Imperial College London
Department of Physics

Computational Evolution: Using Agent-Based Modelling to investigate the evolution of bacteria in the presence of antibiotics

Project Code: ASTRO-Clements

Hameed Khandahari

CID: 01069638

Word Count: 5520

Supervisor: Dr Dave Clements

Assessor: Dr Tim Evans

Submitted in part fulfilment of the requirements for the degree of
Bachelor of Science in Physics

Abstract

A computer model was created to investigate the evolution of meningicoccal bacteria in the presence of penicillin G, a beta-Lactam antibiotic. Agent based-modelling techniques were employed in *Python* to run simulations from which data was collected using *Pandas* dataframes. Three different cases were considered. These were the effects of finishing the prescribed treatment early, skipping one dose, skipping more than one continuous doses. The evolution of other 'genes' including the number of offspring during a reproduction and genes relating to the phenomena of dormancy was also considered. While the parameters used may not match the real-world cases, this project is a proof of concept that agent-based modelling can be used to model systems like bacteria in a human body.

Contents

Chapter 1

Introduction

1.1 Agent-Based Modelling

From economics to sociology, agent-based modelling is a versatile tool used to model complex systems [1]. In this project, agent-based modelling techniques are used in the context of Artificial Life. This is a discipline which, in the words of one of its co-founders, Chris G. Langton, studies natural life by attempting to recreate biological phenomena from scratch within computers [2]. Within the field of artificial life, this project investigates the emergence of antibiotic resistance amongst bacteria. The specific case of resistance to penicillin G amongst the meningococcal bacteria was chosen as a focus.

1.2 Antibiotics

Antibiotics are substances of microbial origin that can inhibit the life processes of other organisms [3]. Prior to 1939, when the first *Sulphonamides*, became the first widely available antibiotic, the annual death toll due to diseases such as meningitis and pneumonia was in the thousands [4] Ever since, antibiotics have completely revolutionised the field of medicine. This is clear to see by looking at Table 1.1, where the death rates due to bacterial illnesses before and after the advent of widespread anitbiotics are presented [5].

Year	Deaths due to bacterial illness	Total U.S. Population	Death Rate due to Bacterial Illness (per 100,000 population)
1930	293,623	118,708,333	247.7
1936	277,541	129,083,333	216.3
1952	93,014	155,758,376	59.7
1960	90,345	179,321,462	50.4
2002	110,202	289,055,601	38.1

Table 1.1: Data showing change in death rates due to Bacterial Illnesses before and after the advent of widely available antibiotics. The following illnesses are included in the data: Typhoid and Paratyphoid Fever, Typhus Fever, Scarlet Fever, Whooping Cough, Diphtheria, Tuberculosis, Pneumonia, Diarrhea and Enteritis, Appendicitis, Puerperal Septicemia. This table has been adapted from a report by Joseph Gottfried, Harvard Law School [5]. The data itself is taken from the *Vital Statistics of the United States.*

It is striking that the total death rate due to the bacterial illnesses stated above dropped by almost five times between 1930 and 1960: a period covering the so-called Golden Age of Antibiotics [6] . This was before the emergence of widespread resistance and subsequent move to less optimal cost-prohibitive drugs, many of which have significant side effects [7].

1.3 Resistance

Continued use of antibiotics can lead to the rise of resistance in bacteria, rendering the antibiotics ineffective. The rise of resistance can be explained using Darwin's theory of evolution by natural selection. Bacteria in a system will have some degree of genetic variability; this can be introduced by random mutations. An estimate of the rate of mutations is approximately 0.0003 mutations/gene/generation [8]. Some mutations may result in a mechanism of resistance to the antibiotics so that when the antibiotics are added, the resistant bacteria are able to survive the treatment. The *'fitness'* of the resistant bacteria is said to be higher than that of the non-resistant bacteria in an environment where the antibiotics as present. This results in an increase of the fraction of the population that are resistant.

Beta-Lactam antibiotics bind to enzymes involved in the synthesis of the cell wall, thus the cell wall is weakened and no longer capable of holding in the pressure from within the cell. This pressure bursts the cell from the inside, killing the bacteria [9] The mutation that gives rise to

resistance against beta-lactam antibiotic results in the production of the protein *beta-lactmase*, this attacks the chemical structure of the beta-lactam antibiotics, leaving them ineffective against the bacteria [10] Therefore an effective treatment must kill the bacteria as soon as possible before resistance emerges. Fleming himself was aware of this as early as 1945 where he warned 'the use of too small doses, so that, instead of clearing up the infection, the microbes are educated to resist penicillin' [11]

1.4 Dormancy

Another mechanism for survival is dormancy. This is a property of bacteria whereby the metabolism of the cell is greatly reduced [12]. As antibiotics can only attack bacteria that are dividing, they are virtually ineffective against dormant bacteria. It has been recently discovered that some bacteria are able to change the period and frequency of their dormancy to match the antibiotic dosing period. This effectively means they are able to survive the antibiotic treatment without developing a mechanism of resistance whereby they kill the antibiotics. These bacteria are known as persistent rather than resistant.

1.5 Specific Setup

In this project, we have tuned our analysis to investigate meningococcal disease. This is caused by a bacteria called Neisseria menigitiditis (meningococcus). The most common treatment for the meningococcus is currently penicillin G. This is a beta-Lactam antibiotic. It is estimated that around 10% of the UK population carries this bacteria in the back of the throat however the concentration is usually so low that the immune system itself can prevent runaway growth that would lead to significant symptoms. The primary reason for looking at this specific case is the vulnerability of university students. A study released in January 2000 shows the carriage rate of the meningococci bacteria amongst first year university students increased from 6.9% on day 1 of the first term to 23.1% on day 4 and 34.2% by December [13]. This sudden increase

in carriage is likely a result of living in shared facilities and interacting with hundreds of new people in a short space of time. These are similar reasons for the so-called 'Freshers Flu'. Another reason for looking at meningococcus is that the meningococcus bacterium is relatively simple to model as it spherically round and does not have any flagella. Some bacteria, like Spirillum, are much more complex as they have more complex shapes and have tails, known as flagella, which they can use to change the direction of their motion.

Chapter 2

Method

2.1 Background

In this project, a simple two-dimensional system containing agents(bacteria), food, and antibiotics was created using Python v3.6. The agents interact with the food and antibiotics, resulting in the evolution of its genetic make-up through random mutations during reproduction. An object-oriented programming approach was used to create this programme. This is for several reasons. Firstly, by using inheritance, a property of object-oriented programme, we can can create a parent class (Particle) from which the agent, food and antibiotic classes inherit properties. This makes the code more readable and elegant. Secondly, it allows the code to be organised neatly and it is more efficient because only the lines of code that are required for a given situation, i.e. the required *methods* are run. Also, it allows both members working on this project to work on different aspects of the code independently and then merge the additions onto a single master branch. This was done using GitHub as it allows for easy collaboration, a good interface for comparing code and a record of the code submitted by each member.

2.2 Genetic Makeup

The genetic makeup of the bacteria are stored as attributes of the *agent* object.

In this model, we consider four genes, these are properties of the bacteria and do not change throughout its life. They can however change from generation to generation through random mutations. To speed up the process of evolution, the probability of mutation for any of the genes is set at 10%.

The genes are:

- whether they are resistant to bacteria
- the number of offspring during a reproduction
- how often they become dormant
- how often they stay dormant

Each of the genes has an implicit or explicit fitness cost. For example, the reproduction threshold of an agent is raised if it is resistant.

2.3 The Graphical Simulation

To aid with debugging, as well as produce a useful graphical representation of the system, the module *'Pygame'* was used to create an animation of the system. *Pygame*, as the name suggests, is often used to create video games, however it is also ideal for this project as it creates a highly customisable animation. The most common way of creating animations in Python is *matplotlib*'s *FuncAnimation* method. However at the early stages of this project, a decision was made to use *Pygame* instead because of its built in clock function. Although the use of the clock function was later abandoned (see *2.8: Development of the Code*), a decision was made to continue using *pygame* because of its simplicity.

A screenshot of the live simulation created using *pygame* can be seen below.

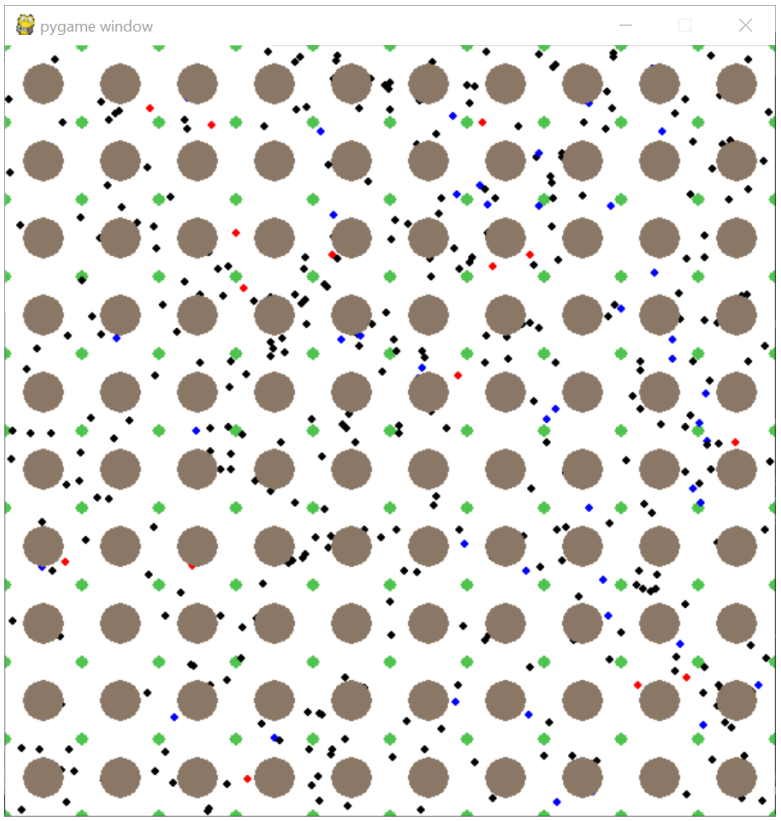

Figure 2.1: A screenshot of the live simulation created using pygame

2.3.1 Environment

The environment is a 2D rectangle. Throughout the experiment, the size of the environment was set to be 600x600 pixels. Although this choice is somewhat arbritary, it is a good compromise between a very large environment, where the environment could not fit on the screen and a small environment where it would be difficult to see what is happening.

The bacteria can be killed directly either by antibiotics or the immune system. The immune system encompasses all of the natural methods used by the human body to fight disease, such as the use of white blood cells or by causing a fever. The immune system has been modelled to kill a set percentage of bacteria (up to a limit), irrespective of their genetic makeup, at set time intervals. Whilst collecting the data for the Results section, the immune system was set to kill 4% of all agents, (up to a limit of 250 agents) every 1100 time steps.

2.3.2 Agents

The black, blue and red circles are the bacteria themselves. The three different colours represent bacteria in three different states. The black bacteria are not resistant to the antibiotics whereas the blue bacteria are resistant. Neither the blue or black bacteria are in a state of dormancy. The red bacteria, however, are dormant. Once dormant,the bacteria do not move and cannot be killed by the antibiotics. However, they may still be killed by the immune system. The bacteria have a food level which is initialised at the start of the simulation. This food level is a measure of the energy of the bacteria. Their food level decreases every time they move, increases when they eat, and if it reaches above a certain threshold, they can reproduce.

2.3.3 Food

The brown circles represent the food. When initialising the set-up, the user can input the desired concentration of the food within the environment. The concentration is calculated by adding the area of all the food points and dividing it by the total area of the environment.

In this model, the food is evenly spaced and non-diminishing as there is a plentiful supply of food for bacteria within the blood supply. During the early stages of the development of the programme, code was implemented to decrease the food concentration when the bacteria eat. However, this was later abandoned because of the realisation that there is virtually an infinite food supply for the bacteria. So the food stays constant throughout the simulation.

2.3.4 Antibiotics

The green circles are the antibiotics. As with the case of the food, the antibiotics are evenly spaced in the environment. Antibiotics have a biological half-life. This is the defined as the time taken for the effectiveness of the antibiotic to half. Biologically, the effectiveness of the antibiotics decreases as natural body processes flush the drug out of the patients system. This means that the half-life of a drug is not only determined by the drug itself, but the patient too. The effectiveness of the antibiotic determines the probability of whether it is successful (see later). The dosing frequency is the time between two successive doses of the antibiotic. This effectively tops up the effectiveness of the antibiotic back to its maximum value. The pygame animation has been set up such that the colour of the antibiotic starts off as a dark green but slowly lightens as it loses its effectiveness. This allows for easier debugging as it is clear when the antbiotic is added and gives an indication of the instantenous effectiveness of the antbiotic at any given point in time.

2.4 Interactions

There are 5 different interactions that take place between the objects.

2.4.1 Moving

After each time step, the agent moves within the environment. As it moves, it uses up energy, hence its food level decreases. During the data collection phase of the project, the decrease in

food level was set to 0.01 per step. It's coordinates are updated according to:

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \mathbf{v}t, \tag{2.1}$$

where $\mathbf{r}_{i+1} = (x_{i+1}, y_{i+1})$ is the position after the move, $\mathbf{r}_i = (x_i, y_i)$ is the position before the move , $\mathbf{v} = (v_x, v_y)$ is the velocity vector and t is the time between each step. If the agent reaches the edge of the environment, it simply bounces back, much like gases in a thermodynamics model.

2.4.2 Eating

If a bacterium passes through food, its food level increases by 0.5. The code has been designed such that the bacteria only eats once if it passes through the food. In other words, if the bacteria remains within a food point for more than one time step, it food level only increases by 0.5. This was required to prevent runaway reproduction whereby the bacteria eat too much, reproduces whilst still inside the food point it's offspring do the same. 'Eating once' was implemented by only allowing the bacteria eat only when it is on the edge of the food point. Although this means the food level increases twice, once upon entering and once upon leaving, the user can simply change the food level increase in the code to be half of the desired food level . Due to the nature of the set up, where we are using continuous coordinates rather than a matrix set up where the agents are only allowed at certain points, there are numerical errors we have to deal with. The tolerance of the width of the allowed region where the eat condition is satisfied was experimented with to ensure the bacteria were not missing the food point.

2.4.3 Dying

When the bacteria move, its food level decreases. If it falls below 0, the bacteria runs out of energy and dies.

2.4.4 Reproduction

When the bacteria eats, its food level increases, if it rises above its reproduction threshold, it can reproduce. The reproduction threshold is one of the genes of the bacteria. It allows us to investigate whether it is advantageous to have a high reproduction threshold and produce more offspring, or to have a low reproduction threshold and produce fewer offspring

The threshold is determined by the bacterias genetics. If the threshold is high, they are less likely to reach it, but if they do, they can produce more offspring. This allows for an interesting analysis whereby one can compare whether it is beneficial to keep a supply of food and then reproduce into many new bacteria, or reproduce as quickly as possible, but with few offspring.

2.4.5 Neutralisation

Neutralisation refers to the event where the antibiotic kills the bacteria. If we consider a non-resistant bacteria, a check is made to see if the bacteria is within the antbiotic. Then based on the effectiveness of the antbiotic (*See Section 2.4.4*), a weighted random choice is made whether or not to kill the bacteria. So if, at the moment a non-resistant bacteria makes contact with an antibiotic point, and the effectiveness of the antibiotic at that moment in time is 0.5, there is a 50% chance that the antibiotic will neutralise, i.e. kill, the bacteria.

2.5 Code Logic

Once the system has been initialised, the code loops through each bacterium and performs checks based on their position and the clock time. If these conditions are satisfied, methods are called accordingly. The logic can be seen in the form of a flow chart in Figure 2.2. The programme stops once the user-input time of simulation is over. Data collected at each time step for each agent is stored in a multidimensional pandas dataframe. Pandas is a library in Python used for easy data analysis and manipulation. In this project, all the data is stored

in a pandas dataframe and from this data, plots are created using matplotlib which is used to analyse the data.

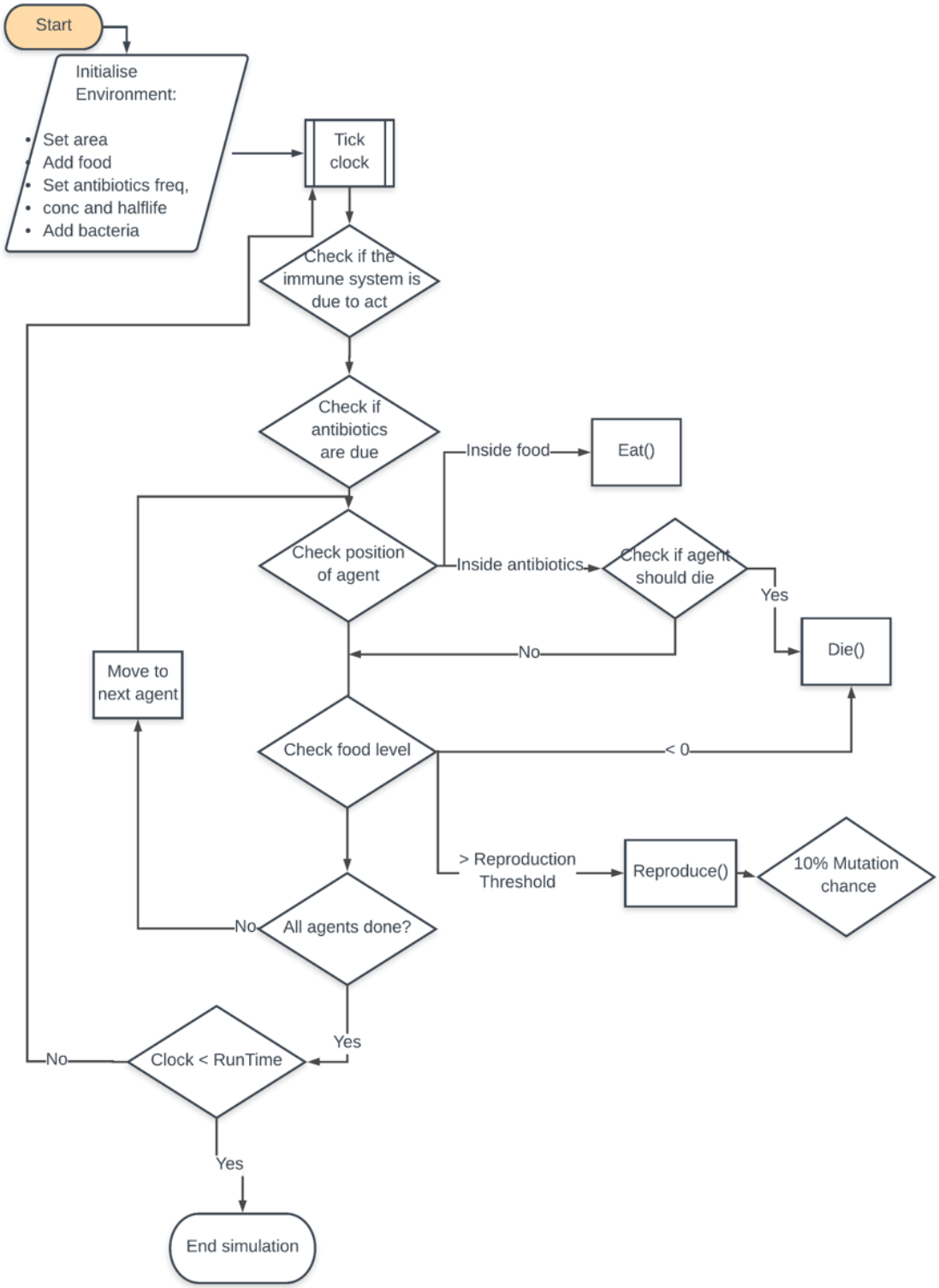

Figure 2.2: Flowchart showing the logic of the code. The flowchart was created using *Lucid-Chart*

2.6 Seeding

The system is initialised with random starting positions to introduce unpredictability into the system. This is done using numpy's built in random function. This uses a Mersene Twister pseudo-random number generator which has a period length of $2^{31} - 1$. The random number generator is also used elsewhere such as the genetic mutations and the neutralisation conditions. By seeding the random number generator, one is able to replicate the results. This is useful for testing comparison cases without compromising the advantages brought in by randomness.

2.7 Development of the code

As highlighted in the previous sections, there were several decisions made throughout the development of the code which shaped fundamental aspects of the final programme.

- Coordinate Setup: Initially, a matrix set-up was used. This has the advantage that it creates a coordinate system where the allowed positions of the agent are known exactly. However, it limits the directions it can move. Due to this limitation, the system was altered such that each agent has a position in continuous x-y space. This however, brings its own problems in the form of numerical errors.

- Use of the clock: Another change made was with the clock. The clock updates through each loop. Initially pygames built in clock function was used. This calculated the time taken between each frame and progressed the system by that value. However, as the number of agents increases, so does the complexity of the system and therefore the computational time required to run the code. So although this gave a smooth animation, it was abandoned in favour of a manually set clock, whereby the clock is updated by a set amount every time it goes through the loop.

- Immune System: Previously, the immune system was set-up such that it killed a much higher percentage of bacteria but the time between which it acted was greater. This

caused great instability in the system so it was changed to killing fewer agents more frequently, resulting in smoother data.

Chapter 3

Results and Discussion

3.1 Finding the base parameters

To compare the effect of different treatments of penicillin G on a patient infected with the meningicoccus bacteria, the system must be calibrated correctly so that the correct conclusions can be drawn.

In the absence of an immune system or antibiotics, the population of the bacteria is free to rise exponentially. This can be confirmed by plotting the rate of change of the population against the population itself. i.e. $\frac{dP}{dt}$ against,P. This is shown in Figure 3.1.

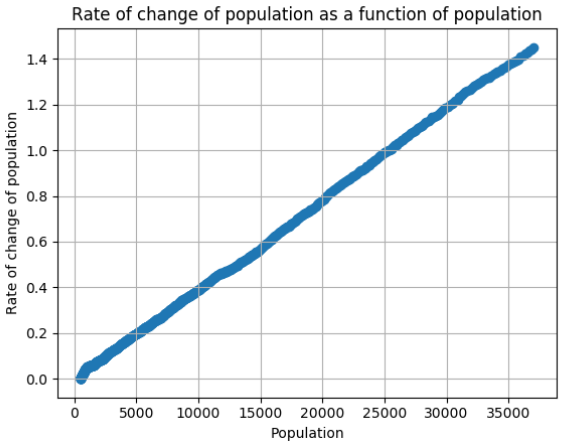

Figure 3.1: A scatter plot of dP/dt against P gives a linear fit. The gradient of this is positive, indicating that the population is exponentially rising.

The gradient of this line was found, using the numpy.polyfit method, to be $k = 3.94 \times 10^{-5}$, therefore the population is given by $P(t) = P(0)e^{kt}$ The greater the initial population, $P(0)$, the quicker it will rise in the early stages. If the rate at which the immune system kills the bacteria is lower than the rate of reproduction, the population will grow indefinitely.

Parameters were set for the immune system such that in the absence of antibiotics, the immune system can deal with 200 agents. This is because the rate of reproduction proportional to the number of agents as shown in Figure 3.1 and the rate of reproduction for a system with 200 agents is smaller than the rate with which the immune system kills the agents. The same parameters for the immune system, when dealing with 500 agents are not strong enough to keep the bacteria population from growing exponentially. Therefore we define an *ill person* as someone who has over 500 agents in their system and a *healthy person* as someone who has 200 or fewer agents in their system. Once the parameters for the immune system were believed to have been found, they were tested by re-running the simulation multiple times. Due to the random nature of agent-based modelling, the simulations are always slightly different but as shown in Figures 3.2 and 3.3, they work as intended because the immune system can deal with 200 agents, but not 500 agents.

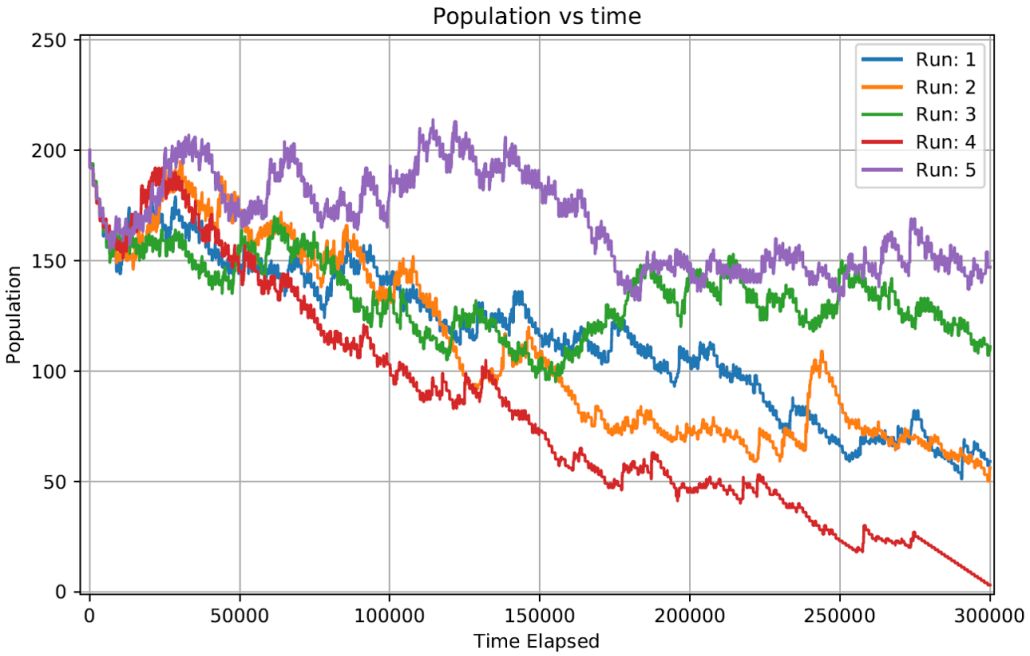

Figure 3.2: Starting with 200 agents,the simulation was run without antibiotics in the presence of just the immune system. It was repeated 5 times and on all occasions, the immune system is able to deal with the infection because the population is decreasing.

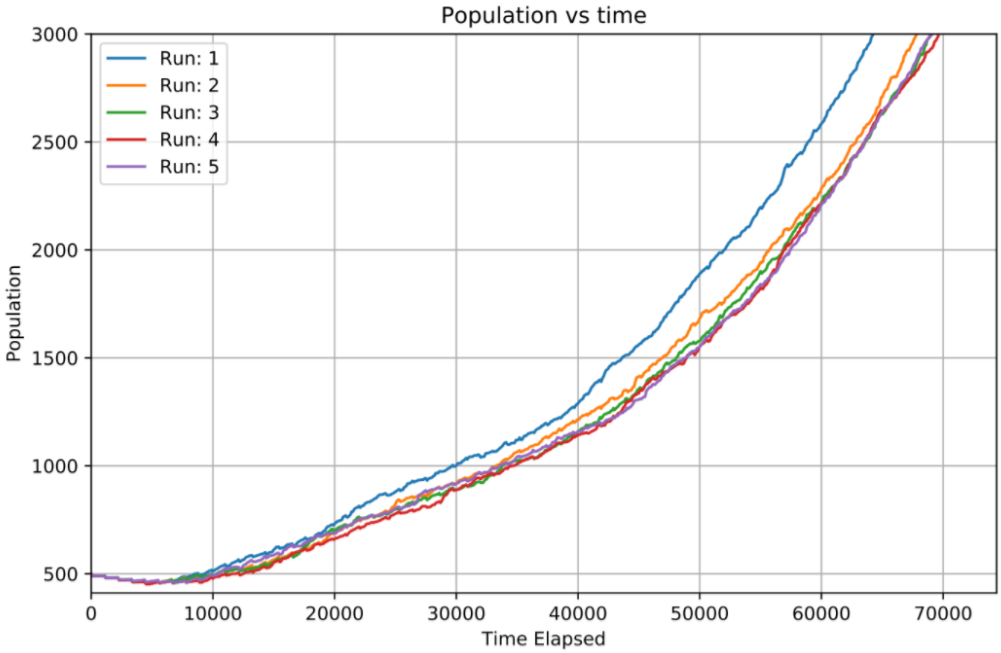

Figure 3.3: Starting with 500 agents,the simulation was run without antibiotics in the presence of just the immune system. It was repeated 5 times and on all occasions, the immune system is unable to deal with the infection because the population increases exponentially. The exponential nature of this increase was confirmed by plotting the same data on a log-linear graph. On this plot, there was a linear relation between the log(population) and time, hence there must be an exponential relationship between population and time.

The standard antibiotics treatment kills the bacteria in around 10 doses. However, the parameters have been set such that 7 doses is usually sufficient and 10 doses are prescribed to be safe. By experimenting with different combinations of parameters, the following were found to replicate the basis experiment well:

Dosage frequency: 16000 time steps, half-lilfe of antibiotic: 4000 time steps, immune system frequency = 1100 time steps, immune system kill fraction = 0.04

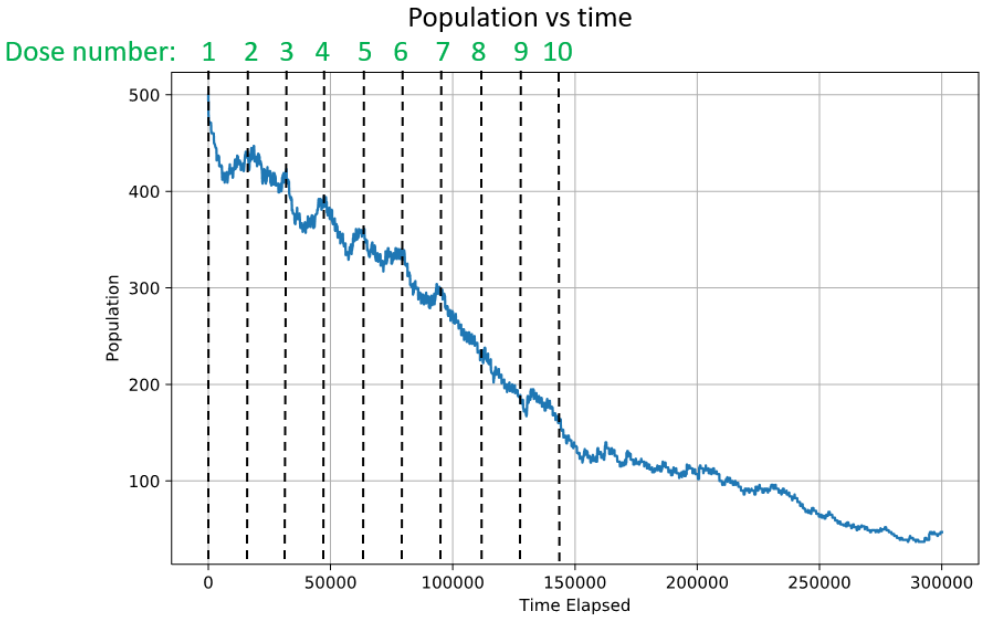

Figure 3.4: The standard antibiotic treatment: In this case, it takes around 8 doses to fall below 200 agents so there is a very high probability that the patient will be cured of the infection

3.2 Finishing the course early

The parameters have been set up such that 10 doses of the antbiotics are usually enough to kill an ill patient (Starting with 500 agents in the environment). A common problem in the medical world is patients often finish their prescribed course of medication earlier than instructed. This is because the medication often relieves the patients symptoms and so the patient may incorrectly assume that the bug has been completely eliminated. However, resistant strands of the bacteria may still prevail so if the level of infection is not reduced to a level where the immune system alone can deal with it, the resistant population may suddenly repopulate the environment. The antibiotics being used would then be ineffective against this strain and a new type of antibiotic would need to be introduced. A set of simulations were performed, stopping after n doses where $1 < n < 9$. This was repeated 5 times with different seeds to ensure the results were not statistically anomalous. In the run shown in Figure 3.5, we can see that with these specific parameters, the infection will still be contained if the patient only takes the first 8 doses. This was done deliberately as doctors often slightly over prescribe to increase the probability that the infection is defeated. This simulation was repeated 5 times in total. From this, we can take the mean and standard deviation to give an estimate of how many doses it is safe to stop with using these parameters. Applying the basic statistics, we conclude that in this set up, stopping the treatment after 7.6 ± 0.8 is doses is likely to be safe.

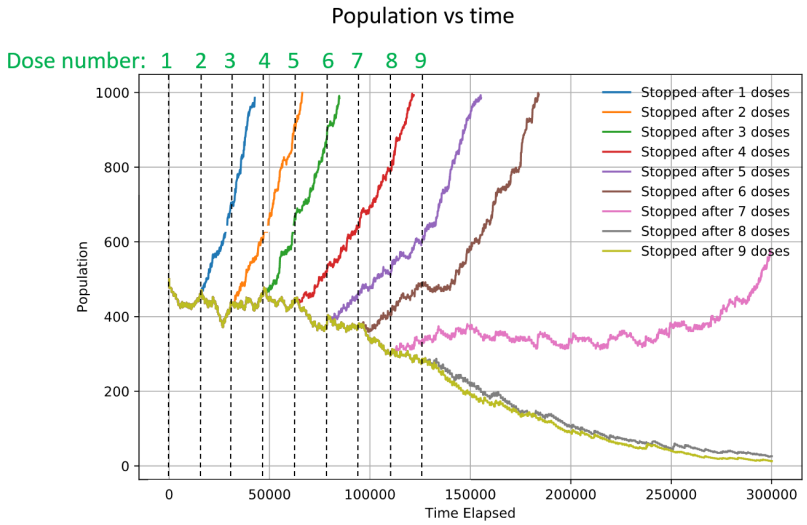

Figure 3.5: Plot showing effect on population of the bacteria if the treatment is stopped early. The dotted lines indicate the doses.

3.3 Skipping Doses

Another common problem is patients skipping one dose during their treatment. Here we investigate the effects of missing one dose at various stages of the treatment. As before each set of runs is initialised with a seed such that the runs within each set can be compared directly. Five sets of runs were taken and the mean and averages of these were used to obtain the numerical results.

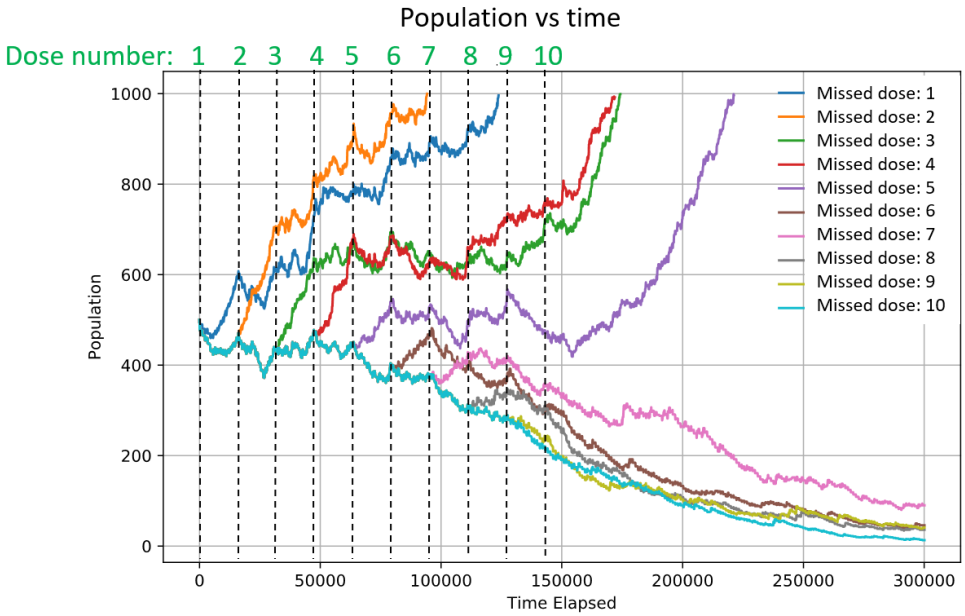

Figure 3.6: Plot showing effect on population of the bacteria if doses are skipped. The dotted lines indicate the doses.

In the set of runs shown in Figure 3.6, we can see interesting results. It is clear that missing the earlier doses is more harmful to the patient. This can be explained due to the fact that the population will be higher earlier on, so missing an early dose will lead to the population increasing well above the critical value and so it will be more difficult to control, due to the increased reproduction that comes with a larger population. By running the simulation for 5 sets of data (with different seeds), an estimate for the earliest single dose one could miss and still recover was found to be 7.4 ± 2.7 (1 s.d.) Similar sets of data were taken for missing two and three successive doses. The earliest dose one could miss and still recover for these cases was found to be 7.6 ± 2.1 and 8.4 ± 2.1 respectively.

One example of missing three successive doses is shown in Figure 3.7 In this run, Doses 1,2,3,4,5 are taken as prescribed, Doses 6,7,8 are skipped and Doses 9,10 are taken.

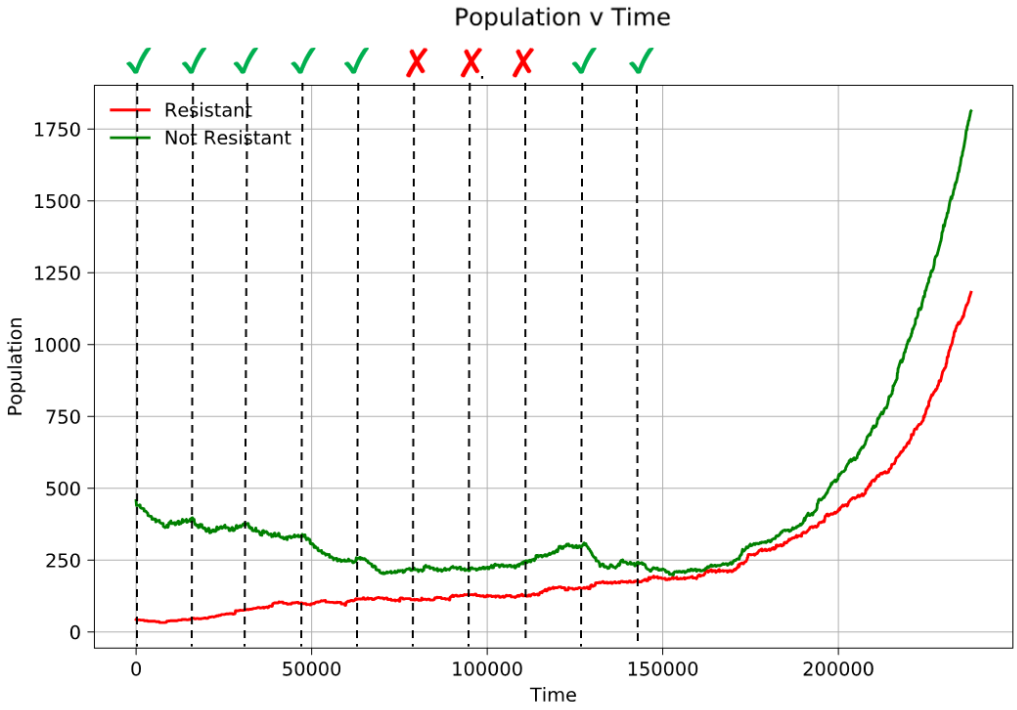

Figure 3.7: Plot showing the total population and the resistant population plotted against time on the same set of axes. The ticks represent doses which were taken, and the crosses represent missed doses

Figure 3.7 shows how the total population decreases during the initial stage of the treatment where the patient is following the prescribed doses. When the three doses are skipped, the population begins to increase until the doses 9 and 10 are added where it drops again. After these doses however, the population begins to rise exponentially as there are no more antibiotics and the population (around 250) is above the critical level so the immune system alone cannot deal with it. It is also interesting to look at the proportion of the bacteria with the resistant gene. This is shown in Figure 3.8. As there is a fitness cost associated with the resistant gene (*see Section 2.2*), the resistant gene becomes unfavourable when it is not needed, i.e. during the skipped dose. We can see how the number with the resistant gene drops during this phase while it increases when antibiotics are present.

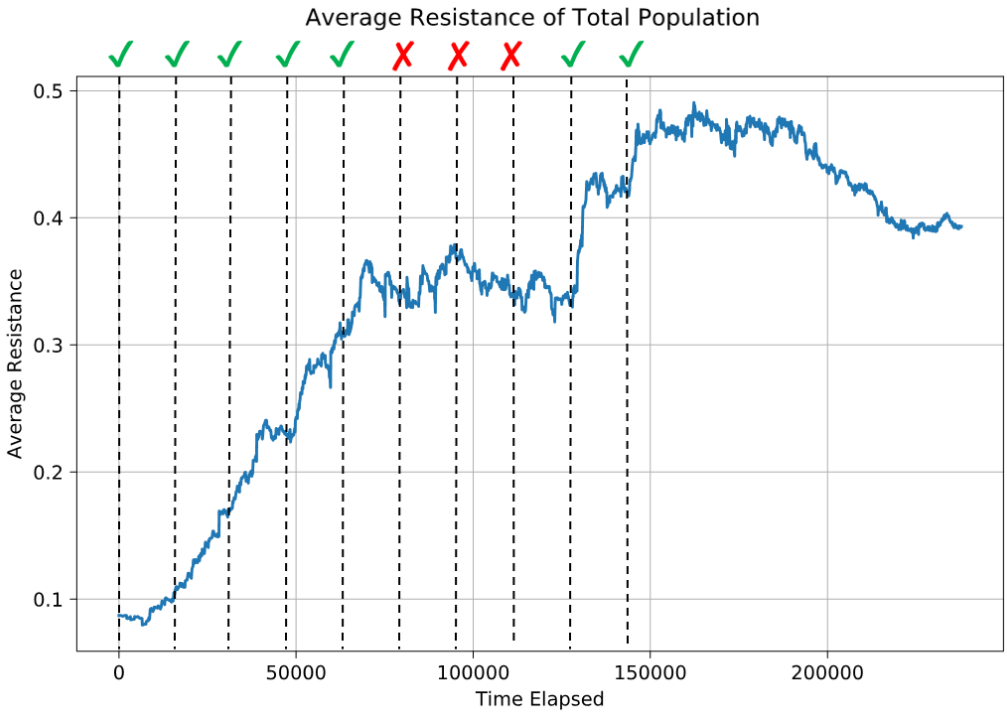

Figure 3.8: Plot showing the effect on the average resistance of the system when antibiotics are present and when doses are missed. Note how the average resistance begins to decrease when the antibiotics are not present for a while.

3.4 Evolution of other genes

Besides resistance, as shown in section 2.2, the bacteria have genes for the number of offspring, the frequency with which they become dormant and the length of the dormancy. Rather than being binary, as is the case with resistance (where a bacteria is either resistant or not resistant), these genes can have a wide range of values. It can therefore be interested to see how the genetic makeup of the system changes as everything interacts and time progresses

3.4.1 Evolution of Dormancy Period and Dormancy Frequency

2D animated histograms are plotted using matplotlib to show the changing fitness landscape of the system. The fitness landscape is a visual representation of how what genes are best suited to the environment. The measure of *fitness* being used is the population. Here, period refers to the length of time the bacteria remains dormant and frequency refers to how often they become dormant. These are properties of the bacteria and do not change during their lifetimes.

The system is initialised randomly. There is a uniform probability for the bacteria to have a dormancy period and length within a user-specified range . This randomness can be seen in Figure 3.9a. As time progresses, the system evolves. Eventually after a time of 150,000 time steps, a clear gradient forms. Short dormancy periods are favoured over longer ones. This can be explained by considering what it means to be dormant. When a bacterium is dormant, it is unable to move, thus unable to eat, thus unable to reproduce. While it cannot be killed by the antibiotics, the immune system can still act on it. So agents with long dormancy periods do not survive because they are killed by the immune system during their dormancy.

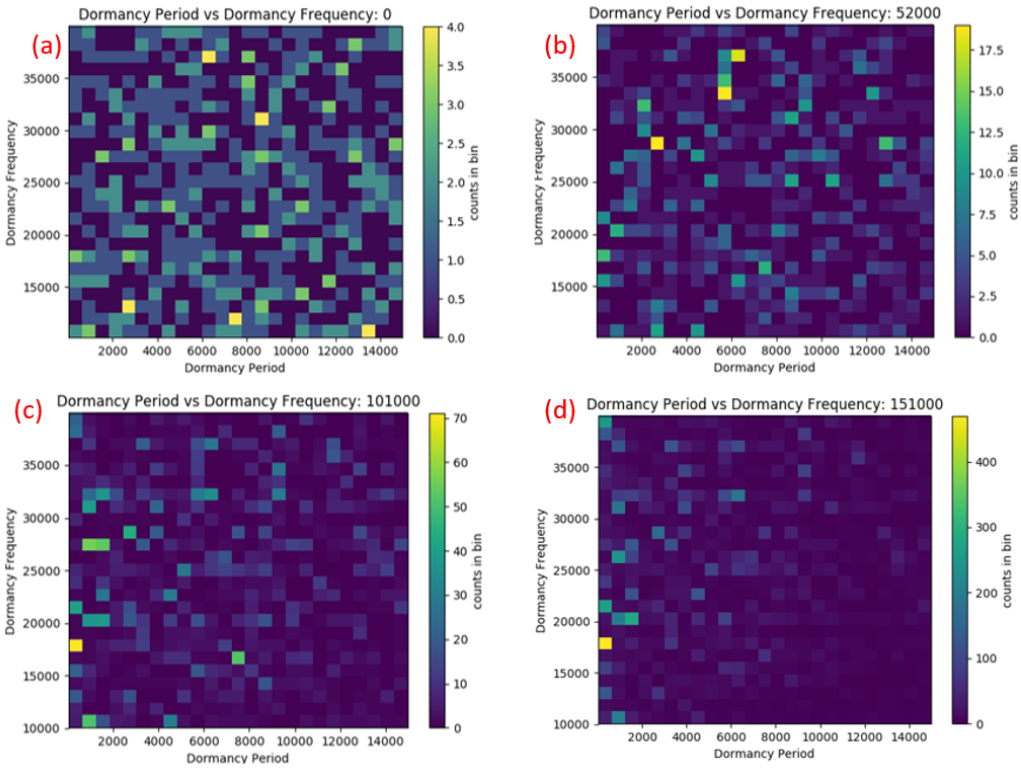

Figure 3.9: 2D Histograms showing the fitness landscape of the system

The simulation shown in Figure 3.9 continues to run until time = 169000 time steps. At this stage, the highest peak in the fitness landscape occurs at Dormancy Period = 300 ± 150 time steps and Dormancy Frequency = 17800 ± 600 time steps

This appears to be the optimal combination for the genetics which maximises the survival of the bacteria against the given treatment regime.

3.4.2 Evolution of the number of Offspring

As explained in *Section 2.2*, there is a cost associated with all the genes. The cost for having a higher offspring number is that the reproduction level increases. Using a Python dictionary, the corresponding reproduction levels for each offspring number is as follows

$$2 : 4, 4 : 6, 6 : 8, 8 : 10, 10 : 15,$$

where the number on the left is the offspring number and the number on the right is the reproduction level. When an agent reproduces, the food is shared equally amongst its offspring.

A 2D animated histogram showing data for the offspring number and the dormancy frequency was generated any potential relationship between the two genes. This is shown in Fig 3.10.

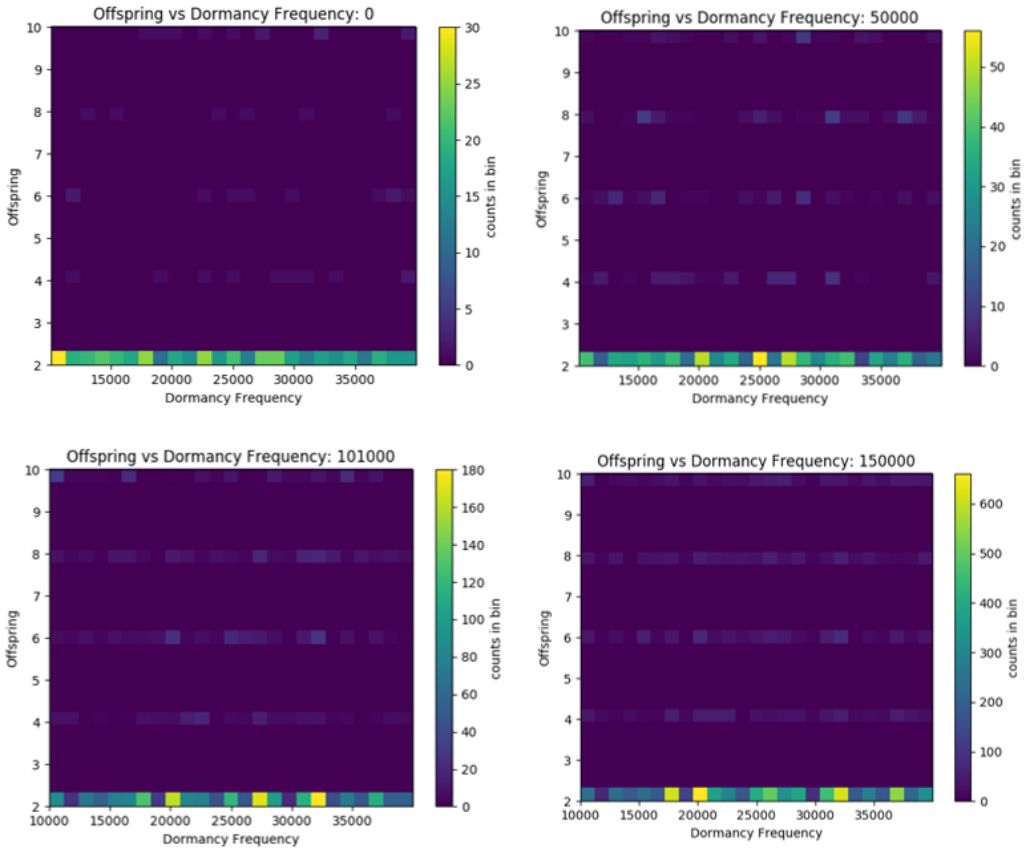

Figure 3.10: 2D Histograms showing the fitness landscape of the system

The distinct bands in this histogram are due to the discrete nature of the offspring numbers as opposed to the previous histogram were continous values were allowed for dormancy period and frequency. We can see from these histograms that the agents with the offspring number 2 dominate. This is because the fitness cost introduced by the higher reproduction levels for other offspring numbers, is too high for non-binary fission.

Chapter 4

Conclusion

The world is facing an antibiotic apocalypse [14]. These are the words of Professor Dame Sally Claire Davies. Should antibiotic resistance continue to spread, and no alternatives to our current antibiotics be found be found, it is not unfeasible to imagine a world in the near future where diseases that claimed so many lives before the 1930s could once again claim victims. A large part of the problem is financial. Many of the large pharmaceutical companies are unwilling to invest large amounts of money into research for new antibiotics because resistance soon catches up and renders them useless [7]. A possible solution to saving costs could potentially be computer simulation. In this project, a simple model has been created and tested in a few different environments. By increasing the complexity of the system, perhaps it may be possible to develop a programme that could help chemists and biologists make more informed choices about the direction of their research thus saving time, money and potentially lives

The short term goal of this project was to demonstrate the feasibility of using agent-based modelling in the context of bacteria and antibiotics and to run some test cases to ensure everything is working as expected. This worked as we were able to produce results that make sense. Also, the live simulation could be used as an educational tool to show people how resistance spreads.Longer term, perhaps it could be developed to have some use in medical research.

Acknowledgements

I would like to thank

- My project partner, Miss Paris-Anne O'Shea, for her equal contribution to this project
- My supervisor, Dr Dave Clements, for his contribution to the direction of this project and his support during weekly meetings
- My assessor, Dr Tim Evans for suggestions to improve aspects of the project during the viva

Bibliography

[1] Uri Wilensky and William Rand. *Introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo.* The MIT Press, 2015.

[2] Sanghamitra Bandyopadhyay, Ujjwal Maulik, Ujwal Maulik, and Jason T.L. Wang. *Analysis of Biological Data*, volume 3. 2007.

[3] Paul R Burkholder. Antibiotics. *Science*, 129(3361):14571465, 1959.

[4] Latest news — before the revolution. *BBC News*, Jun 1998.

[5] Joseph Gottfried. History repeating? avoiding a return to the pre-antibiotic age, Apr 2005.

[6] Julian Davies. Where have all the antibiotics gone? *Canadian Journal of Infectious Diseases and Medical Microbiology*, 17(5):287290, 2006.

[7] P K Singh. Pre-antibiotic era looming large - the world is almost out of time, Jul 2004.

[8] John W Drake. Rates of spontaneous mutation. *GENETICS*, 148(4), Apr 1998.

[9] R.b. Ghooi and S.m. Thatte. Inhibition of cell wall synthesis is this the mechanism of action of penicillins? *Medical Hypotheses*, 44(2):127131, 1995.

[10] K. Poole. Resistance to beta-lactam antibiotics. *Cellular and Molecular Life Sciences*, 61(17), 2004.

[11] Penicillin's finder assays its future. *New York TIMES*, Jun 1945.

[12] Simon Van Vliet. Bacterial dormancy: How to decide when to wake up. *Current Biology*, 25(17), 2015.

[13] D A Ala'Aldeen. Dynamics of meningococcal long-term carriage among university students and their implications for mass vaccination. *Journal of Clinical Microbioligy*, 38(6), Jun 2000.

[14] Robin McKie. 'antibiotic apocalypse': doctors sound alarm over drug resistance, Oct 2017.