Biological Computation 20.181
Homework 10

Note: If your grade on this assignment increases your HW average then it will be weighted as two assignments, otherwise it will be weighted as a single assignment.

Part 1. Accompanying this assignment is some python code that implements Dan Gillesipie’s “Direct Method” for computing the exact behavior of discrete chemical systems (gillespie.py). Modify the code so that it implements Dan’s “First Reaction Method.” See some potentially useful hints at the end of the assignment if you want.

Part 2. You should now have two discrete reaction event simulators. Use each simulator to compute the behavior of the system specified by the provided input files (system1.inp, species1.inp) over a period of 100 seconds, sampling the system state every 0.1 seconds. Report how long it takes each simulator to run and whether or not this time agrees with your expectation of the computational costs of each algorithm.

Part 3. Produce a plot of the numbers of proteins A->D over time (100 seconds). Also, draw a simple sketch of the species and the reactions that define this system. What do you expect this system to do? Is the computed behavior what you expect?

Hints from the TAs.

* to run the direct method in gillespie.py type “python <system input file> <species initialization file name> <output file name> <# of output time points>”. E.g., “python foo.in x.in foo.out 100”

* to do division, be forewarned that python usually defaults to int division, not float division. in other words, 1/5 = 0 in python. to make python use greater precision, give your first value a float (something with decimal places); i.e.: 1.0/5 = 0.2

* if you get ZeroDivisionErrors, it's because you're trying to divide by zero. try adding a really small value to the denominator to erase these errors. for instance, 1/0 makes python barf; 1/(0 + math.exp(-200)) is ok

* to sort a dictionary by its values is a bit tricky. here's one example of how to do it:

```
>>> d = {’bob’:4, ‘krishna’:9, ‘hikari’:5}
>>> e = d.keys()
>>> e.sort(cmp=lambda a,b: cmp(d[a],d[b]))
>>> e
[’bob’, ‘hikari’, ‘krishna’]
```