TALOS

(**T**hree-dimensional, **A**lgorithmically-generated **L**ibrary of DNA **O**rigami **S**hapes)

Software Instructions and Demo

Dr. Hyungmin Jun

Laboratory for Computational Biology and Biophysics, MIT

An online version of this software is available at https://github.com/lcbb/talos.

Table of Contents

Welcome to TALOS!

TALOS simplifies and enhances the process of designing 3D scaffolded DNA origami wireframe nanoparticles from the CAD (Computer-Aided Design) geometry as an input. With this software, you will be able to render almost any target 3D shape as a scaffolded DNA origami nanoparticle composed of six-helix bundle (6HB) edges. By providing a geometry file with PLY ("Polygon File Format") format describing your target 3D geometry, TALOS can generate the following outputs:

- A CVS file of the list of synthetic staple strand sequences. These staple strands, when combined with your scaffold strand (generated by default by TALOS or provided by you), will self-assemble into your scaffolded DNA origami nanoparticle by following the standard annealing protocol provided in our work.
- A CNDO file of your nanoparticle. This output CNDO file (CanDo file format[1]) from TALOS can be used to predict the flexibility of programmed DNA nanoparticles[2]. Also, it can be used to convert the PDB ("Protein Data Bank") file which gives the coordinates of every atom in your structure as predicted by TALOS. With software such as PyMOL[3], VMD[4], UCSF Chimera[5], etc., you will be able to visualize and manipulate your atomic model in 3D space.
- BILD[6] files. BILD files are used for visualizing the target geometry, scaffold routing, staple sequence and cylindrical models, which are rendered by lines, polygons, and geometric primitives built in UCSF Chimera (see Fig 5 and Fig. 6).
- JSON file. This sequence design JSON file can be imported into caDNAno[7] for editing staple paths and sequences.

The major goal of this software is to broaden the use of DNA nanotechnology by the larger scientific community. Even if you are not an expert in scaffold DNA origami design, you can use TALOS to begin to explore the capabilities of this powerful molecular design paradigm!

TALOS features:

- Fully automatic procedure for the sequence design of scaffolded DNA origami nanoparticle composed of 6HB edges
- Two vertex designs; flat vertex (FV) and mitered vertex (MV) design
- Importing PLY file format as input
- JSON output for editing staple paths and sequences from caDNAno
- 3D visualization powered by UCSF Chimera
- 40 pre-defined target geometries

[1] https://cando-dna-origami.org/cndo-file-converter/
[2] https://cando-dna-origami.org/atomic-model-generator/
[3] https://www.pymol.org/
[4] http://www.ks.uiuc.edu/Research/vmd/
[5] https://www.cgl.ucsf.edu/chimera/
[6] https://www.cgl.ucsf.edu/chimera/docs/UsersGuide/bild.html
[7] http://cadnano.org/

- User-friendly TUI (Text-based User Interface)
- Online web resources and pre-compiled binaries for Microsoft Windows and macOS
- Free and open source (GNU General Public License, version 3.0)

Free online web resource:

http://talos-dna-origami.org

We also offer the free online resource TALOS that automatically converts any 3D object specified using a simple Computer-Aided Design file (PLY) into the synthetic DNA sequences that are needed to synthesize the target object. With the same input, the output files from the preceding online resource are the same as those from the compiled version of TALOS.

Part 1. Release package

The release package (as an executable file) is available for both Microsoft Windows and mac OS. These executable files are the console application for Windows (TALOS.EXE) and Mac (TALOS-Finder / TALOS-Terminal), which accept an input (PLY) and generate outputs through the command (Windows) or terminal (Mac). You can download the release package from:

Microsoft Windows

https://github.com/lcbb/talos/raw/master/release/talos-win.zip

macOS / Mac OS X

https://github.com/lcbb/talos/raw/master/release/talos-mac.zip

The current versions of the release package are compiled with Intel (R) Parallel Studio EX Composer Edition for Fortran Windows 2017 (Version 17.0.1.143) under Microsoft Windows 10 64-bit (Intel (R) i7-4470 CPU @ 3.40GHz, 24GB RAM) and Intel (R) Parallel Studio XE Composer Edition for Fortran macOS 2018 (Version 18.0.2) under macOS High Sierra (1.4 GHz Intel Core i5, 4GB RAM), respectively. If you are a Linux user, you will need to download source codes and compile them under the Linux system (see Part 3). In the folder named 'TALOS', you will have the following subfolders and files:

- File 'TALOS.EXE': This is the executable file to run TALOS on Microsoft Windows. The executable file can be run by double-clicking the icon (Part 2.1) or using the command shell (Part 2.2).
- File 'TALOS-Finder': This is an executable file to run TALOS on macOS by double-clicking it. (Part 2.1).
- File 'TALOS-Terminal': This is an executable file to run TALOS on macOS. The executable file can be open on the terminal (Part 2.2).

- File 'env.TXT': This text file contains the sequences of the scaffold as input. The sequences can be replaced with the user-defined sequences of the scaffold (see Table 1).
- Folder 'input': The user-defined geometry file (PLY) must be located here.

Table 1. Definitions of the fields and values in the env.TXT file.

Field	Value	Descriptions
para_platform	win | mac	Depending on user's operating system
para_cut_stap_method	max | opt	Staple-break rule • max: Maximized staple length • opt: Maximized the number of seeds
para_min_cut_stap	20 as a default	The minimum staple length
para_max_cut_stap	60 as a default	The maximum staple length
para_scaf_seq	0 | 1 | 2	Scaffold sequence • 0: M13mp18 (7,249nt) sequence • 1: User-defined sequence • 2: randomly generated sequence

The default scaffold sequence of TALOS is M13mp18 for required length less than or equal to 7,249-nt, a 8,064 sequence if greater than 7,250-nt and less or equal to 8,064-nt, a Lambda phage sequence if greater than 8,064-nt and less or equal to 48,502-nt, and a random sequence if greater than 48,503-nt. User can also define scaffold sequence by changing the 'para_scaf_seq' as 1 and enter sequence as on line at the third line in the 'env.TXT' file.

Part 1.1. Opening the TALOS software by double-clicking

The easiest way to run TALOS is to double-click 'TALOS.EXE' in the file manager of the Windows, or 'TALOS-Finder' in the Finder on Mac. Note that the text file 'env.TXT' should be in the same folder in order to run the software. The user-defined geometry file (PLY) should be in the folder named 'input'. By double-clicking the release package, you can see the TUI (Text based User Interface) on the console, which displays the pre-defined target geometries as first input parameters (Fig. 1). There are 40 pre-defined wireframe 3D structures including Platonic, Archimedean, Johnson, Catalan and Miscellaneous objects. If you have your own your geometry file, just type the name of geometry file with its extension (PLY).

Note: Make sure that TALOS can only read the PLY file format in ASCII[8]. Some PLY files obtained from external sources have been found to have errors, such as missing vertices or vertices with coordinates that do not belong to any face. To make your custom PLY file correct,

[8] http://paulbourke.net/dataformats/ply/

or to convert another 3D structure file format into PLY, you can use some software such as MeshLab[9], Gmsh[10] or Autodesk Netfabb[11].

```
+=============================================================================+
|                                                                             |
|        TALOS by Hyungmin Jun (hyungminjun@outlook.com), MIT, Bathe Lab, 2018        |
|                                                                             |
+=============================================================================+

A. First input - Pre-defined 3D target geometries
================================================

   [ Platonic solids ]
     *1. Tetrahedron,   *2. Cube,   *3. Octahedron,   4. Dodecahedron,   5. Icosahedron

   [ Archimedean solids ]
      6. Cubeocta,                  7. Icosidodeca,             8. Rhombicubocta
      9. Snub Cube,                10. Truncated Cube,         11. Truncated Cubocta
     12. Truncated Dodeca,         13. Truncated Icosa,        14. Truncated Octa
    *15. Truncated Tetra

   [ Johnson solids ]
     16. Gyroelongated Penta Pyramid,            *17. Triangular Bipyramid
    *18. Penta Bipyramid,                         19. Gyroelongated Square Bipyramid
     20. Square Gyrobicupola,                     21. Penta Orthocupolarotunda
     22. Penta Orthobirotunda,                    23. Elongated Penta Gyrobicupola
     24  Elongated Penta Gyrobirotunda,           25. Gyroelongated Square Bicupola

   [ Catalan solids ]
     26. Rhombic Dodeca,         27. Rhombic Triaconta,       28. Deltoidal Icositetra
     29. Penta Icositetra,       30. Triakis Octa,            31. Disdyakis Dodeca
     32. Triakis Icosa,          33. Pentakis Dodecahe,       34. Tetrakis Hexa
     35. Triakis Tetra

   [ Miscellaneous polyhedra ]
     36. Twisted Tri Prism,      37. Heptagonal Bipyramid,    38. Enneagonal Trapezo
     39. Small Stell Dodeca,    *40. Rhombic Hexeconta

Select the number or type geometry file (*.ply) [Enter] :
```

Fig. 1 | The first parameter in TALOS software. The 40 pre-defined target geometries as the first input parameter of TALOS. Users can use the own their geometry with typing the geometry file name with its file extensions (PLY). The negative value as an input terminates this console application immediately.

The second input shown in Fig. 2 is to select the vertex-type; the FV and MV designs (Fig. 3). The following input is to choose the vertex connection layer (Fig. 4a). The final input is to select the minimum edge length which is assigned to the shortest edge and the other edges are scaled.

[9] http://meshlab.sourceforge.net/
[10] http://gmsh.info/
[11] https://www.netfabb.com/

With four inputs, TALOS runs and creates the new folder named ‘output’ where TALOS automatically generates the several outputs (Table 2).

```
B. Second input - Vertex design
===============================

1. Flat vertex
2. Mitered vertex

Select the number [Enter] :

C. Third input - vertex connection
==================================

             [sec ID]                  [sec ID]
   1.  @ @      4 3          2.  @-@      5 4
      @   @    5   2            =@   @=  =0   3=
      =@ @=    =0 1=              @-@      1 2

   [ Inner connection ]      [ Middle Connection ]


Select the number [Enter] :

D. Fourth input - Pre-defined minimum edge length
================================================

   1.  42 bp =  4 turn * 21 bp/turn ->  42 bp * 0.34 nm/bp = 14.28 nm
   2.  63 bp =  6 turn * 21 bp/turn ->  63 bp * 0.34 nm/bp = 21.42 nm
   3.  84 bp =  8 turn * 21 bp/turn ->  84 bp * 0.34 nm/bp = 28.56 nm
   4. 105 bp = 10 turn * 21 bp/turn -> 105 bp * 0.34 nm/bp = 35.70 nm
   5. 126 bp = 12 turn * 21 bp/turn -> 126 bp * 0.34 nm/bp = 42.84 nm

Select the number [Enter] :
```

Fig. 2 | Three inputs for the vertex design, the vertex connection and minimum edge lengths. The negative value as an input terminates TALOS immediately.

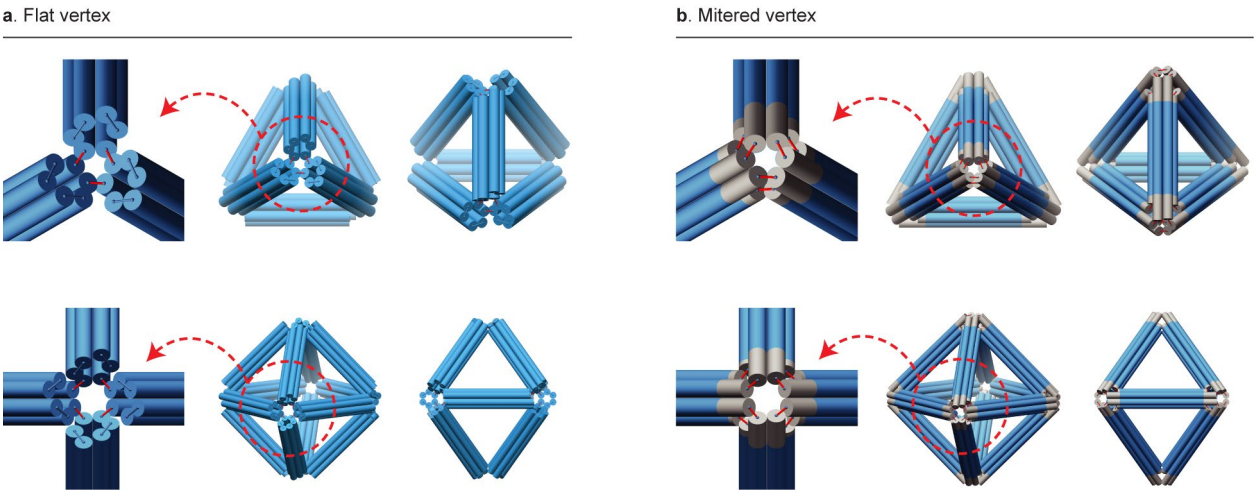

Fig. 3 | The second input parameter to choose the vertex design. a-b, FV (**a**) and MV (**b**) designs. Cylindrical models for 63-bp edge length tetrahedron (top) and 84-bp edge-length octahedron (bottom).

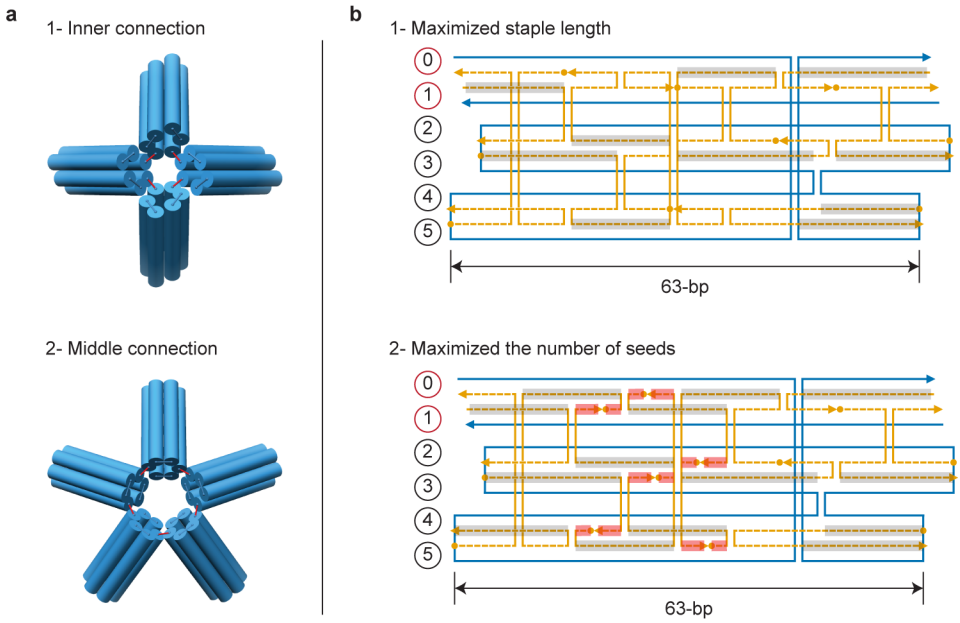

Fig. 4 | The third input parameter for the vertex connection. a, Inner (top) and middle (bottom) connection layers. **b**, Two staple-break rules; maximized staple length and maximized the number of seeds. The blue line represents the scaffold and orange lines indicate staples. The grey and red shaded regions represent the 14-nt and 4-nt dsDNA domain, respectively. The arrow indicates the 5’ to 3’-end direction of each strand.

Part 1.2. Running TALOS with a command prompt

TALOS can run through the command shell (command console / terminal). In Windows, start a command shell with **Start** → **run** → **cmd** (enter) or type **cmd** in Search Windows then use the 'cd' command to move to the folder where the TALOS package exists. To access the Unix command prompt in Mac, open the terminal application. It is located by default the Utilities folder, which in turn is inside the Applications folder. TALOS can be run with the following 4 arguments (Table 2) in the command shell.

```
TALOS.exe argc1 argc2 argc3 argc4          on Windows
./TALOS-Terminal argc1 argc2 argc3 argc4   on macOS
```

Table 2. Command line arguments for TALOS.

Arguments		Descriptions
argc1	String	The file name of the target geometry (including the file extension) Ex) tetrahedron.ply / from 0 to 40 (to select the pre-defined geometry)
argc2	Integer	The vertex-type (see Fig. 3) • 1: FV • 2: MV
argc3	Integer	The vertex connection layer (see Fig. 4a) • 1: Inner connection layer • 2: Middle connection layer
Opt4	Integer	The minimum edge length, which is any number but greater than 42-bp, to have at least two double-crossover per edge. Ex) 1: 42-bp, 2: 63-bp, 3:84-bp, 4:105-bp, and 5:126-bp

For example, the MV octahedron (inner connection) of 42-bp edge-length can be generated by the following command:

```
TALOS.exe octahedron.ply 2 1 42     for Windows
./TALOS octahedron.ply 2 1 42       for macOS
```

Users can run TALOS.EXE through the command shell on Mac and Linux environments, after installing Wine[12] which is a free and open-source compatibility layer that enables software

[12] https://www.winehq.org/

developed for Microsoft Windows to run on Unix-like operating systems. TALOS has been tested to run successfully using Wine on Mac and Linux systems.

Part 2. Compiling the source code

You can download the source code TALOS in zip format from

https://github.com/lcbb/talos/archive/master.zip

or browse the codes on GitHub,

http://github.com/lcbb/talos

You can also clone the project with Git[13] by running:

```
$ git clone https://github.com/lcbb/talos.git
```

The source codes for this project were written in Fortran 90/95. Fortran is a general-purpose, imperative programming language that is particularly well-suited to numeric computation and scientific computing. It is also stable and fast in high performance computing and simulations. In order to compile Fortran source codes, the user can install a Fortran compiler such as gFortran, Intel Fortran, PGI (formerly The Portland Group, Inc) Fortran. gFortran is developed under the GNU Fortran project, which provides a free Fortran 95/2003/2008 compiler for GCC (GNU Compiler Collection). Intel(R) Fortran Compiler known as IFORT was developed by Intel and available for Linux, Windows and macOS. We have developed this project under Intel(R) Fortran Compiler which is available under a free, non-commercial license for qualified students, educators, academic researchers and open source contributors on Linux, OS X and Windows[14]. Before installing Intel(R) Fortran Compiler, the user must have a version of Microsoft Visual Studio installed since the Intel Fortran Compiler integrates into the following versions of Microsoft Visual Studio: Visual Studio 2012 to 2015. Microsoft Visual Studio Community is also free for non-commercial use and it can be downloaded from here[15]. Note that if the installer does not find a supported version of Visual Studio (If the user does not install Visual Studio), a Fortran-only development environment based on the Microsoft Visual Studio 2013 Shell is provided (thus, TALOS can only be compiled using command-mode).

Here, under Windows and macOS systems, we explain how to compile the source codes of TALOS in the following two ways:

- Compiling source codes on command (see Part 2.1)
    - The source codes of TALOS can easily compiled by the build automation tool under Linux and Mac.

[13] https://git-scm.com/
[14] https://software.intel.com/en-us/qualify-for-free-software
[15] https://www.visualstudio.com/vs/community/

- Compiling sources through Visual Studio IDE (Integrated Development Environment) – Window only (see Part 2.2).
    - This provides comprehensive facilities to computer programmers for software development such as a source code editor, build automation tools, a debugger, etc. Microsoft Visual Studio is IDE for Fortran compiler, which can run only under Windows operating system. Mac users can use the alternative IDE, Xcode which supports Intel Fortran.

Part 2.1. Compiling source codes on command

'Makefile' is a simple way to organize or control code compilation. If already installed, the Apple developer tools on Mac can be used with the 'make' command at a terminal. Windows supports a variation of 'makefiles' with its 'nmake' utility. If we have a version of Microsoft Visual Studio installed, one can use NMAKE in Visual Studio Command Prompt to run 'Makefiles'.

(*Alternatives*) The GnuWin32 project provides Win32-version of GNU tools, much of it modified to run on the 32-bit Windows platform. The user can download the Windows version of MAKE from Gnuwin32 project[16]. The easiest way to use the tools is to add them to your search path using the `PATH' environment variable, usually by prepending the /bin folder to your PATH variable.

We provide 'Makefile' in the folder called 'make/makefiles'. User should copy the 'Makefile' file to the location where the source codes exist. The source code is located in the folder named "src". Follow these steps to invoke the compiler using command line:

1. For Windows, open the **Start** menu, and under the 'Intel Parallel Studio XE product group', select a compiler command prompt.

   ex) **Start** → **Program** → Intel Parallel Studio XE 2015 / 2016 / 2017 → Compiler 17.0 Update 1 for IA-32 Visual Studio 2015 environment.

   For macOS, open **Terminal**.

2. Use the **cd** command to move in the folder named 'src'.
3. Copy the 'MakeFile' in the 'src' folder.
4. Type '**make**' to invoke the compiler using 'Makefile'.
5. After compiling sources, you will have the executable file named 'TALOS.EXE' for Windows and 'TALOS' for Mac.
6. To delete object files generated during the compilation, type '**make clean**'.
7. Make sure that the 'env.TXT' file must be at the same folder where the executable file exist, to run TALOS.
8. You can open and modify source codes (*.F90) by the general text editor such as Sublime Text, Notepad++, Vim, Atom, Nano, Emacs, and etc.

We have successfully compiled the source codes of TALOS under RedHat Linux using the Linux version of the Intel Fortran Compiler.

[16] http://gnuwin32.sourceforge.net/packages/make.htm.

Part 2.2. Compiling sources on Microsoft Visual Studio

First, check Microsoft Visual Studio version supported by versions of the Intel compilers. This project was developed under the Intel Parallel Studio XE 2015 / 2016 / 2107 with Microsoft Visual Studio 2015 / 2016 / 2017.

- Launch Microsoft Visual Studio.
- Select **File** > **New** > **Project** to make new project
- In the New Project window, select Empty project under **Intel(R) Visual Fortran**.
- Copy all source files and one text file, 'env.TXT' into project folder and added these in project directory
- Select **Build** > **Build Solution (F7)**
- Select **Debug** > **Starting Without Debugging (Ctrl + F5)**
- The results of the compilation display in the **Output** window

Part 3. Outputs

Once the sequence design from TALOS is completed, the output folder is created. The files, 'TXT_TALOS.txt' and 'TXT_Sequence.txt', contain information on all events for sequence design process and the results of the sequence and routing of the scaffold and staples, respectively. The file, '_17_sequence.CSV', contain staple sequences generated with the provided scaffold sequence. Several BILD files are in ASCII format that describes lines, polygons, and geometric primitives for the visualization of the geometry, routing, strands, edge-staple, and so on. User will be also able to visualize these set of data by UCSF Chimera (Figs 5,6, Table 2).

With the JSON file as one of outputs from TALOS, the user can edit the staple crossover positions and sequences using caDNAno (Fig. 7). The file named '_14_json.guide.BILD' can be loaded in USCF Chimera, which gives the information on edges of the target structure associated with cross-sections in the caDNAno representation.

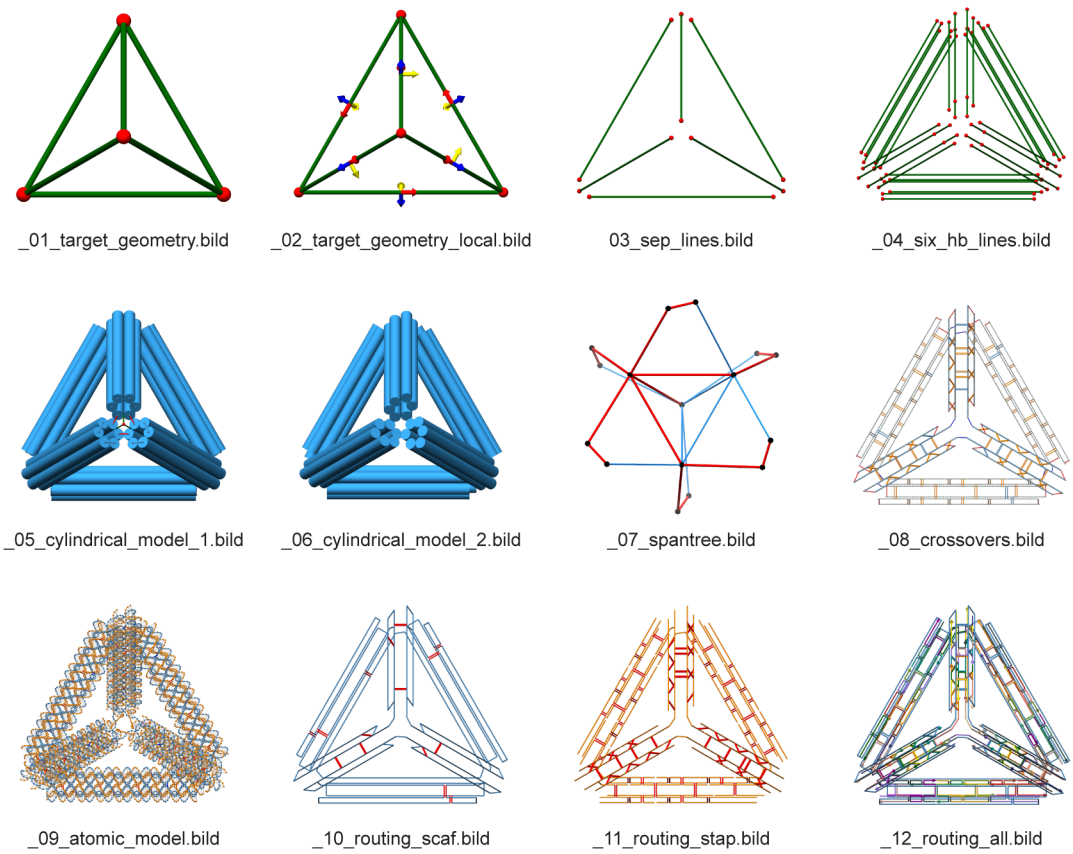

Fig. 5 | 12 rendering from BILD outputs for the FV tetrahedron of 84-bp edge-length.

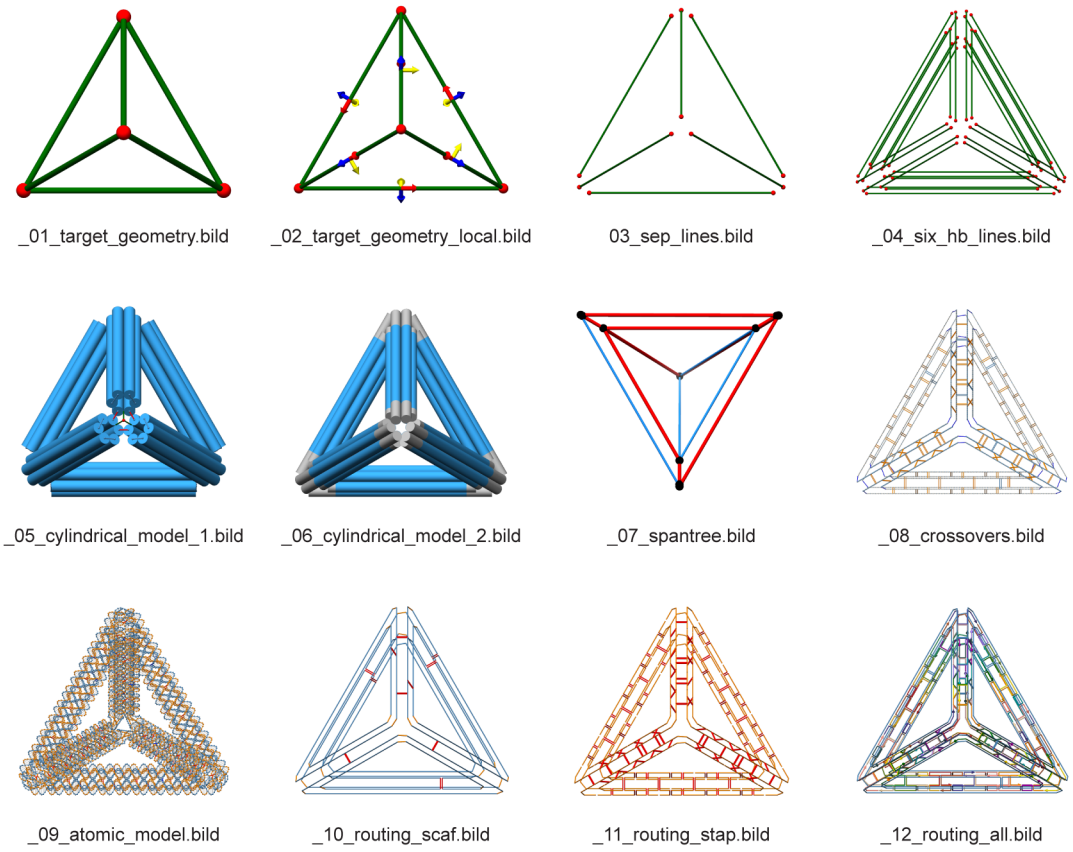

Fig. 6 | 12 rendering from BILD outputs for the MV tetrahedron of 84-bp edge-length.

Using the CNDO (CanDo file format) file, which was designed to describe DNA nanostructures, contains sufficient information to generate the all-atom models of these DNA nanostructures. The atomic model generator[17] uses the CNDO file as its input and creates the PDB file consisting of two phosphates, two deoxyriboses, and two paired bases (Fig. 8). This atomic model generator is written using a MATLAB script that produces a *.PDB file, which can be visualized similarly using UCSF Chimera.

[17] https://cando-dna-origami.org/atomic-model-generator/

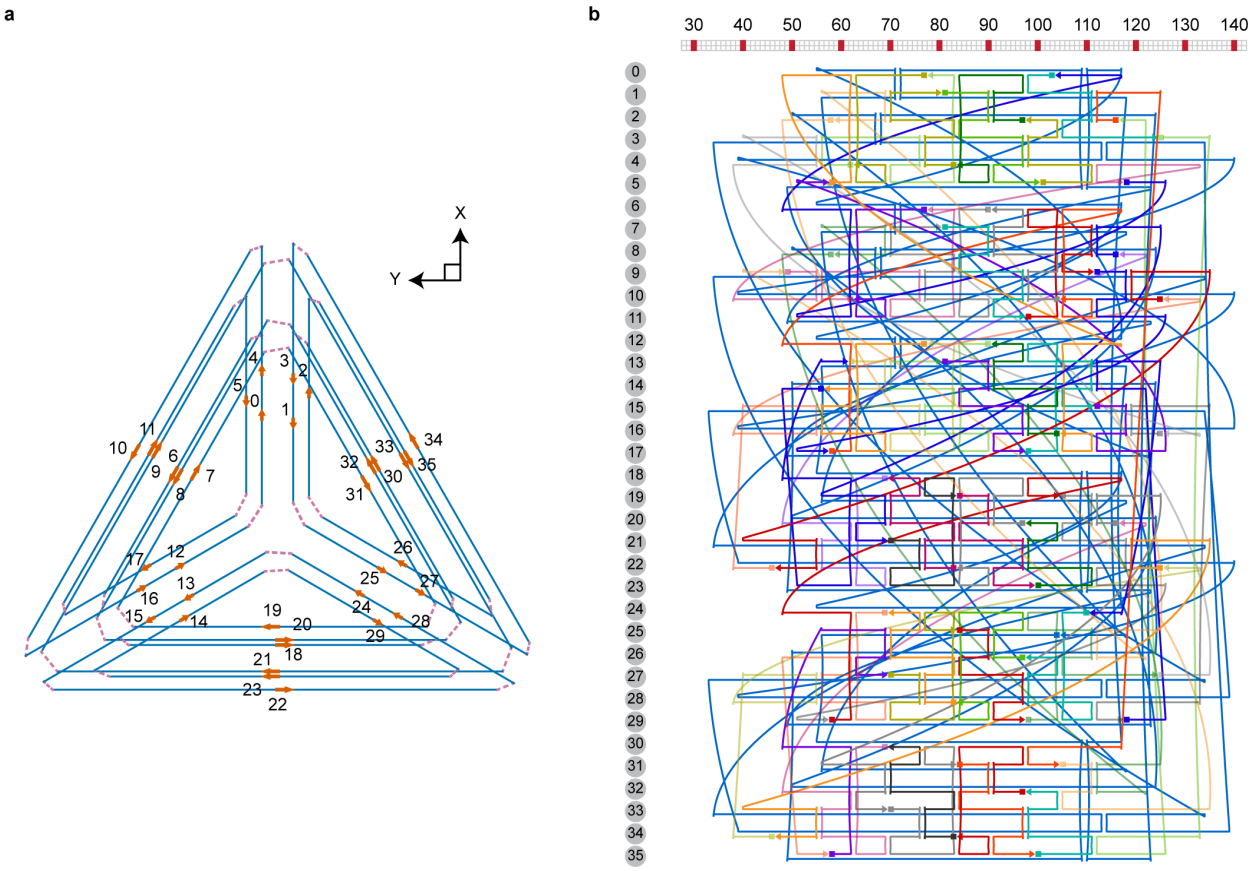

Fig. 7 | JSON caDNAno and guide BILD outputs. a, JSON guide model in which the edge numbers are associated with the cross-section number in caDNAno. **b,** Staple and scaffold organization from caDNAno.

Table 3. Descriptions of 14 BILD outputs generated by TALOS.

BILD file	Colored object	Description
_01_target_geometry	Red circle	Point of the target geometry
	Green edge	Edge of the target geometry
_02_target_geometry_local	Red circle	Point of the target geometry
	Green edge	Edge of the target geometry
	Red arrow	Local vector, $\mathbf{t}_1$
	Yellow arrow	Local vector, $\mathbf{t}_3$
	Blue arrow	Local vector, $\mathbf{t}_2$
_03_sep_lines	Red circle	Point separated from the vertex
	Green line	Line connecting two points
_04_six_hb_lines	Red circle	End point of the double helix

	Green line	Double helix DNA strand
_05_cylindrical_model_1 / _06_cylindrical_model_2	Cylinder	Double helix DNA strand
	Grey cylinder	Extended parts to fill the gap
	Red line	Scaffold strand crossing the vertex
_07_spantree	Black circle	Node of the dual graph
	Blue line	Non-member of the spanning tree
	Red line	Member of the spanning tree
_08_crossovers	Yellow circle	Base pair
	Blue line	Scaffold crossover
	Orange line	Staple crossover
	Dark blue line	Scaffold crossing the vertex
_09_atomic_model	Blue line	Scaffold strand
	Orange line	Staple
_10_routing_scaf	Blue line	Scaffold strand
	Red line	Scaffold crossover
_11_routing_stap	Orange line	Staple
	Red line	Staple crossover
_12_routing_all	Blue line	Scaffold strand
	Multiple colored line	Staple strand
_13_cylindrical_model_xover	Blue band	Scaffold double-crossover
	Orange band	Staple double-crossover
_14_json_guide	Blue line	Scaffold strand
	Red arrow	Scaffold direction, 5’ to 3’-end
	Number	Helix number that is related to the cross-sectional number in caDNAno

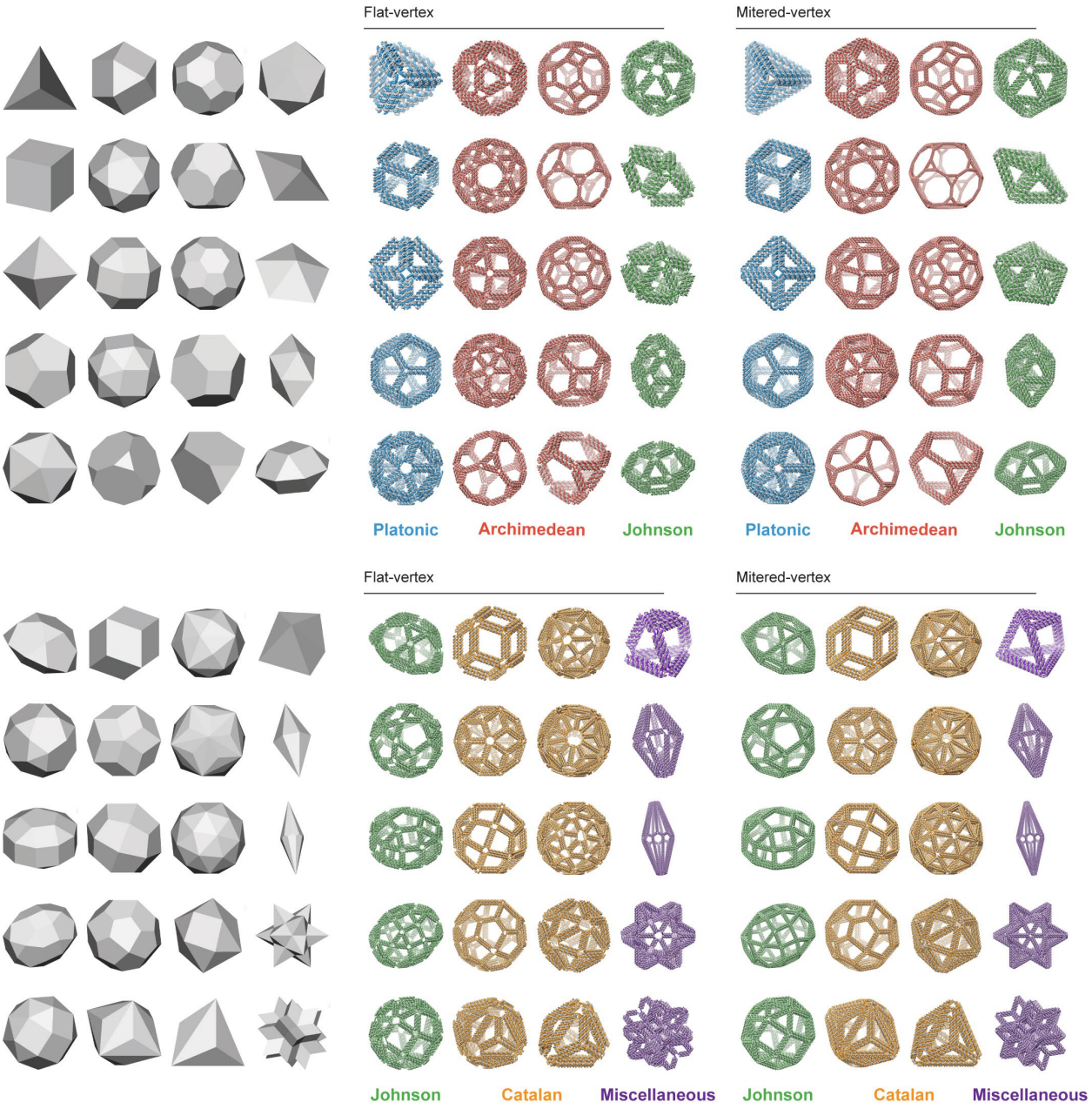

Fig. 8 | Target geometry and atomic model of 40 diverse wireframe nanoparticles generated by TALOS.