



element.

<simulation\_name>

This is the name given to a simulation. This is also the name of the directory within which the data directory that will contain the output csv files will be placed. E.g. if simulation\_name is "fast\_codons" the data will be placed into "data/fast\_codons/".

cell attributes

-----

These are found within the <cell> element inside <simulation>:

<total\_ribosomes>

This is the total number of ribosomes within the cell available to the circuits. This should be an integer number.

monitor attributes

-----

These are found within the <monitor> element inside <simulation>:

<mRNA\_number>

This is the number of mRNA per cell for the monitor circuit. This should be an integer number.

<rbs\_strength>

This is the relative strength of the RBS for the monitor circuit. This should be a number.

<betas>

This declares the elongation rates for the monitor. This is defined as a series of numbers separated by commas. The length of the circuit is defined by the number of values within the <betas> element.

<alpha\_plus\_scale>

This is the scaling for ribosome-RBS affinity. This should not be changed from 0.0001 without an understanding of how this value is used. See models.py for details.

<alpha\_minus\_scale>

This is the scaling for ribosome-RBS affinity. This should not be changed from 200 without an understanding of how this value is used. See models.py for details.

circuit attributes

-----

<mRNA\_number\_steps>

This is the number of different values for the number of mRNA per cell used in the simulation landscape.

<mRNA\_number\_step>

This is the difference between the numbers of mRNA per cell that are to be simulated, it is also the smallest mRNA per cell that will be tested. E.g. if this is 50 and mrna\_number\_steps is 100, then the number of mRNA per cell that will be used to simulate is 50, 100, 150, ..., 5000

<rbs\_strength\_steps>

This is the number of different values for the RBS strength used in the simulation landscape.

<rbs\_strength\_step>

This is the difference between the RBS strengths that are to be simulated, it is also the smallest RBS strength that will be tested. E.g. if this is 0.5 and rbs\_strength\_steps is 20, then the RBS strengths that will be used to simulate is 0.5, 1.0, 1.5, ..., 10.0

<betas>

This declares the elongation rates for the circuit. This is defined as a series of numbers separated by commas. The length of the circuit is defined by the number of values within the <betas> element.

<alpha\_plus\_scale>

This is the scaling for ribosome-RBS affinity. This should not be changed from 0.0001 without an understanding of how this value is used. See models.py for details.

<alpha\_minus\_scale>

This is the scaling for ribosome-RBS affinity. This should not be changed from 200 without an understanding of how this value is used. See models.py for details.

## RUNNING THE SIMULATION

=====

This package should be unzipped and using the command line, the user should navigate into the unzipped directory. Once in the directory, the user can run the simulation using the following command:

```
python simulate_landscape.py [xml_location]
```

where [xml\_location] is to be replaced by the name of the xml file (if in the same directory, e.g. fast\_codons\_parameters.xml), or the location and name of the xml file (e.g. /Users/username/synbio\_simulations/parameter\_files/fast\_codon\_paramters.xml). This will then output the data into csv files. Each row represents a single RBS strength and each column represents a single number of mRNA. The top left cell represents the lowest RBS strength and lowest number of mRNA with the number of RBS increasing by rbs\_strength\_step in each subsequent row and the number of mRNA increasing by mrna\_number\_step in each subsequent column.

## CONSTITUENT FILES

=====

models.py

-----

This python file includes two classes Cell and Circuit that can be used to simulate the behaviour of gene circuits within a cell. In order to simulate a cell containing a synthetic gene and a burden monitor, two separate instances of Circuit must be created (one for the synthetic gene and one for the monitor). These instances can be passed as an argument into the initiation of a cell instance, along with the number of ribosomes. The Cell class has a simulate function which returns the number of free ribosomes and the distribution of ribosomes along the transcripts of each circuit. These distributions, along with the betas for the circuit can be used to calculate the protein production rate for each circuit, the total ribosomes sequestered on each circuit's transcripts and the efficiency of the circuit.

simulations.py

-----

This file contains a SimulationResult class which parses the raw simulation data from the Cell class from models.py. It parses the raw data into key metrics such as circuit output, circuit ribosomal usage etc. This file also contains a function simulate\_landscape that takes key parameters to simulate an array of cells containing a single synthetic circuit and a single monitor circuit. These landscapes are created across various mRNA numbers and RBS strengths.

simulate\_landscape.py

-----

This is a command line executable file that takes the location of an XML file, e.g. fast\_codons\_parameters.xml, as an argument. This then takes parameters from the XML file, runs the simulations across the defined landscape and outputs the files into a defined directory in a series of csv files:

circuit\_out.csv

This file contains the protein production rates per cell for the synthetic circuit.

circuit\_ribs.csv

This file contains the total numbers of ribosomes sequestered on all of the transcripts for the synthetic circuit.

free\_ribs.csv

This file contains the numbers of free ribosomes per cell.

monitor\_out.csv

This file contains the protein production rates per cell for the burden monitor.

monitor\_ribs.csv

This file contains the total numbers of ribosomes sequestered on all of the transcripts for the burden monitor.

mRNA\_amount.csv

This file contains a vector of all the mRNA amounts used in the simulations across the landscape.

RBS\_str.csv

This file contains a vector of all the RBS strengths used in the simulations across the landscape.

## PLOTTING THE SIMULATION RESULTS

=====

The data contained in the .csv files generated once the Python simulations are complete can be visualised using the Matlab script Plot\_Matlab\_3D\_CSV.m. This will create a pdf file called Heat\_Map\_Efficiency.pdf representing the Normalised Efficiency, calculated as the product of circuit output and number of free ribosomes, normalised to the maximum efficiency value.