

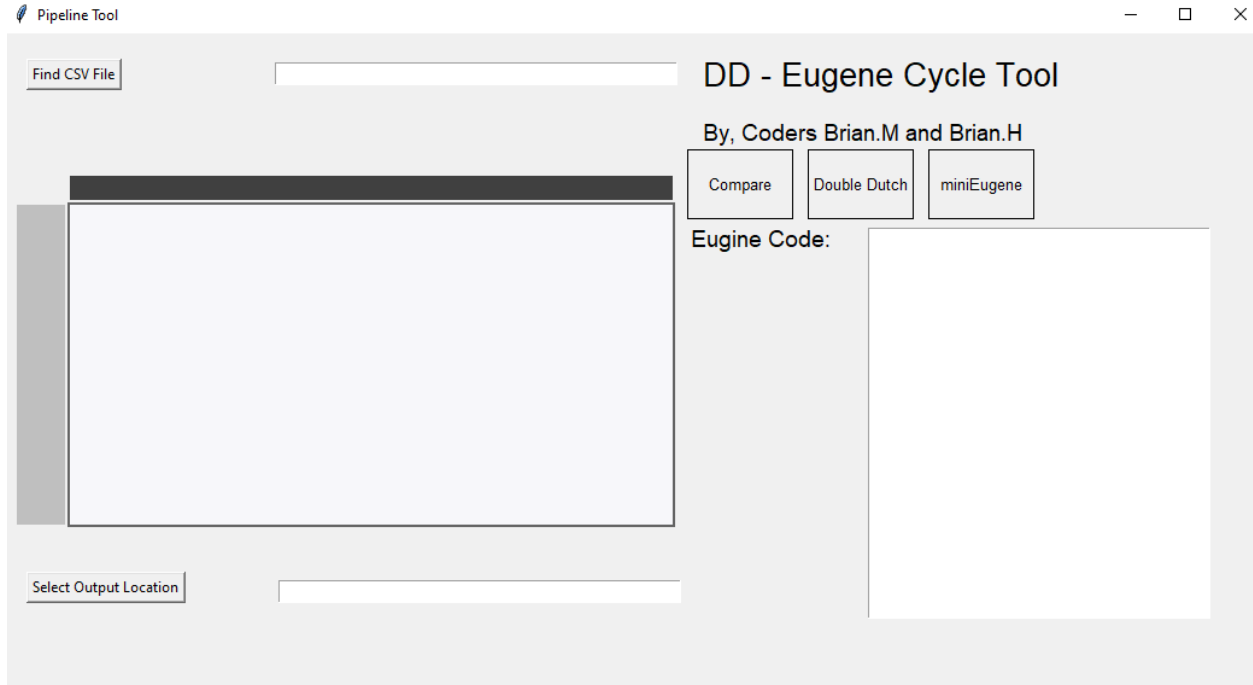
Final Project Report

Brian Mahabir, Landon Kushimi, Brian Jung, Zian Wang

Our project provides a streamlined process for giving users multiple design choices to choose from for Eugene using Double Dutch. We have designed a Graphical User Interface that interacts with Eugene and Double Dutch to give the user a clear comparison graph to showcase which designs might be useful for the user to pick.

The code is written in Python 3.7.9 and uses many preinstalled libraries to get the job done. The first major library the code uses is tkinter which is the main library to create the GUI. Scipy is used for data analysis if necessary when comparing the outputs of Eugene and Double Dutch. Finally, we have selenium which is arguably the most important library that our code uses. Selenium interfaces with front end graphical user interfaces to operate the code from the front end.

To preference why we ended up using selenium we have to delve a little bit into our trouble shooting problems. Originally, we were going to design the code to interact with Eugene and Double Dutch from the backend to both enter the user's inputs and extract the outputs. However, Eugene ended up being too cumbersome to sift through. We opted to go for miniEugene instead. However, miniEugene could not be run locally due to CORS errors surrounding jQuery and google analytics that we spent too much time trying to figure out. Double Dutch could not be run through a webserver since its server was down. We had to compromise with using selenium in tandem with python requests to get both Eugene and Double Dutch running the way we wanted.



First the user can code Eugene code in our built in text editor. Code mirroring is not supported on python and highlighting each technical word would be too hard so the text editor is in plain text.

The user the will hit miniEugene button and input their constraints to get an output. The code will then parse through the SBOL output and save the results

The user will then hit the Double Dutch button which will run the Double Dutch code locally on their machine. The user can then enter in the same credentials to get a plethora of different models. The code will also take save this.

The compare button will compare the outputs of Eugene and Double Dutch and give the user a table to show which models they might like. The user can pick a model for themselves create new constraints and run the code again to get finer results.

