BATHE BIONANOLAB
MIT LABORATORY FOR NUCLEIC ACID NANOTECHNOLOGY

pyDAEDALUSX

Instructions for Use

Bathe BioNanoLab
2022.02.12

The purpose of this software is to design nucleic acid-scaffolded wireframe origami with double duplex edges. With a PLY file as geometry input and with or without a TXT file as the scaffold sequence input, the algorithm calculates scaffold routing along the edges of the geometry and outputs the staple sequences required to fold the scaffold nucleic acid into the target geometry.

There are two ways to call the algorithm:

- Python batch scripts
- Graphical user interface (GUI)

Both require the program to be started as a server in the background.

1 INSTALLING THE PROGRAM

1.1 DOWNLOAD PYDAEDALUSX

Pending update: Download the program files from GitHub at https://github.com/lcbb/pyDAEDALUSX

The pyDAEDALUSX folder contains three subfolders:

- pyDAEDALUSX_program (containing the program files for actually calculating scaffold and staple routing, etc.)
- pyDAEDALUSX-win32-x64 (containing the application for the GUI on Windows)
- pyDAEDALUSX-Darwin (containing the application for the GUI on Macs)

1.2 SETTING UP THE VIRTUAL ENVIRONMENT TO USE PYTHON 2

The software was written in Python v2.7, so you need to call the program with Python 2. For most people, now operating in Python 3, this will require creating a virtual environment that uses Python 2. Instruction for creating and activating such a virtual environment in Anaconda can be found here: https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

Briefly, in terminal or Anaconda Prompt (recommended for Windows users), type:

```
conda create –name python2 python=2.7
```

to create the environment, named here “python2”.

To activate the environment (in terminal or Anaconda Prompt), type:

```
conda activate python2
```

Your command prompt prefix should then look something like:

```
(python2) $
```

The program has several required packages, listed below:

numpy>=1.11.1
matplotlib>=1.5.1
networkx==1.11
scipy>=0.18.0
click>=6.6
mpmath>=0.19
mock>=2.0.0
tqdm>=4.9.0
flake8>=3.2.0

Note that the networkx package must be specifically v 1.11.1.

Use conda to install the required packages (with the python2 virtual environment activated). For example:

```
conda install networkx==1.11 click mpmath mock tqdm flake8
```

2 RUNNING THE PROGRAM

To operate pyDAEDALUSX, you must first start the backend python process, and then submit design jobs either through the GUI or through a python script.

2.1 STARTING THE BACKEND SERVER FOR PYDAEDALUSX

In terminal or Anaconda Prompt, with your virtual environment activated and operating Python v2.7, navigate to the pyDAEDALUSX folder, wherever you installed it. For example:

```
cd C:/users/name/pyDEADALUSX
```

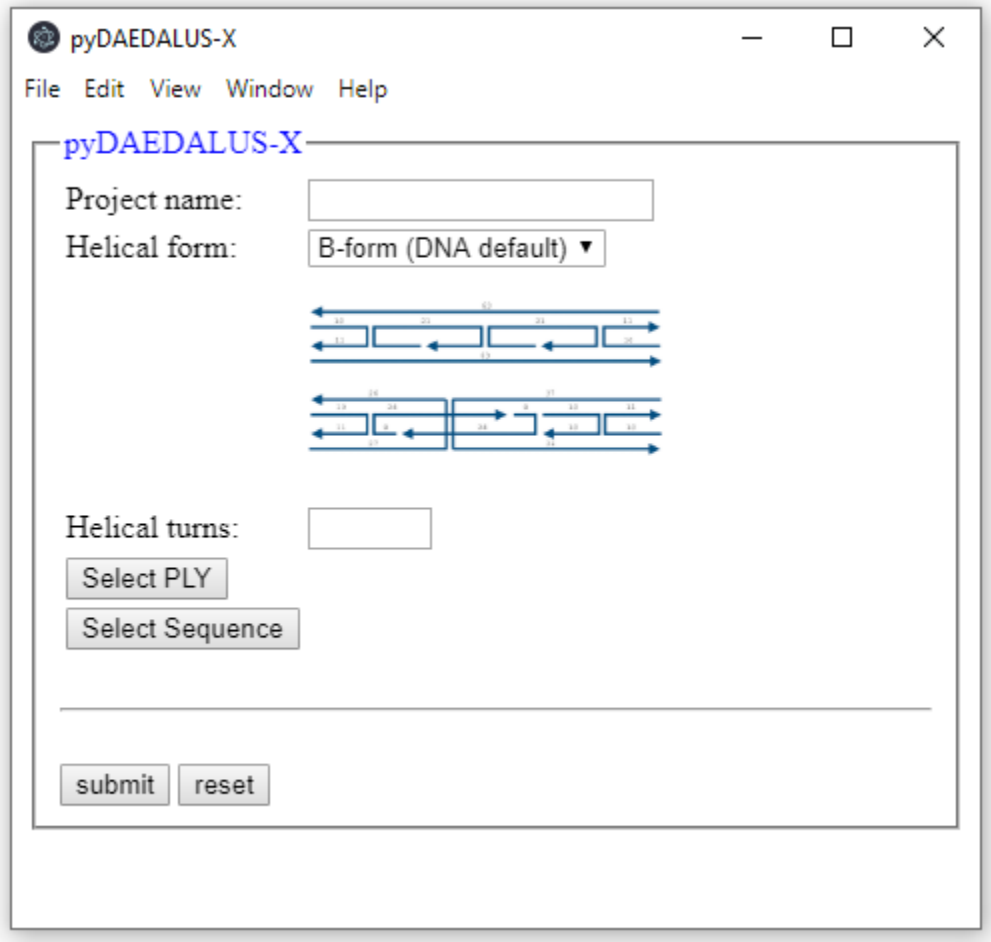
Start the backend process with Python:

```
python pyDAEDALUSX_program/DAEDserve.py
```

Note that output folders will be created in the directory from which you enter the above command. If you want them to output in a different folder, navigate there first and then specify the full pathname to ./pyDAEDALUSX/pyDAEDALUSX_program/DAEDserve.py when you start the backend process with Python.

2.2 USING THE GRAPHICAL USER INTERFACE TO SUBMIT DESIGN JOBS

To use the GUI, navigate in a File Explorer to the appropriate subfolder of pyDAEDALUSX and launch the application. The following window should appear:



Inputs include:

- Project name (a name for your design)
  This will be the name of the output folder created.

- Helical form (A-form or B-form)
  Note: Select A-form if your scaffold and/or staples will be RNA.

- Helical turns (for minimum edge length)
  The minimum edge length will be (# of helical turns) x (bp/turn).
  A-form helices have 11 nt per helical turn.
  B-form helices have approximately 10.5 nt per helical turn.

  Note: Designs using A-form helices must have a minimum edge length of at least 4 helical turns (44 bp). Designs using B-form helices may have a minimum edge length as low as 3 helical turns (31 bp).

- Target geometry (PLY file)

- Optional: Scaffold sequence (TXT file containing only the sequence)
  If you do not select a scaffold sequence file, the program will use M13mp18 phage sequence for scaffold lengths less than 7,249 nt and generate random sequence for larger scaffold lengths.

Click "Submit" to generate the design with your input parameters. The bottom of the window will show the word "Processing…" until the job is complete, at which time it will display "Fail!" or "Success!"

Longer edge lengths and geometries with more edges will require longer processing times.

2.3 SUBMITTING BATCH JOBS WITH A PYTHON SCRIPT

Instead of using the GUI, you can also submit design jobs with a python script. We recommend this method if you want to design many objects. A sample submission script is included in the pyDAEDALUSX folder as "example_submission_script.py"

```
import xmlrpc.client
import os
#
# connect to localhost, port 4242 running the RPC server
#
proxy = xmlrpc.client.ServerProxy("http://localhost:4242/")
multicall = xmlrpc.client.MultiCall(proxy)
#
# calc: function on the RPC server to route the origami
# calc(projectName, helicalForm, helicalTurns, plyFile, sequenceFile):
#   projectName: short descriptive name, name of the generated folder
#                  (write "M13.txt" to use default sequence)
#   helicalForm: 'Bform', 'Aform', 'Hybrid', 'Twisted'
#   helicalTurns: multiplied by 11 (Aforms) or 10.5 (Bform)
#                  to get minimum edge length
#   plyFile: Path to ply file
#   sequenceFile: Path to sequence file
#

ply_dir = "H:/PLY_Files" # directory containing input geometries

for geom in os.listdir(ply_dir):
```

```
    ply = os.path.join(ply_dir, geom) # path to geometry file
    multicall.calc(name,"Aform",4,ply,"M13.txt");

result = multicall()
```

To use the program default scaffold sequence, the equivalent of not choosing a scaffold sequence file in the GUI, use “M13.txt” as your scaffold sequence input in the script. The program will use M13mp18 phage sequence for scaffold lengths less than 7,249 nt and generate random sequence for larger scaffold lengths.

Run your script in python, either from a terminal or Anaconda Prompt window (separate from the instance running the backend process), or in your favorite python editor e.g. Spyder. In the former case, navigate in your command prompt window to the folder containing your submission script (using the command “cd”), and run the script with python, e.g.:

```
python example_submision_script.py
```

3 TROUBLESHOOTING

If a design fails (identifiable by a “Fail!” note at the bottom of the GUI, by missing output files, or by missing strand complements in the .pdb file), make sure:

- Your input scaffold sequence is long enough for the geometry and minimum edge length you have specified.
  Try submitting the job with the same inputs but no scaffold sequence. If the design is successful, note the length of the scaffold sequence used (in the “staples_[…].csv” file in the output folder, the final sequence entry corresponds to the scaffold sequence).
- You have used at least 3 helical turns as the minimum edge length for B-form designs, or at least 4 helical turns as the minimum edge length for A-form designs.
- Your PLY file is formatted correctly (http://paulbourke.net/dataformats/ply/).
  Try opening the .ply file in a 3D Viewer.
  Try generating a design with one of the included sample PLY file inputs.