SEMESTER 20232024 – 2

FACULTY OF COMPUTING

SECB4313 - 01

BIOINFORMATICS MODELING AND SIMULATION

ASSIGNMENT 1 - Modeling a disease model in Python

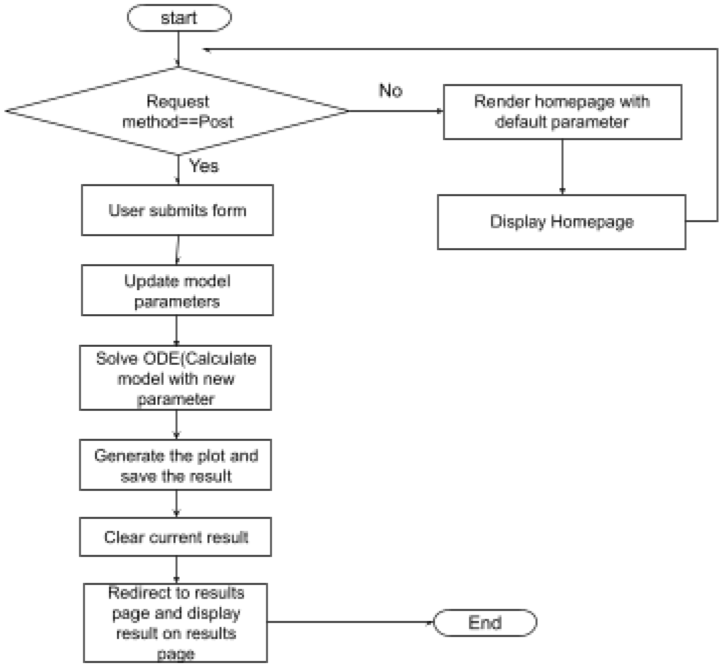
NAME	MATRIC NO
NURZARIFAH BINTI AZIZAN	A20EC0127
ROHAIZAAZIRA BINTI MOHD ZAWAWI	A20EC0138

TABLE OF CONTENTS

1.0 Description of the simulation model (Introduction and objective of model)

The simulation model is designed to simulate the dynamics of a biological system involving various cell populations and factors. The objective is to study the interaction between immune cells, tumor cells, and immune-suppressive factors, and how different parameters affect these dynamics.

2.0 Flow of the simulation model (how the codes are constructed)


start
Request method==Post
No
Render homepage with default parameter
Display Homepage
Yes
User submits form
Update model parameters
Solve ODE(Calculate model with new parameter
Generate the plot and save the result
Clear current result
Redirect to results page and display result on results page
End

3.0 List of mathematical equations used and its descriptions

$$\frac{dC}{dt} = rC.C.(1 - \frac{T}{K}).(1\text{-}S)\text{-}dC.C$$

- This equation represents the rate of change of CTL cells (*C*) over time. It takes into account the following factors:
    - *rC*: Rate of increase of CTL cells.
    - *dC*: Rate of decrease of CTL cells
    - *T*: Tumor cells.
    - *K*: Carrying capacity of tumor cells.
    - *S*: Immune suppression factor.

$$\frac{dt}{dH} = rH \cdot H$$

- This equation represents the rate of change of Th cells (*H*) over time. It is a simple exponential growth equation where *rH* represents the rate of increase of Th cells.

$$\frac{dIL}{dt} = kIL.H$$

- This equation represents the rate of change of IL-2 (*IL*) over time. It describes the production of IL-2 by Th cells (*H*) with a rate constant *kIL*

$$\frac{dT}{dt} = kCT.C.T$$

- This equation represents the rate of change of tumor cells (*T*) over time. It describes the killing of tumor cells by CTL cells (with a rate constant *kCT*. The negative sign indicates a decrease in tumor cell population due to CTL cell activity.

$$\frac{dS}{dt} = s.T$$

- This equation represents the rate of change of the immune suppression factor (*S*) over time. It describes the increase in the immune suppression factor due to the presence of tumor cells (*T*) with a rate constant *s*

4.0 Python libraries used

- Flask: For creating the web application.
- NumPy: For numerical computations.
- Matplotlib: For plotting.
- SciPy: Specifically, odeint from scipy.integrate for solving ODEs.

5.0 Input of the simulation model

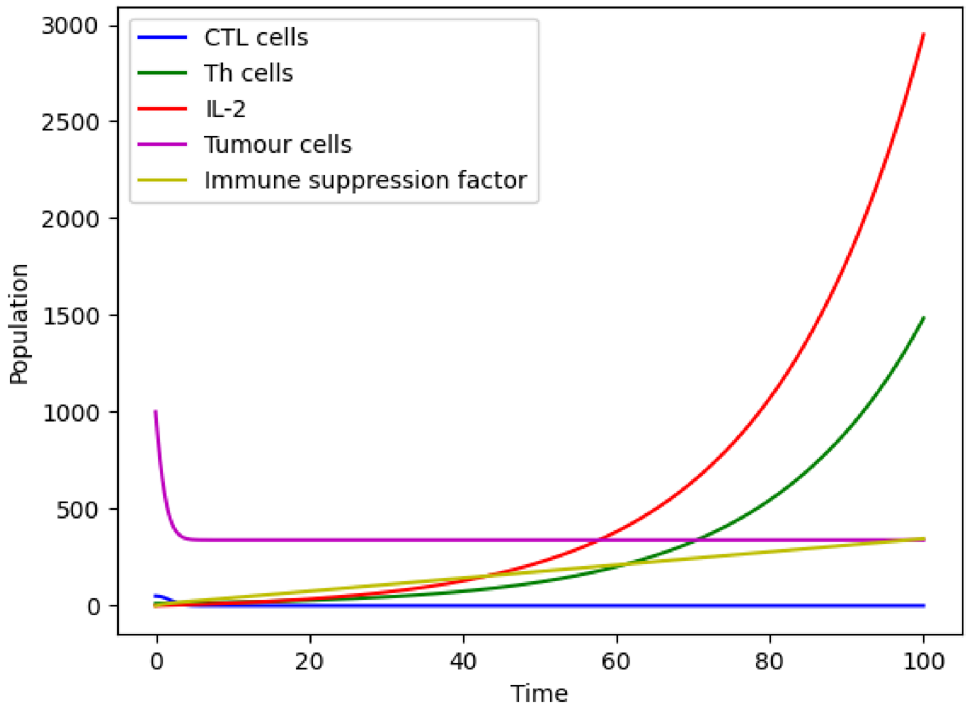
The input for these simulation models is initial condition like starting state of the system, such as the initial populations of different cell types (e.g., CTL cells, Th cells) and environmental conditions (e.g., nutrient levels). Secondly, model parameters dictate the behavior of the system, including rates of cell proliferation, death, and interaction between cell types. For example, parameters like the reproduction rate of CTL cells or the rate of cytokine production by Th cells influence the dynamics of the immune response. Thirdly, experimental conditions define the specific setup under which the simulation is conducted, such as the dosage and timing of drug treatments or the presence of environmental stimuli.Fourthly, simulation settings determine technical aspects like the duration of the simulation, time steps used for numerical integration, and convergence criteria for solving differential equations. Finally, external inputs, such as experimental data or clinical measurements, can be incorporated to enhance the realism and relevance of the simulation.

6.0 Model parameters

These parameters are important because they determine how the biological system behaves. Users can change these parameters to see how different situations affect the populations of cells over time. In the Flask app, users can modify these parameters on the homepage and see the new results on the results page.

Model Parameters	Description	Parameter Values
rC	Rate of increase of CTL cells (Cytotoxic T Lymphocytes)	0.1
dC	Rate of decrease of CTL cells.	0.05
rH	Rate of increase of Th cells (Helper T cells)	0.05
kIL	Rate constant for production of IL-2 (Interleukin-2) by Th cells.	0.1
kCT	Rate constant for killing of tumor cells by CTL cells.	0.01
s	Rate of increase of immune suppression factor.	0.01
K	Carrying capacity of tumor cells.	1000

7.0 Description of simulation output and the generated graph



The initial graph shows how the population of CTL cells (Cytotoxic T Lymphocytes) changes over time, influenced by factors like reproduction rate (rC) and interactions with other cells. Results indicate that CTL cell population starts high but declines slightly, stabilizing after about 20 units of time.

The second graph examines the dynamics of Th cells (Helper T Cells) over time, emphasizing growth patterns and responses to stimuli. Results show a gradual increase in Th cell population over 100 units of time.

The third graph tracks IL-2 (Interleukin-2) levels produced by Th cells, revealing fluctuations influenced by production rate (kIL) and environmental factors. IL-2 levels appear to fluctuate slightly before stabilizing at a low level.

The fourth graph illustrates tumor cell population dynamics, reflecting interactions with immune cells and environmental conditions. Results show an initial sharp decline, followed by a period of low levels, then significant increase after about 80 units of time.

Lastly, the fifth graph portrays the concentration of an immune suppression factor over time, influenced by tumor cell presence and rate of increase (s). Results indicate low levels until approximately 60 units of time, followed by a sharp increase.

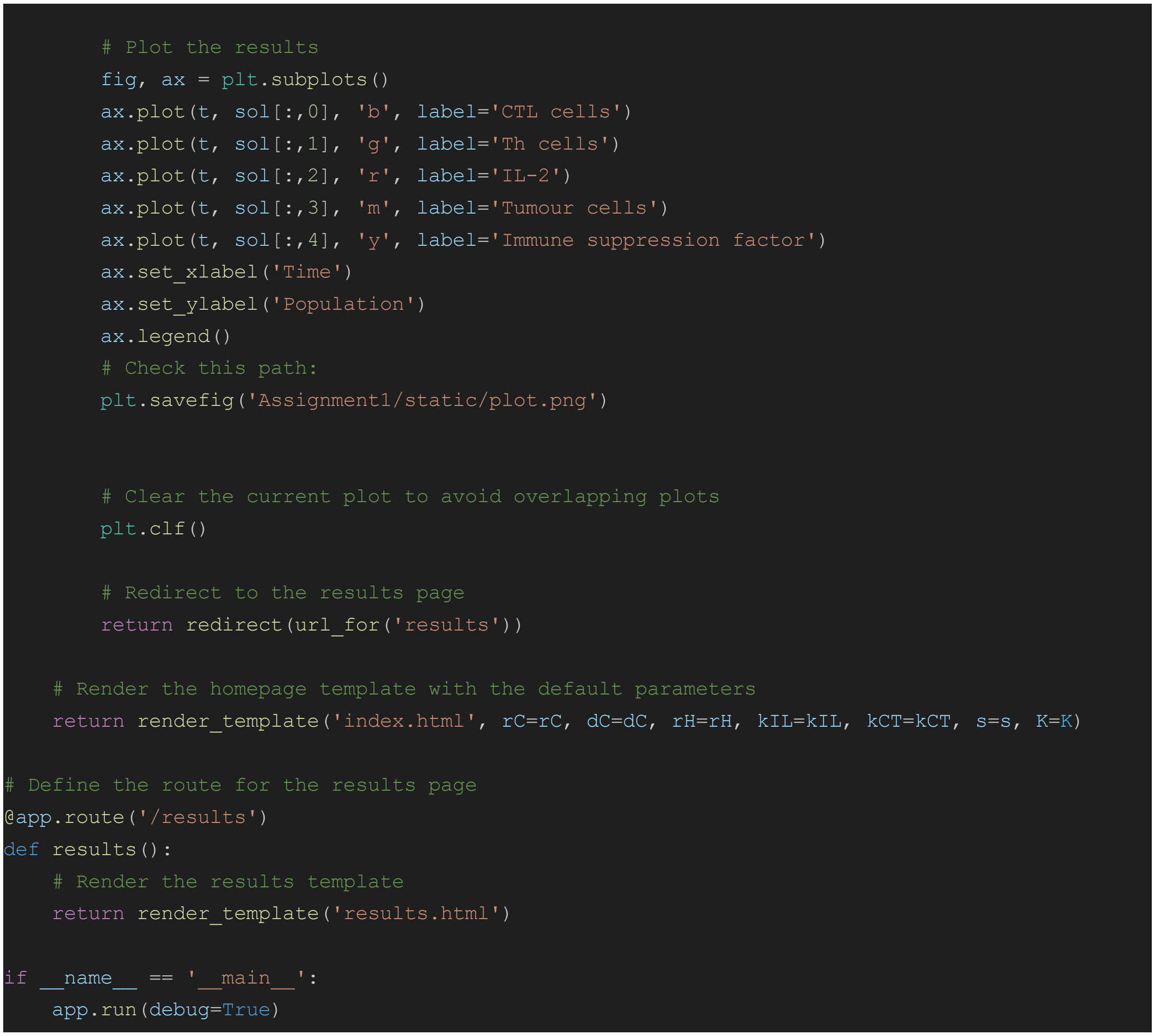
Collectively, these graphs provide insights into how parameters and interactions between cell types shape the behavior of the biological system. This understanding allows for evaluation of intervention effectiveness and informed decision-making for system management.

8.0 Experimentation that can be carried out using the simulation model

Using the simulation model, several experiments and analyses can be conducted to gain insights into the behavior and dynamics of the biological system being modeled.First example of experiments is drug dose optimization by Optimize drug dosing regimens by simulating different dosage levels and schedules. Assess how varying drug concentrations and administration frequencies impact tumor suppression and immune cell activation.Secondly, use in disease progression studies like model disease progression scenarios by altering initial conditions or introducing mutations in the system. Explore how different stages of disease development affect the dynamics of immune cell populations and tumor growth. Furthermore, the simulation model can be utilized in predicting clinical outcomes. Researchers can use the model to forecast clinical outcomes and treatment responses in real-world scenarios. By validating model predictions against clinical data and refining model parameters, researchers can enhance the accuracy of their predictions.

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)
```