Modeling Bacterial Chemotaxis Using a Biased, Density-Based Random Walk

Authors: Hudson Tien, Yeowon Yoon

Cosmos Cluster 5 Biophysics

Abstract

Chemotaxis, the directed movement of organisms in response to chemical gradients, is a fundamental phenomenon in biology, exemplified by bacterial behavior. In this study, we modeled bacterial chemotaxis using Python libraries such as NumPy, Random, Math, and Matplotlib, creating a computational framework that simulates bacterial movement within diffusing food fields. Our model employs a biased random walk, wherein bacteria dynamically adjust their motion—reducing tumbling frequency and increasing step size—based on the detection of favorable concentration gradients. This approach not only replicates the behavior observed in wild bacteria but also demonstrates the utility of random walk models in exploring the interplay between stochasticity and environmental feedback.

Biased random walks represent a unifying theme across biology, underlying diverse processes such as animal foraging patterns, immune cell migration, and the spread of infectious agents. By incorporating localized probabilistic adjustments influenced by environmental conditions, these models reveal how small, stochastic variations at the individual level can significantly impact macroscopic outcomes, such as resource acquisition, population distribution, and ecological stability. Our findings highlight the role of diffusion dynamics and gradient detection in shaping bacterial navigation, offering insights into how microorganisms exploit stochastic strategies to thrive in heterogeneous environments.

The implications of this study extend beyond microbiology to broader fields such as ecology, developmental biology, and systems biology. By bridging bacterial chemotaxis with the principles of biased random walks, this work contributes to the understanding of stochastic processes in complex biological systems. Additionally, our model provides a foundation for potential applications, including the development of artificial navigation systems inspired by biological behaviors, improved drug delivery

mechanisms that mimic chemotactic principles, and predictive models for population dynamics in response to environmental changes.

Introduction

Bacterial chemotaxis, the movement of bacteria in response to chemical gradients, is a fundamental process in microbiology. Bacteria adjust their motion to increase their proximity to nutrients by reducing tumbling frequency and moving more persistently in favorable conditions. This study develops a computational model that replicates these behaviors, exploring chemotactic efficiency under varying diffusion coefficients and initial conditions.

Methods

1. Computational Framework

- **Random Walk**: Movement is modeled as a sequence of random turns, with each step direction randomized by an angle in radians.
- **Food Diffusion**: The concentration of food is represented using a Gaussian distribution:

$$C(x, y, t) = \frac{N}{4\pi Dt} e^{-(\frac{r^2}{4Dt})}$$

where:

- C: concentration
- N: number of molecules
- D: diffusion coefficient
- t: time
- r: radial distance from the origin.

Chemotactic Bias: Movement length depends on concentration gradients:

- If the current concentration (C) exceeds the previous concentration (Cprev), bacteria take a longer step.
- If C < Cprev, bacteria reduce their movement.

2. Simulation Procedure

The program consists of:

1. **Initialization**: Arrays posesX and posesY store bacterial coordinates over time. Environmental parameters, including grid dimensions, diffusion coefficients, and starting positions, are defined.
2. **Randomized Movement**: At each step:
- Compute Cprev.
- Randomize the turning angle and move the bacteria accordingly.
- Check boundaries and adjust positions if needed.
3. **Chemotactic Adjustment**: Compare C to Cprev and adjust step length using the timeWalk function.
4. **Visualization**: Generate heatmaps of food concentration with bacterial paths overlaid.

Results

Model Descriptions

Model 1: High Diffusion Coefficient

- **Diffusion Coefficient (D):** Large
- **Outcome**: Bacteria detect the food gradient from a distance and move directly toward the target.

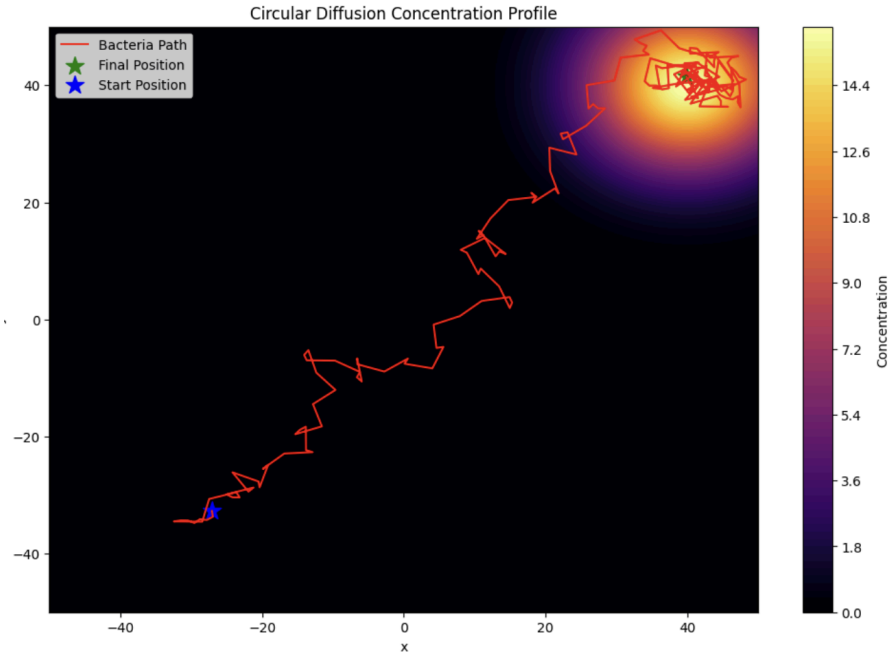
Model 2: Moderate Diffusion Coefficient

- **Diffusion Coefficient (D):** Medium
- **Outcome**: Bacteria alternate between exploration and direct movement upon detecting the food source.

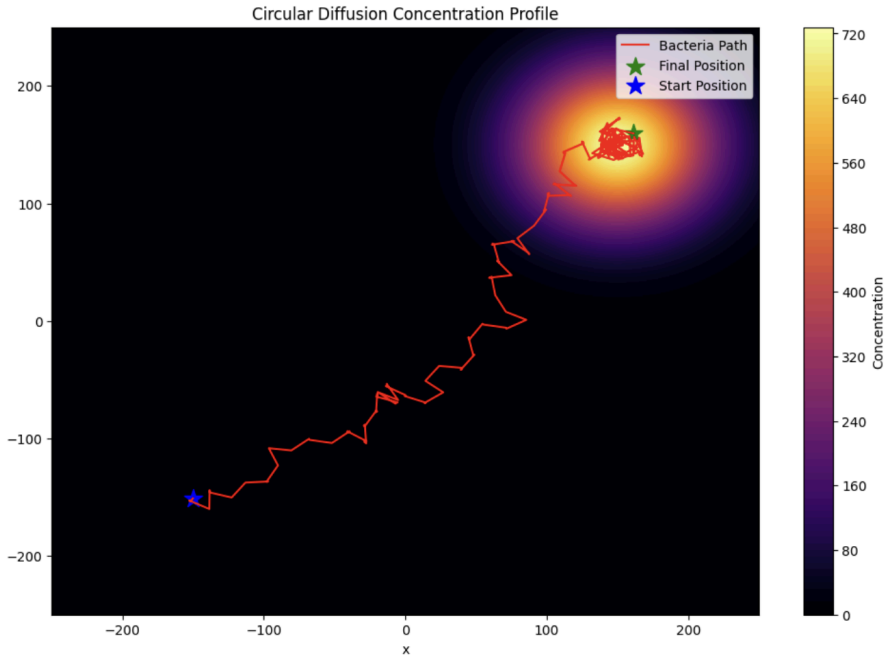
Model 3: Low Diffusion Coefficient

- **Diffusion Coefficient (D):** Small
- **Outcome**: Bacteria exhibit more localized exploration, relying on proximity to the food source due to the weak gradient.

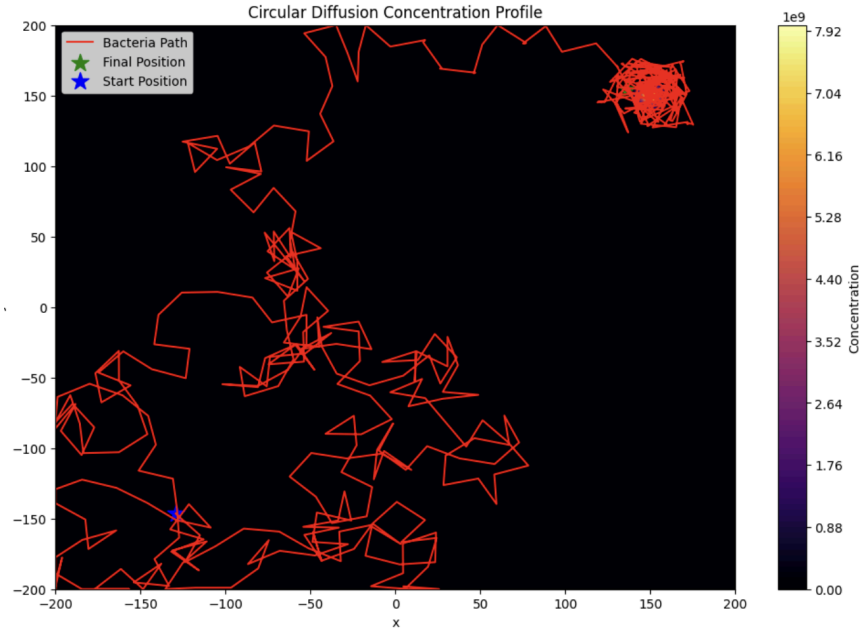
Visualization



- **Figure 1**: Heatmap and bacterial path for Model 1, showing a direct trajectory to the target.



- **Figure 2**: Heatmap and bacterial path for Model 2, illustrating mixed exploratory and targeted behaviors.



- **Figure 3**: Heatmap and bacterial path for Model 3, depicting localized movement near the food source.

Discussion

Key Findings

- Higher diffusion coefficients enable bacteria to detect gradients from farther distances, resulting in more direct movement.
- Lower diffusion coefficients reduce gradient detectability, forcing bacteria to rely on localized exploration.
- The model effectively replicates natural bacterial behaviors and demonstrates how environmental conditions influence chemotactic efficiency.

Limitations

- Assumption of uniform Gaussian diffusion may oversimplify real-world scenarios with multiple competing gradients or physical barriers.
- The model does not account for stochastic noise or interactions between multiple agents.

Future Directions

- Incorporate three-dimensional environments to simulate complex scenarios.
- Introduce multi-agent systems to study collective chemotactic behaviors.
- Enhance realism by modeling non-Gaussian diffusion patterns or adding environmental obstacles.

Conclusion

This study successfully modeled bacterial chemotaxis using a density-based biased random walk. The findings demonstrate the adaptability of bacterial movement under varying diffusion conditions and provide insights into chemotactic behavior in nutrient-rich environments.

Python Implementation

```
import numpy as np
import random as rd
import math
import matplotlib.pyplot as plt

def circularDiffusion(origin, point, timeOfSpread, diffusion_coefficient,
numMolec):
    x_origin, y_origin = origin
    x_point, y_point = point

    if timeOfSpread <= 0:
        return 0

    # Calculate the distance from the origin to the point
    distance = math.sqrt((x_point - x_origin) ** 2 + (y_point - y_origin)
** 2)

    # Calculate the concentration using a Gaussian distribution model
    concentration = numMolec * (1 / (4 * math.pi * diffusion_coefficient *
timeOfSpread)) * math.exp(-distance**2 / (4 * diffusion_coefficient *
timeOfSpread))

    return concentration
```

```
def randAngle():
    return rd.uniform(0, 2 * np.pi)

def BactSimulSetSpread(width, height, subdiv, timeOfSpread,
diffusion_coefficient, numMolec, steps, bactPos, goal, distance):
    def timeWalk(Cprev, C,distance):
        return distance if C >= Cprev else 0

    # Initialize bacteria position and concentration arrays
    posesX = np.zeros(steps)
    posesY = np.zeros(steps)

    for i in range(steps):

        # Calculate previous concentration
        Cprev = circularDiffusion(goal, bactPos, timeOfSpread,
diffusion_coefficient, numMolec)
        bactAngle = randAngle()

        # Bacteria "tumbles"
        new_x = bactPos[0] + np.cos(bactAngle)
        new_y = bactPos[1] + np.sin(bactAngle)

        # Keep bacteria within the bounds
        if new_x <-width / 2:
          bactPos[0] = -width/2
        elif new_x > width / 2:
          bactPos[0] = height/2
        else:
          bactPos[0] = new_x

        if new_y <- height / 2:
          bactPos[1] = -height/2
        elif new_y > height/ 2:
          bactPos[1] = height/2
        else:
```

```
          bactPos[1] = new_y

        # Calculate current concentration
        C = circularDiffusion(goal, bactPos, timeOfSpread,
diffusion_coefficient, numMolec)
        time = timeWalk(Cprev, C, distance)

        # Bacteria "runs"
        new_x = bactPos[0] + np.cos(bactAngle) * time
        new_y = bactPos[1] + np.sin(bactAngle) * time



        # Keep bacteria within the bounds
        if new_x <-width / 2:
          bactPos[0] = -width/2
        elif new_x > width / 2:
          bactPos[0] = height/2
        else:
          bactPos[0] = new_x

        if new_y <-height / 2:
          bactPos[1] = -height/2
        elif new_y > height/ 2:
          bactPos[1] = height/2
        else:
          bactPos[1] = new_y

        #updates the values of the array to visualize the motion of the
bacteria
        posesX[i] = bactPos[0]
        posesY[i] = bactPos[1]

    # Create a grid for the contour plot
    x = np.linspace(-width / 2, width / 2, subdiv)
    y = np.linspace(-height / 2, height / 2, subdiv)
    meshx, meshy = np.meshgrid(x, y)
```

```
    # Calculate concentrations on the grid
    concentrations_circular = np.array([[circularDiffusion(goal, (x[i],
y[j]), timeOfSpread, diffusion_coefficient, numMolec)
                                        for j in range(subdiv)] for i in
range(subdiv)])

    # Plot the results
    plt.figure(figsize=(12, 8))
    plt.contourf(meshx, meshy, concentrations_circular, levels=100,
cmap='inferno')
    plt.colorbar(label='Concentration')

    # Plot bacteria path
    plt.plot(posesX, posesY, 'r-', label='Bacteria Path')
    plt.scatter(posesX[-1], posesY[-1], marker='*', color='g', s=200,
label='Final Position')
    plt.scatter(posesX[0], posesY[0], marker='*', color='b', s=200,
label='Start Position')
    plt.xlabel('x')
    plt.ylabel('y')
    plt.title('Circular Diffusion Concentration Profile')
    plt.legend()
    plt.show()

# Example usage
      # Model 1
BactSimulSetSpread(100, 100, 500, 50, 1, 10000, 200, [-30, -30], (40, 40),
3)

      # Model 2
BactSimulSetSpread(400, 400, 2000, 1000, 0.01, 1000000000000, 500, [-150,
-150], (150, 150), 20)

      # Model 3
```

```
BactSimulSetSpread(500, 500, 2000, 1000, 1.1, 10000000, 200, [-150, -150], (150, 150), 15)
```

Figures

- **Figure 1**: High diffusion coefficient results in direct movement.
- **Figure 2**: Moderate diffusion coefficient balances exploration and targeting.
- **Figure 3**: Low diffusion coefficient emphasizes localized exploration.