

Video Article

Designing a Bio-responsive Robot from DNA Origami

Eldad Ben-Ishay¹, Almogit Abu-Horowitz¹, Ido Bachelet¹

¹Faculty of Life Sciences and the Institute for Nanotechnology & Advanced Materials, Bar-Ilan University

Correspondence to: Ido Bachelet at Ido.Bachelet@biu.ac.il

URL: <https://www.jove.com/video/50268>

DOI: [doi:10.3791/50268](https://doi.org/10.3791/50268)

Keywords: Bioengineering, Issue 77, Genetics, Biomedical Engineering, Molecular Biology, Medicine, Genomics, Nanotechnology, Nanomedicine, DNA origami, nanorobot, caDNAno, DNA, DNA Origami, nucleic acids, DNA structures, CAD, sequencing

Date Published: 7/8/2013

Citation: Ben-Ishay, E., Abu-Horowitz, A., Bachelet, I. Designing a Bio-responsive Robot from DNA Origami. *J. Vis. Exp.* (77), e50268, doi:10.3791/50268 (2013).

Abstract

Nucleic acids are astonishingly versatile. In addition to their natural role as storage medium for biological information¹, they can be utilized in parallel computing^{2,3}, recognize and bind molecular or cellular targets^{4,5}, catalyze chemical reactions^{6,7}, and generate calculated responses in a biological system^{8,9}. Importantly, nucleic acids can be programmed to self-assemble into 2D and 3D structures¹⁰⁻¹², enabling the integration of all these remarkable features in a single robot linking the sensing of biological cues to a preset response in order to exert a desired effect.

Creating shapes from nucleic acids was first proposed by Seeman¹³, and several variations on this theme have since been realized using various techniques^{11,12,14,15}. However, the most significant is perhaps the one proposed by Rothemund, termed scaffolded DNA origami¹⁶. In this technique, the folding of a long (>7,000 bases) single-stranded DNA 'scaffold' is directed to a desired shape by hundreds of short complementary strands termed 'staples'. Folding is carried out by temperature annealing ramp. This technique was successfully demonstrated in the creation of a diverse array of 2D shapes with remarkable precision and robustness. DNA origami was later extended to 3D as well^{17,18}.

The current paper will focus on the caDNAno 2.0 software¹⁹ developed by Douglas and colleagues. caDNAno is a robust, user-friendly CAD tool enabling the design of 2D and 3D DNA origami shapes with versatile features. The design process relies on a systematic and accurate abstraction scheme for DNA structures, making it relatively straightforward and efficient.

In this paper we demonstrate the design of a DNA origami nanorobot that has been recently described²⁰. This robot is 'robotic' in the sense that it links sensing to actuation, in order to perform a task. We explain how various sensing schemes can be integrated into the structure, and how this can be relayed to a desired effect. Finally we use Cando²¹ to simulate the mechanical properties of the designed shape. The concept we discuss can be adapted to multiple tasks and settings.

Video Link

The video component of this article can be found at <https://www.jove.com/video/50268/>

Protocol

The robot we will design in this paper responds to a protein *P* by making a cargo *C* available to bind to receptors on the surface of a chosen target cell. The robot is shown in **Figure 1**. *C* may be a receptor-blocking drug; a growth factor etc., and a way to chemically link it to a DNA oligonucleotide must be available that does not destroy its function. The robot has two states. When inactive, DNA gates on the two external 'lips' are hybridized, making sure the robot remains closed such that any cargo loaded within it is securely sequestered. In the presence of protein *P*, the gates open by either one of several mechanisms (discussed below) allowing the robot to open and expose the cargo. When designing the structure, consider that the robot has to be flexible enough to close onto itself in the closed state, and spring to the open state when the gates enable it to do so. Modeling the behavior of a DNA structure integrating thermodynamic and mechanical components is difficult, and the actual object might require some iterative improvement. Nevertheless, here we focus on the design process using a general working model, which can be built upon.

Note:

For a more comprehensive understanding of the process of DNA origami design and folding, we highly recommend reading the original caDNAno paper by Douglas and colleagues¹⁹ which explains the abstract representation of DNA in the design interface and how it relates to the actual molecular structure of a 3D DNA shape. This paper is accompanied by two video tutorials describing the caDNAno representation and interface in a very clear way. Additionally, we recommend reading the more recent paper by Dietz and colleagues describing many important aspects and detailed protocols of the folding process, including the Cando analysis tool²¹.

1. Download and Install caDNA 2.0 and Autodesk Maya 2012

Note: Autodesk software is free for students and academic use. The instructions below include setting up an academic account at Autodesk.

1. Create an academic account at <http://students.autodesk.com/>. After receiving the account setup e-mail, click the activation link and fill in your preferences as desired.
2. Download the free version of Maya 2012 from the download center.
3. Install Maya 2012 on your computer.
4. Run Maya once before installing caDNA 2.0.
5. Download and install the latest version of caDNA 2.0 from <http://cadnano.org/>.
6. Run Maya 2012. A caDNA icon should appear at the top right corner of the graphical user interface. Click the icon to go into caDNA.

2. Outline the Desired Shape and Scaffold Strand Path

1. The design interface of caDNA within Maya includes 3 panels (**Figure 2**):
 1. Top panel: lattice view, where the shape is initially outlined. This panel enables double helix-level actions and provides a section view of the shape.
 2. Bottom panel: editing panel, enabling single base-level actions.
 3. Right panel: a Maya-generated real time 3D model of the shape
2. Click the "Honeycomb" icon. Zooming in on and out of the lattice in the top panel can be done by mouse scroll up and down, respectively. caDNA enables two possible design lattices, honeycomb and square; in this paper we will use the honeycomb layout, although the square lattice could be generally used as well²².
3. Start by drawing the section of the desired shape on the left panel. Each circle represents a double DNA helix. To choose the helices which build the shape, simply left-click on their center (**Figure 3**). Continue helix by helix until the entire shape is outlined. Alternatively, the shape can be drawn by pressing the left mouse button and continuously drawing the shape's outline. Any action can be undone by clicking Edit menu and "Undo", or by the keyboard shortcut CTRL+Z (PC) or CMD +Z (Mac).
At this point, the selected helices will appear yellow. At the same time, the bottom panel will display a side view of the shape, comprised of these helices. The helix numbering in the bottom panel is consistent with numbering in the top one.
4. Observe the bottom panel. Each helix is represented by two rows of squares: the rows are the two strands of the double helix, with each square representing a base (**Figure 4**).
The orange vertical bar determines where editing actions take place along a helix. The base position along the grid appears as a number above the orange bar. The helix framework's default length is 42 bases. The length can be extended by clicking one of the grey arrow icons at the top right corner of the editing panel and choosing the extension length (in multiples of 21, which correspond to two full turns of the DNA helix, in which one turn spans 10.5 bases) (**Figure 4**). The grid will extend to the direction of the chosen arrow.
5. To plot the actual scaffold strand path throughout the shape, press the mouse button, start from the first helix and go continuously over all the helices following the same order they were initially selected in section 2.3. **Note that:**
 1. The helices selected this time will change color to orange.
 2. In the bottom panel, scaffold strand fragments will be automatically drawn at the selected helices.
 3. The right panel will show the 3D model of the shape being built in real time. At the end of this process, a draft of the scaffold strand path will be automatically drawn in the bottom panel (**Figure 5**).
6. Draw a rectangle around all the leftmost edges of the scaffold path. **Note** that edges so selected will appear red (**Figure 6**).
7. Extend the scaffold path by dragging the selected edges as a group to the left side of the grid. Repeat this process for the right edges until the path is properly extended. **Note** that scaffold extension also extends the 3D shape in the right panel (**Figure 7**).
8. Locate the scaffold path parts that are isolated from the rest, and connect them. In our shape, for example, helices 0-9 form an isolated part. Helix 9 needs to be connected to helix 12 (**note** that helices 9 and 10 are not adjacent in the shape [top panel] so they cannot be connected).
9. Zoom in on the strands to be connected, and using the "Select" tool click on any point on one of the strands. Upon clicking any point along a blue scaffold fragment, 'bridge' icons appear between helices, denoting the positions where crossovers are allowed. In these positions, bases in adjacent helices face each other directly, allowing the strands to cross over from helix to helix without deforming or twisting the DNA. The number appearing next to each bridge icon indicates the number of helix it will crossover to (**Figure 8**).
10. To create crossovers, left click the bridge icon of choice. A scaffold crossover will be generated, meaning the scaffold crosses at this point from helix to helix (**Figure 9**). Repeat this process until the scaffold traverses all helices and creates a closed loop that spans the entire shape, leaving no regions that are isolated from the rest of the shape.
Note that while crossovers appear to span a distance in the software, in reality they do not include any DNA base. Physically, the crossover "bridge" contains only one phosphate unit of the DNA backbone that links the two bases from the adjacent helices together.
11. Before moving on to the next step, make sure the entire scaffold is continuous, and no part of it is isolated from the others.

3. Define Opening Mechanism Axes

The described robot opens in response to a defined biological input to expose its payload. Opening takes place in a shell-like manner, with two halves (helices 0-29 make up one half, helices 30-61 make up the second half) revolving around two axes. The axes are formed by crossovers between helices 29-30 and 61-0, which are the only crossovers between those halves and are positioned only in or close to the left edge of the grid. The right edge will contain the gate strands (discussed below).

1. Erase the existing crossover between helices 29-30. To erase the crossover, click the "knee" point in either strand. This leaves a nick in both strands where the crossover used to be. To seam the nicks, press SHIFT and click each nick.

2. Create a new crossover between helices 29-30 as close as possible to the left edge of the grid (**Figure 10**).
3. Create a new crossover between helices 61 and 0 as close as possible to the left edge of the grid.

4. Define Payload Attachment Sites

After we finish plotting the scaffold strand path, we need to define the payload attachment (loading) sites. Loading sites are in fact staple strands that extend out of their helices as single stranded 'branches'. It is therefore important to define very precisely where along the helix this branching occurs, to make sure it extends to the desired direction. If we define staple extensions arbitrarily, loading sites might occur on the external side of the robot instead of the internal side.

To make sure a staple extends to a specific direction only, we plot an additional helix, which serves as guides for the directional branching of the staple from the main body. After extending the desired loading site staple, the guide helix is removed.

1. Let us define 4 loading sites facing towards the internal side of the robot. The loading sites will branch out of helices 3, 27, 34, and 58. For each site, in the top panel click the helix immediately adjacent to these helices that faces the internal side (**Figure 11**). This will add the helices to the grid in the bottom panel. Do not second-click these helices yet.

5. Add and Edit Staples

1. Click "AutoStaple". The software will automatically add staple sequences in various colors (**Figure 12**). **Note** that staples have been added to the 3D shape in the right panel. Staple colors are consistent for the bottom and right panels. In addition, there is an indicator on the bottom left corner of the interface, which indicates a staple.
Note: staples cannot be too long, too short or circular. Most of the staples generated here do not meet these criteria, and have to be edited. The first step in editing them is automatic (see next step).
2. Click "AutoBreak". A dialogue box will open (**Figure 13**), asking for user-defined parameters for this action:
 1. Target length (bp): anticipated length of staple if possible
 2. Min length (bp): minimal length allowed for a staple
 3. Max length (bp): maximal length allowed for a staple
 4. Min dist to xover (bp): the minimal number of base pairs a staple can traverse between its edge and a crossover or between two crossovers.
 Use the default parameters, click OK. The software will break the staples according to these parameters to the best of its ability (**Figure 14**).
3. Erase all the staple crossovers between helices 29-30 and 61-0, to enable these helices to separate and enable the robot to open. Erasing staple crossovers will require some manual editing to correct staples that become too short or irrational as a result of this action. To do this properly, follow the instructions in the following sections.
Make sure to leave the scaffold crossovers created in sections 3.2 and 3.3 intact.
4. Consider, for example, the first staple crossover (cyan and black staples) from the left between helices 29 and 30 (**Figure 15**). Erase both bridges of this crossover by clicking each knee point or bridge so it appears red, then hitting DELETE (**Figure 16**).
5. Seam the two staples on helix 29 by pressing SHIFT and clicking the nick between them. Similarly, seam the three staples on strand 30 to a single staple (**Figure 17**). Staples can be manually extended or shortened by clicking an edge and dragging it as desired. Take care not to circularize any staple. **Figure 18** shows the gap between helices 29-30 after complete editing of staple crossovers. Repeat this process for helices 0 and 61, and manually edit all the staples in each helix.
6. Locate staples that are drawn by a thick line, meaning they require further editing. Examine each one and correct as necessary. For example, staples that are too short can be erased (**Figure 19**) or extended if possible.

6. Create Loading Sites and Gates

1. Second-click the loading site helices in the top panel, and extend the resulting scaffold strand fragments in the bottom panel by clicking an edge and dragging it as desired (**Figure 20**).
2. Manually add staples to these scaffold fragments by placing the orange vertical bar at the desired position along the scaffold, going over the guide helices on the left panel, holding SHIFT and clicking. This will add a staple precursor at each helix (**Figure 21**).
3. Extend the staple precursors to full length as well by clicking and dragging.
4. Locate the red bridge icons, denoting allowed crossover positions between the guide strand (for example, helix 62) and the chassis (for example, helix 3).
5. Choose the most convenient location to introduce a crossover and click the bridge icon (**Figure 22**). A convenient location requires minimal editing of existing staples in the chassis.
6. In the guide helix (helix 62), delete the staple part that is not a part of the loading site, and shorten the participating part to the desired length. The desired length should provide both specificity for loading different types of cargo, and binding strength. Typically, an 18-mer tail should be fine. Make sure the staple remains drawn by a thin line, otherwise edit it until it is.
7. In the chassis, edit the changed staples as necessary.
8. Erase the guide (helix 62) leaving only the staple extension.
9. Repeat steps 6.4-6.8 for all loading sites (**Figure 23**).

7. Designing Gate Strands

The gate strands are the only strands, except for the axes, linking helices 29-30 and 61-0. In contrast to the axes, the gate strands are not crossovers. Rather, they hybridize to form a double stranded segment that serves as the sensor for the biological input of choice. Once the gate duplexes are displaced, the entire robot can entropically revolve around the axes and open.

1. Locate the proper positions for gate strands. These will be staples on helices 29, 30, 61, and 0.
2. For example, examine the 29-30 gate region. There are convenient staple strands flanking helices 29 and 30 on the right side of the grid, which can be used as gate strands. **Note** that they face opposite directions.
3. Click the edge of one of the potential gate strands to extend it outside of the shape. If the edge lies over a scaffold crossover, its selection could be simplified by making sure only "Stap"(les) are selectable, by clicking off "Scaf"(fold) in the "Selectable" toolbar on the top right side of the interface.
4. Extend both staples to form the gate strands. Edit the staples if this extension requires it (**Figure 24**). Repeat this for the gate strands of helices 0 and 61.
Note that for now, the actual length does not matter, since the sensor DNA (e.g. aptamer) will replace the gate strand sequences at the sequence completion step.

8. Choose Scaffold Sequence

1. Click the "Seq" tool. Place the cursor anywhere on the scaffold strand and click. A dialog box will open asking us to choose the scaffold DNA source (**Figure 25**).
2. Choosing the source DNA greatly depends on the robot size. For example, M13mp18 ssDNA (p7249), and its extended derivatives (p7308 etc.) which have generally been the choice for large DNA origami shapes, fit when the scaffold strand is ~7 kb long. If the scaffold of the designed shape is significantly shorter than the chosen source, the excess scaffold strand that is not hybridized to any staple will create a loop of ssDNA protruding from the folded shape. While this usually poses little problem for relatively short loops, multi-kb long loops could drastically interfere with folding and function of the robot. Therefore it is important to fit the chosen source to the shape scaffold length.

For example, if the scaffold strand needed to fold a small shape is ~1,600 bases long, which is significantly shorter than the preset sources in the dialog box, a custom sequence can be used as scaffold. Several sources can be considered. For example, the M13mp18 can be digested with a specific restriction enzyme that produces a fragment of the desired length. Designing such a source can be done at [NebCutter](http://tools.neb.com/NEBcutter2/) (<http://tools.neb.com/NEBcutter2/>) by pasting the M13mp18 sequence (http://www.neb.com/nebecomm/tech_reference/restriction_enzymes/dna_sequences_maps.asp?#UAggyzFWomR) in the NebCutter input window, and mapping restriction sites. Another option is to use pre-digested ssDNA, such as the phiX174 virion ssDNA HaeIII digest, available from New England Biolabs.

3. In the dialog box, click "M13mp18". **Note** that the chosen DNA sequence has been added to the scaffold and staple strands in the bottom panel.

9. Export Staple Sequence as a Spreadsheet

1. Click "Export" on the top toolbar, and choose a destination filename for the staple list. Click "Save".
2. Locate the destination .csv file and open it.
3. The spreadsheet shows the staple list, which can be sent as is to a DNA synthesis company. The first two columns display the start and end coordinates, with the number outside the brackets denoting helix number and the number inside brackets denoting base position.

10. Assign Gate and Loading Sequences

1. In the staple list, you will notice that some sequences begin or end with a string of question marks "?????". These question marks denote that since no scaffold strand hybridizes with these specific staple regions, they cannot be assigned complementary sequences. These are in fact the extensions we designed for the gate strands and loading sites, and therefore these need to be assigned manually now.
2. Gate:
 1. The gates determine the nature of the biological input upon which the robot will switch from inactive to active state and expose its payload. Each single dsDNA gate can encode response to one biological input (or more), so a profile of inputs required for robot activation can be defined.
Let us assume for this example that the biological cue triggering robot activation is a restriction enzyme, which could indicate the presence of infectious bacteria.
 2. First consider that the gate ssDNA strands do not hybridize immediately after branching out of their helices. Designing the gate otherwise might impede hybridization during folding. Therefore, each branch should start with a spacer string. We typically use poly-T as spacer strings, as this sequence provides flexibility.
 3. We also assume that the length of the gate hybridization region is 20 bases, containing the target restriction site in its middle.
 4. Therefore the gate might look like this:
[Helix 29]-5'-.....TTTTTTGTGAGTTxxxxxGCTAGAG-3'
[Helix 30]-3'-.....TTTTTTCACTCAAxxxxxCGATCTC-5'
The "....." denote the staple region that hybridizes with the scaffold strand, therefore it has a sequence already and should not be changed.
The random duplex "GTGAGTT" and its complement ensures the restriction site is not partly open, and provides some extra bases to ensure effective digestion by the enzyme.
The "x" denotes the restriction site.

The random duplex "GCTAGAG" and its complement provide some extra bases for the enzyme to work efficiently, but also makes sure the gate strand is sufficiently long to ensure good robot closure.

Before choosing a restriction site as a target, make sure the entire robot structure, loading sites and other part of the gate itself are not digested by the enzyme of choice. In this examination, the NEBCutter 0-cutter list (enzymes which do not cut the entire sequence) highlighted *EagI*, isolated from the *Enterobacter Pantoea agglomerans*, as a potential enzyme that could indicate the presence of an enterobacterial infectious.

5. The gate now looks like this (yellow marks *EagI* restriction site):

[Helix 29]-5'-.....TTTTTTTGTGAGTTCGGCCGGCTAGAG-3'

[Helix 30]-3'-.....TTTTTTTCACTCAAGCCGGCCGATCTC-5'

Note that this design assumes that after digestion, the sequence "GTGAGTTCGG" ($T_m = 32^\circ\text{C}$) is not sufficiently long or thermodynamically stable to hold the robot closed anymore. This assumption will most likely need to be verified experimentally.

6. The second gate can be the same in which case the robot would only respond to one enzyme, or can be designed with a different site, increasing specificity of robot. More restriction sites can be added onto the same strand, increasing the complexity and specificity of the robot.

3. Loading sites:

1. The loading site can be a universal sequence. Alternatively, loading sites can be based on unique sequences, which will decrease modularity but improve control over cargo orientation and ratios (for different types of cargo).
2. Finally, the loading site oligonucleotides should include a chemical functional group enabling them to conjugate with any payload: protein, nanoparticle, etc. Make sure the chemical group is mounted on the correct end (5' or 3'), according to the staple direction.

11. Simulate Results in CANDO

1. After the job is saved as a .json file, it can be uploaded to CANDO for analysis. CANDO is a finite-element based simulation of the DNA structure that can estimate its rigidity and stability in solution²¹.
2. Go to <http://cando-dna-origami.org/>
3. Click "Submit a caDNAno file for analysis" and fill all the necessary information.
4. Analysis in CANDO usually takes up to 15-20 min. At the end, an e-mail message lets us know the analysis is complete, providing a link to download the simulation results (**Figure 26**).

12. Order DNA and Fold the Robot

Once the design process is complete and CANDO analysis shows satisfactory prediction of the product, the staple strand list generated in sections 9-10 can be ordered. Typically, staple strands do not require particular purification; however, it is recommended that special purpose strands such as gates or loading sites be purified by HPLC.

The steps following DNA order, namely folding, purification and evaluation of product, including visualization of the folded structure by either atomic force microscopy (AFM) or transmission electron microscopy (TEM) are outside the scope of this paper, and can be found in previous reports^{17,18,20,21}. A TEM image of the robot designed here is brought as an example (**Figure 27**). Sample preparation and staining was carried out exactly as described elsewhere²¹.

Representative Results

Figures 1-25 are screenshots of the caDNAno 2.0 interface showing the design process step-by-step. The cross-section of the shape was first outlined (**Figure 3**), followed by automatic addition of scaffold strand fragments and completion of the entire scaffold path (**Figure 7**). Staple strands are automatically added (**Figure 12**), broken according to user-defined parameters (**Figure 14**), and manually edited to adapt the staples to the desired function of the device (**Figures 15-18**). **Figures 23-24** describe how loading site and gate strands are added and edited. Finally, **Figure 27** shows a TEM image of the model designed here.

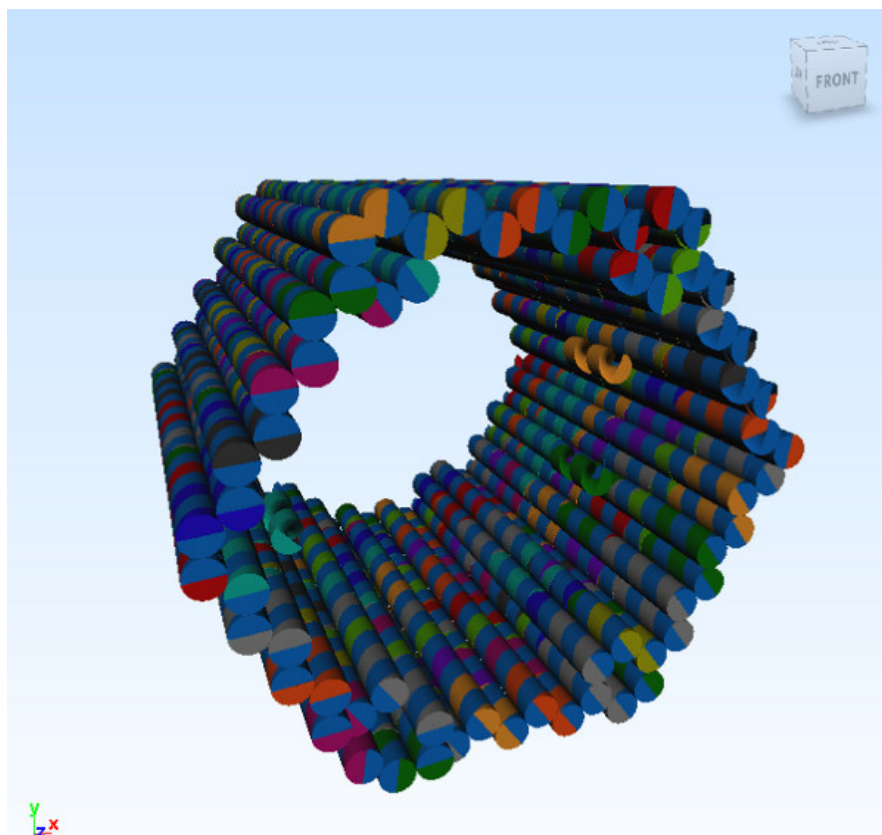


Figure 1. A 3D model of the finished robot, designed by caDNano 2.0 and generated by Autodesk Maya 2012.

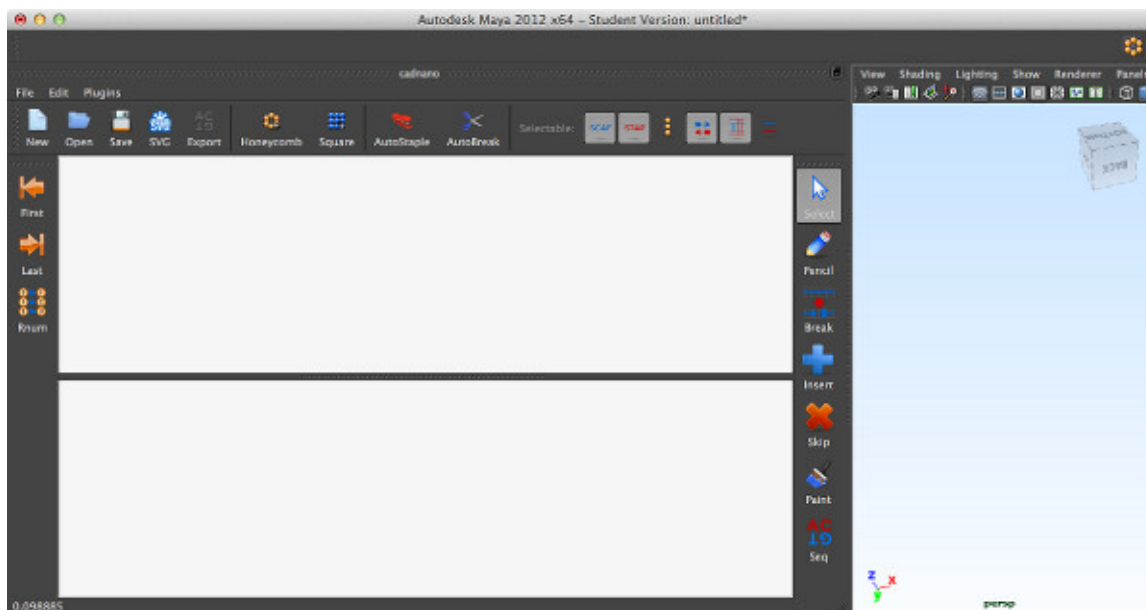


Figure 2. A view of the caDNano 2.0/Autodesk Maya 2012 design interface. Top panel: lattice panel for outlining the initial shape. Bottom panel: editing panel. Right panel: 3D model generator (see section 2.1). [Click here to view larger figure.](#)

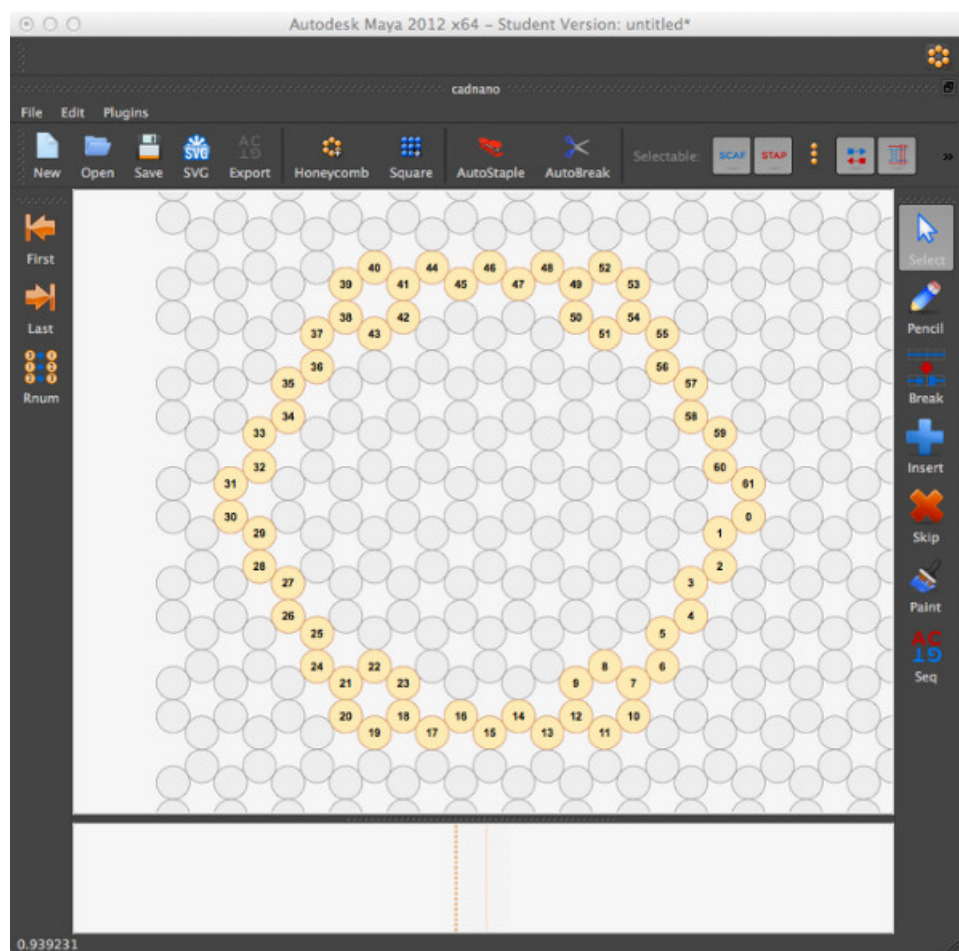


Figure 3. Drawing the section of the shape on the top panel (see section 2.3).

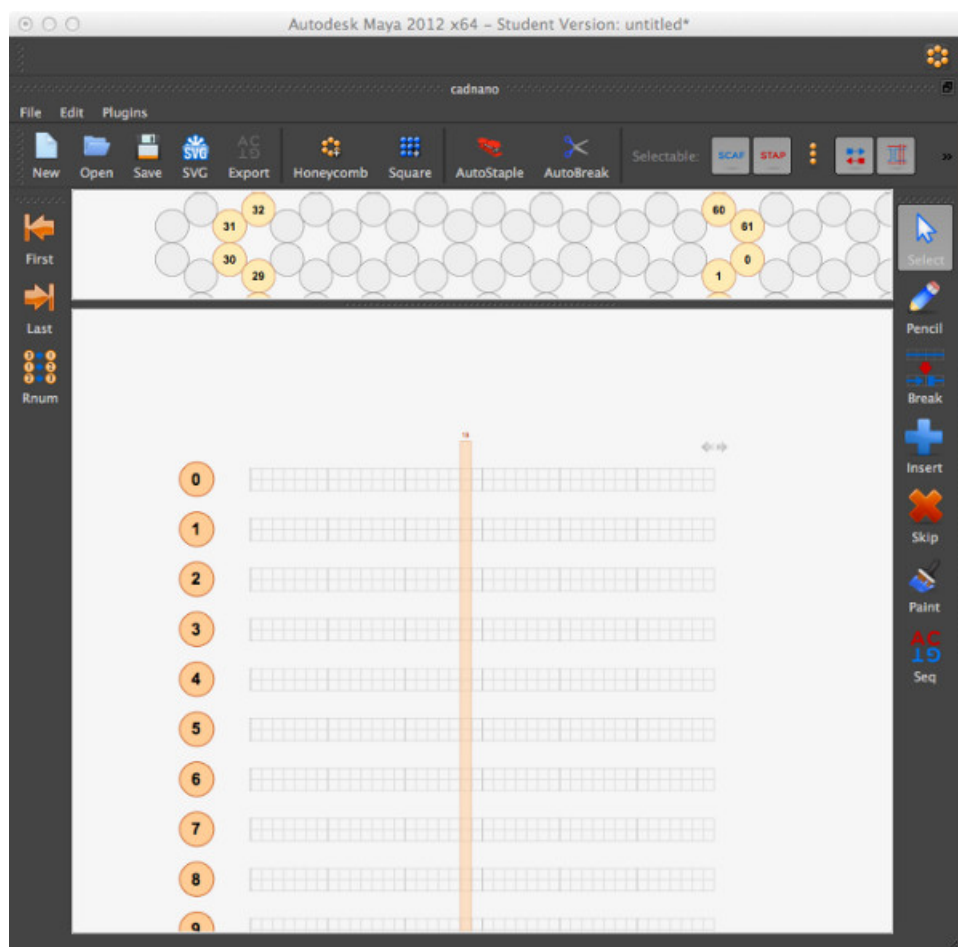


Figure 4. The bottom (editing) panel of caDNano 2.0. The orange vertical bar determines where along the grid editing actions will occur. The grey arrows on the top right corner are used to extend the grid to either side (see section 2.4).

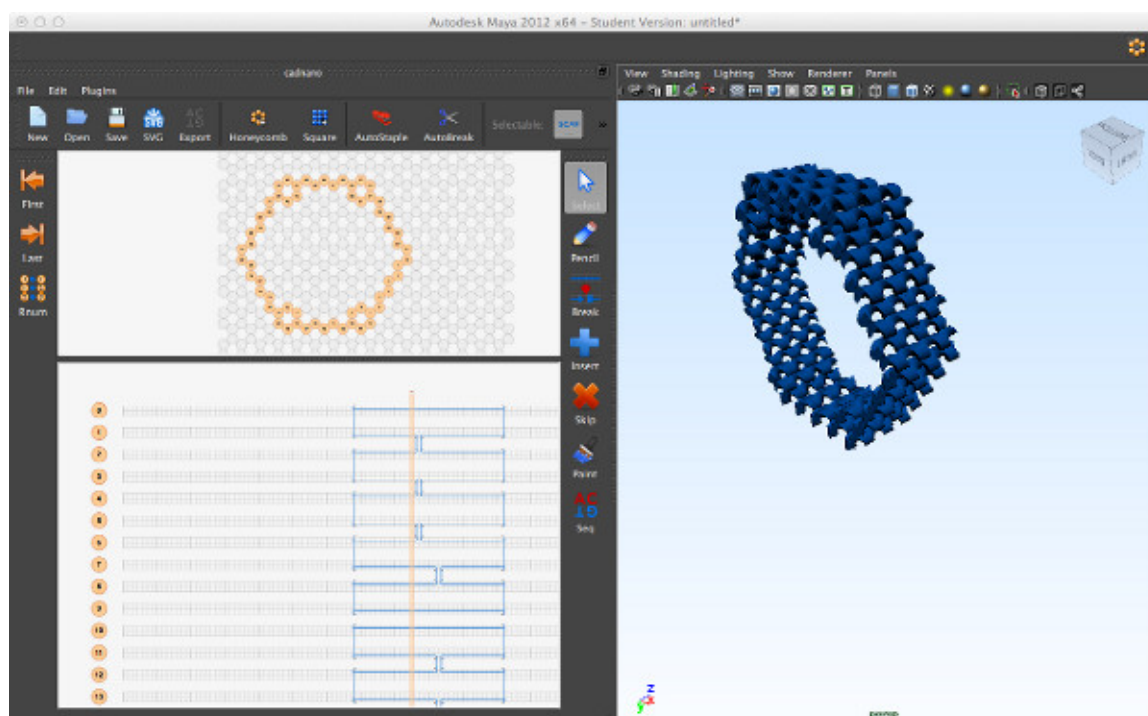


Figure 5. A draft of the scaffold strand after the initial outline in the top panel (see section 2.5). [Click here to view larger figure.](#)

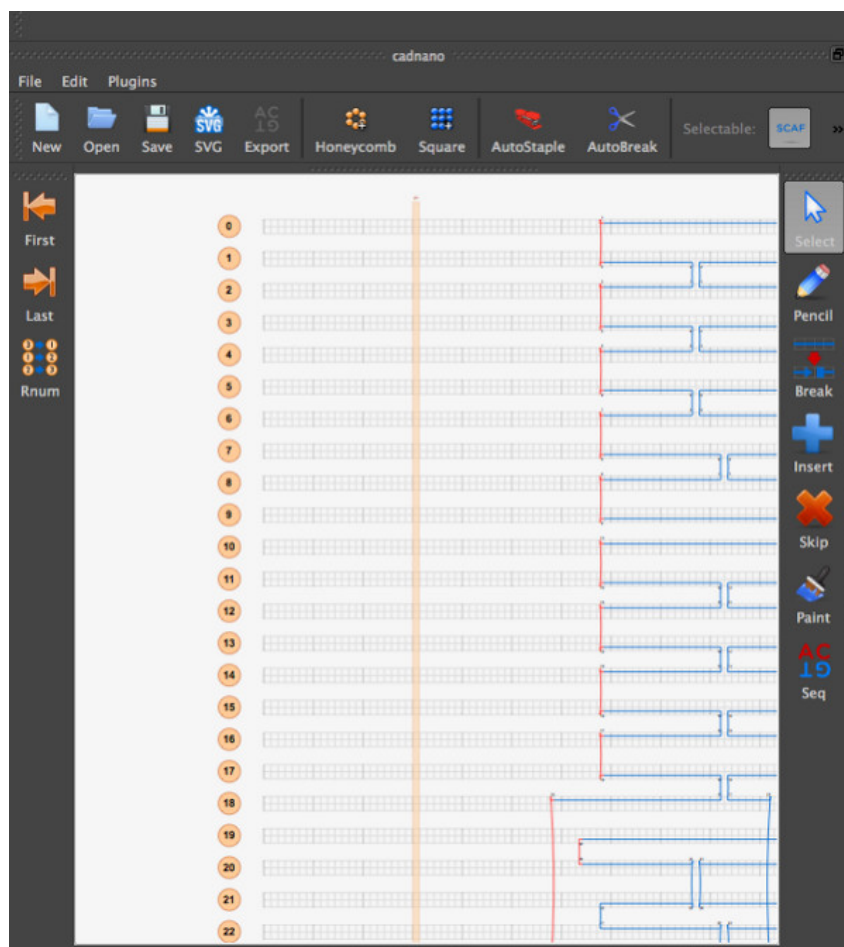


Figure 6. Selecting all the scaffold strand path edges and extending the path to the desired length (see section 2.7).

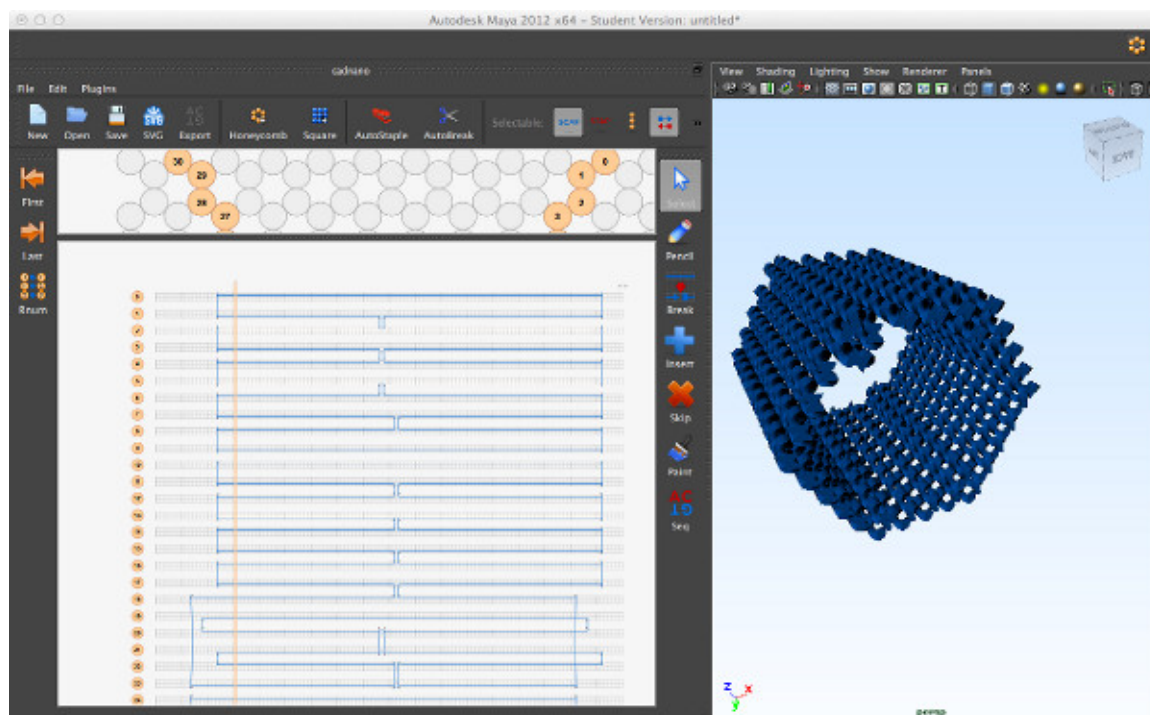


Figure 7. A general view of the bottom and right panels demonstrating how the 3D model changes in real time along with editing actions. [Click here to view larger figure.](#)

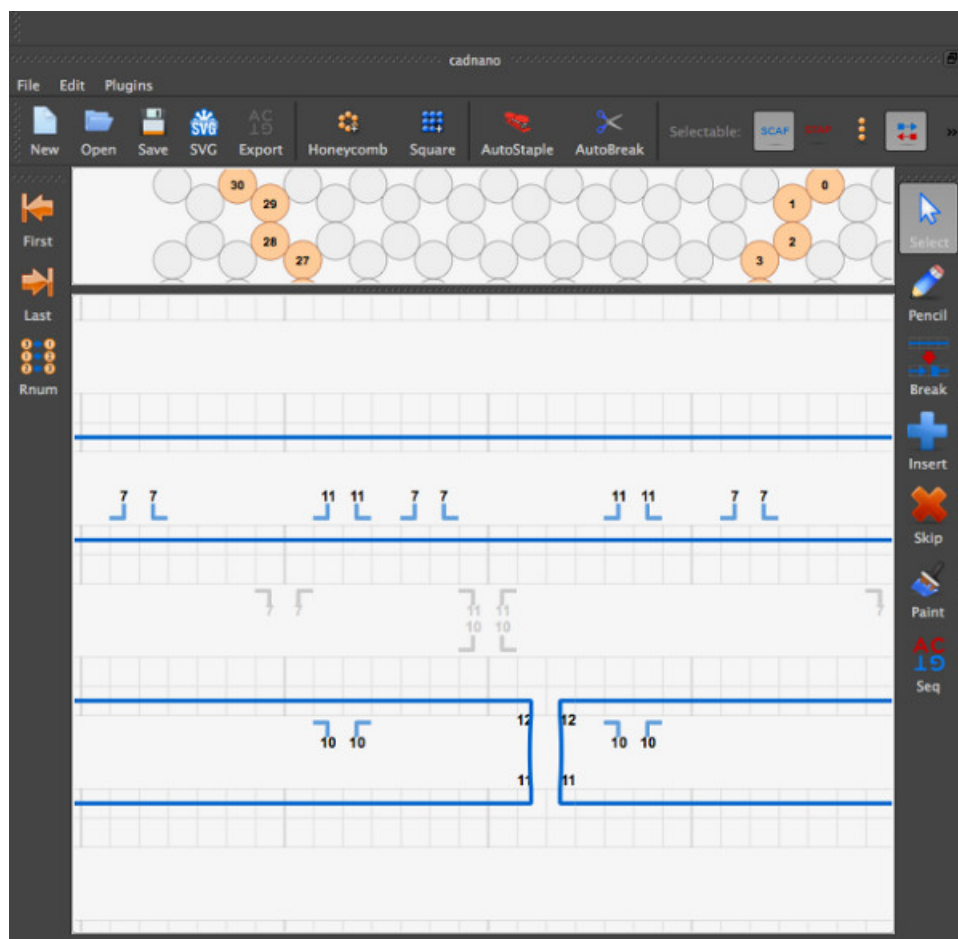


Figure 8. The blue bridge icons between helices denote the positions where scaffold crossovers are allowed (red icons refer to staple crossovers and are not yet shown, see section 2.9).

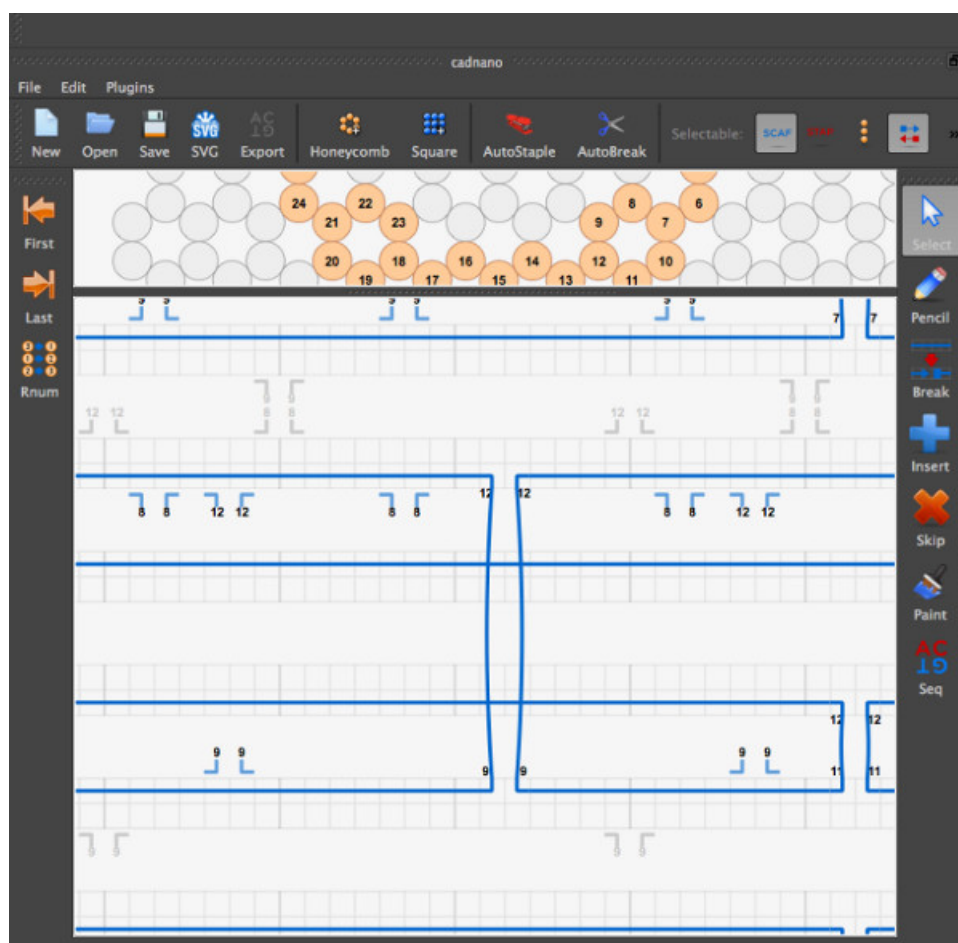


Figure 9. Creating new scaffold crossovers by clicking the bridge icons of choice (see section 2.10).

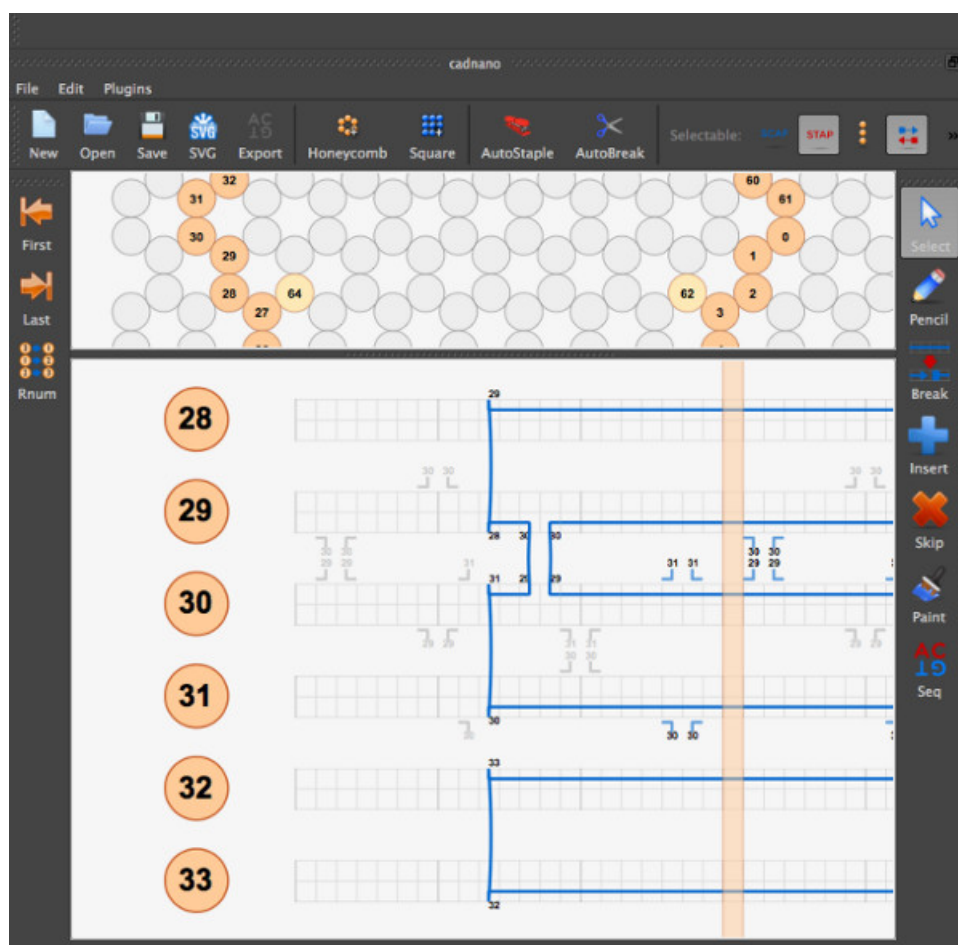


Figure 10. Creating an axis (a crossover a close as possible to the left side of the grid) between helices 29 and 30 (see section 3.2).

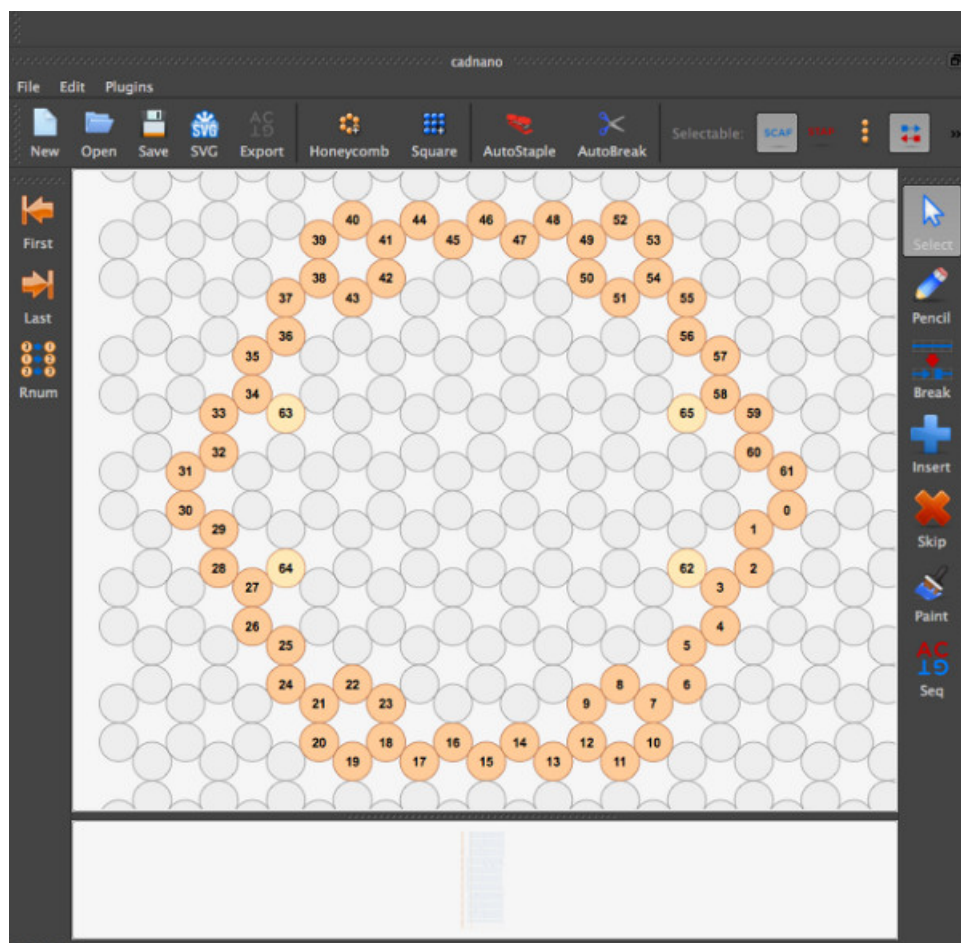


Figure 11. Adding helices that guide the branching of loading sites (see section 4.1).

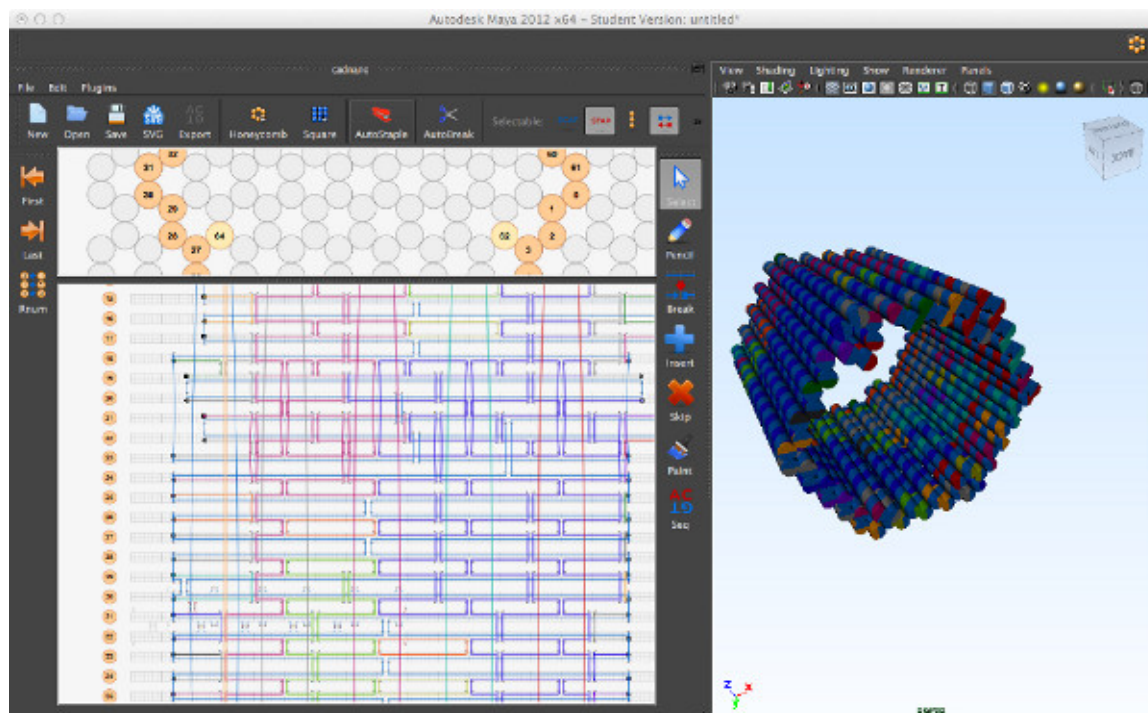


Figure 12. The blueprint after the "AutoStaple" action. The staple colors in the bottom panel and the right panel are consistent (see section 5.1). [Click here to view larger figure.](#)

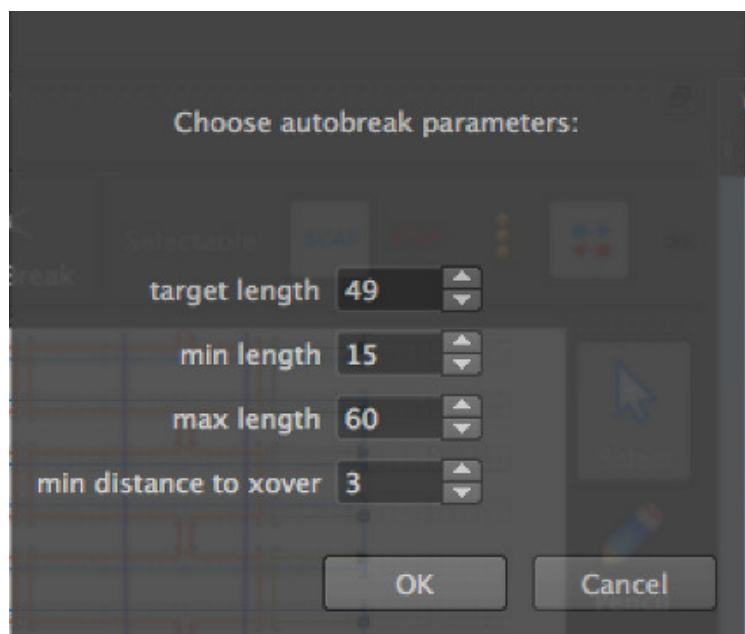


Figure 13. The "AutoBreak" dialog box in which the user can define AutoBreak parameters (see section 5.2).

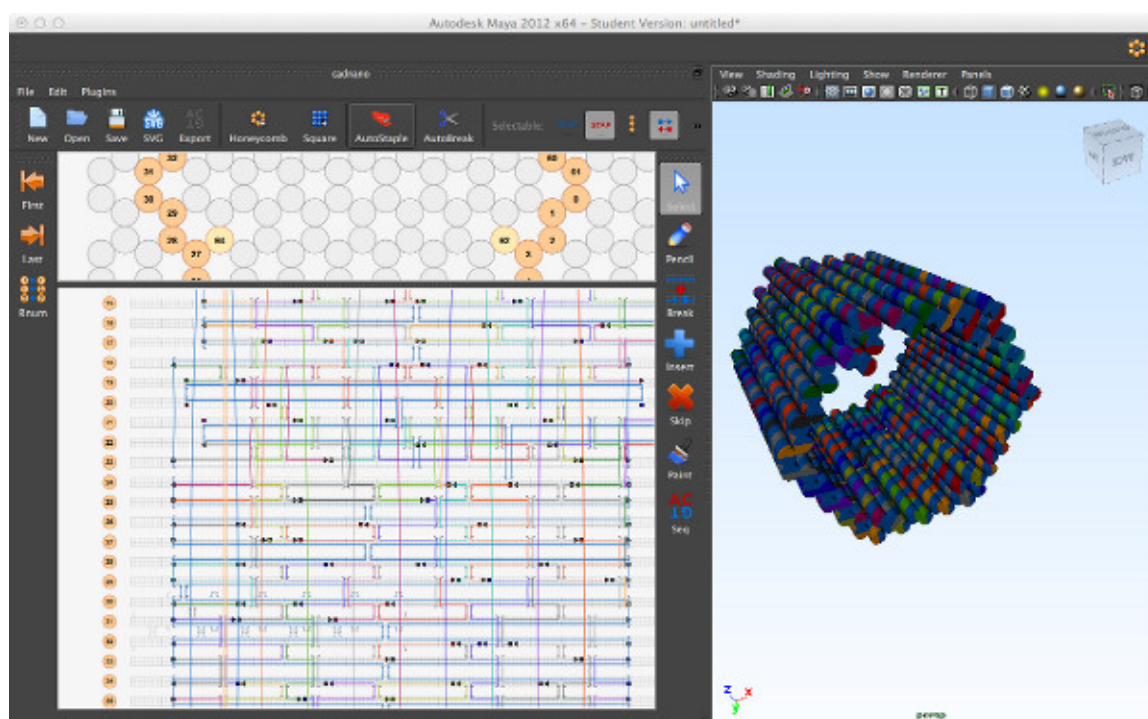


Figure 14. The blueprint after the "AutoBreak" action (see section 5.2). [Click here to view larger figure.](#)

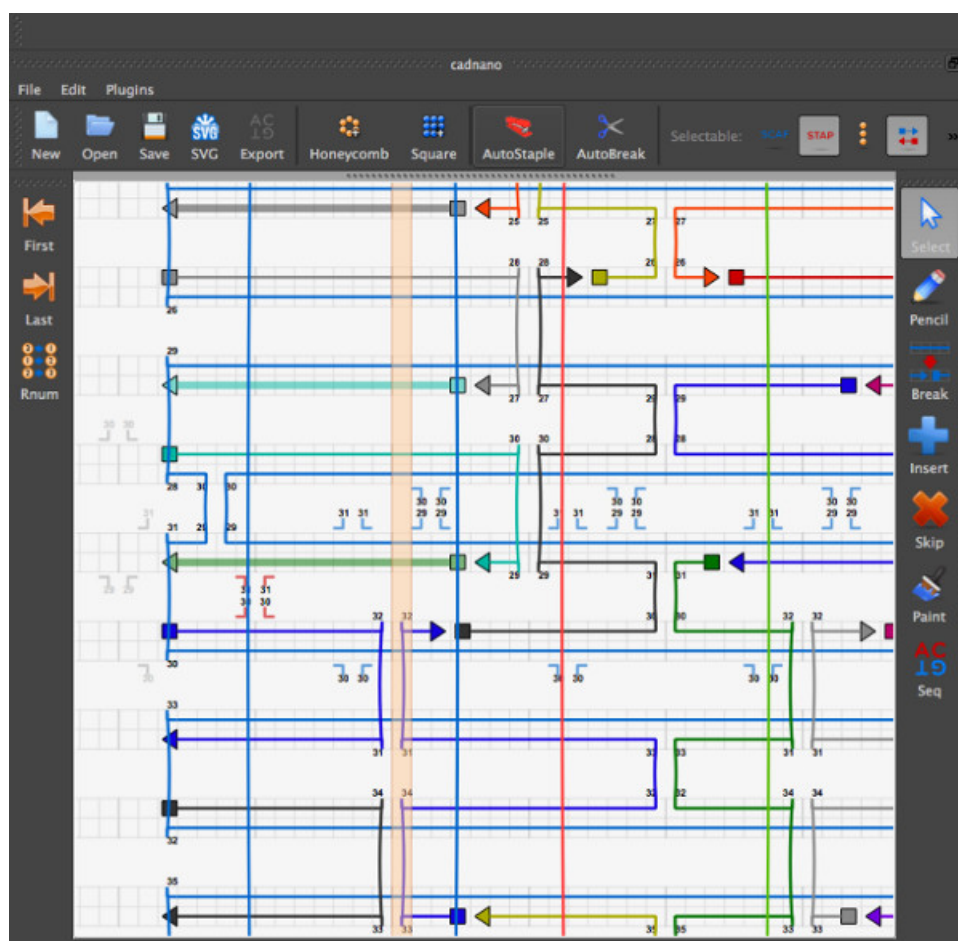


Figure 15. Manual editing of staples I: locating staples that cross over from helix 29 and 30 and should be deleted.

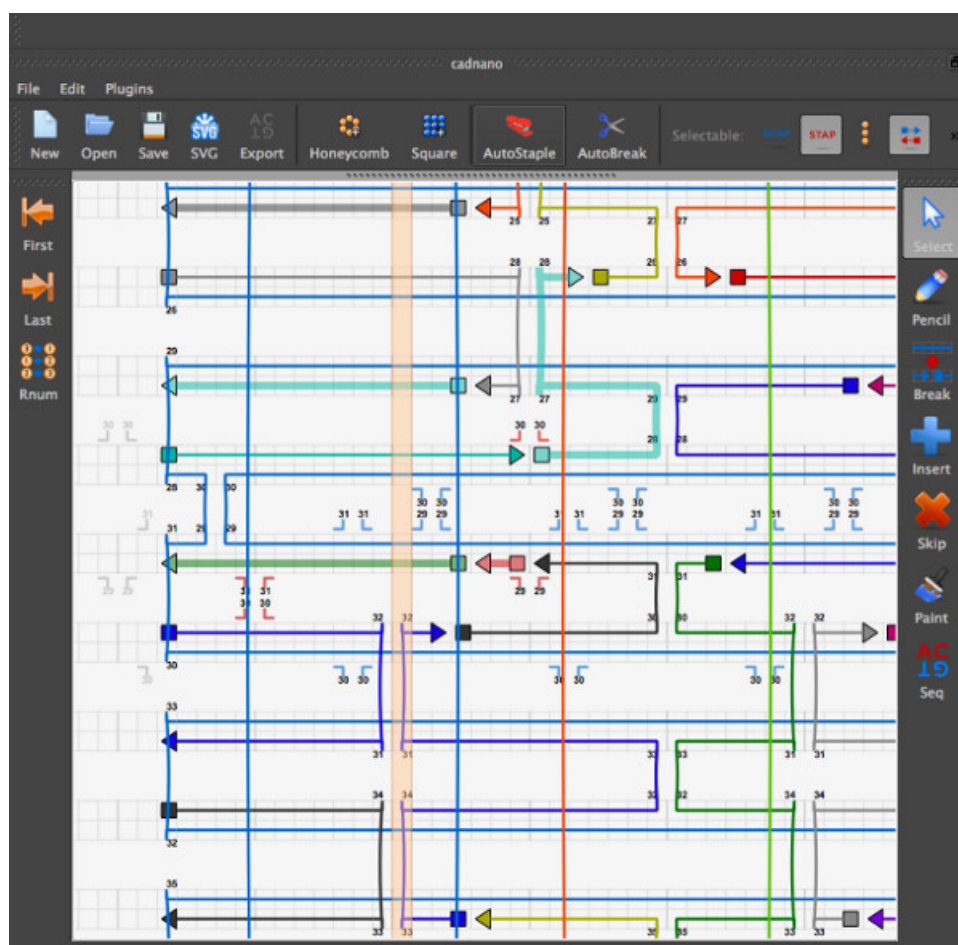


Figure 16. Manual editing of staples II: deleting the bridges between the located staples.

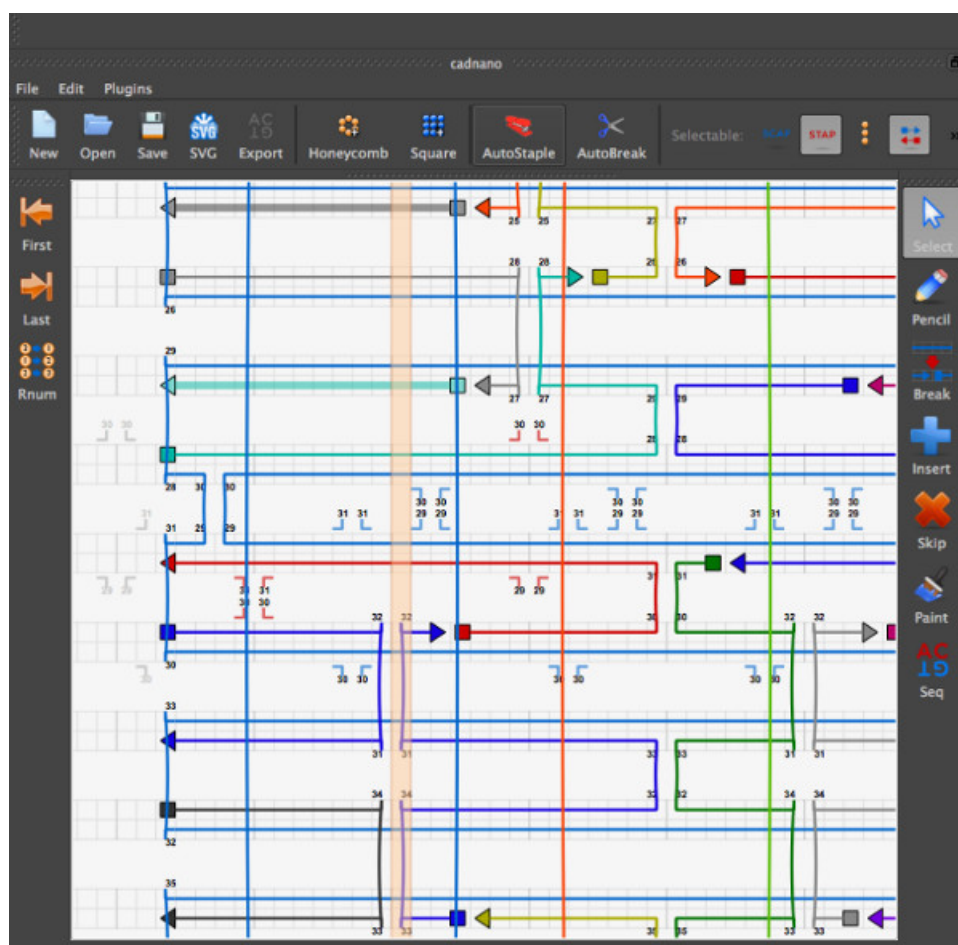


Figure 17. Manual editing of staples III: seaming the nicks along fragmented staples (see section 5.5).

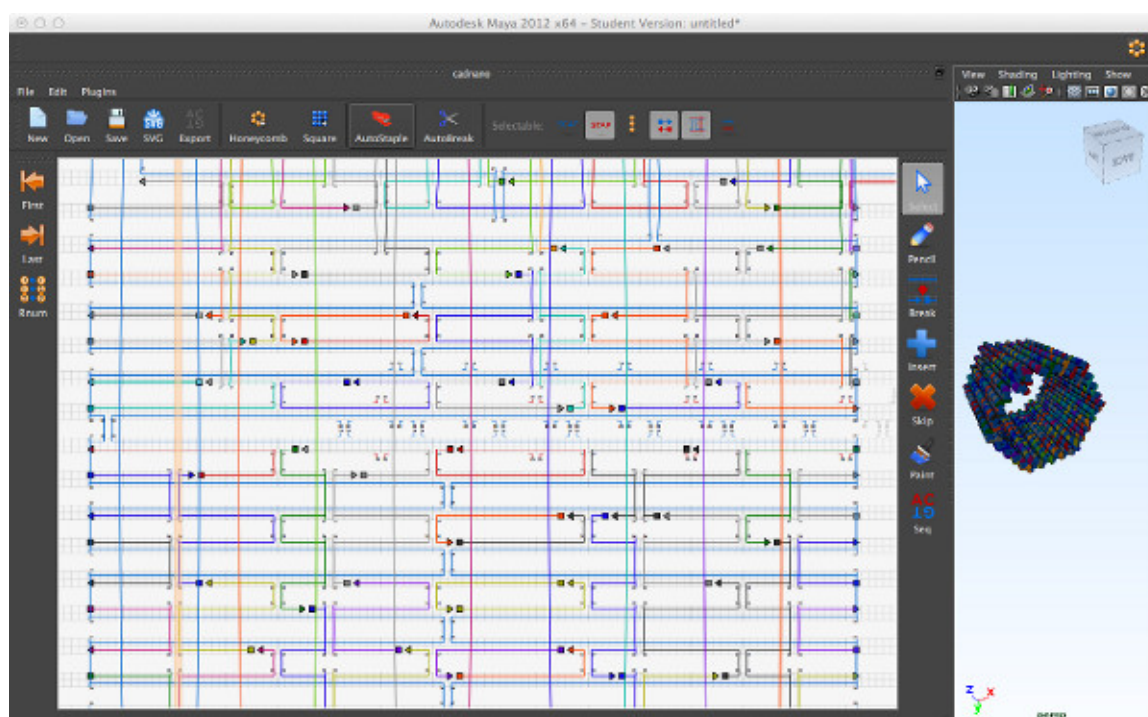


Figure 18. The entire gap between helices 29-30 showing no crossovers link the two (see section 5.5). [Click here to view larger figure.](#)

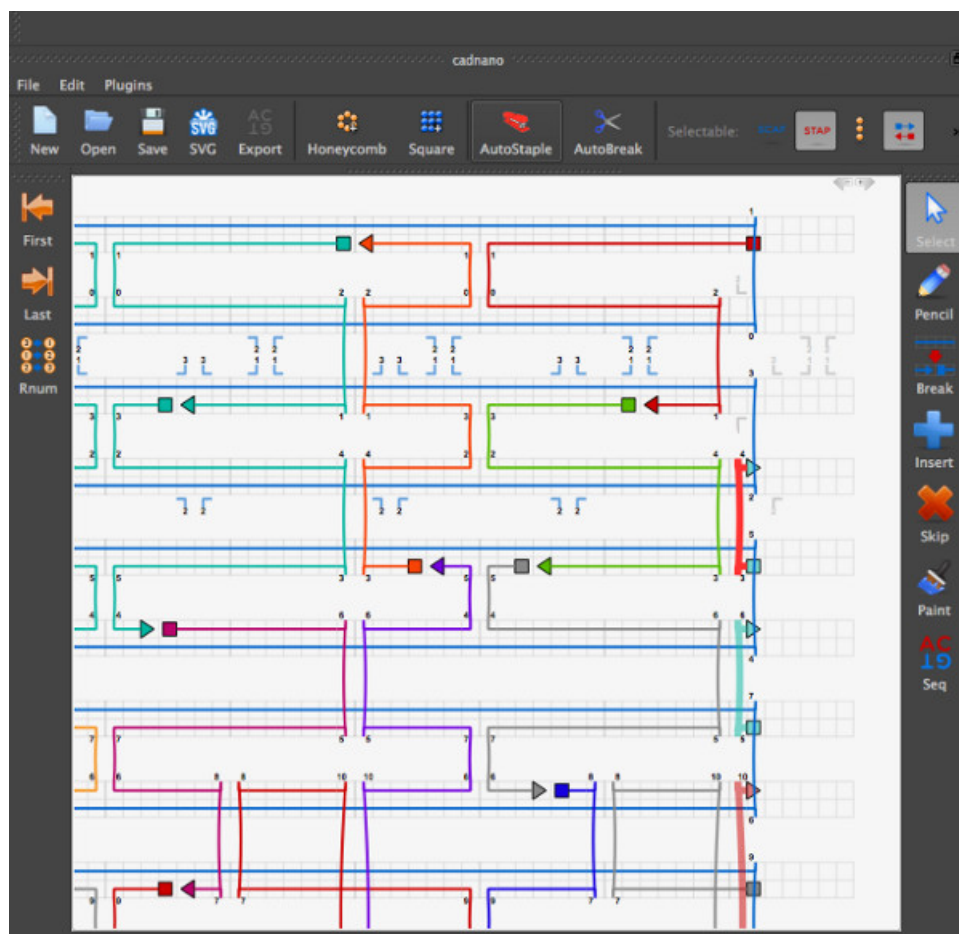


Figure 19. Manual editing of staples drawn in thick line (denoting they are either too short, too long or circular, see section 5.6).

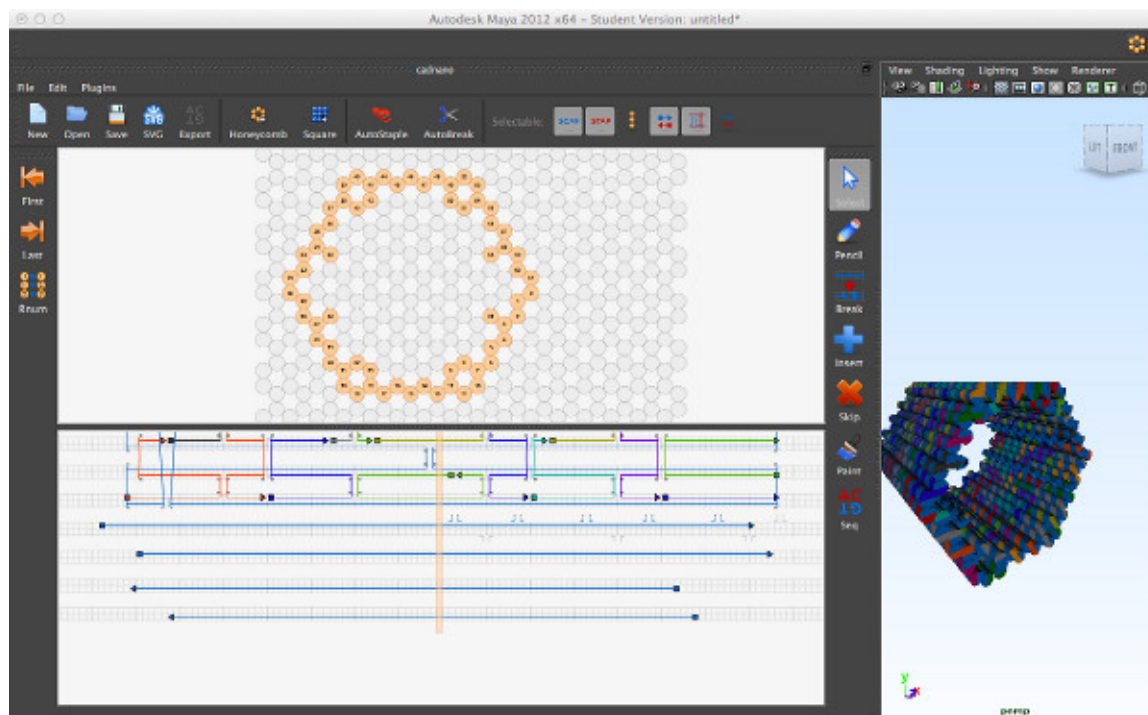


Figure 20. Adding guide helices for loading site branching (see section 6.1). [Click here to view larger figure.](#)

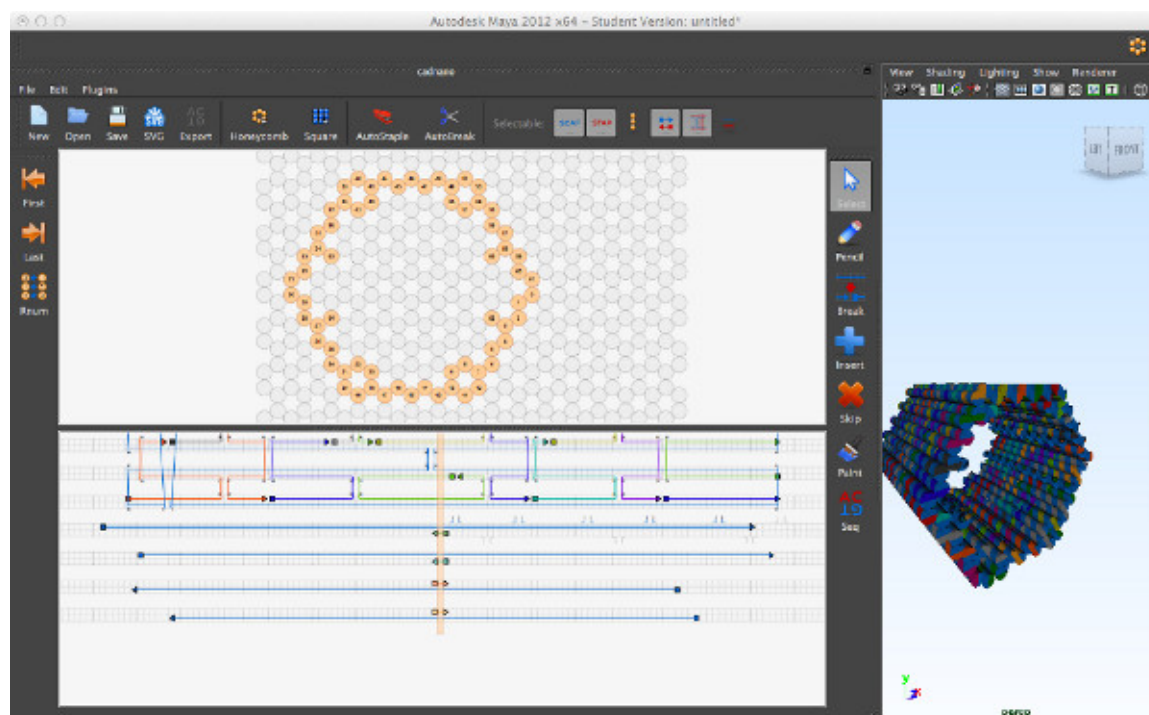


Figure 21. Manual addition of staple strands to the guide helices, so branching points can be located (see section 6.2). [Click here to view larger figure.](#)

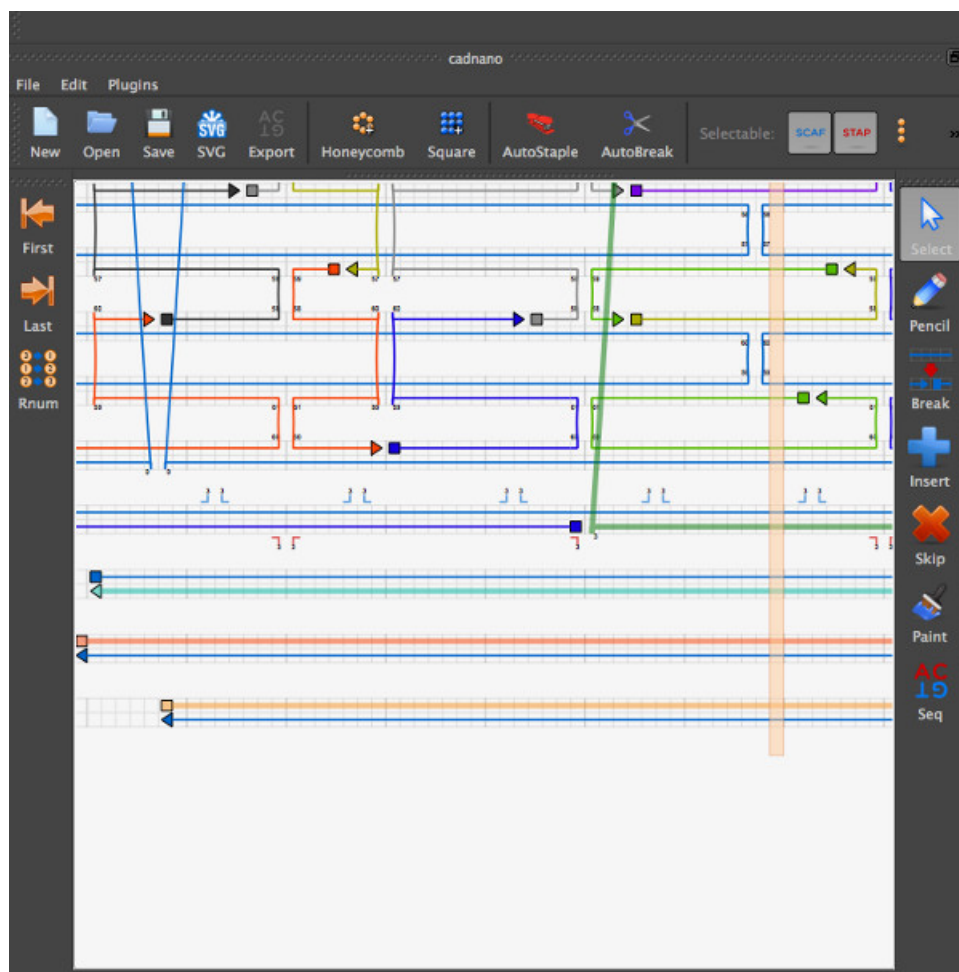


Figure 22. Introducing a loading site crossover to the robot chassis scaffold in a convenient location (one that requires minimal editing of chassis staples, see section 6.5).

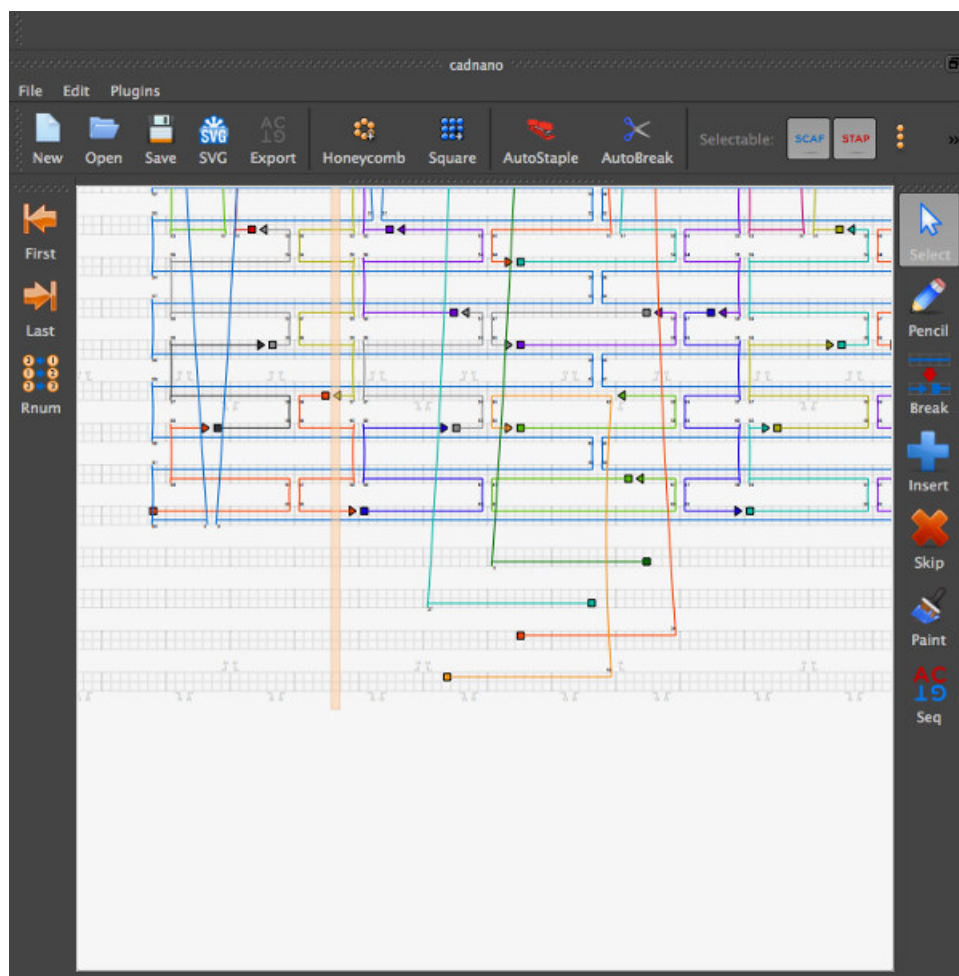


Figure 23. View of the loading site staples as seen in the bottom panel after removing the guide helices, which are no longer necessary (see section 6.9).

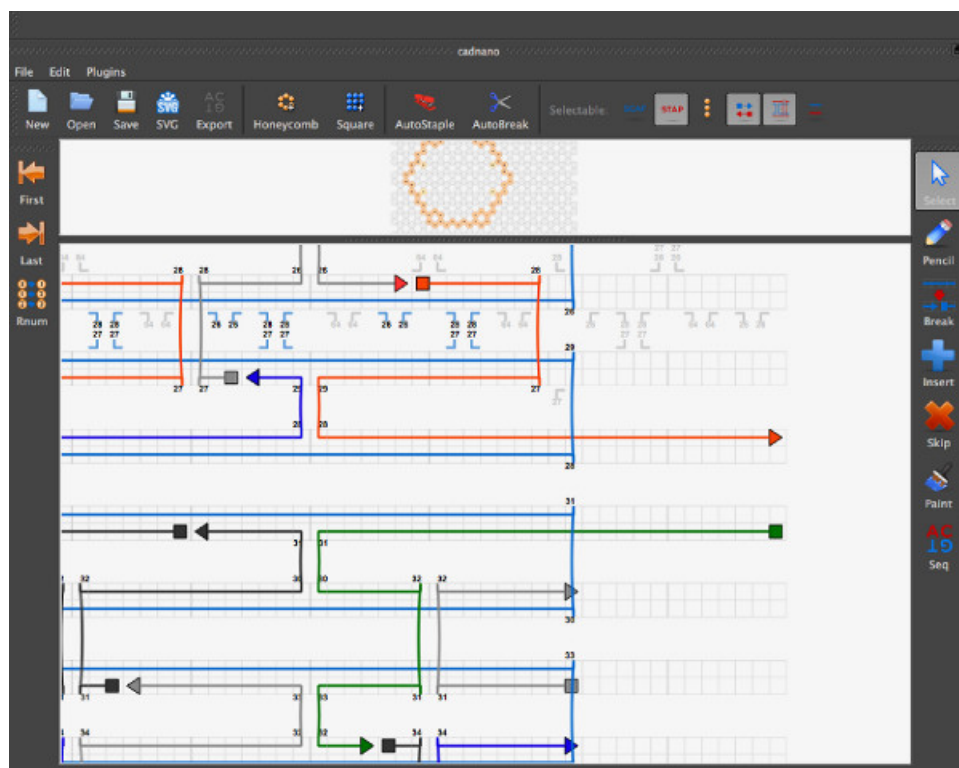


Figure 24. Extending two staples, which are going to be used as gate strands, from helices 29 and 30. Note that the two strands face opposite directions, which is mandatory for the formation of the gate duplex (see section 7.4). [Click here to view larger figure.](#)

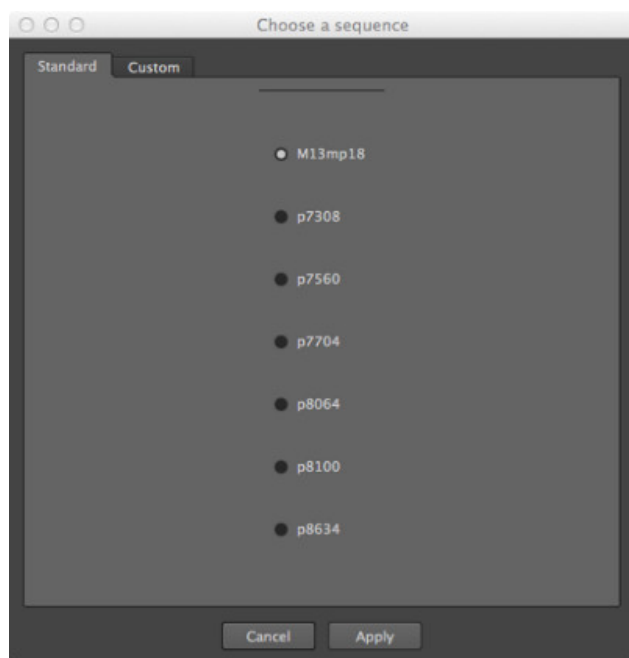


Figure 25. The scaffold sequence addition ("Seq" tool) dialogue box, allowing to choose either one of pre-defined scaffolds, or to insert a custom sequence (see section 8.1).

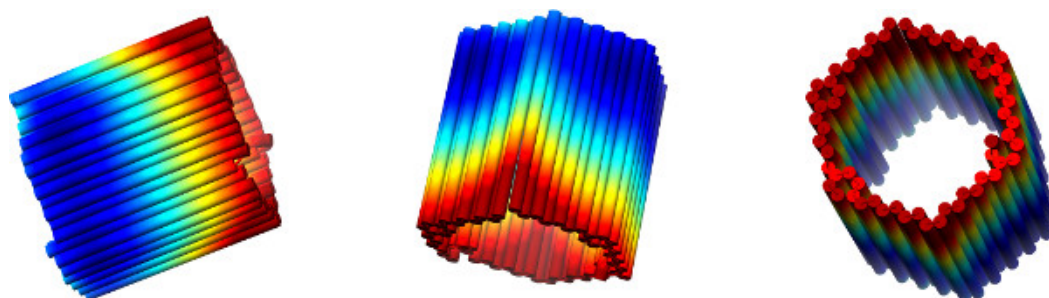


Figure 26. Results of CANDO analysis of the design described here. The simulation generates a .zip archive containing the various files that provide the requested information. Here the RMSF (root mean square fluctuation) files (.png) are depicted, showing a model of the design from 3 view angles, colored according to the key detailed in the accompanying file named "HeatMap4RMSF.txt". In this case, minimum RMSF (bluest) is 1.03 nm, and 95% RMSF (reddest) is 3.19 nm. The gradient of color across the model derives from the polarity of the robot (gates in 'front', axis in 'back') and the fact that there are no connecting staples along helices 29-30 and 61-0, causing the 'front' side to fluctuate more than the 'back' side.

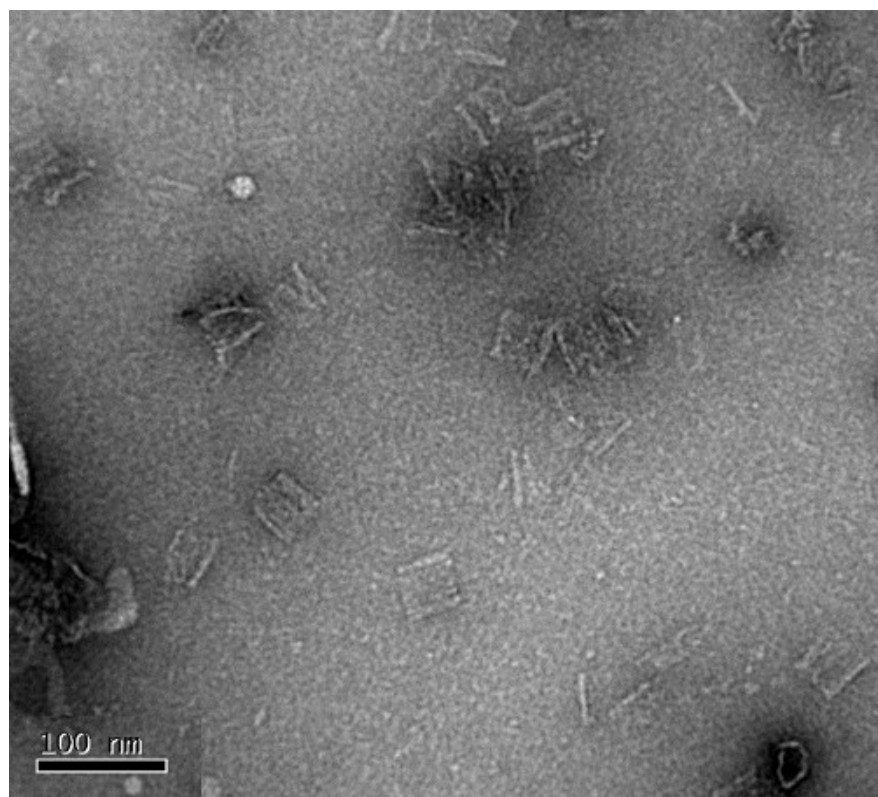


Figure 27. TEM image of the robot designed in this article. Sample preparation and staining was carried out exactly as described elsewhere²¹.

Discussion

DNA origami enables us to fabricate accurately defined objects with arbitrary features at the nanoscale. An important next step would be the integration of function into these designs. While many applications and challenges could be addressed with this technology, there is a particular interest in fabricating therapeutic and scientific robots from DNA origami, as these represent a natural milieu of DNA. DNA already interfaces with molecular machinery in cells as a genetic information storage medium. Interestingly, the folded DNA in a nanorobot or another machine can still serve as genetic information in addition to being a construction material, which can be relayed to the expression of a desired protein after the nanorobot disintegrates, as parts of a sequence of outputs.

In the example discussed in this paper, we use a restriction enzyme to operate the robot. However, additional mechanisms by which DNA robots can respond to inputs include the following.

Molecular recognition: we recently demonstrated aptamer-based gates for DNA robots that recognize protein molecules on the surface on target cells²⁰. Aptamers can be selected *in-vitro* using methods such as SELEX²³, outsourced from companies, or used from the aptamer database (<http://aptamer.icmb.utexas.edu/>). When aptamers are employed, it is important to consider that the strand complementary to the aptamer, which together forms the gate, can be designed to include mismatches, which will facilitate binding of the aptamer's ligand and displacement of the

complementary strand. While the mechanism allowing this is unknown, the sensitivity and specificity of an aptamer-based gate can be tuned by increasing or decreasing the % of mismatch between the two strands, to get either a very stringent but inefficient gate, or a fast but leaky one.

Enzymatic cleavage: for this, the gates should be designed such that they contain the substrate of that enzyme. For example, a small peptide substrate of a protease can be tethered from both sides to the gate, which will keep the robot closed in the absence of the enzyme.

Remote control: a potential approach that has not been applied to DNA machines is using a gold nanocrystal antenna in a high-frequency electromagnetic field to induce dsDNA melting²⁴. This may provide a user-operated switch in addition to bio-responsive ones. Although DNA origami robots are relatively straightforward to design and make, they pose several technical challenges as a therapeutic platform. DNA is not an ideal material for drug delivery as it is highly vulnerable to cleavage by nucleases. Moreover, it might precipitate an immune response. A thorough study of the behavior of DNA origami objects in an organism is needed to define their fate and make sure they do not aggregate in tissues or integrate into the host genome.

In summary, we presented the use of caDNAno, a straightforward, robust CAD tool to designing DNA origami shapes. We hope to start seeing application-driven research in DNA origami, in areas such as therapeutics, energy, metamaterials, and education. In all these places, caDNAno is expected to have a significant impact on realizing the solutions. In the future, it might become an industrial and design standard, which can be replaced (or parts of which can) by any user because they are all compatible.

Disclosures

The authors have nothing to disclose.

Acknowledgements

The authors wish to thank S. Douglas for extremely valuable discussions and advice, and all the members of the Bachelet lab for helpful discussions and work. This work is supported by grants from the Faculty of Life Sciences and Institute of Nanotechnology & Advanced Materials at Bar-Ilan University.

References

1. Watson, J.D. & Crick, F.H. Genetical implications of the structure of deoxyribonucleic acid. *Nature*. **171**, 964-967 (1953).
2. Adleman, L.M. Molecular computation of solutions to combinatorial problems. *Science*. **266**, 1021-1024 (1994).
3. Qian, L., Winfree, E., & Bruck, J. Neural network computation with DNA strand displacement cascades. *Nature*. **475**, 368-372, doi:10.1038/nature10262 (2011).
4. Ellington, A.D. & Szostak, J.W. *In vitro* selection of RNA molecules that bind specific ligands. *Nature*. **346**, 818-822, doi:10.1038/346818a0 (1990).
5. Tang, Z., Parekh, P., Turner, P., Moyer, R.W., & Tan, W. Generating aptamers for recognition of virus-infected cells. *Clin. Chem.* **55**, 813-822, doi:10.1373/clinchem.2008.113514 (2009).
6. Baskerville, S. & Bartel, D.P. A ribozyme that ligates RNA to protein. *Proc. Natl. Acad. Sci. U.S.A.* **99**, 9154-9159, doi:10.1073/pnas.142153799 (2002).
7. Bartel, D.P. & Szostak, J.W. Isolation of new ribozymes from a large pool of random sequences [see comment]. *Science*. **261**, 1411-1418 (1993).
8. Benenson, Y., Gil, B., Ben-Dor, U., Adar, R., & Shapiro, E. An autonomous molecular computer for logical control of gene expression. *Nature*. **429**, 423-429, doi:10.1038/nature02551 (2004).
9. Xie, Z., Wroblewska, L., Prochazka, L., Weiss, R., & Benenson, Y. Multi-input RNAi-based logic circuit for identification of specific cancer cells. *Science*. **333**, 1307-1311, doi:10.1126/science.1205527 (2011).
10. Rothmund, P.W., Papadakis, N., & Winfree, E. Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biol.* **2**, e424, doi:10.1371/journal.pbio.0020424 (2004).
11. Chen, J.H. & Seeman, N.C. Synthesis from DNA of a molecule with the connectivity of a cube. *Nature*. **350**, 631-633, doi:10.1038/350631a0 (1991).
12. He, Y., *et al.* Hierarchical self-assembly of DNA into symmetric supramolecular polyhedra. *Nature*. **452**, 198-201, doi:10.1038/nature06597 (2008).
13. Seeman, N.C. Nucleic acid junctions and lattices. *J. Theor. Biol.* **99**, 237-247 (1982).
14. Wei, B., Dai, M., & Yin, P. Complex shapes self-assembled from single-stranded DNA tiles. *Nature*. **485**, 623-626, doi:10.1038/nature11075 (2012).
15. Yin, P., *et al.* Programming DNA tube circumferences. *Science*. **321**, 824-826, doi:10.1126/science.1157312 (2008).
16. Rothmund, P.W. Folding DNA to create nanoscale shapes and patterns. *Nature*. **440**, 297-302, doi:10.1038/nature04586 (2006).
17. Dietz, H., Douglas, S.M., & Shih, W.M. Folding DNA into twisted and curved nanoscale shapes. *Science*. **325**, 725-730, doi:10.1126/science.1174251 (2009).
18. Douglas, S.M., *et al.* Self-assembly of DNA into nanoscale three-dimensional shapes. *Nature*. **459**, 414-418, doi:10.1038/nature08016 (2009).
19. Douglas, S.M., *et al.* Rapid prototyping of 3D DNA-origami shapes with caDNAno. *Nucleic Acids Res.* **37**, 5001-5006, doi:10.1093/nar/gkp436 (2009).
20. Douglas, S.M., Bachelet, I., & Church, G.M. A logic-gated nanorobot for targeted transport of molecular payloads. *Science*. **335**, 831-834, doi:10.1126/science.1214081 (2012).
21. Castro, C.E., *et al.* A primer to scaffolded DNA origami. *Nature Methods*. **8**, 221-229, doi:10.1038/nmeth.1570 (2011).

22. Ke, Y., *et al.* Multilayer DNA origami packed on a square lattice. *Journal of the American Chemical Society*. **131**, 15903-15908, doi:10.1021/ja906381y (2009).
23. Mallikaratchy, P., *et al.* Using aptamers evolved from cell-SELEX to engineer a molecular delivery platform. *Chem. Commun. (Camb)*. 3056-3058, doi:10.1039/b823258j (2009).
24. Hamad-Schifferli, K., Schwartz, J.J., Santos, A.T., Zhang, S., & Jacobson, J.M. Remote electronic control of DNA hybridization through inductive coupling to an attached metal nanocrystal antenna. *Nature*. **415**, 152-155, doi:10.1038/415152a (2002).