Systems Biology – Exercises

Week 9: Genetic Algorithm (GA) - Part 2

Genetic Algorithm

- We have covered mechanisms of evolution by natural selection and want to investigate how this idea can be realized by actual computer code. Last week, we considered two mechanisms—and their respective Python code—namely mutation and recombination.
- The second part of this exercise is to understand and code a process that favors beneficial traits of a gene or DNA. In preparation, we have three variations of a fitness function (or the inverse, as given here, which is closer to the Python code, Equations 1 to 3).

Fitness Function -1

$$1/f_{\text{fitness\#1}} = \sum_{i=1}^{n} |x_i - y_i|$$ Equation 1

$$1/f_{\text{fitness\#2}} = \sum_{i=1}^{n} (x_i - y_i)^2$$ Equation 2

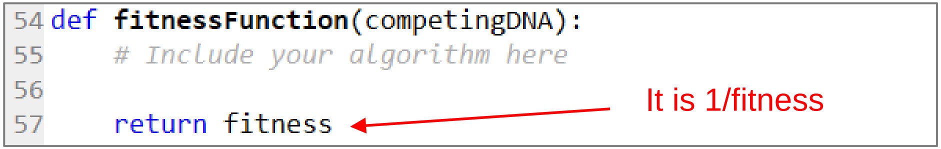
$$1/f_{\text{fitness\#3}} = \sum_{i=1}^{n} 1 \quad \forall x_i \neq y_i$$ Equation 3

Fitness Function -2

- In the last exercise class, the provided *dummy fitness function* assigned to any passed DNA the same penalty of 0 (or any other number that you chose). With this tutorial, we will implement different fitness functions and look at their performance.
- **Fitness Function:** This is a crucial part of the Genetic Algorithm, as computational time and the outcome can heavily depend on the choice. A fitness function should be least complex to compute and not too quickly converge to a local optimum. In this case, we are looking at the rate of discrepancy between the DNA of a current population and the target (or environment).

Exercise 1

- Following the Equations 1 to 3, implement at least three different types of penalty (three different fitness functions). To convert a character into a numerical value, we can use the decimal representation of the corresponding ASCII binary number. In Python, this can be done by using the ord() function in the form ord("a")



```python
def fitnessFunction(competingDNA):
    # Include your algorithm here

    return fitness
```

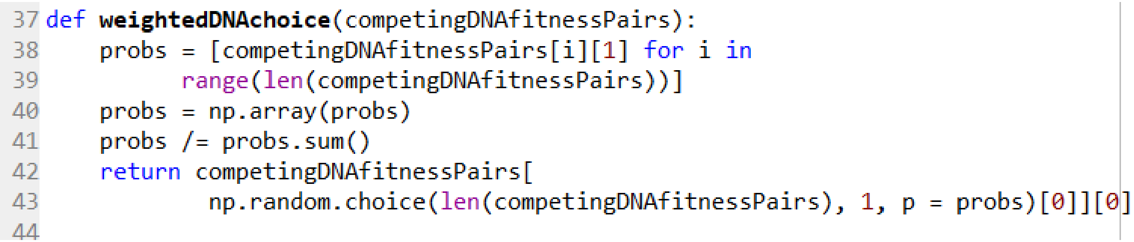
Code 1. Fitness Function.

Weighted Selection of DNA

- During the lecture, you have seen how the weighted choice function is working. The fitness value of each DNA is replaced by their respective ratio or proportion.
- The line probs /= probs.sum() normalizes these values so the list can be passed on as an array of probabilities. It will then be implemented in the form of

np.random.choice(list, frequency, p = probabilities)

Source code of Weighted Selection



```python
def weightedDNAchoice(competingDNAfitnessPairs):
    probs = [competingDNAfitnessPairs[i][1] for i in
          range(len(competingDNAfitnessPairs))]
    probs = np.array(probs)
    probs /= probs.sum()
    return competingDNAfitnessPairs[
            np.random.choice(len(competingDNAfitnessPairs), 1, p = probs)[0]][0]

```

Code 2. Weighted Selection of DNA for Next Generation.

Exercise 2

- To get a better understanding of the mechanism, create a separate Python file, strip the function definition and declaration (remove the line starting with def and replace the return command with a print expression) and change the code according to the following instructions.
- Feel free to change the array names. The first line should read probs = [0.001, 0.001, 0.01, 0.988]. Rewrite the code so that the following values from an array are selected with their respective chances:
- "very low", "still very low": 1/1000
- "rather low": 1/100
- "high": 988/1000.

Homework

Due next Wednesday (17:00, Dec. 1st, 2021) electronically to manaba+R.

• File format: YourStudentID_Week09_n.py/pdf (ID without hyphen, e.g., 12345678901_Week09_01.py/pdf).

• Your code must include your own comments for all code sections. Go line-by-line, if necessary. Comments in your program must be full sentences and reflect your understanding of the code.

Q1. Implement at least three different fitness functions (Equations 1 to 3 at least) in your source code (can be improved) of the Week-8 homework, and submit the improved version (including your fitness functions) of Genetic Algorithm.

2 How many fitness functions were you able to implement? How was their performance? Describe how your fitness functions work and which one showed good or best results. Can you think of reasons, why fitness functions perform this differently? Aim for a total of at least 200 words.

Q1: YourStudentID_Week09_01.py

Q2: YourStudentID_Week09_02.pdf

Systems Biology – Exercises

Week 10: Neural Network Part (1)