

PERDIX-6P

(Purine / pyrimidine Engineering Routing Design Inverse X – 6HB Particle)

Software Instructions and Demo

Dr. Hyungmin Jun

Laboratory of Computational Biology and Biophysics, MIT

Copyright 2018. Massachusetts Institute of Technology. Rights Reserved. M.I.T. hereby makes following copyrightable material available to the public under GNU General Public License, version 3 (GPL-3.0). A copy of this license is available at <https://opensource.org/licenses/GPL-3.0>.

An online version of this software is available at <https://github.com/lcbb/PERDIX-6P>.

Table of Contents

Welcome to PERDIX-6P!	3
Part 1. Release package	4
Part 1.1. Opening the PERDIX-6P software by double-clicking	5
Part 1.2. Running PERDIX-6P with command prompt	8
Part 2. Compiling source code	9
Part 2.1. Compiling source codes on command	10
Part 2.2. Compiling sources on Microsoft Visual Studio	11
Part 3. Outputs	12

Welcome to PERDIX-6P!

PERDIX-6P simplifies and enhances the process of designing 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 PLY (“Polygon File Format”) file of your design that describes your target 3D geometry, PERDIX-6P 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 PERDIX-6P 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 format¹) from PERDIX-6P can be used to predict the flexibility of programmed DNA nanoparticles². 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 PERDIX-6P. With software such as PyMOL³, VMD⁴, UCSF Chimera⁵, etc., you will be able to visualize and manipulate your atomic model in 3D space.
- Several BILD files. These BILD files are used for visualizing the target geometry, scaffold routing, staple paths and cylindrical models, which are rendered by lines, polygons, and geometric primitives built in UCSF Chimera (see Fig. 5-6).
- A JSON file. This JSON file can be imported into caDNAno⁷ software that allows to edit the staple paths and sequences.

One of the goals of this software is to broaden the usage of DNA nanotechnology to the larger community. We hope that, even if you are not an expert on DNA origami, you can use PERDIX-6P to begin to explore the capabilities of this powerful molecular design paradigm.

PERDIX-6P features:

- Fully automatic procedure of the sequence design for scaffold DNA 6HB based wireframe particles
- Two vertex designs; flat and mitered vertices
- Importing PLY file formats
- JSON output for editing staple path and sequences from caDNAno
- 3D visualization powered by UCSF Chimera

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/>

- 45 pre-defined target geometries
- User-friendly TUI (Text-based User Interface)
- Executable files on Windows and Mac OS X systems
- Free and open source (GNU General Public License, version 3.0)

PERDIX variants:

- PERDIX-6P – Designer scaffolded DNA 6HB-based wireframe nanoparticles
- PERDIX-2L – Designer scaffolded DNA DX-based wireframe lattices

Part 1. Release package

The release package (as an executable file) is available for Windows and Mac OS X systems. These executable files are the console application for Windows (PERDIX-6P) and Mac OS X (PERDIX-6P), which accept an input (PLY) and generate outputs to the console through the command prompt. You can download PERDIX-6P from

<https://github.com/lcbb/PERDIX-6P/archive/master.zip>

and find the release packages in the folder named as ‘release’ after extracting zip file. The current version of the release package is compiled with Intel (R) Visual Fortran compiler (Ver. 17.0.1.143) under 64-bit Microsoft Windows 10 with Intel(R) Core(TM) i7-4470 CPU @ 3.40GHz and Intel (R) Visual Fortran compiler (Ver. 18.0.1) under macOS 10.13 High Sierra, respectively. If you are a user on a Linux system, you will need to download source codes and compile them properly under your OS (see Part 3).

- File ‘Win/PERDIX-6P.exe’: This is an executable file to run PERDIX-6P under Microsoft Windows. The executable file can be run by double-clicking the icon (Part 2.1) or on command shell (Part 2.2).
- File ‘Mac/PERDIX-6P-EXE’: This is an executable file to run PERDIX-6P under Mac OS X by double-clicking. (Part 2.1).
- File ‘Mac/PERDIX-6P’: This is an executable file to run PERDIX-6P under Mac OS X. The executable file can be open using 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 **Error! Reference source not found.**).
- Folder ‘input’: The user-defined geometry file (PLY) must be here.

Table 1. The sequences must be defined at third line and the ‘para_scaf_seq’ must be defined as 1 to use user-defined scaffold sequence.

Field	Value	Descriptions
-------	-------	--------------

para_platform	win mac	Depending on user's operating system
para_scaf_seq	0 1 2	Scaffold sequence <ul style="list-style-type: none"> • 0: M13mp18(7249nt) sequence • 1: User-defined sequence • 2: randomly generated sequence

Part 1.1. Opening the PERDIX-6P software by double-clicking

The easiest way to run PERDIX-6P is to double-click ‘PERDIX-6P.EXE’ in the file manager of the Windows, or ‘PERDIX-6P-EXE’ in the Finder of the Mac. Note that the text file ‘env.TXT’ should be in the same folder in order to properly run the software. The user-defined geometry file (PLY) should be in the folder named as the ‘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 45 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 PERDIX-6P can only read the PLY file format in ASCII⁸ (If you open the PLY file externally, it should be human-readable). Some PLY files obtained from external sources have been found to have errors, like missing vertices or vertices with coordinates that do not belong to any face. To make your custom PLY file correct, or to convert another 3D structure file format into PLY, you can use some software such as MeshLab⁹, Gmsh¹⁰ or Autodesk Netfabb¹¹.

⁸ <http://paulbourke.net/dataformats/ply/>

⁹ <http://meshlab.sourceforge.net/>

¹⁰ <http://gmsh.info/>

¹¹ <https://www.netfabb.com/>

```
+=====
| PERDIX-6P 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. Heptagonal Bipyramid,      37. Enneagonal Trapezo,    38. Small Stell Dodeca
  #39. Rhombic Hexeconta,       40. Goldberg dk5dgD,       *41. Double Helix
  42. Nested Cube,              *43. Nested Octa,          *44. Torus
  45. Double Torus

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

Fig. 1 | The first parameter in PERDIX-6P software. The 45 pre-defined target geometries as the first input parameter of PERDIX-6P. Users can use their own geometry with typing the geometry file name with 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 types; the flat and mitered vertex designs (Fig. 3). The following input is to choose the vertex connection (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.

Even though each edge length should be a multiple of 21-bp with a minimum of 42-bp in the design with the flat vertex, there is no limitation for edge lengths if the length of edge is over 42-bp, for the mitered vertex design. Thus, user can directly type the any arbitrary edge length that should be larger than 42 as the fourth input. With four inputs, it runs and creates the new folder named as ‘output’ where PERDIX-6P automatically generates the several outputs (Table 3).

```

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 PERDIX-6P immediately.

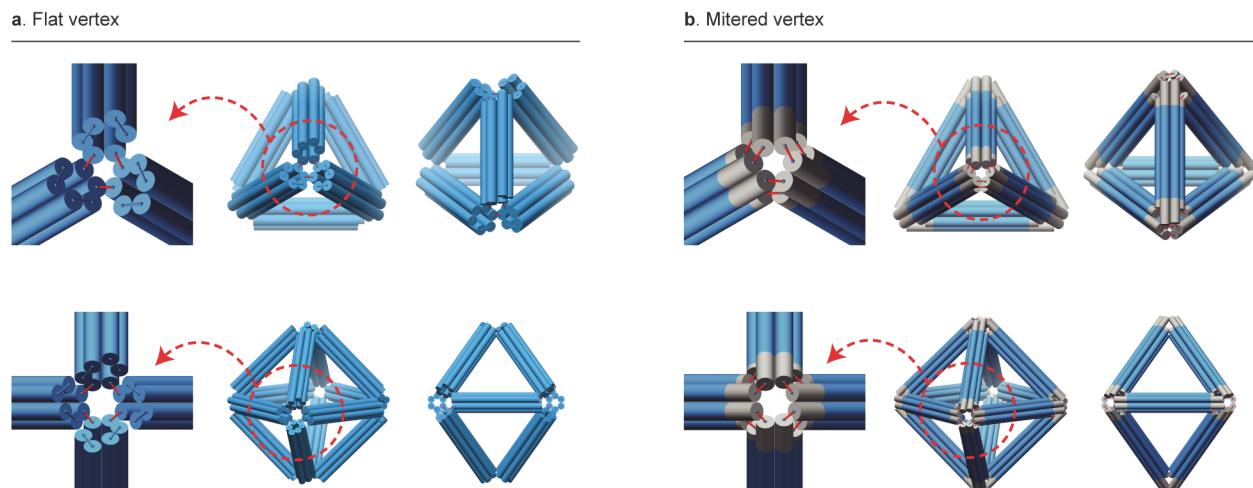


Fig. 3 | The second input parameter to choose the vertex design. a-b, Flat vertex (a) and mitered vertex (b) designs. Cylindrical models for 63-bp edge length tetrahedron (top) and 84-bp edge-length octahedron (bottom).

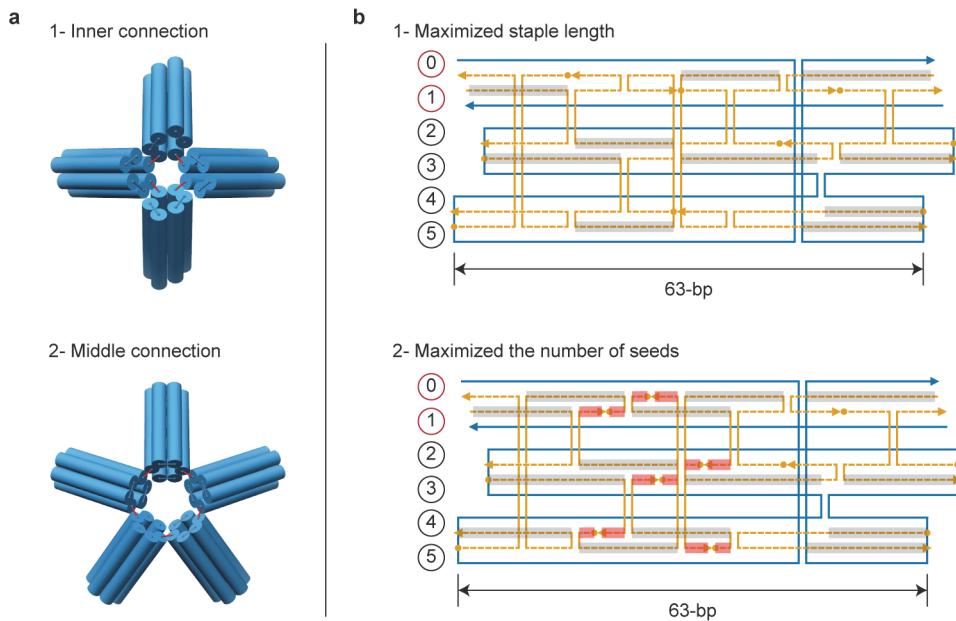


Fig. 4 | The third input parameter for the vertex connection. **a**, Inner (top) and middle (bottom) connections. **b**, Two staple-break rules; Maximized staple length and maximized the number of seeds. The blue line represents the scaffold and orange lines indicate staples. The grey and red shaded 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 PERDIX-6P with command prompt

PERDIX-6P 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 PERDIX-6P package exists. To access the Unix command prompt in Mac OS X, open the terminal application. It is located by default the Utilities folder, which in turn is inside the Applications folder. PERDIX-6P can be run with the following 4 arguments (Table 2) from the command shell.

`PERDIX-6P.exe argc1 argc2 argc3 argc4 argc5` for Windows

`./PERDIX-6P argc1 argc2 argc3 argc4 argc5` for Mac OS X

Table 2. Command line arguments for PERDIX-6P.

Parameter	Descriptions
<code>argc1</code>	String The file name of the target geometry (including the file extension) Ex) tetrahedron.ply / from 0 to 45 (to select the pre-defined geometry)

<code>argc2</code>	Integer	The vertex design (see Fig. 3) <ul style="list-style-type: none"> • 1: Flat vertex • 2: Mitered vertex
<code>argc3</code>	Integer	The vertex connection (see Fig. 4a) <ul style="list-style-type: none"> • 1: Inner connection • 2: Middle connection
<code>Opt4</code>	Integer	The minimum edge length, which is any number but greater than 37-bp, to have at least two double-crossover per edge. Ex) 42 – 42-bp as minimum edge length
<code>Opt5</code>	String	The staple-break rule (see Fig. 4b) <ul style="list-style-type: none"> • max: Maximized staple length • opt: Maximized the number of seeds

For example, the octahedron with 42-bp, inner connection, mitered vertex, and maximized the number of seeds can be generated by the command as below:

`PERDIX-6P.exe octahedron.ply 2 1 42 opt` for Windows

`./PERDIX-6P octahedron.ply 2 1 42 opt` for Mac OS X

User can run PERDIX-6P.EXE through the command shell on Mac and Linux environments, after installing Wine¹² which is a free and open-source compatibility layer that aims to allow computer programs developed for Microsoft Windows to run on Unix-like operating systems. We successfully tested PERDIX-6P working with Wine on Mac and Linux systems.

Part 2. Compiling source code

You can download the source code PERDIX-6P in zip format from

<https://github.com/lcbb/PERDIX-6P/archive/master.zip>

or browse the codes on GitHub,

<http://github.com/lcbb/PERDIX-6P>

You can also clone the project with Git¹³ by running:

`$ git clone https://github.com/lcbb/PERDIX-6P.git`

¹² <https://www.winehq.org/>

¹³ <https://git-scm.com/>

The source codes for this project were written in Fortran 90/95. Fortran is a general-purpose, imperative programming language that is especially 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, you can install the 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 Mac OS X. 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 Windows¹⁴. Before installing Intel(R) Fortran Compiler, you 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 here¹⁵. Note that if the installer does not find a supported version of Visual Studio (If you do not install Visual Studio), a Fortran-only development environment based on the Microsoft Visual Studio 2013 Shell is provided (thus, PERDIX-6P can only be compiled on command).

Here, under Windows system and Mac OS X, we will explain how to compile the source codes of PERDIX-6P in two ways as follow:

- Compiling source codes on command (see Part 2.1)
 - The source codes of PERDIX-6P can easily compiled by the build automation tool under Linux and Mac OS X.
- Compiling sources through Visual Studio IDE (Integrated Development Environment) – Window only (see Part 2.2).
 - It 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. The users for Linux and Mac can find the alternative IDE, Xcode which is support to Intel Fortran.

Part 2.1. Compiling source codes on command

‘Makefile’ is a simple way to organize or control code compilation. Windows supports a variation of ‘makefiles’ with its ‘nmake’ utility. If we have a version of Microsoft Visual Studio installed, we 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. You can download the Window version of ‘Make’ from Gnuwin32 project¹⁶. The easiest way to use the tools is to add them to your search path

¹⁴ <https://software.intel.com/en-us/qualify-for-free-software>

¹⁵ <https://www.visualstudio.com/vs/community/>

¹⁶ <http://gnuwin32.sourceforge.net/packages/make.htm>.

using the 'PATH' environment variable, usually by prepending the /bin folder to your PATH variable.

We provide 'Makefile' in the folder named as 'make/makefiles'. User should copy the 'Makefile' file to where the source codes exist. The source code is located at the folder named as "src". Follow these steps to invoke the compiler from the 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 Mac OS X, open **Terminal**.

2. Use the **cd** command to move in the folder named as '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 as 'PERDIX-6P.EXE' for Windows and 'PERDIX-6P' for Mac OS X.
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 PERDIX-6P.
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.

It is note that, by the same way mentioned above, we successfully compiled the source codes of PERDIX under RedHat Linux using the Linux version of the Intel Fortran Compiler.

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 2016 with Microsoft Visual Studio 2015.

- 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 PERDIX-6P is completed, the output folder is created. The files, ‘TXT_PERDIX_6P.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 generated sequences of the staples with the given sequence of the scaffold. Several BILD files are ASCII format that describes lines, polygons, and geometric primitives for the visualization of the geometry, routing, strands, edge-staple and so on. You will be also able to visualize these set of data by UCSF Chimera (Figs 5-6, and Table 3).

With the JSON file as one of outputs from PERDIX-6P, user can edit the staple crossover positions and sequences using caDNAno (Fig. 7).The file named as ‘_15_json.guide.BILD’ can be loaded in UCSF Chimera, which give the information which edges of the target structure is associated with the which cross-sections of caDNAno representation.

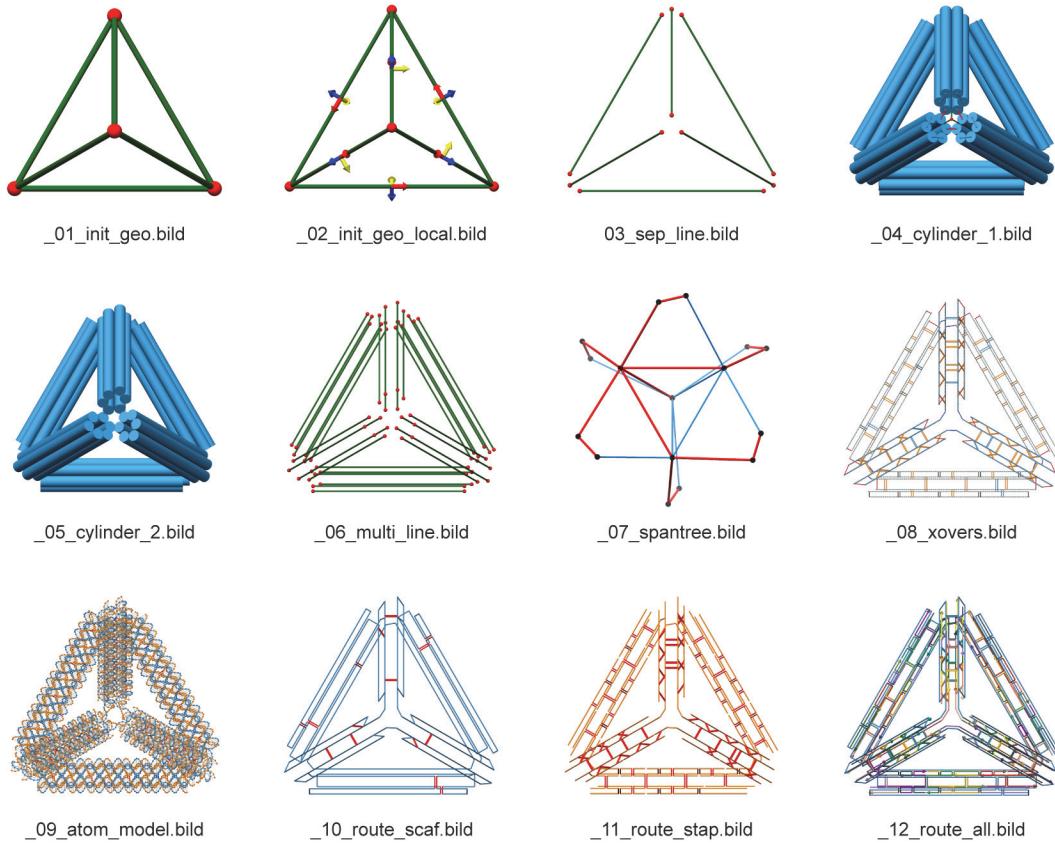


Fig. 5 | 12 rendering from BILD outputs for 84-bp edge length DNA tetrahedron with the flat vertex design.

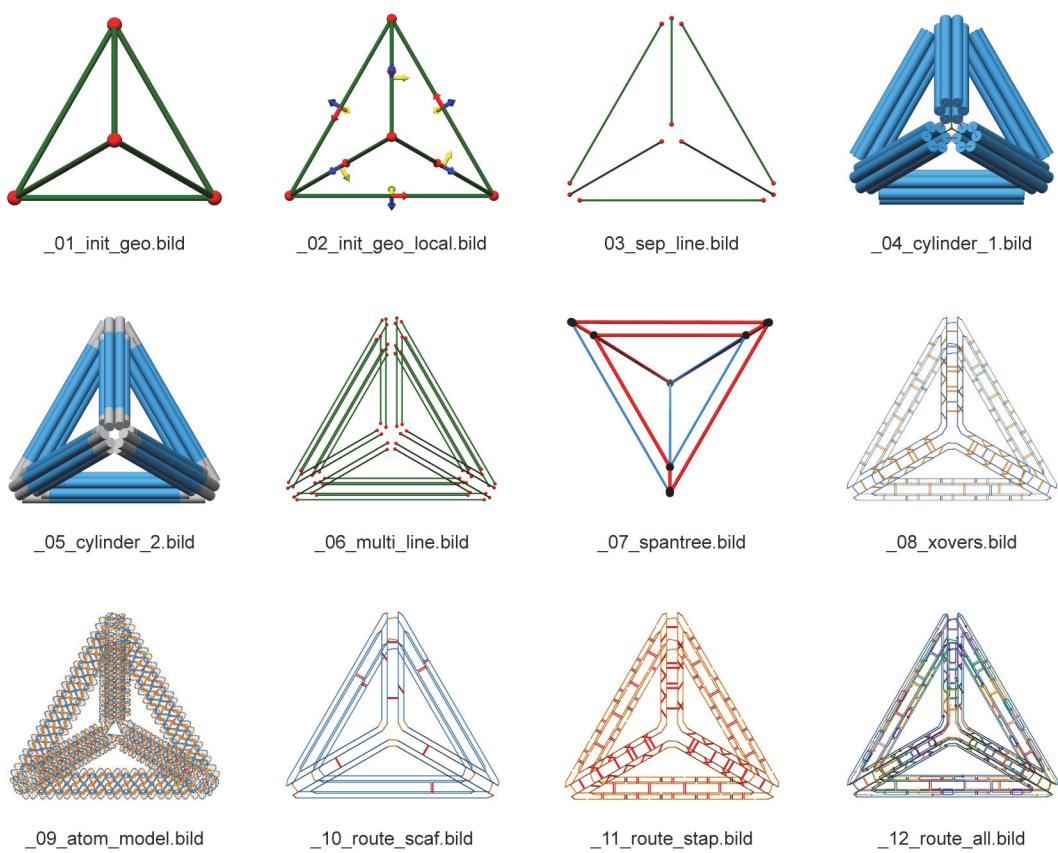


Fig. 6 | 12 rendering from BILD outputs for 84-bp edge length DNA tetrahedron with the mitered vertex design.

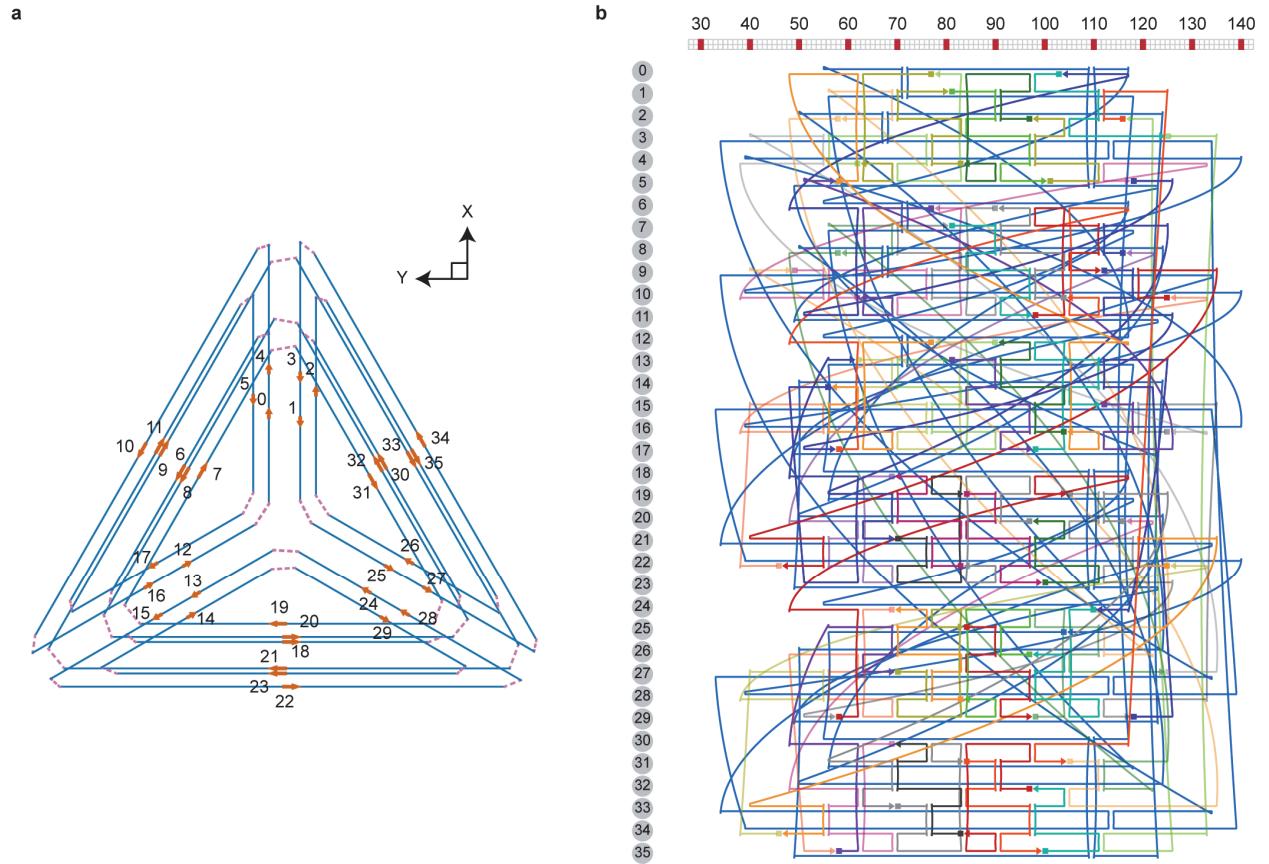


Fig. 7. JSON caDNAno and guide BILD outputs. a, JSON guide model in which the edge numbers are associated with the cross-section number in caDNAno. **b,** Staple and scaffold organization from caDNAno.

Table 3. Descriptions of 12 BILD outputs generated by PERDIX-6P.

BILD file	Colored object	Description
_01_init_geo	Red circle Green edge	Point of the target geometry Edge of the target geometry
_02_init_geo_local	Red circle Green edge Red arrow Yellow arrow Blue arrow	Point of the target geometry Edge of the target geometry Local vector, t_1 Local vector, t_3 Local vector, t_2
_03_sep_line	Red circle Green line	Point separated from the vertex Line connecting two points
_04_cylinder_1 /_05_cylinder_2	Blue cylinder Grey cylinder Red line	Double helix DNA strand Extended part to fill the gap Scaffold strand crossing the vertex

<u>_06_multi_line</u>	Red circle	End point of the double helix
	Green line	Double helix DNA strand
<u>_07_spantree</u>	Black circle	Node of the dual graph
	Blue line	Non-member of the spanning tree
<u>_08_xovers</u>	Red line	Member of the spanning tree
	Yellow circle	Base pair
<u>_08_xovers</u>	Blue line	Scaffold crossover
	Orange line	Staple crossover
<u>_09_atom_model</u>	Dark blue line	Scaffold crossing the vertex
	Blue line	Scaffold strand
<u>_10_route_scaf</u>	Orange line	Staple
	Red line	Scaffold crossover
<u>_11_route_stap</u>	Blue line	Scaffold strand
	Red line	Staple crossover
<u>_12_route_all</u>	Blue line	Scaffold strand
	Multiple colored line	Staple strand

With the CND0 (The 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 generator¹⁷ uses the CND0 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 by a MATLAB script which produces a PDB file, which can be similarly visualized using UCSF Chimera.

¹⁷ <https://cando-dna-origami.org/atomic-model-generator/>

PERDIX-6P Software Instructions and Demo

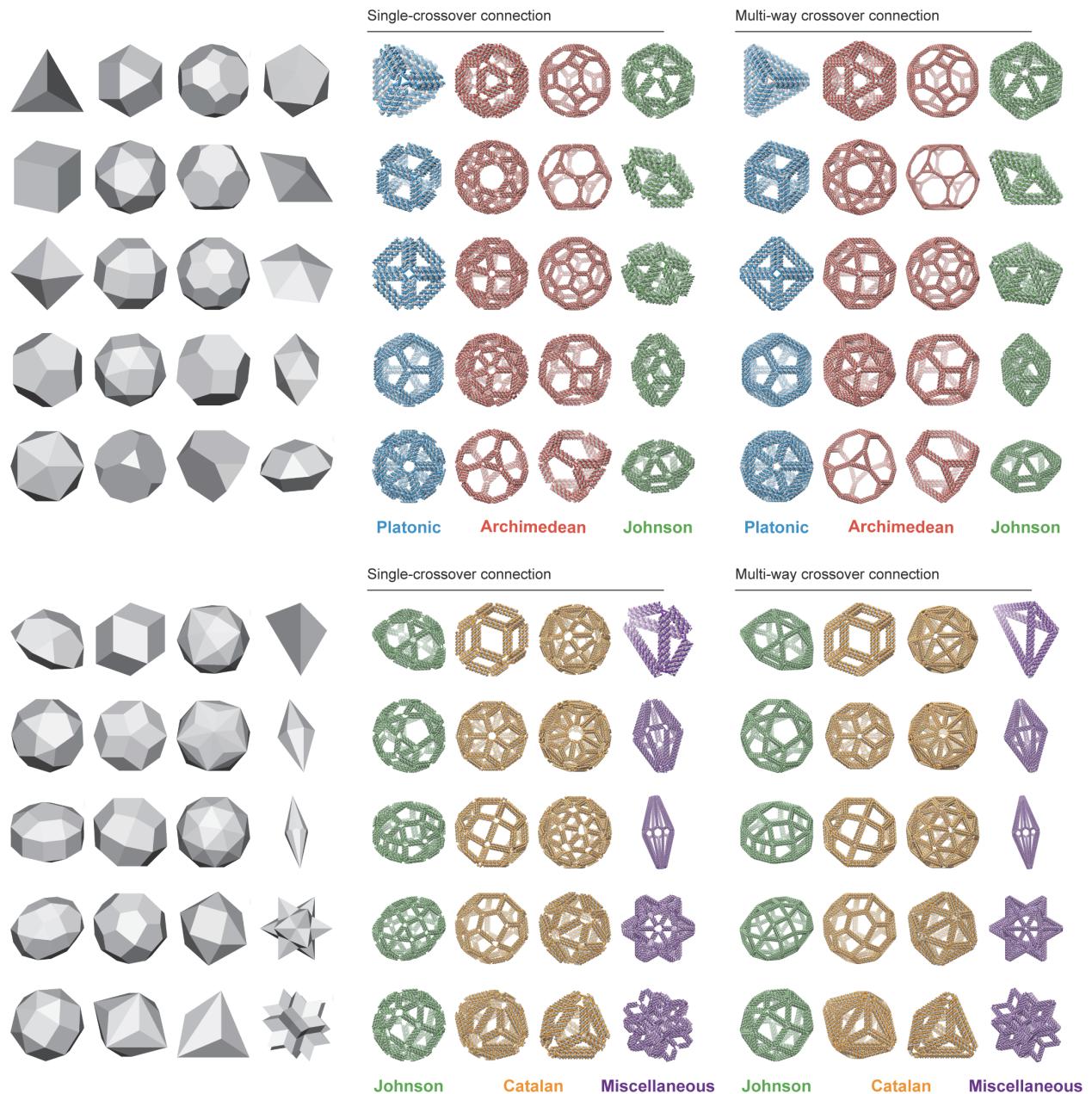


Fig. 8 | Atomic model of 45 diverse nanoparticles generated by PERDIX-6P.