SEMESTER 2 SESSION 2023/2024

SECB4313 BIOINFORMATICS MODELLING AND SIMULATION

ASSIGNMENT 1

LECTURER : DR AZURAH BTE SAMAH

SUBMITTED BY :

NAME	MATRIC NO
ERICA DESIRAE MAURITIUS	A20EC0032
NUR HAZNIRAH BINTI HAZMAN	A20EC0114

Table of Contents

1. Description of the simulation model

1.1. Introduction

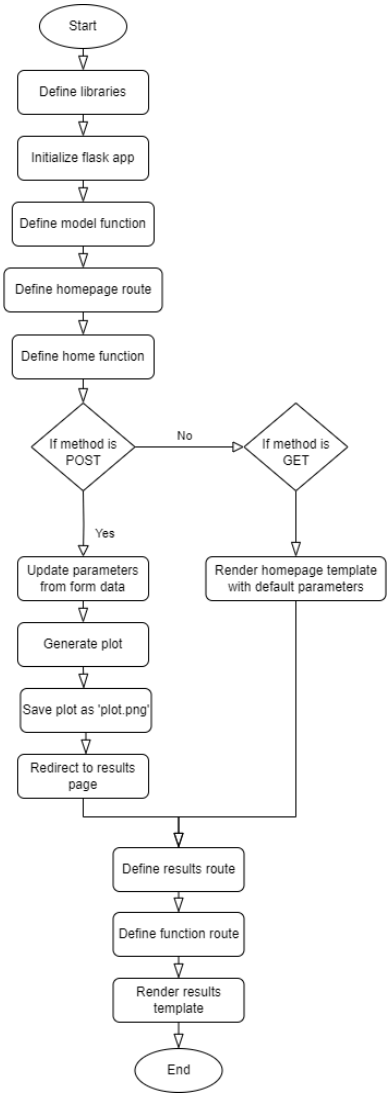
Understanding the intricate relationship between cells and cancers in the human body is paramount for effective disease treatment. Computational disease modeling provides a crucial lens to simulate infections and cancer growth, aiding researchers in unraveling the complexities of cellular dynamics and guiding therapeutic interventions. This report outlines a simulation model aimed at deciphering the dynamic interactions between immune cells and cancer cells, with a focus on elucidating the effects of therapeutic interventions and addressing drug resistance.

1.2. Objectives

1) To explore how different types of cells, like immune cells and cancer cells, interact with each other in the body.
2) To predict how these interactions change over time, helping us understand how cancer develops and progresses.
3) To understand how treatments impact cancer and immune cells, guiding treatment choices.
4) To address why treatments may become ineffective over time, offering solutions for better outcomes.
5) To develop new therapies by studying these interactions and responses to treatments.

2. Flow of the simulation model (how the codes are constructed)

2.1. Flowchart



3. List of mathematical equations used and its descriptions

eq1. dCdt = rC * C * (1 — (T/K)) * (1 — S) — dC * C

The rate of change in the concentration of cancer cells (dCdt) depends on their growth rate (rC), competition for resources with other cells (1-(T/K)), treatment effect (1-S), and cancer cell death rate (dC).

eq2. dHdt = rH * H

The rate of change in the concentration of healthy cells (dHdt) depends on their growth rate (rH).

eq3. dILdt = kIL * H

The rate of change in the concentration of interleukins (dILdt) depends on the production rate of interleukins (kIL) and the concentration of healthy cells (H).

eq4. dTdt = -kCT * C * T

The rate of change in the concentration of tumor cells (dTdt) depends on the competition for resources with other cells and the death rate of tumor cells due to treatment (kCT).

eq5. dSdt = s * T

The rate of change in the effectiveness of treatment (dSdt) depends on the concentration of tumor cells (T) and the effectiveness of the treatment (s).

4. Python libraries used

Python Libraries Used	**Purpose**
Flask	For rapid web application development.
Numpy	For array operations and numerical calculations.
matplotlib	For creating visualizations.
scipy	For solving ordinary differential equations.

5. Input of the simulation model

The simulation model integrates essential input parameters that influence the behavior of the biological system. These parameters include the growth rates of Th cells

(rH), CTL cells (rC), their death rates (dC), the rate of immune suppression (s), the carrying capacity of tumor cells (K), and the rate of IL-2 production (kIL).

6. Model parameters

This Flask code establishes a homepage route where users can adjust model parameters. Default values are provided for parameters like CTL and Th cell growth rates, IL-2 production rates, T cell activation rates, immune suppression, and tumor cell carrying capacity. If users submit the form, parameters are updated, and a plot is generated based on the new values. Users are then redirected to view the results. If no form is submitted, the homepage displays default or updated parameters for user interaction. This code handles both GET and POST requests effectively, allowing parameter adjustment and result visualization.

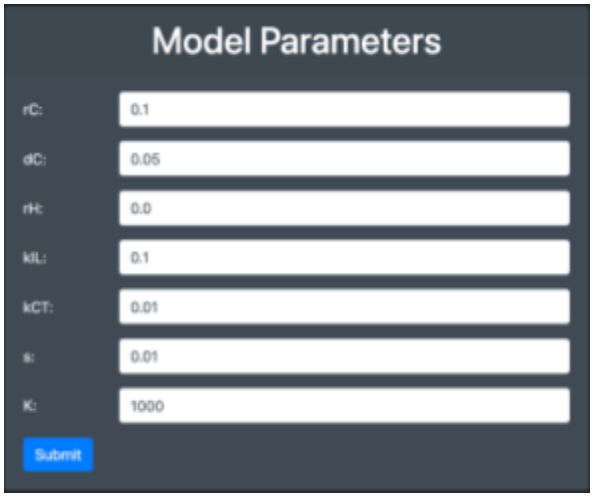


Figure 1 index.html

7. Description of simulation output and the generated graph

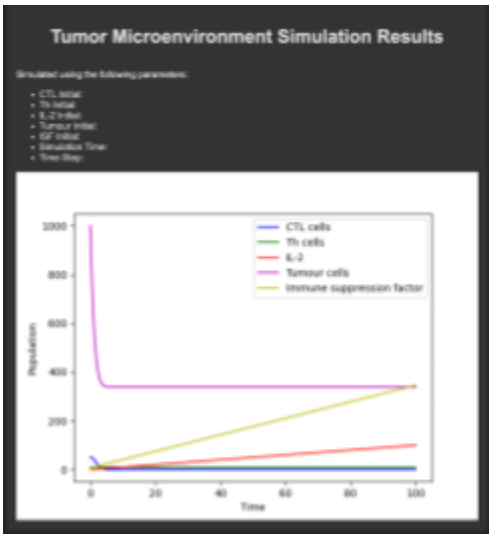


Figure 2 results.html

The line graph illustrates a notable decrease in Tumor Cell (Pink Line) population from an initial count of 1000 to around 380, followed by a relatively stable population size thereafter. Besides that, the Immune Suppression Factor (Gold Line) shows a rapid and continuous increase over time, indicating escalating immunosuppressive mechanisms within the tumor microenvironment. However, IL-2 (Red Line) displays a gradual and marginal increase over the course of time. Moreover, CTL cells indicate an initial decrease in population, followed by a consistent maintenance at zero population over time. Meanwhile, Th cells (Green Line) demonstrate a consistent zero population maintained over time.

8. Experimentation that can be carried out using the simulation model

Experimentation with the simulation model allows researchers to investigate how changes in parameters like IL-2 production rate and immune suppression level affect tumor growth and immune response. This enables evaluation of therapeutic interventions and identification of strategies to overcome drug resistance, advancing cancer biology understanding and optimizing personalized therapies.

9. Appendix

```
from flask import Flask, render_template, request, redirect, url_for, after_this_request
import numpy as np
import matplotlib
matplotlib.use('Agg')  # Use Agg backend
import matplotlib.pyplot as plt
from scipy.integrate import odeint

app = Flask(__name__)

# Define the model
def model(y, t, rC, dC, rH, kIL, kCT, s, K):
    C, H, IL, T, S = y
    dCdt = rC * C * (1 - (T/K)) * (1 - S) - dC * C
    dHdt = rH * H
    dILdt = kIL * H
    dTdt = -kCT * C * T
    dSdt = s * T
    return [dCdt, dHdt, dILdt, dTdt, dSdt]

# Define the route for the homepage
@app.route('/', methods=['GET', 'POST'])
def home():
    # Default values for the model parameters
    rC = 0.1
    dC = 0.05
    rH = 0.0
    kIL = 0.1
    kCT = 0.01
    s = 0.01
    K = 1000

    # If the form has been submitted, update the parameters
    if request.method == 'POST':
        rC = float(request.form.get('rC', 0.1))
        dC = float(request.form.get('dC', 0.05))
        rH = float(request.form.get('rH', 0.0))
        kIL = float(request.form.get('kIL', 0.1))
        kCT = float(request.form.get('kCT', 0.01))
        s = float(request.form.get('s', 0.01))
        K = float(request.form.get('K', 1000))

        # Generate the plot
        t = np.linspace(0, 100, 1000)
        y0 = [50, 10, 0, 1000, 0]
        sol = odeint(model, y0, t, args=(rC, dC, rH, kIL, kCT, s, K))

        # Plot the results
        fig, ax = plt.subplots()
        ax.plot(t, sol[:,0], 'b', label='CTL cells')
        ax.plot(t, sol[:,1], 'g', label='Th cells')
        ax.plot(t, sol[:,2], 'r', label='IL-2')
        ax.plot(t, sol[:,3], 'm', label='Tumour cells')
        ax.plot(t, sol[:,4], 'y', label='Immune suppression factor')
        ax.set_xlabel('Time')
        ax.set_ylabel('Population')
        ax.legend()
        # Check this path:
        plt.savefig('Assignment1/static/plot.png')


        # Clear the current plot to avoid overlapping plots
        plt.clf()

        # Redirect to the results page
        return redirect(url_for('results'))

    # Render the homepage template with the default parameters
    return render_template('index.html', rC=rC, dC=dC, rH=rH, kIL=kIL, kCT=kCT, s=s, K=K)

# Define the route for the results page
@app.route('/results')
def results():
    # Render the results template
    return render_template('results.html')

if __name__ == '__main__':
    app.run(debug=True)
```

Link GitHub: https://github.com/NurHaznirah/eportfolio-SECB4313/tree/main/Assignment1

https://github.com/EricaDesirae/eportfolio-SECB4313/tree/main/Assignment1