PERDIX-2L

(**P**urine / pyrimidine **E**ngineering **R**outing **D**esign **I**nverse **X** – DX lattice)

Software Instructions and Demo

Hyungmin Jun

Laboratory of Computational Biology and Biophysics, MIT

An online version of this software is available at https://github.com/hmjeon/PERDIX-2L.

Table of Contents

Welcome to PERDIX-2L!

PERDIX-2L simplifies and enhances the process of designing 2D DNA wireframe structures from the CAD (Computer-aided Design) geometry as an input. With this software, you will be able to render almost any target 2D shape as a scaffolded DNA origami lattice array composed of DX-based edges. By providing a geometry file such as GEO (PERDIX geometry definition file format), IGES / IGS ("Initial Graphics Exchange Specification"), and PLY ("Polygon File Format") files of your design that describes your target 2D geometry, PERDIX-2L 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-2L 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 PERDIX-2L 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 PERDIX-2L. 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.
- Several BILD files. These BILD[6] 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 caDNAno[7] software that enable users to edit the staple connections and sequences. When opening the BILD file named '_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-2L to begin to explore the capabilities of this powerful molecular design paradigm.

PERDIX-2L features:

- Fully automatic procedure for the sequence design of the DX-based 2D DNA lattice arrays
- Importing GEO, IGES/IGS, or PLY file formats as an input

[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/

- Arbitrary edge-lengths to design asymmetric and irregular shapes
- JSON output for editing staples from caDNAno[8]
- 3D visualization powered by UCSF Chimera
- Pre-defined 34 target geometries
- User-friendly TUI (Text-based User Interface)
- Free and open source (GNU General Public License, version 3.0)

PERDIX variants:

- PERDIX-6P – Automatic procedure for design of 6HB nanoparticles
- PERDIX-2L – Automatic procedure for design of DX-based 2D wireframe structures

Part 1. Preparing geometry

Discretizing the target geometry with the polygon meshes is tedious and time-consuming requiring special software such as MeshLab[9], Gmsh[10] or Autodesk Netfabb[11]. For ease-of-use, PERDIX-2L use a set of 2-points straight lines specifying the target geometry as an input and the algorithm automatically converts a set of lines to polygon meshes using the external scripts (Shapely[12] and DistMesh[13]). To render the target geometry, user can simply define points and lines in the text document (**Part 1.1**), or draw target geometry with lines from pre-existing CAD software with GUI environments (**Part 1.2**).

Part 1.1. Points and lines in the text document

PERDIX-2L is compatible with two drawing methods using straight lines; (1) drawing the border of their target object ("boundary design", **Fig. 1a**), leaving the internal structure up to the algorithm, or (2) drawing the exact lines that will be converted into DX edges in the final origami ("boundary design & internal mesh design", **Fig. 1b**).

Users can use general text editors (e.g, Notepad) to create the user editable input file (*.GEO) in which the number of points, lines, and faces should be sequentially contained at the first row. The following rows are corresponding to the x and y positions of points. Then, following rows contain the connectivity of the lines if the number of lines are not zero, or of the faces if the number of faces are not zero.

[8] http://cadnano.org
[9] http://meshlab.sourceforge.net/
[10] http://gmsh.info/
[11] https://www.autodesk.com/products/netfabb/
[12] https://pypi.python.org/pypi/Shapely
[13] http://persson.berkeley.edu/distmesh/

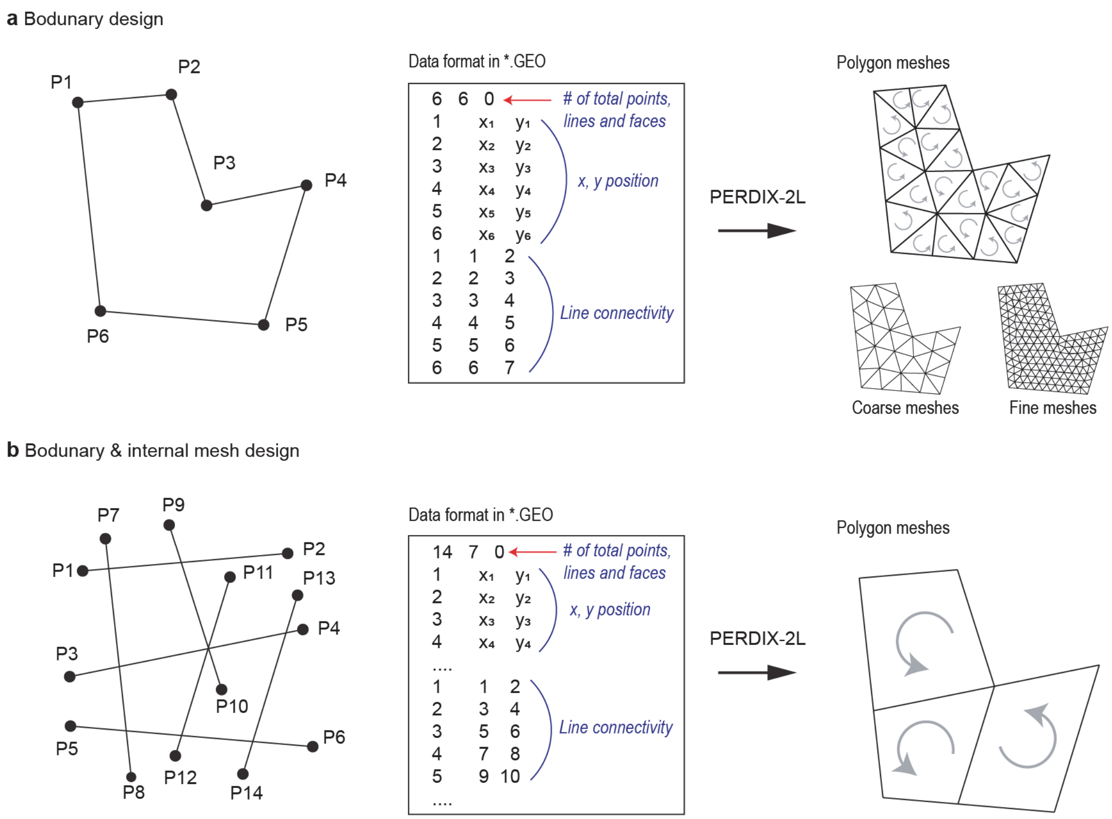

Fig. 1. Target geometry specification using two-points straight lines in the text file, *.GEO (**a**) drawing the border of their target object, leaving the internal structure up to the algorithm and (**b**) drawing the exact lines that will be converted into DX edges in the final origami. When the number of faces are one ("boundary design"), auto-generated triangular meshes are fill in the border of target geometry with the mesh spacing parameters that determine the mesh density.

It is note that, for "Bodunary & internal mesh design", the internal polygon mesh is constructed when intersectional points are formed the closed loop and the branching lines that are not involved in the generation of polygon meshes are deleted. Two drawing methods can be easily integrated with pre-existing CAD software such as FreeCAD[14] with GUI environments (see **Part 1.2**).

Part 1.2. Drawing geometry from GUI tools

With the unique feature converting from piecewise lines to polygon meshes, users can use pre-existing CAD software to render the target geometry with straight lines. Here is the guide examples using open-source FreeCAD[15] software, which has simple GUI interface and easy to design real-life objects of any size.

[14] https://www.freecadweb.org/
[15] https://www.freecadweb.org/

After installing FreeCAD, open it and create a new empty document (Ctrl + N). To draw 2D geometry, change the workbench to ‘Draft’ (drop-down list in main toolbar). Select ‘2-points line’ in the drawing tool and draw any 2D geometry. To save it, select all lines (Ctrl + A) and go the File → Export (Ctrl + E) and Save as with IGES format (IGES or IGS). Then, PERDIX-2L can import the IGES file format as an inputs (see **Part 2.1**)

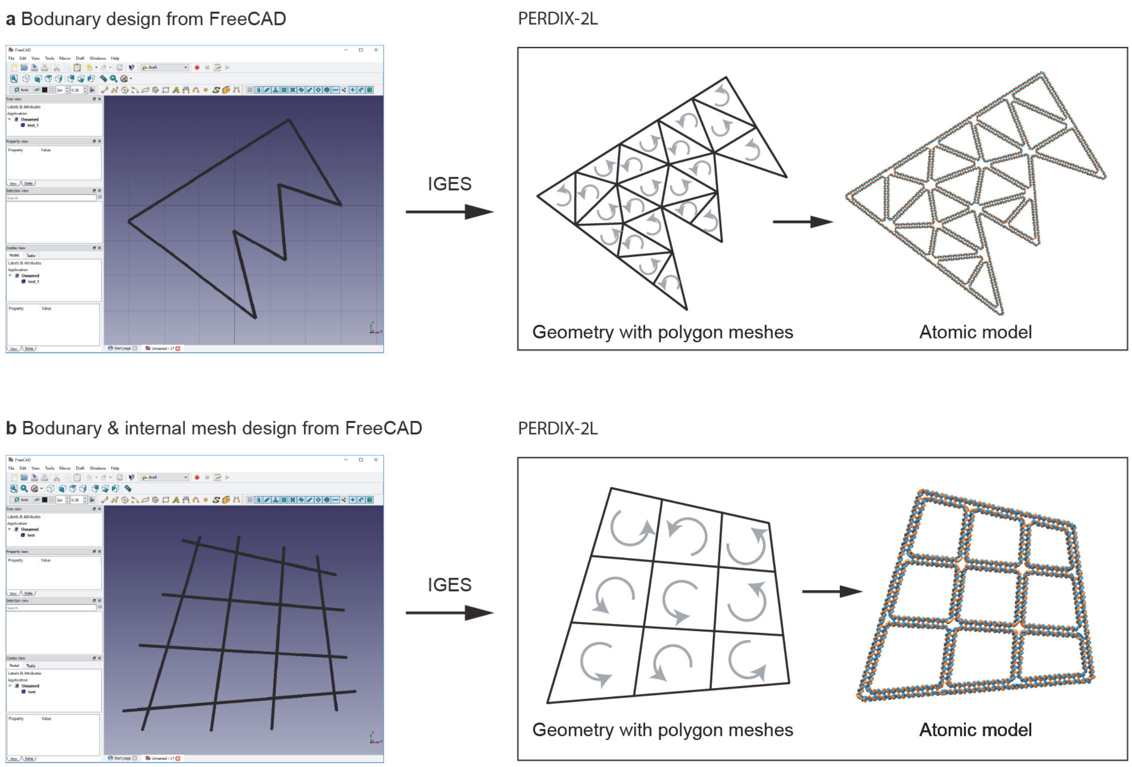

Fig. 2. Drawing geometry from FreeCAD software and import the geometry by IGES format as an input of PERDIX-2L. (**a**) “Boundary design” and (**b**) “boundary & internal mesh design”.

Part 2. Release package

The release package (as an executable file .EXE) is available only for Microsoft 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-2L

and compile it properly under your OS (see **Part 3**). 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-2L/files/1830573/released.zip

Part 2.1. Opening the PERDIX-2L software by double-clicking the .EXE icon

After download, unzip the file, 'PERDIX-2P.zip'. In the folder named 'PERDIX-2P', you will have the following subfolders and files:

- File 'PERDIX-2P.exe': This is an executable file to run PERDIX-2P 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-2P (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 'tools': This folder contains the MATLAB script of DistMesh to generate automatic meshes and stand-alone executable file to convert lines to polygon meshes using Shapely package.
- Folder 'input': This folder is empty and user should copy their geometry files into this directory.
- Folder '/Examples': This folder contains the output files generated by PERDIX-2P from the square and honeycomb shape DNA lattice objects with 42-bp edge length.

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_scaf_seq	**0** | 1 | 2	Scaffold sequence • 0: M13mp18(7249nt) sequence • 1: sequence from the file, '*seq.txt*' • 2: randomly generated sequence

The easiest way to run PERDIX-2L is to double-click the icon or file name of 'PERDIX-2L.exe' in the file manager of the Window operating systems (It can be also run by command prompt (see **Part 2.2**)). Note that the three files, 'PERDIX-2L.exe', 'env.txt', and 'seq.txt' should be in the same folder in order to properly run the software (the geometry file, *.GEO, *.IGES / IGS, *.PLY should also be in the folder named as the 'input'). 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. 3**). There are pre-defined 24 geometries including the 2D wireframe arrays with the triangular, quadrilateral, and *N*-polygon meshes. If you have your own GEO, IGES / IGS, or PLY files specifying the target geometry, just type the name of geometry file with its extension.

Note: Make sure that PERDIX-2L can only read the PLY file format in ASCII[16] (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 structure file format into PLY, you can use some software such as MeshLab[17], Gmsh[18] or Autodesk Netfabb[19].

```
 +=============================================================+
 |                                                             |
 |  PERDIX-2L by Hyungmin Jun (hyungminjun@outlook.com), 2018  |
 |                                                             |
 +=============================================================+

A. First input - Predefined 2D target geometry
=============================================

 [Triangular-mesh objects]
    1. Square,               2. Honeycomb
    3. Circle,               4. Wheel,                    5. Ellipse

 [Quadrilateral-mesh objects]
    6. Rhombic Tiling,       7. Quarter Circle
    8. Cross,                9. Arrowhead,               10. Annulus

 [N-polygon-mesh objects]
   11. Cairo Penta Tiling,  12. Lotus
   13. Hexagonal Tiling,    14. Prismatic Penta Tiling,  15. Hepta Penta Tiling

 [Variable vertex-number, edge-length, and internal mesh]
   16. 4-Sided Polygon,     17. 5-Sided Polygon,         18. 6-Sided Polygon
   19. L-Shape [42-bp],     20. L-Shape [63-bp],         21. L-Shape [84-bp]
   22. Curved Arm [Quad],   23. Curved Arm [Tri],        24. Curved Arm [Mixed]

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

Fig. 3. The interface user can see when opening the PERDIX-2L software. The 24 pre-defined target geometries as the first input parameter of PERDIX-2L. Users can use the own their surfaced geometry with typing the geometry file name including file extensions (GEO, IGES / IGS, or PLY). Displayed another input for minimum edge lengths. The negative value as input terminates this console application immediately.

The second input shown in **Fig. 4** is to select minimum edge lengths which is assigned to the shortest edge and the other edges are scaled. User can directly type the number as the second input. With two 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.

16 http://paulbourke.net/dataformats/ply/
17 http://meshlab.sourceforge.net/
18 http://gmsh.info/
19 https://www.netfabb.com/

```
B. Second input - Pre-defined minimum edge length
=================================================

   [Honeycomb lattice]

      1.  31 bp =  3 turn * 10.5 bp/turn ->  31 bp * 0.34nm/bp = 10.54nm
   *  2.  42 bp =  4 turn * 10.5 bp/turn ->  42 bp * 0.34nm/bp = 14.28nm
      3.  52 bp =  5 turn * 10.5 bp/turn ->  52 bp * 0.34nm/bp = 17.85nm
   *  4.  63 bp =  6 turn * 10.5 bp/turn ->  63 bp * 0.34nm/bp = 21.42nm
      5.  73 bp =  7 turn * 10.5 bp/turn ->  73 bp * 0.34nm/bp = 24.99nm
   *  6.  84 bp =  8 turn * 10.5 bp/turn ->  84 bp * 0.34nm/bp = 28.56nm
      7.  94 bp =  9 turn * 10.5 bp/turn ->  94 bp * 0.34nm/bp = 32.13nm
   *  8. 105 bp = 10 turn * 10.5 bp/turn -> 105 bp * 0.34nm/bp = 35.70nm
      9. 115 bp = 11 turn * 10.5 bp/turn -> 115 bp * 0.34nm/bp = 39.27nm
   * 10. 126 bp = 12 turn * 10.5 bp/turn -> 126 bp * 0.34nm/bp = 42.84nm

Select the number or type the minimum edge length [Enter] :
```

Fig. 4. Displayed second input to determine minimum edge lengths. The negative value as input terminates this console application immediately.

User can specify the edge ID to apply the minimum edge length (see **Part 2.2**) but should be careful for the design to have at least two double-crossovers in the shortest edge.

Part 2.2. Running PERDIX-2L with command prompt

PERDIX-2L 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-2L package exists. The various geometries can easily run on command prompt using the batch file. PERDIX-2L can be run with the following four parameters (**Table 2**) from the command shell.

C:\PERDIX\PERDIX-2L **Opt1** **Opt2** **Opt3**

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

Parameter		Description
Opt1	String	The file name of the target geometry (including the file extension) Ex) square.geo
Opt2	Integer	Specific edge ID, user can check the edge ID in the output file named as "_03_sep_line.bild" file. See **Fig. 5**. The shortest edge in the target is assigned when Opt2 = 0. Ex) 1 – Edge that has ID of 1

Opt3	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

For example, in the sequence design of square (**Opt1** = "square.geo") with the 48-bp (**Opt3** = 48) edge length for edge ID of 2 (**Opt2** = 2), the command as below:

C:\PERDIX\PERDIX-2L **square.geo** **2** **48**

To run PERDIX-2L through the command shell on Mac and Linux environments, user should installed Wine[20] 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 3. Compiling source code

You can download the source code PERDIX-2L in zip format from

https://github.com/hmjeon/PERDIX-2L/archive/master.zip

or browse the codes on GitHub,

http://github.com/hmjeon/PERDIX-2L

You can also clone the project with Git[21] by running:

$ git clone https://github.com/hmjeon/PERDIX-2L.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 Windows[22]. Before installing Intel(R) Fortran Compiler, you must have a version of Microsoft Visual Studio installed since the Intel Fortran Compiler integrates into

[20] https://www.winehq.org/
[21] https://git-scm.com/
[22] https://software.intel.com/en-us/qualify-for-free-software

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[23]. 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 3.1**)
    - 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 3.2**).
    - 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.

Part 3.1. 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 project[24]. 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 2015 / 2016 / 2017 → Compiler 17.0 Update 1 for IA-32 Visual Studio 2015 environment.

2. Use the **cd** command to move in the folder, *Sources.*

[23] https://www.visualstudio.com/vs/community/
[24] http://gnuwin32.sourceforge.net/packages/make.htm.

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-2L.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 3.2. 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 / 2017with 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

Part 4. Outputs

Once the sequence design from PERDIX-2L is completed, the output folder is created. The files, 'TXT_PERDIX_2L.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 (**Fig. 5** and **Table 3**).

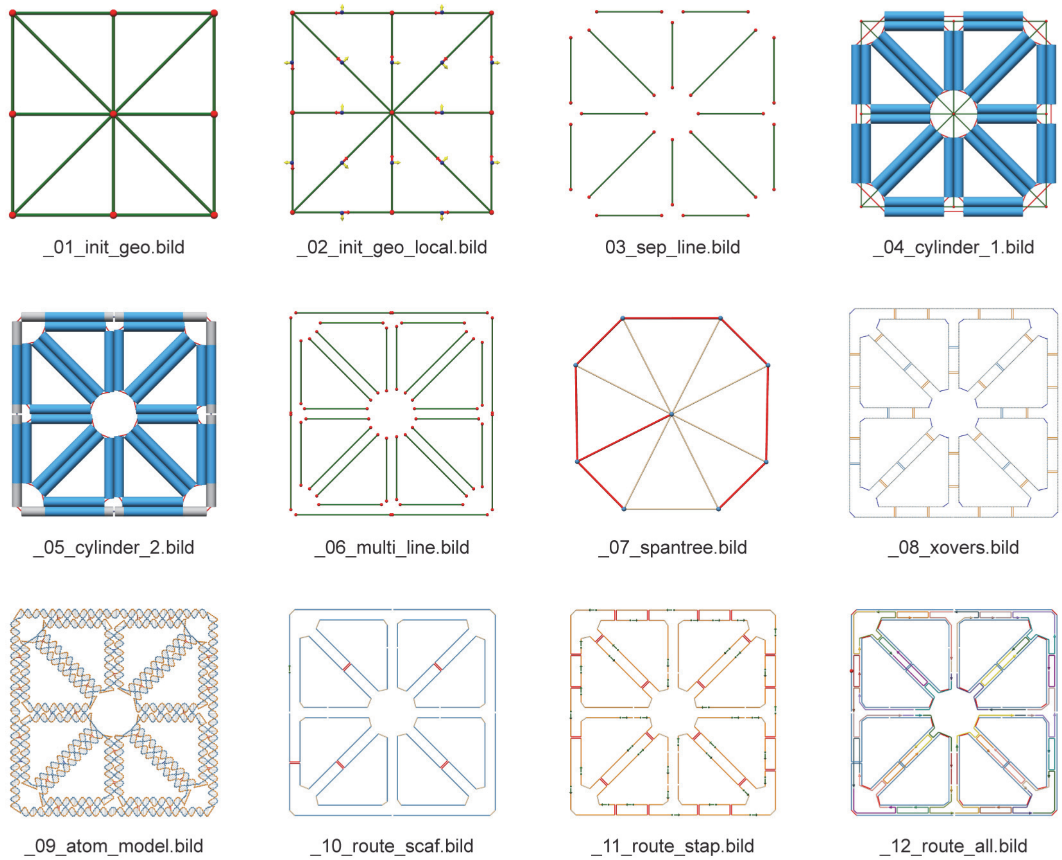

Fig. 5. Twelve rendering from BILD outputs of PERDIX-2L for 42-bp edge length DNA L-shape.

With the JSON file as one of outputs from PERDIX-2L, user can edit the staple crossover positions and sequences using caDNAno (**Fig. 6**).The file named '_15_json.guide.bild' can be loaded in USCF Chimera, which give the information which edges of the target structure is associated with the which cross-sections of caDNAno representation.

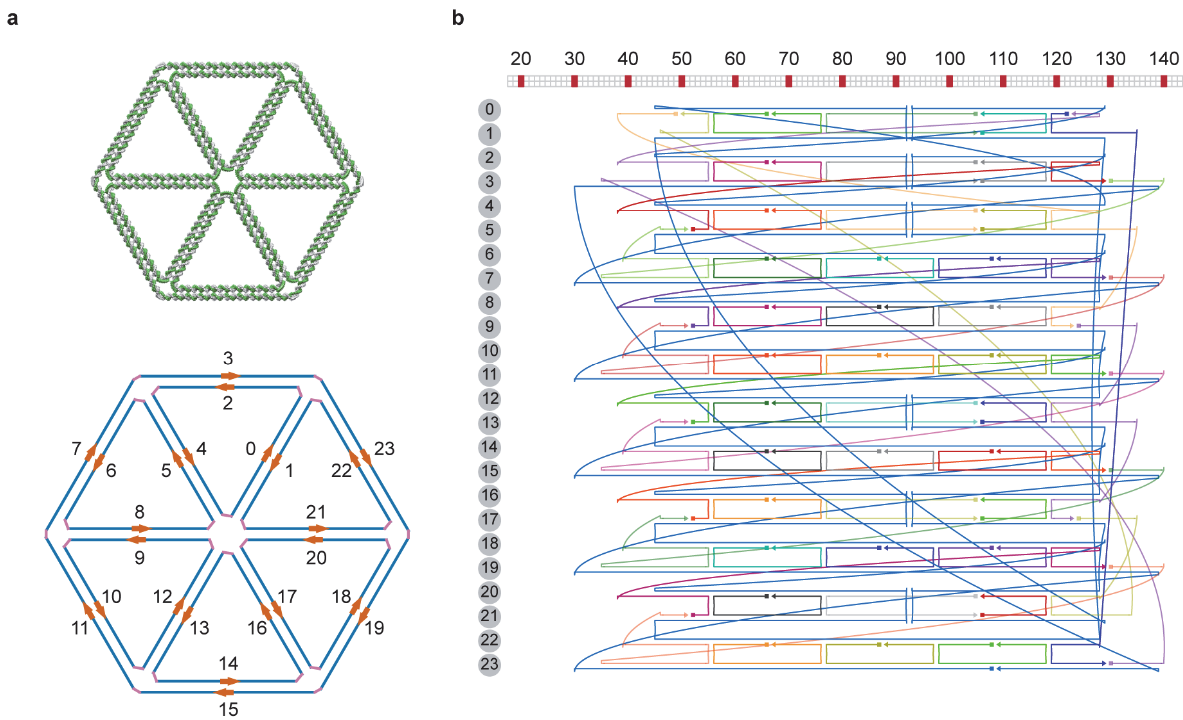

Fig. 6. (**a**) JSON guide model in which the edge numbers are associated with the cross-section number in (**b**) staple and scaffold organization from caDNAno.

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	Point of the target geometry
	Green edge	Edge of the target geometry
_02_init_geo_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_line	Red circle	Point separated from the vertex
	Green line	Line connecting two points
_04_cylinder_1 / _05_cylinder_2	Blue cylinder	Double helix DNA strand
	Grey cylinder	Extended part to fill the gap
	Red line	Scaffold strand crossing the vertex
_06_multi_line	Red circle	End point of the double helix
	Green line	Double helix DNA strand
_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_xovers	Yellow circle	Base pair
	Blue line	Scaffold crossover
	Orange line	Staple crossover
	Dark blue line	Scaffold crossing the vertex
_09_atom_model	Blue line	Scaffold strand
	Orange line	Staple
_10_route_scaf	Blue line	Scaffold strand
	Red line	Scaffold crossover
_11_route_stap	Orange line	Staple
	Red line	Staple crossover
_12_route_all	Blue line	Scaffold strand
	Multiple colored line	Staple strand

With the CNDO (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 generator[25] uses the CNDO file as its input and creates the PDB file consisting of two phosphates, two deoxyriboses, and two paired bases (**Fig. 7**). This atomic model generator is written by a MATLAB script which produces a *.PDB file, which can be similarly visualized using UCSF Chimera.

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

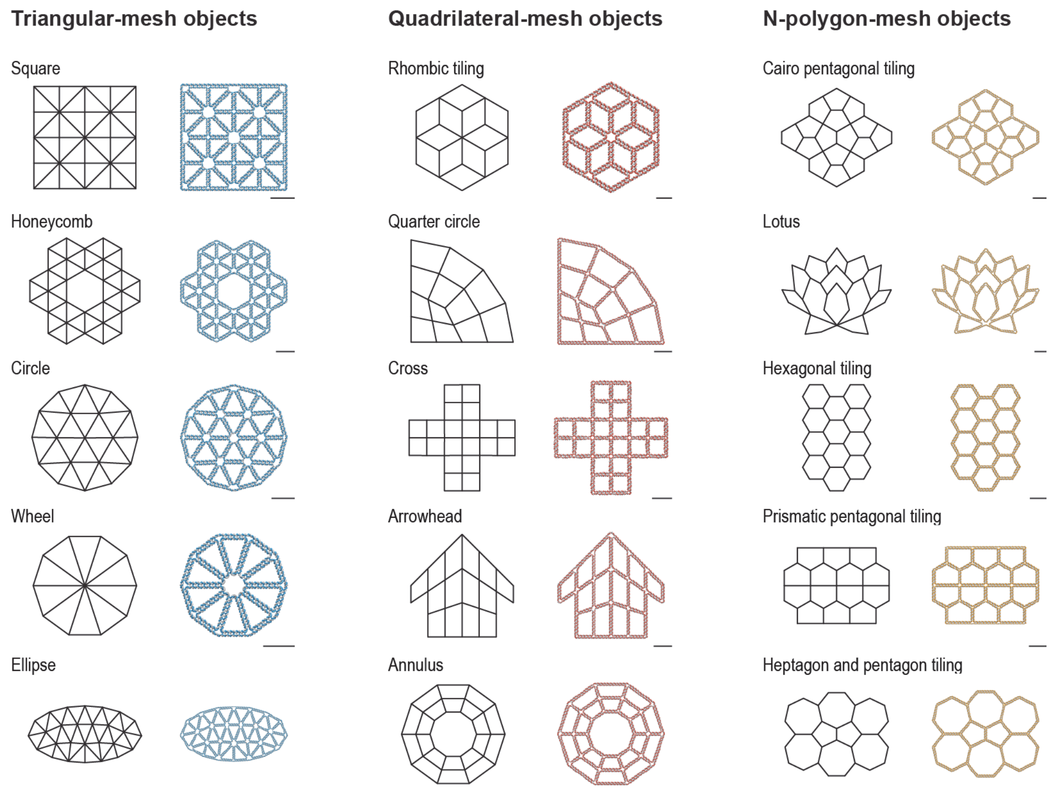

Fig. 7. Atomic model of 15 diverse nanoparticles generated by PERDIX-2L. Scale bar for atomic structures is 20 nm.