# PERDIX-2L

(**P**rogrammed **E**ulerian **R**outing of **D**NA des**i**gn using **X**-overs – DX lattice)

Software Instructions and Demo

Dr. Hyungmin Jun

Laboratory for Computational Biology and Biophysics, http://lcbb.mit.edu, MIT



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

## Table of Contents

## Welcome to PERDIX-2L!

PERDIX-2L simplifies and enhances the process of designing 2D scaffolded DNA origami wireframe geometries using CAD (Computer-aided Design) files as input. With this software, you will be able to render practically 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"), or PLY ("Polygon File Format") files describing 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 nanostructure by following the standard annealing protocol provided in our work.
- A CNDO file of your nanostructures. This output CNDO file (CanDo file format[1]) from PERDIX-2L can be used to predict the flexibility of programmed DNA nanosturctures[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.
- BILD[6] files. BILD files are used for visualizing the target geometry, scaffold routing, staple sequence and cylindrical models, which are rendered using lines, polygons, and geometric primitives built in UCSF Chimera (see Fig. 5).
- JSON file. The 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 scaffolded DNA origami design, you can use PERDIX-2L to begin to explore the capabilities of this powerful molecular design paradigm!

PERDIX-2L features:

- Fully automatic procedure for sequence design of scaffolded DNA origami objects using DX-based wireframe lattices
- Importing GEO, IGES/IGS, or PLY file formats as input
- Exact edge-lengths to design highly asymmetric and irregular shapes

[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/

- JSON file output for editing staple paths and sequences using caDNAno
- 3D visualization using UCSF Chimera
- 24 pre-defined target geometries
- User-friendly TUI (Text-based User Interface)
- Online web resources and release packages for Microsoft Windows and macOS
- 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

Free online web resource:

http://perdix-dna-origami.org/

We also offer the free online resource PERDIX that automatically converts any 2D object specified using a simple Computer-Aided Design file (GEO, IGS/IGES, or 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 PERDIX-2L.

## Part 1. Preparing target geometry designs

It may be tedious and time-consuming to manually discretize target, boundary-based geometries using polygonal meshes. Instead, fully automated software can be used including MeshLab[8], Gmsh[9] or Autodesk Netfabb[10]. For ease-of-use, PERDIX-2L uses a target input geometry that is specified using a set of straight lines that are automatically converted into polygonal meshes by external MATLAB and Python scripts (Shapely[11] and DistMesh[12]). Thus, the arbitrary target geometry can be defined simply using a set of points and their connectivity in the text file with the file extension GEO (Part 1.1), or lines from pre-existing CAD software packages that have GUI environments (Part 1.2).

### Part 1.1. Definition of straight lines in the input text file

PERDIX-2L is compatible with two drawing methods using straight lines; (1) drawing the border of their target object (“free-form boundary design”, Fig. 1a), leaving the internal structure

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

up to the algorithm, or (2) drawing the exact lines that will be converted into DX edges in the final origami (“free-form boundary design and internal geometric design”, Fig. 1b).

Users can use general purpose text editors (e.g., Notepad++ and Atom) to create an editable input file (*.GEO) in which the number of points, lines, and faces should be sequentially contained starting from the first row. The following rows correspond to the x and y positions of points. Subsequently, following rows contain the connectivity of the lines if the number of lines is not zero, or of the faces if the number of faces is not equal to zero.

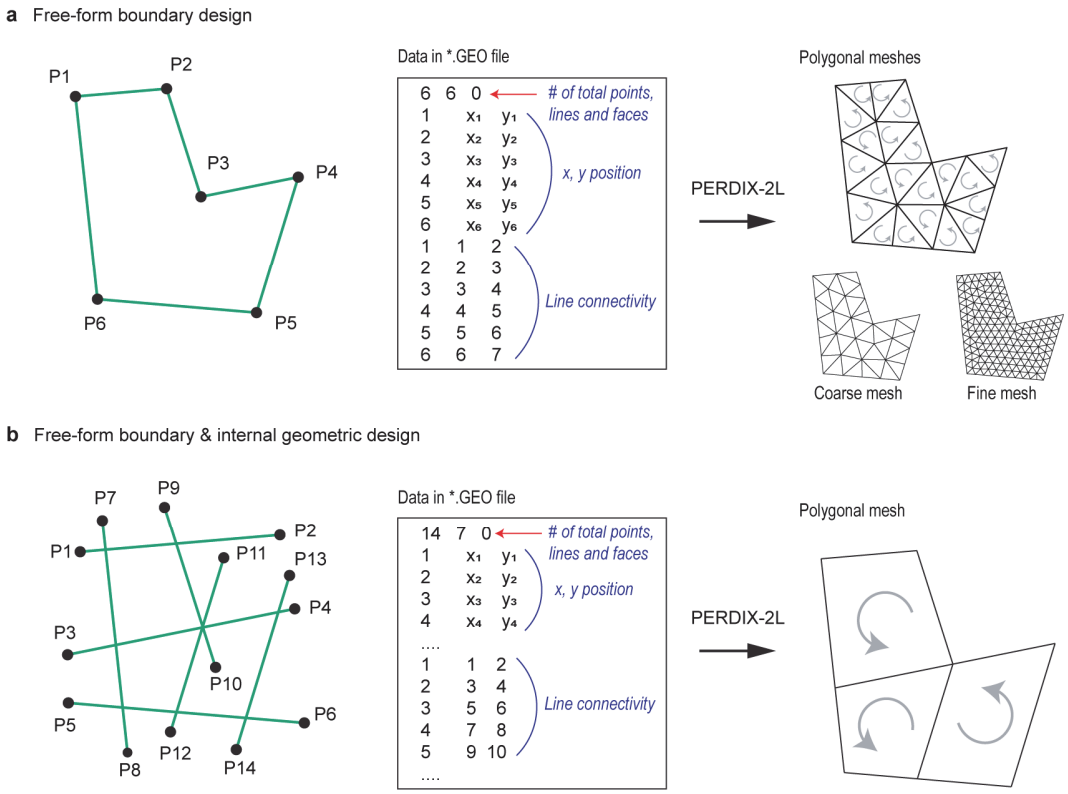


**Fig. 1 | Target geometry specification using straight lines in the text file, *.GEO. a**, drawing the border of their target object, leaving the internal structure up to the algorithm and **b**, drawing the exact lines that will be converted into DX edges in the final origami. When the number of faces is one in case of the “free-form boundary design”, triangular meshes are automatically filled in the border of the target geometry with the mesh spacing parameters that determine the mesh density.

It is noteworthy that, for “Free-from boundary & internal geometric design”, the internal polygonal mesh is constructed when intersectional points are formed in? the closed loop and the branching lines that are not involved in the generation of polygonal meshes are deleted. Two

drawing methods can be easily integrated with pre-existing CAD software such as FreeCAD[13] with GUI environments (see the following section).

## Part 1.2. Specifying the geometry using GUI drawing software

Using lines instead of polygonal meshes to define your target geometry, pre-existing CAD software can be used to render the target geometry and easily integrated with PERDIX-2L. FreeCAD software is the best option since it is open-source with a simple GUI interface and it supports the IGES/IGS format as an export format.

After installing FreeCAD (for Windows or Mac or Linux), open the software and create a new empty document (Ctrl + N). To draw the 2D geometry, change the workbench to 'Draft' (drop-down list in main toolbar). Select '2-points line' in the drawing tool and draw the target geometry. To save it, select all lines (Ctrl + A) and go to File → Export (Ctrl + E) and save your design in IGES format (IGES or IGS). Finally, PERDIX-2L can be used to import the IGES file format as input (see Part 2.1).

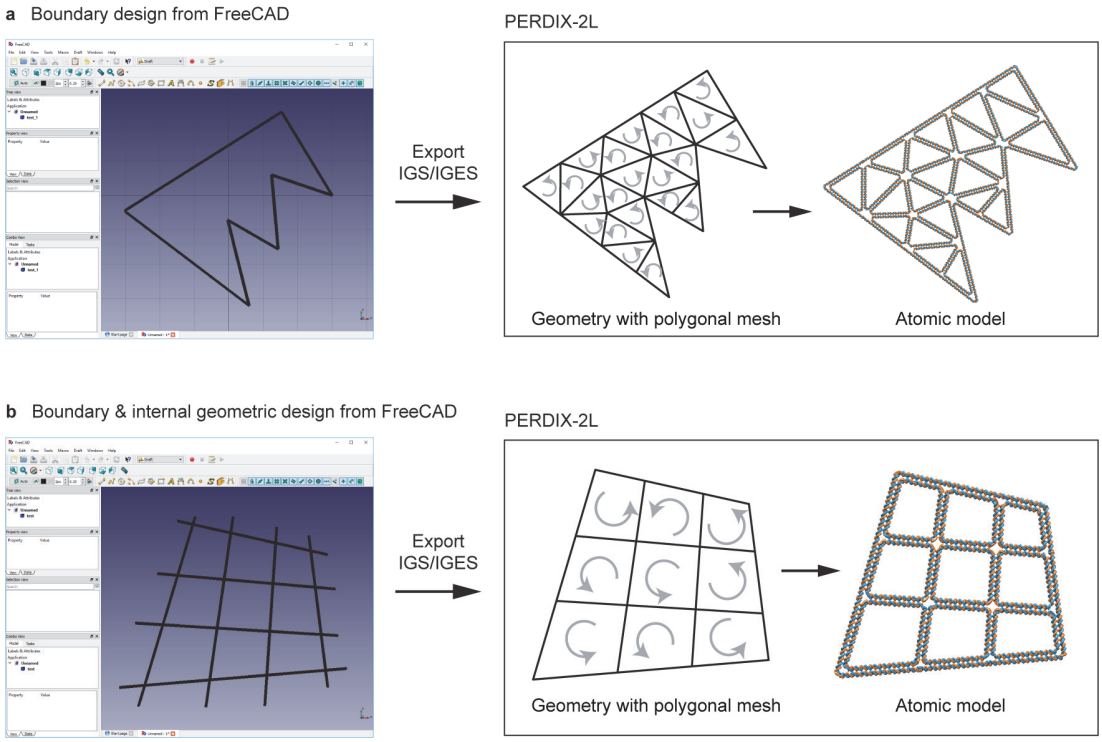


**Fig. 2 | Drawing geometry from FreeCAD software and import the IGES output as input to PERDIX-2L. a**, Free-form boundary design. **b**, Free-form boundary & internal geometric design.

[13] https://www.freecadweb.org/

## Part 2. Release package

The release package (as an executable file) is available for both Microsoft Windows and macOS. These executable files are the console application for Windows (PERDIX-2L.EXE) and Mac (PERDIX-2L), which accept an input (GEO, IGES/IGS, and PLY) and generate outputs through the command (Windows) or terminal (Mac). You can download the release package from:

Microsoft Windows - MCR (MATLAB Compiler Runtime 2015[14]) version

https://github.com/lcbb/PERDIX-2L/raw/master/release/PERDIX-2L-Win-MCR.zip

Microsoft Windows - MATLAB version

https://github.com/lcbb/PERDIX-2L/raw/master/release/PERDIX-2L-Win-MATLAB.zip

macOS / Mac OS X

https://github.com/lcbb/PERDIX-2L/raw/master/release/PERDIX-2L-Mac.zip

The current versions of the release package are compiled with Intel (R) Parallel Studio XE 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 the source codes and compile them under the Linux system (see Part 3). In the folder named 'release', you will have the following subfolders and files:

- File 'PERDIX-2L.EXE': This is the executable file to run PERDIX-2L 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 'PERDIX-2L-RUN': This is the executable file to run PERDIX-2L on macOS by double-clicking it. (Part 2.1).
- File 'PERDIX-2L': This is the executable file to run PERDIX-2L 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 'tools': This folder contains the DistMesh.EXE (for MCR version), MATLAB script of DistMesh (for MATLAB version) or Python wrapper script (PyDistMesh.py for Mac version) to generate internal triangular meshes. It also contains the Python wrapper script (Shapely.py) to convert a set of lines to polygonal meshes using Shapely.
- Folder 'input': The user-defined geometry file (GEO, IGES/IGS, or PLY) must be located here.

[14] https://www.mathworks.com/products/compiler/matlab-runtime.html

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

| Field | Value | Descriptions |
|---|---|---|
| para_platform | win \| mac | Depending on the User's operating system |
| para_scaf_seq | 0 \| 1 \| 2 | Scaffold sequence<br>• 0: M13mp18(7249nt) sequence<br>• 1: User-defined sequence<br>• 2: randomly generated sequence |

In order to use the two drawing methods: free-form boundary design and free-form boundary & internal geometric design, users should install Python and the Python package, Shapely (Python package for set-theoretic analysis and manipulation of planar features). Mac users can install Shapely by downloading 'get-pip.PY' as below:

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
```

Inspect get-pip.py for any malevolence. Then run the following:

```
python get-pip.py
```

```
pip install Shapely
```

Windows users have two good installation options: the wheels at
http://www.lfd.uci.edu/~gohlke/pythonlibs/#shapely

and the Anaconda platform's conda-forge channel.

https://conda-forge.org/

The default scaffold sequence of PERDIX-2L is M13mp18 for required length less than or equal to 7,249nt, a Lambda phage sequence if greater than 7,250nt and less or equal to 48,502nt, and a random sequence if greater then 48,503nt. 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 2.1. Opening the PERDIX-2L software by double-clicking

The easiest way to run PERDIX-2L is to double-click 'PERDIX-2L.EXE' in the file manager of the Windows, or 'PERDIX-2L-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 (GEO, IGES/IGS, 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. 3). There are 24 pre-defined wireframe lattices with the triangular, quadrilateral, and *N*-polygonal meshes. If you have your own your geometry file, just type the name of geometry file with its extension (GEO, IGES / IGS, or PLY).

*Note*: Please ensure that PERDIX-2L can only read the PLY file format in ASCII[15]. 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, or to convert another structure file format into PLY, you can use software such as MeshLab[16], Gmsh[17] or Autodesk Netfabb[18].

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

A. First input - Pre-defined 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 first parameter in PERDIX-2L software.** The 24 pre-defined target geometries as the first input parameter of PERDIX-2L. Users can use the own their geometry with typing the geometry file name with file extensions (GEO, IGES/IGS, or PLY). The negative value as an input terminates this console application immediately.

The second input shown in Fig. 4 is to select the minimum edge length which is assigned to the shortest edge and the other edges are scaled. User can directly type any arbitrary edge length that should be larger than 37 to ensure at least 2 double-crossovers. Also, users can specify the edge ID to apply the minimum edge length (see Part 2.2). With two inputs, it runs and creates the new folder named 'output' where PERDIX-2L automatically generates several output files (Table 3).

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

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

   * 1.  42 bp =  4 turn * 10.5 bp/turn ->  42 bp * 0.34nm/bp = 14.28nm
     2.  52 bp =  5 turn * 10.5 bp/turn ->  52 bp * 0.34nm/bp = 17.85nm
   * 3.  63 bp =  6 turn * 10.5 bp/turn ->  63 bp * 0.34nm/bp = 21.42nm
     4.  73 bp =  7 turn * 10.5 bp/turn ->  73 bp * 0.34nm/bp = 24.99nm
   * 5.  84 bp =  8 turn * 10.5 bp/turn ->  84 bp * 0.34nm/bp = 28.56nm
     6.  94 bp =  9 turn * 10.5 bp/turn ->  94 bp * 0.34nm/bp = 32.13nm
   * 7. 105 bp = 10 turn * 10.5 bp/turn -> 105 bp * 0.34nm/bp = 35.70nm
     8. 115 bp = 11 turn * 10.5 bp/turn -> 115 bp * 0.34nm/bp = 39.27nm
   * 9. 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 | The second input to determine minimum edge lengths.** The negative value as an input terminates PERDIX-2L immediately.

## Part 2.2. Running PERDIX-2L using a command prompt

PERDIX-2L can run through the command shell (command console / terminal). In Windows, start a command shell usign **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 is located. To access the Unix command prompt in Mac, open the terminal application. It is located by default in the Utilities folder, which in turn is inside the Applications folder. PERDIX-2L can be run using the following 3 arguments (Table 2) in the command shell.

PERDIX-2L.exe argc1 argc2 argc3 on Windows

./PERDIX-2L-Terminal argc1 argc2 argc3 on macOS

**Table 2.** Command line arguments for PERDIX-2L.

| Arguments | | Descriptions |
|---|---|---|
| argc1 | String | The file name of the target geometry (including the file extension)<br>Ex) square.geo / square.iges / from 0 to 24 (to select the pre-defined geometry) |
| argc2 | 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 argc2 = 0.<br>Ex) 1 – Edge that has ID of 1 |
| argc3 | Integer | The minimum edge length, which is any number but greater than 37-bp, to have at least two double-crossover per edge.<br>Ex) 42 – 42-bp as minimum edge length |

For example, the square with a 48-edge length at edge ID = 2 can be generated by the following command:

```
PERDIX-2L.exe square.geo 2 48          on Windows
./PERDIX-2L-Terminal square.geo 2 48   on macOS
```

Users can run PERDIX-2L.EXE through the command shell in Mac and Linux environments, after installing Wine[19], which is a free and open-source compatibility layer that enables software developed for Microsoft Windows to run on Unix-like operating systems. PERDIX-2L has been tested to run successfully using Wine on Mac and Linux systems.

## Part 3. Compiling the source code

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

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

or browse the software on GitHub:

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

You can also clone the project with Git[20] by running:

```
$ git clone https://github.com/lcbb/PERDIX-2L.git
```

The source code 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, Mac and Windows[21]. 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-

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

commercial use and it can be downloaded here[22]. 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, PERDIX-2L can only be compiled using command-mode).

Here, in Windows and macOS systems, we explain how to compile the source code of PERDIX-2L in the following two ways:

- Compiling source codes on command (see Part 3.1)
  - The source codes of PERDIX-2L can easily compiled by the build automation tool under Linux and Mac.
- Compiling sources through Visual Studio IDE (Integrated Development Environment) – Window only (see Part 2.2).
  - This provides comprehensive facilities for 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 the Windows operating system. Mac users can use the alternative IDE, Xcode which supports Intel Fortran.

## Part 3.1. 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 version of 'makefiles' with its 'nmake' utility. If one has a version of Microsoft Visual Studio installed, one can use NMAKE in Visual Studio Command Prompt to run 'Makefile'.

(*Alternatives*) The GnuWin32 project provides Win32-version of GNU tools, much of it modified to run on the 32-bit Windows platform. The usser can download the Windows version of MAKE from Gnuwin32 project[23]. 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'. The 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 to the folder named 'src'.

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

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 ‘PERDIX-2L.EXE’ for Windows and ‘PERDIX-2L’ 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 PERDIX-2L.
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 PERDIX under RedHat Linux using the Linux version of the Intel Fortran Compiler.

### Part 3.2. Compiling sources on Microsoft Visual Studio

First, check the Microsoft Visual Studio version that is supported by the Intel compilers. This project was developed under the Intel Parallel Studio XE 2015 / 2016 / 2017 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 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 the sequence design process and the results of the sequence and routing of the scaffold and staples, respectively. The file, ‘_17_sequence.CSV*’*, contains 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-staples, and so on. You will be also able to visualize these sets of data using UCSF Chimera (Fig. 5 and Table 3).

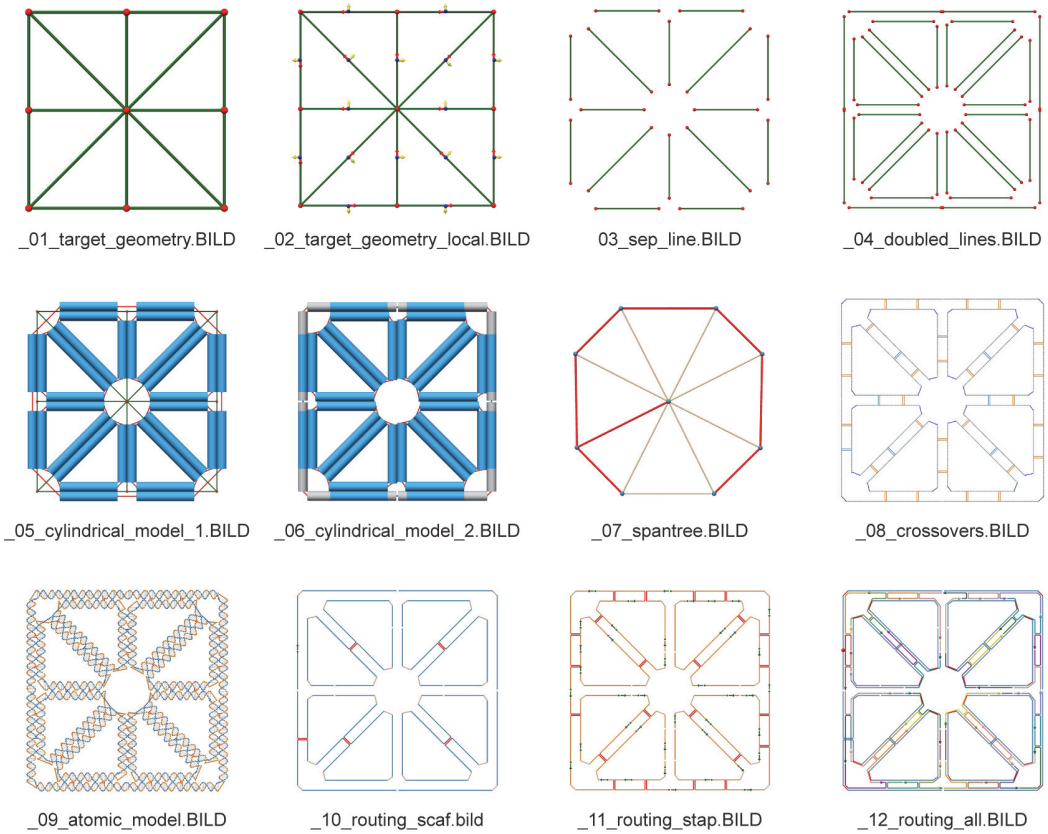


**Fig. 5 | 12 rendering from BILD outputs for a 42-bp edge length DNA object.**

With the JSON file as one of the outputs from PERDIX-2L, the user can edit the staple crossover positions and sequences using caDNAno (Fig. 6). The file named as '_15_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.

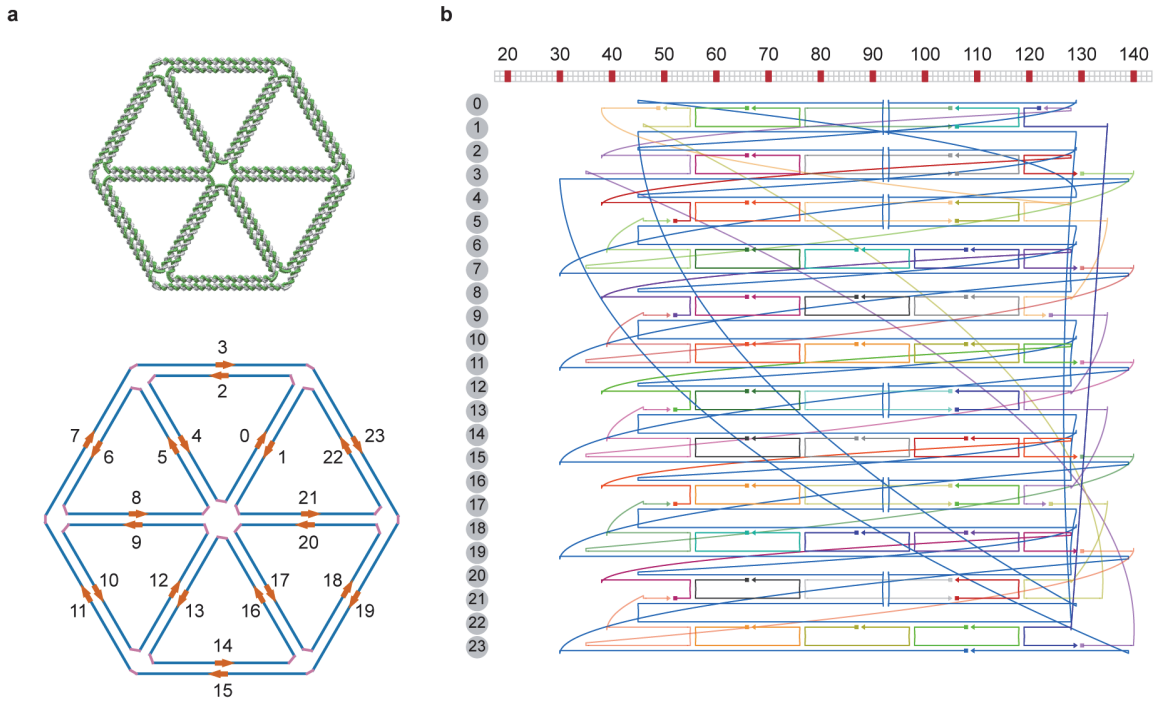


**Fig. 6 | JSON caDNAno and guide BILD outputs. a**, JSON guide model in which the edge numbers are associated with the cross-section number in caDNAno. **b,** Staple and scaffold organization from caDNAno.

**Table 3.** Descriptions of 14 BILD outputs generated by PERDIX-2L.

| 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_line | Red circle | Point separated from the vertex |
| | Green line | Line connecting two points |
| _04_doubled_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 |

Using the CNDO (CanDo file format) file, which was designed to describe DNA nanostructures, all information needed to generate the all-atom models of DNA nanostructures is provided. The atomic model generator[24] 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 using a MATLAB script that produces a *.PDB file, which can be visualized similarly using UCSF Chimera.

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

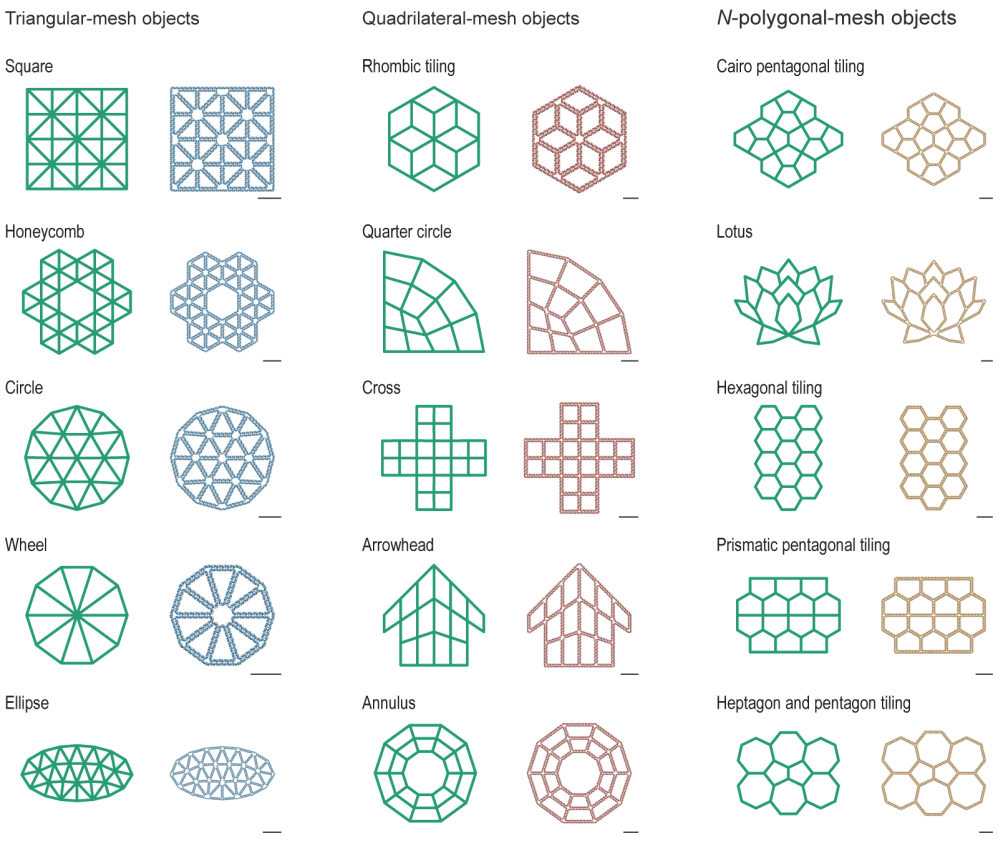


**Fig. 7 | Target geometry and atomic model of 15 diverse wireframe lattices generated by PERDIX-2L.** Scale bar for atomic structures is 20 nm.