Jo-Ann Meunier
Pranay Singh
EC 500 Hwk 1 Report

Jello, A Tool For Genetic Logic Synthesis

Introduction:

For this homework we were challenged to create a tool, which improves upon Cello. Cello is used for the simulation and evaluation of digital genetic logic circuits. This tool uses the concept of a circuit score to define how well the circuit works. We have created a working tool called Jello, which will aim to optimize a user specified circuit design to get the best possible circuit score. Jello is written in javascript using the node.js framework. The tool allows the user to upload a custom UCF and verilog file, which will then be submitted to Cello using the python cello API. The tool then runs an optimization algorithm which will greatly improve upon the original circuit score.

Methods:

Jello is a webapp that is written in javascript using the node.js framework. The optimization algorithm was written in python. In order to run python scripts within node.js we utilized the python-shell package, which gave us more flexibility because one member was well versed in python and the other in javascript. The tool makes use of the existing cello python API to connect to cello, submit and retrieve results. A python script then runs an optimization algorithm on the user's existing UCF file and displays the improved circuit score.

The optimization algorithm utilizes operations from both protein and DNA engineering in order to improve the original circuit score produced by Cello. The circuit score is used to evaluate how closely a genetic logic circuit models a specific boolean function. In order to improve upon this certain operations can be performed on the response function of each gate. For example if the user wants to use protein engineering the possible operations are to stretch and increase slope of the response function. If the user wants to use DNA engineering possible operations are to increase/ decrease the strength of the rbs or use a more/less active promoter. Methods are written within the Jello python script bio.py that can perform all of these operations.

The current methodology used to optimize the circuit score is to stretch and increase the slope of the response function as well as weaken the rbs. This involves modifying the user's UCF file. Specifically the values of ymin, ymax, K and n of each gate's response function are manipulated in such a way to give an improved score. Changing the values of each parameter does different operations on the circuit. In our algorithm we first search and find all of the intermediate gates that are used in the original circuit. This is done by searching the output of Cello after submitting the user's original json file. Then the gate's response function will be stretched by multiplying the ymax and dividing the ymin value by 1.5. This means that each gate will have higher ON

input values and lower OFF values, which will give the gate a more predictable behavior. Next the slope of the response function is increased by multiplying each gate's "n" value by a factor of 1.05. Finally, DNA engineering is done by weakening the ribosome binding site strength, which means modifying the value of "K". A weaker rbs would mean that higher "ON" value would be required to induce the promoter. This leaves room for a greater margin of error, which will allow the circuit to behave in a more predictable way. This approach is done iteratively to find the best possible combination of multiplying factors. The improved score is then returned to the user.

Results:

In order to test this tool several different circuit configurations were used. It was found that by using Jello the circuit designer could increase their circuit score by an average of 121% and completion time of 27. on the various circuit topologies used. The detailed results are shown below:

Circuit Topology	Original Score	Jello Score	% increase	Time to Completion (s)
3 in NOR	333.3	755.29	126.59	29.76
2 in AND	328.65	758.87	130.91	25.24
0xFE.v topology	47.14	97.52	106.86	27.63

Conclusions:

While there still exists many problems in the new field of genetic logic synthesis, it is clear that with the assistance of efficient computing huge advances are possible. Jello is a tool that can reliably provide the user with a solution to increase the score of their circuit and therefore obtain reliable results when the circuit is produced in the lab. This is hugely important for synthetic biologist because developing just a few boolean like gates can yield incredible results in genetic therapies and treatment. Jello is equipped with the framework to be expanded and improved upon. The current working algorithm makes use of protein engineering as well as DNA engineering. Because it has been shown that protein engineering is very difficult to realize in the lab, the algorithm should be improved in the future to only use DNA engineering. The current algorithm does not make the promoters more or less active. This is an avenue that could be explored further, and methods have been written in the existing python scripts to do this. Ultimately, the results of this homework have shown Jello to be a tool in which can be built off of and used to improve current knowledge of genetic circuit design.