REPORT

Introduction to Focus Areas - Complex Systems - Group 8

Maximilian Otto*, Dominik Bannwitz and Florian Herzler

*Correspondence:
maximilian.otto@fu-berlin.de
Department of Mathematics and Computer Science, Free University of Berlin, Takustraße 9, 14195 Berlin, Germany
Full list of author information is available at the end of the article
†Equal contributor

Abstract

Goal of the project: Model within-host dynamics of viral infection in the human body and predict the unknown parameters. Analyze the influence of initial free viruses in regards to the infection probability.

Methods used in the project: implement and solve an ODE using *SciPy 1.9.3*

Main results of the project: The amount of initial viruses is in a non-linear relation to the probability of infection. If the initial virus concentration is high enough, the probability of infection reaches 100% over multiple independent simulation runs.

Possible improvements: Perform a higher number of simulations to get a better estimate. Analyze the influence of the different definitions of "virus elimination". Use the exact same version of *SciPy* as all other groups on the same platform to have comparable results. Modify the *Params.txt* so it does not contain 0

Keywords: ODE; SSA; Complex System; Infection; Prediction; Optimization

1 Introduction

The rates of a virus spreading within the human body from cell to cell over time, as well as the reactions of the host can be mathematically modeled and simulated to a certain extend. This helps to understand under which initial conditions a virus manages to infect the cells of a host organism or not and is important to get a better understanding of the virus in general. Performing this kind of simulations could also give approximated predictions of infection rates within populations. Tweaking different variables and conditions of a system can help to understand the dynamics and how it can be possible to achieve a desired outcome after a specified time.

1.1 Goal

In this report we wanted to tackle the question of infection probability of a single host in regards to an initial exposure to a virus by implementing a virus infection model as an ODE. For this, we had to simulate the in-host dynamics of the infection with an integral solver and a stochastic simulation algorithm. The estimate of all unknown parameters of the virus by fitting the dynamics to one patients' virus dynamics data had to be computed as well.

2 Methods

2.1 Modelling

The first step was to implement the following model with its corresponding propensities and set up an ODE solver using the initial states $x_0(t_0) = k_0/k_1$, $x_1(t_0) = 0$ and $x_2(t_0) = 20$ based on the patients' data.

Model of virus infection

$$R_0 : \emptyset \xrightarrow{r_0} X_0 \qquad R_1 : X_0 \xrightarrow{r_1} \emptyset$$
$$R_2 : X_0 + X_2 \xrightarrow{r_2} X_1 \qquad X_1R_3 : X_1 \xrightarrow{r_3} \emptyset$$
$$R_4 : X_1 \xrightarrow{r_4} X_1 + X_2 \qquad R_5 : X_2 \xrightarrow{r_5} \emptyset$$

Reaction rates

$$r_0 = k_0 \qquad r_1 = k_1 \cdot x_0$$
$$r_2 = c_2 \cdot x_0 \cdot x_2 \qquad r_3 = c_3 \cdot x_1$$
$$r_4 = c_4 \cdot x_2 \qquad r_5 = c_5 \cdot x_2$$

In this model, X_0 represents the amount of uninfected target cells, x_1 the infected cell and x_2 the amount of free viruses. Target cells get produced with rate r_0 and cleared with rate r_1. Cells get infected by free viruses with rate r_2 and cleared by T-cells with rate r_3. New viruses get produced by infected cells with rate r_4 and cleared by antibodies with r_5. Based on this model, we implemented the model and following tasks using the programming language Python [1].

2.2 Simulation

For the model prediction the functions `integrate.odeint()` and `optimize.curve_fit()` from the package `scipy` [2] were used. For plotting the packages `matplotlib` [3] as well as `seaborn` [4].

Optimization: The second step was to estimate all unknown parameters $(c_2, ..., c_5)$. For that we were given $k_0 = 100$ as well as $k_1 = 0.1$ from a previous study and a data set of the virus dynamic of an infected patient. The setting of the function `optimize.curve_fit()` included the given data as *t_data* and *y_data*. The bounds were set to $(0, np.inf)$ and *p0* was set to $[0.1, 1, 5, 1]$.

Parameter identity 1st round: After fitting the model to the given data, we initialized the unknown parameters in each simulation run randomly (maximum value: 4) to evaluate how their estimation would change.

Parameter identity 2nd round: We were able to choose one parameter to get determined by a biological experiment. To investigate, which parameter should be determined, we plotted the spread of the estimations. Subsequently, we ran the model fitting again with the new defined parameter.

2.3 The Prediction

Exposure - Infection Probability: To simulate the virus infections, we implemented the stochastic simulation algorithm (SSA) [5, 6]. This algorithm can be used in a discrete state space to simulate stochastic dynamics and to calculate trajectories. Next, we wanted to find out how the infection probability differs, when a patient is exposed to various numbers of viruses. For this we set up a simulation representing our virus infection model and collected the results of each run. We performed 300 simulations and stopped each one when the number of viruses exceeded 50, meaning the host is infected, or the virus has been eliminated within the host. We defined the latter event as having no infected cells and no free viruses. After this, the infection probability was calculated from the results of the simulations.

3 Results

The predictions of our model, the distribution of the corresponding parameters and the results of our simulations are presented in this section.

3.1 Modeling & Simulation

The results of our model and parameter fitting are presented below.

3.1.1 Optimization

Figure 1 shows the fitted model to the input virus infection data. It is shown that the number of viruses increased rapidly during the first time step, and exponentially decreasing afterwards. The fitted function followed closely the first three data points, but did not increase as much as the fourth. Between time points two to four, most data points are below the fitted function, but about after 6.3, nearly all other data points are higher than it.

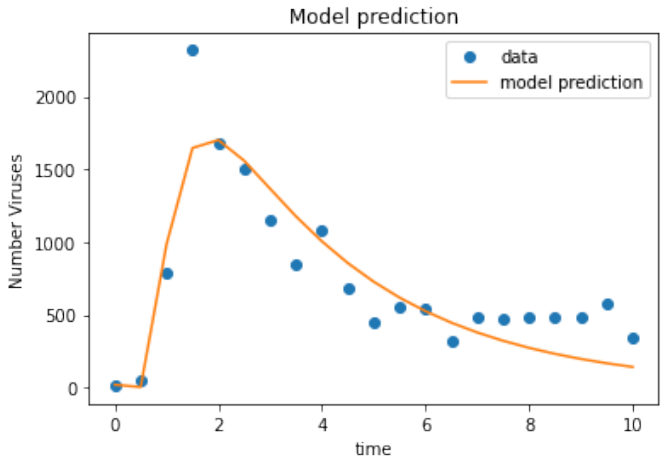

Figure 1 In blue, the input data points and in orange, the fitted function calculated with `curve_fit()`

3.1.2 Parameter identity 1st round:

The comparison between the different optimal model parameters after prediction can be seen in Figure 2 on the next page. It can be seen that $c2$ and $c3$ are lower than $c5$, which in turn is lower than $c4$. In the plot without normalization (left) it is is shown that the parameter $c4$ is not only the highest, but also the one with the the largest standard deviation. After normalization, this has changed and the spread of $c2$ is higher than that of the other three.

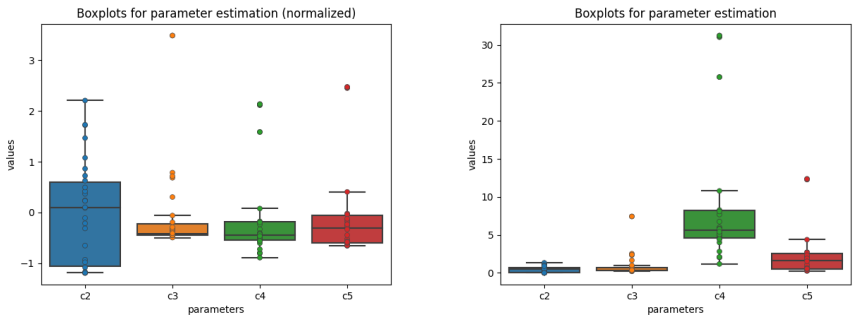

Figure 2 Distribution of the estimated parameters from 30 different randomly initialized parameters. Left is normalized by subtracting the mean and dividing by the standard deviation

3.1.3 Parameter identity 2nd round:

After setting the parameter $c4$ in the beginning to 10, the variance of the optimal parameter are decrease. This is shown in figure 3.

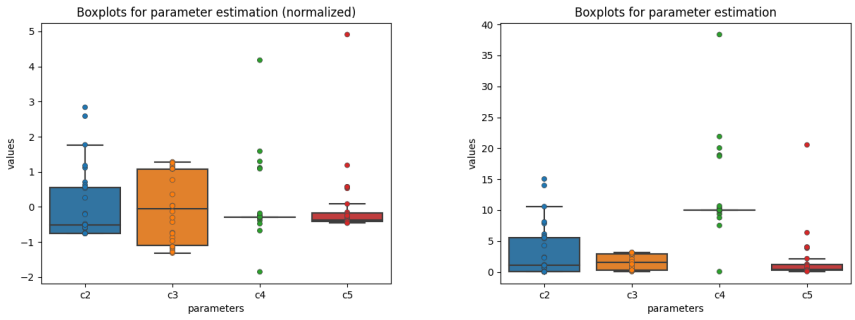

Figure 3 Distribution of the estimated parameters after initializing the parameter $c4$ to 10 for 30 different simulations with randomly initialized parameters $c2$, $c3$ and $c5$. Left is normalized by subtracting the mean and dividing by the standard deviation.

3.2 The Prediction

The following section redners the outcomes of the SSA.

3.2.1 Exposure - Infection Probability:

Figure 4 compares the outcomes of a different amount of free viruses at the beginning are compared. Both plots - the left bar chart and the right line chart - show the same data. It can be seen that the probability of infection with more than one virus is always above 90% and reaches 100 % with five free viruses. Only with one free virus in the beginning, the infection probability is 73.7%. A more detailed view

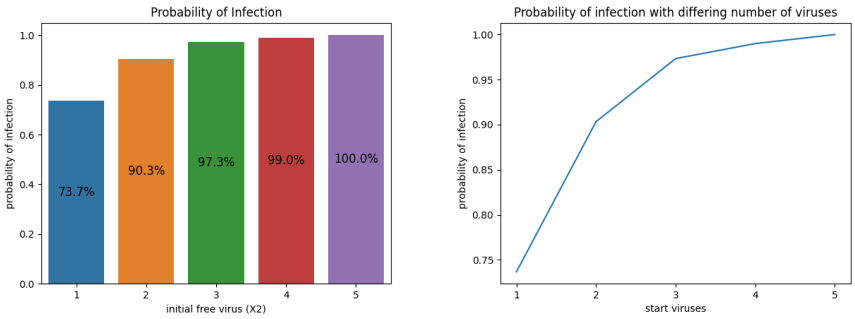

Figure 4 Infection Probability in regards to different initial free viruses.

on the outcome on the infection probabilities, when starting with three free viruses in the beginning, can be seen in figure 6 on page 7. The left box plot shows the amount of uninfected cells, how many cells are infected can be seen in the middle and the amount of free viruses is displayed on the right. It can be seen that most of the time there are 800 to 850 uninfected cells and 60 to 80 infected cells. Free viruses are only 0 or 50 which correlates with the two termination conditions. In figure 6 on page 7, similar plots for starting with one free virus can be seen. Figure 7 on page 7 depicts the results for an initial virus amount of two, and figure 8 on page 8 was obtained with the initial amount of 4, and figure 9 on page 8 with an initial amount of 5.

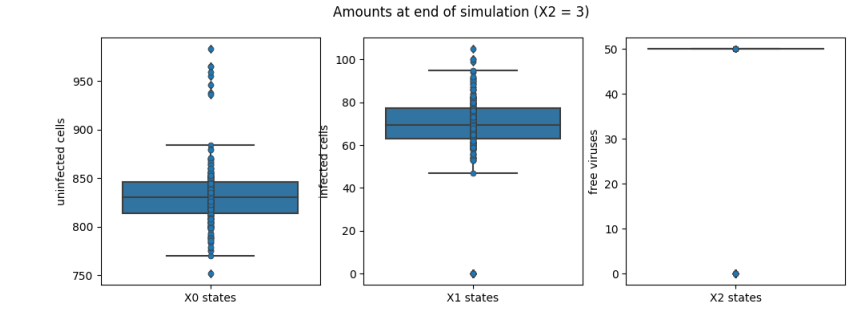

Figure 5 Last states of all simulations with initial value $X2 = 3$

4 Discussion

4.1 Modeling & Simulation

4.1.1 Parameter identity: 1st round

It appeared to us, that parameter c_4 was harder to determine than the other parameters that had a smaller standard deviation and less drastic outliers. Apart from that, the identification of parameters worked reliably.

4.1.2 Parameter identity: 2nd round

We chose to fix parameter c_4 because it had the biggest standard deviation and seemed to be hardest to determine, as displayed in figure 2.

We did not achieve an optimal solution as provided in *Params.txt*, so we were not convinced by our results. However, we got differing results when the exact same code was executed on different machines and on *Google Colab*, so we suspect that this is a versioning or platform issue with the `SciPy` library. Also the results in *Params.txt* included a zero-value, which is not useful for generating further plots for comparing different initial values because the second reaction rate r_2 will never produce any infected cells.

By fixing the parameter with the biggest standard deviation, we were hoping to decrease the standard deviations of the other estimated parameters. This partly worked, as it can be seen in figure 3. The standard deviation for c_4 decreased by a lot, so that the boxplot is just a line at the fixed value 10 with a few outliers in both directions. The standard deviations for the remaining parameters did not change drastically, neither in an in- or decreasing direction.

4.2 The Prediction

Because we were unable to obtain the exact same results for the parameter estimation, we continued witch the provided values. We chose the termination criteria to be one of the following:

- number of free viruses $x_2 \geq 50$
- number of free viruses and infected cells $x_0 = x_2 = 0$

After performing 300 simulations for initial values of x_2 between 1 and 5, we plotted the last value of each x_0, x_1, x_2 and visualized them in 5 different boxplots. In Figure 5 on the previous page, you can see the results of the 300 trials for initial value $x_2 = 3$. The remaining boxplots can be found in the Appendix. It is clear from looking at the plots for x_2 that the values are either 50 or 0, just as one would expect from the termination criteria. The states of x_0 are not so clearly separated, ranging from 0 to 1000, but never more than that. States of x_1, the infected cells, tend to be mostly around 65, with a smaller group at value 0 (a partial termination condition). As it can be seen in Figure 4, the probability of infection increases with the initial number of free viruses, which is consistent to our expectation. The certainty of infection for the initial $x_2 = 5$ was surprising to us. After discussing the predictions, we did not come up with a concise formula using only the exposure with a single virus. This question was not really clear, because one cannot come up with a simple formula based on one value ("based on the elimination probability after exposure with a single virus"). In our modeling, the exposure with 5 free viruses led to an infection probability of 100% after 300 simulations, so we assume that

any additional virus will not decrease this probability. We can also imagine, that with the increase in simulations it gets increasingly more improbable to achieve a 100% certainty. Because of this, we would assume, that the $\lim_{x\to\infty} f(n_{viruses}) = 1$, something like a modified version of $f(x) = 1 - \frac{1}{x}$.
Furthermore, splitting the termination condition into two different, independent termination conditions could lead to a drastic change in our results, because the amount of of uninfected cells would not be allowed to bounce back up, after it got decreased to 0. Therefore, it would be also more probable that one of the termination conditions would be fulfilled far earlier in time and more different initial conditions would be also interesting for further investigations.

5 Appendix

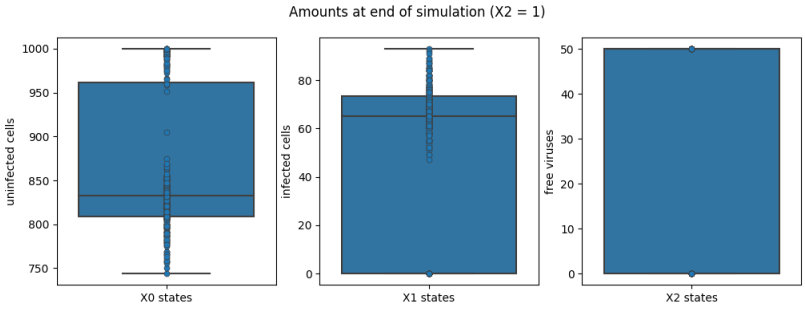

Figure 6 Last states of simulations with initial value $X2 = 1$

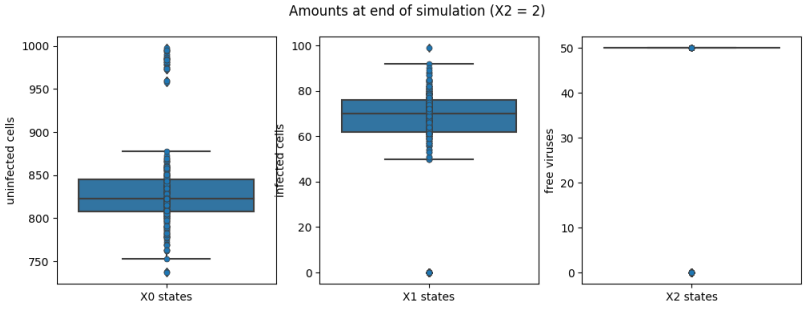

Figure 7 Last states of simulations with initial value $X2 = 2$

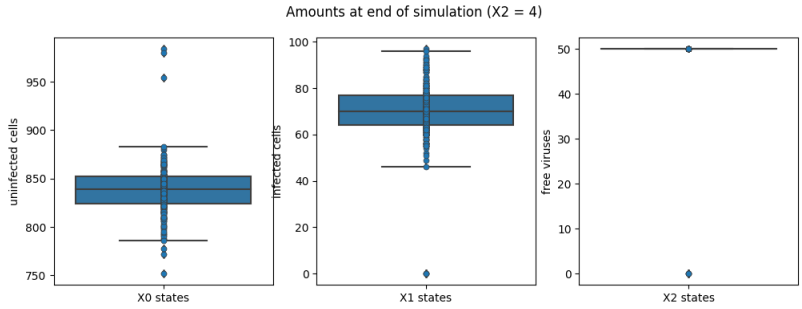

Figure 8 Last states of simulations with initial value $X2 = 4$

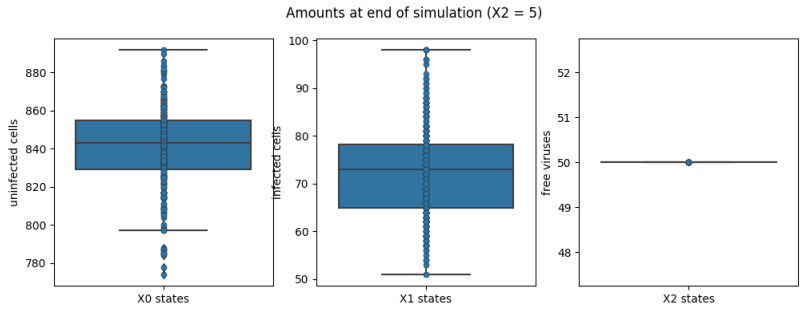

Figure 9 Last states of simulations with initial value $X2 = 5$

References

1. Van Rossum, G., Drake Jr, F.L.: Python Reference Manual. Centrum voor Wiskunde en Informatica Amsterdam, ??? (1995)
2. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods **17**, 261–272 (2020). doi:10.1038/s41592-019-0686-2
3. Hunter, J.D.: Matplotlib: A 2d graphics environment. Computing in Science & Engineering **9**(3), 90–95 (2007). doi:10.1109/MCSE.2007.55
4. Waskom, M.L.: seaborn: statistical data visualization. Journal of Open Source Software **6**(60), 3021 (2021). doi:10.21105/joss.03021
5. Gillespie, D.T.: A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. Journal of computational physics **22**(4), 403–434 (1976)
6. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. The journal of physical chemistry **81**(25), 2340–2361 (1977)