B.A.N.G.O.S.

Bayesian Analysis of Genomic (circuit) Optimizer for Sensitivity

Sofia Briquet, Scott Cazier, Kevin Delgado, and Guadalupe Garcia

Project Description

Our project is a command line tool for analysis of genetic circuits. The tool takes in user input such as number of input signals, circuit name, and a priority ranking list of toxicity, GC content, part cost and circuit score. The tool then performs a stochastic simulation and sensitivity analysis to the input signals and parts parameters by applying gaussian noise. Weights are then added to each component based on the priority ranking. The result is the average score and the resulting GC content, part cost, and toxicity of the circuit.

Main Software Components

USER INPUT

The circuit name and number of input signals provides the program with the gate to be used and how many signals to be used in it. The inputted priority list will help determine the weights added to each category after the stochastic simulation and sensitivity analysis is performed.

STOCHASTIC SIMULATION

The stochastic simulation applies gaussian (normal) noise to each parameter of the input signals to later determine which is most affected. The team used normal variation, but it would be trivial to add other distribution selections if desired.

GC CONTENT -B(x)

The GC content function calculates the GC content of each part sequence. This served as a proxy for ease of assembly for a finished design. Extending to homology and sticky ends would be even better representations of assembly/manufacturability.

PART COST CALCULATION - C(x)

The cost function calculates the cost of each part by applying a 0 to parts found in an added repository or 1 to parts not found. This can be extended to “parts in the freezer” library or online repository based on whether a part already exists.

CELLO SCORE (Circuit Gain) - A(x)

The score for the circuit based on its gain, pulled directly from Cello and is returned to be a key variable in the optimization (BANGO Score). - Not done by this project but used.

TOXICITY SCORE - D(x)

This is already an output of the CelloAPI, our tool utilizes it so the user can decide whether to move forward with the circuit design based on toxicity. Cello presents this as a vector based on logic gate input states, and should be averaged over the returned values. Note this appears to have not made it into our submitted code.

OVERALL PERFORMANCE (BANGO Score)

Rates a design from 0 to 1 - counting user weighting to provide a relative scoring metric. Each factor is normalized with experimentally or literature derived values and then multiplied by its previously assigned weight. This provides an objective function to be optimized as a single unitless parameter.

$$Max: G(x) = \sum_{X} A(x) - B(x) - C(x) - D(x)$$

SENSITIVITY ANALYSIS

The sensitivity provides a visualization of which parameter is most affected by applied noise and allows the user assesses the robustness of the circuit design. This assesses the variation associated with each input parameter, it's iterative perturbations and displays its effect on the circuit's Cello score and could be used for the other performance variables using variation based sensitivity, plotting the values as follows in the equation below.

$$Var(E_{X\sim i}\,(G|X_i)$$

Instructions to Install and Run BANGOS

INSTALLATION

1. Make sure you can install and run CelloAPI on your machine
    a. To install the CelloAPI, follow the instructions described in: https://github.com/CIDARLAB/celloapi2
2. Download the modified iGEM's Registry of Standard Biological Parts at https://drive.google.com/file/d/1DIQIdmrU2aYTFxdpX7-2oLN5a5YmRYYI/view?usp=sharing
    a. This is a modified parts list that originally comes from here: http://parts.igem.org/Registry_API.
3. Clone the B.A.N.G.O.S project from the Github Repository
    a. https://github.com/sebriquet/banjos
4. To install the remaining dependencies used for B.A.N.G.O.S, run the following command:
    a. *cd CODE*
    b. *pip install -r requirements.txt*

RUN INSTRUCTIONS

1. Before running ensure your input file contains the following:
    a. Input JSON file
    b. UCF JSON file
    c. Output JSON file
    d. Verilog file with circuit description
2. In the command window type: "*python3 main.py input*"
3. You will be asked to specify your preferences for the weighting scheme: which metrics matter most to you, as well as the name of the device, your circuit (example: **and.v**) and the number of input signals your circuit needs.
4. As an output, users should expect plots relating to sensitivity analysis, with toxicity, manufacturing cost, assembly cost, and cello score outputted on the main terminal window.