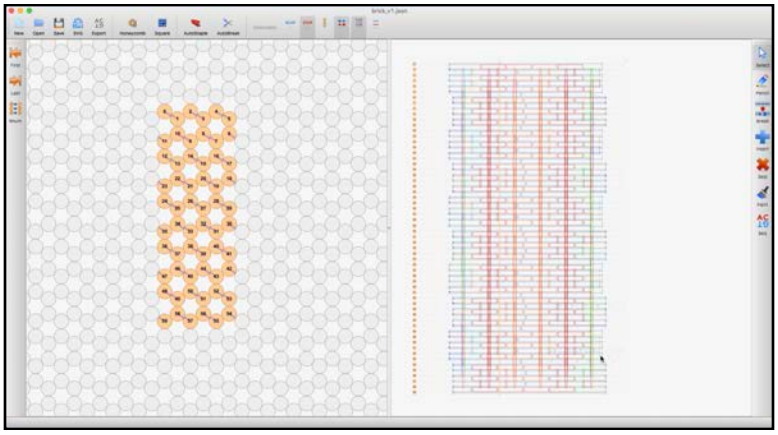
Design 1: Honeycomb ‘10x6’ Brick

Introduction

In this section we describe how to create a DNA origami “brick” structure. The simplicity of the shape is intentional, so we can quickly gain exposure to the entire **CTK** workflow before delving into more subtle aspects of designing more complex shapes.



The initial design steps are similar to those outlined in the original Cadnano paper[1]. We first route a scaffold to approximate a desired shape. The interface tools are used to insert short stretches of scaffold DNA strands, which are then edited to form one contiguous scaffold path whose final length should be equal to a corresponding scaffold DNA that will be used in the lab. Once the scaffold route is set, the “AutoStaple” tool is used to automatically generate a set of staple paths that are complementary to the scaffold strands, and include “crossovers” between proximal neighboring helices.

Our original design approach would normally continue at this step by manually “breaking” the staple paths into short oligonucleotides, typically 20–50 bases in length. Cadnano renders staples with lengths outside this range with thicker lines to facilitate the process of systematically working through breaking the staples in the entire design. Decisions for where to break the staple paths were made according to mostly *ad hoc* rules or heuristics derived from limited empirical data[2]. Once the scaffold and staple routes were finalized, a DNA sequence is applied to the scaffold at a chosen start position. When the scaffold sequence is applied, the staples inherit the complementary base at each location that they pair with the scaffold. Staple sequences can then be exported for synthesis.

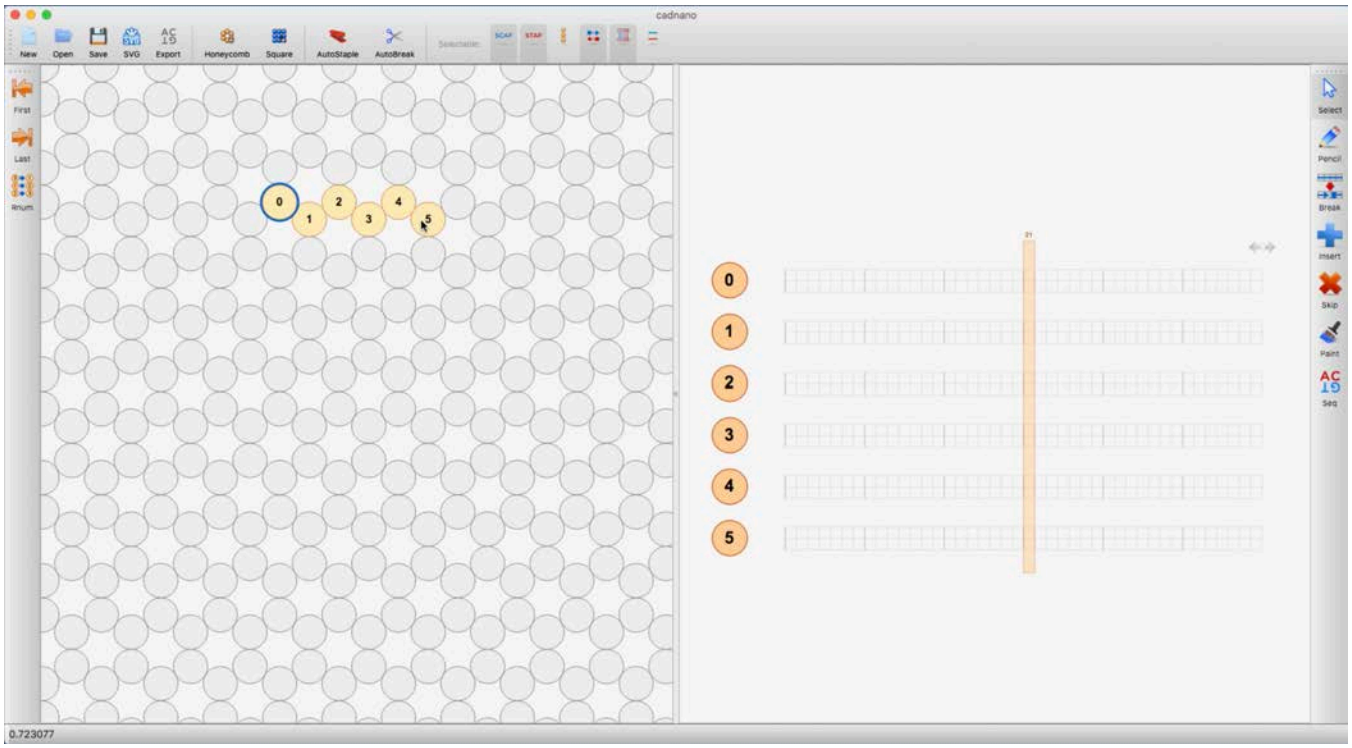
In this tutorial we introduce an improved workflow for staple breaking that uses the “AutoBreak” tool in the Cadnano Toolkit (CTK). AutoBreak attempts to find globally optimized staple breaking solutions based on a simple thermodynamic model. It also carries out the steps of sequence addition, and automatically exports the staple sequences. We also demonstrate the basic use of the structure prediction tools in CTK. We will generate a coarse-grained model of the DNA brick structures and compare them. The resulting 3D models can be useful for making figures and animations. In future tutorials, we will show how the tools can help in debugging multi-domain shapes, and aid in visualizing functional attachments, such as proteins or nanoparticles.

---

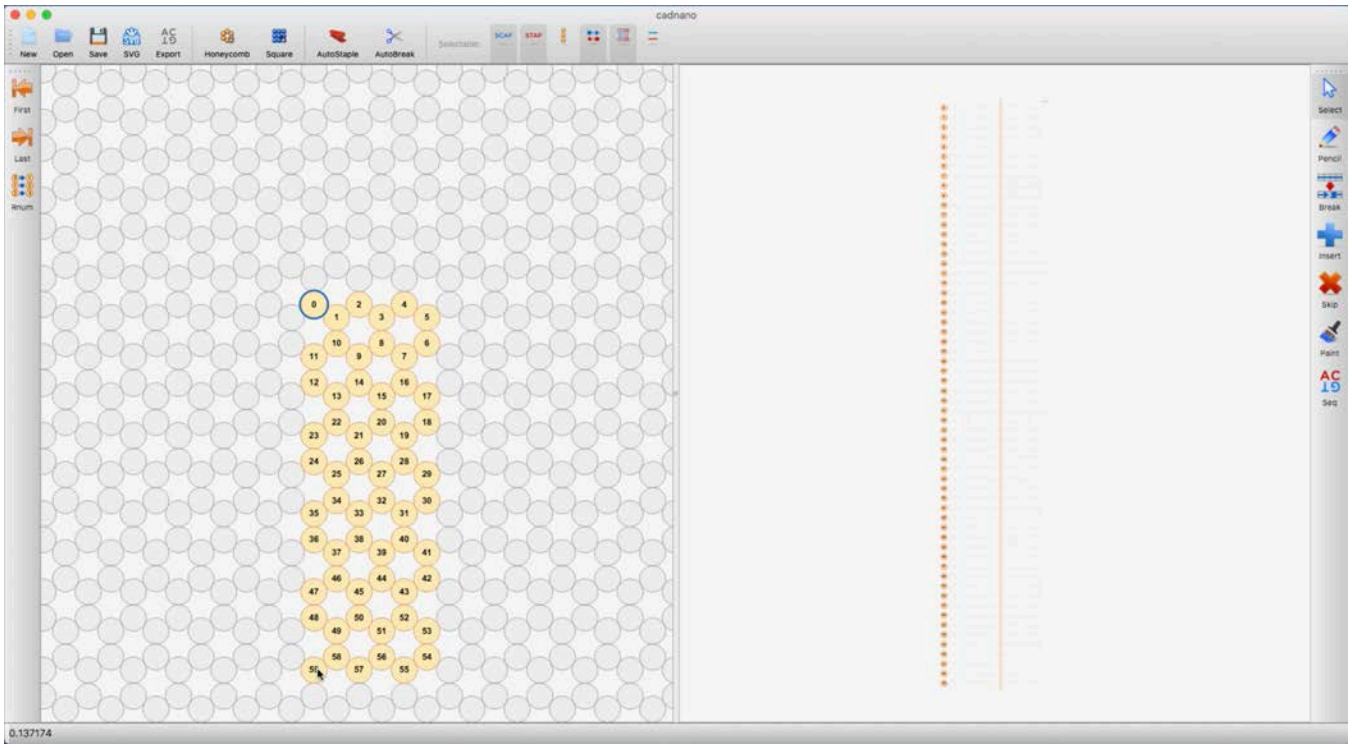
[1] Douglas *et al.* 2009. *Nucleic Acids Res.* 37(15):5001–6. doi.org/10.1093/nar/gkp436

[2] Martin and Dietz, 2012. *Nat Commun.* 3:1103. doi.org/10.1038/ncomms2095

Step 1: Define the cross-section



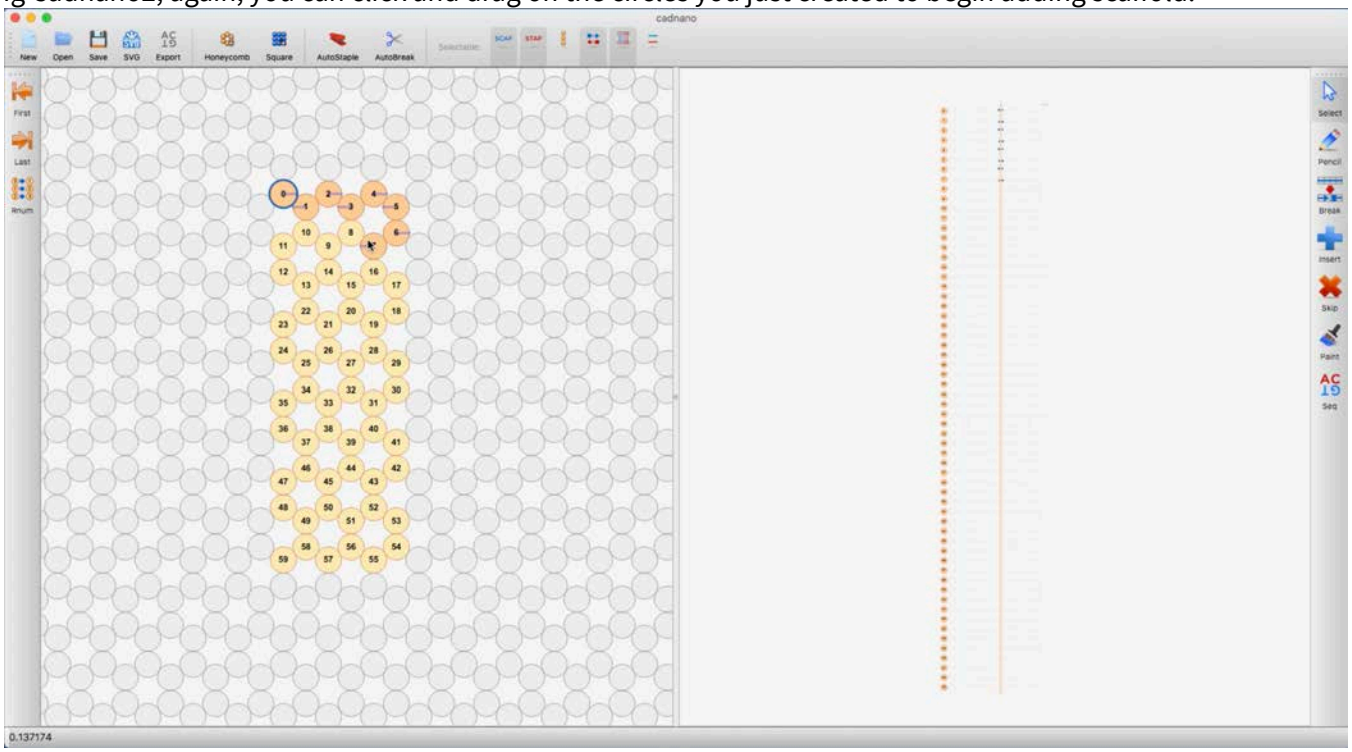
Using Cadnano1, you have to click each circle one at a time to add a “virtual helix” to your design. In Cadnano2, you can click and drag in the Slice interface to add several at once.



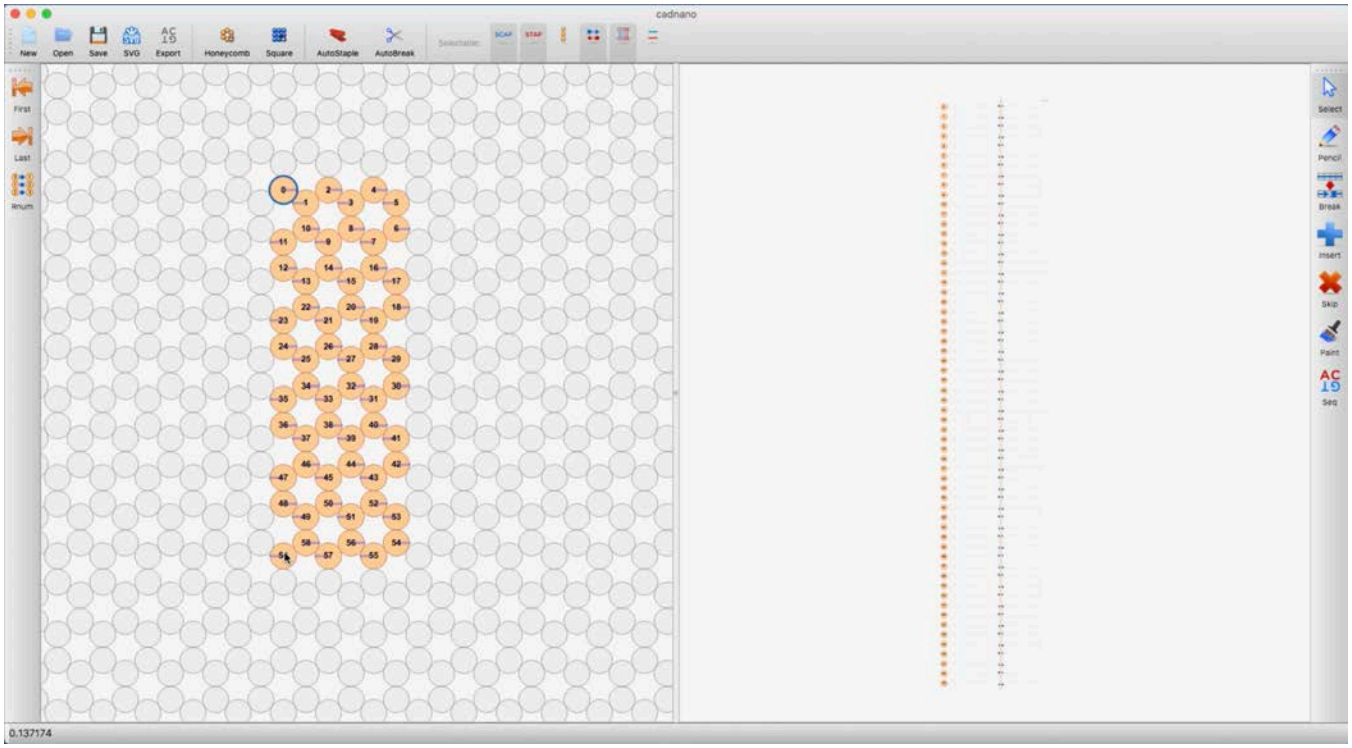
This is a good save point to avoid duplicating work if the next step doesn’t work on the first try.

Step 2: Add Scaffold to your data structure

Using Cadnano2, again, you can click and drag on the circles you just created to begin adding scaffold.

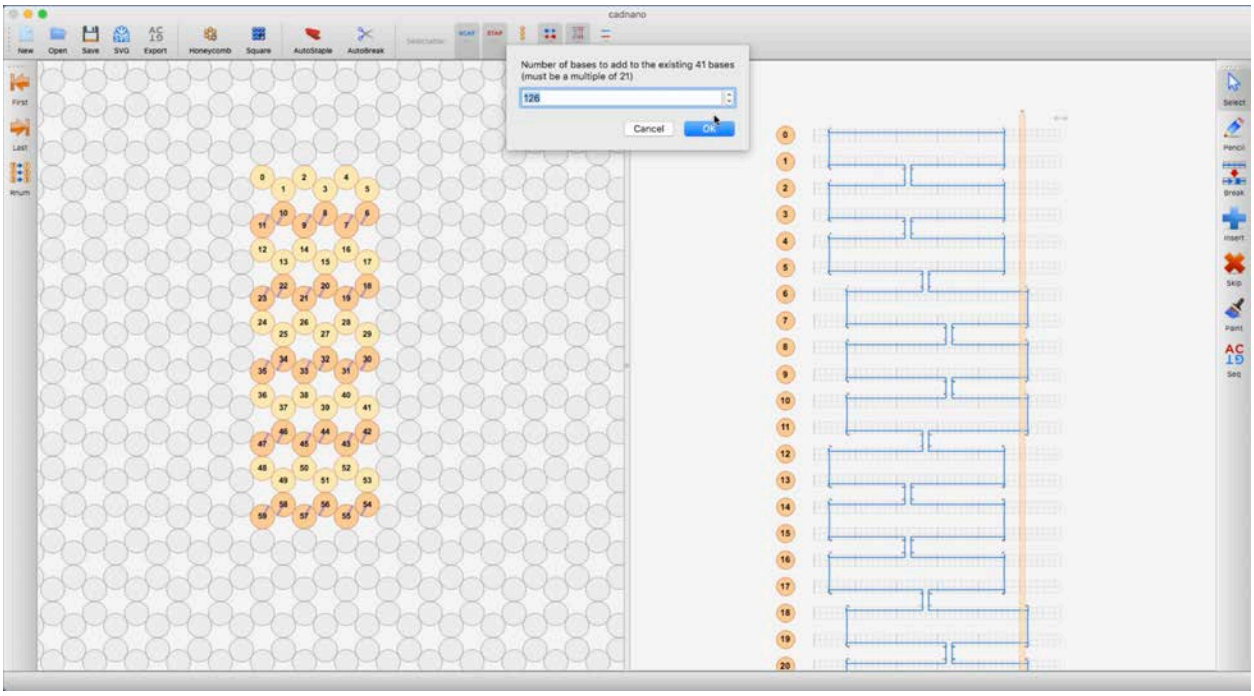


If you can do this in a single mouse-press from start to finish, Cadnano will automatically expand the scaffold and connect it with crossovers. In Cadnano1, this needs to be done manually.



The above image shows just before the mouse release. If this step didn't quite work, you can either undo, or revert to a saved version of the file.

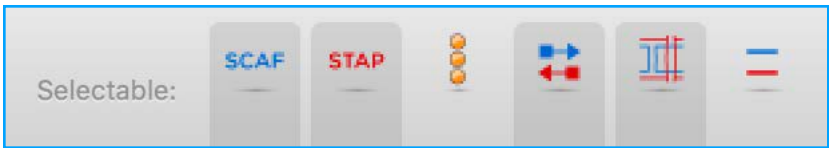
Once we are satisfied with the initial scaffold route, we can move over to the Path panel (right side) and start to expand the shape to match our scaffold size. We have 60 helices, and are aiming to use the 7560 scaffold, or lengths of 126 bases. We prefer some “buffer” space on either side of the scaffold, so we usually expand a bit larger than the minimum needed region. Click on the right arrow at the very top-right corner of the design. It will open a dialog box to extend the canvas.



Once we have expanded the data structure to allow for longer virtual helix lengths in the design, we can use the **Select Tool** click and drag the edges of the scaffold route to extrude the scaffold.

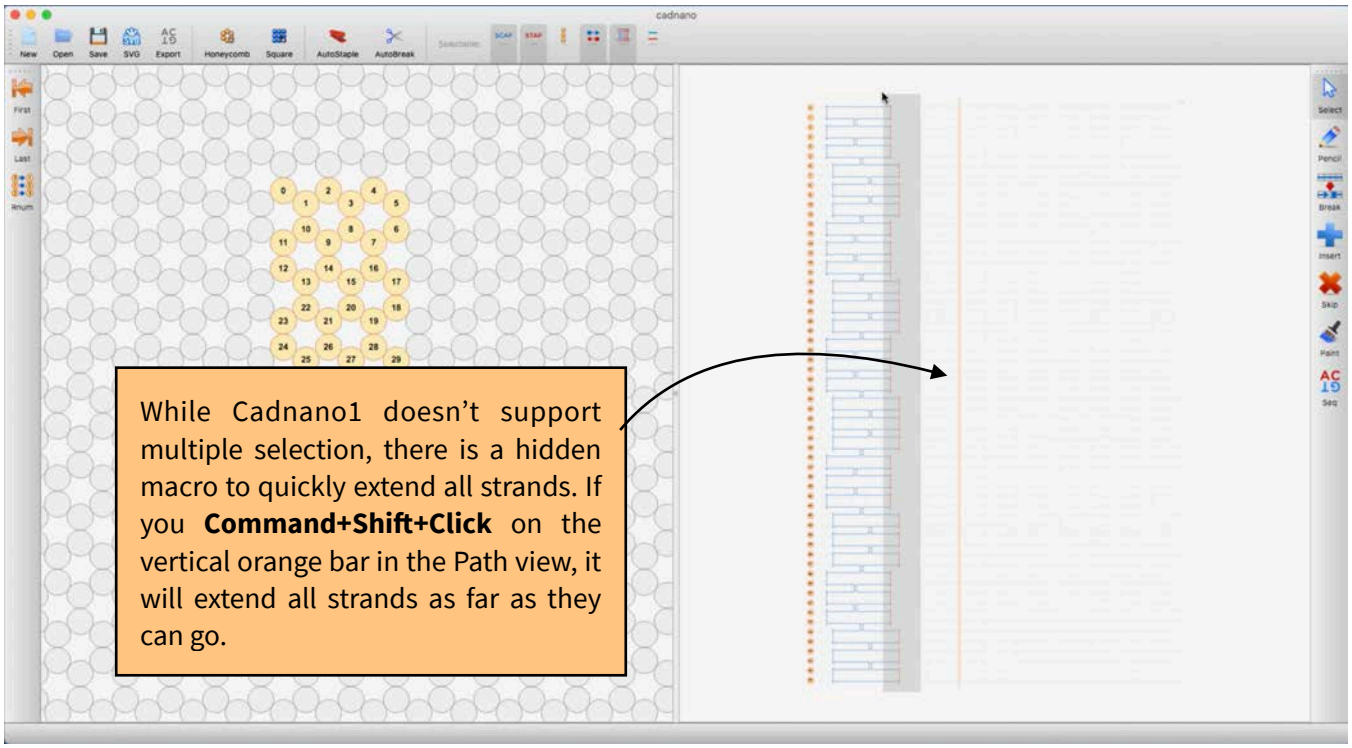
Note about Selection Filters

In order to allow precise selections, we also have added selection filters. If you ever try to click in the interface and nothing happens, you probably need to change the active tool and/or selection filters. The buttons that toggle filters on and off are located on the top menu bar:

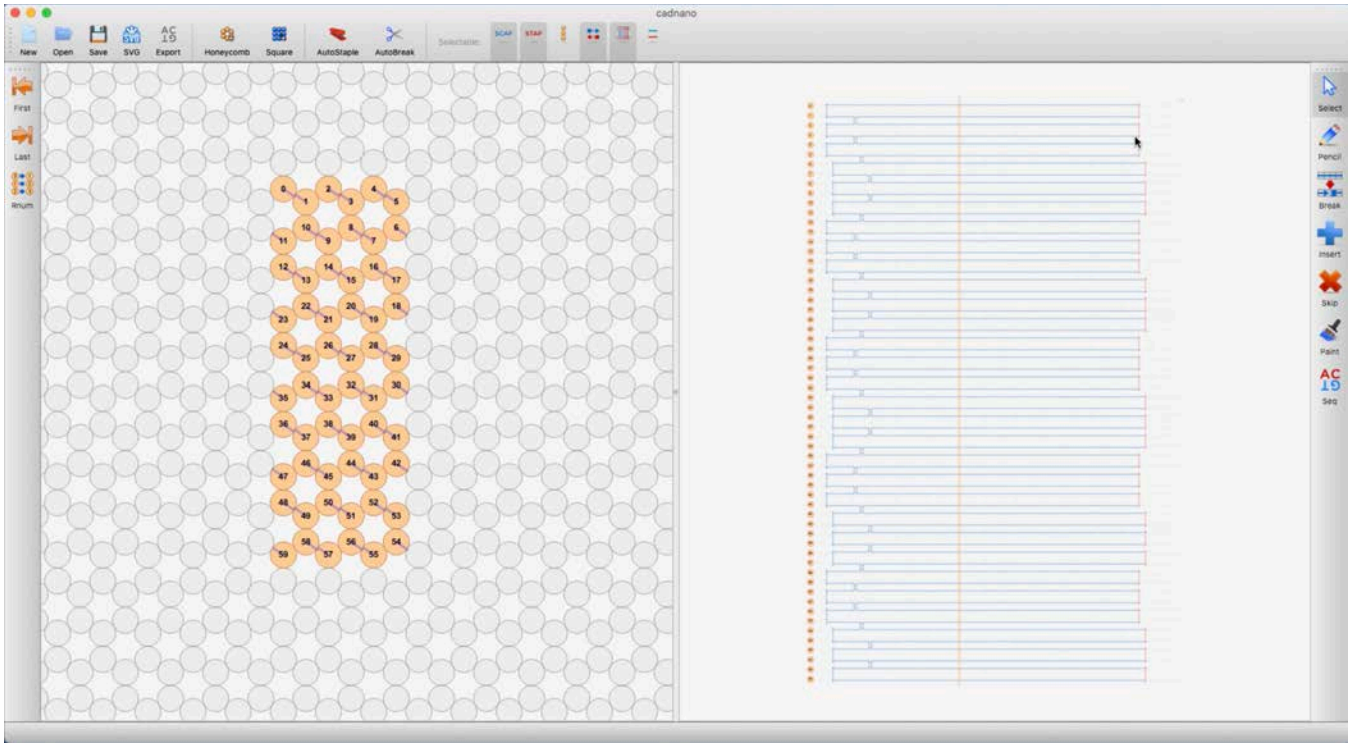


The six filters (and keyboard shortcuts) are: scaffold (C), staple (T), handle (H), endpoint (E), crossover (X), and strand (D). The scaffold & staple buttons filter strand type. At least one must be enabled, so for example to disable scaffold selection, staple selection must first be turned on. The **Handle** button allows selection of the handles on the left side of each helix in the path view, and can be used to reorder the virtual helices. This is useful when strand crossovers overlap and you need to reach something that is obscured. The filter is exclusive with the other item filters (endpoint, xover, strand). The **endpoint** filter allows selection of 5' and 3' ends. If both endpoints of a strand are selected, the strand is considered selected as well (for display). The **crossover** (or **xover**) filter allows selection of crossovers. The

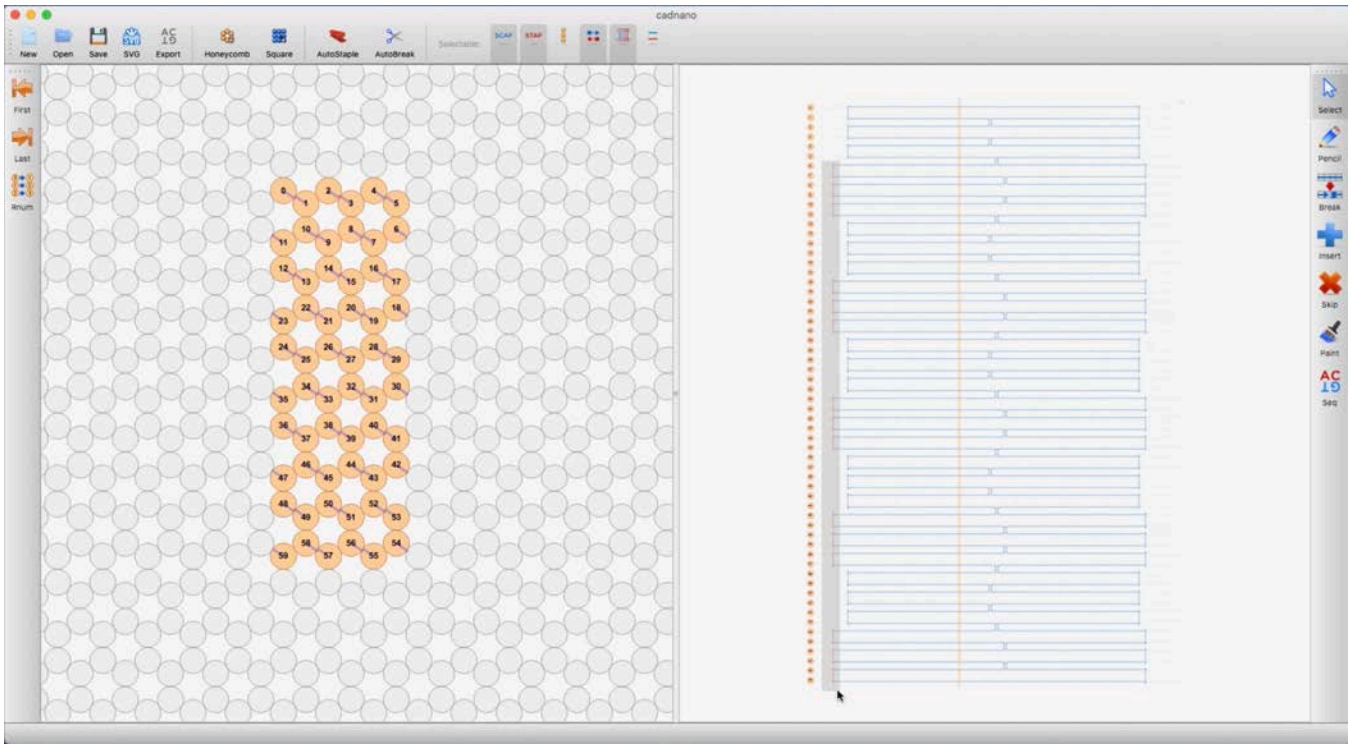
strand filter selects entire strands (including their endpoints & xovers). Most editing can be done without the strand filter, but it is included for completeness.



Once selected, the crossovers will turn pink. You can now click and drag the selected region to resize everything at once. Watch the "status bar" at the bottom left of the window to see the number of bases you are expanding.

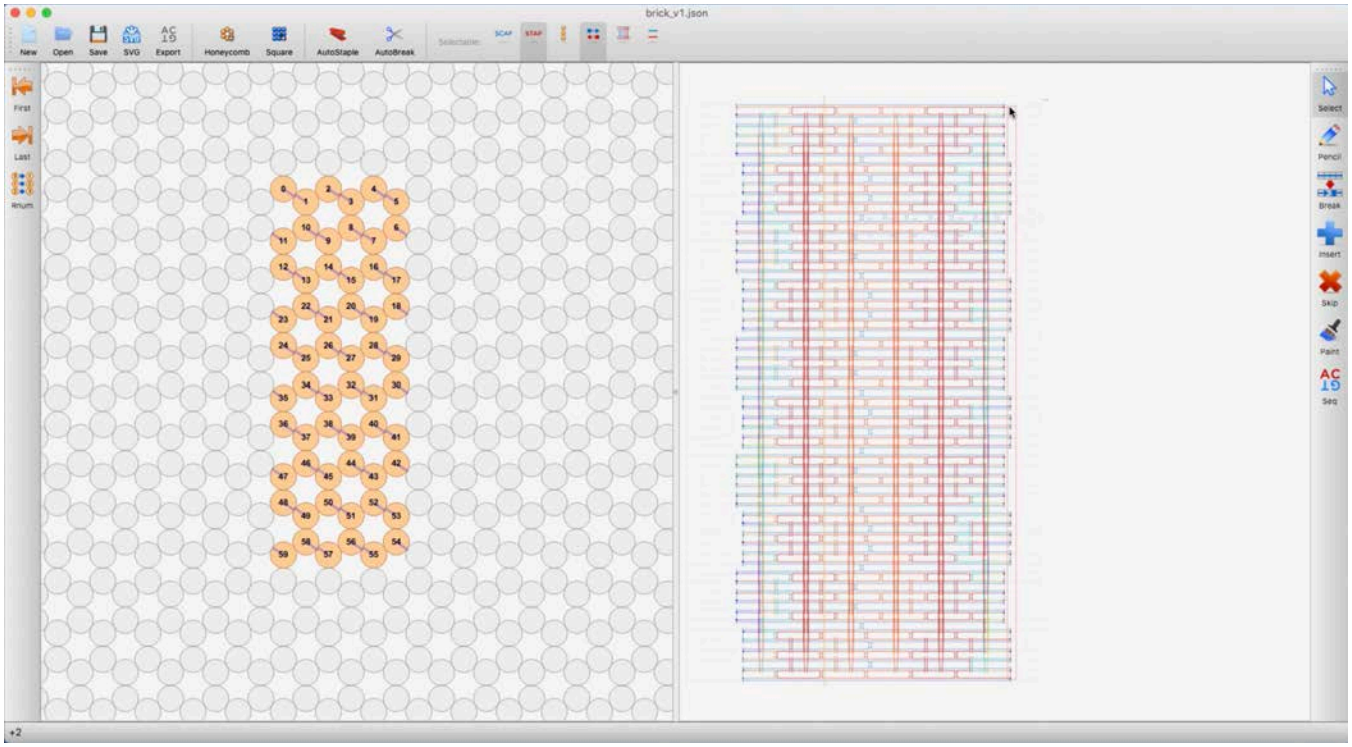


We're getting pretty close to the final scaffold layout. Next we use the **Select Tool** to select and drag the internal seem to the middle of the design.

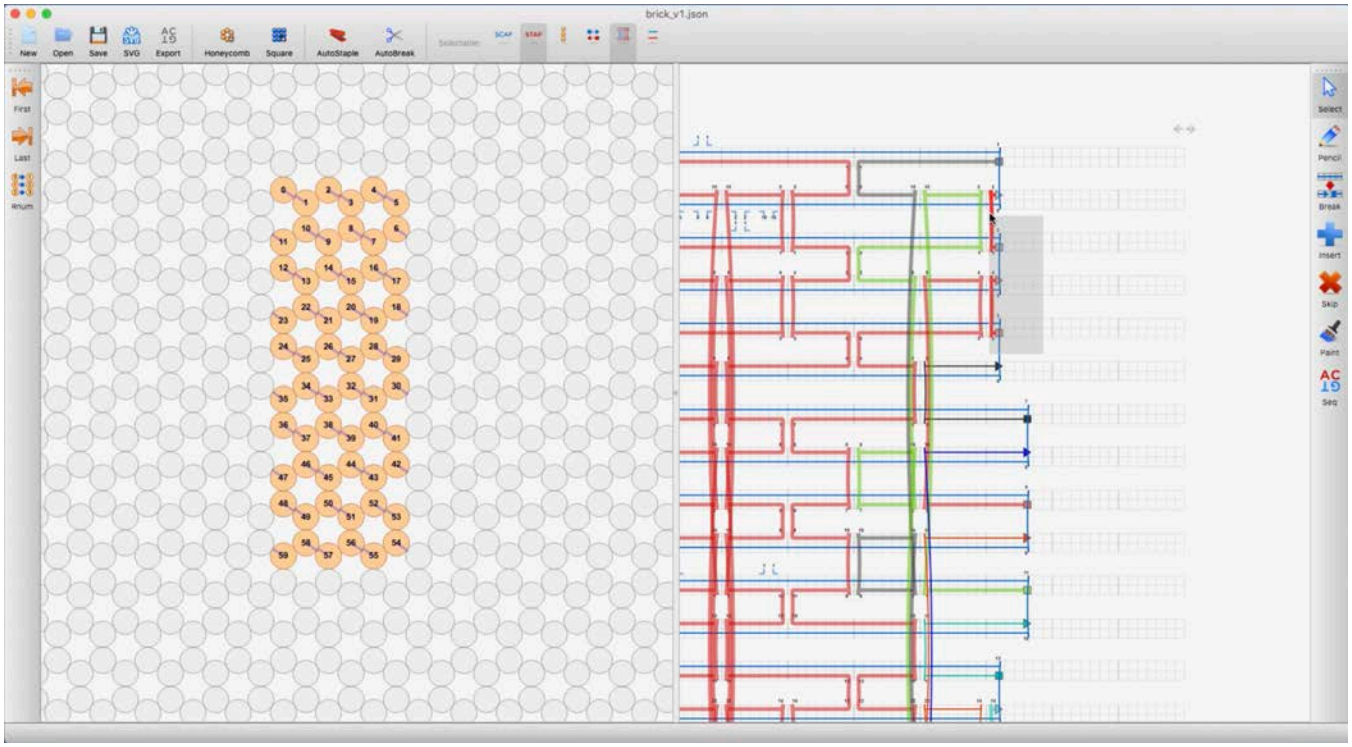


We next mouse over the scaffold and check its total length in the status bar. If it's not 7560, then we need to resize the scaffold again. If we overextended, then we can shorten by resizing at the left edge.

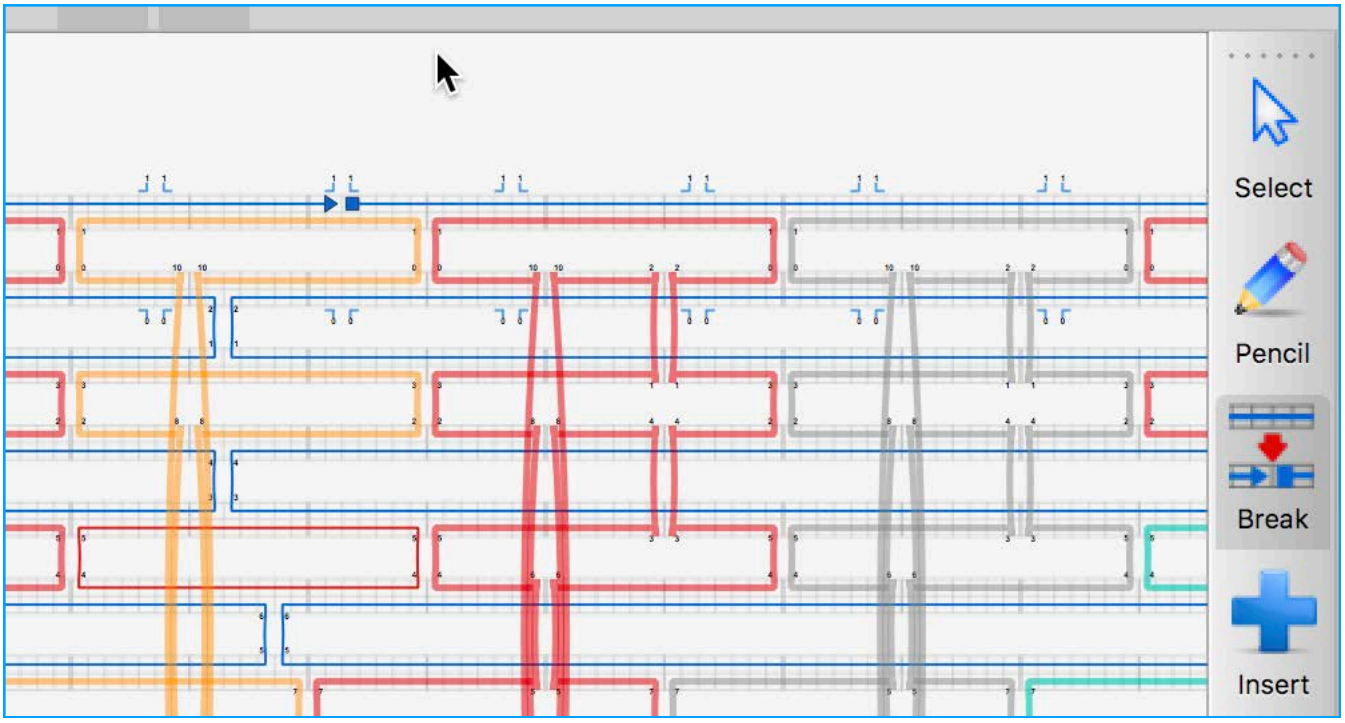
Step 3: Add proto-staples to your design



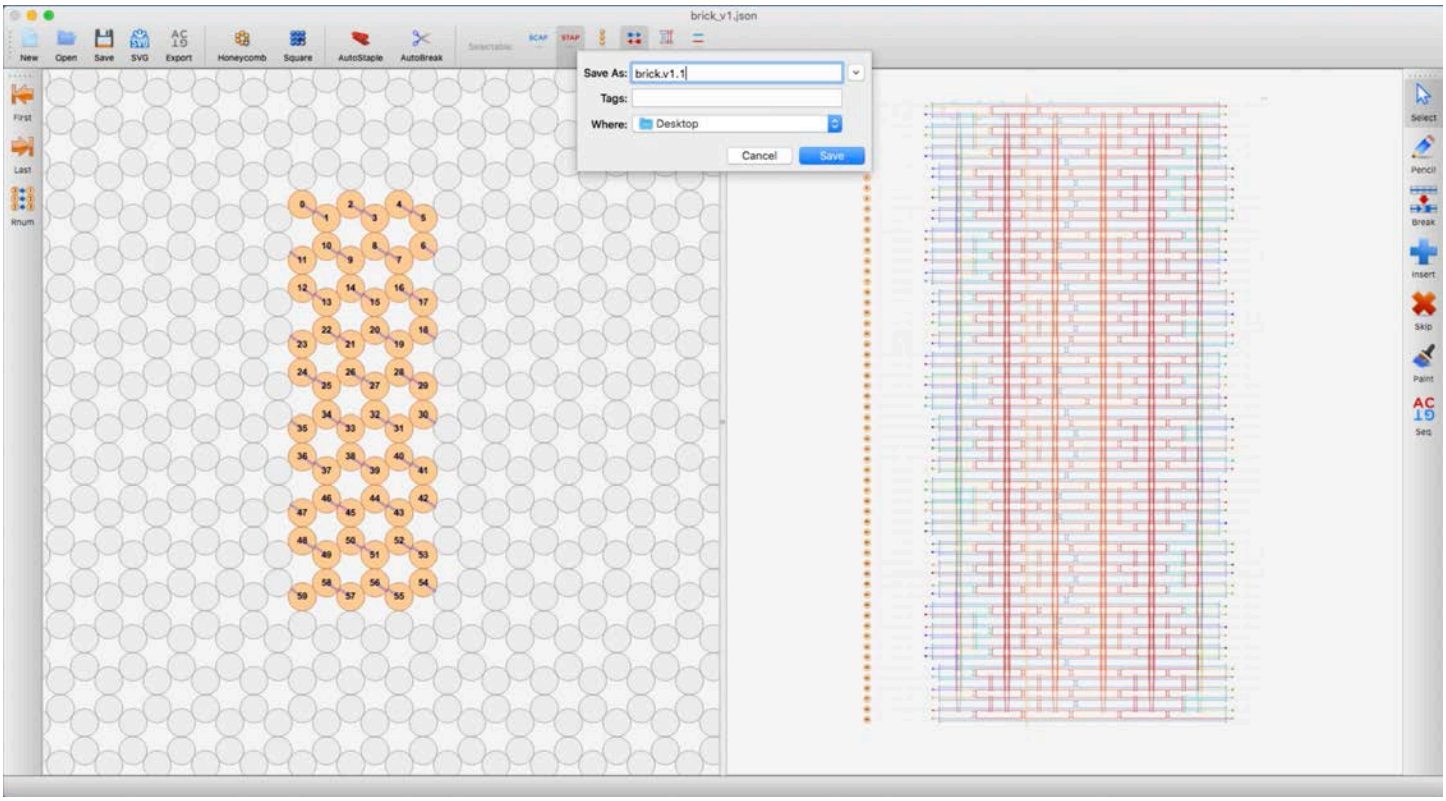
Once satisfied with the scaffold route, click the "AutoStaple" button. After a few seconds, a full set of "proto staples" should appear, or unbroken staples that will need further processing. We usually need to clean them up a bit, since the algorithm includes a few crossovers that we probably don't want in our design.



If we notice any super-short staples near the edges, we delete those crossovers and shift-click on the adjacent endpoints to seal the nicks. We repeat this step for the entire edge.



Before saving the design, we introduce a break in the scaffold using the **Break** tool. This will make the **CTK AutoBreak** tool run faster since it will not try to explore all the scaffold permutations. In practice, we do not notice a big difference in final scores.



Finally, we save at this point since we're ready to pass the structure over to **AutoBreak** and **OrigamiSim** tools.

We perform a poor-man's version control by saving each cadnano design file with an incrementing number at the end of the filename, e.g. brick.v1.0, brick.v1.1, etc. That way, if you make a mistake, you can always go back to a previous version.

Save or copy the file into your **Dropbox** swap folder.

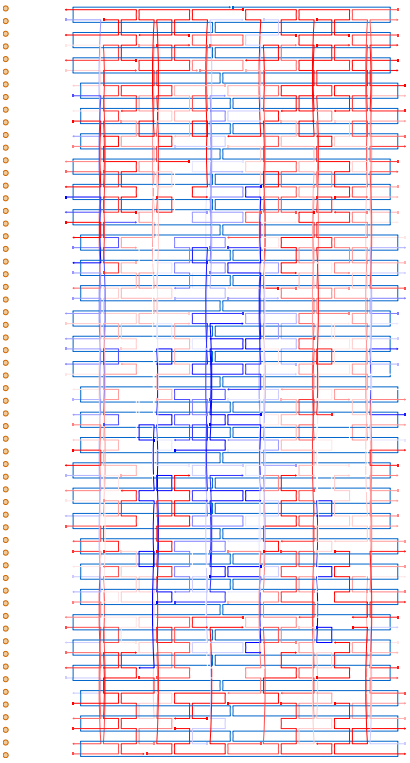
Step 4: Run AutoBreak

Switch over to your AMI VNC window. Open a **Terminal** window and navigate to your swap folder (e.g., `cd ~/Dropbox/AMI-swap`).

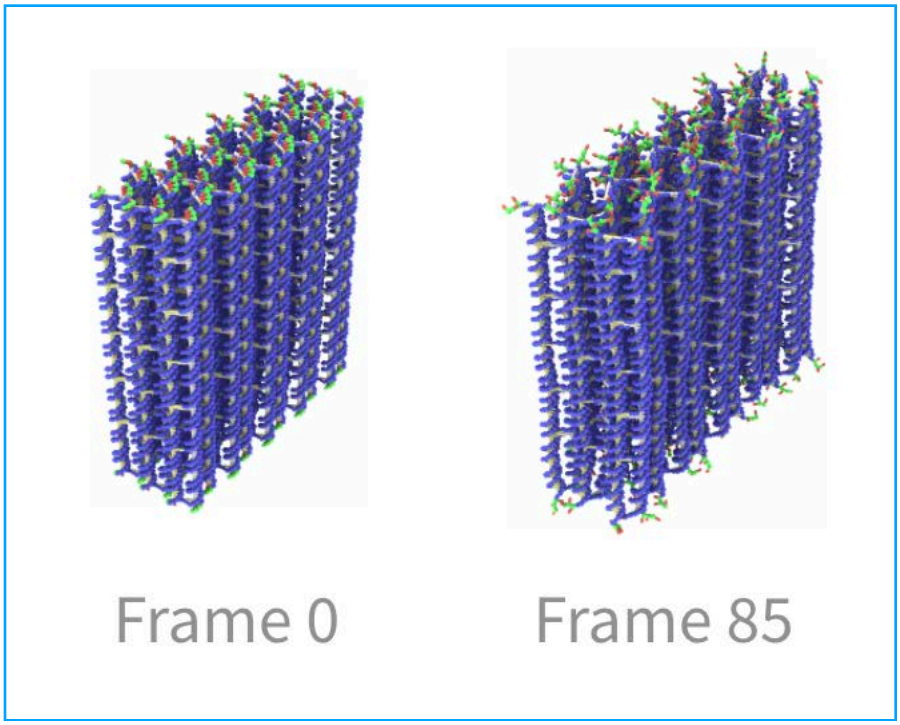
Run the **AutoBreak** script using the following command:

```
autobreak.py -i brick.v1.1 -rule xstap.all3 -seq p8064
```

The script takes a few moments to start up, and then a progress bar will appear. It may take 5–10 minutes to complete. It will create a new folder for each run based on the design name, and an incrementing counter (001, 002, etc). Once the run is complete, you can open the output design in Cadnano, and look for **blue** staples that are predicted to fold at a lower temperature. The design can be edited to optimized to increase the total number of **red** staples. The total score can be found in the Excel output, e.g. "brick_v1.1_summary.xlsx", and a lower total score is better.



Example output from AutoBreak



Example output from OrigamiSim

Step 5: Run OrigamiSim

Once **AutoBreak** is complete, you can use the **Terminal** to run a quick structure prediction with **OrigamiSim**. First, navigate into the AutoBreak output folder:

```
cd brick_v1.1_autobreak_001
```

Then run the following command:

```
origamisim.py -i brick_v1.1_autobreak_legacy.json -vmd
```

The -vmd flag will spawn a VMD window so you can view the progress of the simulation in realtime.

Tural Aksel will explain how AutoBreak and OrigamiSim work in detail during **Day 2** of the workshop.