

PERDIX-6P

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

Software Instructions and Demo

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/hmjeon/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 the .exe icon	4
Part 1.2. Running PERDIX-6P with command prompt	9
Part 2. Compiling source code	9
Part 2.1. Compiling sources on command	11
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 three-dimensional DNA origami nanoparticles from the CAD 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 BILD⁶ files. These BILD files are used for the visualization of the target geometry, scaffold routing, staple sequence and cylindrical model, which are rendered by lines, polygons, and geometric primitives built in UCSF Chimera (see **Fig. 5**).
- A JSON file. This output JSON file can be imported into caDNAno⁷ software that enable users to edit the staple connections and sequences. When opening the BILD file, ‘guild.bild’ in Chimera, it displays information on which edge of the target geometry is associated with which the section number in caDNAno software.

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 for the sequence design of the DNA nanoparticle composed of 6HB edges
- 3D visualization powered by UCSF Chimera
- JSON output for editing staples from caDNAno

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)
- Free and open source (GNU General Public License, version 3.0)
- Two vertex designs; “single-crossover” and “multi-way crossover” connection

Part 1. Release package

The release package (as an executable file .exe) is available only under Windows. This software is a Win32 console application that accepts input and sends output to the console through the command prompt. If you are a user on a Linux or Mac OS X system, you will need to download the source codes from

<https://github.com/hmjeon/PERDIX-6P>

and compile it properly under your OS (see Part 2). The current version of the 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. You can download the release package directly below:

[https://github.com/hmjeon/PERDIX-6P/files/1830330/released.zip.](https://github.com/hmjeon/PERDIX-6P/files/1830330/released.zip)

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

After download, unzip the file, ‘PERDIX-6P.zip’. In the folder named ‘PERDIX-6P’, you will have the following subfolders and files:

- File ‘PERDIX-6P.exe’: This is an executable file to run PERDIX-6P under Microsoft Windows. Opening the software requires double-clicking the icon.
- File ‘env.txt’: This text file contains a set of environment variables that can affect the way running processes of PERDIX-6P (see **Table 1**).
- File ‘seq.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.
- Folder ‘/Examples’: This folder contains the output files generated by PERDIX-6P from the tetrahedron with 63-bp edge length (for single-crossover and multi-way crossover connection vertex design).
- Folder ‘/PLY_Files’: This folder contains the 45 geometries rendered as scaffolded DNA origami in the manuscript that this software accompanies. When running PERDIX-6P yourself, you may use these PLY files as inputs.

Table 1. The description of environment variables in the file, ‘env.dat’. The default is the first value in bold.

Field	Value	Descriptions
para_ext	on off	It should be always “on” to use parameters defined in the file, ‘env.dat’.
para_stap_break	max opt	<p>Staple-break rule</p> <ul style="list-style-type: none"> • max: maximized staple length • opt: maximized the number of 14-nt seeds
para_scaf_seq	0 1 2	<p>Scaffold sequence</p> <ul style="list-style-type: none"> • 0: M13mp18(7249nt) sequence • 1: sequence from the file, ‘seq.txt’ • 2: randomly generated sequence

The easiest way to run PERDIX-6P is to double-click the icon or file name of ‘PERDIX-6P.exe’ in the file manager of the Window operating systems (It can be also run by command prompt (see **Part 1.2**)). Note that the three files, ‘PERDIX-6P.exe’, ‘env.txt’, and ‘seq.txt’ should be in the same folder in order to properly run the software (the geometry file, ‘*.ply’ should also be in the same folder). By double-clicking the icon, you will see the TUI (Text based User Interface) by the Win32 console, which displays the pre-defined target geometries as first input parameters (**Fig. 1**). There are pre-defined 45 geometries including the Platonic and Archimedean, Johnson, Catalan solids and miscellaneous structures. If you have your own PLY file specifying the target geometry, just type the name of geometric file you designed with the file 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 Software Instructions and Demo

```
+=====+  
| PERDIX-6P by Hyungmin Jun (hyungminjun@outlook.com), MIT, Bathe Lab, 2018 |  
+=====+  
A. First input – Geometry discretized by surface mesh  
======  
  
I - Platonic solids  
-----  
1. Tetrahedron, 2. Cube, 3. Octahedron, 4. Dodecahedron, 5. Icosahedron  
  
II - Archimedean solids  
-----  
6. Cubeoctahedron, 7. Icosidodecahedron, 8. Rhombicuboctahedron  
9. Snub Cube, 10. Truncated Cube, 11. Truncated Cuboctahedron  
12. Truncated Dodecahedron, 13. Truncated Icosahedron, 14. Truncated Octahedron  
15. Truncated Tetrahedron  
  
III - Johnson solids  
-----  
16. Gyroelongated Pentagonal Pyramid, 17. Triangular Bipyramid  
18. Pentagonal Bipyramid, 19. Gyroelongated Square Bipyramid  
20. Square Gyrobicupola, 21. Pentagonal Orthocupolarotunda  
22. Pentagonal Orthobirotunda, 23. Elongated Pentagonal Gyrobicupola  
24. Elongated Pentagonal Gyrobicupola, 25. Gyroelongated Square Bicupola  
  
IV - Catalan solids  
-----  
26. Rhombic Dodecahedron, 27. Rhombic Triacontahedron, 28. Deltoidal Icositetrahedron  
29. Pentagonal Icositetrahedron, 30. Triakis Octahedron, 31. Disdyakis Dodecahedron  
32. Triakis Icosahedron, 33. Pentakis Dodecahedron, 34. Tetrakis Hexahedron  
35. Triakis Tetrahedron  
  
V - Miscellaneous polyhedra  
-----  
36. Heptagonal Bipyramid, 37. Enneagonal Trapezohedron, 38. Small Stell Dodecahedron  
39. Rhombic Hexecontahedron, 40. Goldberg dk5dg0, 41. Double Helix  
42. Nested Cube, 43. Nested Octahedron, 44. Torus  
45. Double Torus  
  
Select the number or type the geometry file (*.ply) [Enter] :
```

Fig. 1. The first interface user can see when opening the PERDIX-6P software. The 45 pre-defined target geometries as first input parameter of PERDIX-6P. Users can use their surfaced geometry with typing the geometry file name (PLY). The negative value as input terminates this console application immediately.

Once pressing the “Enter” key with the first input parameter, the console application will wait for the following input parameters (**Fig. 2**). The second input parameter is to determine the design type of the vertex called the “single-crossover connection” and “multi-way crossover connection” (**Fig. 3**). The third input is to choose the position of the origin of the local coordinate system (**Fig. 4a**). The final input is to select minimum edge lengths which is assigned to the shortest edge and the other edges are scaled.

Even though the length of each edge should be a multiple of 21-bp with a minimum of 42-bp in the design from the “single-crossover connection” vertex, there is no limitation for edge lengths if the length of edge is over 42-bp, for the “multi-way crossover connection” vertex design. Thus, user can directly type the number as the fourth input. With four inputs, it runs and creates the new folder named as ‘output’ where it automatically generates the output files for the scaffold route and sequence design.

```

B. Second input – Vertex design
=====
1. Single-crossover connection
2. Multi-way crossover connection

Select the number [Enter] :

C. Third input – Origin of the local coordinates
=====

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

[ Bottom origin ] [ Middle origin ]

Select the number [Enter] :

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

[Honeycomb lattice]

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. Displayed three input parameters for the vertex design, the position of the origin of the local coordinate system and minimum edge lengths. The negative value as input terminates this console application immediately.

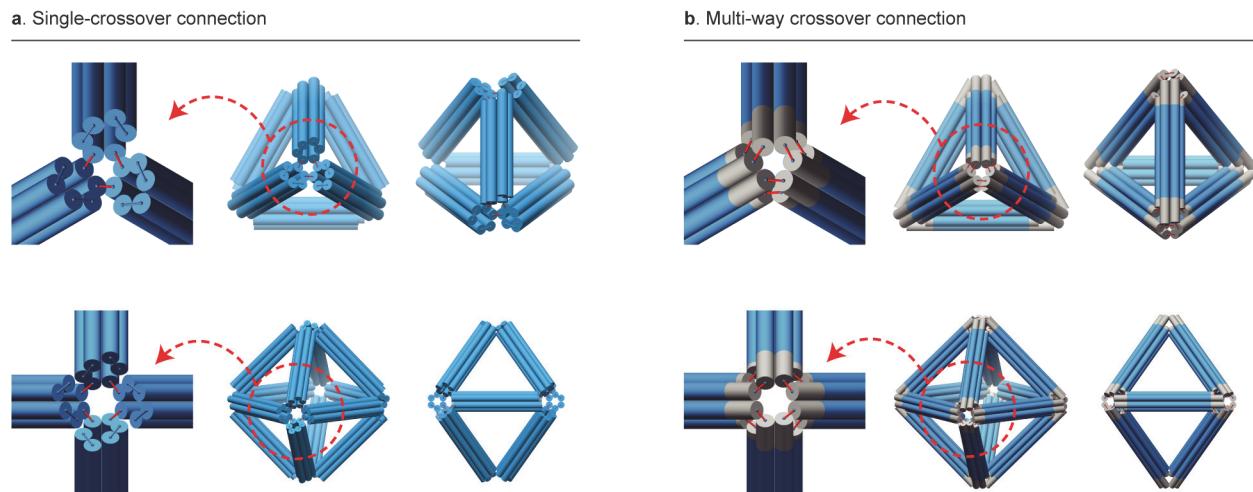


Fig. 3. The second input parameter to choose the vertex design; (a) “Single-crossover connection” and (b) “Multi-way crossover connection” vertex design. Cylindrical models for (Top) 63-bp edge length DNA tetrahedron and (Bottom) 84-bp edge-length DNA octahedron.

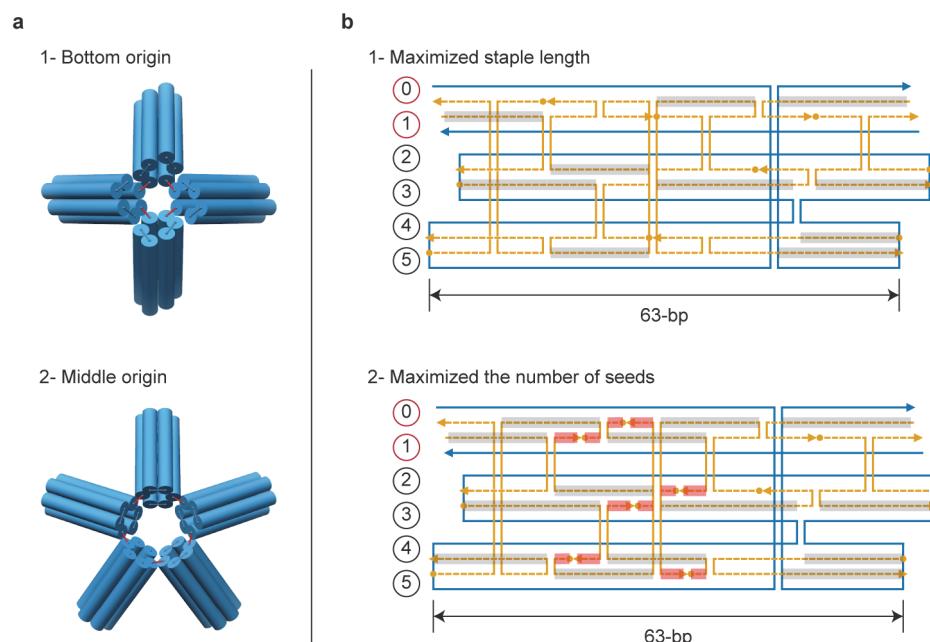


Fig. 4. The input parameters for (a) the position of the local coordinate system and (b) the two staple-break rules. In the panel b, the blue line represents the scaffold and green lines indicate staples. The grey and red shaded regions represent the 14-nt and 4-nt dsDNA domain, respectively. The arrow 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). 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. PERDIX-6P can be run with the following four parameters (**Table 2**) from the command shell.

C:\PERDIX\PERDIX-6P ***Opt1 Opt2 Opt3 Opt4 Opt5***

Table 2. Parameters to run PERDIX-6P in the command console

Parameter	Description
<i>Opt1</i> String	The file name of the target geometry (including the file extension) Ex) tetrahedron.ply
<i>Opt2</i> Integer	The vertex design (see Fig. 3) <ul style="list-style-type: none"> • 1: Single-crossover connection • 2: Multi-way crossover connection
<i>Opt3</i> Integer	The location of the origin of the local coordinate system (see Fig. 4a) <ul style="list-style-type: none"> • 1: Bottom origin • 2: Middle origin
<i>Opt4</i> Integer	The minimum edge length, which is multiples of 21-bp and greater than 21-bp Ex) 42 – 42-bp as minimum edge length
<i>Opt5</i> 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, in the sequence design of the 42-bp (**Opt4** = 42) octahedron (**Opt1** = “octahedron.ply”) with the “single-crossover connection” vertex design (**Opt2** = 1), bottom origin (**Opt3** = 1) and the maximized the number of seeds (**Opt5** = opt), the command as below:

C:\PERDIX\PERDIX-6P ***octahedron.ply 1 1 42 opt***

To run PERDIX-6P through the command shell on Mac and Linux environments, user should installed 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.

Part 2. Compiling source code

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

¹² <https://www.winehq.org/>

<https://github.com/hmjeon/PERDIX-6P/archive/1.0.zip>

or browse the codes on GitHub,

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

You can also clone the project with Git¹³ by running:

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

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 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 systems, we will explain how to compile the source codes of PERDIX-6P in two ways as follow

- Compiling sources on command (see **Part 2.1**)
 - It can easily compile source codes with the simple text editor. Also, with Intel Fortran compiler for Linux and Mac OS X, user can compile codes and make execution file on the user's environment.
- Compiling sources in Visual Studio IDE (Integrated Development Environment) (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.

¹³ <https://git-scm.com/>

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

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

Part 2.1. Compiling sources 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 using the 'PATH' environment variable, usually by prepending the /bin folder to your PATH variable.

Follow these steps to invoke the compiler from the command line:

1. 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 2017' → 'Compiler 17.0 Update 1 for IA-32 Visual Studio 2015 environment'.
2. Use the **cd** command to move in the folder, 'src'.

Make sure that the 'env.txt' and 'seq.txt' must be at the same folder where source codes exist.

3. Type 'make' to invoke the compiler using 'Makefile'.
4. After compiling sources, you will have 'PERDIX-6P.exe' file.
5. Move using **cd** to the folder where the source codes exist.
6. You can open and modify Fortran source codes (**F90**) by the general text editor such as Sublime Text, Notepad++, Vim, Atom, Emacs, etc.

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 two text file, 'env.txt' and 'seq.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

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

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, ‘_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 (**Fig. 5, Fig. 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 ‘_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.

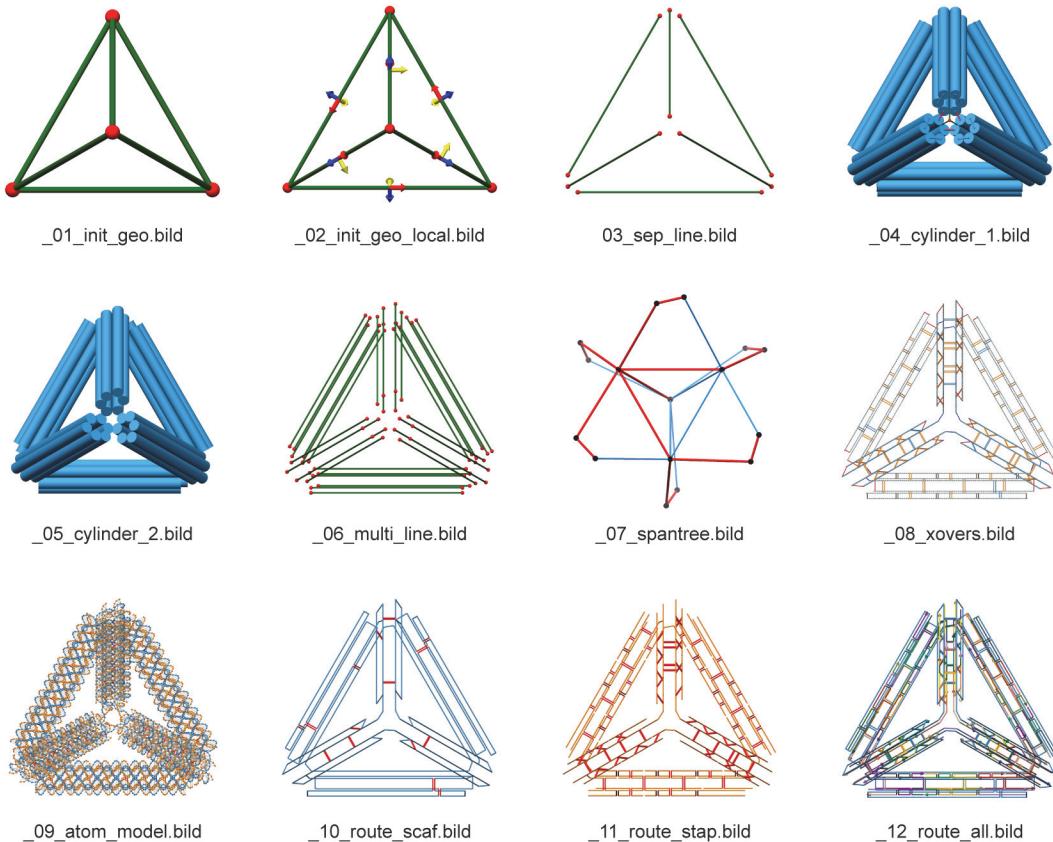


Fig. 5. 12 different rendering from BILD outputs when considering the “single-crossover connection” vertex design for 84-bp edge length DNA tetrahedron.

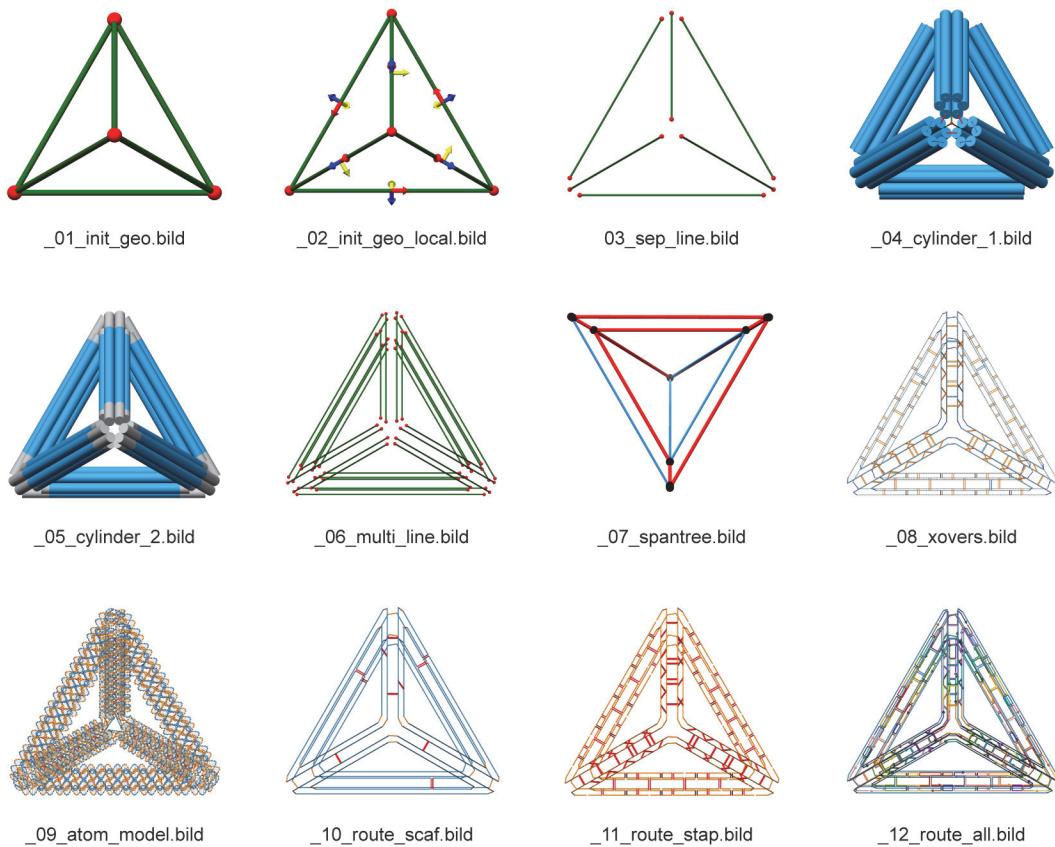


Fig. 6 12 different rendering from BILD outputs when considering the “multi-way crossover connection” vertex design for 84-bp edge length DNA tetrahedron.

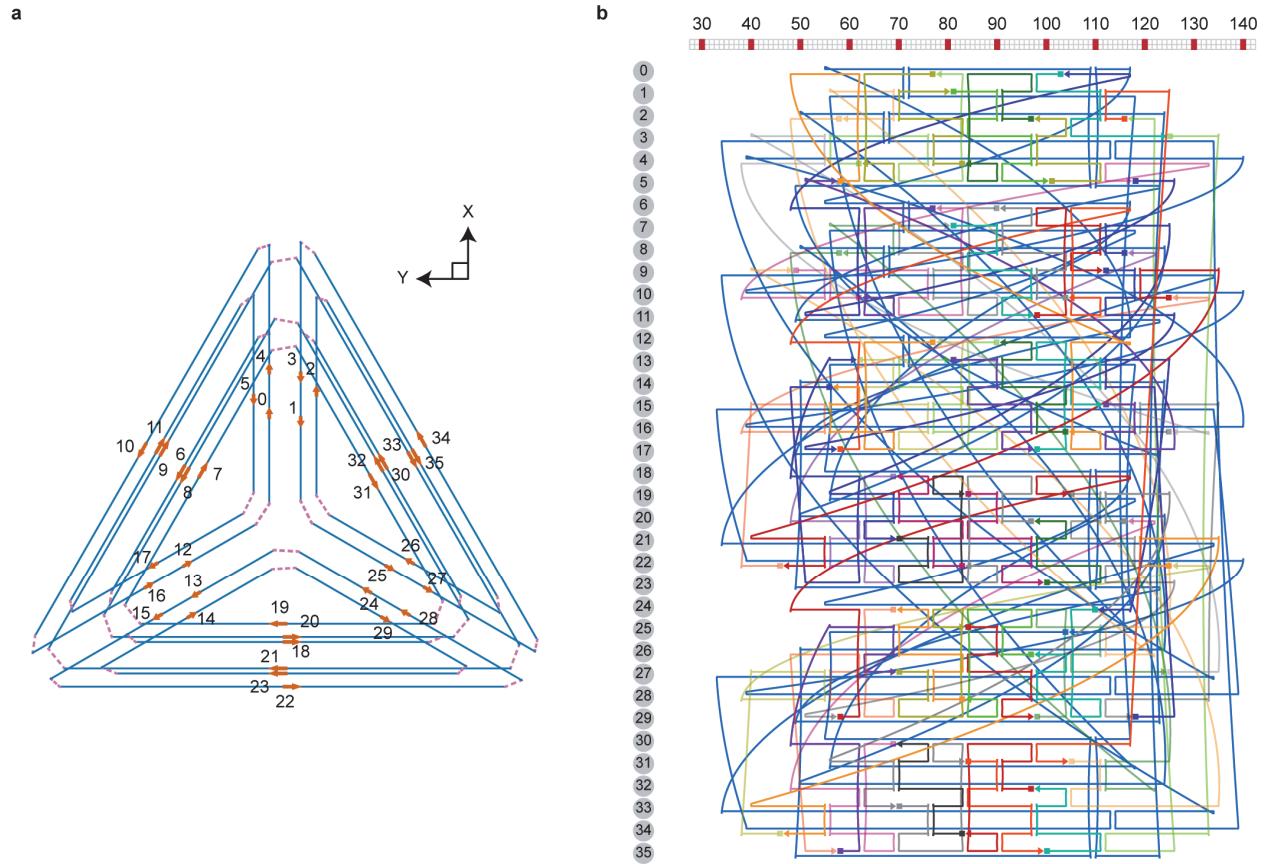


Fig. 7. (a) JSON guide model in which the edge numbers are associated with the cross-section number in (b) staple and scaffold organization from caDNAno.

Table 3. The meaning of colored line, circle, cylinder in each *bild* output generated by PERDIX-6.

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

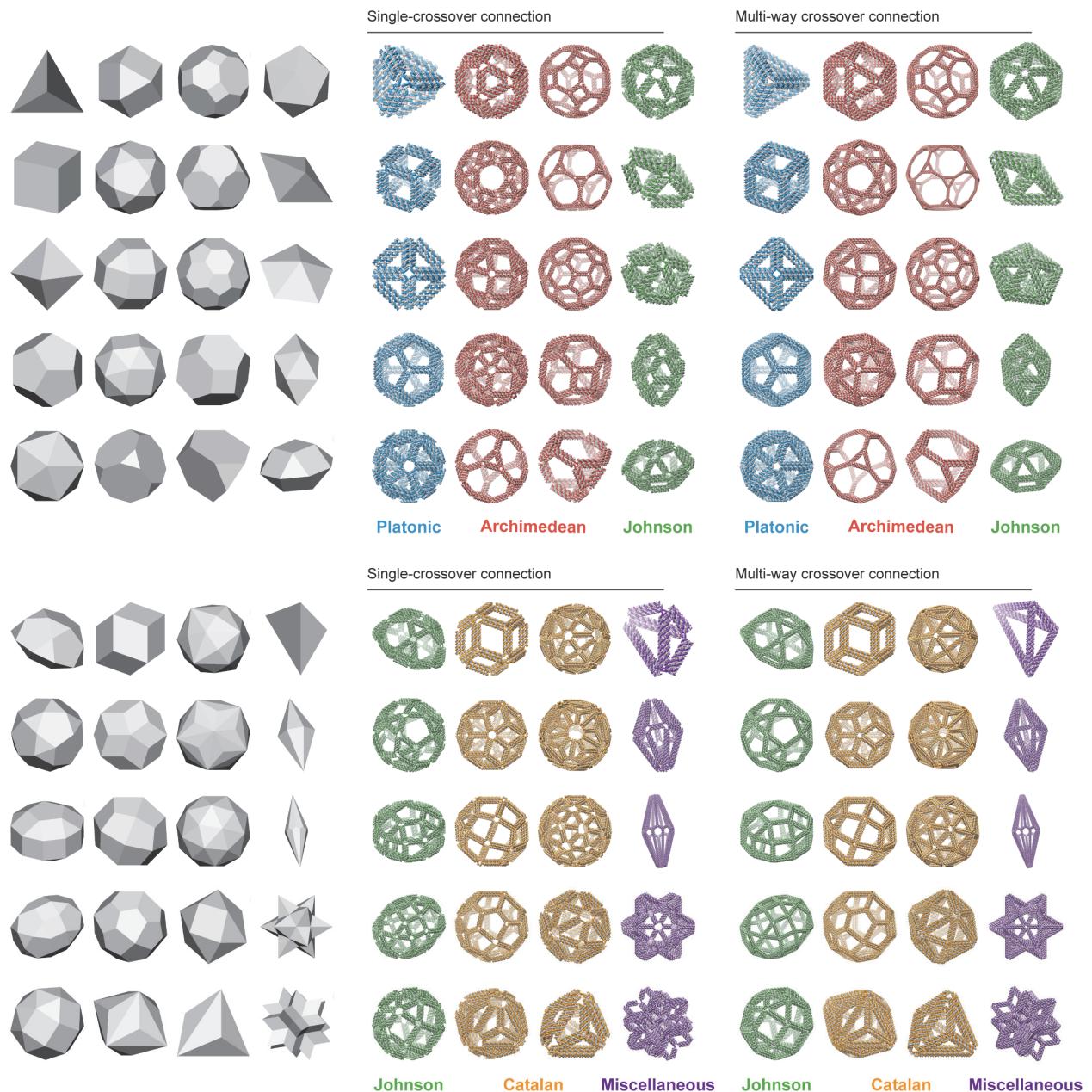


Fig. 8. Atomic model of 45 diverse nanoparticles generated by PERDIX-6P.