACCCTCAGA
G5,shflip-h22-b151,AGAAGGATTAGGATTATAGATAATACA
H5,shflip-h23-b120,CAATTCGACAAGCGGGGTTTTGCTCAG
A6,shflip-h24-b151,GTAACGATCTAAAGTTCTCGTATTAAA
B6,shflip-h25-b120,TTGAGTAACATTTGTCGTCTTTCCAGA
C6,shflip-h26-b151,TTTCTTAAACAGCTTGTATCATTTTGC
D6,shflip-h27-b120,ATCATCATATTATACCGATAGTTGCGC
E6,shflip-h28-b151,GGTAGCAACGGCTACACCTGATTATCA
F6,shflip-h29-b120,GTTTGGATTATGAGGCTTTGAGGACTA
G6,shflip-h31-b136,AGAGGCAAAAGAATACACCAACCTAAAACGAAACTTCTGAATA
A7,shflip-h16-b167,ATTAACACCGCCTGCTTATTTATCCCAATCACACCACG
B7,shflip-h18-b159,GAATAAGTGGTTTACCAGCGCCGAGCCGCC
C7,shflip-h20-b159,ACCCTCAGCCGCCACCAGAACCCTGAGACT
D7,shflip-h22-b159,CCTCAAGTACCAGGCGGATAAGGCCCTCAT
E7,shflip-h24-b159,AGTTAGCCGTTAGTAATGAATTGCTTTCG
F7,shflip-h26-b159,AGGTGAACGACAATGACAACAACAGCATCG
G7,shflip-h18-b175,ACGCAAAGCAAATAAGAAACGATTAGGTGAGGCGGTCAGT
H7,shflip-h20-b175,CTCCCTCAAAAGACAAAAGGGCGATAAAAGAA
A8,shflip-h22-b175,TTAAGAGGACCACCAGAGCCGCCGGGAACCGC
B8,shflip-h24-b175,CACAGACATGCCGTCGAGAGGGTTTGAAAGTA
C8,shflip-h26-b175,TATCAGCTTTTCTGTATGGGATTTTAGCATTC
D8,shflip-h28-b175,GCGAAAGACCATCGCCCACGCATATATCGGTT
E8,shflip-h16-b199,GCAGAAGATAAAACAGTTTTGTTTAACGTCAATAAAGGTG
F8,shflip-h18-b191,GCAACATACATTCAACCGATTGAGAACCAGAG
G8,shflip-h20-b191,CCACCACCCCAGCATTGACAGGAGCTATTATT
H8,shflip-h22-b191,CTGAAACAGATATAAGTATAGCCCCAAACTAC
A9,shflip-h24-b191,AACGCCTGTGCTAAACAACTTTCAAAGGAGCC
B9,shflip-h26-b191,TTTAATTGACCGATATATTCGGTCTTTGCGGGATCGTCAC
C9,shflip-h18-b207,ACATACAAAATGAAAATAGCAGCCGAACGAACCACCA
D9,shflip-h20-b207,TCACCGGGGAGGGAAGGTAAATTAGAAAAT
E9,shflip-h22-b207,TCGGAACGTTGAGGCAGGTCAGAATCAAAA
F9,shflip-h24-b207,ACCAGTAGGAATAGGTGTATCATGCCTATT
G9,shflip-h26-b207,GCTCCAAACAGTTTCAGCGGAGGTTTCGTC
H9,shflip-h16-b231,ATCGCCATTAAAAATACCTTTACAGAGAGAATAGTATGTT
A10,shflip-h18-b223,AGCAAACGATTGACGGAAATTATTTTTGCCAT
B10,shflip-h20-b223,CTTTTCATACGATTGGCCTTGATAAACAGTTA
C10,shflip-h22-b223,ATGCCCCCCCGTACTCAGGAGGTTTGTACCGT
D10,shflip-h24-b223,AACACTGATGAGAATAGAAAGGAAACGTTGAAAATCTCCA
E10,shflip-h18-b239,TATTACGCAACATAAAAACAGGGACTGATAGCCCTAAAAC
F10,shflip-h20-b239,TATTAGCGCATTAAAGGTGAATTAAGACTCCT
G10,shflip-h22-b239,CCCGTATATTCACAAACAAATAAAAGCCCCCT
H10,shflip-h24-b239,GGAACCCATAGTACCGCCACCCTCTAACAGTG
A11,shflip-h16-b263,TCTTTAATGCGCGAAAGCGCATTAGACGGGGAACTGGC
B11,shflip-h18-b255,ATGATTATCACCGTCACCGACTGGCATTTT
C11,shflip-h20-b255,CGGTCATTCCTCATTAAAGCCAGGTCAGTG
D11,shflip-h22-b255,CCTTGAGAGAACCGCCACCCTCTTTCAGGGATAGCAA
E11,shflip-h18-b271,ACCCAAAAAGAATTAACTGAACACTTTGAATGGCTATTAG
F11,shflip-h20-b271,TTTTCATCTGAGCCATTTGGGAATAACGGAAT
G11,shflip-h22-b271,TTTAACGGGAATGGAAAGCGCAGTGTAGCGCG
H11,shflip-h16-b295,TGGCACAGACAATATTCCTGAACAAAGTCAGAGAGGAAAC

plate shape-flip-bottom-core (arm2) continued....
A12,shflip-h18-b287,GCAATAATTAGAGCCAGCAAAATCCCTTTAGC
B12,shflip-h20-b287,GTCAGACTCTCTGAATTTACCGTTTACAGGAGTGTACTGG
C12,shflip-h18-b303,GGAAACCGGGTAATTGAGCGCTAAGCGTAAGAATACG
D12,shflip-h20-b303,AAGTTTGACCAGTAGCACCATTTACCAGAA
E12,shflip-h16-b327,AACCCTTCTGACCTGAAATATCAGAGAGATAAGATAGCCG
F12,shflip-h18-b319,AACAAAGTACCATTAGCAAGGCCGACCGTAATCAGTAGCG
G12,shflip-h18-b335,AGTAAGCACCCACAAGAATTGAGTTGGCCAACAGAGATAG
H12,shflip-h16-b351,GACATTCTAAGCCCAATAATAATTACCGAAGCCCTTT

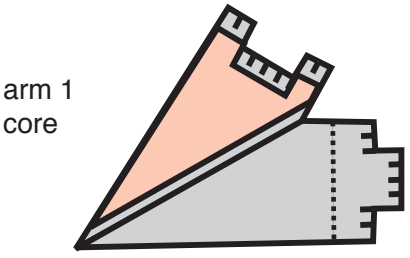
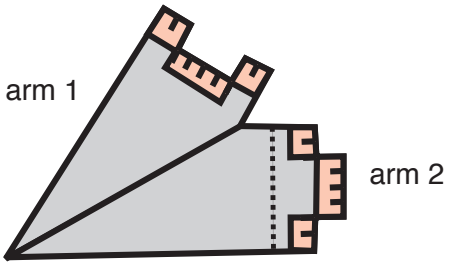
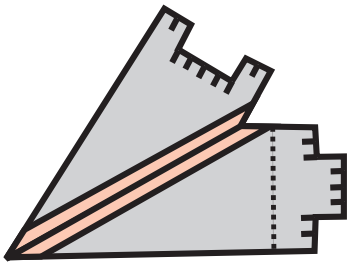


plate shape-flip-top-core (arm1)
A1,shflip-h1-b96,ATCATAAGCGAGAAACACCAGAACTTTCAACTTTAATCAT
B1,shflip-h13-b96,AGCTGTTTCGGAACCCTAAAGGGACGGCGAACGTGGCGAG
C1,shflip-h1-b112,ACTGACCAGCCTGATAAATTGTGTCGAAATCCGCGACCTGGACGGTCA
D1,shflip-h15-b104,AAAGGAAGGGAAGAAACCGTAAAGCACTAAATCCTGTGTG
A2,shflip-h1-b128,AAGAGGACCTGCTCATTCAGTGACGATTTT
B2,shflip-h3-b128,AAGAACTGGTTTACCAGACGACGAAGAAGT
C2,shflip-h5-b128,TTTGCCAGATTCGAGCTTCAAAGAGGATTA
D2,shflip-h7-b128,GAGAGTACAAATTAAGCAATAAAAAAAACA
E2,shflip-h9-b128,TTATGACCATTGTAAACGTTAATAATCAGC
F2,shflip-h11-b128,TCATTTTTCGACGGCCAGTGCCAATCCGCT
G2,shflip-h13-b128,CACAATTCTTTGGGGTCGAGGTGGCGAAAGGAGCGGGC
A3,shflip-h1-b144,GGTGTACAGCGCGAAAGTACAACGGAGATTTGTATCATCACTTTGA
B3,shflip-h3-b144,TACCAGTCTTACCCAATAACAAAGAGATGAAC
C3,shflip-h5-b144,ATAGTAAAAAAGGAATTAACCCTCGCTCATTA
D3,shflip-h7-b144,GCTCCTTTTCAAATATGCGTTTTAAGGGGGTA
E3,shflip-h9-b144,CTTTTGCGTAGTAGTAAATTAGCACTTTAATT
F3,shflip-h11-b144,AGGAACGCCAAAAACAATATTTAACTGTAATA
G3,shflip-h13-b144,ATACGAGCAGGGTTTTTTTGTAAAATAACCAAT
H3,shflip-h15-b136,GCTAGGGCCGATTAAAACTACGTGTCAAGTTTCACACAAC
A4,shflip-h3-b160,GGGAAGAACAGATACATAACGCCAATGTTTAG
B4,shflip-h5-b160,ACTGGATAGGAAGCCCGAAAGACTTGATAAGA
C4,shflip-h7-b160,GGTCATTTGCATCAATTCTACTAAGGAGAAGC
D4,shflip-h9-b160,CTTTATTTATAATCAGAAAAGCCCCATCAAAA
E4,shflip-h11-b160,ATAATTCGTAAGTTGGGTAACGCCCGGAAGCA
F4,shflip-h13-b160,TAAAGTGTATCAGGGCGATGGCCCGGGATTTTAGACAGGA
A5,shflip-h5-b176,ATACTGCGCACATTCAACTAATGAAATCTA
B5,shflip-h7-b176,TGGCTTAGTCAAAAAGATTAAGAGCGTCCA
C5,shflip-h9-b176,AGGATAAACGAGCTGAAAAGGTGTTGCGGA
D5,shflip-h11-b176,CCTTCCTGTATGTACCCCGGTTGCAACGCA
E5,shflip-h13-b176,GGGGTGCCGTGCTGCAAGGCGATCGTCTGG
F5,shflip-h15-b168,ACGGTACGCCAGAATGGCGAAAAACCGTCTAAAGCCT
A6,shflip-h3-b184,AAACGAACTAACGGAGATCGAGATTTAGGAATACGAATCGTC
B6,shflip-h5-b192,ATAAATATAGCAAAGCGGATTGCAAGCTTAAT
C6,shflip-h7-b192,TGCTGAATATTTTCATTTGGGGCGAATTTTTA
D6,shflip-h9-b192,GAACCCTCACTAGCATGTCAATCATAGCCAGC
E6,shflip-h11-b192,TTTCATCACTGGCGAAAGGGGGATTAATGAGT
F6,shflip-h13-b192,GAGCTAACGACTCCAACGTCAAAGCCTGAGAAGTGTTTTT
A7,shflip-h7-b208,GTAGCTCAGACTATTATAGTCAGATCATTGAA
B7,shflip-h9-b208,TAAATGCATAACCTGTTTAGCTATATAATGCT
C7,shflip-h11-b208,TGTGAGCGGAACGGTAATCGTAAAATATATTT
D7,shflip-h13-b208,ATTGCGTTTTCGCTATTACGCCAGACATTAAA
E7,shflip-h15-b200,ATAATCAGTGAGGCCAACTATTAAAGAACGTGTCACATTA
A8,shflip-h5-b216,AAATGCTTTAAACAGTCAGGTCTTTACCCTACATGTT
B8,shflip-h7-b224,TTAAATATTTCGCAAATGGTCAAATGCCTG
C8,shflip-h9-b224,AGTAATGTAACAAGAAATCGATAGTAACA
D8,shflip-h11-b224,ACCCGTCGATCGGTGCGGGCCTCGCGCTCA
E8,shflip-h13-b224,CTGCCCGCTTGGAACAAGAGTCCCCGAGTAAAAGAGTC
A9,shflip-h9-b240,AGATTCAATGACCATTAGATACATGCAACTAA
B9,shflip-h11-b240,GTGGGAACCTGAGAGTCTGGAGCAGTAGGTAA
C9,shflip-h13-b240,CGGGAAACACTGTTGGGAAGGGCGGATTCTCC
D9,shflip-h15-b232,TGTCCATCACGCAAATTGAGTGTTGTTCCAGTTTTCCAGT
E9,shflip-h7-b248,GTCTGGAAGTTTCATTACGAGTAGATTTAGTTAAGGGTGA
F9,shflip-h9-b256,GAAAGGCCGCTATCAGGTCATTGCAAACGGCG
G9,shflip-h11-b256,GATTGACCCCATTCAGGCTGCGCACTGTCGTG
H9,shflip-h13-b256,CCAGCTGCTAGCCCGAGATAGGGTTAACCGTTGTAGCAAT
A10,shflip-h15-b264,ACTTCTTTGATTAGTTATAAATCAAAAGAAATTAATG
B10,shflip-h11-b272,GATAGGTCGAGAGATCTACAAAGGGAGACA
C10,shflip-h13-b272,AATCGGCCGCAAAGCGCCATTCGGTAATGG
D10,shflip-h9-b280,ACCATCAATATGATATGAGGGTAGCTATTTTTACGTTGGT
E10,shflip-h11-b288,GTAGATGGTGGTGCCGGAAACCAGAACGCGCG
F10,shflip-h13-b288,GGGAGAGGAATCGGCAAAATCCCTAATAACATCACTTGCC
G10,shflip-h13-b304,GTATTGGGTTTCCGGCACCGCTTCGCGCATCG
H10,shflip-h15-b296,TGAGTAGAAGAACTCATTGATGGTGGTTCCGACGGTTTGC
A11,shflip-h11-b312,CATCTGCCAGTTTGACGCACTCCAGCCAGCCGCCAGG
B11,shflip-h13-b320,GTGGTTTTAGGCGAAAATCCTGTAACTATCGGCCTTGC
C11,shflip-h15-b328,TGGTAATATCCAGAACGCTGGTTTGCCCCAGCTCTTTTCA
D11,shflip-h13-b344,ACGGGCAACAGCTGATGCAGCAAGCGGTCCACAATATTAC





edge staples

arm 1

A1, shflip-h1-b72,CCGGAACGAGGCGCACTCCATGTTACTTAG
B1, shflip-h3-b72,CTTGAGATGGTTTAAGAGTAGTAAATTGGG
C1, shflip-h3-b104,TGTGAATTACCTTATGATAAGGCTTGCCCTGAGGAACCGA
D1, shflip-h5-b104,CGAGAGGCTTTTGCAAATAAAAACCAAAATAG
E1, shflip-h7-b104,CAAACTCCAACAGGTCCGAACCAGACCGGAAG
F1, shflip-h9-b104,CTAAATCGGTTGTACCGCCTCAGAGCATAAAG
G1, shflip-h11-b104,CATTAAATTTTTGTTAATTTTGTTAAAATTCG
H1, shflip-h13-b112,AAATTGTTAGCTTGCATGCCTGCAGGTCGACTCTAGAGGAATGGTCAT
A2, shflip-h13-b72,CTCGAATTCGTAATCTCCCCGGGTACCGAG
B2, shflip-h15-b72,CTTGACGGGGAAAGCGCCCCCGATTTAGAG

arm 2

A3, shflip-h33-b56,CTATTTTGCACCCAGCAAATCAAGATTAGTTG
B3, shflip-h17-b56,CAAATCAGATATAGAACGCGCCCAATAGCAAG
C3, shflip-h20-b47,AATATCCAATAATCGGCTGTCTTTCCTTATCATTCCAAAGATAAGT
D3, shflip-h19-b24,CATGTAGAAACCAATCCATCCTAATTTACGAG
E3, shflip-h21-b24,CCAGTAATAAGAGAATCAGAGGCATTTTCGAG
F3, shflip-h23-b24,CGTTAAATAAGAATAACGTGTGATAAATAAGG
G3, shflip-h25-b24,CTGAGAAGAGTCAATAAGCTTAGATTAAGACG
H3, shflip-h24-b63,ATTTAATGTCATAGGTCTGAGAGATTTCCCTTAGAATCCT
A4, shflip-h27-b56,CAAAAGAAGATGATGAATTTCAATTACCTGAG
B4, shflip-h29-b56,CGTAGATTTTCAGGTTAGAAATAAAGAAATTG

bridge staples

strands adjacent to bridges

plate shape-flip-bridges
A1, shflip-h0-b156,ACCAAGACCAGGCGCATAGGCAAGAACCGGATATTCAAGGACGTT
B1, shflip-h3-b176,CGTTAATAAAGAGTAATCTTGACTGGCT
C1, shflip-h5-b208,TCCCCCTCAGAAAGATTCATCAGTCAACA
D1, shflip-h7-b240,AGTACGGTGACCATAAATCAAAAATTCAG
E1, shflip-h9-b272,GTCAAATCTTCCCAATTCTGCGACCATA
F1, shflip-h11-b304,TAACCGTGATAAATTAATGCCGGATCAAC
G1, shflip-h13-b336,CCAGTGAGTCGGCCTCAGGAAGATGGGGA
H1, shflip-h15-b352,CGCCAGCCATTGCAACAGGAAAAGGCCCTGAGAGAGTTTGCCC
A2, shflip-h18-b364,CTATCGAGCAAGAAACAATGAAGTCACACGACCAGTAATAAAAGG
B2, shflip-h20-b332,GCAGCGAAACGTCACCAATGATTAAGAAA
C2, shflip-h22-b300,GATGACCAGTAAGCGTCATAACAGAATC
D2, shflip-h24-b268,CTCATAGAACCGCCACCCTCATAATAAGT
E2, shflip-h26-b236,TTTTCCAACTAAAGGAATTGCGCCCAATA
F2, shflip-h28-b159,GAACGAGAAGACTTTTTCATGATGCCACTACGAAGGCACTAA
G2, shflip-h30-b172,CGTAAGGAAGTTTCCATTAAACCTCAGCA
H2, shflip-h28-b204,CCGCTGCTGAGGCTTGCAGGAAAAAAAG

bridges

A3, shflip-h31-b157,AACACTCATCCCAGCGATTAT
B3, shflip-h1-b173,GACCTTCATCCGGGTAAAATA
C3, shflip-h3-b205,TTATTACAGGTGAGTTAAAGG
D3, shflip-h5-b237,AAAACGAGAATGAATAATAATT
E3, shflip-h7-b269,TAACAGTTGAGAGCCACCACC
F3, shflip-h9-b301,CGTTCTAGCTGCATGGCTTTT
G3, shflip-h11-b333,CGACGACAGTAAACCATCGATA
H3, shflip-h13-b365,TTCACCGCCTAATAGCAATAG
A4, shflip-h15-b397,TTTTGACGCTCGAAATGGAT

strands at corner

B4, shflip-h15-b376,ACGCTCATGGAAATACCTACA
C4, shflip-h16-b396,ATTTACATTGGCAGATTCACC

Sequence diagram for 60° corner origami with edge shapes and reversed stacking polarity