

GenePy Description

GenePy is a Python tool that ports core functionality from miniEugene, a genetic specification tool within the Eugene ecosystem that is focused on user-declared rules for specifying DNA sequences. The goal of this project was to demonstrate the potential benefits for porting the functionality of Eugene to a Python package. Through a general use programming language, Eugene should become more accessible to users without a strong programming background that would like to explore Eugene. A Python port also encourages developers to integrate Eugene into custom synthetic biology workflows that require specification.

Similar to miniEugene, GenePy accepts user-declared rules to generate DNA sequences. To accomplish this, GenePy frames the placement of parts as a constraint solving problem in order to satisfy declared rules when generating potential DNA sequence solutions. In summary, we were largely successful in implementing the core functionalities of miniEugene and a graphical user interface that simplifies the use of the toolkit.

GenePy Components

GenePy Class

One of the major software components of GenePy is the genepy Python class. This class enables users to use miniEugene functionality when writing Python scripts. The class contains methods for declaring Eugene rules in a manner that should be familiar to Eugene users. A README file highlights all the methods available, along with an example script that uses the class to specify a DNA sequence.

Additionally, the class stores user-declared rules and DNA sequence solutions in a dictionary. Users may save this information in JSON files, enabling smooth data input/output with other software tools in a custom synthetic biology automated workflow.

GenePy GUI

The second major software component of GenePy is the GUI. It is launched with a single line of code. The main window pops up with buttons for adding rules, launching the console window, and running the solver.

It is recommended to open the console window first because it displays all the rules that are declared by the user along with the output DNA sequences.

Clicking the buttons for adding rules will open another window where the user can enter the parameters pertaining to the rule. There is a button for each of the rules.

Lastly, running the solver displays all DNA sequence solutions in the console window. If there are no sequence solutions, nothing is added in the console window.

There are spaces for defining the number of parts to be used and the number of solutions the user desires. When the button for running the solver is pressed, these values will be passed into the code backend.

GenePy Setup

To use GenePy in a Python script users must download the project repository from GitHub. Once downloaded, import the `genepy.py` file, which contains the `genepy` class, in your working Python script. Information regarding how to use the class can be found in the project documentation.

To use the GUI, import `genepy` and `tkinter` into the main script. Tkinter is the de facto standard GUI toolkit for Python. It comes included in the windows installer of Python but this may not be true for MacOS or Linux.