DATASCI2G03 Final Project: Modifying a model of Trypanosome Infections to account for Co-infection of the Malaria Parasite

Alexander Turco, TA: Hector Robin

December 7, 2022

Student Number: 400235250

Contents

1 A Brief Background

Trypanosomes are single celled eukaryotic parasites which maintain a chronic parasitaemia (parasites present in the blood) in a variety of mammalian hosts (Tyler et al., 2001). The way in which these Trypanosomes are able to infect mammalian hosts stems from a balance between Trypanosome proliferation, differentiation and cell death (Tyler et al., 2001). In a study conducted by Tyler et al. (2001), researchers utilized data pertaining to the number of parasite cells/ml of blood in mice to develop mathematical models which could help to better understand Trypanosome infections.

The way in which the researchers modelled these complex interactions of cell proliferation, differentiation, and cell death was based upon two types of cells that were measured in blood samples obtained from the mice. The first cell type is called 'slender' and these multiply exponentially in the host. The second cell type is 'stumpy' cells and these no longer divide. Slender cells can differentiate into stumpy cells and this is initiated when the cell concentration of these slender cells becomes high (Tyler et al., 2001). This differentiation of slender cells to stumpy cells will stop the exponential growth of slender cells and prevent these parasites from killing the host in a quick manner.

To model the slender-to-stumpy cell transition, (Tyler et al., 2001) utilized differential equations to describe the rate at which the total parasite changes with respect to time. This project will utilize `C++` to employ a fourth-order Runge-Kutta method that will allow us to find the solution to this set of differential equations. We will compare results to the findings of (Tyler et al., 2001) to confirm that our model is working as it should, and we will also expand the model to account for co-infection.

2 Methods

The concentration of slender and stumpy cells was first proposed by (Turner et al., 1995) using two differential equations, one to model replication of slender cells at density X (1) and one to model the differentiation of slender to stumpy cells at density Y (2).

$$\frac{dX}{dT} = (r - f)X(t) \tag{1}$$

$$\frac{dY}{dT} = fX(t) - mY(t) \tag{2}$$

X is the concentration of slender cells at time t
Y is the concentration of stumpy cells at time t
r is the rate of division of slender cells
f is the rate of differentiation of slender-to-stumpy cells
m is the mortality rate of stumpy cells

In order to achieve a solution to the differential equations, (Tyler et al., 2001) modified the model by introducing density dependent trypanosome differentiation. The assumption is made that $f = f_0(X + Y)$ which means that the differentiation rate is proportional to the total population/concentration of both cell types. This can be inserted into the equations above to get two slightly different equations (3) and (4) which are shown below.

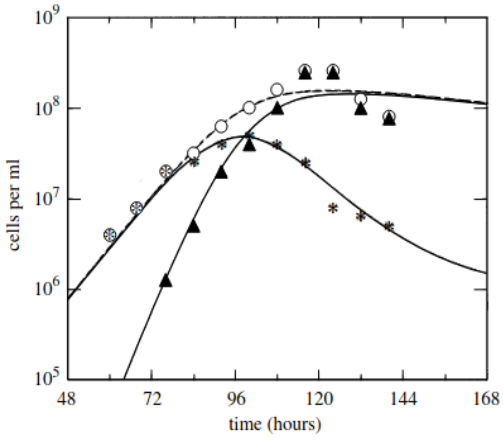
$$\frac{dX}{dT} = (r - f_0(X + Y))X \tag{3}$$

$$\frac{dY}{dT} = f_0(X + Y)X - mY \tag{4}$$

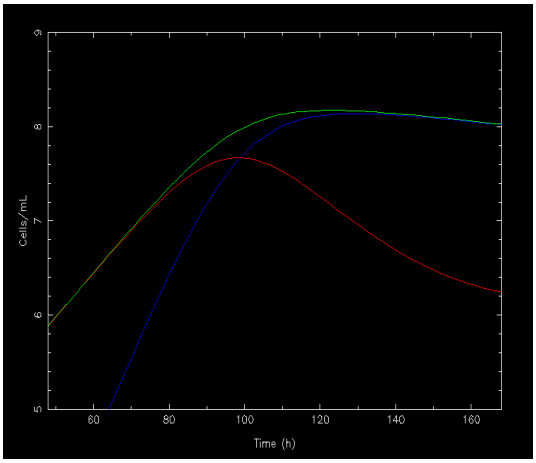
Now, these equations are able to be solved by utilizing a fourth-order Runge-Kutta method, in the same way as (Tyler et al., 2001). The parameters r and m have already been mentioned above, but f_0 and X_0 are two parameters that are used to set initial conditions and create a solution using a fourth-order Runge-Kutta method. This method was implemented in `C++` and in order to see if the system behaved correctly, plots were used to visualize the populations of the two different cell types, as well as the total population of both cell types (X+Y). This study will compare results to those of (Tyler et al., 2001), and build upon the model by accounting for co-infection.

3 Testing the Program

To test if the initial `C++` program functioned correctly, the same parameter values used in the study performed by (Tyler et al., 2001) were used to produce a plot. Below are the results of the comparison.

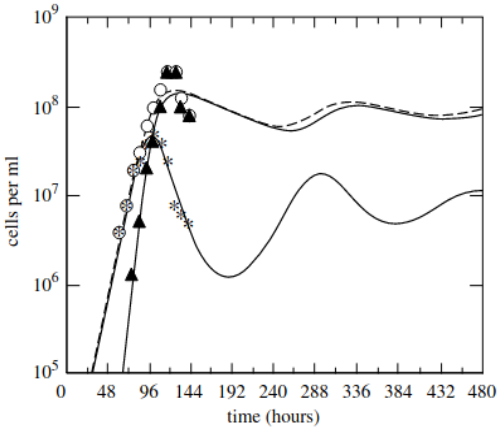


(a) Figure taken from (Tyler et al., 2001)

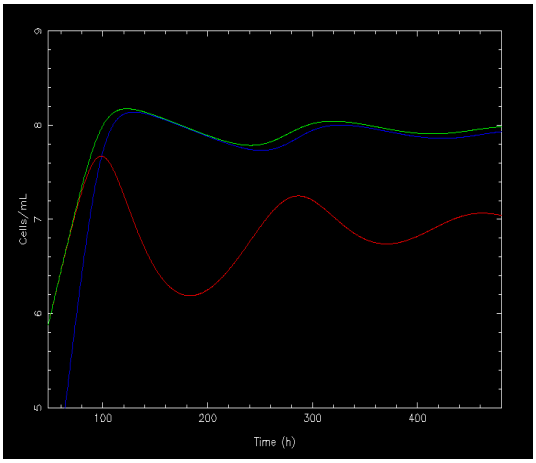


(b) Figure made with `pgplot` by me

Figure 1: Fitting differential equations (3) and (4) to infected mice with parameters $r = 0.11$, $f_0 = 1.2$ x 10^-9, $m = 0.012$ and $X_0 = 3.91$ x 10^3. This figure shows a comparison between the different cell populations over a time period of 168 hours found by (Tyler et al., 2001) (a) and found by me (b). Both solutions were found using a fourth-order Runge-Kutta method. The green line represents the total trypanosome concentration, the red line represents the concentration of slender cells and the blue line represents the concentration of stumpy cells. In Figure (a) the circles, stars, and triangles are representative of data the researchers obtained from the mice, pertaining to the total trypanosome concentration, slender cell concentration, and stumpy cell concentration respectively. These findings suggest that first of all this is an accurate model and second of all the `C++` program works.



(a) Figure taken from (Tyler et al., 2001)



(b) Figure made with `pgplot` by me

Figure 2: Fitting differential equations (3) and (4) to infected mice with parameters $r = 0.11$, $f_0 = 1.2$ x 10^-9, $m = 0.012$ and $X_0 = 3.91$ x 10^3. This figure shows a comparison between the different cell populations over a time period of 480 hours found by (Tyler et al., 2001) (a) and found by me (b). These plots are essentially continuations of the theoretical curves from Figure 1 and the oscillations in trypanosome concentration in all three lines provides good evidence of density dependent differentiation as we can see slender cells differentiate to stumpy cells when the concentrations of slender cells reach a high concentration. The matching of results also further confirms the `C++` program I wrote is running correctly.

4 Results and Extensions

An excellent way to modify/extend this model is to consider the effects of the immune repsonse, which is something that Tyler et al. (2001) have already done. In section (4.1). To try something new, I want to modify the model proposed by Tyler et al. (2001) by accounting for co-infection. In a given host, pathogens are rarely found in isolation, meaning they interact with each other and potentially affect the composition of other bacteria (Venter et al., 2022). A study performed by Sanches-Vaz et al. (2019) reported that *T. brucei* infection can actually protect mice from *P. berghei* infection (malaria).

4.1 Co-infection Model

Upon modifying this system, we now have 3 equations.

$$\frac{dX}{dT} = (r - f_0(X + Y))X \tag{5}$$

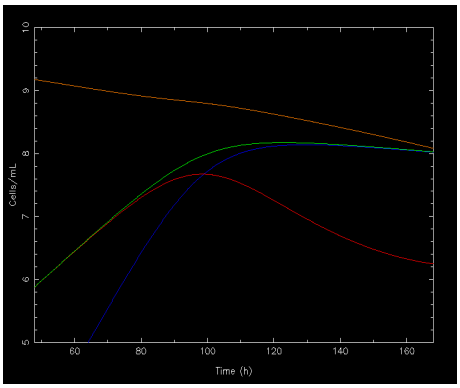
$$\frac{dY}{dT} = f_0(X + Y)X - mY \tag{6}$$

$$\frac{dC}{dT} = (r_2 - f_0(X + Y))X - mY - m_2C \tag{7}$$

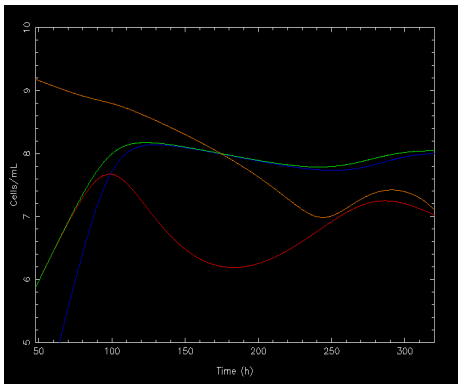
C is the concentration of malaria causing cells at time t
r_2 is the rate of division of malaria causing cells
m_2 is the mortality rate of malaria cells

To not over-complicate the model the assumption is once again made that $f = f_0(X + Y)$. In this modified model, I developed a third equation (7) to model the population/concentration of *Plasmodium berghei* which causes malaria. I modelled this equation based off the findings from Sanches-Vaz et al. (2019) which suggest that a Trypanosome infection can provide protection against malaria infections. New parameters I introduced are r_2 and m_2 and C.

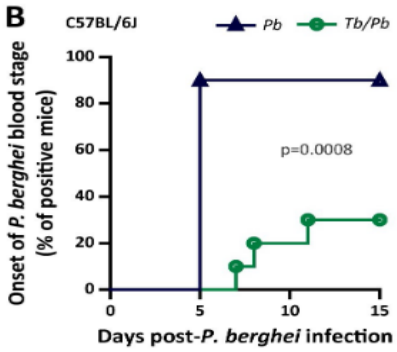
4.2 Results Using Parameters $r = 0.11$, $r_2 = 0.20$, $m = 0.012$, $m_2 = 0.02$, $f_0 = 1.2e-9$, $X_0 = 3910.00$, $Y_0 = 1e-10$, $C_0 = 3.91e9$



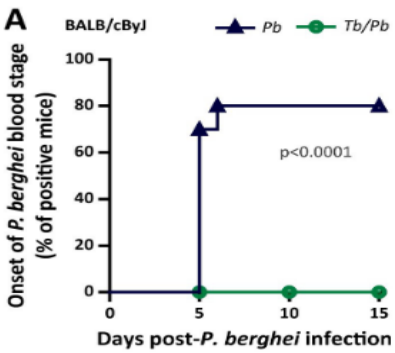
(a) Figure made with `pgplot` by me (168h)



(b) Figure made with `pgplot` by me (300h)



(c) Figure taken from (Sanches-Vaz et al., 2019)



(d) Figure taken from (Sanches-Vaz et al., 2019)

Figure 3: This figure reveals similarities in the findings of my expanded model, and real data collected from mice by Sanches-Vaz et al. (2019). Figures (c) and (d) show the percentage of mice displaying *Plasmodium berghei* parasitaemia, measured by flow cytometry. The black line represents those who were innoculated with *P. berghei* and the green line represents those innoculated with *P. berghei* after being infected with *T. brucei* (trypanosome) five days earlier. It is clear to see the significant differences between the lower number of mice that have parasitaemia and a were infected with trypanosomes prior to malaria innculation, and the higher number of mice who were that have parasitemia and were infected by malaria cells only. Figures (a) and (b) were produced by me and the orange line represents the theoretical concentration of *P. berghei*. What is important to notice is the initial decrease in the concentration of *P. berghei* as the total concentration of slender and stumpy cells (green line) remains fairly consistent. Figure (b) shows the concentrations over 300h and from these results we can see clearly the population of malaria causing cells decreasing, but we also see it increase for a little as the population of slender cells increase.

5 Conclusions and Future Work

Overall, the findings of this study provide further evidence for a role played by Trypanosomes in the prevention of malaria infection. Utilizing past studies, I developed a modified system of differential equations to model a trypanosome infection and account for the addition of another bacterial population known to commonly co-infect hosts who also have Trypanosomes present. *P. berghei* causes malaria in its host but it has been found that upon co-infection, Trypanosomes suppress the effects of the malaria causing cells. The equation I added to account for this new population of malaria causing cells is based on the same assumptions used by Tyler et al. (2001). The results from this theoretical model do show a decrease in the concentration of malaria causing cells over time, but it is also interesting to note the relationship between the curve of slender cells and the curve of *P. berghei* cells. As the concentration of slender cells rises after reaching a low point, so does the concentration of malaria causing cells.

It would be extremely interesting to take this model even further to explore the reason why the presence of Trypanosomes impacts the presence of malaria cells. Does the immune system have anything to do with it? Should we account for spontaneous cell death? There are so many biological questions to explore and so many parameters we can incorporate into this model to get a more accurate representation of how this process actually works.

6 References

Sanches-Vaz, Margarida et al. (2019). "Trypanosoma brucei infection protects mice against malaria". In: *PLoS Pathogens* 15.11, e1008145.

Turner, CMR, N Aslam, and C Dye (1995). "Replication, differentiation, growth and the virulence of Trypanosoma brucei infections". In: *Parasitology* 111.3, pp. 289–300.

Tyler, Kevin M et al. (2001). "Limitation of Trypanosoma brucei parasitaemia results from density-dependent parasite differentiation and parasite killing by the host immune response". In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 268.1482, pp. 2235–2243.

Venter, Frank, Keith R Matthews, and Eleanor Silvester (2022). "Parasite co-infection: an ecological, molecular and experimental perspective". In: *Proceedings of the Royal Society B* 289.1967, p. 20212155.

7 C++ Code

trypanosomes.cpp

```cpp
#include "functions.cpp"
#include <cmath>
#include "cpgplot.h"
#include <iostream>

int main()
{
  //taking interval input from terminal
  int n;
  std::cout << "Enter number of steps: " << "\n";
  std::cin >> n;

  //setting float arrays for plotting
  float tp[n+1], yp[n+1], xp[n+1], xpyp[n+1], cp[n+1];

  // Open a plot window
  if (!cpgopen("/XWINDOW")) return 1;

  // Set-up plot axes
  cpgenv(48.,320., 5., 10., 0, 1);
  // Label axes
  cpglab("Time (h)", "Cells/mL", "");

  //Setting up the runge-kutta method
  double vx1, vy1, vc1, vx2, vy2, vc2, vx3, vy3, vc3, vx4, vy4, vc4, dt, x, y, c,
    t, t_end, r, r2, f0, m2, m;

  //setting parameters
  t_end = 320.00;
  dt = t_end/n;
  y = 1e-10; //change this
  t = 0.00;
  x = 3910.00; //X0 value from paper
  r = 0.11;
  f0 = 1.2e-9;
  m = 0.012;
  m2 = 0.02; //mortality rate of malaria causing cells
  c = 3.91e9; //population of malaria causing cells
  r2 = 0.20;  //rate of division of malaria causing cells

  //Initializing arrays
  tp[0] = t;
  yp[0] = log10(y);
  xp[0] = log10(x);
  xpyp[0] = log10(x + y);
  cp[0] = log10(c);

  // Compute the function at the points
  for (int i=1;i<=n;i++) {

    vx1 = dt * (dxdt(x,y,r,f0));
    vy1 = dt * (dydt(x,y,f0,m));
    vc1 = dt * (dcdt(x,y,c,f0,m,m2,r2));

    vx2 = dt * (dxdt((x + 0.5 * vx1), (y + 0.5 * vy1), r, f0));
```

```
    vy2 = dt * (dydt((x + 0.5 * vx1), (y + 0.5 * vy1), f0, m));
    vc2 = dt * (dcdt((x + 0.5 * vx1), (y + 0.5 * vy1), (c + 0.5 * vc1), f0, m, m2,
     r2));

    vx3 = dt * (dxdt((x + 0.5 * vx2), (y + 0.5 * vy2), r, f0));
    vy3 = dt * (dydt((x + 0.5 * vx2), (y + 0.5 * vy2), f0, m));
    vc3 = dt * (dcdt((x + 0.5 * vx2), (y + 0.5 * vy2), (c + 0.5 * vc2), f0, m, m2,
     r2));

    vx4 = dt * (dxdt((x + vx3), (y + vy3), r, f0));
    vy4 = dt * (dydt((x + vx3), (y + vy3), f0, m));
    vc4 = dt * (dcdt((x + vx3), (y + vy3), (c + vc3), f0, m, m2, r2));

    t = t + dt;

    x = x + (vx1 + (2.00 * vx2) + (2.00 * vx3) + vx4 )/6;
    y = y + (vy1 + (2.00 * vy2) + (2.00 * vy3) + vy4 )/6;
    c = c + (vc1 + (2.00 * vc2) + (2.00 * vc3) + vc4 )/6;

    std::cout << x << "\n";

    tp[i] = t;
    yp[i] = log10(y);
    xp[i] = log10(x);
    xpyp[i] = log10(x + y);

    if (c <= 0) {
        cp[i] = 0.00;
    } else {
        cp[i] = log10(c);
    }
  }

  std::cout << "Number of steps: " << n << "\n";
  std::cout << "timestep (deltat): " << dt << "\n";
  std::cout << "final y(t=10): " << y << "\n";
  //std::cout << "final error at t=10: " << exact_y - y << "\n";
  //std::cout << "final error at t=10 divided by deltat: " << (exact_y - y)/pow(dt
    ,4) << "\n";

  // Plot the curve
  // This line is
  cpgsci(4);
  cpgline(n+1,tp,yp);
  // Plot the next curve
  // This line is
  cpgsci(2);
  cpgline(n+1,tp,xp);
  // Plot the next curve
  // This line is
  cpgsci(3);
  cpgline(n+1, tp, xpyp);
  //Plot the next curve
  cpgsci(8);
  cpgline(n+1, tp, cp);
  // Pause and then close plot window
  cpgclos();
}
```

functions.cpp

```cpp
#include <iostream>
#include <math.h>

//This file will have the equations for dxdt and dydt
//0.11 = r, 0.0000000012 = f0, 0.012 = m, 3.91x10^3 = X0
//I am using these parameters now to try and replicate paper
//These values come from mouse 1 in a table on the paper
//Maybe try other values?

double dxdt(double a, double b, double r, double f0) {
    double value = (r - (f0 * (a+b))) * a;
    return value;
}

double dydt(double a, double b, double f0, double m) {
    double value = (f0 * (a+b) * a) - (m*b);
    return value;
}

double dcdt(double a, double b, double c, double f0, double m, double m2, double
    r2) {
    double value = (r2 - (f0 * (a + b))) * a - (m*b) - (m2*c);
    return value;
}
```

Makefile

```
trypanosomes: trypanosomes.o
  c++ -o trypanosomes trypanosomes.o -ltrapfpe -lpgplot -lcpgplot -lX11 -lm

trypanosomes.o: trypanosomes.cpp
  c++ -c trypanosomes.cpp
```