Supplementary Information

Custom-Size, Functional, and Durable DNA Origami with Design-Specific Scaffolds

Floris A. S. Engelhardt [1§], *Florian Praetorius* [1§], *Christian H. Wachauf* [1], *Gereon Brüggenthies* [1], *Fabian Kohler* [1], *Benjamin Kick* [1], *Karoline L. Kadletz* [1], *Phuong Nhi Pham* [1], *Karl L. Behler* [2], *Thomas Gerling* [1], *and Hendrik Dietz* [1*]

Affiliations

1 Technical University of Munich, Physics Department, Am Coulombwall 4a, 85748 Garching bei München, Germany.

2 Technical University of Munich, Lehrstuhl für Bioverfahrenstechnik, Boltzmannstr. 15, 85748 Garching bei München, Germany.

§ These authors contributed equally to this work.

* Correspondence to dietz@tum.de

Supplementary Note S1: Generating a design-specific scaffold sequence with the scaffold smith.

Supplementary Note S2: Overview of methods for phage-based production of ssDNA scaffolds

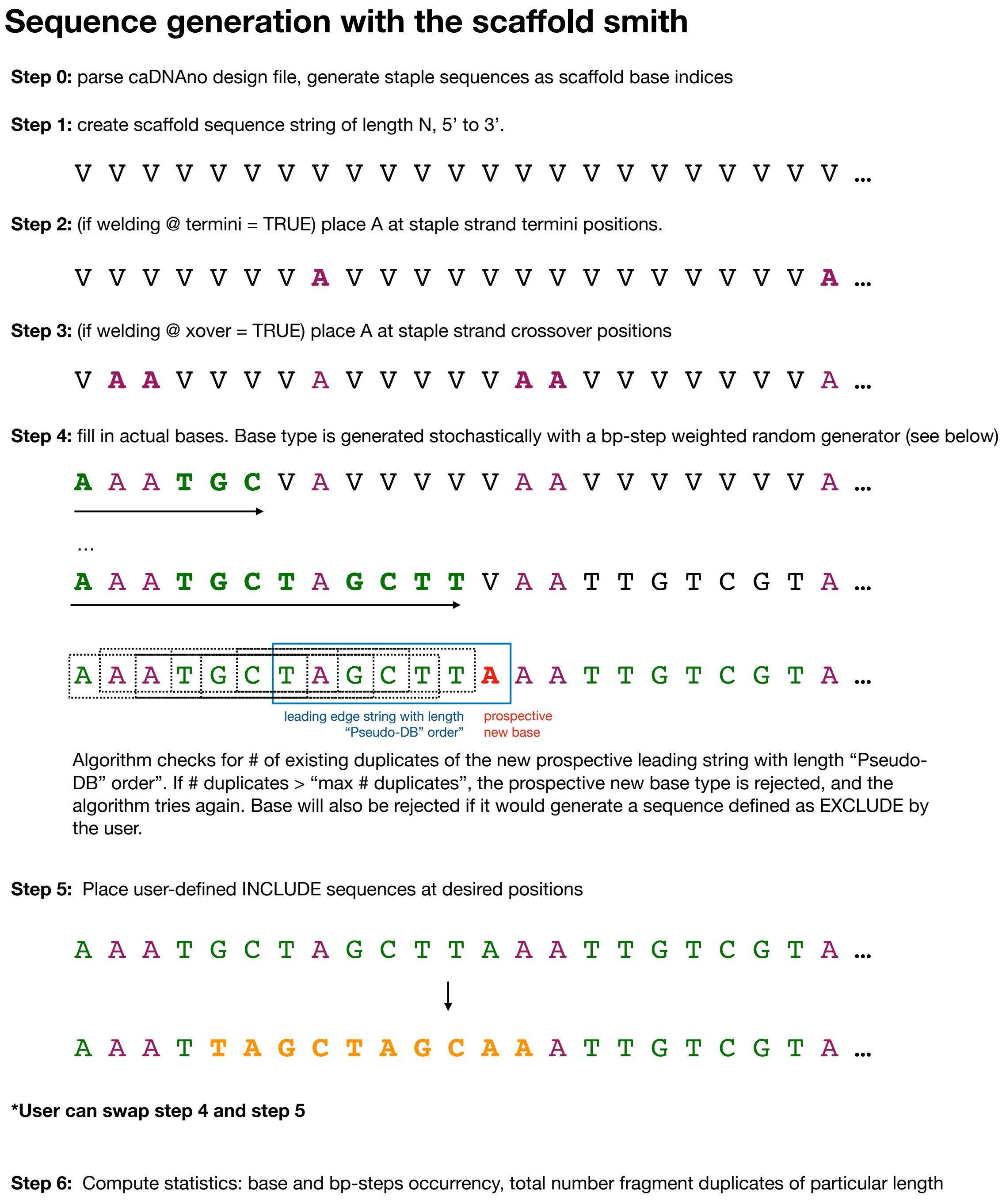
Supplementary Note S3: Workflow for the design, cloning, and production of a new custom scaffold

Supplementary Figures S1-S17

Supplementary Table ST1: Sequences.xlsx

Supplementary Note S1:

Generating a design-specific scaffold sequence with the scaffold smith.



Sequence generation with the scaffold smith

Step 0: parse caDNAno design file, generate staple sequences as scaffold base indices

Step 1: create scaffold sequence string of length N, 5’ to 3’.

V V V V V V V V V V V V V V V V V V V V V V V ...

Step 2: (if welding @ termini = TRUE) place A at staple strand termini positions.

V V V V V V V **A** V V V V V V V V V V V V V V **A** ...

Step 3: (if welding @ xover = TRUE) place A at staple strand crossover positions

V **A** **A** V V V V A V V V V V **A** **A** V V V V V V V A ...

Step 4: fill in actual bases. Base type is generated stochastically with a bp-step weighted random generator (see below)

A A A **T** **G** **C** V A V V V V V A A V V V V V V V A ...

...

A A A **T** **G** **C** **T** A **G** **C** **T** **T** V A A T T G T C G T A ...

A A A T G C T A G C T T **A** A A T T G T C G T A ...

Algorithm checks for # of existing duplicates of the new prospective leading string with length “Pseudo-DB” order”. If # duplicates > “max # duplicates”, the prospective new base type is rejected, and the algorithm tries again. Base will also be rejected if it would generate a sequence defined as EXCLUDE by the user.

Step 5: Place user-defined INCLUDE sequences at desired positions

A A A T G C T A G C T T A A A T T G T C G T A ...

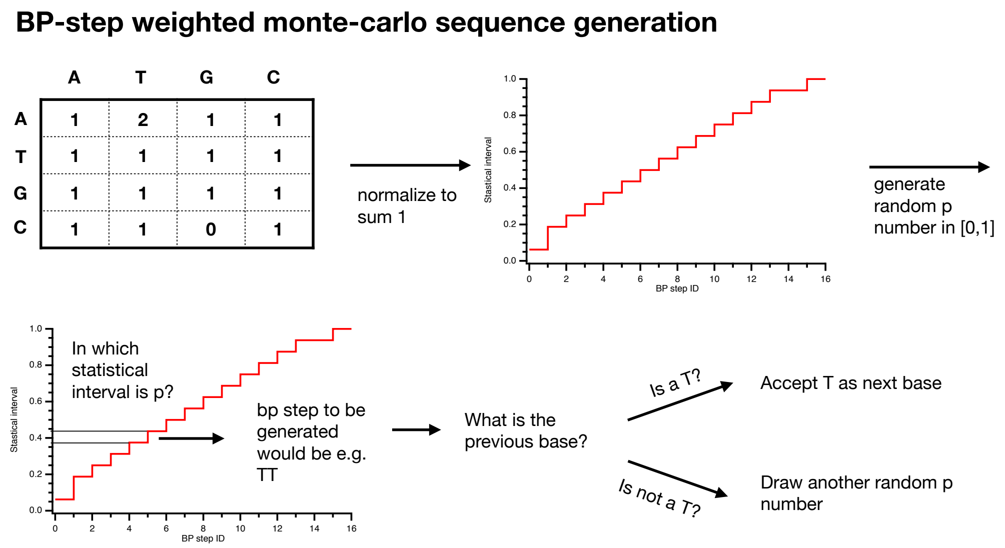
↓

A A A T **T** **A** **G** **C** **T** **A** **G** **C** **A** **A** A T T G T C G T A ...

***User can swap step 4 and step 5**

Step 6: Compute statistics: base and bp-steps occurrency, total number fragment duplicates of particular length

Figure S1: Overview of the main steps in the sequence construction workflow in the scaffold smith.



BP-step weighted monte-carlo sequence generation

	A	T	G	C
A	1	2	1	1
T	1	1	1	1
G	1	1	1	1
C	1	1	0	1

normalize to sum 1
Stastical interval
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14
16
BP step ID
generate random p number in [0,1]
In which statistical interval is p?
bp step to be generated would be e.g. TT
What is the previous base?
Is a T?
Accept T as next base
Is not a T?
Draw another random p number

Figure S2: Illustration of the stochastic sequence generation method used by the scaffold smith.

Instructions for using the scaffold smith graphical user interface.

The application was compiled using LabView 2017 (National Instruments) and runs as a standalone program on either Mac OSX or Windows machines after installing the LabView2017 runtime engine. The runtime engine is available freely on the National Instruments website.

After having installed the runtime, open the app. It enters automatically execution mode. Depending on the screen resolution of the user, some display items may be scrambled or overlaid a bit. In this case, simply resize the window and make use of the scroll bars. The GUI should be relatively straightforward to use, nonetheless below is a screenshot in which the key features are explained.

Follow these steps to generate a design specific scaffold sequence:

(1) Prepare a cadnano design (*.json) file and a plain (!) txt file with sequence constraints. Make backup copies of both!

(2) Write constraints in a plain text file as follows

```
INCLUDE
helixID,baseID|sequence
...
EXCLUDE
sequenceA
sequenceB
```

Example:

```
INCLUDE
0,16|ATGC
EXCLUDE
GCTTTAGC
```

Note: Sequences always refer to sequences to appear in the scaffold and are written 5' to 3'.

(3) Do NOT define constraints directly at skip or loop positions.

(4) Hierarchy: user constraints specified in the file will override "fix staple ends" and "welding".

(5) Use the option to include fixed scaffold motifs to conveniently modify all staple ends (e.g. with DNAzyme overhangs as in the biotechnological mass production method described in Praetorius et al, Nature 2017). When using the fix-staple-ends feature, do not place scaffold crossovers within length of fixed termini sequence next to the staple termini.

(6) Consider including the split-ori 234 constant backbone bases motif as a constraint.

(7) Consider excluding the restriction enzyme sites needed for gene assembly (i.e. CGTCTC and GAGACG).

(8) For multi-chain designs, the scaffold start point defines which scaffold loop sequence will be constructed here.

(9) Define the scaffold starting point in design, choose construction options, define base step weights. Then press LOAD & BUILD.

(10) Adjust weights and “max # duplicates” to achieve completion of the sequence build. Fixed staple ends and constraints can cause many duplicates, which can cause the construction to stall.

(11) The program outputs a .txt file named "[json file name]_custom_scaffold.txt". Insert the custom sequence thus produced in caDNAno, export a .svg file and do a thorough inspection whether the scaffold sequence is as desired.

Advanced: the GUI can also be used to generate new variants of a previously obtained scaffold sequence. To this end, paste the old sequence into the field “TEMPLATE TO BE PLACED AT START LOC”. If the field is not empty, the program will insert this sequence starting at scaffold start location. Additional options such as welding or user constraints will be (over)written at the locations specified in the constraint file. If the template string is too short to parse the entire caDNAno design file scaffold loop, the missing bases will be built stochastically as described above.

GUI available under:
https://showcase.dropbox.com/doc/Design-tools-from-Engelhardt-et-al-ACS-Nano-2019--Aa2PU7wwwDRvVqIP~Qyv2qhFAQ-vQgqALYeDzI5GH8MgRvCn

Scaffold smith graphical user interface

Abort execution
(e.g when sequence generation does not converge)

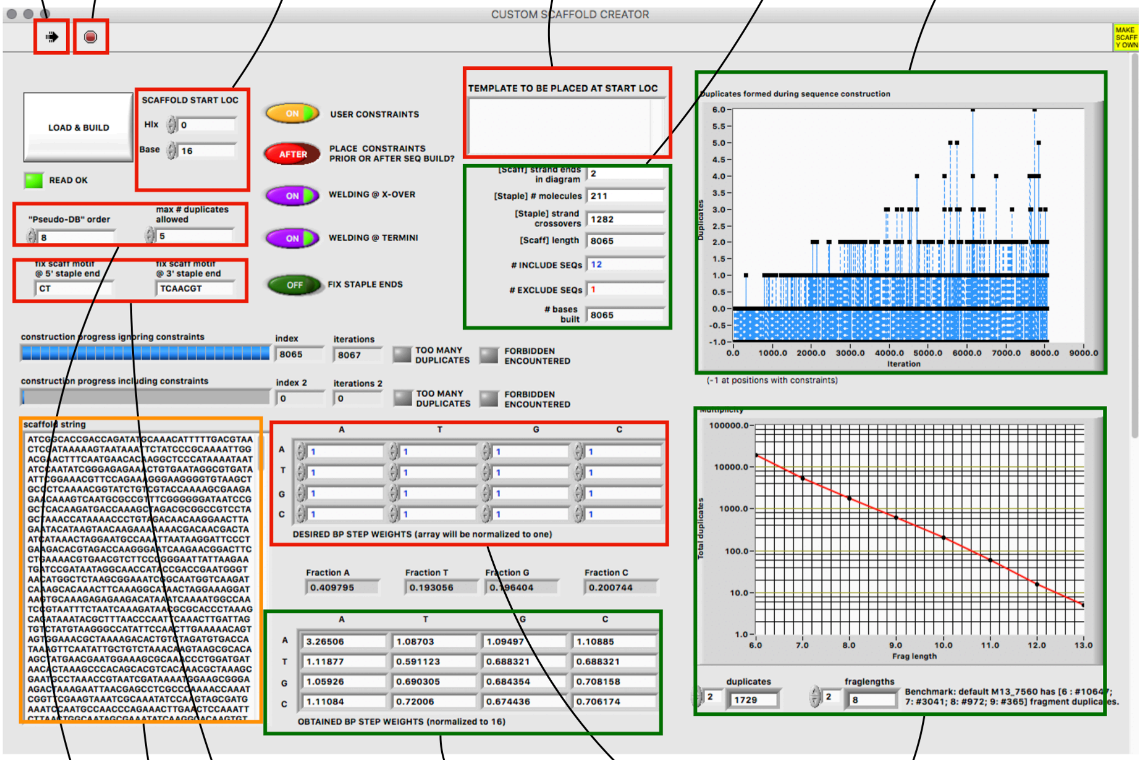
Activate app

Must correspond to scaffold 5'end position in caDNAno diagram

Enter an existing sequence here in case user wishes to create new variants with e.g. UV welding sites or other constraints.

Some design file information. Check whether # EXCLUDE and #INCLUDE corresponds to input.

Duplicates generated during sequence build. Useful to find practical "max # duplicates".



CUSTOM SCAFFOLD CREATOR

SCAFFOLD START LOC
Hlx 0
Base 16

"Pseudo-DB" order 8
max # duplicates allowed 5

fix scaff motif @ 5' staple end CT
fix scaff motif @ 3' staple end TCAACGT

ON USER CONSTRAINTS
AFTER PLACE CONSTRAINTS PRIOR OR AFTER SEQ BUILD?
ON WELDING @ X-OVER
ON WELDING @ TERMINI
OFF FIX STAPLE ENDS

TEMPLATE TO BE PLACED AT START LOC

[Scaff] strand ends in diagram 2
[Staple] # molecules 211
[Staple] strand crossovers 1282
[Scaff] length 8065
INCLUDE SEQs 12
EXCLUDE SEQs 1
# bases built 8065

construction progress ignoring constraints
index 8065
iterations 8067
TOO MANY DUPLICATES
FORBIDDEN ENCOUNTERED

construction progress including constraints
index 2 0
iterations 2 0
TOO MANY DUPLICATES
FORBIDDEN ENCOUNTERED

scaffold string

```
ATCGGCACCGACCAGATATGCAAACATTTTTGACGTAA
CTCGATAAAAAGTAATAAATTCTATCCCGCAAAATTGG
ACGAACTTTCAATGAACACAAGGCTCCCATAAAATAAT
ATCCAATATCGGGAGAGAAACTGTGAATAGGCGTGATA
ATTCGGAAACGTTCCAGAAAGGGAAGGGGTGTAAGCT
GCCCTCAAAACGGTATCTGTCGTACCAAAAGCGAAGA
GAACAAAGTCAATGCGCCGTTTCGGGGGGATAATCCG
GCTCACAAGATGACCAAAGCTAGACGCGGCCGTCCTA
GCTAAACCATAAAACCCTGTAGACAACAAGGAACTTA
GAATACATAAGTAACAAGAAAAAAACGACAACGACTA
ATCATAAACTAGGAATGCCAAATTAATAAGGATTCCCT
GAAGACACGTAGACCAAGGGAATCAAGAACGGACTTC
CTGAAAACGTGAACGTCTTCCCGGGAATTATTAAGAA
TGATCCGATAATAGGCAACCATACCGACCGAATGGGT
AACATGGCTCTAAGCGGAAATCGGCAATGGTCAAGAT
CAAAGCACAAACTTCAAAGGCATAACTAGGAAAGGAT
AAGTGCAAAGAGAGAAGACATAAATCAAAATGGCCAA
TCCGTAATTTCTAATCAAAGATAACGCGCACCCTAAAG
CAGATAAATACGCTTTAACCCAATTCAAACTTGATTAG
TGTCTATGTAAGGGCCATATTCCAACTTGAAAAACAGT
AGTGGAAACGCTAAAAGACACTGTCTAGATGTGACCA
TAAAGTTCAATATTGCTGTCTAAACAAGTAAGCGCACA
AGCTATGAACGAATGGAAAGCGCAAACCCTGGATGAT
AACACTAAAGCCCACAGCACGTCACAAACGCTAAAGC
GAATGCCTAAACCGTAATCGATAAAATGGAAGCGGGA
AGACTAAAGAATTAACGAGCCTCGCCCAAAACCAAAT
CGGTTCGAAGTAAATCGCAAATATCCAAGTAGCGATG
AAATCCAATGCCAACCCAGAAACTTGAACTCCAAATT
CTTAACTGGCAATAGCGAAATATCAAGGCACAAGTGT
```

	A	T	G	C
A	1	1	1	1
T	1	1	1	1
G	1	1	1	1
C	1	1	1	1

DESIRED BP STEP WEIGHTS (array will be normalized to one)

Fraction A	Fraction T	Fraction G	Fraction C
0.409795	0.193056	0.196404	0.200744

	A	T	G	C
A	3.26506	1.08703	1.09497	1.10885
T	1.11877	0.591123	0.688321	0.688321
G	1.05926	0.690305	0.684354	0.708158
C	1.11084	0.72006	0.674436	0.706174

OBTAINED BP STEP WEIGHTS (normalized to 16)

(-1 at positions with constraints)

duplicates 2 1729
fraglengths 2 8

Benchmark: default M13_7560 has [6 : #10647; 7: #3041; 8: #972; 9: #365] fragment duplicates.

Use these parameters to reduce sequence redundancy

Will be applied if "FIX STAPLE ENDS" active

Actual bp step occurrency

Coarse sequence redundancy analysis.
Sequences that have similar properties as M13 should be fine.

Define desired statistical bp step weights in this table. Default is uniform distribution.
When using "welding option", one may wish to suppress AA at other sites.

Output scaffold string. Will be saved to disc as [design_file_name]_custom_scaffold.txt

Figure S 3: Key features of the graphical user interface of the scaffold smith. Red items are controls / inputs, green items are read-outs / indicators, the orange item is the output sequence.

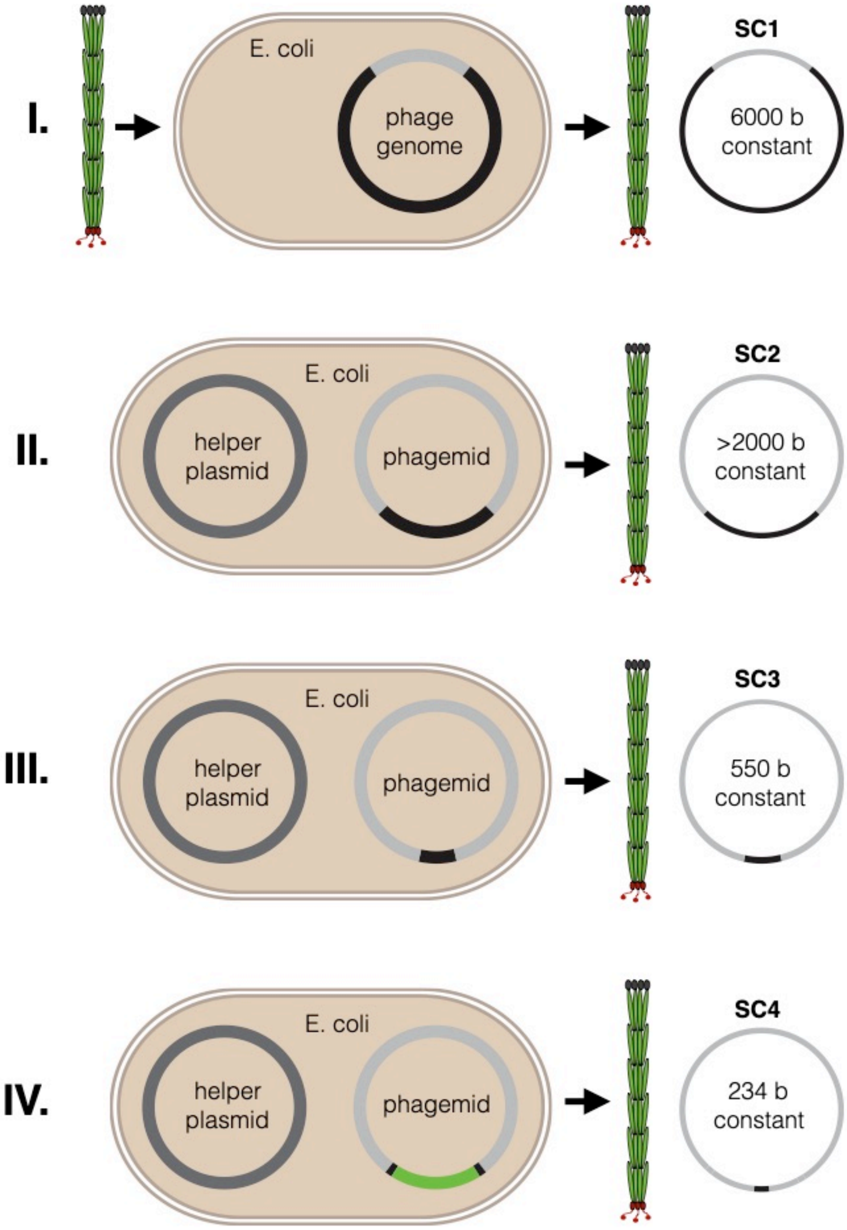
Supplementary Note S2: Overview of methods for phage-based production of ssDNA scaffolds

Method I: conventional M13-based phage production
Production of conventional M13-based scaffolds relies on infection of E. coli cultures with the desired variant of the M13 phage (Figure S4, method I). The ssDNA phage genome is replicated by the host cell and packaged into phage particles that can be easily separated from the cells, enabling efficient isolation of highly pure ssDNA. The M13 genome variants must contain all information needed for the phage replication process, including the phage origin of replication (ori), the phage packaging signal, and the genes encoding for the phage proteins. Altogether, these sequences make up a fixed part of the phage genome of approximately 6k bases. Longer scaffolds may be made by inserting additional sequence stretches into the phage genome.[16, 19]

Method II: Helper plasmid and phagemid
The length of the constant part can be reduced by moving the genes that encode for the phage proteins to a separate helper plasmid,[21, 28] enabling the production of single-stranded phagemid DNA (Figure S4, method II). A typical phagemid backbone contains a phage ori, a phage packaging signal, a plasmid ori, and a selection marker, usually in the form of an antibiotic resistance, resulting in a constant part of >2000 bases. In cells that are co-transformed with helper plasmid and phagemid, the ssDNA version of the phagemid is produced and packaged into phage-like particles from which it can be easily isolated.

Method III: Minimized phagemid backbone
Here, we reduced the length of the fixed portion of the phagemid backbone by removing the plasmid ori from the phagemid and replaced the region conferring antibiotic resistance with a shorter, RNA-based selection marker.[57] As a result, we were able to shorten the fixed section from to 520 bases. However, due to the missing plasmid ori the reduced-backbone phagemid must now be propagated together with a helper plasmid. The RNA-based selection mechanism was also less tight in our hands than the previous antibiotic selection mechanism. cloning and production of scaffolds based on method III is thus more tedious than for the other methods.

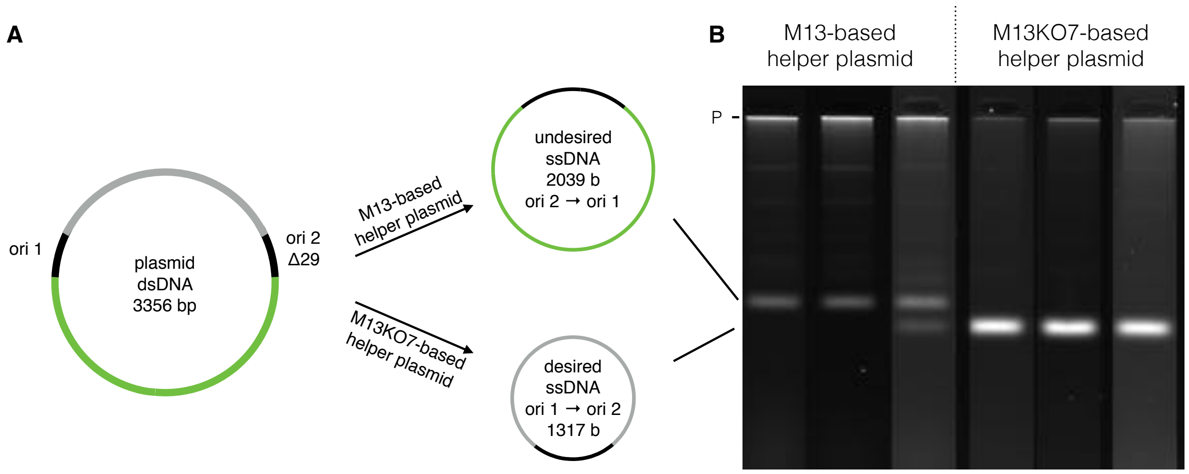
Method IV: Split-ori approach
Phagemid design in method IV relies on a split-ori system originally developed by Specthrie *et al.* to produce “microphages”.[30] In this system, a full copy of the phage ori is placed upstream of the insert sequence, and a second, truncated copy of the phage ori is placed downstream of the insert. We co-transformed E. coli cells with such a phagemid and a suitable helper plasmid to produce circular DNA single strands that comprise one phage ori and the insert sequence, but do not contain the backbone (Supplementary Figure S5). The residual fixed section in ssDNA produced by method IV comprises only 234 bases. Since the backbone is not included in the final product, a conventional backbone with plasmid ori and antibiotic resistance can be used. When using our helper plasmid that was constructed using the genome of the helper phage M13KO7, we were able to produce highly pure ssDNA without detectable side products (Supplementary Figure S5).



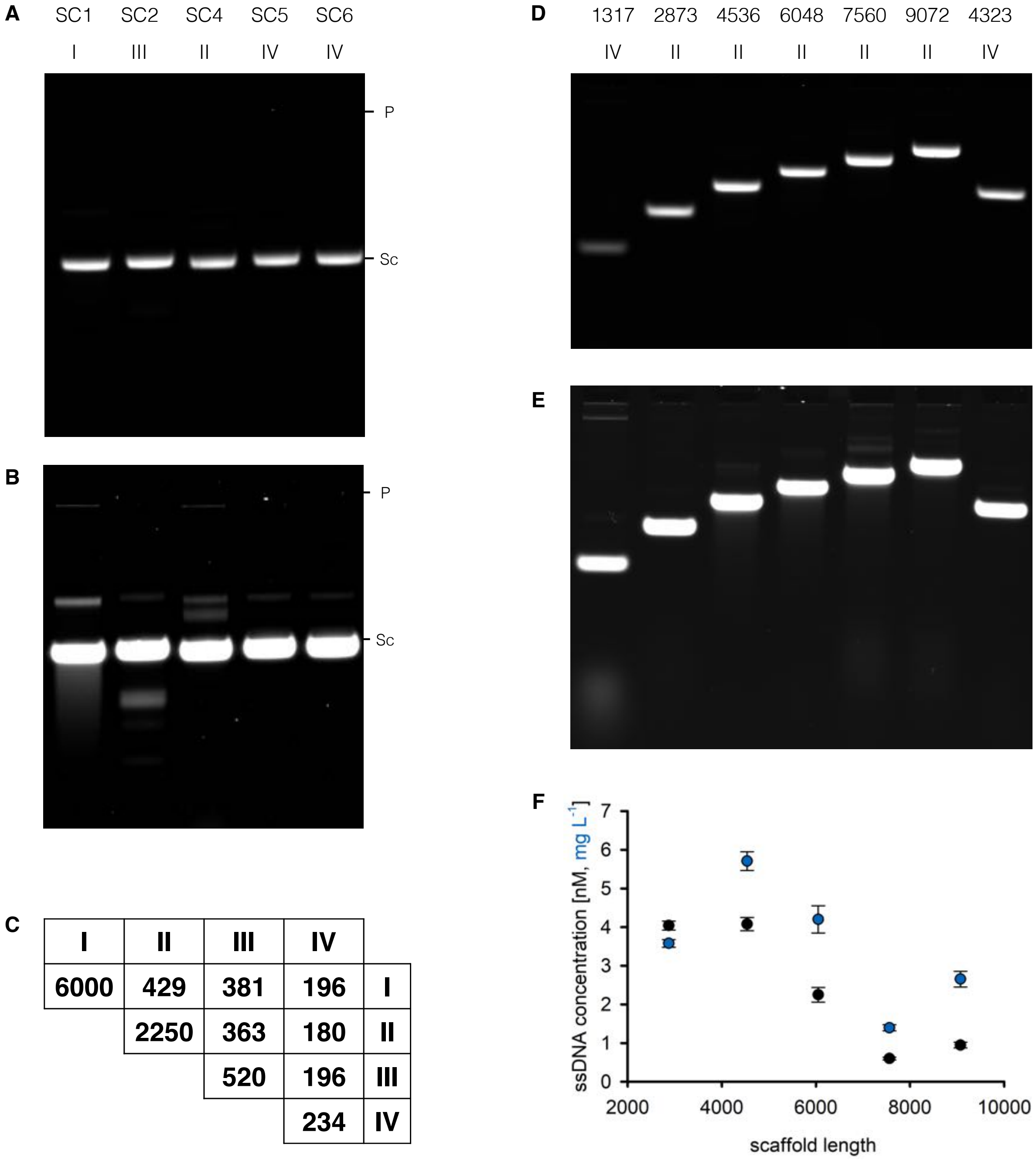
Supplementary Figure S4: Methods for Biotechnological production of ssDNA.
Illustration of scaffold production processes using the conventional M13 phages (I), phagemids with full-length plasmid backbones (II), phagemids with minimized backbones herein developed (III), or a split-ori approach (IV). Black: constant parts for each type of scaffold; grey: user-definable parts; light green: backbone present only in the double-stranded precursor plasmid and not in the single-stranded product.

Supplementary Note S3: Workflow for the design, cloning, and production of a new custom scaffold

- Use the scaffold smith to generate the full scaffold sequence
- Using the gene splitter tool, split the insert sequence into parts that can be synthesized as genes by your favorite gene synthesis provider (e.g. for a 7560-bases-long scaffold, split the insert into 4 fragemnts of less than 2 kb each)
- Order the resulting fragments from any gene synthesis provider, preferably in a plasmid with a kanamycin resistance that does not contain the sequence CGTCTC (or GAGACG) in its backbone
- Set up a golden gate assembly reaction including the fragments and the target plasmid in equimolar ratio as well as EspI, T4 Ligase, and T4 Ligase buffer (1x final).
    - subject the assembly mixture to 15 cycles of (5 minutes 37 °C, 15 minutes 16 °C) followed by a final incubation at 37 °C for 30 min
- transform competent E. coli cells with the assembly mixture, pick individual colonies, purify plasmid DNA and check for correct assembly using suitable restriction enzymes in a control digest. Verify sequence identity of the plasmid DNA using DNA sequencing.
- perform a double transformation of competent DH5alpha E. coli cells using your correctly assembled precursor plasmid and our helper plasmid
- Grow liquid cultures and isolate ssDNA as described in the method section

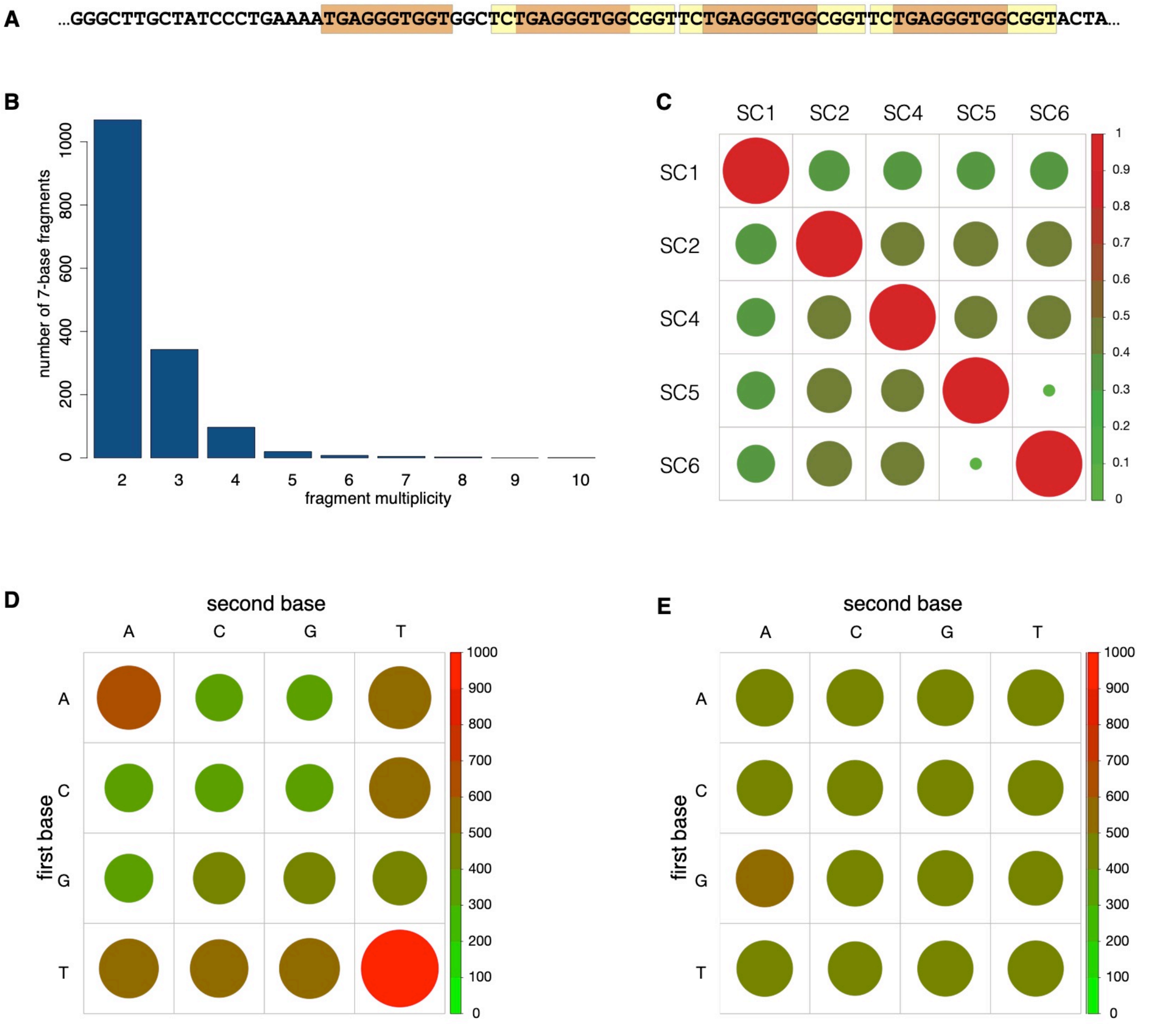


Supplementary Figure S5: Production of ssDNA using the split-ori system (method IV) requires a helper plasmid that is based on the helper phage M13KO7. A Schematic representation of a precursor plasmid comprising a plasmid backbone (green), a full M13 phage ori and a truncated M13 phage ori (both black), and a user-defined insert sequence (grey). Depending on the choice of helper plasmid, two different DNA species can be produced. When using a helper plasmid that is based on the genome of the helper phage M13KO7, the desired ssDNA species is produced, starting at the full-length ori and ending at the truncated ori (bottom). When using a helper plasmid that is based on the genome of the M13 phage variant M13mp18, an undesired species is produced, starting at the truncated ori and ending at the full-length ori. **B** Laser scanned image of a agarose gel on which ssDNA produced in bacteria containing the precursor plasmid shown in A and either a M13-based helper plasmid or a M13KO7-based helper plasmid were separated electrophoretically. Identity of the ssDNA products was confirmed using DNA sequencing. All lanes were autoleveled individually to facilitate comparison.

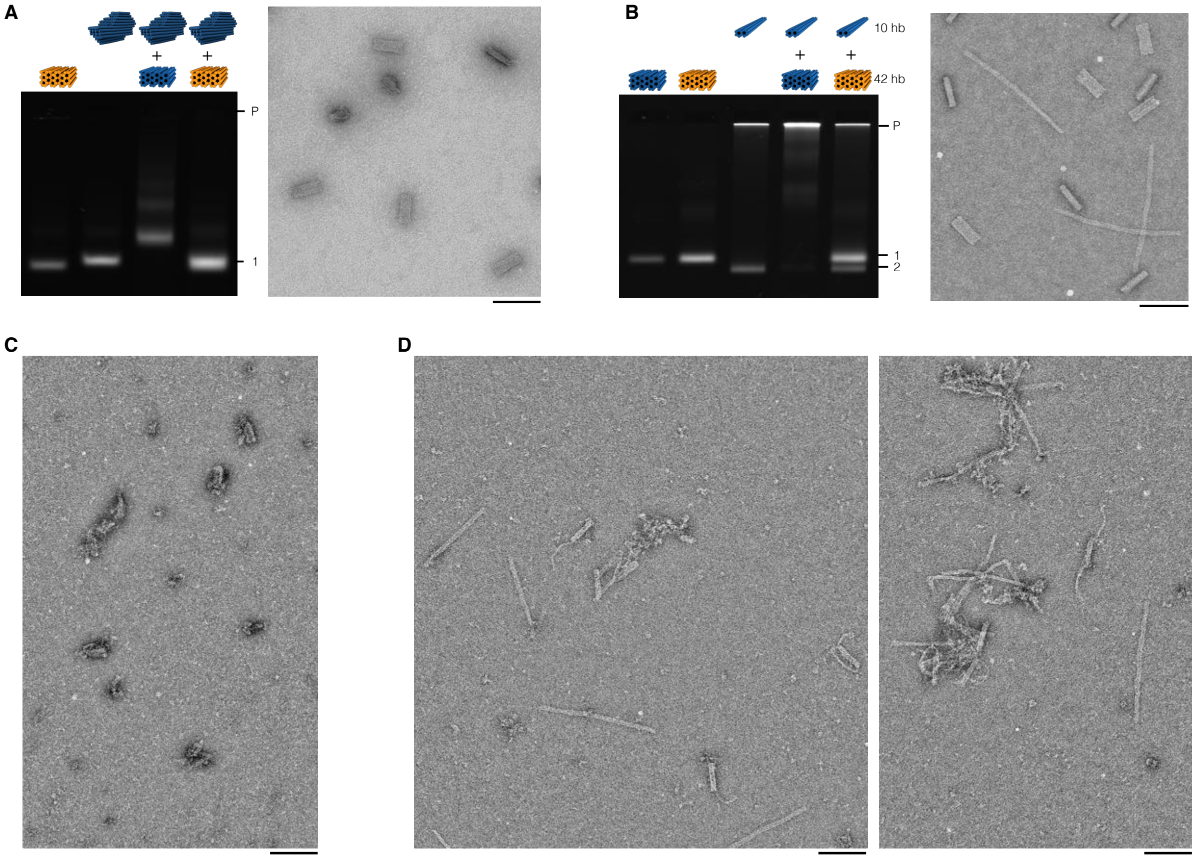


I	II	III	IV	
6000	429	381	196	I
	2250	363	180	II
		520	196	III
			234	IV

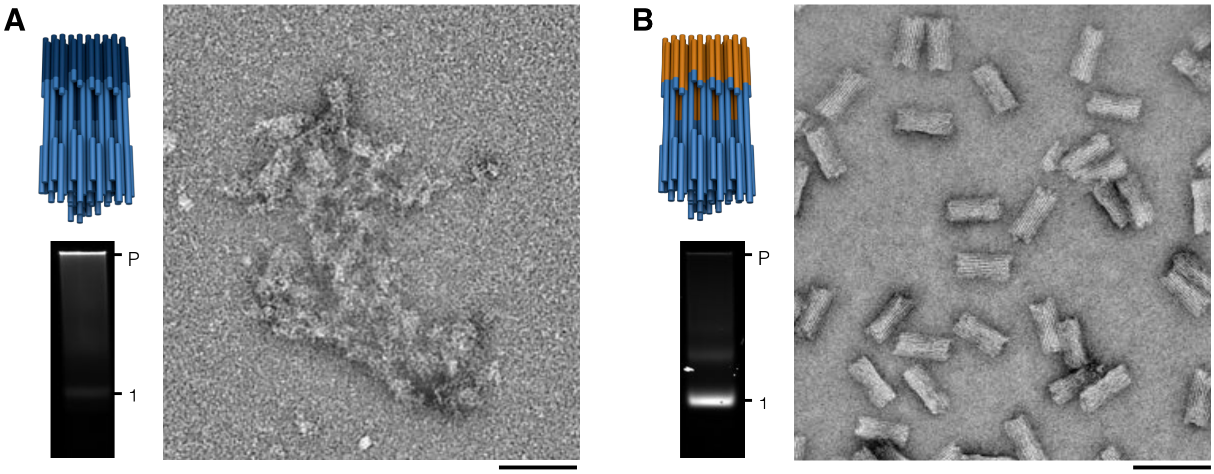
Supplementary Figure S6: Purity and yield for different scaffold production methods. A Gel-electrophoretic purity analysis for the five orthogonal 7560-bases-long scaffolds. SC1-SC6 refers to the scaffold nomenclature used in Figure 2, roman numerals indicate the scaffold production method as described in Supplementary Note S2. The images were brightness auto-leveled. Only the desired product bands are visible. **B** Highly oversaturated image of the gel shown in A to reveal the existence of small amounts of impurities and side products which appear in methods II and III. **C** Pairwise overlap sequences between scaffolds produced with methods I, II, III and IV. Numbers indicate overlap sequence length in bases. **D** Gel-electrophoretic purity analysis for custom scaffolds of different lengths. Numbers above each lane indicate the length of the corresponding scaffold (in bases), roman numerals indicate the scaffold production method. **E** Highly oversaturated image of the gel shown in D. **F** Yield quantification for the production of the five scaffolds from lanes 2-6 in D. blue: yield in ng ssDNA per liter of bacterial culture; black: ssDNA titer in the bacterial culture (in nM). Error bars indicate the standard deviation from three biological replicates. Data shown here is based on 15 individual bacterial cultures that were grown and processed in parallel. The overall yield varied somewhat between productions performed on different days, presumably due to differences in incubation times and experimental deviations.



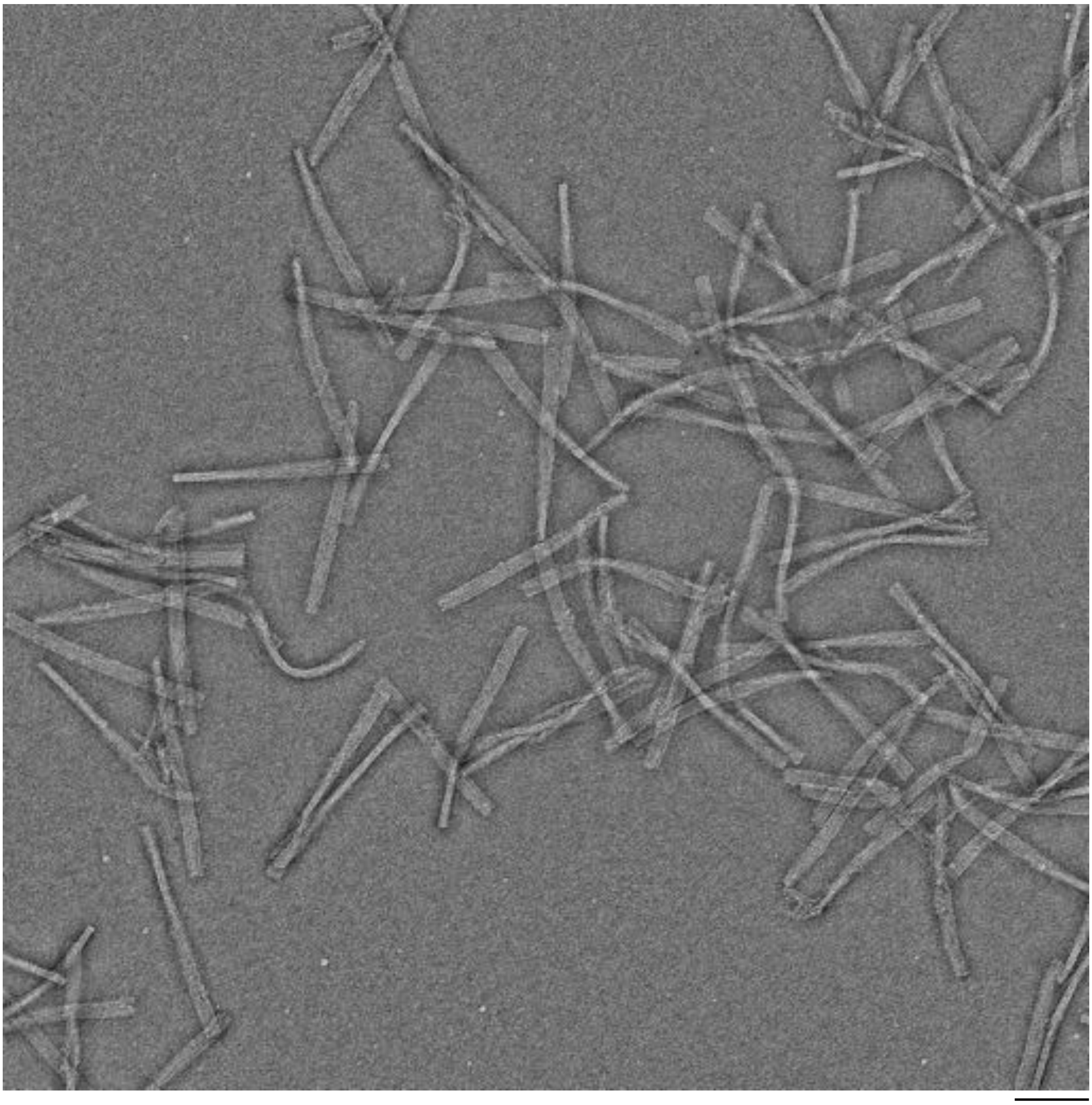
Supplementary Figure S7: Statistical analysis of scaffold sequence redundancy. A Sequence of a 42-base-fragment that appears twice in the M13 phage genome, and thus appears twice in all conventional M13-based scaffold sequences. Red and yellow boxes indicate a 9-base-fragment and a 15-base fragment that appear four times or three times inside this 42-mer, respectively. **B** Histogram of redundant 7-base-sequences in the conventional M13 scaffold with a length of 7560 bases. "Fragment multiplicity" indicates how often a segment appears in the scaffold sequence, "number of 7-base fragments" indicates how many fragments appear with a given multiplicity. **C** Orthogonality plot for the 5 different scaffold sequences used in this work. The scale represents the probability of a 7-base-fragment that appears in one scaffold sequence to also appear in the other scaffold sequence. SC1 is the conventional M13 scaffold, SC2, 4, 5, and 6 are the De Bruijn-based scaffolds with synthetic sequences constructed in this work. The insert sequences for SC5 and SC6 are part of one de bruin sequence of order 7. **D, E** Frequencies of individual base steps in the conventional M13 scaffold SC1 (D) and in the de Bruijn-based custom scaffold SC6 (E). The scale indicates how often a given base is followed by a specific second base. In the de-Bruijn sequences, base steps are much more uniformly distributed.



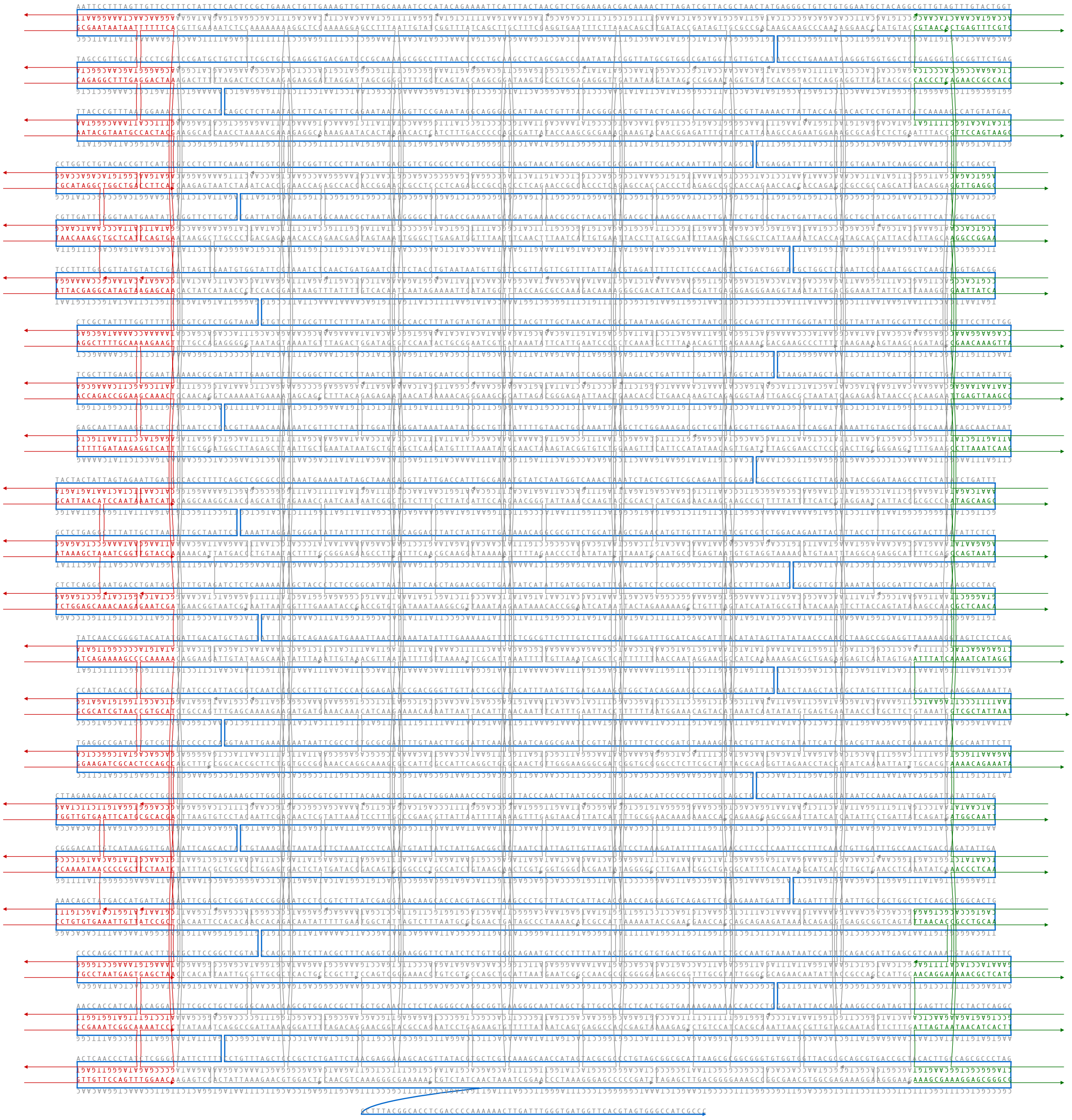
Supplementary Figure S8: One-pot assembly of two different DNA origami structures requires orthogonal scaffold sequences. Blue indicates conventional M13-based scaffolds; orange indicates a custom scaffold with synthetic sequence that is orthogonal to the M13-based scaffolds. **A** left: electrophoretic mobility analysis of single reaction setups (lane 1-2) and one-pot reaction setups (lane 3-4) containing 42-helix bundles and pointer objects. Right: TEM image of the reaction mixture from lane 4. **B** Left: electrophoretic mobility analysis of single reaction setups (lane 1-3) and one-pot reaction setups (lane 4, 5) containing 42-helix bundles (42hb) and 10-helix bundles (10hb). Right: TEM image of the reaction mixture from lane 5. **C** TEM image of the reaction mixture from lane 3 in A containing 42-helix bundles and pointer objects that are both designed using conventional, non-orthogonal M13-based scaffolds. **D** TEM images of the reaction mixture from lane 4 in B containing 42-helix bundles and 10-helix bundles that are both designed using conventional, non-orthogonal M13-based scaffolds. Scale bars: 100 nm



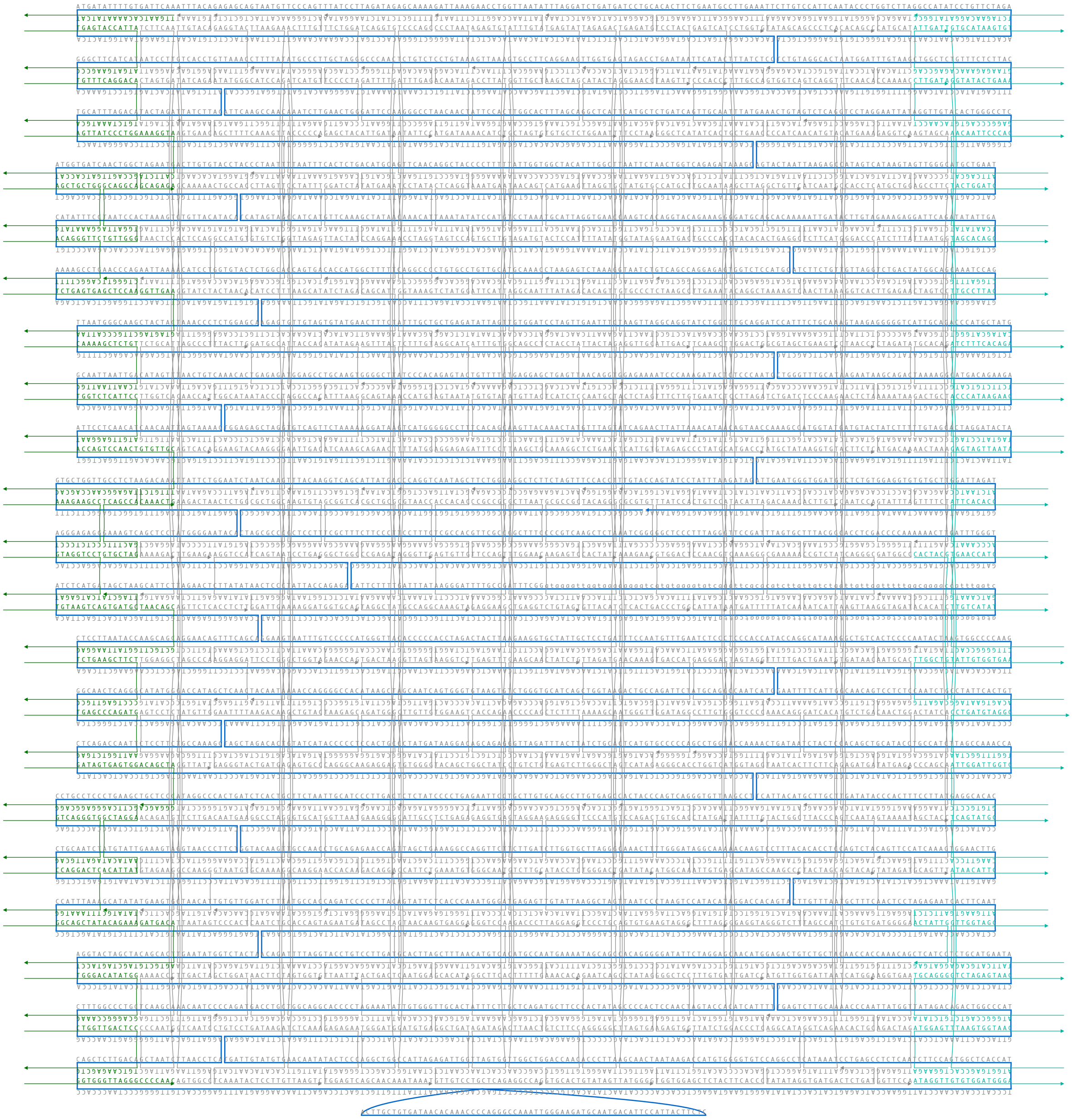
Supplementary Figure S9: Control Experiments for design of dual-scaffold structures. A, B Schematic representations, electrophoretic mobility analysis, and TEM images of 94-helix bundle variants assembled using two M13-based scaffolds (A) and two orthogonal scaffolds (B).



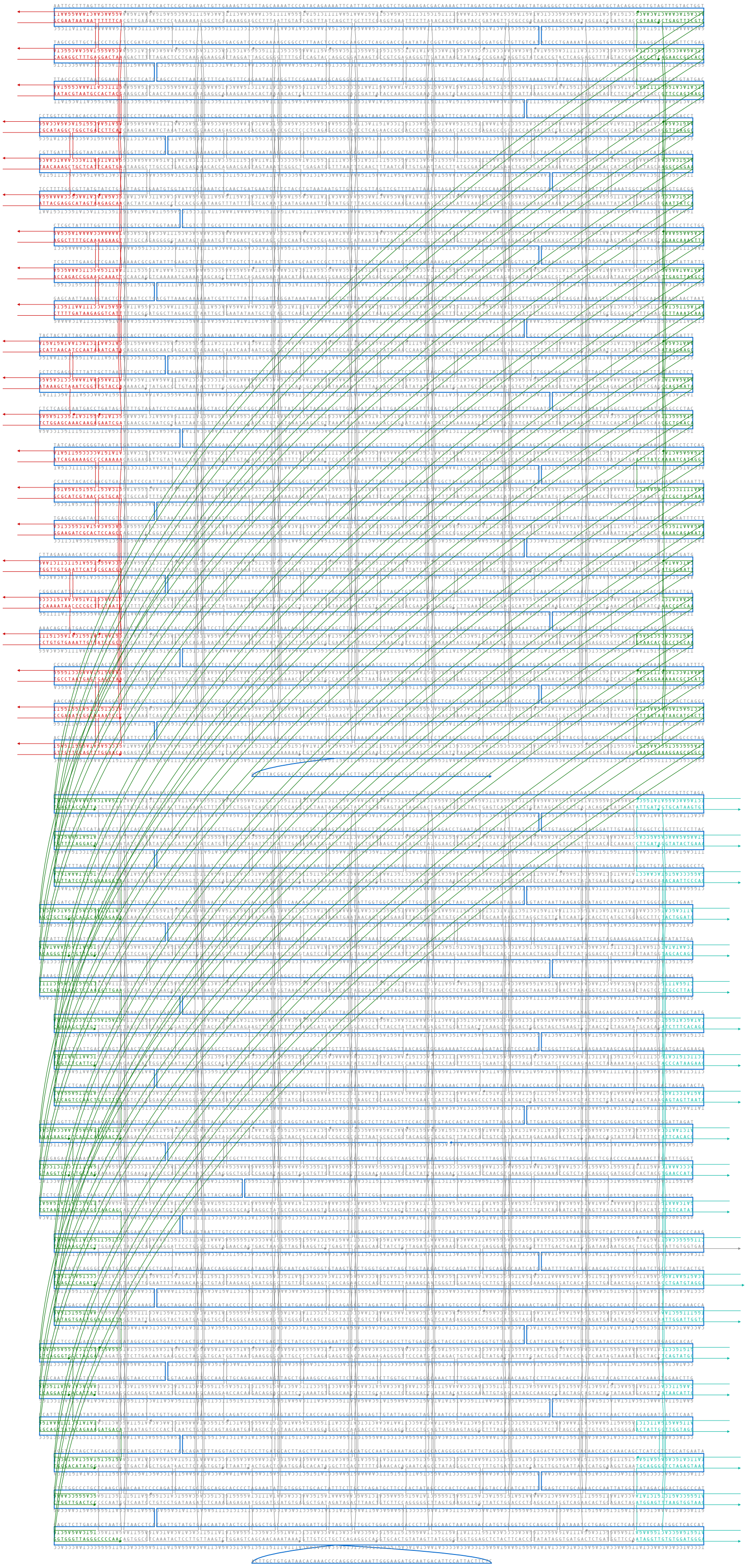
Supplementary Figure S10: Field-of-view TEM image of aggregates formed in 42-helix bundle pentamers. As evident from the image, the objects are well formed, which corroborates the possibility for one-pot assembly with up to five orthogonal scaffold strands. Here, aggregation is presumably caused by strand exchange due to long binding segments in the connecting staple strands. Scale bar: 100 nm.



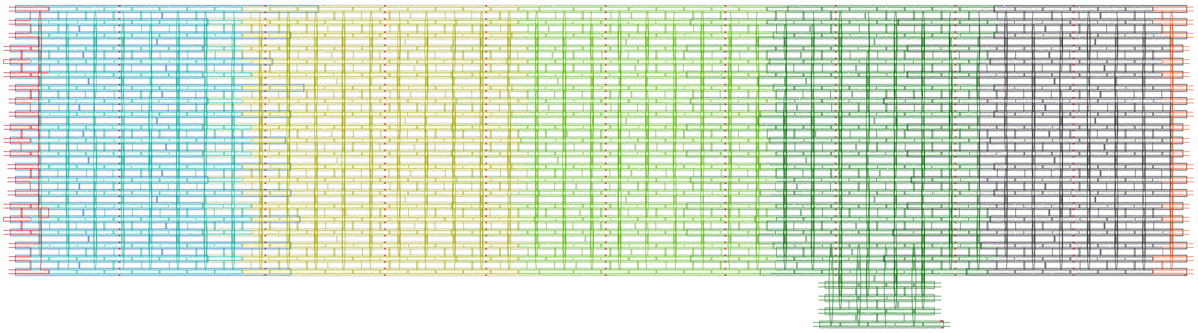
Supplementary Figure S11: caDNAno design diagram of the 42-helix bundle variant designed using the conventional M13-based scaffold. blue: scaffold; grey: core staples; red and green: edge staples carrying 3 thymidines at the termini pointing out of the object. Diagram was prepared with caDNAno v0.1.[22]



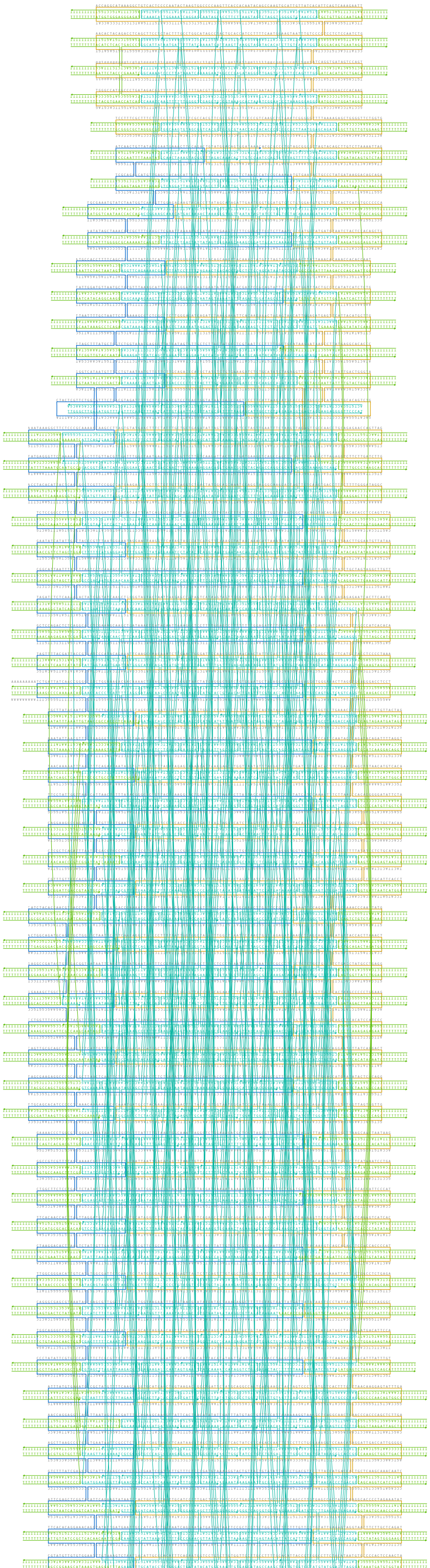
Supplementary Figure S12: caDNAno design diagram of the 42-helix bundle variant designed using our custom scaffold SC2. blue: scaffold; grey: core staples; green and cyan: edge staples carrying 3 Thymidines at the termini pointing out of the object. Strand topologies of the versions designed using our custom scaffolds SC3, SC4, SC5, and SC6 are identical, but edge oligos in these versions carry 10 thymidines instead of 3. Diagram was prepared with caDNAno v0.1.[22]



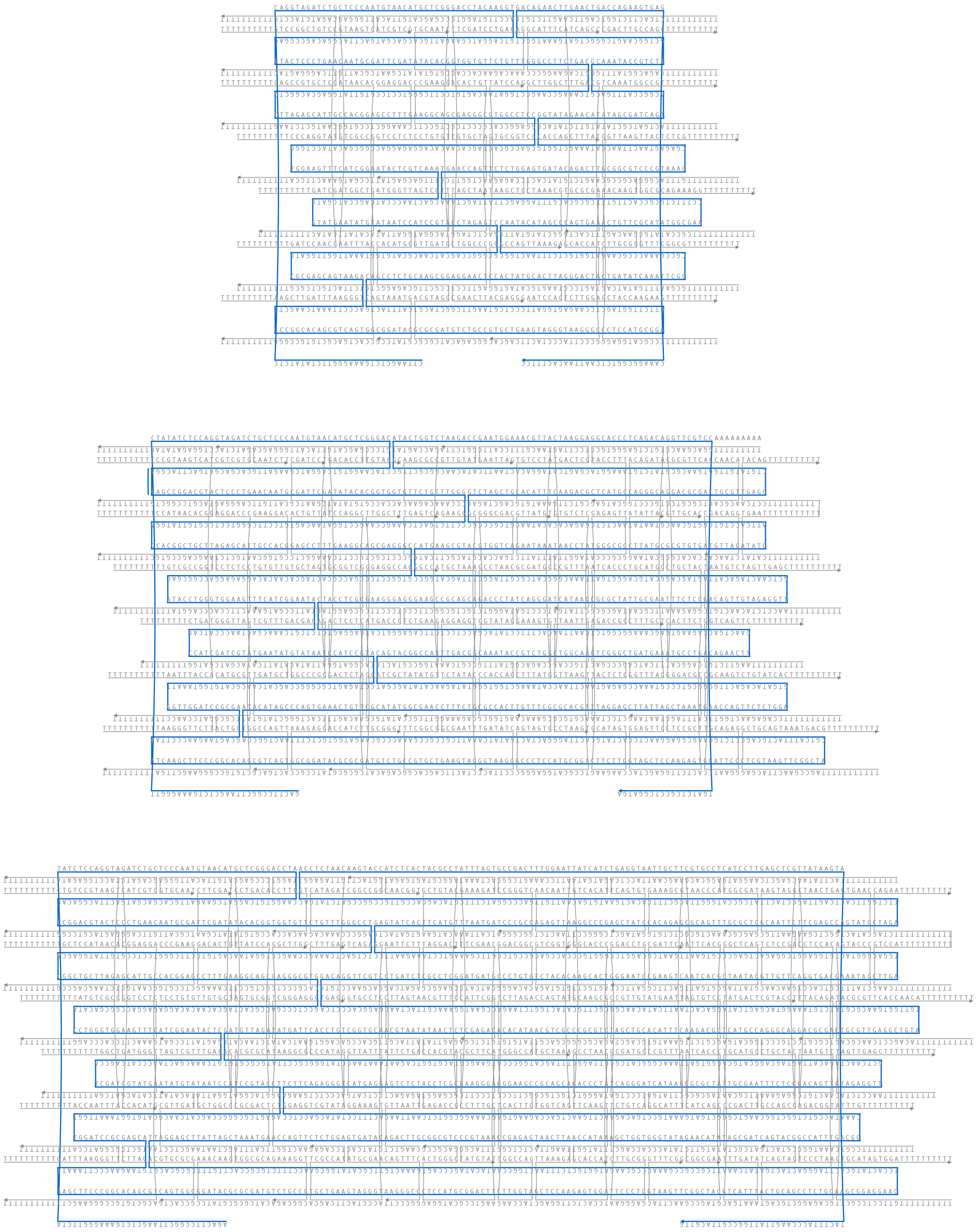
Supplementary Figure S13: caDNAno design diagram of a 42-helix bundle dimer in which the green edge staples of the two monomers shown in Supplementary Figures S12 and S13 are connected. Trimers, tetramers and pentamers are constructed accordingly. Diagram was prepared with caDNAno v0.1.[22]



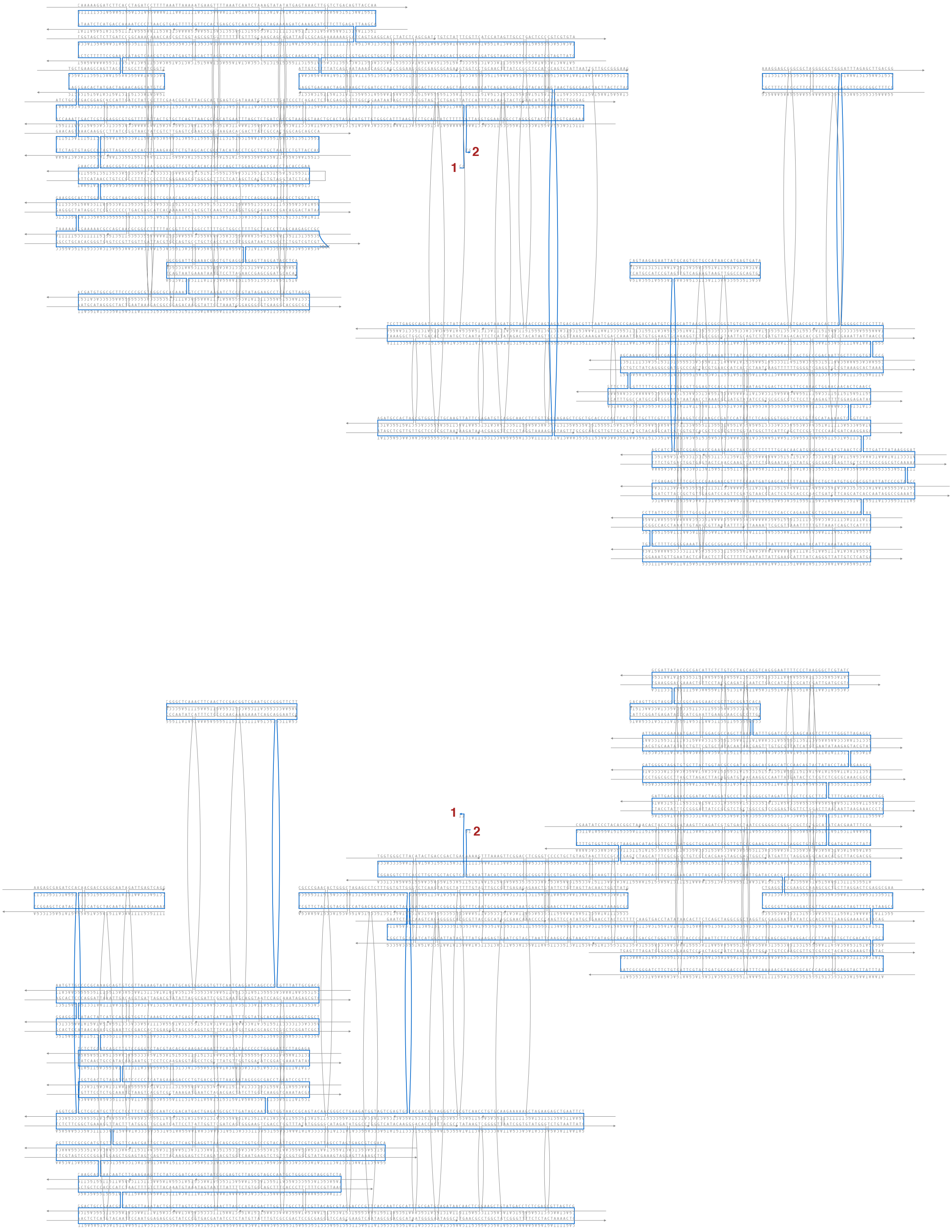
Supplementary Figure S14: caDNAno design diagram of a 42-helix bundle pentamer shown in Figure 3. Red: edge staples carrying 3 thymidines at the termini pointing out of the object; light blue: 1rst fifth of 42hb build by core staples binding and scaffold SC1; yellow: 2nd fifth of 42hb build by core staples binding and scaffold SC2; light green: 3rd fifth of 42hb build by core staples binding and scaffold SC4: dark green: 4th fifth of 42hb build by core staples binding and scaffold SC5; black: 5th fifth of 42hb build by core staples binding and scaffold SC6. Diagram was prepared with caDNAno v0.1.[22]



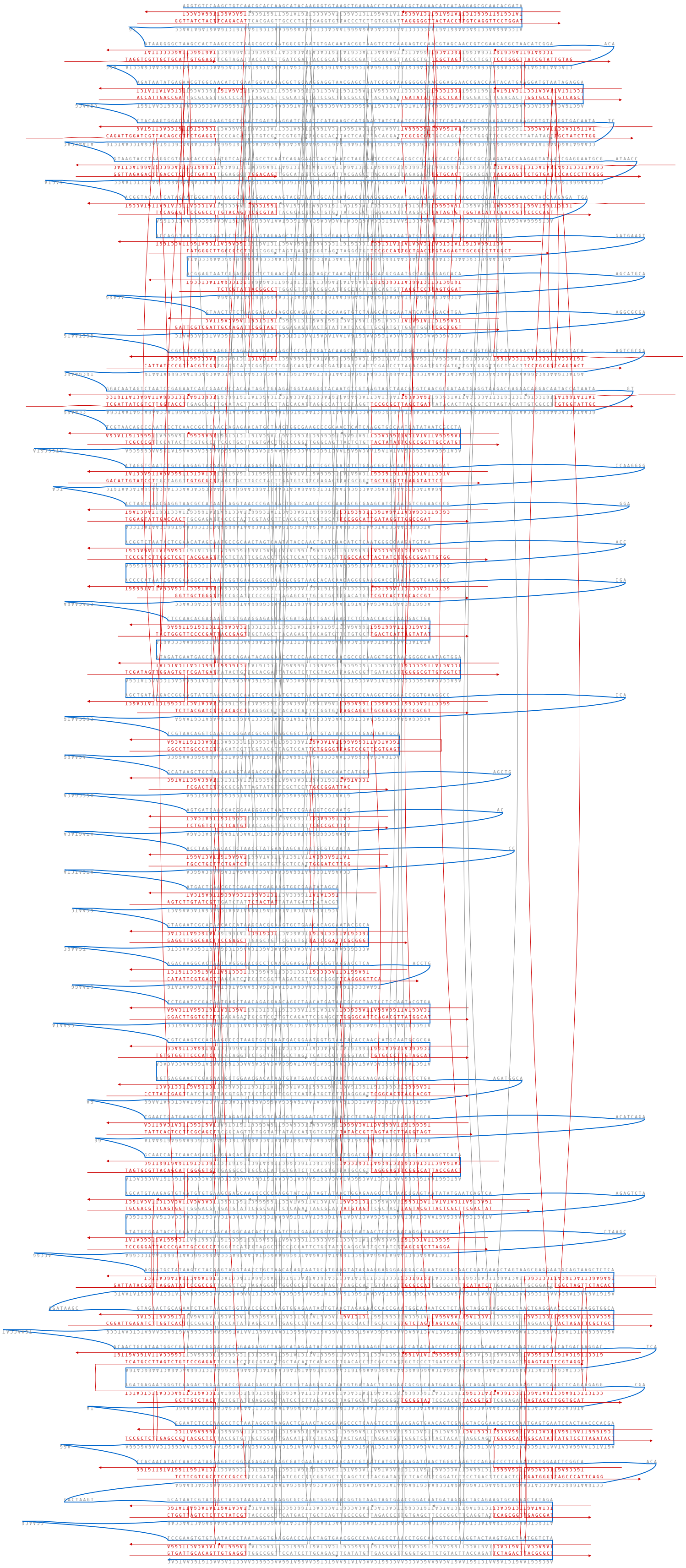
Supplementary Figure S15: caDNAno design diagram of the 126-helix bundle shown in Figure 3. Blue: SC1; orange: SC2; light blue: core staples; green: edge staples; Diagram was prepared with caDNAno v0.1.[22]



Supplementary Figure S16: caDNAno design diagrams of 13-helix bundles designed using 1024-, 1536-, and 2048-bases-long scaffolds. Diagrams were prepared with caDNAno v0.1.[22]



Supplementary Figure S17: caDNAno design diagram of the switch object shown in Figure 4. Both arms are depicted individually. In the uncleaved switch, both arms are connected via one DNAzyme cassette between the two positions labeled as “1”, and another DNAzyme cassette between the two positions labeled as “2”. Diagram was prepared with caDNAno v0.1.[22]



Supplementary Figure S17: caDNAno design diagram of the welding pointer object Figure 6. grey: core staples; red: edge staples. Diagram was prepared with caDNAno v0.1.[22]