FACULTY OF COMPUTING

UNIVERSITI TEKNOLOGI MALAYSIA

BIOINFORMATICS MODELING AND SIMULATION - SECB4313(01)

Assignment 1

TITLE : Modeling a Disease Model in Python

PREPARED BY : GROUP

NAME	MATRIC NO
MOHD FIRDAUS BIN ZAMRI	A20EC0080
LUE GUO MING	A20EC0073

PREPARED FOR:

DR AZURAH BTE A SAMAH

Table of Contents

Introduction

Disease modeling is essential for understanding cellular dynamics and interactions within cells and tumors. By simulating infection processes and tumor growth, researchers gain insights into treatment strategies and biological mechanisms. These models help explore immune-cancer cell interactions, treatment effects, and drug resistance. Researchers use various quantitative methods, from simple equations to agent-based models, calibrated with experimental data to validate hypotheses and enhance their understanding of biology.

Flow of the Simulation Model

The simulation model follows a structured flow, beginning with parameter initialization, mathematical model definition, integration of differential equations, and visualization of outcomes.

Parameter initialization involves setting essential parameters such as cancer cell growth rate (rC), healthy cell growth rate (rH), interleukin growth rate (kIL), treatment efficacy, and cell death rates. These parameters provide the foundation for the subsequent simulation process.

Once parameters are initialized, the mathematical model is defined. It represents cancer progression and treatment response dynamics using a system of ordinary differential equations (ODEs). These equations encapsulate biological processes such as cell growth, immune response, and treatment effects.

Next, the integration of differential equations begins. Using numerical methods provided by the SciPy library, particularly the ‘odeint’ function, the system of ODEs is solved over a defined time interval. This computational process generates temporal trajectories for cancer cells, healthy cells, interleukins, tumor cells, and treatment effects.

After the integration is complete, the simulation results are visualized to facilitate interpretation and analysis. Matplotlib, a powerful plotting library, is employed to create visual representations of temporal changes in cell populations and treatment effects. Line plots are utilized to illustrate trends and patterns, enabling researchers to discern insights into disease progression and treatment outcomes.

Finally, researchers have the flexibility to interactively adjust model parameters. This enables sensitivity analysis and exploration of diverse scenarios. Modifications to parameters such as cell growth rates, treatment efficacy, or immune response allow investigation of their influence on disease dynamics and treatment effectiveness.

Mathematical Equations

The disease modeling framework employs a set of mathematical equations to depict the progression of cancer and the interaction dynamics within a biological system. The model

comprises five variables: C, H, IL, T, and S, each representing different components of the disease and its treatment.

Equation 1: dCdt = rC * C * (1 — (T/K)) * (1 — S) — dC * C

Equation 1 describes the rate of change of cancer cell concentration over time (dCdt). This change is determined by multiple factors:

- The growth rate of cancer cells (rC).
- Competition for resources with other cells, represented by 1-(T/K), where T/K is the ratio of tumor cells to the carrying capacity of the environment.
- The impact of treatment, denoted by 1 - S.
- The death rate of cancer cells (dC).

Equation 2: dHdt = rH * H

Equation 2 represents the rate of change of healthy cell concentration over time (dHdt). This change is solely determined by the growth rate of the healthy cells (rH).

Equation 3: dILdt = kIL * H

Equation 3 describes the rate of interleukin concentration over time (dILdt). This change is influenced by the production rate of interleukins (kIL) and the concentration of healthy cells.

Equation 4: dTdt = -kCT * C * T

Equation 4 represents the rate of change of tumor cell concentration over time (dTdt). This change is determined by multiple factors:

- Competition for resources with other cells, denoted by -kCT * C * T.
- The death rate of tumor cells due to treatment (kCT).
- The concentration of tumor cells themselves (T).

Equation 5: dSdt = s * T

Equation 5 describes the rate of change of treatment effect over time (dSdt). This change is influenced by two main factors:

- The concentration of tumor cells (T), which represents the target of the treatment.
- The effectiveness of the treatment (s), which determines how efficiently it impacts the tumor cells.

Python Libraries

The implementation of the disease modeling project relies on several Python libraries that provide essential functionalities for mathematical computation, data visualization, and integration of differential equation solvers. These libraries are widely used in scientific computing and enable the simulation and analysis of complex systems. Below are the key Python libraries utilized in the project:

a. NumPy: NumPy is a foundational package for numerical computing in Python. It provides support for large arrays and matrices, along with a collection of mathematical functions to perform operations on these arrays efficiently. NumPy is extensively used for array manipulation, mathematical operations, and numerical computations in the disease modeling project.

b. Matplotlib: Matplotlib is a versatile library for creating static, interactive, and animated visualizations in Python. It offers a wide range of plotting functions to generate various types of graphs, including line plots, scatter plots, histograms, and heatmaps. Matplotlib is utilized for visualizing the simulation results, such as the dynamics of cell populations over time and the effects of different treatment scenarios.
c. SciPy: SciPy is a comprehensive library that extends NumPy and provides additional functionalities for scientific computing. It includes modules for optimization, integration, interpolation, linear algebra, and differential equation solving. The 'odeint' function from the 'scipy.integrate' module is used in disease modeling project to solve the system of ordinary differential equations governing the dynamics of cell populations.

Input of the Simulation Model

The simulation model requires specific input parameters to initialize the system and define the conditions under which the simulation will be conducted. These inputs include:

1. Time Array (t): The time array (t) represents the time points at which the system's state variables will be evaluated during the simulation. It is generated using NumPy's linspace function to create a linearly spaced array of time points from 0 to 100, inclusive, with 1000 data points evenly distributed along the interval.
2. Initial State (y0): The initial state (y0) defines the initial concentrations of the system's state variables at the beginning of the simulation. It is specified as an array containing the initial concentrations of cancer cells, healthy cells, interleukins, tumor cells, and treatment effect, respectively. In this case, the default initial state is set to [50, 10, 0, 1000, 0].

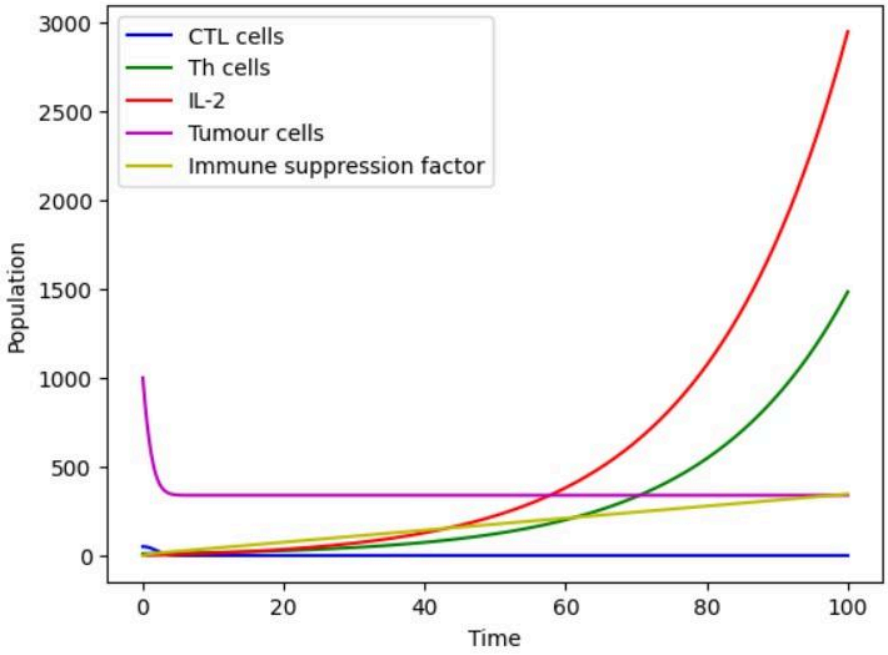
Model Parameters

The model parameters are intrinsic properties of the mathematical model that govern the behavior and interactions of the biological entities within the system. These parameters are predefined with default values and include:

1. Growth Rate of Cancer Cells (rC): The growth rate of cancer cells (rC) determines the rate at which cancer cells proliferate in the absence of external factors. The default value is set to 0.1.
2. Death Rate of Cancer Cells (dC): The death rate of cancer cells (dC) represents the rate at which cancer cells die. It influences the decline in cancer cell population over time. The default value is set to 0.05.
3. Growth Rate of Healthy Cells (rH): The growth rate of healthy cells (rH) governs the rate at which healthy cells proliferate. It contributes to the maintenance and regeneration of healthy tissue. The default value is set to 0.05.

4. Interleukin Production Rate (kIL): The interleukin production rate (kIL) determines the rate at which interleukins, proteins involved in the immune response, are produced. It influences the modulation of immune cell activity. The default value is set to 0.1.
5. Death Rate of Tumor Cells due to Treatment (kCT): The death rate of tumor cells due to treatment (kCT) represents the rate at which tumor cells are eliminated in response to treatment. It reflects the effectiveness of the treatment in targeting tumor cells. The default value is set to 0.01.
6. Effectiveness of Treatment (s): The effectiveness of treatment (s) determines the degree to which treatment inhibits cancer cell growth. It represents the efficacy of the treatment regimen in suppressing tumor progression. The default value is set to 0.01.
7. Carrying Capacity (K): The carrying capacity (K) defines the maximum population size that the environment can support. It influences the competition for resources among cell populations and affects tumor growth dynamics. The default value is set to 1000.

Simulation Output and Graph



Upon execution, the simulation model generates output data that provides insights into the temporal dynamics of cell populations over the simulated time interval. The primary output consists of numerical data representing the concentrations of cancer cells, healthy cells, interleukins, tumor cells, and treatment effect at discrete time points throughout the simulation duration.

From the figure above, Matplotlib is utilized to create visual representations of the simulation results, focusing on the temporal evolution of cell populations. A line plot is generated to depict the concentrations of cancer cells, healthy cells, interleukins, and tumor cells

as functions of time. The x-axis represents time, while the y-axis denotes the concentrations of the respective cell populations.

With the default parameter values provided (t=np.linspace(0,100,1000) and *y*0=[50,10,0,1000,0]), the generated graph would illustrate the temporal dynamics of cell populations. The plot showcases the changes in the concentrations of cancer cells, healthy cells, interleukins, and tumor cells over the simulated time interval. Each population's trajectory reflects its growth, decline, or stabilization over time, influenced by factors such as cell proliferation, immune response, and treatment efficacy.

Experimentation

The simulation model offers a versatile platform for experimentation, allowing researchers to explore various scenarios, assess parameter impacts, and evaluate treatment strategies:

1. **Sensitivity Analysis**: Researchers systematically vary model parameters to understand their influence on simulation outcomes. By adjusting parameters such as growth rates, treatment efficacy, or immune response strength, researchers analyze disease progression and treatment response dynamics. Identifying influential parameters helps prioritize further study and intervention design.
2. **Treatment Optimization**: The model facilitates the evaluation and optimization of different treatment regimens. Researchers adjust parameters related to treatment efficacy, dosage, or scheduling to find optimal intervention combinations. Comparative analysis identifies the most effective approach to inhibit tumor growth while minimizing adverse effects on healthy tissues.
3. **Drug Screening and Development**: The simulation model serves as a virtual screening tool for testing potential therapeutic agents and drug combinations. Researchers simulate candidate drug effects on cancer cell proliferation, tumor growth inhibition, and immune response modulation. Promising candidates are prioritized for further experimental validation and clinical trials based on simulated interactions with biological pathways.
4. **Prediction of Treatment Outcomes**: Researchers predict treatment outcomes under various conditions and scenarios. Different treatment protocols and disease states are simulated to anticipate long-term effects on tumor progression, recurrence risk, and patient survival. Personalized treatment planning considers individual patient characteristics, tumor biology, and treatment response profiles.

Overall, the simulation model provides a powerful tool for conducting experiments to enhance understanding of disease dynamics, optimize treatment strategies, and guide decision-making in cancer research and clinical practice. Through systematic experimentation and analysis, researchers can advance knowledge in the field of oncology and contribute to the development of more effective and personalized cancer therapies.

Appendix

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

# 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]

# Default values for the model parameters
rC = 0.1
dC = 0.05
rH = 0.05
kIL = 0.1
kCT = 0.01
s = 0.01
K = 1000

# Define time array and initial conditions
t = np.linspace(0, 100, 1000)
y0 = [50, 10, 0, 1000, 0]

# Solve ODE
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()
plt.show()
```

Python Source Code

(https://colab.research.google.com/drive/1mqyhi6FRXgjL6w_zSavZ7czC8-oPa_7L?usp=sharing)