INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Homework Assignment - 5

Indraprastha Insitute of Information Technology,
Delhi

Computer Science and Applied Mathematics

Introduction to Quantitative Biology (BIO213)

Author:
Aditya Chetan
(2016217)

Date: February 19, 2018

1 Solution (a)

In the steady state, we will have,

$$
\begin{aligned}
\frac{dP_O}{dt} &= 0 \\
-k_{-1}P_O + k_1 P_C &= 0 \\
-k_{-1}P_O &= -k_1 P_C \\
\frac{P_O}{P_C} &= \frac{k_1}{k_{-1}}
\end{aligned}
$$

[Same can be obtained by taking $\frac{dP_C}{dt} = 0$]

2 Solution (b)

From the given single channel recording data, we can see that the open intervals are as follows:

- First open interval is approx. 1ms
- Second open interval is approx. 2ms
- Third open interval is approx. 1ms

Hence, we have,

$$P_O = \frac{\text{Sum of all open intervals}}{\text{Total time duration}} = \frac{1+2+1}{27} = \frac{4}{27}$$

Similarly, we have,

$$P_C = \frac{\text{Total time - Total open time}}{\text{Total time duration}} = \frac{27-4}{27} = \frac{23}{27}$$

$$\therefore \frac{P_O}{P_C} = \frac{\frac{4}{27}}{\frac{23}{27}} = \frac{4}{23}$$

Hence, answer is $\frac{4}{23}$

3 Solution (c)

The code for the simulation is given below:

```python
import random
import matplotlib.pyplot as plt

def generate_data(n, p_oc, p_co):

    """
        This function is for generating a random array
        that will denote the state of the ion channel
        for the number of iterations provided as input
    """

    # Variable to denote state of the channel
    # 1 => Channel is open
    # 0 => Channel is closed
    state = 0

    # List to store values in all iterations
    trials = [state]

    for i in range(n - 1):

        trial = random.uniform(0, 1)
        if state == 1:

            if trial < p_oc:

                state = 0

        elif state == 0:

            if trial < p_co:

                state = 1

        trials.append(state)

    return trials


def plot_graph(trials):

    plt.plot(trials, linewidth=0.8)
    plt.xlabel("Time (in ms)")
    plt.ylabel("Current (in pA)")
    plt.show()



p_oc = 2/5
p_co = 1/5
n = 100

val = generate_data(n, p_oc, p_co)
plot_graph(val)

open_t = sum(val)
p_open = open_t/n
p_close = 1 - p_open

print("P_open =", p_open)
print("P_close =", p_close)
print("P_open / P_close =", p_open/p_close)
```

I calculated the ratio $\frac{P_O}{P_C}$ twice. The resulting graphs and outputs are shown below:

Trial 1

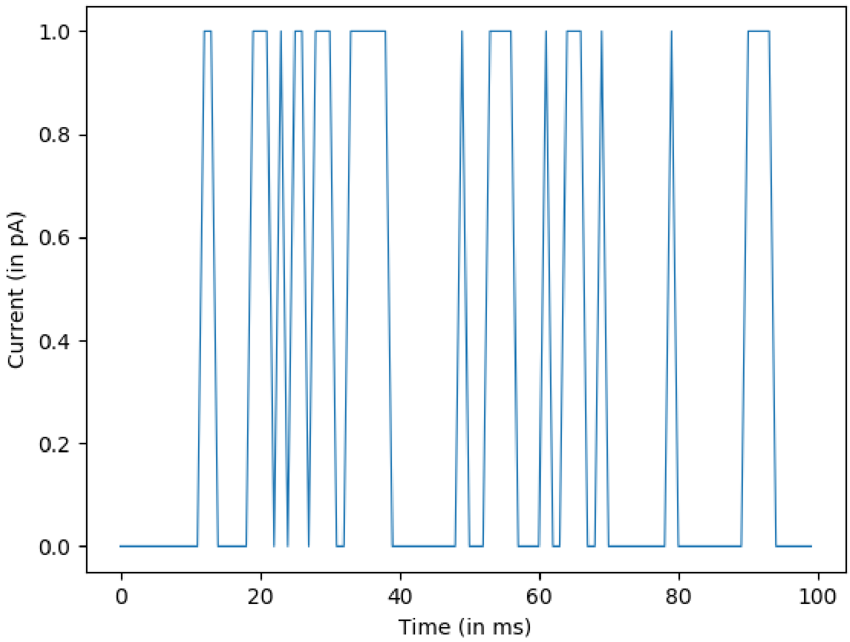

Figure 1: The first Monte-Carlo Simulation

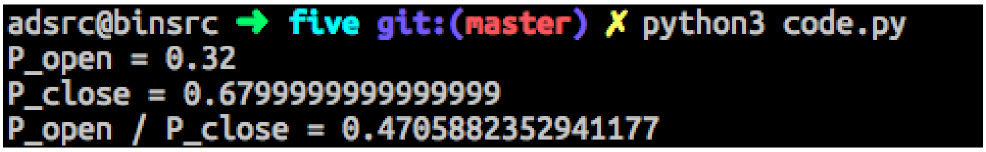

Figure 2: The console output from the first trial

Trial 2

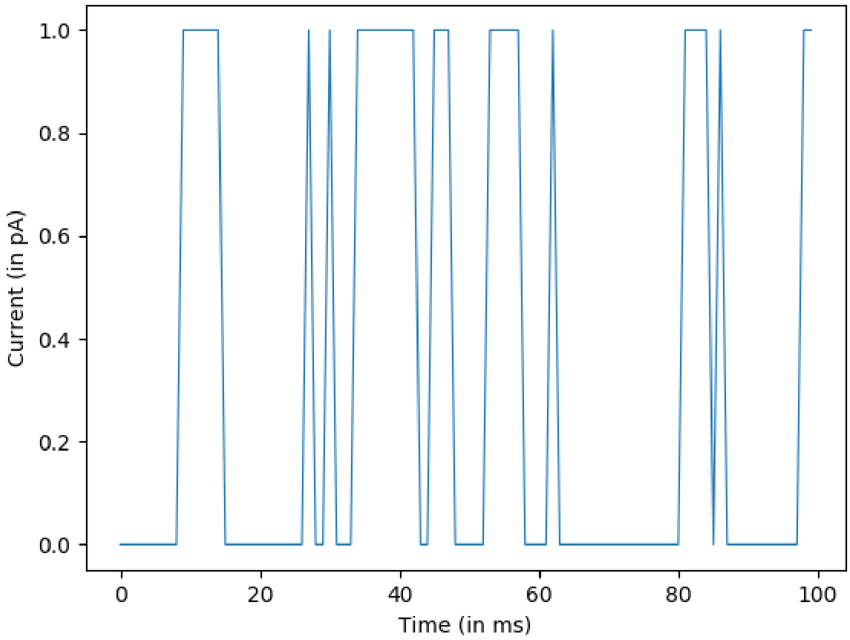

Figure 3: The second Monte-Carlo Simulation

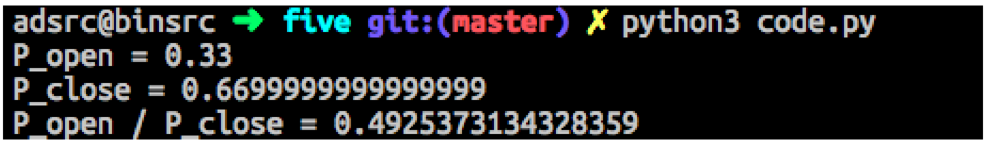

Figure 4: The console output from the second trial

4 Challenge Solution

From the master equation, we know that at equilibrium,

$$\frac{P_O}{P_C} = \frac{k_1}{k_{-1}}$$

Now multiplying and dividing the RHS by $\Delta t = 1$ms, we get,

$$\frac{P_O}{P_C} = \frac{k_1 \Delta t}{k_{-1} \Delta t} = \frac{P_{\text{close}\to\text{open}}}{P_{\text{open}\to\text{close}}}$$

This multiplication by Δt helps in converting the rate constants, k_1 and k_{-1} which have units of s^{-1} to probabilities $P_{\text{close}\to\text{open}}$ and $P_{\text{open}\to\text{close}}$ respectively with no units.

Now, in our case,

$$P_{\text{close}\to\text{open}} = \frac{1}{5}$$
$$P_{\text{open}\to\text{close}} = \frac{2}{5}$$
$$\therefore \frac{P_O}{P_C} = \frac{k_1 \Delta t}{k_{-1} \Delta t} = \frac{P_{\text{close}\to\text{open}}}{P_{\text{open}\to\text{close}}} = \frac{\frac{1}{5}}{\frac{2}{5}} = \frac{1}{2} = 0.5$$

For Trial 1

As we can see in 3, the terminal output of the code gives,

$$\frac{P_O}{P_C} = 0.47 \approx 0.5$$

Hence, the output of Trial 1 concurs with the result that we obtained in Solution (a) (1).

For Trial 2

As we can see in 3, the terminal output of the code gives,

$$\frac{P_O}{P_C} = 0.49 \approx 0.5$$

Hence, the output of Trial 2 concurs with the result that we obtained in Solution (a) (1). Hence, the results of both the trials have been successfully verified.