

Increasing Redundancy Exponentially Reduces Error Rates During Algorithmic Self-Assembly

Rebecca Schulman,^{*, †, ‡} Christina Wright^{§, ¶} Erik Winfree^{||}

1 Controls and Additional Data

1.1 Error rates in inner vs. outer ribbon rows

To test whether the unusually high error rate observed for the 1-redundantly encoded row within the the 3-redundant ribbon was caused by a higher error rate in rows adjacent to boundary tile layers, we tabulated the error rate for 1-redundant copying in “inner” rows (those not adjacent to a boundary tile) vs. “outer” rows (those adjacent to a boundary layer.) Our results show no discrepancy between error rates in these rows; in inner rows, the error rate was 0.0097 ± 0.0017 per bit and in outer rows, the error rate was 0.0096 ± 0.0017 per bit.

*Computation and Neural Systems, California Institute of Technology

†Current address, Chemical and Biomolecular Engineering and Computer Science, Johns Hopkins University

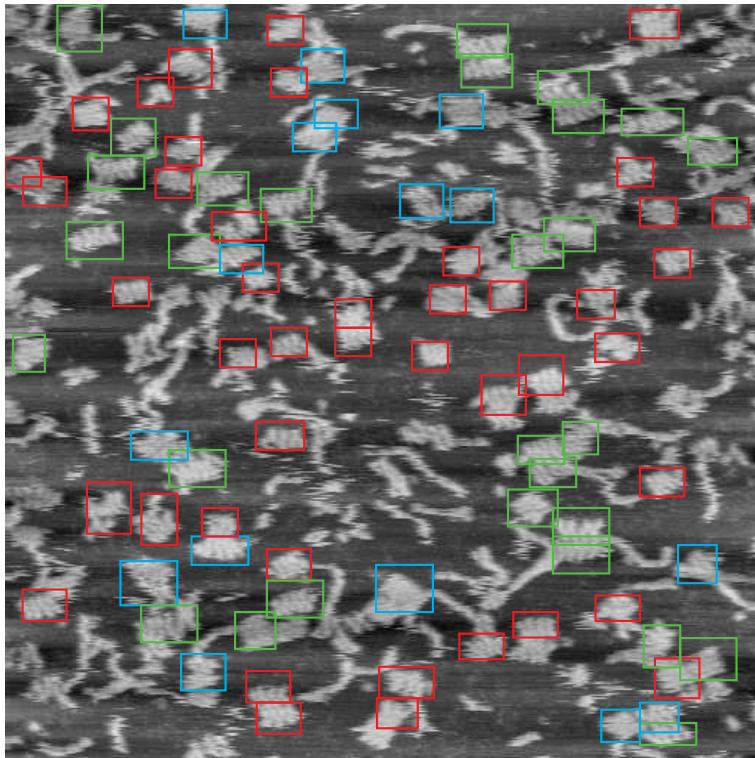
‡E-mail: rschulm3@jhu.edu

§Massachusetts Institute of Technology

¶Current address, Google, Inc.

|| Computer Science, Computation and Neural Systems, and Bioengineering, California Institute of Technology

1.2 Assembly yield of crystal seeds



Supplemental Figure S1: AFM image (scan size of 500 nm) used to measure the synthesis yield and synthesis quality of crystal seed structures. Structures 5 tiles in height were considered well-formed, structures 3-4 tiles in height were considered malformed, and smaller structures were considered irrelevant for determining the synthesis quality. Green boxes indicate well-formed structures, red boxes indicate malformed structures and blue boxes indicate relevant structures for which the height in tiles could not readily be discerned. To estimate an upper bound for synthesis yield, we counted the number of pixels (i.e. area) of all presumably-DNA structures (i.e. above the mica background) as well as the number of pixels within the green boxes (minus half the overlap with red or blue boxes), and took the ratio; this gave an upper bound of 8%. To estimate the synthesis quality, we counted the number of green-or-red boxes and the number of green boxes, and took the ratio; this estimate ignores structures that could not be classified and came out to be 39%. Assuming all unclassifiable structures (blue boxes) are malformed would imply a synthesis quality of 32% and assuming all unclassifiable structures are well-formed would imply a synthesis quality of 49%. In the main text we report a synthesis quality of approximately 40%.

2 Strand design and sequences

2.1 Sequence Design

The sequences for the double tiles and the double stranded regions of the “0” tiles were those used in earlier experiments [1]. The sequences for the 8 “1” tiles were newly designed. Design began with the structure of the tiles: the size and placement of the hairpins on the “1” tiles, used for contrast under the atomic force microscope, was identical to those used previously [2].

The design of the sequences themselves proceeded according to the sequence design method described in detail previously [1]. The goal of the design process was to ensure that areas designed to hybridize would do so while minimizing spurious interactions between regions that were designed not to interact [3, 4]. Sticky ends were also designed with the criterion that their binding energies should be similar to one another; binding energies were predicted using the nearest neighbor model of hybridization [5]. The design of all new sequences was done to ensure they would be compatible with existing DNA using these criteria. Additionally, the designed sticky end sequences were arranged on the tiles in such a way that all sticky ends encoding “0” and “1” in the same row have 0 cross-hybridization energy of 0 according to the nearest neighbor model.

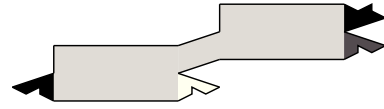
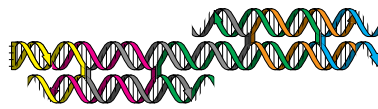
The crystal seeds in the tile rows that nucleated the ribbon used sequences were identical to those used in the “0” tiles in each row of the ribbons. The tail region of the seed consisted of sequences that were used in a previous experiment [6] and were also designed to minimize spurious interactions with the sequences of the ribbon tiles.

2.2 Tile sequences

Ribbons were constructed from two types of tiles. Double tiles run along the top and bottom rows and “single” tiles assemble in the middle 4 rows of each ribbon. Each experiment used two types of double tiles, 8 tile types that made up the blocks propagating the “0” bit and 8 tile types that made up the blocks propagating the “1” bit. To create the 1-, 2-, 3- and 4-redundant tile sets, the single-stranded (“sticky”) ends of the tiles were altered, but tile sequences otherwise remained the same in all experiments. Sticky ends that changed between experiments are denoted nnnnn here.

2.2.1 Boundary tiles

Z78



```

                    5 <gaagcagg-acaagcga\ctcagtgt-ccgattgg> 3
                    6 >catcac-cttcgtcc tgttcgct/\gagtcaca ggctaacc-tccaa< 4
                                | |           Z78           | |
1  /tt-gg-tcttg 2>cattctgg-tgaccata\tctatcct-ccgatgac-gacag-ccgtgccca ggtagcgg\tacactcc tgcttctg-ttct> 4
  \tt-cc-agaac----gtaagacc actggtat/\agatagga ggctactg-ctgtc-ggcacggt-ccatcgcc/\atgtgagg-acgaagac< 3
                                | |           Z78           | |
1  >aggtt-ctaccgca gcctattc\tgacgtgg tgccctagc-acctt< 5
2  <gatggcgt-cggataag/\actgcacc-acggatcg> 6

```

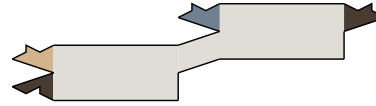
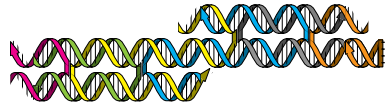
Strands:

```

1  aggttctaccgcaccagaatgcaagaccttttggtcttg
2  cattctgggtgaccataagataggaggtgcagtgaataggctgcggtag
3  cagaagcaggagtgtaggcgatggtgttcgctctcagtgtccgattgg
4  aacctccaatcggtgcttctgttcct
5  ttccacgatccgtggctactgctgtcggcacggtccatcgccctacactccacactgagagcgaacaggacgaag
6  cataccttcgtccaccgtgccgacagcagtagcctcctatcttatggtcagcctattcactgcaccacggatcg

```

Z56



```

5 <gcaagcgt-ccacttgg\gcagtagg-acgcctcg> 3
6 >gtgat-cgttcgca ggtgaacc\cgtcatcc tgcggagc-gcaat< 4
      | |          Z56          | |
2 >gtttgagg-acgctatg\|tttaggt-ccatgagc-acgaa-cgaaagcc tgagctag\|tccagaca ggtcatcg---aaggc-cc-tt\ 4
1 <ctgta-caaactcc tgcgatac\|aacatcca ggtactcg-tgctt-gctttcgg-actcgatc\|aggctgt-ccagtagc<3 <ttccg-gg-tt/
      | |          Z56          | |
1 >cgta-gctcggca ggtgtctc\|acgaatcc tggttacg-aaggc< 5
2 <cgagccgt-ccacagag\|tgcttagg-accaatgc-ttccg> 6

```

Strands:

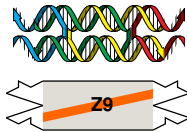
```

1 cgttagctcggcacctcaaacaatgtc
2 gtttgaggacgctatgaacatccacctaaagcagagacacctgccgagc
3 cgatgacctgtctggagatcgagtggtgaaccgcagtaggacgcctcg
4 taacgcgagggcgtggtcatcgaagggccttttgggcctt
5 cggaagcattgggtggtactcgtgcttgctttcggactcgatctccagacacctactgcggttcacctgcgaacg
6 gtgatcgttcgcaccgaaagcaagcagcagtagtacctggatggtcatagcgtggtgtctctgcttaggaccaatgcttccg

```

2.2.2 "0" tiles

Z9



```

3 >gagcgagt-ccatatca\|tacttagg-acgactgg< 2
4 <nnnn-ctcgtca ggtatagt\|atgaatcc tgcctgacc-nnnn> 1
      | |          Z9          | |
4 >nnnn-gttcatcc tgcgact\|tcaacgca ggctattg-nnnn< 1
3 <caagtagg-acgctgca\|agtgcgt-ccgataac> 2

```

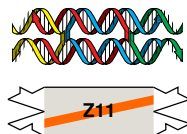
Strands:

```

1 nnnnngttatcgggtgctgaccnnnnn
2 ggtcagcaggattcattgatatggtgcgacgtagttgcgtccgataac
3 gagcaggtccatatacaatgaatccacgcaactacgtcgcaggatgaac
4 nnnnngttcatccactcgtcnnnnn

```

Z11



```

3 <gagaatca-ggctctca\|atctagcc-tggttcgg> 2
4 >nnnn-ctcttagt ccgagagt\|tagatcgg accaagcc-nnnn< 1
      | |          Z11          | |
4 <nnnn-gctgaagg actattgt\|ttttagt cccttag-nnnn> 1
3 >cgacttc-tgataaca\|aacatca-gggaaatc< 2

```

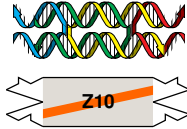
Strands:

```

1 nnnnccgaaccaccctttagnnnnn
2 ctaaagggactacaaatggtatcaccgagagtagtctagcctgggttcgg
3 cgacttcctgataacatttgtagtggttagatctctcggactaagag
4 nnnnctcttagtggaagtcgnnnnn

```

Z10



```

3 >gaatgagg-actgagta\tccgctgt-cccaaat< 2
4 <nnnn-cttactcc tgactcat/\aggcgaca gggtttag-nnnn> 1
      | |      Z10      | |
4 >nnnn-ctagaaca ggaacgc/\tagatgcc tgaagacg-nnnn< 1
3 <gatctgt-cctttgcg/\atctacgg-acttctgc> 2

```

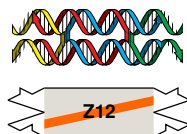
Strands:

```

1 nnnnngcagaagtgggtttagnnnnn
2 ctaaaccctgtgccttactcagtggaaacgcacatctacggacttctgc
3 gaatgaggactgagtaaggcgacaccgtagatgcgtttctgttctag
4 nnnnnctagaacacctcattcnnnnn

```

Z12



```

3 <caaagcgg-acaacgta\ttccatgt-ccttagac> 2
4 >nnnn-gtttcgcc tgttgc/\aaggtaca ggaatctg-nnnn< 1
      | |      Z12      | |
4 <nnnn-gctctaca ggcattag/\gttatgcc tgtatgc-nnnn> 1
3 >cgagatgt-cgtaatc/\caatcgg-acatagcg< 2

```

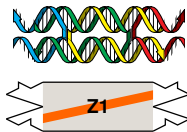
Strands:

```

1 nnnnngtctaaggtgtatcgcnnnnn
2 gcgatacaggcataacgattacggtggttcatttccatgtccttagac
3 cgagatgtccgtaatcgttatgccacatggaaatgcaacaggcgaaac
4 nnnnngtttcgccacatctcgnnnnn

```

Z1



```

2 >cagagtgg-acgaaagc/\agtccgt-ccgatgtc< 3
1 <nnnn-gtctcacc tgcttctg/\tcacggca ggtacag-nnnn> 4
      | |      Z1      | |
1 >nnnn-caaacgca ggaacctg/\tatgaacc tgctcaac-nnnn< 4
2 <gtttgcgt-ccttggac/\atacttgg-acgagttg> 3

```

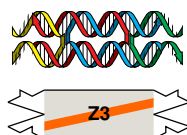
Strands:

```

1 nnnnncaaacgcaccactctgnnnnn
2 cagagtggacgaaagctcacggcaccaagtatcaggttcctgcgtttg
3 ctgtagcctgccgtgagctttcgtggaacctgatacttggacgagttg
4 nnnnncaactcgtggctacagnnnnn

```

Z3



```

2 <gtcggta-ggctcgtc/\acgacacc-tgagacgg> 3
1 >nnnn-cagccagt ccgagcag/\tgctgtgg actctgcc-nnnn< 4
      | |      Z3      | |
1 <nnnn-gaggatgg acgcttag/\tctgtagt ccgcattg-nnnn> 4
2 >ctctacc-tgcaatc/\agacatca-ggcgtaac< 3

```

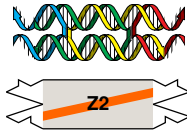
Strands:

```

1 nnnnncagccagtggtaggagnnnnn
2 ctctacctgcgaatctctgtagtgggtgctgctcgtcggactggctg
3 caatgcggaactacagagattcgcaccgagcagacgacacctgagacgg
4 nnnnncgctctcaccgcattgnnnnn

```

Z2



```

2 >gatgatgt-ccttgtaa\tgaagcgg-acaacgag< 3
1 <nnnnn-ctactaca ggaacatt/\acttcgcc tgttgctc-nnnnn> 4
      | |      Z2      | |
1 >nnnnn-gaacgacc tgattgcg\taatctca ggcattcg-nnnnn< 4
2 <cttgctgg-actaacgc/\attagagt-ccgtaagc> 3

```

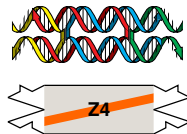
Strands:

```

1 nnnnngaacgaccacatcatcnnnnn
2 gatgatgtccttgtaaacttcgccactctaatacgcaatcaggtcgttc
3 gagcaacaggcgaagtttacaaggtgattgcgattagagtcgtaagc
4 nnnnngcttacggtgttgctcnnnnn

```

Z4



```

2 <cgtagg-acattgca\cggcttgt-cggttcg> 3
1 >nnnnn-ggcaatcc tgaactg/\gccgaaca ggcaagcg-nnnnn< 4
      | |      Z4      | |
1 <nnnnn-cgccaaca ggttgaat\ccagatcc ttagagc-nnnnn> 4
2 >gcggttgt-ccaactta\ggtctagg-acatctcg< 3

```

Strands:

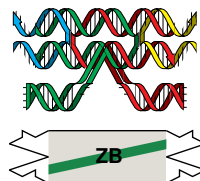
```

1 nnnnnggcaatccacaaccgcnnnnn
2 gcggttgtccaacttaccagatccacaagccgacggttacaggattgcc
3 gctctacaggatctggtaagttggtgtaacgtcggcttgtccggttcgc
4 nnnnngcgaacggtgttagagcnnnnn

```

2.2.3 "1" tiles

ZB



```

2 >gatcggat-ccgaccat\ggatcgtg-agttggtg< 3
1 <nnnnn-ctacgcta ggctggta\cctagcac tcaaccac-nnnnn> 4
      | |      ZB      | |
1 >nnnnn-gtactcc tactctgc\tccgacga ggcatagg-nnnnn< 4
2 <cactgagg-atgagacg\aggctgct-ccgtatcc> 3
      / | | \
      /tt-cgcttcggtt || ttcagcaccg-tt\
      \tt-gcgaagcc--/ \-gtcgtggc-tt/

```

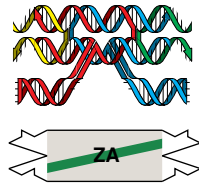
Strands:

```

1 nnnnngtgactccatcgcatcnnnnn
2 gatcggatccgaccatcctagcacagcagcctccgaagcggttttcgcttcggttcgagagtaggagtcac
3 gtgggttgagtgctaggatggcggtaactctgcgctcgtggcttttgccacgacttaggctgctccgtatcc
4 nnnnnggatcgggtcaaccacnnnnn

```

ZA



```

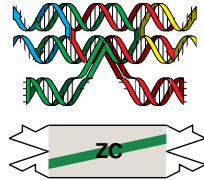
2 <gactggaa-gtggtcgg\attcggac-ttggaggc> 3
1 >nnnnn-ctgacctt caccagcc\taagcctg aacctcgg-nnnnn< 4
      | |      ZA      | |
1 <nnnnn-gcacgacg agtccgag\agtaggtt cagcttcg-nnnnn> 4
2 >cgtgctgc-tcaggctc\tcatccaa-gtcgaagc< 3
      / || \
      /tt-cgctctgc || ggcacggc-tt\
      \tt-gcgagagctt\ttccgtgccg-tt/

```

Strands:

- 1 nnnnnctgaccttgacgacgnnnnn
- 2 cgtgctgctcaggctccgtctcgcttttgcgagacgttagtaggttgcgaatggctggtgaaggtcag
- 3 cgaagctgaacctactggcacggcttttgcctgccttgagcctgacaccagccattcggacttgaggc
- 4 nnnnngcctccaacagcttcgnnnnn

ZC



```

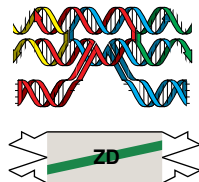
2 >gagtgtgg-aacctgtc\gttctgtc-ccgctgac< 3
1 <nnnnn-ctcacacc ttggacag\caagacga ggcactg-nnnnn> 4
      | |      ZC      | |
1 >nnnnn-cgttcga ggtggctg\tccacgcc tctcctcg-nnnnn< 4
2 <gcaaggct-ccaccgac\aggtgctgg-ataggagc> 3
      / || \
      /tt-cgcttgctt || ttggcatgct-tt\
      \tt-gcgaaacgg-/\ --ccgtagcg-tt/

```

Strands:

- 1 nnnnncgttccgaccacactcnnnnn
- 2 gagtgtggaacctgtccaagacgaccgcacctggcaagcgttttcgcttgcttcagccacctcggaaacg
- 3 cagtgcctcctgctcttgacaggttggctggctgcccgtagcgttttcgctacggttaggtgctggataggagc
- 4 nnnnngctcctatggcgactgnnnnn

ZD



```

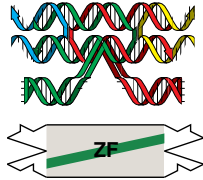
2 <cgatgcgg-aagctct\ggtctggt-ccgttgcg> 3
1 >nnnnn-gctacgcc ttgcaga\ccagacca ggcaacgc-nnnnn< 4
      | |      ZD      | |
1 <nnnnn-gacgaaca ggagcttg\ggaatgcc tgatggac-nnnnn> 4
2 >ctgcttgt-cctcgaac\ccttacgg-actacctg< 3
      / || \
      /tt-gtcgctgg || gttgagcg-tt\
      \tt-cagcgacctt\ttcaacctgc-tt/

```

Strands:

- 1 nnnnngctacgccacaagcagnnnnn
- 2 ctgcttgtcctcgaacggctcgctgttttcagcgaccttggaaatgccaccagacctctgcgaaggcgtagc
- 3 gtccatcaggcattccgtggagcgttttcgctccacttgcttcgaggttcgcagaggtctggctccgttgcg
- 4 nnnnncgcaacggatgatggacnnnnn

ZF



```

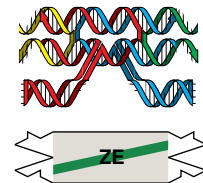
2 >ctgtgccg-agcatcgc\|gtcgtgat-ctcggaac< 3
1 <nnnnn-gacacggc tctagcgc\|cagcacta gagccttg-nnnnn> 4
      | |      ZF      | |
1 >nnnnn-ctgacaga gctacgag\|tccgcacc tggctactc-nnnnn< 4
2 <gactgtct-cgatgctc\|aggcgtgg-accatgag> 3
      / || \
      /tt-cggagacgctt || ttgccttggc-tt\
      \tt-gcctctgc--/ \--cggaacgg-tt/

```

Strands:

- 1 nnnnnctgacagacggcacagnnnnn
- 2 ctgtgccgagcatcgccagcactaccacgcctcgtctccgttttcggagacggttctcgtagctctgtcag
- 3 caaggctctagtgtggcgatgctgctacgagcggaacggttttcggttccggttaggcgtggaccatgag
- 4 nnnnnctcatggtgagccttgnnnnn

ZE



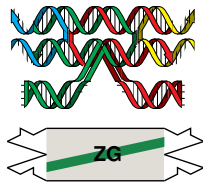
```

2 <gacgtaga-gcgacct\|ttcggatc-tggcaacg> 3
1 >nnnnn-ctgcatct cgctggag\|aacctag accgttc-nnnnn< 4
      | |      ZE      | |
1 <nnnnn-gactcctg agagctgc\|ggtcaggt ctcacagg-nnnnn> 4
2 >ctgaggac-tctcgacg\|ccagtcca-gagtgtcc< 3
      / || \
      /tt-gtgcctcg || gacaccgc-tt\
      \tt-cacggagctt\|ttctgtggcg-tt/

```

- 1 nnnnnctgcatctgtcctcagnnnnn
- 2 ctgaggactctcgacggctccgtgttttcacggagcttggtcaggtgatccgaactccagcgagatgcag
- 3 cctgtgagacctgaccgacaccgcttttgoggtgtcttcgtcgagacgctggagttcggatctggcaacg
- 4 nnnnnctgttgcactcacaggnnnnn

ZG



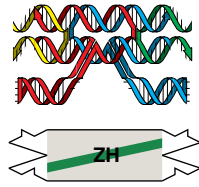
```

2 >gaagtct-cgagactg\|atcgaccg-aaggtccg< 3
1 <nnnnn-cttcagga gctctgac\|tagctggc ttcaggc-nnnnn> 4
      | |      ZG      | |
1 >nnnnn-gccaaggc tgagcacc\|ctgcctaa gctgatcg-nnnnn< 4
2 <cggttccg-actcgtgg\|gacggatt-cgactagc> 3
      / || \
      /tt-cggcatcgtt || ttcacaggcg-tt\
      \tt-gccctagc--/ \--gtgtccgc-tt/

```

- 1 nnnnngccaaaggcaggacttcnnnnn
- 2 gaagtctcgcgagactgtagctggcaatccgtccgatgccgttttcggcatcgttgggtgctcagccttggc
- 3 gcctggaagccagctacagtctcgtgagcaccgtgtccgcttttcggacacttgacggattcgactagc
- 4 nnnnnctagtctgttccaggcnnnnn

ZH



```
2 <ctgtgagg-atgctccg\agatcggg-caaggacc> 3
1 >nnnnn-gacactcc tacgaggc\tctagcca gttcctgg-nnnnn< 4
      | |      ZH      | |
1 <nnnnn-caactcca ggtcaacc\cacagtcc tgtctcac-nnnnn> 4
2 >gttgaggt-ccagttgg\gtgtcagg-acagagtg< 3
      / || \
      /tt-gcctcagc || gcgagccg-tt\
      \tt-cggagtcggt\ttcgctcggc-tt/
```

```
1 nnnnngacactccacctcaacnnnnn
2 gttgaggtccagttggcgactccgttttcgagtcggtccacagtcaccgatctgcctcgtaggagtgtc
3 gtgagacaggactgtggcgagccgttttcggtcgcgttccaactggtacgaggcagatcggtcaaggacc
4 nnnnnggtccttgtgtctcacnnnnn
```

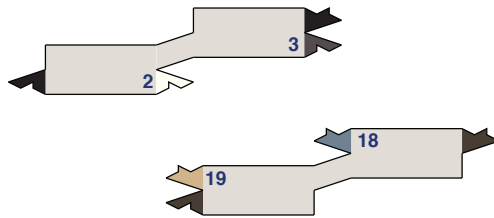
2.3 Tile schematics

This section shows how the sticky ends were programmed to produce the four levels of redundancy. To accomplish this, the same tile cores were used and the row in which each tile could assemble remained the same, but the sticky ends of some tiles changed. The following schematics show which sticky ends were used in which experiment and the function of each of the sticky ends.

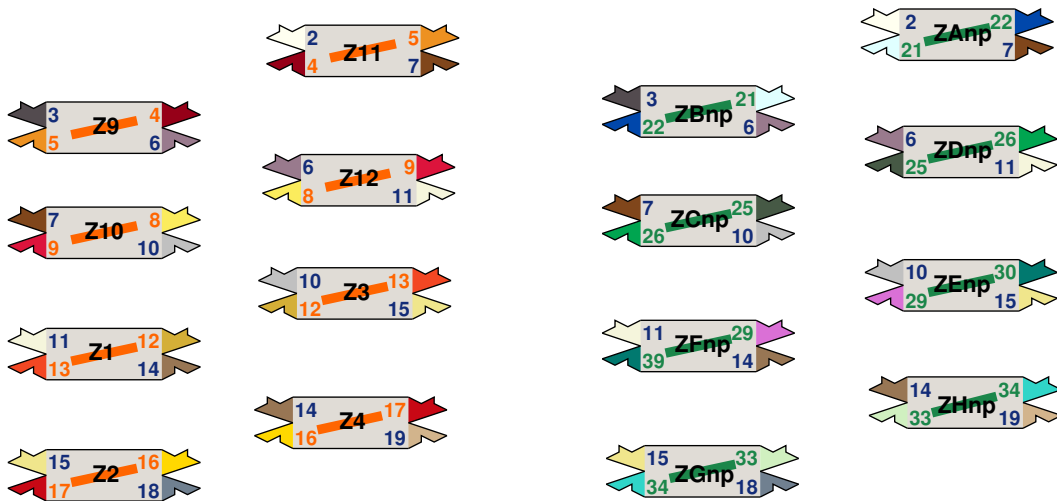
Numbers next to each claw denote which sticky end was used (these are also indicated by the color of the claws, but the numbers can be more easily matched to the sticky end sequences given below than can subtle color changes). The colors of the numbers indicate whether the sticky end type is used only in 0 tiles (orange), only in 1 tiles (green) or in tiles encoding both bits (blue). While the sticky ends for the 0 tiles were the same in all experiments, the colors of some of the 0 tile sticky ends change in the schematics below depending on whether they are shared by the 1 tiles or not.

While each sticky end in a matching pair have the same number label, the two sticky ends have complementary (as opposed to identical) sequences.

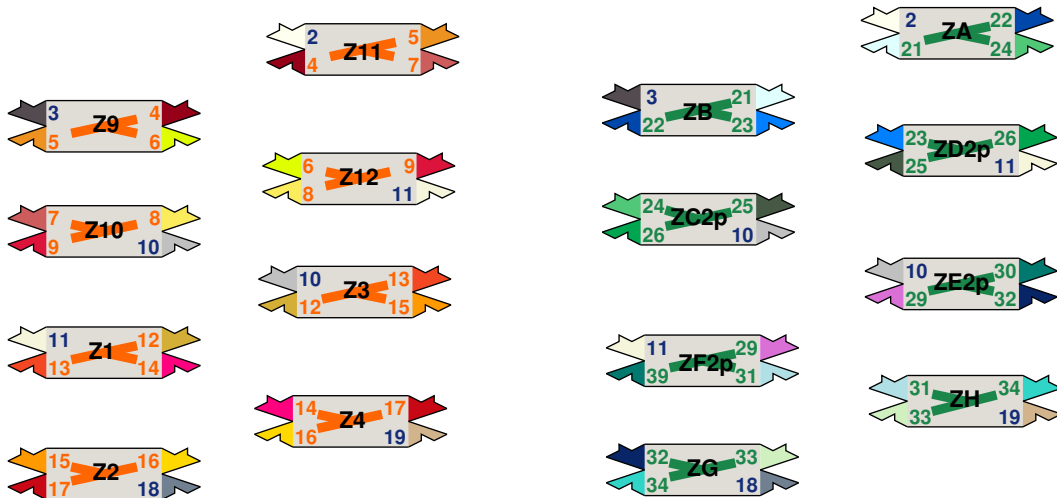
Schematic arrangement of and sticky ends for the boundary tiles, which are included the 1-redundant, 2-redundant, 3-redundant and 4-redundant tile sets:



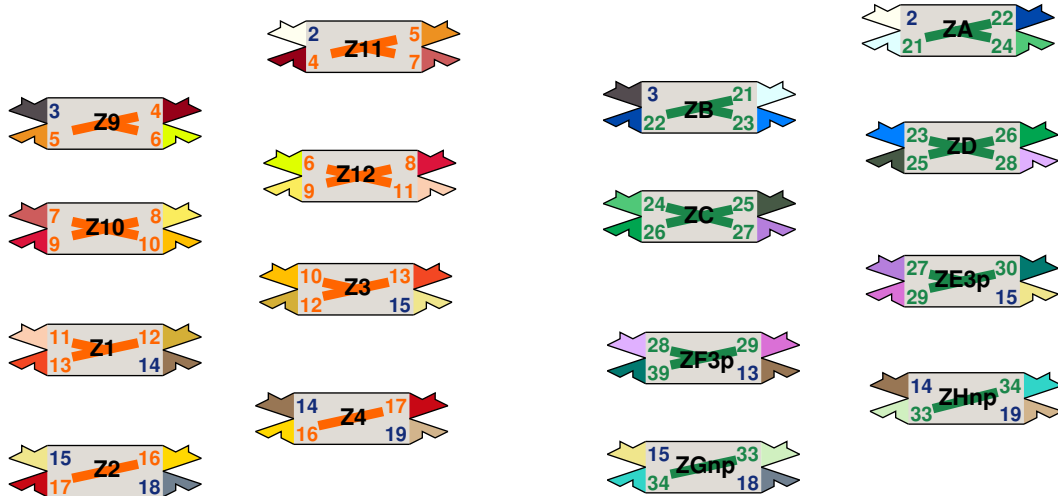
Schematic arrangement of and sticky ends for the 1-redundant proofreading tile set:



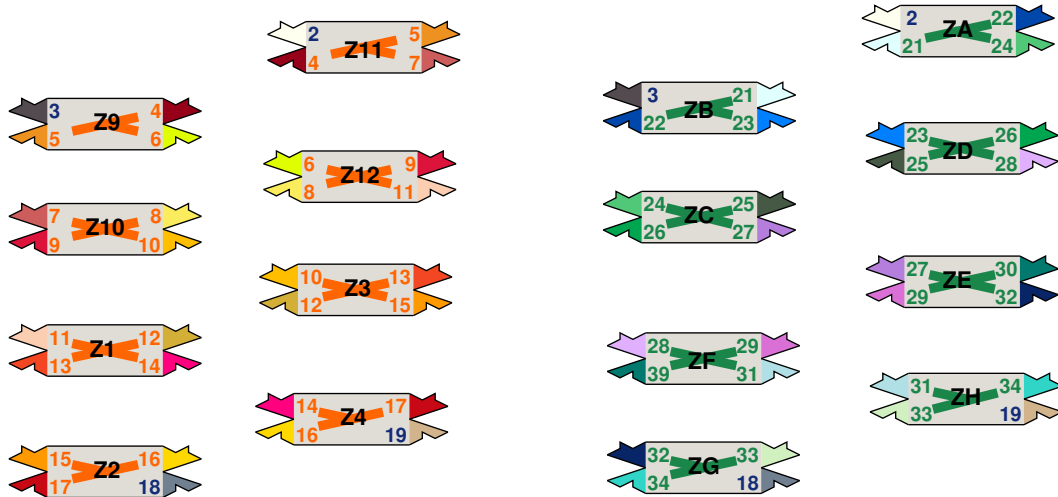
Schematic arrangement of and sticky ends for the 2-redundant proofreading tile set:



Schematic arrangement of and sticky ends for the 3-redundant proofreading tile set:



Schematic arrangement of and sticky ends for the 4-redundant proofreading tile set:



2.4 Tile sticky end sequences

1-redundant “0” tile sticky end sequences

Z1-1	13 — 11	cttgt—caaacgcaccactctg—ttggt
Z1-4	14 — 12	atgct—caactcgtggctacag—aaagag
Z2-1	17 — 15	aagtc—gaacgaccacatcatc—cgatt
Z2-4	18 — 16	atcac—gcttacgggtgttgctc—gctaa
Z3-1	10 — 12	accaa—cagccagtggtaggag—ctctt
Z3-4	13 — 15	acaag—ccgtctcaccgcattg—aatcg
Z4-1	14 — 16	agcat—ggcaatccacaaccgc—ttagc
Z4-4	17 — 19	gactt—gcgaacgggtgtagagc—gacat
Z9-1	6 — 4	ggtaa—gttatcgggtgctgacc—agaac
Z9-4	5 — 3	tgagt—gttcatccactcgctc—aggaa
Z10-1	10 — 8	ttggt—gcagaagtgggtttag—tgagt
Z10-4	9 — 7	aatgg—ctagaacacctcattc—ggtta
Z11-1	5 — 7	actca—ccgaaccaccctttag—taacc
Z11-4	2 — 4	tggaa—ctcttagtggaagtcg—gttct
Z12-1	9 — 11	ccatt—gtctaaggtgtatcgc—accaa
Z12-4	6 — 8	ttacc—gtttcgccacatctcg—actca

1-redundant “1” tile sticky end sequences:

ZA-1	2 — 21	tggaa—ctgaccttgccagcacg—atgac
ZA_np-4	22 — 7	gagat—gcctccaacagcttcg—taacc
ZB-1	22 — 3	atctc—gtgactccatcgcac—aggaa
ZB_np-4	6 — 21	ggtaa—ggatcgggtcaaccac—gtcat
ZC_np-1	26 — 7	tcaga—cgttccgaccacactc—ggtta
ZC_np-4	10 — 25	ttggt—gctcctatggcgactg—acaca
ZD_np-1	6 — 25	ttacc—gctacgccacaagcag—tgtgt
ZD_np-4	26 — 11	tctga—cgcaacgggtgatggac—accaa
ZE_np-1	10 — 29	accaa—ctgcatctgtcctcag—tacga
ZE_np-4	30 — 15	tcctt—cgttgccactcacagg—aatcg
ZF_np-1	39 — 11	aagga—ctgacagacggcacag—ttggt
ZF_np-4	14 — 29	atgct—ctcatgggtgagccttg—tcgta
ZG_np-1	34 — 15	gctta—gccaaaggcaggacttc—cgatt
ZG-4	18 — 33	atcac—gctagtcgttccaggc—atgac
ZH_np-1	14 — 33	agcat—gacactccacctcaac—gtcat
ZH-4	34 — 19	taagc—ggtccttgtgtctcac—gacat

2-redundant “0” tile sticky end sequences

Z1-1	13 — 11	cttgt—caaacgcaccactctg—ttggt
Z1-4	14 — 12	atgct—caactcgtggctacag—aaag
Z2-1	17 — 15	aagtc—gaacgaccacatcatc—cgatt
Z2-4	18 — 16	atcac—gcttacgggtgttgctc—gctaa
Z3-1	10 — 12	accaa—cagccagtggtaggag—ctctt
Z3-4	13 — 15	acaag—ccgtctcaccgcattg—aatcg
Z4-1	14 — 16	agcat—ggcaatccacaaccgc—ttagc
Z4-4	17 — 19	gactt—gcgaacgggtgtagagc—gacat
Z9-1	6 — 4	ggtaa—gttatcgggtgctgacc—agaac
Z9-4	5 — 3	tgagt—gttcatccactcgctc—aggaa
Z10-1	10 — 8	ttggt—gcagaagtgggtttag—tgagt
Z10-4	9 — 7	aatgg—ctagaacacctcattc—ggtta
Z11-1	5 — 7	actca—ccgaaccaccctttag—taacc
Z11-4	2 — 4	tggaa—ctcttagtggaagtcg—gttct
Z12-1	9 — 11	ccatt—gtctaagggtgatcgc—accaa
Z12-4	6 — 8	ttacc—gtttcgccacatctcg—actca

2-redundant “1” tile sticky end sequences

ZA-1	2 — 21	tggaa—ctgaccttgcagcacg—atgac
ZA-4	22 — 24	gagat—gcctccaacagcttcg—agcat
ZB-1	22 — 3	atctc—gtgactccatcgcatc—aggaa
ZB-4	23 — 21	atgac—ggatacgggtcaaccac—gtcat
ZC-1	26 — 24	tcaga—cgttccgaccacactc—atgct
ZC_np-4	10 — 25	ttggt—gctcctatggcgactg—acaca
ZD-1	23 — 25	gtcat—gctacgccacaagcag—tgtgt
ZD_np-4	26 — 11	tctga—cgcaacgggtgatggac—accaa
ZE_np-1	10 — 29	accaa—ctgcatctgtcctcag—tacga
ZE-4	30 — 32	tcctt—cgttgccactcacagg—cttac
ZF_np-1	39 — 11	aagga—ctgacagacggcacag—ttggt
ZF-4	31 — 29	caaac—ctcatgggtgagccttg—tcgta
ZG-1	34 — 32	gctta—gccaaaggcaggacttc—gtaag
ZG-4	18 — 33	atcac—gctagtcgttccaggc—atgac
ZH-1	31 — 33	gtttg—gacactccacctcaac—gtcat
ZH-4	34 — 19	taagc—ggtccttgtgtctcac—gacat

3-redundant “0” tile sticky end sequences

Z1-1	13—11	cttgt—caaacgcaccactctg—ttggt
Z1-4	14—12	atgct—caactcgtggctacag—aaag
Z2-1	17—15	aagtc—gaacgaccacatcatc—cgatt
Z2-4	18—16	atcac—gcttacgggtgttgctc—gctaa
Z3-1	10—12	accaa—cagccagtggtaggag—ctctt
Z3-4	13—15	acaag—ccgtctcaccgcattg—aatcg
Z4-1	14—16	agcat—ggcaatccacaaccgc—ttagc
Z4-4	17—19	gactt—gcgaacgggtgtagagc—gacat
Z9-1	6—4	ggtaa—gttatcgggtgctgacc—agaac
Z9-4	5—3	tgagt—gttcatccactcgctc—aggaa
Z10-1	10—8	ttggt—gcagaagtgggtttag—tgagt
Z10-4	9—7	aatgg—ctagaacacctcattc—ggtta
Z11-1	5—7	actca—ccgaaccaccctttag—taacc
Z11-4	2—4	tggaa—ctcttagtggaagtcg—gttct
Z12-1	9—11	ccatt—gtctaagggtgatcgc—accaa
Z12-4	6—8	ttacc—gtttcgccacatctcg—actca

3-redundant “1” tile sticky end sequences

ZA-1	2—21	tggaa—ctgaccttgcagcacg—atgac
ZA-4	22—24	gagat—gcctccaacagcttcg—agcat
ZB-1	22—3	atctc—gtgactccatcgcatc—aggaa
ZB-4	23—21	atgac—ggatacgggtcaaccac—gtcat
ZC-1	26—24	tcaga—cgttccgaccacactc—atgct
ZC-4	27—25	ctatc—gctcctatggcgactg—acaca
ZD-1	23—25	gtcat—gctacgccacaagcag—tgtgt
ZD-4	26—28	tctga—cgcaacgggtgatggac—gatag
ZE-1	27—29	gatag—ctgcatctgtcctcag—tacga
ZE_np-4	30—15	tcctt—cgttgccactcacagg—aatcg
ZF-1	39—28	aagga—ctgacagacggcacag—ctatc
ZF_np-4	13—29	atgct—ctcatgggtgagccttg—tcgta
ZG_np-1	34—15	gctta—gccaaaggcaggacttc—cgatt
ZG-4	18—33	atcac—gctagtcggtccaggc—atgac
ZH_np-1	14—33	agcat—gacactccacctcaac—gtcat
ZH-4	34—19	taagc—ggtccttgtgtctcac—gacat

4-redundant “0” tile sticky end sequences

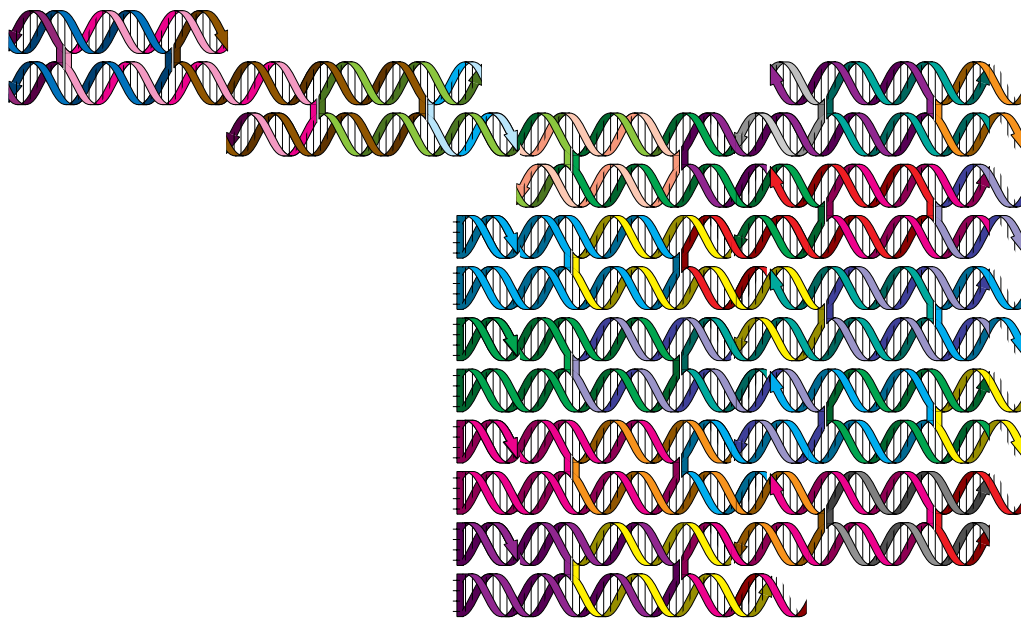
Z1-1	13—11	cttgt—caaacgcaccactctg—ttggt
Z1-4	14—12	atgct—caactcgtggctacag—aaag
Z2-1	17—15	aagtc—gaacgaccacatcatc—cgatt
Z2-4	18—16	atcac—gcttacgggtgttgctc—gctaa
Z3-1	10—12	accaa—cagccagtggtaggag—ctctt
Z3-4	13—15	acaag—ccgtctcaccgcattg—aatcg
Z4-1	14—16	agcat—ggcaatccacaaccgc—ttagc
Z4-4	17—19	gactt—gcgaacgggtgtagagc—gacat
Z9-1	6—4	ggtaa—gttatcgggtgctgacc—agaac
Z9-4	5—3	tgagt—gttcatccactcgctc—aggaa
Z10-1	10—8	ttggt—gcagaagtgggtttag—tgagt
Z10-4	9—7	aatgg—ctagaacacctcattc—ggtta
Z11-1	5—7	actca—ccgaaccaccctttag—taacc
Z11-4	2—4	tggaa—ctcttagtggaagtcg—gttct
Z12-1	9—11	ccatt—gtctaagggtgatcgc—accaa
Z12-4	6—8	ttacc—gtttcgccacatctcg—actca

4-redundant “1” tile sticky end sequences

ZA-1	2—21	tggaa—ctgaccttgcagcacg—atgac
ZA-4	22—24	gagat—gcctccaacagcttcg—agcat
ZB-1	22—3	atctc—gtgactccatcgcatc—aggaa
ZB-4	23—21	atgac—ggatacggccaaccac—gtcat
ZC-1	26—24	tcaga—cgttccgaccacactc—atgct
ZC-4	27—25	ctatc—gctcctatggcgactg—acaca
ZD-1	23—25	gtcat—gctacgccacaagcag—tgtgt
ZD-4	26—28	tctga—cgcaacggatgatggac—gatag
ZE-1	27—29	gatag—ctgcatctgtcctcag—tacga
ZE-4	30—32	tcctt—cgttgccactcacagg—cttac
ZF-1	39—28	aagga—ctgacagacggcacag—ctatc
ZF-4	31—29	caaac—ctcatggtgagccttg—tcgta
ZG-1	34—32	gctta—gccaaggcaggacttc—gtaag
ZG-4	18—33	atcac—gctagtcgttccaggc—atgac
ZH-1	31—33	gtttg—gacactccacctcaac—gtcat
ZH-4	34—19	taagc—ggtccttgtgtctcac—gacat

2.5 Crystal seed schematics

Schematic diagram of the crystal seed structure.




```

1  ttccacgatccgtggctactgctgtcggcacggtccatcgctacactccacactgagagcgaacaggacgaag
2  ctctcgtccaccgtgccgacag
3  cagaagcaggagtgtaggcgatggtgttcgctctcagtgccgattgg
4  aacctccaatcggtgcttctgttcct

```

Weave-11

```

1  acaagagctggcagacttgg
2  ctaccgcaccagaatgaccaagtctgccacatcttccgtgttacggttaggaatcaaccttgctcttgt
3  ttgcacggttgagtcgtaaacacggaagatgaagggtgattcctaaagactatgcaagagccagattgaaacgtgg

```

Weave-12

```

1  ccacgttctcgtcagaggtacagaccattgtgaagtatggaatactggctcttgcatagtcactcaacgtgcaa
2  ggctaactgtgctccg
3  cggagcactctgtaccagcagtcctcatacttcacaatggagttagcc

```

2.6.1 Crystal seed sticky end sequences

A particular sequence is nucleated from the seed by controlling the sticky ends presented by the structure: for each vertical position on the seed, the sticky ends presented match the sticky ends of the tiles that propagate the desired bit. This presentation requires both a long strand (#2) that binds to the “scaffold” of the seed that does not change with sequence and a shorter strand (#3) that binds specifically to the long strand and has the presented sticky ends. An example arrangement for the 0101 sequence is shown following the sequences.

1-redundant sequences for 0101 seed

```

Weave8-2  ctaaagggactacaaatggtatcaccgagagtatctagcctgggtcgg
Weave8-3  actcaccgaaccacccttagtaacc
Weave6-2  gtccatcaggcataacgattacggtgttgcatctccatgtccggtgcg
Weave6-3  tctgacgcaacgggtgatggacaccaa
Weave4-2  caatgcggtactacagagattcgcaccgagcagacgacacctgagacgg
Weave4-3  acaagccgtctcaccgcattgaatcg
Weave2-2  gtgagacaggatctggtaagttggtgtaacgtcggcttgt caaggacc
Weave2-3  taagcggtccttg tgtctcac

```

1-redundant sequences for 1010 seed

```

Weave8-2  cgaagctgactacaaatggtatcaccgagagtatctagccttgaggc
Weave8-3  gagatgcctccaacagcttcgtaacc
Weave6-2  gcgatacaggcataacgattacggtgttgcatctccatgtccttagac
Weave6-3  ccattgtctaaggtgtatcgacaccaa
Weave4-2  cctgtgagactacagagattcgcaccgagcagacgacacctggcaacg
Weave4-3  tccttcggttgccactcacaggaatcg
Weave2-2  gctctacaggatctggtaagttggtgtaacgtcggcttgtccggtcgc
Weave2-3  gacttgccaacgggtgtagagc

```

2-redundant sequences for 0011 seed

Weave8-2 ctaaagggactacaaatgttatcaccgagagtatctagcctgggttcgg
Weave8-3 actcaccgaaccacccttttagtaacc
Weave6-2 gcgatacaggcataaacgattacgggtggtgcatttccatgtccttagac
Weave6-3 ccattgtctaaggtgtatcgaccaa
Weave4-2 cctgtgagactacagagattcgcaccgagcagacgacacctggcaacg
Weave4-3 tccttcggttgccactcacaggcttac
Weave2-2 gtgagacaggatctggtaagttggtgtaacgtcggcttgtcaaggacc
Weave2-3 taagcggtccttgtgtctcac

2-redundant sequences for 1100 seed

Weave8-2 cgaagctgactacaaatgttatcaccgagagtatctagccttggaggc
Weave8-3 gagatgcctccaacagcttcgagcat
Weave6-2 gtccatcaggcataaacgattacgggtggtgcatttccatgtccgttgcg
Weave6-3 tctgacgcaacgggtgatggaccaa
Weave4-2 caatgctggactacagagattcgcaccgagcagacgacacctgagacgg
Weave4-3 acaagccgtctcaccgcattgaatcg
Weave2-2 gctctacaggatctggtaagttggtgtaacgtcggcttgtccgttcgc
Weave2-3 gacttgcaacgggtgtagagc

3-redundant sequences for 0001 seed

Weave8-2 ctaaagggactacaaatgttatcaccgagagtatctagcctgggttcgg
Weave8-3 actcaccgaaccacccttttagtaacc
Weave6-2 gcgatacaggcataaacgattacgggtggtgcatttccatgtccttagac
Weave6-3 ccattgtctaaggtgtatcgaccaa
Weave4-2 caatgctggactacagagattcgcaccgagcagacgacacctgagacgg
Weave4-3 acaagccgtctcaccgcattgaatcg
Weave2-2 gtgagacaggatctggtaagttggtgtaacgtcggcttgtcaaggacc
Weave2-3 taagcggtccttgtgtctcac

3-redundant sequences for 1110 seed

Weave8-2 cgaagctgactacaaatgttatcaccgagagtatctagccttggaggc
Weave8-3 gagatgcctccaacagcttcgagcat
Weave6-2 gtccatcaggcataaacgattacgggtggtgcatttccatgtccgttgcg
Weave6-3 tctgacgcaacgggtgatggacgatag
Weave4-2 cctgtgagactacagagattcgcaccgagcagacgacacctggcaacg
Weave4-3 tccttcggttgccactcacaggaatcg
Weave2-2 gctctacaggatctggtaagttggtgtaacgtcggcttgtccgttcgc
Weave2-3 gacttgcaacgggtgtagagc

4-redundant sequences for 0000 seed

Weave8-2 ctaaagggactacaaatgttatcaccgagagtatctagcctgggttcgg
Weave8-3 actcaccgaaccacccttttagtaacc
Weave6-2 gcgatacaggcataaacgattacgggtggtgcatttccatgtccttagac
Weave6-3 ccattgtctaaggtgtatcgaccaa
Weave4-2 caatgctggactacagagattcgcaccgagcagacgacacctgagacgg
Weave4-3 acaagccgtctcaccgcattgaatcg
Weave2-2 gctctacaggatctggtaagttggtgtaacgtcggcttgtccgttcgc
Weave2-3 gacttgcaacgggtgtagagc

4-redundant sequences for 1111 seed

Weave8-2 cgaagctgactacaaatgttatcaccgagagtatctagccttggaggc
Weave8-3 gagatgcctccaacagcttcgagcat
Weave6-2 gtccatcaggcataaacgattacggtggtgcatttccatgtccggtgcg
Weave6-3 tctgacgcaacggtgatggacgatag
Weave4-2 cctgtgagactacagagattcgcaccgagcagacgacacctggcaacg
Weave4-3 tccttcgttgccactcacaggcttac
Weave2-2 gtgagacaggatctggtaagttggtgtaacgctcggttgtcaaggacc
Weave2-3 taagc ggtccttg tgtctcac

Sequences for programming a particular sequence can in principle be mixed and matched to nucleate any desired sequence.

Arrangement of sequences within the crystal seed structure for sequence 0101

```

3 >cgagcac-tctgtacc\agactgct-cttgcaac< 1
2 <gacctgtg agacatgg\tctgacga gaacgttg>
| | Weave-12 |
2 >ggctaact ccattgtg\tctcaacc tlatgacc-gagaa-cgtatcag-aaatcctt\tcaacctt-gctcttgt> 2
3 <ccgattga-ggtaaacac\aaqdatgg-aatactgg-ctctt-gcattgctc ttaggaa\agttggaa cgaagaac< 1
| | | | Weave-11 | |
| | <aacgt-gcaactca gcatttgt\gaaagatg tggagac-----ttggt>1 |>catctgtg-tgaccata\tctatcct-ccgatgac<2 |>gacag-----csgtgcca ggtagcga\tcacctcc tgcctctg-ttctct> 4
| | | | Weave-9 | | | |
| | <ctaacgca gctctctg\tgacgtgg tgcctagg-----acctt<1 |>gagaatca-ggctctca\atcaacc-nmmnmn> 2
| | >gatgggct-cggataag\actgcaacc-eggatcg-----tgaa-----ctcttagt |csgagagt\tagatcgg |mnmnmnm-nmmn|< 3
| | | | Weave-7 | | | |
| | /tt-gg-ttctct> 1 >gagggagt-ccatata\tacttagg-acgactgg<2 |>ctctgg-----gctgaagg actatctg\tttgtagt |mnmnmnm-nmmn|> 3
| | \tt-cc-agga-----ctcgctca ggtatagc\atgaaatcc tctgacc-----agaa-----cgacttcc-tgataca\aaacatca-nmmnmnm|< 2
| | | | Weave-5 | | | |
| | /tt-gg-ttctct> 1 >gagggagt-ccatata\tacttagg-acgactgg<2 |>ctctgg-----gctgaagg actatctg\tttgtagt |mnmnmnm-nmmn|> 3
| | \tt-cc-agga-----ctcgctca ggtatagc\atgaaatcc tctgacc-----agaa-----cgacttcc-tgataca\aaacatca-nmmnmnm|< 2
| | | | Weave-3 | | | |
| | /tt-gg-ttctct> 1 >gatgagg-actgata\tccgctgt-cccaaat<2 |>actca-----gctctaca ggcattag\gattgoc |mnmnmnm-nmmn|> 3
| | \tt-cc-attgg-----cttactcc tgactcat\aggcgaca gggtttag-----tgat-----cgagatgt-cogtaac\caatacgg-nmmnmnm|< 2
| | | | Weave-1 | | | |
| | /tt-gg-ttctct> 1 >cgagtggt-acgaaagc\agtgccgt-ccgatgc<2 |>ttctc-----gagtagtg acgcttag\tctgtag |mnmnmnm-nmmn|> 3
| | \tt-cc-aagga-----gctcacc tgccttcg\tcaacgca ggcatacg-----aagag-----ctcctacc-tgcgaatc\agacatca-nmmnmnm|< 2
| | | | Weave-3 | | | |
| | /tt-gg-aatcg> 1 >gatgatgt-cttgtaa\tgaaaggc-acaacag<2 |>cgatt-----cgcaaca ggttgaat\tccagatcc |mnmnmnm|> 3
| | \tt-cc-ttagc-----ctactca ggaacatc\acttcgct-ctctgctc-----gctaa-----gggttgt-ccaactta\ggcttagg-nmmnmnm|< 2
| | | | Weave-1 | | | |
| | /tt-gaacgacc tgatttgg\taataca ggcattcg-----cactac
| | \tt-cttgttg-actaacg\attagat-ccgtaacg>

```

References

- [1] Rebecca Schulman and Erik Winfree. Synthesis of Crystals with a Programmable Kinetic Barrier to Nucleation. *P. Natl. Acad. Sci.*, 104(39):15236–15241, 2007.
- [2] Paul W. K. Rothemund, Nick Papadakis, and Erik Winfree. Algorithmic Self-Assembly of DNA Sierpinski Triangles. *PLOS Biology*, 2:424–436, 2004.
- [3] Nadrian C. Seeman. *De Novo* Design of Sequences for Nucleic Acid Structural Engineering. *J. Biomol. Struct. Dyn.*, 8(3):573–581, 1990.
- [4] Robert M. Dirks, Milo Lin, Erik Winfree, and Niles A. Pierce. Paradigms for Computational Nucleic Acid Design. *Nucleic Acids Res.*, 32:1392, 2004.
- [5] John SantaLucia, Jr., Hatim T. Allawi, and P. Ananda Seneviratne. Improved Nearest-Neighbor Parameters for Predicting DNA Duplex Stability. *Biochemistry*, 35(11):3555–3562, 1996.
- [6] Ho-Lin Chen, Rebecca Schulman, Ashish Goel, and Erik Winfree. Reducing Facet Nucleation During Algorithmic Self-Assembly. *Nano Lett.*, 7(9):2912–2919, 2007.