EC 552 Computational Synthetic Biology for Engineers Project Report

Team Name: MARS BUILDER

Team Members: Ryuichi Ohhata, Yangruirui (Ron) Zhou

Short Description

MARS stands for Microfluidics Applications for Research in Synbio created by a BU team in iGEM 2017. The team designed standardized microfluidics chips that can be used in any synthetic biology lab. Our project is to implement the MARS library to LFR, add a missing component to 3DuF to enable it to recreate MARS device, and have the Guide Tool to give a control guidebook to generated JSON files of MARS designs from 3DuF. We also created the carrier pipelining algorithm in order to pipeline meter components in MARS devices, deleting unecessary carrier ports to simplify the design.

Function Description

3DuF

https://github.com/CIDARLAB/3DuF/pull/237

In this pull request, I created the meter component used in the MARS devices that was originally missing from the 3DuF. Channel width, design width and length, valve gap, valve radias, and rotation are parametrized. The meter component can now be rendered on 3DuF to create MARS devices.

LFR

https://github.com/CIDARLAB/pyLFR/pull/21

In this pull request, I added the MARS library to generator.py and created marsstrategy.py. This allows the LFR to compile LFR files with MARS designs using the library.

LFR Test Cases

https://github.com/CIDARLAB/LFR-TestCases/pull/2

This pull request contains the LFR files I created for eight MARS devices. These files were used to test LFR functionality with the MARS library.

Carrier Pipelining Algorithm

“code/pipelining_algorithm”

This folder contains my pipelining algorithm code that aims to be intergrated with LFR in the future. The pipelining_algorithm.py file contains the algorithm code, and other files are four MARS test cases and one failed test cases that’s more compelx than the MARS devices. The “out” folder includes all the generated dot files.

Guide Tool

“code/guide_tool/js/upload_read.js”

This file works for the step 1, which implement the function of upload and analyze. We analyze the content from the uploaded json file and transfer their data structure into the inner data structure used in guide tool.

“code/guide_tool/js/pointchoose.js”
This file works to add the components we click into a list, which can be used by other functions. The methodology of it is get the position of where MouseClick event happens and find the cloesest components to that position, whose type should be decided as an input parameter.

“code/guide_tool/js/drawCanvas.js”
This file works to draw a given components list with different colors on the canvas. The input information include components informations, which layer they are from, and some other parameters for special situations.

“code/guide_tool/js/canvas_auto.js”
This file works for pathway searching algorithm. If two components have overlap, we can think them connected. When one of the component is searched and added into the pathway, the other one should also be added unless it breaks the operation rule.

Special Instructions

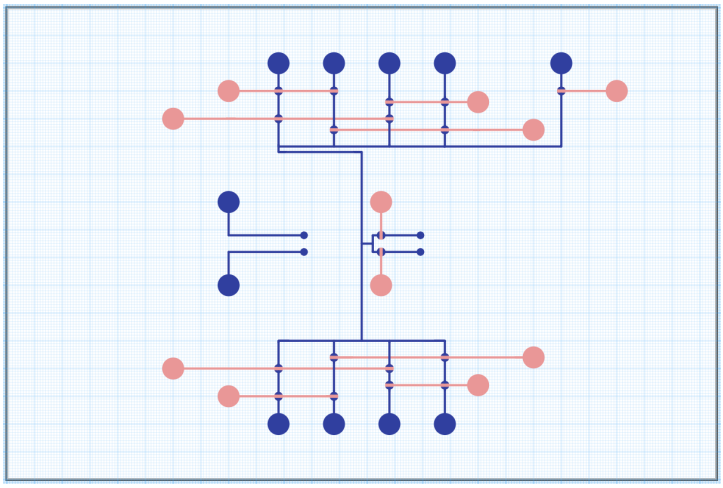
Pipelining Algorithm

Requirements: Python3.8, networkx dependency, and dot dependency

Each MARS device test cases includes *pa.draw([name])* in the end. This is the function that generates a dot file. If only the first argument (name of the file) is passed, it will generate an unpipelined graph. If True is passed in as the second argument (i.e. *pa.draw(“ligation”, True)*), it will generate a pipelined graph using the algorithm. *python3 [file_name]* to run the program.

Guide Tool

Just click the index.html, then you can see the webpage, you will see the similar layout after you upload the design files from 3duf



Step 1

Step 2

Step 3

Step 4

Step 5

Step 1

Upload 3Duf design file in json. Click upload below.

Upload

Type of Elements	Numbers
Via	4
RoundedChannel	55
Port	11
Valve3D	19
Valve3D_control	19
Port_control	11

Next