

Design, optimization and analysis of large DNA and RNA nanostructures through interactive visualization, editing and molecular simulation

Erik Poppleton¹, Joakim Bohlin¹, Michael Matthies¹, Shuchi Sharma¹, Fei Zhang^{1,3} and Petr Šulc^{1,*}

¹School of Molecular Sciences and Center for Molecular Design and Biomimetics, The Biodesign Institute, Arizona State University, 1001 South McAllister Avenue, Tempe, AZ 85281, USA, ²Department of Physics, Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, UK and ³Department of Chemistry, Rutgers University-Newark, 73 Warren St, Newark, NJ 07102, USA

Received January 27, 2020; Revised April 22, 2020; Editorial Decision May 06, 2020; Accepted May 07, 2020

Downloaded from https://academic.oup.com/nar/article/48/12/e72/5843822 by guest on 06 February 2021

ABSTRACT

This work seeks to remedy two deficiencies in the current nucleic acid nanotechnology software environment: the lack of both a fast and user-friendly visualization tool and a standard for structural analyses of simulated systems. We introduce here oxView, a web browser-based visualizer that can load structures with over 1 million nucleotides, create videos from simulation trajectories, and allow users to perform basic edits to DNA and RNA designs. We additionally introduce open-source software tools for extracting common structural parameters to characterize large DNA/RNA nanostructures simulated using the coarse-grained modeling tool, oxDNA, which has grown in popularity in recent years and is frequently used to prototype new nucleic acid nanostructural designs, model biophysics of DNA/RNA processes, and rationalize experimental results. The newly introduced software tools facilitate the computational characterization of DNA/RNA designs by providing multiple analysis scripts, including mean structures and structure flexibility characterization, hydrogen bond fraying, and interduplex angles. The output of these tools can be loaded into oxView, allowing users to interact with the simulated structure in a 3D graphical environment and modify the structures to achieve the required properties. We demonstrate these newly developed tools by applying them to design and analysis of a range of DNA/RNA nanostructures.

INTRODUCTION

The field of nucleic acid nanotechnology (1) uses DNA and RNA as building blocks to construct nanoscale struc-

tures and devices. Using the high programmability of pairing combinations between oligonucleotides, it is possible to construct 2D and 3D nanostructures up to several thousand nucleotides. Over the past three decades, designs of increasing complexity have been proposed, such as DNA/RNA tiles and arrays (2), DNA multi-bundle origamis (3), wireframe nanostructures (4,5) single-stranded tile (SST) nanostructures (6), single-stranded DNA (ssDNA) and RNA (ssRNA) origami structures (7), and larger multi-origami tile assemblies (8). The nanostructures have promising applications ranging from photonic devices (9) to drug delivery (10).

There are many available nucleic acid nanotechnology design tools, including CaDNAno (11), Tiamat (12), vHelix (13,14), Adenita (15), MagicDNA (Huang *et al.*, in preparation) and the CAD converters DAEDALUS (16) and PERDIX (17). CaDNAno is frequently used to design very large structures on either a square or hexagonal lattice, which requires components be made of parallel helices. Tiamat is an intuitive lattice-free design tool that supports both DNA and RNA. MagicDNA is a Matlab-based tool that specializes in the design of large 3D structural components on a 3D cubic lattice using CaDNAno-like parallel DNA bundles as the base unit of each edge. VHelix and Adenita are DNA design plugins for the commercial design platforms Maya and SAMSON. VHelix facilitates conversion of polyhedral meshes to DNA sequences, with further free-form editing available in Maya. Adenita combines the functionality of CAD converters with free-form design, allowing users to load structures from a variety of sources with additional editing tools available in the SAMSON interface. DAEDALUS and PERDIX are software that facilitate conversion of meshes designed in CAD software into DNA representations. Currently, the nanotechnology field lacks a universal method for assembling structures made in different design tools, especially if small changes need to be made.

*To whom correspondence should be addressed. Tel: +1 480 965 9601; Email: psulc@asu.edu

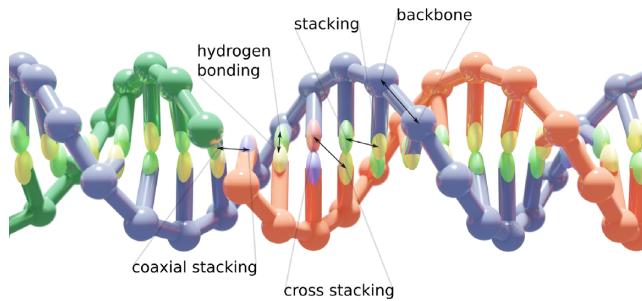


Figure 1. The oxDNA model. A DNA duplex as modeled in oxDNA with labels corresponding to the potentials defining the force field. oxDNA is a coarse-grained model with each nucleotide represented as a rigid body with specific interaction sites that approximate the geometry and interactions of the 20+ atoms that make up each nucleotide. The coarse-grained force field is parameterized to reconstruct the structural and dynamic properties of both single- and double-stranded DNA and RNA.

Continued development of tools is thus necessary to integrate the previous efforts and enable design of more complex DNA and RNA nanostructures. Additionally, with the exception of Tiamat, all available tools focus only on DNA nanostructure designs.

Molecular simulations have proved indispensable in the field of nucleic acid nanotechnology, providing detailed information about bulk structural characteristics (18,19), folding pathway kinetics (20,21), conformational space and kinetics of complex nanostructures (22–24), and active devices such as DNA walkers (25,26). Due to the size of the designed nanostructures and the laboratory timescales involved, traditional fully atomistic simulation methods are often infeasible for nucleic acid nanotechnology applications. To remedy this, several coarse-grained models have been developed (27–35), each of which with a unique focus on a specific part of the DNA nanostructural design and characterization pipeline. In particular, the oxDNA/oxRNA models have grown in popularity in recent years and have been used for studying DNA/RNA nanostructures and devices (22,31,36–38) as well as RNA/DNA biophysics (29,39,40). The models represent each nucleotide as a single rigid body, where the interactions between nucleotides are empirically parameterized to reproduce basic structural, mechanical and thermodynamic properties of DNA and RNA (Figure 1).

However, the standalone simulation package only provides simulation trajectory with recorded 3D positions of all nucleotides in the simulation. Users usually have to develop in-house evaluation tools that post-process the simulation trajectory to extract desired properties of the studied nanostructures.

In this paper, we present two open-source tools to fill these unmet needs in the field of DNA/RNA nanotechnology and illustrate their use for design and optimization of DNA and RNA nanostructures. The first tool we introduce here is oxView, a browser-based visualization and editing platform for DNA and RNA structural design and analysis of nanostructures simulated in oxDNA/oxRNA. The tool is able to accommodate nanostructures containing over a million nucleotides, which is beyond the reach of most other visualization tools. It allows the user to load multiple large nanostructures simultaneously and edit them by addi-

tion or deletion of individual nucleotides or entire regions, providing a way to create new, more complex designs from smaller, individually designed subunits, even from different design tools. All of the previously mentioned design tools can be converted to the oxDNA format using either built-in tools (Adenita, MagicDNA, vHelix), the TacoxDNA web-server (41) (CaDNAno, Tiamat, vHelix), or by converting first to PDB using built-in tools and then to oxDNA using TacoxDNA (DAEDALUS, PERDIX). The visualization tool is integrated with oxDNA/oxRNA simulations and loads long simulation trajectories quickly (including files which are tens of gigabytes in size) for interactive analysis and video export of nanostructure dynamics. It can also load data overlays from the analysis scripts introduced in this paper, allowing users to interactively explore features such as hydrogen bond occupancy and structure flexibility and then use this information to iteratively redesign nanostructures based on simulation feedback using oxView. Finally, oxView implements rigid-body dynamics code so that individual parts of the structures can be selected and interactively rearranged. The structure will then be relaxed on-the-fly using rigid-body dynamics to a conformation which can be used as an initial structure in simulations.

The second tool introduced here is a set of standardized structure-agnostic geometry analysis scripts for oxDNA/RNA which cover a number of common molecular simulation use cases. Many groups that work with oxDNA/RNA have developed their own analysis tools in-house, resulting in many duplicate functionalities and scripts that are limited to single experiments. To facilitate the simulation-guided design of DNA/RNA nanostructures and lower the barrier of entry into the simulation field, we have developed a toolkit that is easy to use, generically applicable to numerous studied systems, and extensible. The tool set includes the following: (i) calculation of mean structure and root-mean squared fluctuations to quantify structure flexibility; (ii) hydrogen-bond occupancy to quantify fraying and bond breaking during the simulation; (iii) angle and distance measurements between respective duplex regions in a nanostructure; (iv) a covariance-matrix based principle component analysis tool for identification of nanostructure motion modes and (v) unsupervised clustering of sampled configurations based on structural order parameters or global difference metrics.

We demonstrate the versatility of the analysis tools and visualization platform functionality by analyzing simulations of previously published structure and a few novel designs. In particular, we study two RNA tiles, a Holliday junction, the tethered multi-fluorophore structure, two wireframe DNA origamis, and a single-stranded RNA origami nanostructure. We make no custom modifications to the analysis tools for each of the designs to demonstrate their versatility and general utility for distinct nanostructures. The visualization and analysis software developed in this work is freely available under a public license.

MATERIALS AND METHODS

System and software requirements

The analysis tools were written and tested using the following dependencies: Python 3.7 (minimum version 3.6),

NumPy 1.16 (42), Matplotlib 3.0.3 (minimum version 3.0) (43), BioPython 1.73 (44), SciKitLearn 0.21.2 (45) Pathos 0.2.3 (46), oxDNA 6985 (minimum version June 2019) (30,31,47).

OxView will run as-is on any modern web browser with WebGL support; though, we note that Google Chrome performs best at very large structure sizes. To make modifications to the code, the following dependencies are required: JavaScript ES6, Typescript 2.9.0

Simulation details

The oxDNA simulations of systems that were used in this work have been carried out using the standard molecular dynamics and Monte Carlo approaches. The simulation parameters and file formats produced by the simulations are described in the Supplementary Material.

RESULTS

OxView - Web browser visualization, analysis and editing of nanostructures

We introduce oxView, a JavaScript app built on the Three.js visualization library to provide fast, user-friendly, and flexible visualization capabilities with low technical overhead (Figure 2). OxView uses hardware instancing to offload most calculation of object geometry to the computer's GPU, allowing it to smoothly visualize structures containing millions of nucleotides (Figure 2 A). Standard Three.js scenes encounter a bottleneck in the rate of CPU draw calls with only a few thousand objects on the screen. By using instanced materials and custom properties written into the WebGL shaders, oxView bundles many objects with similar geometries into a single draw call that calculates edges and vertices in the compiled shader code.

Loading a simulation is as simple as dragging and dropping a trajectory/topology file pair onto a browser window with the app running. Simulation trajectory files can be stepped through using onscreen buttons or the keyboard, and the trajectory movie can also be downloaded as a video file. Available formats are .webm, .gif, and .jpg/.png image archives.

In addition to visualization, oxView also has basic editing capabilities (Figure 2B and Supplementary video 2). Particles can be selected individually, or whole strands and systems can be selected as a whole. Box selection, range selection (shift+click) and cluster selection are also available. Clustering can be done automatically using a Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm (48) or can be assigned manually from other selection methods. Briefly, DBSCAN compares distances between points and classifies groups of points meeting a specified minimum size and within a specified mutual minimum distance as members of the same cluster. It also characterizes points as central or peripheral, with central points having at least the minimum number of neighbors in the cluster and peripheral points being within the cutoff distance of one or more, but fewer than the specified number of central points. When manually selecting nucleotides, holding the control key while making a selection

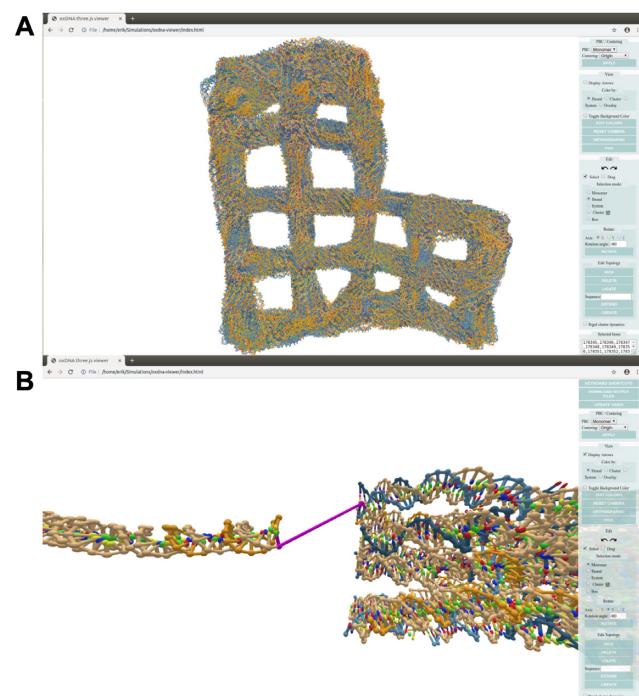


Figure 2. Screenshots from usage of oxView. (A) 100 configurations from an oxDNA simulation of design 24 from (17) merged into a single file and loaded into oxView; illustrating the ability to smoothly visualize over 10^6 nucleotides. The origami design has 11 382 nucleotides, resulting in a combined file containing 1 138 200 nucleotides, which renders as 5 691 000 individual objects in the scene. (B) Using oxView to assemble a simulation of the tethered multifluorophore (TMF) structure used in (49). Each of the subunits is a separate CaDNAno file converted into oxDNA format using (41). The two subunits and the algorithmically generated tether had to be ligated prior to simulation.

will combine the new and previous selections. Selected particles can be translated and rotated, and the topology can be edited via strand extension and creation, nicking, deletion, and ligation. Edits can be undone and redone using the standard ctrl-z/ctrl-shift-z keyboard shortcuts. Strand extensions will attempt to approximate either an A-form or B-form helix depending on the parent nucleotide's identity: RNA or DNA. The final edited version can be downloaded as an oxDNA file pair for further simulation or as a CSV sequence list for experimental validation.

We envision this tool being used to prototype DNA/RNA nanostructural designs in an iterative process before realization in the lab. The structure can be simulated for a short time, analyzed for defects, and then iteratively modified in the viewer and returned to simulation to verify success. This tool is also useful as a neutral ground between structures designed in other editing tools, allowing researchers to merge together structures from many sources to realize a complex vision.

OxView also allows the creation of mutual trap external force files for oxDNA/RNA. These files define artificial pairwise spring potentials between nucleotides that can be loaded in an oxDNA simulation and be very helpful when simulating the relaxation of a complex structure, assembled from multiple components, or when relaxing a structure imported from the CaDNAno format.

Implementation details. The underlying architecture of oxView has two parallel data streams. The first mirrors the physical arrangement of nucleic acid monomers into strands, with each configuration/topology pair representing a system. This data structure contains the topological information relating to particle identities, connectivity, and relation to the system. Monomers, strands and systems all inherit from the Three.js Group object and are related through an inheritance hierarchy, which allows interaction with structural units as a group. Additionally, each system contains a set of data arrays that define the positions, orientations, sizes, and colors of every particle. These arrays are passed into a custom implementation of the WebGL Lambert shader, where they are compiled on the GPU and drawn as a single object. This scheme allows loading of over 1 million nucleotides into a single scene (Figure 2A and Supplementary video 1).

Selection is handled through a GPU-picker, which avoids the need for computationally-expensive raycaster intersection calculation. Briefly, each nucleotide has a mesh with a color corresponding to its global ID at the same position as its backbone site which is rendered in an invisible scene. The color of this mesh can quickly be determined via the x-y coordinates of the mouse on the screen. When the color is converted from the hexadecimal color to the corresponding decimal value, it returns the ID of the nucleotide under the mouse pointer. As the arrays passed to the shader are of constant-size, new nucleotides added to the scene after initialization, are placed in a temporary system object with its own instancing arrays.

Data overlays in oxView. Many of the simulation analysis scripts introduced in this work output overlay files that can be viewed in oxView. This allows interactive visualization of different properties (such as flexibility, discussed in Figure 4) of respective parts of the structure obtained from simulations. These are JSON-format files that define the name of the overlay and the data. There are three types of overlays recognized by oxView. The most frequently used is the color overlay. These files contain one value per particle. When dragged and dropped into oxView, along with the corresponding configuration/topology pair, the color overlay file will create a superimposed colormap on the structure based on the value associated with each particle. All 256-value colormaps from Matplotlib (43) are available in addition to the default Three.js colormaps. The displayed colormap can be altered via a simple API implemented in the browser console. In addition to per-nucleotide coloring, oxView can also read two JSON formats corresponding to arrows drawn on the scene. The first is a three-component vector for each nucleotide, which is produced by the principal component analysis script and draws a vector, emanating from each particle, using the magnitude and orientation defined in the overlay file. The second format, which can contain any number of vectors, takes pairs of three-component vectors and draws arrows of the corresponding position and orientation on the scene.

Relaxing structures using rigid body dynamics. There has been a recent push to develop software that converts structures designed in the various design tools to simulation for-

mats (41). Due to the lattice-based drawing platform with parallel helices used by CaDNAno, exported structures can be very difficult to relax to a physically reasonable state in oxDNA. Initial configurations imported from CaDNAno (shown in Figure 3 A) will generally be planar with highly stretched bonds between individual structural units. Thus, without 3D information on how to reorient the helices, neither MC nor MD simulations are able to find the relaxed arrangement. This can also lead to topological impossibilities, where structures are knotted in a nonphysical manner. Additionally, starting simulations from a state with very stretched bonds can result in numerical instabilities that crash the simulation. For origami structures consisting of multiple origami blocks, connected by initially stretched backbone bonds, rigid-body manipulation has previously been used to arrange the converted oxDNA structure into a more realistic initial configuration (50). The translation and rotation tools in oxView allow users to select and rearrange blocks of nucleotides as rigid bodies. Furthermore, oxView also includes a rigid-body dynamics (RBD) (51) mode, that automatically transforms groups of nucleotides based on a simple force field. It is also possible to drag and rotate groups during RBD, allowing the user to nudge the design into the desired topology. Groups can either be created manually via the selection interface or through the implemented DBSCAN algorithm (48) that automatically identifies and categorizes spatially separated groups of particles. The latter option works particularly well with designs developed in CaDNAno.

Each group is represented as a rigid body with a position and an orientation. The groups are held together with spring forces at each shared backbone bond, with a magnitude of

$$f_{\text{spr}} = c_{\text{spr}}(l - l_r),$$

where c_{spr} is a spring constant, l is the current bond length and l_r is the constant relaxed bond length. To avoid overlaps, a simple linear repulsive force, of magnitude

$$f_{\text{rep}} = \max \left(c_{\text{rep}} \left(1 - \frac{d}{r_a + r_b} \right), 0 \right)$$

is added between the center of each group, where c_{rep} is a repulsion constant, d is the distance between the two centers of mass, and $r_a + r_b$ is the sum of the group radii (the greatest distance they can be while still overlapping). An example of the dynamics in action can be seen in Figure 3 and Supplementary video 3, where each side of a DNA icosahedron (52) is automatically arranged into the intended shape.

General-purpose analysis tools

Popular molecular simulation tools programs, such as GROMACS (53), not only perform molecular simulations, but also include analysis tools for common use-cases. The access to reliable and maintained tools, as part of the distribution, allows for standardization between many researchers using the core tool, as well as simplifying the learning curve for new researchers working with the tool. At this time, although there are over a hundred publications using oxDNA/RNA, no standardized set of tools for structural analysis has emerged. We present here a set of tools

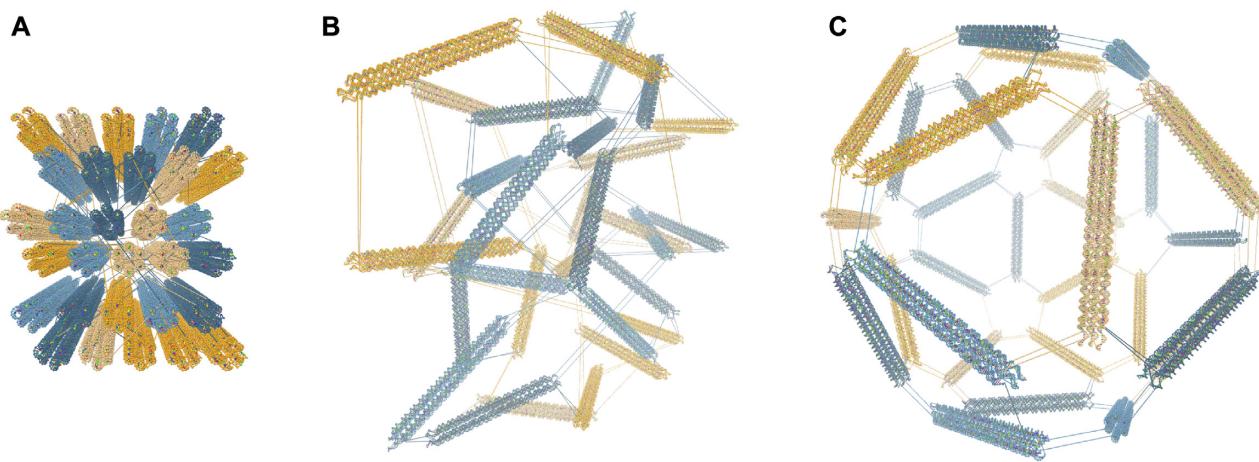


Figure 3. Rigid-body dynamics of clusters. Snapshots from the automatic rigid-body relaxation of an icosahedron, starting with the configuration converted from caDNAno (**A**), through the intermediate (**B**) where the dynamics are applied, and (**C**) the final resulting relaxed state.

covering many common structure analyses: mean structure, root mean squared fluctuations (RMSF), hydrogen bond occupancy, interaction energy, interduplex angles, contact mapping, the distance between nucleotides, and principal component analysis of structure motion. These are primarily written in Python, with some portions embedded in the oxDNA C++ code for enhanced speed. Moreover, we provide additional utilities including a parallelization scheme for analyses, trajectory alignment, and unsupervised clustering based on data outputs.

Mean structure determination and RMSFs. This package includes two methods for determining the mean structure. One utilizes the Biopython (44) singular value decomposition (SVD)-based structure superimposer. This is a popular method (54) that finds a translation and rotation to superimpose two distinct conformations on top of each other to minimize the root mean square distance between their components. Either a user-defined or random configuration in the trajectory is selected as the reference structure. In the example structures displayed here, this choice was found to have little impact on the final outcome. Each configuration is then superimposed onto the reference, and the average position of each nucleotide is calculated by taking the mean of each particle's coordinates in the aligned reference frame. The alignment can also be performed on a subset of particles in the structure. These are assigned from a space-separated index file that can be produced by clicking the 'Download Selected Base List' button in oxView. Sometimes, a mean structure is undesirable because they are frequently not physically possible state. To obtain a physically reasonable, but representative structure, this package also includes a centroid-finding script which finds the structure in a trajectory that has the lowest total RMSF to the a provided reference (such as a mean structure). To find the per-particle RMSF, a second script uses the mean structure produced by the first script as the reference configuration for alignment. The squares of the distances between the alignment and the mean structure for each nucleotide are then summed and divided by the total number of configurations. The square root is then taken to find the RMSF per parti-

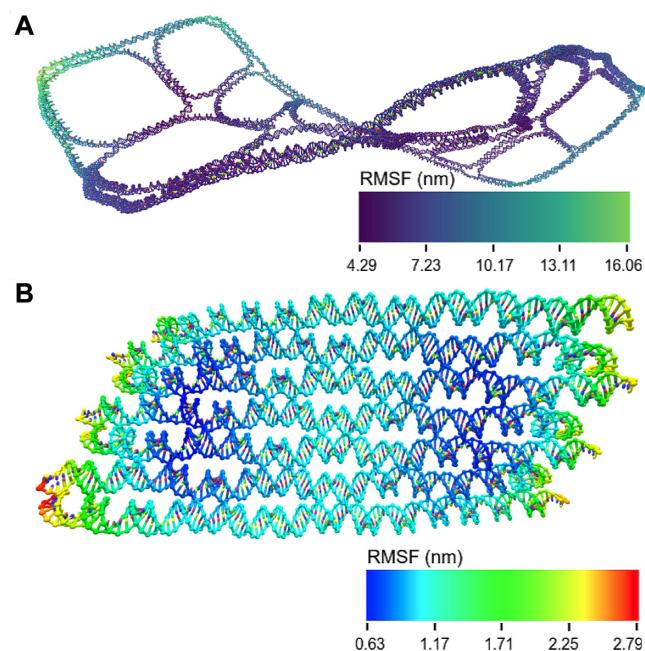


Figure 4. Mean structures and RMSF. (**A**) The mean and deviations scripts were used to compute the mean structure and RMSFs of design 19 from (17). In the initial report of these designs, they were characterized by AFM, showing complete, flat structures. In the simulations here, the structures were stable; however, the mean structure shows a significant right-handed global twist. (**B**) To demonstrate the patterns that appear in RMSF calculations, this is the mean structure of a single-stranded RNA origami (60) with the RMSF shown using a colormap with high spectral contrast. The center of the origami appears to have an RMSF twice as high as the surrounding regions. This is simply an artifact of the alignment and not an accurate characterization of particle motion.

cle in nanometers. The final output from this script is a .json format color overlay that can be loaded into oxView.

As noted in (22), averaging methods that use full structure alignment work very well for rigid structures; However, there are some caveats. Large planar structures frequently appear to have the smallest RMSF in a ring midway be-

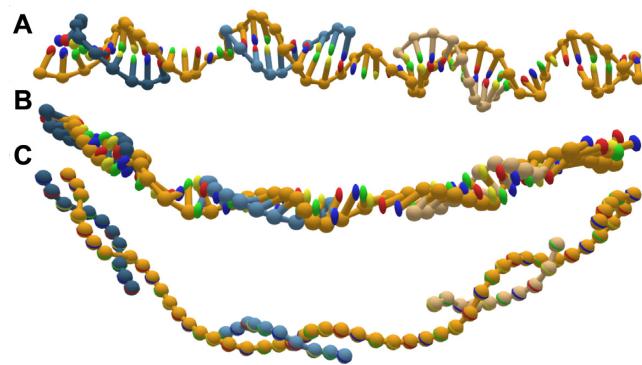


Figure 5. Improving mean structures of flexible designs. (A) The initial configuration of a 50-nucleotide duplex interrupted with 5 nucleotide gaps, created using the editing tools in oxView. Each individual configuration encountered during simulation displayed helical geometry. (B) The mean structure computed using SVD of the whole simulation. Because of the high backbone and rotational flexibility of this structure, it collapses into a linear shape that has little correspondence to the double helix geometry that is maintained throughout the simulation. (C) The mean structure computed using MDS. In this case, since only local contacts are used to construct the mean structure, the helical geometry is maintained. MDS comes at the cost of losing nucleotide orientation information, however. Thus, the visualization only shows the center of mass for each nucleotide.

tween the center and the edge (Figure 4B). This does not correspond to lower flexibility, but instead reveals an artifact of the single-value decomposition. If a structure can bend in two possible directions, the stationary point in the oscillation will appear to have very low flexibility. Highly flexible regions tend to collapse towards a center line, which is particularly problematic for rigid structures connected by a flexible linker, exemplified by the interrupted duplex shown in Figure 5A. When the average structure is computed for this design, the entire structure collapses into a linear blob that does not have any resemblance to any of the individual configurations. This is because the average position for these flexible particles is drawn towards the center. For such structures, another mean structure calculation based on interparticle distance is employed.

The second option for mean structure determination uses a common machine learning technique, multidimensional scaling (MDS) (55), to reconstruct a mean structure from local contact maps. MDS is one of a class of algorithms known collectively as manifold learning, which are traditionally used to perform dimensionality reduction in high-dimensional datasets. MDS takes a set of pairwise distances between points in an arbitrary number of dimensions, as an input. The algorithm then uses eigenvalue decomposition to find distances d_{ij} in the embedded space that minimize

$$f(\delta, d) = \sum_{i,j=1}^N (\langle \delta_{ij} \rangle - d_{ij})^2,$$

where N is the number of data points, $\langle \delta_{i,j} \rangle$ is the mean distance between centers of mass of nucleotides i and j (averaged over the whole simulated trajectory) and $d_{i,j}$ is their embedded distance (45). In the implementation presented here, pairs of nucleotides, where average distance $\langle \delta_{i,j} \rangle$ is longer than the cutoff of $r_{\text{cut}} = 2.07$ nm (approximate

ly the interhelix gap in an origami), are not considered in the embedding. The MDS-based mean structure calculation uses the MDS algorithm (56), implemented in the Python machine learning toolkit, SciKit-Learn (45), to reconstruct these local distances into a three-dimensional embedded representation. This method loses orientation data, and thus, nucleotides are simply visualized as spheres at their centers of mass (Figure 5). Once a mean structure (in the embedded space) is calculated, the script then calculates the mean deviation in distance between each particle and its nearest neighbors and outputs an oxView color overlay file to quantify the flexibility.

We used the SVD-based mean structure script to study flexibility and curvature in large wireframe origami structures (17). In the original research, these structures were visualized using atomic force microscopy (AFM), which tends to overestimate the flatness of structures due to electrostatic interactions between the mica surface and the DNA origami (4). Though the wireframes appear flat in the published AFM results, our simulations suggest that in solution they would be more crumpled or have some degree of global helical twist. Particularly striking is the helical shape of the mean structure of design number 19 from (17) (shown in Figure 4A and Supplementary video 4). OxDNA was parameterized to correctly reproduce the global twist of large 3D DNA structures (47,57), suggesting that this twist is likely significant while in solution. We note, however, that the global twist of 2D DNA nanostructures in the bulk remains a topic of active research (58), and more experimental data is needed to establish a better comparison of oxDNA parametrization with experimentally determined structures. Mean structures are also the best method to compare simulation results to cryo-EM maps. Both produce an averaged structure over thousands of individual snapshots. Thus, converting mean structures to PDB format using existing conversion tools (41) for use with cryo map fitting software, such as can be found in Chimera (59), is a method to correlate simulations and experimental data.

Because of the limitations of SVD-based mean structure calculation, the MDS approach was also used to determine the mean structure and deviations. Unfortunately, because average distance data is noisy and does not precisely map to a single configuration, this method does not work for structures larger than a few thousand particles. In all tests of the algorithm at origami scales, every particle was placed at the origin, a trivial solution that is a known issue of manifold learning methods. However, at smaller scales, this method provides a reasonable mean structure, that respects the geometry of the double helix, and a measure of deviation that reveals areas of flexibility without global artifacts due to fitting (Figure 6).

Geometric parameters: interduplex angles and distances. The simplest structural unit of nanotechnology structures is the duplex—antiparallel strands of sequentially bonded nucleotides. We have implemented a script that automatically determines the duplexes present in each configuration within a trajectory and fits a vector through the axis of the duplex. This is trivial for DNA, where the center points of each base pair lie roughly co-linear and the axis can be defined by a linear regression through the points in the center

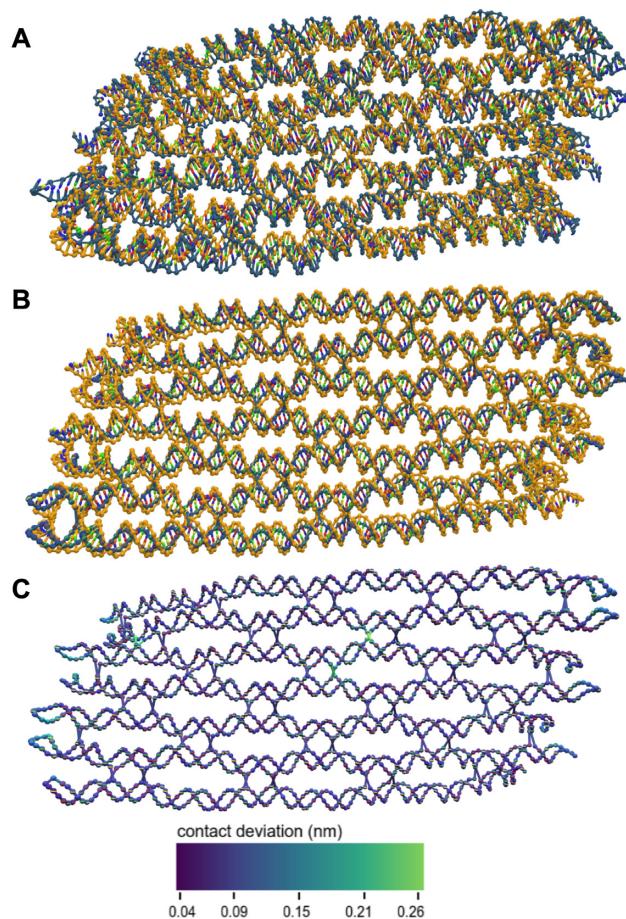


Figure 6. Centroid structure and mean computed via multidimensional scaling. (A) The centroid structure (blue) observed during a simulation of a single-stranded RNA origami from (60) overlaid on the SVD-computed mean (yellow). This is the structure with the lowest RMSF to the mean structure. (B) The mean structure as computed both by SVD (yellow) and MDS (blue). Because MDS does not preserve orientation data, the nucleotides are visualized simply as spheres at their center of mass, rather than having distinct base/backbone sites. (C) The deviation in local contacts from the mean structure calculated in (B). This measure shows most of the structure to be homogeneously stable, with higher flexibility at helix ends and at junctions capable of sliding.

of the duplex. For RNA, the A-form helix is slightly more difficult to characterize. The duplex is defined by the normal vector to an average plane fit through the displacements along the backbones as described in (30,61). This script creates a text file that contains information about all duplexes found at each step. This can be visualized using a separate script, which uses the ID of nucleotides at the edge of the duplex, found using oxView's selection feature. This method can compare angles either within or between structures.

Determining the angle between two duplexes can be useful in assessing design outcomes as well as quantifying twist within nanostructures. The output from the angle script is a list of all duplexes found in each configuration of the trajectory. This output can then be fed into the partnered visualization script along with the starting nucleotide IDs of the duplex. The output will be the median, mean and standard deviation of the angle between the two duplexes, as

well as the fraction of analyzed configurations in which that pair of duplexes are both present. This number is an indication of both how stable the structure is and whether or not the chosen duplex is representative of the entire trajectory. The script will also provide a histogram and/or trajectory of the angle over the course of the simulation. Here, we show an example of the angle script again using the wireframe origami designs from (17). Each origami has a designed junction angle corresponding to the number of arms joined at each junction (Supplementary Figure S2 in the Supp. Mat.). Deviation from this designed angle is a measure of strain and how non-planar the structure is in simulation. This can be particularly revealing in combination with the mean structure, showing that an on-average flat structure has a significant degree of flexibility over the course of the simulation.

The tethered multi-fluorophore (TMF) structure from (49) was used as a demonstration of the distance script. This structure is used to measure binding kinetics through the large change in radius of gyration induced by binding and unbinding of compatible sequences near the ends of the double-stranded tether. End-to-end distance of the tether in both the bound and unbound states are shown in Supplementary Figure S3 in the Supplementary Materials. Knowing the end-to-end distance of this structure can be used in predicting the radius of gyration for various states of the structure, which is useful in corroborating experimental results.

Base pair occupancy. The hydrogen bonds defining Watson–Crick base-pairing are the single most important parameter defining DNA/RNA nanotechnology geometries. Since structures are designed towards a theoretical global free-energy minimum that maximizes hydrogen bonds, deviations from the designed structures point to regions of significant topological strain or that have found a kinetically trapped structure distinct from the intended design. OxDNA/RNA defines hydrogen bonds between base-paired nucleotides as a base-pairing potential between two base particle beads less than $-0.1 k_b T$, about 10% of the magnitude of the equilibrium value of the base pairing potential of a base pair in a duplex. The script compares the hydrogen bonds in a simulation with a provided list of pairs present in the intended design. The fraction of the configurations in which the intended bonds are formed are reported as an oxView overlay file, with color coding intensity corresponding to the fraction of the time where the bonds are formed. Bonding is considered 0 for nucleotides without designed complements.

Since the structures exported from design tools represent an idealized form, deviations from the original vision imply unmet design constraints. In Figure 7, we use this script to explore a poorly-formed RNA tile structure. We first simulated the original tile design, as shown in Figure 7A. The hydrogen bond occupancy data revealed intense stress in a single duplex, with individual bonds ranging from 0 to 60% occupancy. This introduced considerable flexibility to the structure, disrupting the intended planar design. When the duplex was redesigned to extend it by one base pair, it no longer suffered from the same disruption, and the intended design was observed in the simulation (Figure 7B).

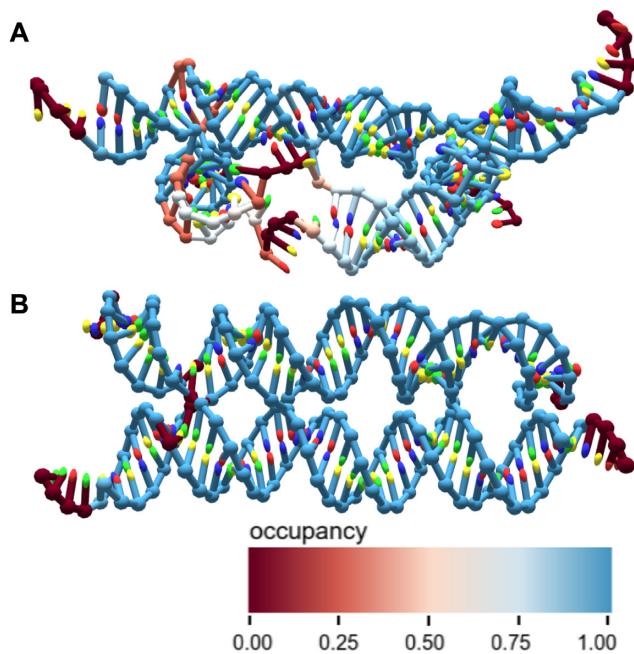


Figure 7. Bond occupancy of an RNA tile. (A) The hydrogen bond occupancy during an oxRNA simulation, overlaid on a structure of an RNA tile. The structure was known to form poorly in the lab, and the simulation revealed significant strain on one duplex. The structure used here is the centroid of a trajectory based on the global fitting parameters discussed later. This was used as a visualization instead of the mean structure, as the unpaired duplex made the structure so flexible that the mean structure collapsed. (B) The broken duplex from the structure in (A) was extended by one base pair, and the simulation was re-run. Shown here are the hydrogen bond occupancies overlaid on the mean structure. In simulation, this significantly improved rigidity.

Principal component analysis of nanostructure motion modes. Principal component analysis (PCA) is a common method for analyzing molecular simulation data that extracts the largest sources of deviation from the dataset (62). First, using SVD, each configuration is aligned to a mean configuration (produced by either SVD or MDS) to remove rotations and translations from the data. Each nucleotide's deviation from its reference position in x- y- and z-coordinates is stored as its difference matrix. A covariance matrix is then constructed from the difference matrices, and the eigenvalues and eigenvectors are found through eigenvalue decomposition. These are then sorted in descending order with the highest eigenvalues representing the largest sources of variation in the structure. The eigenvectors generated by PCA represent an orthogonal basis for the reconstruction of every structure visited during the trajectory, and these reconstructions can then be used for clustering of distinct sampled conformations. Finally, the PCA script outputs a json file for the oxView tool, which displays arrows on the structure corresponding to the sum of a user-defined number of components weighted by their respective eigenvalues.

To demonstrate the principal component analysis of DNA/RNA structures developed in this work, we ran it on a simulation of a Holliday junction (Figure 8). As one would expect for this structure, PCA reveals strong collective mo-



Figure 8. Principal component analysis of a Holliday junction visualized on oxView. Shown here is the top mode, which corresponds to a scissoring motion in the junction, with the arm ends having significantly higher average displacement than the crossover point.

tion for the junction arms. The motion grows stronger at the ends of duplexes, while the crossover point shows little motion.

Unsupervised clustering of configurations encountered in simulation. The trajectories produced in an oxDNA/RNA simulation can be tens of gigabytes in size and explore an expansive amount of the configuration space available to the structure. In cases where multiple metastable states are visited during the trajectory, aggregate structural data, such as mean structures or base pair occupancy, might not be representative of the ensemble. This is due to the presence of these distinct metastable states. Here, we once again use the DBSCAN clustering algorithm (48), as implemented in SciKit Learn (45), to automatically extract clusters of geometrically distinct structures from large trajectories and save each cluster as a separate file containing a collection of configurations that can be analyzed independently. The clustering algorithm can take any matrix of positions as an order parameter, whether that be principal component coefficients of each configuration, or simply the distance between two particles. The DBSCAN algorithm is particularly good at clustering molecular simulation data where metastable states tend to form distinct clusters separated by a large energy barrier, such that observing transition states is relatively rare and multiple distinct densities are observed.

To demonstrate the utility of clustering using structural order parameters, we analyzed a simulation of an RNA tile structure (Figure 9), that is known to form two distinct structural isomers in experiment (unpublished results). In the simulation, two states were encountered, the correctly-folded structure, with three crossovers, and an unfolded structure, in which the paramecic cohesion (63) between two of the crossovers is lost, leaving essentially a Holliday junction (Figure 9, cluster 2). There are many potential order parameters that can be used to separate out these two structures. In this case, we chose to work with the most aggregate data: each configuration's position in principal component space.

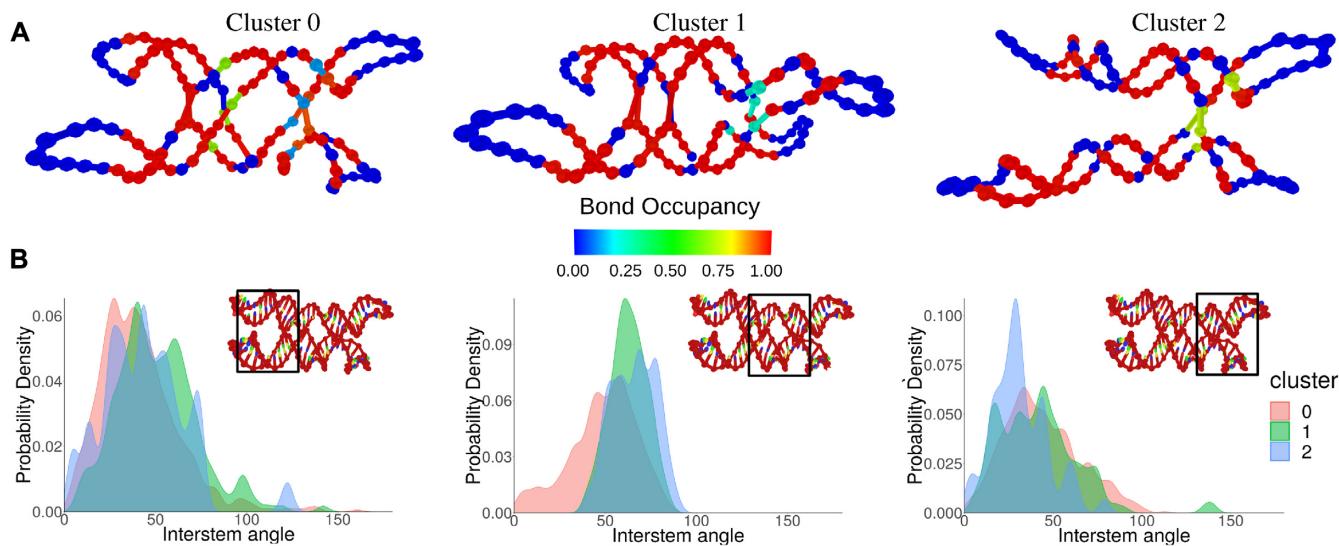


Figure 9. Unsupervised clustering to isolate isomers of an RNA tile (A) The three clusters found in a simulation of a single-stranded RNA tile. The mean structure of each cluster was determined using MDS, and the hydrogen bond occupancy compared with the original design was used as an overlay. (B) Histograms of the angles found in each cluster showing the distinct structures found in each cluster. The black frame on the tile snapshot indicates pairs of double-stranded RNA regions that were used to calculate the interstem angle.

The components produced by PCA (Supplementary Figure S1 in the Supplementary Material) represent a linearly independent basis for describing structures relative to the provided mean structure. This also means that every configuration used to compute the components can be mapped to a unique point in $3N - 6$ dimensional space. When applying DBSCAN to the positions of configurations in this space (described in detail in Supplementary Material), the distinct conformational isomers can be separated without further processing. In addition to the two expected configurations, this method also separated out another cluster (cluster 1 in Figure 9) of structures where the paramecic cohesion was correctly formed, but stacking was interrupted at the nick point, resulting in a non-planar kinked structure. The overlay in Figure 9A shows the fractional hydrogen bond occupancy compared with the original design. Of particular note is the large stretch of blue on the left side of cluster 2 where the bonds that form the paramecic cohesion are missing. The clusters were further analyzed using the angle script, identifying the distinct interduplex angles between each duplex in the structures (Figure 9B). These distributions show the fully formed structure (cluster 0) as having the lowest angle between the left duplexes in the first panel of Figure 9B and cluster 1 having a very defined angle between the central duplexes (Figure 9B, center).

Other utilities. In addition to the specific structural measures discussed here, this package also contains additional utility functions for processing and displaying data. The first are two scripts that utilize the SVD superimposer from Biopython (44) for improving visualizations. The superimposing script takes multiple configuration files that share the same topology and returns them with their translations and rotations removed relative to the first configuration provided. We find this very helpful for comparing mean structures of similar designs or of the same design under dif-

ferent simulation conditions. There is also an alignment script, which takes a trajectory file and aligns all configurations to the first one in the file. This makes for a much smoother visualization experience when exploring trajectories in oxView or when making movies of a trajectory.

We have found the alignment scripts to be very useful for producing figures and movies (see Supplementary video 5 and Figure 6 A) and for making comparisons between designs. These scripts are limited, however, by the need to align discrete units. Therefore, the structures must have the same number of particles in mostly the same position. Thus, the scripts are best used for comparing simulation conditions, changing sequences, and changing crossover positions in designs.

There is also a utility that reports the energy contribution of every interaction in the model. This has options of a text output to check specific values, as well as an oxView overlay showing the average energy of all nucleotides over the course of a simulation. Checking the base pairing or stacking interactions of specific nucleotides can be very helpful in identifying properties or defects in a given design. Additionally, we have found the visualization option useful for identifying excluded volume clashes during relaxations of large structures, as these cause extremely high total energies, which visually pop in oxView.

There are two further scripts that work with base pairs. One takes the current arrangement of base pairs in the structure and generates either the designed pairs file used by the base pair analysis script, or an oxDNA mutual trap force file, which can be used to enforce a particular base pairing configuration during relaxation. This can be particularly helpful when relaxing multi-component structures edited in oxView, as the forces pulling stretched bonds back together can cause unwanted fraying of base pairs in otherwise stable structures. The second script converts oxDNA force files into a designed pair file. The Tiamat converter

from (41) can produce force files as part of the conversion process, and this script can convert those force files into the format needed for the duplex angle script.

Finally, we provide a parallelization scheme for analyzing oxDNA trajectories. The parallelization module breaks down a trajectory into a number of chunks equal to the number of CPUs you have available, and uses the Pathos Multiprocessing library (46) to map trajectory chunks, CPUs, and functions. If the user has enough computational resources available, this facilitates analysis of even very large structures or long trajectories in a matter of minutes. The implementation of parallel functions is standardized across all scripts used here, and users are encouraged to follow the example given here in developing further analyses specific to their own designs.

Most of the analysis discussed fall into the class of tasks known as ‘embarrassingly parallel’, where there is no communication required between processes, and the final joining step is relatively easy. For all structure analysis algorithms described here, each configuration can be calculated independently of all the others. The only limitations to parallelization come from calculating split points in the trajectory and if a data trajectory is required, combining the outputs together in the proper order. As an example, we benchmarked parallelizing the computation of the mean structure of two structures: one with 423 nucleotides, and the other with 11 385. In both cases, runtime decreased by more than a factor of 10 when run on 30 CPUs compared with a single CPU, with diminishing returns past that point.

DISCUSSION

We developed this collection of tools to remedy two gaps that we have perceived in the oxDNA software environment. First is the lack of an all-in-one visualizer that loads files within a reasonable timeframe, has a user-friendly UI, and performs edits on structures that could then be further simulated. All-atom simulations have such tools in the form of VMD, Chimera and PyMol. While tools exist to convert between all-atom and oxDNA formats, this is a cumbersome process that we felt could be remedied by the development of oxView. The use of hardware instancing allows oxView to load structures of unprecedented sizes and facilitates our work on million-nucleotide oxDNA simulations of multi-origami structures. Furthermore, because oxView is built using the open-source 3D library Three.js, opens the possibilities for features from other Three.js projects to be added to oxView. For example, virtual reality oxDNA visualization was easily added by following the Three.js WebXR examples. Similarly, it is easy to export the visualized scene to other 3D formats, such as GLTF, for photorealistic rendering (Figure 1) or 3D printing (Supplementary Figure S4 in the Supp. Mat.).

The features of oxView and simulation analysis tools are designed to help researchers in DNA and RNA nanotechnology to prototype *in silico* their structures, simplify the design and optimization process, and better understand the functioning of the designed structures. We demonstrated the utility and versatility of the visualization and analysis tools on multiple DNA and RNA nanostructure designs, ranging in size from hundreds to multiple thousands of nu-

cleotides per structure. We also demonstrated that the tools can, in principle, handle structures of sizes over a million nucleotides.

These tools, particularly mean structure calculation and hydrogen bond occupancy, provide significant utility for iterative design of nanostructures. In many structures where unbounded growth is a goal, global curvature of the nanostructure due to subtleties in crossover placement is a significant bottleneck, that is difficult to solve using rational design principles. We have found that the curvature of mean structures calculated from oxDNA simulations (unpublished results) is a good predictor of lattice formation in the laboratory. We also note that mean structures are the best proxy for comparing simulations with cryo-EM structures, which have become important characterizations for 3D nanostructures in the nucleic acid nanotechnology field.

Hydrogen bond occupancy is a good proxy measure for the amount of stress built up in a structure. Even with the speed and level of coarse-graining that oxDNA provides, modelling assembly pathways for large structures remains out of reach for all but the most ambitious simulations(64). Because of this limitation, we perform simulations with the assumption that the structure forms as designed, and initiate the simulation with all hydrogen bonds present. Designed pairs that become unbonded or find different partners, particularly at junction points, are a good indication for points in the design that are stressed and would benefit from iterative design. In general, we found that successfully published structures had near 100% bond occupancy, while those that were proving difficult to obtain in the lab had regions with low occupancy.

We demonstrated the functionality and versatility of these tools by applying them to a range of DNA and RNA nanostructures, such as DNA and RNA origamis, as well as optimizing and analyzing an RNA tile.

All software discussed here is open-source and freely available through our GitHub under the GNU Public License. Pull requests, bug reports and feature suggestions are welcome, as we hope that these will provide fundamental support long into the future. All tools that were introduced here are documented on their respective GitHub repositories, with examples of use reproducing the figures in this paper.

DATA AVAILABILITY

The oxDNA code is available online on the oxDNA webpage dna.physics.ox.ac.uk. OxView is available as a web-based application on github.com/sulcgroup/oxdnviewer. The analysis package can be downloaded from github.com/sulcgroup/oxdna_analysis_tools.

SUPPLEMENTARY DATA

[Supplementary Data](#) are available at NAR Online.

ACKNOWLEDGEMENTS

We would like to thank R. Hariadi, C. Simmons, and X. Qi for the design files from their previous publications used as examples in this work. We would like to thank members of

the Šulc and Yan labs, specifically H. Liu, J. Procyk, and L. Chen for their feedback and beta-testing of the tools presented here.

FUNDING

National Science Foundation [1931487]; European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie [765703]; the EU funding concerns the contribution of J.B. Funding for open access charge: NSF [1931487].

Conflict of interest statement. None declared.

REFERENCES

1. Seeman,N.C. (1982) Nucleic Acid Junctions and Lattices. *J. Theor. Comput. Chem.*, **99**, 237–247.
2. Yan,H., LaBean,T.H., Feng,L. and Reif,J.H. (2003) Directed nucleation assembly of DNA tile complexes for barcode-patterned lattices. *Proc. Natl. Acad. Sci. U.S.A.*, **100**, 8103–8108.
3. Gerling,T., Wagenbauer,K.F., Neuner,A.M. and Dietz,H. (2015) Dynamic DNA devices and assemblies formed by shape-complementary, non-base pairing 3D components. *Science*, **347**, 1446–1452.
4. Matthes,M., Agarwal,N.P., Poppleton,E., Joshi,F.M., Šulc,P. and Schmidt,T.L. (2019) Triangulated Wireframe Structures Assembled Using Single-Stranded DNA Tiles. *ACS Nano*, **13**, 1839–1848.
5. Veneziano,R., Ratanalert,S., Zhang,K., Zhang,F., Yan,H., Chiu,W. and Bathe,M. (2016) Designer nanoscale DNA assemblies programmed from the top down. *Science*, **352**, 1534–1534.
6. Wei,B., Dai,M. and Yin,P. (2012) Complex shapes self-assembled from single-stranded DNA tiles. *Nature*, **485**, 623–626.
7. Han,D., Qi,X., Myhrvold,C., Wang,B., Dai,M., Jiang,S., Bates,M., Liu,Y., An,B., Zhang,F. et al. (2017) Single-stranded DNA and RNA origami. *Science*, **358**, eaao2648.
8. Tikhomirov,G., Petersen,P. and Qian,L. (2017) Fractal assembly of micrometre-scale DNA origami arrays with arbitrary patterns. *Nature*, **552**, 67.
9. Gopinath,A., Miyazono,E., Faraon,A. and Rothemund,P.W. (2016) Engineering and mapping nanocavity emission via precision placement of DNA origami. *Nature*, **535**, 401.
10. Li,S., Jiang,Q., Liu,S., Zhang,Y., Tian,Y., Song,C., Wang,J., Zou,Y., Anderson,G.J., Han,J.-Y. et al. (2018) A DNA nanorobot functions as a cancer therapeutic in response to a molecular trigger in vivo. *Nat. Biotechnol.*, **36**, 258.
11. Douglas,S.M., Marblestone,A.H., Teerapittayanon,S., Vazquez,A., Church,G.M. and Shih,W.M. (2009) Rapid prototyping of 3D DNA-origami shapes with caDNAno. *Nucleic Acids Res.*, **37**, 5001–5006.
12. Williams,S., Lund,K., Lin,C., Wonka,P., Lindsay,S. and Yan,H. (2009) Tiamat: a three-dimensional editing tool for complex DNA structures. In: Goe,A., Simmel,F.C. and Sosik,P. (eds). *DNA Computing*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 90–101.
13. Benson,E., Mohammed,A., Gardell,J., Masich,S., Czeizler,E., Orponen,P. and Höglberg,B. (2015) DNA rendering of polyhedral meshes at the nanoscale. *Nature*, **523**, 441–444.
14. Benson,E., Mohammed,A., Bosco,A., Teixeira,A.I., Orponen,P. and Höglberg,B. (2016) Computer-Aided Production of Scaffolded DNA Nanostructures from Flat Sheet Meshes. *Angew. Chem. Int. Ed.*, **55**, 8869–8872.
15. de Llano,E., Miao,H., Ahmadi,Y., Wilson,A.J., Beeby,M., Viola,I. and Barisic,I. (2019) Adenita: interactive 3D modeling and visualization of DNA Nanostructures. bioRxiv doi: <https://doi.org/10.1101/849976>, 29 November 2019, preprint: not peer reviewed.
16. Veneziano,R., Ratanalert,S., Zhang,K., Zhang,F., Yan,H., Chiu,W. and Bathe,M. (2016) Designer nanoscale DNA assemblies programmed from the top down. *Science*, **352**, 1534.
17. Jun,H., Zhang,F., Shepherd,T., Ratanalert,S., Qi,X., Yan,H. and Bathe,M. (2019) Autonomously designed free-form 2D DNA origami. *Sci. Adv.*, **5**, eaav0655.
18. Suma,A., Stopar,A., Nicholson,A.W., Castronovo,M. and Carnevale,V. (2020) Global and local mechanical properties control endonuclease reactivity of a DNA origami nanostructure. *Nucleic Acids Res.*, **48**, 4672–4680.
19. Maffeo,C., Yoo,J. and Aksimentiev,A. (2016) De novo reconstruction of DNA origami structures through atomistic molecular dynamics simulation. *Nucleic Acids Res.*, **44**, 3013–3019.
20. Fonseca,P., Romano,F., Schreck,J.S., Ouldridge,T.E., Doye,J.P.K. and Louis,A.A. (2018) Multi-scale coarse-graining for the study of assembly pathways in DNA-brick self-assembly. *J. Chem. Phys.*, **148**, 134910.
21. Reinhardt,A. and Frenkel,D. (2014) Numerical evidence for nucleated self-assembly of DNA brick structures. *Phys. Rev. Lett.*, **112**, 238103.
22. Snodin,B.E., Schreck,J.S., Romano,F., Louis,A.A. and Doye,J.P.K. (2019) Coarse-grained modelling of the structural properties of DNA origami. *Nucleic Acids Res.*, **47**, 1585–1597.
23. Ouldridge,T.E., Šulc,P., Romano,F., Doye,J.P.K. and Louis,A.A. (2013) DNA hybridization kinetics: zippering, internal displacement and sequence dependence. *Nucleic Acids Res.*, **41**, 8886–8895.
24. Schreck,J.S., Ouldridge,T.E., Romano,F., Šulc,P., Shaw,L.P., Louis,A.A. and Doye,J.P.K. (2015) DNA hairpins destabilize duplexes primarily by promoting melting rather than by inhibiting hybridization. *Nucleic Acids Res.*, **43**, 6181–6190.
25. Šulc,P., Ouldridge,T.E., Romano,F., Doye,J.P.K. and Louis,A.A. (2014) Simulating a burnt-bridges DNA motor with a coarse-grained DNA model. *Nat. Comput.*, **13**, 535–547.
26. Ouldridge,T.E., Hoare,R.L., Louis,A.A., Doye,J.P.K., Bath,J. and Turberfield,A.J. (2013) Optimizing DNA nanotechnology through coarse-grained modeling: a two-footed DNA walker. *ACS Nano*, **7**, 2479–2490.
27. Ouldridge,T.E., Louis,A.A. and Doye,J.P.K. (2011) Structural, mechanical, and thermodynamic properties of a coarse-grained DNA model. *J. Chem. Phys.*, **134**, 085101.
28. Snodin,B.E., Randisi,F., Mosayebi,M., Šulc,P., Schreck,J.S., Romano,F., Ouldridge,T.E., Tsukanov,R., Nir,E., Louis,A.A. et al. (2015) Introducing improved structural properties and salt dependence into a coarse-grained model of DNA. *The J. Chem. Phys.*, **142**, 234901.
29. Šulc,P., Romano,F., Ouldridge,T.E., Rovigatti,L., Doye,J.P.K. and Louis,A.A. (2012) Sequence-dependent thermodynamics of a coarse-grained DNA model. *The J. Chem. Phys.*, **137**, 135101.
30. Šulc,P., Romano,F., Ouldridge,T.E., Doye,J.P.K. and Louis,A.A. (2014) A nucleotide-level coarse-grained model of RNA. *J. Chem. Phys.*, **140**, 235102.
31. Doye,J.P.K., Ouldridge,T.E., Louis,A.A., Romano,F., Šulc,P., Matek,C., Snodin,B.E.K., Rovigatti,L., Schreck,J.S., Harrison,R.M. et al. (2013) Coarse-graining DNA for simulations of DNA nanotechnology. *Phys. Chem. Chem. Phys.*, **15**, 20395–20414.
32. Maffeo,C. and Aksimentiev,A. (2020) MrDNA: a multi-resolution model for predicting the structure and dynamics of DNA systems. *Nucleic Acids Res.*, **48**, 5135–5146.
33. Kim,D.-N., Kilchherr,F., Dietz,H. and Bathe,M. (2011) Quantitative prediction of 3D solution shape and flexibility of nucleic acid nanostructures. *Nucleic Acids Res.*, **40**, 2862–2868.
34. Castro,C.E., Kilchherr,F., Kim,D.-N., Shiao,E.L., Wauer,T., Wortmann,P., Bathe,M. and Dietz,H. (2011) A primer to scaffolded DNA origami. *Nat. Methods*, **8**, 221.
35. Hinckley,D.M., Freeman,G.S., Whitmer,J.K. and De Pablo,J.J. (2013) An experimentally-informed coarse-grained 3-site-per-nucleotide model of DNA: Structure, thermodynamics, and dynamics of hybridization. *J. Chem. Phys.*, **139**, 144903.
36. Engel,M.C., Smith,D.M., Jobst,M.A., Sajfutdinow,M., Liedl,T., Romano,F., Rovigatti,L., Louis,A.A. and Doye,J.P.K. (2018) Force-induced unravelling of DNA origami. *ACS Nano*, **12**, 6734–6747.
37. Sharma,R., Schreck,J.S., Romano,F., Louis,A.A. and Doye,J.P.K. (2017) Characterizing the motion of jointed DNA nanostructures using a coarse-grained model. *ACS Nano*, **11**, 12426–12435.
38. Hong,F., Jiang,S., Lan,X., Narayanan,R.P., Šulc,P., Zhang,F., Liu,Y. and Yan,H. (2018) Layered-crossover tiles with precisely tunable

- angles for 2D and 3D DNA crystal engineering. *J. Am. Chem. Soc.*, **140**, 14670–14676.
39. Matek,C., Šulc,P., Randisi,F., Doye,J.P.K. and Louis,A.A. (2015) Coarse-grained modelling of supercoiled RNA. *J. Chem. Phys.*, **143**, 243122.
 40. Ouldridge,T.E., Šulc,P., Romano,F., Doye,J.P.K. and Louis,A.A. (2013) DNA hybridization kinetics: zippering, internal displacement and sequence dependence. *Nucleic Acids Res.*, **41**, 8886–8895.
 41. Suma,A., Poppleton,E., Matthies,M., Šulc,P., Romano,F., Louis,A.A., Doye,J.P.K., Micheletti,C. and Rovigatti,L. (2019) TacoxDNA: A user-friendly web server for simulations of complex DNA structures, from single strands to origami. *J. Comput. Chem.*, **40**, 2586–2595.
 42. Van Der Walt,S., Colbert,S.C. and Varoquaux,G. (2011) The NumPy array: A structure for efficient numerical computation. *Comput. Sci. Eng.*, **13**, 22–30.
 43. Hunter,J.D. (2007) Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.*, **9**, 90–95.
 44. Cock,P.J., Antao,T., Chang,J.T., Chapman,B.A., Cox,C.J., Dalke,A., Friedberg,I., Hamelryck,T., Kauff,F., Wilczynski,B. *et al.* (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422–1423.
 45. Pedregosa,F., Weiss,R. and Brucher,M. (2011) Scikit-learn: machine Learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.
 46. McKerns,M.M., Strand,L., Sullivan,T., Fang,A. and Aivazis,M.A.G. (2011) Building a Framework for Predictive Science. *Proceedings of the 10th Python in Science Conference*. pp. 1–12.
 47. Snodin,B.E., Randisi,F., Mosayebi,M., Šulc,P., Schreck,J.S., Romano,F., Ouldridge,T.E., Tsukanov,R., Nir,E., Louis,A.A. *et al.* (2015) Introducing improved structural properties and salt dependence into a coarse-grained model of DNA. *J. Chem. Phys.*, **142**, 234901.
 48. Ester,M., Kriegel,H.-P., Sander,J. and Xu,X. (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Kdd Vol. 96*, pp. 226–231.
 49. Schickinger,M., Zacharias,M. and Dietz,H. (2018) Tethered multifluorophore motion reveals equilibrium transition kinetics of single DNA double helices. *Proc. Natl. Acad. Sci. U.S.A.*, **115**, 201800585.
 50. Huang,C.-M., Kucinic,A., Le,J.V., Castro,C.E. and Su,H.-J. (2019) Uncertainty quantification of a DNA origami mechanism using a coarse-grained model and kinematic variance analysis. *Nanoscale*, **11**, 1647–1660.
 51. Baraff,D. (1997) An introduction to physically based modeling: rigid body simulation I unconstrained rigid body dynamics. SIGGRAPH Course Notes, 1–31.
 52. Douglas,S.M., Dietz,H., Liedl,T., Högberg,B., Graf,F. and Shih,W.M. (2009) Self-assembly of DNA into nanoscale three-dimensional shapes. *Nature*, **459**, 414.
 53. Berendsen,H.J., van der Spoel,D. and van Drunen,R. (1995) GROMACS: A message-passing parallel molecular dynamics implementation. *Comput. Phys. Commun.*, **91**, 43–56.
 54. Kabsch,W. (1976) A solution for the best rotation to relate two sets of vectors. *Crystallogr. A*, **32**, 922–923.
 55. Young,G. and Householder,A.S. (1938) Discussion of a set of points in terms of their mutual distances. *Psychometrika*, **3**, 19–22.
 56. Borg,I. and Groenen,P. (2003) Modern multidimensional scaling: theory and applications. *J. Educ. Meas.*, **40**, 277–280.
 57. Dietz,H., Douglas,S.M. and Shih,W.M. (2009) Folding DNA into twisted and curved nanoscale shapes. *Science*, **325**, 725–730.
 58. Baker,M.A., Tuckwell,A.J., Berengut,J.F., Bath,J., Benn,F., Duff,A.P., Whitten,A.E., Dunn,K.E., Hynson,R.M., Turberfield,A.J. *et al.* (2018) Dimensions and global twist of single-layer DNA origami measured by small-angle X-ray scattering. *ACS Nano*, **12**, 5791–5799.
 59. Pettersen,E.F., Goddard,T.D., Huang,C.C., Couch,G.S., Greenblatt,D.M., Meng,E.C. and Ferrin,T.E. (2004) UCSF chimera - a visualization system for exploratory research and analysis. *J. Comput. Chem.*, **25**, 1605–1612.
 60. Han,D., Qi,X., Myhrvold,C., Wang,B., Dai,M., Jiang,S., Bates,M., Liu,Y., An,B., Zhang,F. *et al.* (2017) Single-stranded DNA and RNA origami. *Science*, **358**, eaao2648.
 61. Schomaker,V., Waser,J., Marsh,R.E. and Bergman,G. (1959) To fit a plane or a line to a set of points by least squares. *Acta Crystallogr.*, **12**, 600–604.
 62. David,C.C. and Jacobs,D.J. (2014) Principal component analysis: a method for determining the essential dynamics of proteins. *Methods Mol. Biol.*, **1084**, 193–226.
 63. Qi,X., Zhang,F., Su,Z., Jiang,S., Han,D., Ding,B., Liu,Y., Chiu,W., Yin,P. and Yan,H. (2018) Programming molecular topologies from single-stranded nucleic acids. *Nat. Commun.*, **9**, 4579.
 64. Snodin,B.E., Romano,F., Rovigatti,L., Ouldridge,T.E., Louis,A.A. and Doye,J.P.K. (2016) Direct simulation of the self-assembly of a small DNA origami. *ACS Nano*, **10**, 1724–1737.