Matteo Mansi 890976

Corso di Laurea in Ingegneria Fisica

Anno Accademico 2022/2023

Per iniziare vorrei scusarmi perchè scrivendo questo report mi sono reso conto che probabilmente non le avevo avvisato che avevo cambiato il tema del mio progetto: quando ho saputo che l’opportunità con Carlos ad Altos Labs si stava concretizzando, ho deciso di fare qualcosa che avesse a che fare con la biologia. Poi ho pensato al cancro, che è peculiare per le sue caratteristiche di evadere il processo apoptotico e di potersi replicare indefinitamente, e in più è una malattia di cui sapevo veramente poco. Vista la complessità dei sistemi biologici, restava da capire il livello a cui simulare, se molecolare, cellulare o dell’organismo; alla fine ho optato per il livello cellulare integrato con due specie chimiche, la citochina IL-2 e l’ossigeno.

A questo punto ho cercato dei paper che potessero aiutarmi per organizzare il sistema di simulazione, basato su concetti di biologia abbastanza tecnici.

Allego il link dei due paper da cui ho tratto molte delle equazioni e delle stime iniziali dei parametri che ho utilizzato:

**On-lattice agent-based simulation of populations of cells within the open-source Chaste framework:**

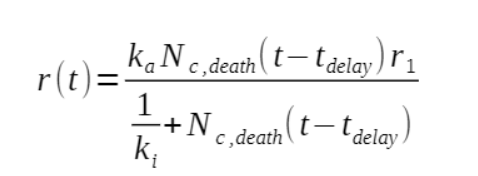
<https://royalsocietypublishing.org/doi/full/10.1098/rsfs.2012.0081>

**A computational multiscale agent-based model for simulating spatio-temporal tumor immune response to PD1 and PDL1 inhibition:**

<https://royalsocietypublishing.org/doi/full/10.1098/rsif.2017.0320#RSIF20170320F2>

Ho scelto una modellazione ad agenti usando la libreria agentpy, dove lo spazio è una griglia quadrata in cui ogni cella può contenere al massimo un agente alla volta.

Il prodotto finale della simulazione è un’animazione dello sviluppo del cancro a partire da una singola cellula proliferativa inizializzata nel centro della griglia. La cellula può replicarsi, spostarsi, o entrare in uno stato quiescente se non c’è abbastanza ossigeno. Dopo un certo tempo predefinito, se non viene risvegliata da un’adeguata quantità di ossigeno, anche la cellula quiescente morirà e sarà rimossa dalla griglia. Il numero di cellule tumorali morte ad ogni iterazione è salvato perchè determina il numero di cellule T che verranno reclutate, secondo l’equazione:



N è il numero di cellule tumorali morte, k\_a e k\_i, rispettivamente mutational burden e neoantigen\_strength del paziente, sono parametri geneticamente determinati che influenzano la rapidità di riconoscimento da parte del sistema immunitario delle cellule tumorali.

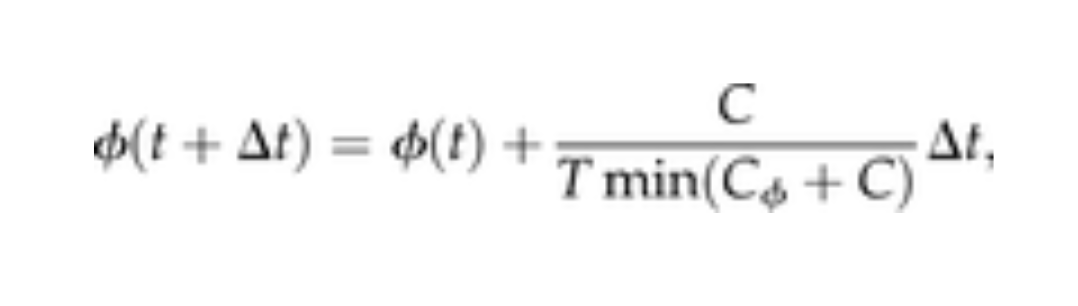
Anche r serve a determinare la velocità di reclutamento.

T\_delay indica il tempo di reazione del sistema immunitario, che deve far arrivare le cellule dendritiche ai linfonodi ed effettuare il priming delle cellule T, che entrano nel micro ambiente tramite i vasi sanguigni, estratti aleatoriamente sulla griglia. Le cellule T arrivano in stato effettore, e passano allo stato attivo quando individuano una cellula tumorale nel loro intorno di Moore. Nello stato attivo possono identificare una cellula cancerogena ed attaccarla.

Le cellule cancerogene sono di due tipi, a seconda della presenza o meno della proteina PDL1.

Questa proteina nelle cellule normali previene l’uccisione del materiale *self*, e quindi dell’insorgere di malattie autoimmuni, inibendo l’azione del sistema immunitario. Questa proteina ha una probabilità di manifestarsi ogni volta che la cellula tumorale è sollecitata dall'attacco di una cellula T. Quando è attivata, la probabilità di essere uccisi da una cellula T è inferiore. Ad ogni replicazione della cellula tumorale la presenza o meno della proteina si conserva.

Ogni cellula è contrassegnata da un attributo che è la fase, , che viene aumentata ad ogni iterazione secondo questa equazione



Dove Tmin è il periodo minimo del ciclo cellulare, e è la concentrazione di ossigeno alla quale la velocità metabolica è la metà di quella massima. C è appunto la concentrazione di ossigeno della posizione nella griglia della cellula; questa equazione sostanzialmente indica che più una cellula consuma ossigeno, più “invecchia” rapidamente. Quando la fase è maggiore di 1, si ha la replicazione, azzerando la fase per le due cellule figlie.

La citochina IL-2 è prodotta dalle cellule T appena attivate per un periodo di tempo predeterminato, e una sufficiente concentrazione di questa specie chimica è condizione necessaria per la replicazione delle cellule T di qualsiasi tipo.

Sia la diffusione di 02, che di IL-2, sono ottenute risolvendo l’equazione differenziale alle derivate parziali:

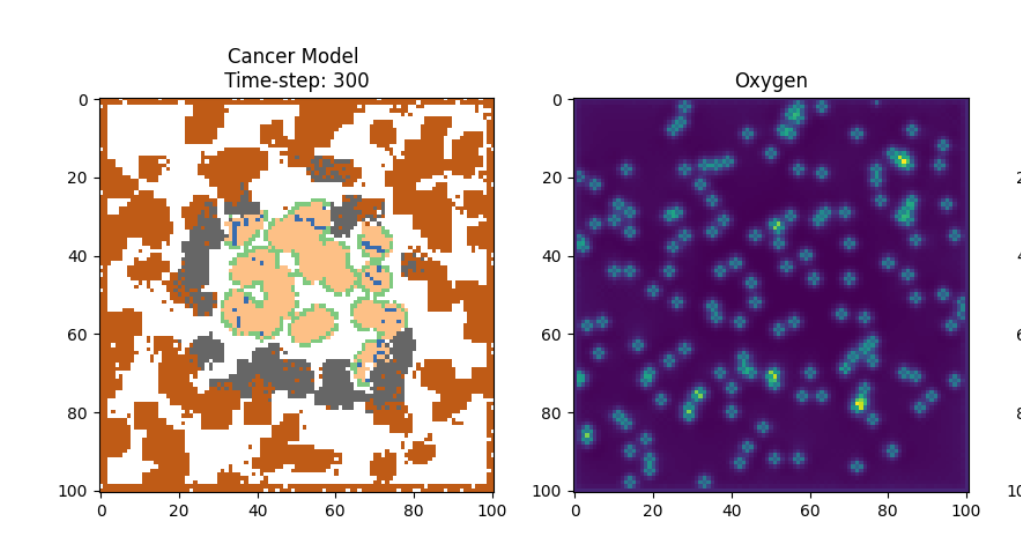
Laplaciano =

Dove C è la concentrazione, D è il coefficiente di diffusione, d è il rate di degradazione (nel caso dell’ossigeno è di consumo e viene applicato nelle posizioni della griglia dove è presente una cellula) e lambda è il rate di produzione, per l’ossigeno viene dai vasi sanguigni, e per IL-2 dalle cellule T appena attivate.

Ho risolto questa equazione discretizzandola e risolvendo per ogni time-step sincronizzato con il resto della simulazione.

**Risultati**

Ho cercato di aggiustare i parametri manualmente per ottenere una “battaglia” prolungata tra tumore e sistema immunitario, ma quello che succede sempre nei miei tentativi è che cluster di cellule si formano attorno ai punti di collegamento con i vasi sanguigni (entry\_points nel codice) e schermano con il loro consumo l’ossigeno che fuoriesce da queste sorgenti, arrivando nel giro di relativamente poche iterazioni ad uno stato di equilibrio. Non ho avuto il tempo per sbloccare questa situazione e trovare un risultato un po’ più interessante, anche perchè penso richiederebbe una conoscenza più approfondita di come si comportano le equazioni parziali differenziali ed inoltre i tempi di simulazione sono abbastanza alti.



Questo è il risultato di una simulazione con i parametri settati come nel file consegnato su Moodle, ed è facile vedere come le cellule si accalchino attorno alle fonti di ossigeno.

In rosa pallido sono le proliferative cell PDL1 negative, in blu quelle PDL1 positive, in verde quelle in stato quiescente, in grigio le cellule T attivate, e in arancio le cellule T non attivate

**Istruzioni per il Codice**

Servono le seguenti librerie:

import numpy as np

import matplotlib.pyplot as plt

import matplotlib as mpl

import agentpy as ap

import random

import IPython

from math import sqrt,exp

IPython l’ho usato per l’animazione in jupyter.

Utilizzando la funzione agentpy.animate, il codice plotterà 3 animazioni in parallelo, quella con le cellule, quelle della diffusione dell’ossigeno e quella della diffusione di IL-2.

I parametri sono tutti in un dizionario commentato, e si modificano manualmente.

Per tempi di simulazione più brevi consiglio di diminuire il parametro Tmin.

Il numero di iterazioni è dato dal parametro step.