Modeling Dual Predation of *Escherichia coli* by Bacteriophage and *Bdellovibrio Bacteriovorus*

Kaiya Taura, Emmett Myers, Russell Ramsay

02-251: Great Ideas in Computational Biology
May 1, 2024

Abstract

Antibiotics remain our primary and most reliable method of combating bacterial infection and their subsequent illnesses. However, the threat of developing bacterial strains resistant to our treatments presents a severe and costly threat to the development and deployment of new treatments. Nature shines a ray of hope on the scene, providing us with natural predators for the problematic bacteria, Bacteriophages, and *Bdellovibrio bacteriovorus*. Lab results show that individually, both predators are incapable of completely eradicating the prey population in their samples. In response, we propose a new treatment method: dual predation. By deploying both of the predators on the prey sample at the same time, we can take advantage of their varied predation methods to curb the development of resistant prey. To model our sample, we employed a particle diffusion simulation, optimized with a QuadTrees collision algorithm to simulate the interactions between the predators and prey. We also made use of a Gray-Scott model to visualize the diffusion of the entities in our sample. Our lab results and simulations were very promising, displaying complete prey lysis within 48 hours. Despite only conducting tests *in vitro*, our results have promising implications for the development of future antibacterial treatments.

1. Introduction

In recent years, antibiotic resistance has become an increasingly prevalent challenge in the world of modern medicine. As bacteria become resistant to treatments, antibiotics become less effective which leads to prolonged illness. According to the World Health Organization, this is a critical threat to the field of medicine as the misuse of antibiotics becomes more common, resulting in more drug resistance. In turn, infections are becoming harder to treat and medical procedures are becoming riskier due to the lack of treatments to combat antimicrobial resistance. Therefore, we are interested in finding alternatives to the typical antibiotic treatments. One such option is the use of the bacteriophages and *Bdellovibrio bacteriovorus,* both natural predators of bacteria. The paper considered this potential solution and sought to model the interactions between these predators against *Escherichia coli,* using differential equations. Ultimately, these equations are very complex and their interpretability and applicability in future *in vivo* tests seem to be limited due to their high threshold for understanding. Thus, beginning with these differential equations, we wanted to simplify them by considering the base interactions. Using our understanding of biological systems present during these interactions, we model the cells applying simple rules. Basing our research on the paper, we wanted to simulate the population dynamics using particle simulation to verify that the interactions stated in the paper coincide with the lab data and the differential equations. Utilizing these "chemical" reactions instead of the differential equations as the foundation for the simulation allows for a more complete biological understanding of the phenomenon of dual predation. Therefore, we will begin by introducing the main actors and their reactions.

Bacteriophages

Bacteriophages, commonly referred to as phages, are viruses that specifically target and infect bacteria. They do this by attaching themselves to the surface of bacterial cells and injecting their genetic material into them. The phage's genetic material then hijacks the bacterial cellular machinery, using it to produce more phages instead of performing its usual functions. However, this process does not kill the bacteria instantly, rather it continues until lysis, a process which results in the cell membrane bursting and releasing numerous phage particles that go on to infect other bacterial cells. This process of using bacterial cells to replicate is the basis of phage therapy, a potential alternative or complement to antibiotic treatment for bacterial infections. Thus, phage therapy has emerged as a potential solution to combat antibiotic-resistant bacteria, as phages can specifically target and kill certain bacterial strains. Unlike broad-spectrum antibiotics, which can harm beneficial bacteria along with harmful ones, phages are highly specific, which could potentially reduce the collateral damage to the body's microbiome.

However, the potential for phage resistance is still possible. Just as bacteria can evolve resistance to antibiotics, they can also develop mechanisms to evade phage attacks. This can occur through various means, such as mutations in the bacterial surface receptors or the production of enzymes that degrade phage genetic material. At the same time, phages can execute a counter strategy against antiphage defense mechanisms. We can see this by analyzing the lab data from the paper by Hobley et al., where we observed that relying on only phages was insufficient in eliminating the entire bacterial population of E. coli. Those surviving the initial phage attack were the E. coli cells with innate genetic mutations allowing for resistance or the cells capable of developing resistance. Consequently, subsequent generations of bacteria were able to rebound and reach a stable population once again.

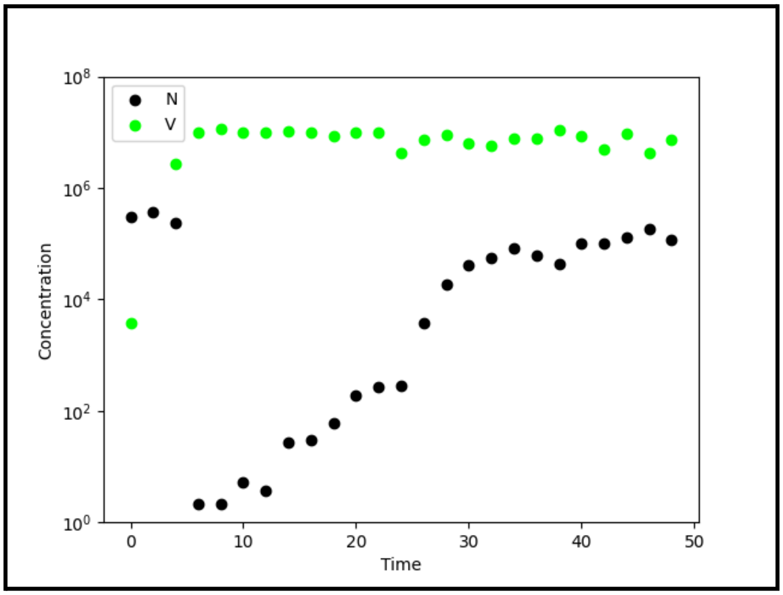

Figure 1. Impact of bacteriophages on E. coli population over time

Bdellovibrio bacteriovorus

Bdellovibrio bacteriovorus is a unique type of bacteria, preying on other bacteria in its environment to reproduce. This is achieved in a similar process to bacteriophage infection, with some key exceptions. The initial entry remains the same, *B. bacteriovorus* attaches itself to the host cell and begins to penetrate the outer layer. The process diverges here, as once the layer is penetrated, the target cell immediately undergoes apoptosis, and dies. At this point, the target cell is designated a bdelloplast as the process continues. Inside the bdelloplast, *B. bacteriovorus* continues to grow and eventually begins to proliferate, producing progeny. Finally, lysis occurs, and the bdelloplast bursts open, releasing the new *B. bacteriovorus* cells out into the system, ready to repeat the process.

The properties of *B. bacteriovorus* as a bacterial predator imply a strong case for using it as a therapeutic method, but such use cases are sparsely documented. This is likely due to the phenomenon of plastic resistance reducing the effectiveness of *B. bacteriovorus* over time. The phenomenon is observed as a protective outer coating growing around the target bacteria cells in the presence of *B. bacteriovorus*. This outer layer protects against the penetrative ability of *B. bacteriovorus*, preventing infection in the first place. That said, the protection is not perfect, and irregularities in the layer can be exploited to penetrate the cell. The prevailing theory explaining plastic resistance is that upon a cell transforming into a bdelloplast, an extracellular signal is produced that influences neighboring cells into raising a plastic layer. This greatly slows down the proliferation of *B. bacteriovorus*, reducing its therapeutic potential.

The lab results in the paper by Hobley et al. reflect this fact, initially displaying a steady decline in E. coli (prey) populations, relative to the amount of *B. bacteriovorus* present in the system. Roughly 40 hours into the trial though, the prey population begins to reach a steady state, indicating that the remaining members have developed plastic resistance. Our prey will then be able to repopulate, but since plastic resistance is not heritable, the new generation will not be innately protected, and we predict the system will reach this steady state again, in a continuous oscillation.

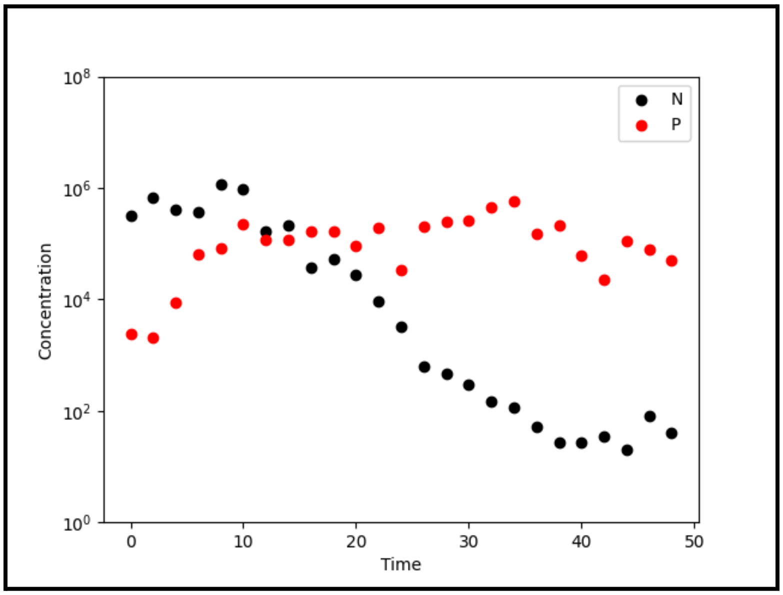

Figure 2. Impact of *Bdellovibrio bacteriovorus* on E. coli population over time

Dual Predation

While both bacteriophages and *B. bacteriovorus* have shown promise in combating bacterial infections individually, their effectiveness is limited due to the potential of bacterial resistance. This can lead us to consider employing a strategy of dual predation, utilizing both bacteriophages and *B. bacteriovorus* simultaneously. Dual predation capitalizes on the complementary mechanisms of action of these predators. Bacteriophages target and infect bacterial cells, injecting their genetic material and ultimately causing cell lysis, while *B. bacteriovorus* directly invade and prey upon bacterial cells, using them as a nutrient source for replication. In theory, by combining these two predators, there is a synergistic effect that enhances their efficiency. Fortunately, in practice, according to the lab data, this is also the case.

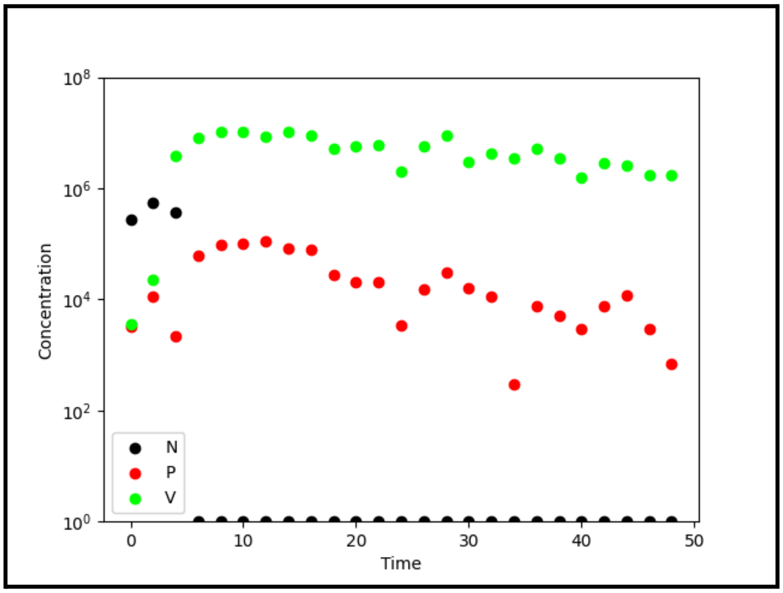

Figure 3. Impact of dual predation by *Bdellovibrio bacteriovorus* and bacteriophage on E. coli population over time

2. Methodologies and Algorithms

Experimental Data

In the lab experiment done by Hobley et al., an initial concentration of 2.9╳10^8 CFU/mL of E. coli, 5.0╳10^6 PFU/mL of *Bdellovibrio bacteriovorus*, and 5.0╳10^6 PFU/mL of bacteriophage were used in 4 different experiments of dual predation by both *Bdellovibrio bacteriovorus* and bacteriophage, single predation by just *Bdellovibrio bacteriovorus*, single predation by just bacteriophage, and a control with no predation. The concentrations of each organism were tracked every 2 hours for 48 hours total for each of the tests. The above figures show the results of the experiments, and we used this data to fit our model.

Particle Simulation

In our first method, we used a particle simulation approach to model the interactions between the organisms. In this simulation, each organism type was represented by different particles, and reactions were simulated by the interactions between these particles with certain probabilities, mimicking the chemical kinetics of the biological system. We tracked the population of each cell type over time, which could be compared with the data to adjust the probabilities and fit the data.

Cell Types

We abbreviate the cell types as follows:

All E. coli (Prey)	N
Sensitive E. coli	N_S
Bacteriophage Resistant E. coli	N_R
Plastic-Resistant E. coli	N_P
Bdellovibrio bacteriovorus (Predator)	P
Bdelloplast	B
Bacteriophage (Virus)	V
Phage-Infected Prey	I
Medium	M
Signal	S

Reactions

The particles reacted according to the following reactions:

Bdellovibrio Attacks Sensitive E. coli:	$P + N_S \rightarrow B$	(1)
Bdellovibrio Attacks Phage-Resistant E. coli:	$P + N_R \rightarrow B$	(2)
Phage Attacks Sensitive E. coli:	$V + N_S \rightarrow I$	(3)
Phage Attacks Plastic-Resistant E. coli:	$V + N_P \rightarrow I$	(4)
Bdelloplast Maturation and Lysis:	$B \rightarrow Y_{P/B}P + S + Y_{M/B}M$	(5)
Infected Prey Maturation and Lysis:	$I \rightarrow Y_{V/I}V + Y_{M/I}M$	(6)
Sensitive Prey Growth:	$N_S + M \rightarrow 2\ N_S$	(7)
Phage-Resistant Prey Growth:	$N_R + M \rightarrow 2\ N_R$	(8)
Plastic Resistant Prey Growth:	$N_P + M \rightarrow 2\ N_S$	(9)
Phage Resistance:	$N_S + M \rightarrow N_S + N_R$	(10)
Phage Resistance Reversion:	$N_R + M \rightarrow N_R + N_S$	(11)
Plastic Resistance:	$N_S + S \rightarrow N_P$	(12)
Predator Mortality:	$P \rightarrow \text{ø}$	(13)
Phage Mortality (unused):	$V \rightarrow \text{ø}$	(14)

Note that the *Bdellovibrio* can attack sensitive (1) and phage-resistant prey (2) since phage resistance does not affect it, but it cannot attack plastic-resistant prey. Similarly, the bacteriophage can attack sensitive (3) and plastic-resistant prey (4) since plastic resistance does not affect it, but it cannot attack phage-resistant prey. For reactions (5) and (6), in addition to the predators and viruses released by the lysis, we assume a small amount of nutrients are released, which the prey can consume for limited growth with reactions (7), (8), and (9), though (9) results in two sensitive cells due to the growth-coupled reversion discussed previously. We represent $Y_{X/Y}$ as the “yield” or the number of X produced per particle of Y; though $Y_{P/B}$ is fixed to a value of 4.17 by experimental results (Fenton et al.), $Y_{V/I}$, $Y_{M/B}$, and $Y_{M/I}$ were adjusted to fit the data. Phage resistance arises from mutations during cell division in the sensitive prey, so we represent this as a growth reaction (10), instead of a direct conversion from sensitive to resistant. We also add (11) to represent the reverse of this mutation. Plastic resistance occurs when the prey receives a signal instead of through mutations, so we represent this by (12). However, we do not include a reaction or particle for an E. coli with both plastic and phage resistance, as this was not observed in the paper by Hobley et al. Predator mortality (13) was also included as it is observed in experimental results, but the phage mortality reaction (14) was not utilized in the implementation because phage mortality did not have a large enough effect within the 48-hour time span of the experiment to have statistical significance.

The Final Model

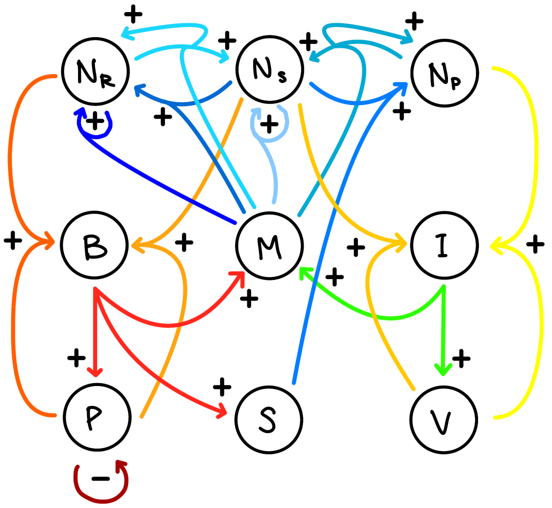

Figure 4. The interaction graph of the reactions in the final model

Initialization

Due to the large number of initial cells that would be needed to use the same scale from the experimental setup (2.9 × 10^8 CFU/mL of E. coli), simulating every individual prey and predator would be computationally intensive and impractical. Therefore, we scaled down the simulation to manageable levels while still capturing the essence of the system dynamics.

Thus, we employed two scales for our simulations:

Primary Scale (1/100,000): In this scale, we reduced the number of particles representing the organisms by a factor of 100,000. Thus, the scaled number of prey, predators, and viruses became 2900, 5, and 5, respectively. This reduced scale allowed us to quickly conduct tests and adjust parameters without the extremely long waiting times.

Secondary Scale (64/100,000): To account for higher concentrations and interactions, we scaled up the primary scale by a factor of 64 and the grid space (width and height) of the simulation by 8, resulting in 185,600 prey, 320 predators, and 320 viruses. This larger scale ensured that the initial number of resistant cells (just 1 resistant cell) was correctly scaled, unlike the smaller scale where the existence of a single resistant prey in the initial population of prey would be far overscaled compared to the actual experimental concentration of resistant cells.

Then for each scale, we conducted four sets of tests to match the four tests from the lab data:

1.	Dual Predation	E. coli + *bdellovibrio bacteriovorus* + bacteriophage
2.	Predator Only	E. coli + *bdellovibrio bacteriovorus*
3.	Virus Only	E. coli + bacteriophage
4.	Prey Only	E. coli

For all tests in the primary scale, we ran the simulation for 4,800 timesteps/frames (dt = 0.01) to account for all 48 hours of lab data, where 1 simulation time step = 0.01 hours = 36 seconds in real life. However for the secondary scale, we were unable to do all 48 hours due to time constraints; thus we limited this to 1,000 frames (10 hours of experimental data), which should be enough to give us an idea of what the simulation would do after, since the rest of the data is mostly flat.

Particle Movement

For simplicity's sake, we implemented a constant random movement model for the particles. Each particle, whether it represents prey, predators, or viruses, was assigned a random initial rate of movement (bounded by some constant) and direction. These particles continued to move in their respective directions until they encountered another particle and reacted, or self-reacted.

Upon a reaction event, the new product particles resulting from the reaction will start moving in random directions at random speeds, starting from the average of the positions of the two particles that reacted (or just from the parent particle for self-reactions). This process mimicked the dynamic movement of biological organisms within a fluid environment, such as a liquid culture in a laboratory setting. Though we did simulate different types of organisms having different movement speeds, this made no difference as we could change the probabilities of the reactions to adjust for this change. Thus, we proceeded with using the same speed bounds for all particles.

Particle Size

We assume all particles are circular, though we do consider the particle sizes in our simulation since they represent different organisms and molecules, which have different sizes in real life. These size values were adjusted to fit the data, but some parts were kept consistent. For example, the E. coli particles, including all three sensitivities and infected variants, should have the same size and the predators and viruses should be much smaller than the E. coli particles. Furthermore, signal and medium particles should be even smaller in size as they represent the different molecules in the system. (An extra benefit of making some particles smaller is that the smaller bounding boxes make computation time faster with our implementation, and since smaller particles usually exist in much larger quantities than the larger ones, this balances everything out.)

Particle Reactions

There are two types of reactions that can occur in our simulation: pair reactions and single reactions. Pair reactions involve interactions between two colliding particles, while single reactions occur within individual particles.

For pair reactions, we employ a collision detection algorithm in every frame to identify intersecting particles and run the corresponding reaction with a certain probability, which we adjust to fit the experimental data. Single reactions are executed on every particle in every frame with a certain probability, which we also adjust to fit the data. To prevent particles from trying to react again after failing to react in the previous frame, we implement a hard-coded cooldown period after each reaction that gives enough time for the particles to separate. Note that this cooldown time needs to be adjusted if we want to adjust the time step (dt) since it changes the distance each particle moves in a unit of time.

Collision Detection: QuadTrees

The naive approach in implementing the collision detection algorithm for the pair reactions would be to check every pair of particles for an intersection. However, since we would have to compare every one of the n particles to every other (n – 1) particle, this results in a time complexity of O(n^2) which becomes impractical for the large numbers of particles we will be using in the simulation. So, to efficiently handle collisions between particles, we use the QuadTree data structure, which offers a much more efficient solution by organizing particles into a hierarchical tree structure, reducing the complexity to O(nlogn).

In a QuadTree, the simulation space is recursively divided into quadrants until each quadrant contains at most N particles, where N is a hard-coded constant known as the bucket capacity (we found that ~30 worked the best for our case). So, when a particle is inserted into a QuadTree, it is recursively placed into the corresponding quadrant based on its position in the simulation space. If the leaf that the particle is inserted into is full, the space covered by that leaf is split into 4 smaller quadrants. Then, the particles originally stored in that leaf are placed into the smaller quadrants, and that leaf becomes an internal node (so it doesn't store any particles). This makes sure that we only create sub-quadrants as needed.

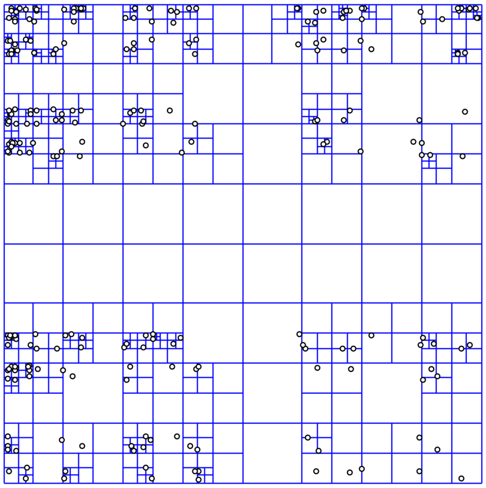

Figure 5. A diagram of the subsections created by a QuadTree. Note that N = 1 in this diagram, so each quadrant stores at most 1 particle. Source: https://en.wikipedia.org/wiki/Quadtree

The benefit of using this data structure is that it allows us to query for particles in a certain region in logarithmic time complexity. So, for each particle, we can query inside the square-shaped space around the particle with width and height = 2 (radius(particle) + max(radius of all particles)). Then, for each of the particles we find that isn't the original one, if the distance between it and the original particle is at most radius(original) + radius(other), then we can confirm that the two are intersecting. Since this significantly reduces the number of pairwise comparisons we do, this leads to improved computational efficiency, and we repeat this for every particle, we can search for all colliding particles in O(nlogn) time.

So, during the simulation, we insert each particle into a QuadTree and then query for all intersecting particles, and then run the reactions on them. In the next iteration, we create a new QuadTree and re-insert all particles after they move. Though this may seem very inefficient, creating a new QuadTree doesn't take much time compared to the actual collision detection, so although we did try an implementation that could handle particle movement inside the tree (and re-insertion if it moves out of the quadrant bounds), this didn't improve the speed by much. Thus, we decided to use the much simpler approach of creating a new tree every time. (We also used Cython to convert the Python code into C for just the QuadTree implementation to speed it up even more.)

Gray-Scott Simulation

While the particle simulation provides us insights into dual predation, it also has limitations, particularly in scalability. Simulating every individual cell present in the experimental setup is computationally intensive and impractical. To address this, we scaled down the simulation to manageable levels while still capturing the essential parts of the system. An alternative approach to this problem is to instead use differential equations that represent the reactions, and approximate the reactions that occur by solving the equations. However, since the differential equations only give up the total population and assume all particles are evenly spread out, we would like to include aspects of spatial distributions of concentrations of the different particles that the particle simulation allows us to do. Thus, we can use the Gray-Scott Model to allow us to do this.

Concentration Grid

In the Gray-Scott Simulation, the space is discretized into a grid of cells, where each cell represents a small volume of the experimental environment with its own concentrations, and we track these concentrations at each grid point over time. This simulation approximates this system based on the idea that within each cell, the organisms are well-mixed and interact homogeneously, so all particles in the cell interact with each other and react. While this may not fully capture the complexity of the actual biological system, it provides a computationally efficient way to model the interactions between the organisms and observe their dynamics over time. By discretizing the space into grid cells, we can simulate the movement and interactions of the organisms at a macroscopic level, allowing us to observe emergent patterns without having to track the movements of individual particles.

Initialization

Each cell has an initial concentration of E. coli with an upper bound of 2.9×10^8 CFU/mL as defined in the experimental results with added simplex noise to add variation. We add 5.0×10^6 PFU/mL of *Bdellovibrio* *bacteriovorus* in the dual predation and predator-only predation models and 5.0×10^6 PFU/mL of bacteriophage in the dual predation and virus-only predation models in a similar method (with simplex noise) as we do with the E. coli.

Reaction Phase

This model consists of two main phases that alternate in each frame: the reaction phase, where chemical reactions occur, and the diffusion phase, where molecules diffuse through the environment.

In the reaction phase, every cell in the grid uses the particle concentrations within itself to calculate the rates of chemical reactions that occur with the differential equations stated previously. The differential equations and constant values come from the paper by Hobley et al.

Differential Equations

$$\frac{dM}{dt} = -\frac{\mu_N M\,(N_S + N_P + N_R)}{(K_{M,N} + M)Y_{N/M}} + Y_{M/P}k_pB + Y_{M/V}k_VI$$

$$\frac{dN_S}{dt} = N_S\frac{\mu_N M}{K_{M,N} + M} + N_P\frac{\mu_N M}{K_{M,N} + M} + P\frac{\mu_P N_S}{(K_{N,P} + N_S + N_P + N_R)Y_{B/N}} - V\frac{\mu_V N_S}{Y_{I/N}} - k_DSN_S - k_MN_S$$

$$\frac{dN_P}{dt} = -V\frac{\mu_V N_P}{Y_{I/P}} + k_DSN_S \; \frac{dN_R}{dt} = N_R\frac{\mu_N M}{K_{M,N} + M} - P\frac{\mu_P N_R}{(K_{N,P} + N_S + N_P + N_R)Y_{B/N}} + k_MN_S$$

$$\frac{dP}{dt} = k_PB - mP - P\frac{\mu_P N_R}{(K_{N,P} + N_S + N_P + N_R)Y_{B/P}} - P\frac{\mu_P N_S}{(K_{N,P} + N_S + N_P + N_R)Y_{B/P}}$$

$$\frac{dB}{dt} = -\frac{k_PB}{Y_{P/B}} + P\frac{\mu_P N_R}{K_{N,P} + N_S + N_P + N_R} + P\frac{\mu_P N_S}{K_{N,P} + N_S + N_P + N_R}$$

$$\frac{dI}{dt} = -\frac{k_VI}{Y_{V/I}} + V\mu_V N_P + V\mu_V N_S$$

$$\frac{dV}{dt} = k_VI - V\frac{\mu_V N_P}{Y_{I/V}} - V\frac{\mu_V N_S}{Y_{I/V}}$$

$$\frac{dS}{dt} = \frac{k_PB}{Y_{P/B}}$$

Note that the particle abbreviations used in these equations are the same as the ones from the particle simulation. Furthermore, we will call just solving these differential equations (without Gray Scott) with initial concentration values as defined in the experimental results as the Differential Equation Model, which we use to compare with our simulation in our results. In their paper, Hobley et al. used a Sequential Monte Carlo method to fit the model parameters to the data.

Diffusion Phase

After simulating the reactions in the reaction phase, the system transitions to the diffusion phase, where the concentrations diffuse through the environment nearby, which allows the particles to interact with each other across the entire simulated space, influencing the dynamics of the system as a whole. The rates at which the concentrations diffuse are based on the diffusion equation,

$$\frac{\partial C(x,y,t)}{\partial t} = D\nabla^2 C(x, y, t) = D\left(\frac{\partial^2 C(x,y,t)}{\partial x^2} + \frac{\partial^2 C(x,y,t)}{\partial y^2}\right)$$

where D is the diffusion constant and C(x,y,t) is the function for the concentration of the cell at (x,y) at time t. So, we calculate the sum of the second derivatives of the concentrations of the grid in the x and y directions multiplied by the constant to get the rate of change for the concentrations each cell should have in the diffusion phase.

3. Results and Discussion

Differential Equation Graphs

Solving the Differential Equations

First, we used the equations and parameters provided by Hobley et al. to solve for the fitted model which we use as the basis for comparison with our other simulations. To do this, we first used a hand-written piece of code that solves the equations using Newton's method, but later decided to instead use a library that uses the Runge-Kutta 4 method to solve them more quickly. We also created a model with adjustable parameters with real-time visual feedback to see how different parameters affect the graph, though we did not include these in the results as they were just for our understanding. However, the code for the graphs with adjustable parameters is included in the Code Availability section.

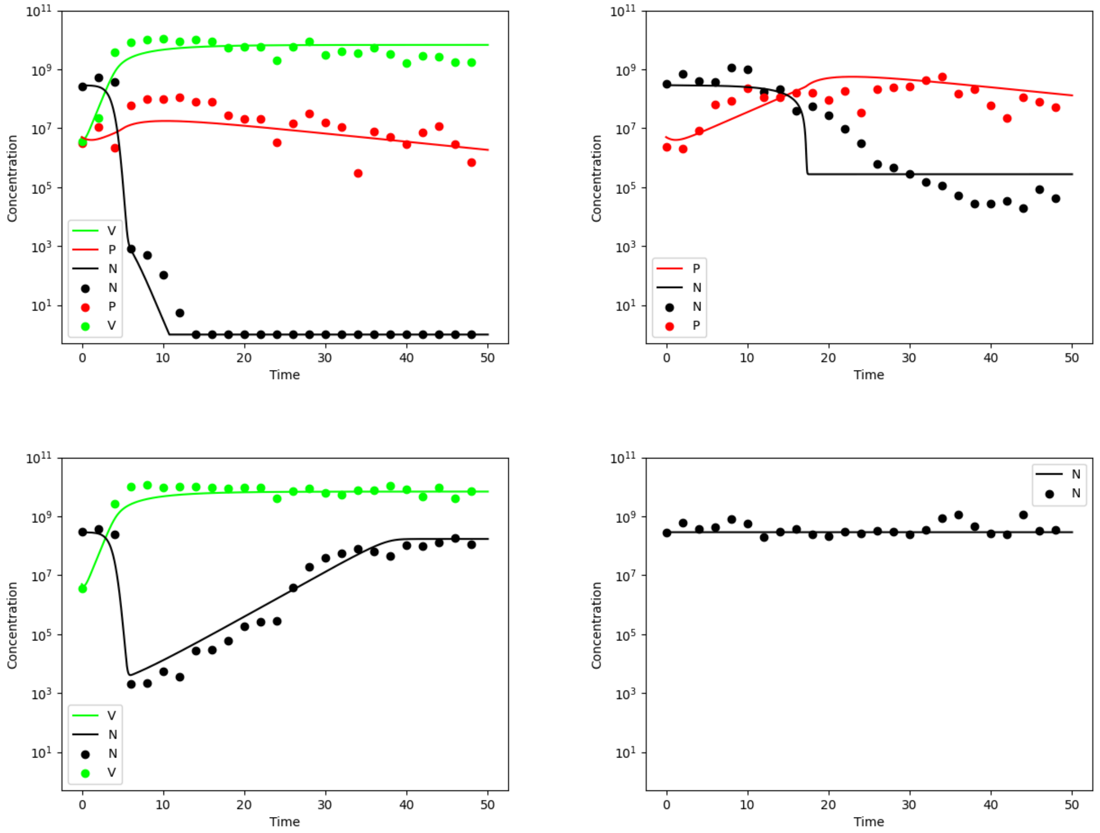

Figure 6. Solved differential equations from the results of the paper by Hobley et al. in fitting their differential equations to the experimental data; we use these graphs as a basis for comparison for our simulations, in addition to the data itself.

Particle Simulation Results

Primary Scale Results

For all simulations, the populations of each type of particle were saved and plotted with the results of the differential equation model and experimental data.

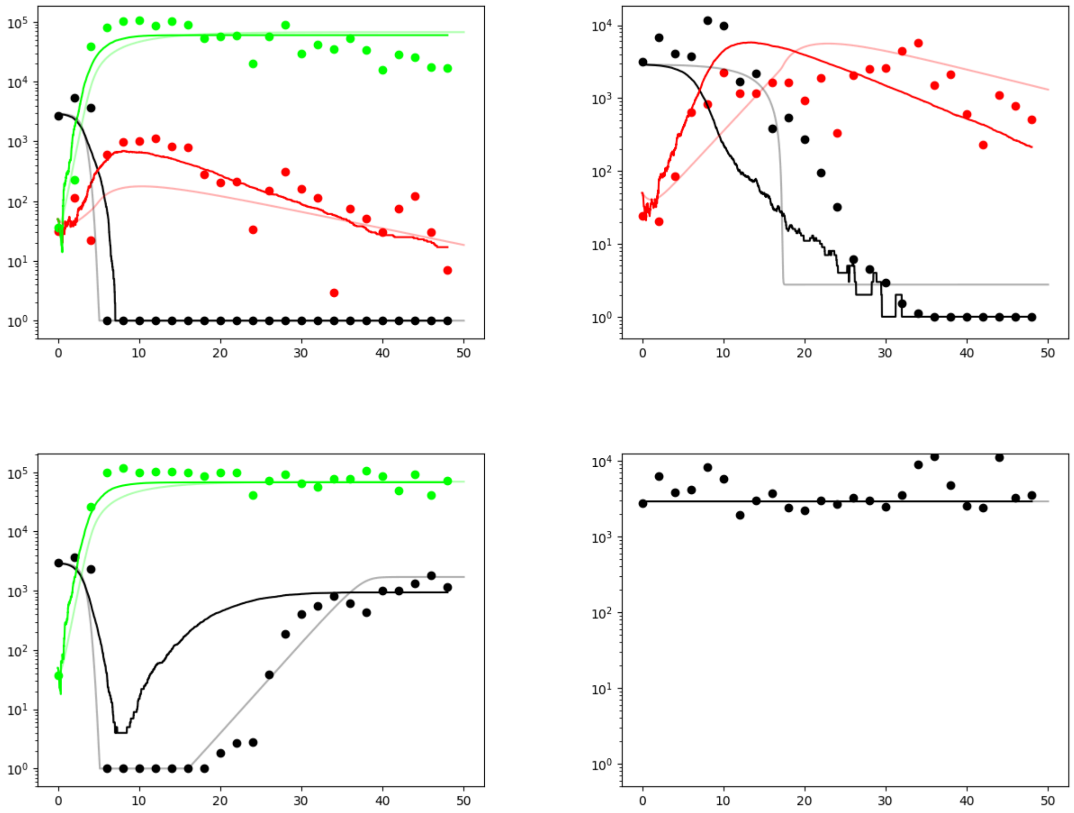

Figure 7. A visual, log-scale comparison of the primary scale particle simulation and differential equation model with the experimental results. For all graphs, black represents the total population of E. coli, red represents the population of *bdellovibrio bacteriovorus*, and green represents the population of bacteriophages. The darker-colored lines represent the particle simulation results, and the lighter-colored lines represent the differential equation results. Top left is the simulation of dual predation; top right is single predation by *bdellovibrio bacteriovorus*; bottom left is single predation by bacteriophages; bottom right is no predation. Note that the differential equation and experimental data were scaled down to the primary scale with the particle simulation. The dual predation and virus-only predation simulation took ~30 minutes, and the predator-only predation and no predation simulations took ~10 min.

We see that in the dual predation model, the particle simulation was able to exterminate the E. coli and the populations of the other two predators matched the experimental data well, even better than the results from the differential equation model. For the *Bdellovibrio bacteriovorus*-only simulation, we see that the differential equations for the prey population decrease far too quickly at around the ~17-hour point. Though we did try to play around with the fit parameters given to us by the original paper, we found that none of the parameters actually adjusted this rapid decrease in population; thus we assume that the differential equations miss an aspect that keeps the E. coli population up in the experimental data. In the particle simulation data, however, we see that although the sensitive population quickly decreases similarly to the differential equation version, there is a significant temporary increase in the plastic-resistant E. coli population that increased the total population of E. coli to match more closely to the data, as we can see in the graph. It's also worth noting that we see that the E. coli population does reach 0 in the simulation, which shouldn't occur. However, we also see this in the experimental data since we scaled everything down by the primary scale, so technically our simulation does match better with the data than the differential equation does. This is because the actual concentration of remaining E. coli is much lower than the amount we scaled down by, so we would need many more particles to see this effect.

In the bacteriophage-only experiment, we see that the differential equations do match the data much better for the E. coli population. However, this again is caused by the limitations of scale. As we previously stated, even a single resistant E. coli particle in the initial population is very overscaled; however, including no initial resistant strain would mean that we would have to rely on the random (very low) chance of a mutation occurring that changes a sensitive prey into a resistant one. Thus, we decided to keep exactly one resistant prey, which allowed the population of E. coli to increase too soon, since as the population of sensitive prey decreases significantly, the number of medium particles that the resistant one can interact with increases. If there were more particles and the resistant strain were correctly scaled, then we would see that the population would have to decrease to the same point to see an increase in the resistant population, meaning a later population increase (to match the data better). For the virus population itself, we see that both the differential equations model and particle simulation models do very good. However, we also saw that this population usually fit very well even if we adjusted the probability parameters by a significant amount, so we assume this pattern is caused by the nature of how the virus works producing a huge number of viruses per lysis.

Though the prey-only test is not very interesting, it did confirm to us that the prey population does stay stable if no other predators exist, which we would expect.

Secondary Scale

For our secondary scale tests, because we were unable to run the simulation for all 48 hours of experimental data (adjusting the dt would increase inaccuracies and affect the probabilities, as stated before), we can only compare the first 1000 frames or 10 hours of experimental data for the dual predation and bacteriophage single predation tests. However, we were able to simulate the *Bdellovibrio bacteriovorus* only and no predation tests for the full 4800 frames as they don't have the bacteriophages which increases the computation times significantly. Nevertheless, we were able to see a much better match to the data that we were unable to get from just the primary scale due to scaling limitations:

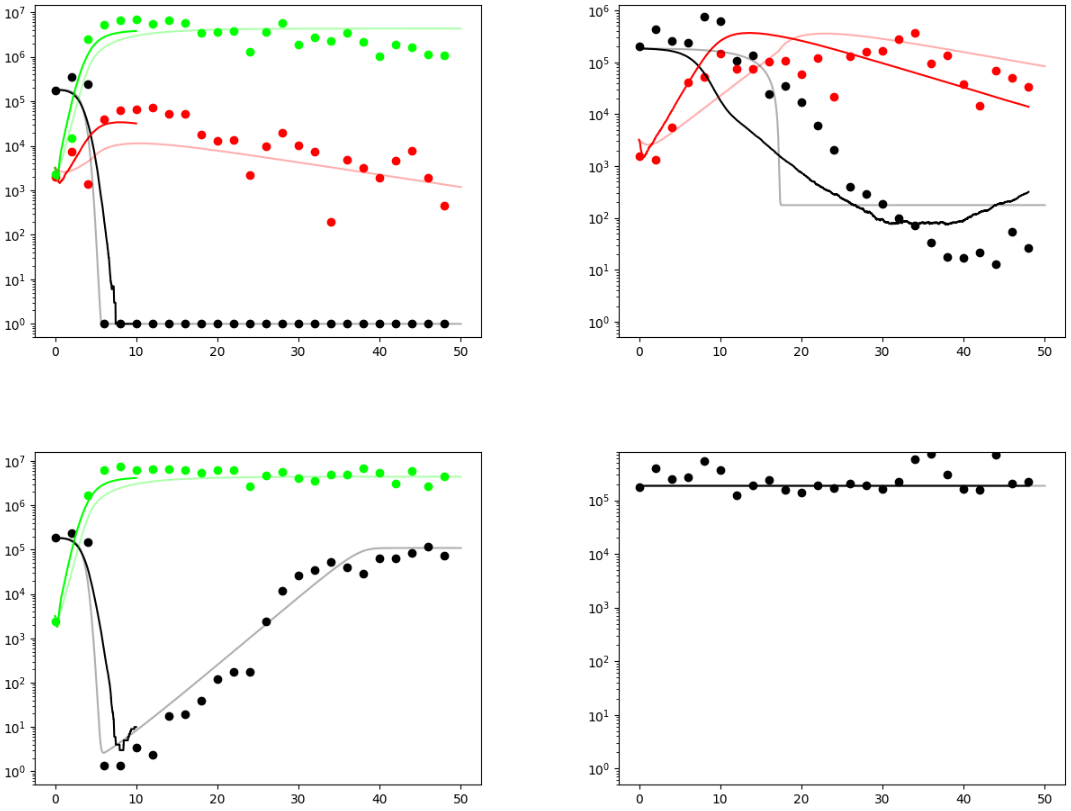

Figure 8. A visual, log-scale comparison of the secondary scale particle simulation and differential equation model with the experimental results. For all graphs, black represents the total population of E. coli, red represents the population of *Bdellovibrio bacteriovorus*, and green represents the population of bacteriophages. The darker-colored lines represent the particle simulation results, and the lighter-colored lines represent the differential equation results. Top left is the simulation of dual predation; top right is single predation by *Bdellovibrio bacteriovorus*; bottom left is single predation by bacteriophages; bottom right is no predation. Note that the differential equation and experimental data were scaled down to the secondary scale with the particle simulation and that the dual predation and bacteriophage predation tests were stopped at 1000 frames, but the other two were able to be fully simulated for 4,800 frames. The 1,000 frames for dual predation and virus-only predation simulation both took ~6 hours, and the full 4,800 frames for predator-only predation and no predation simulations both took ~2.5 hours.

In the dual predation test, we see that the populations for all 3 organisms were similar to the one from the primary scale; this is likely due to the fact that the limitations of scale not existing in this model, since the prey gets killed off too quickly for it to become an issue. Either way, we do see that the virus and predator populations match the data better than the differential equation versions, and though the match for the prey population is worse, we still see that it correctly decreases rapidly near where that occurred in the experimental data. For the single predation by *Bdellovibrio bacteriovorus* test, we see that the predator population is similar to that of the primary scale simulation. However, we do see that the prey population correctly increased back up, unlike with the primary scale. As stated before, this is because of the increased scale which prevents the prey population from going to zero before the population of the predators can decrease enough for the prey population to increase back up.

For the bacteriophage simulation, we again see that the population of viruses matches the data very well, similar to the differential equation model. However, we now see that the population of prey matches well with the data unlike that from the primary scale version, which was what we expected to see, as stated before. Since we do see that the population starts increasing slightly early, this again would likely be fixed if the scale were increased further. In the prey-only test, again everything stayed constant as expected.

Particle Simulation Visualization

Furthermore, we do have the rendered videos of each of the particle simulation tests we conducted. Here are two frames from the two scales of particle simulations. Full videos are provided in the Data Availability section.

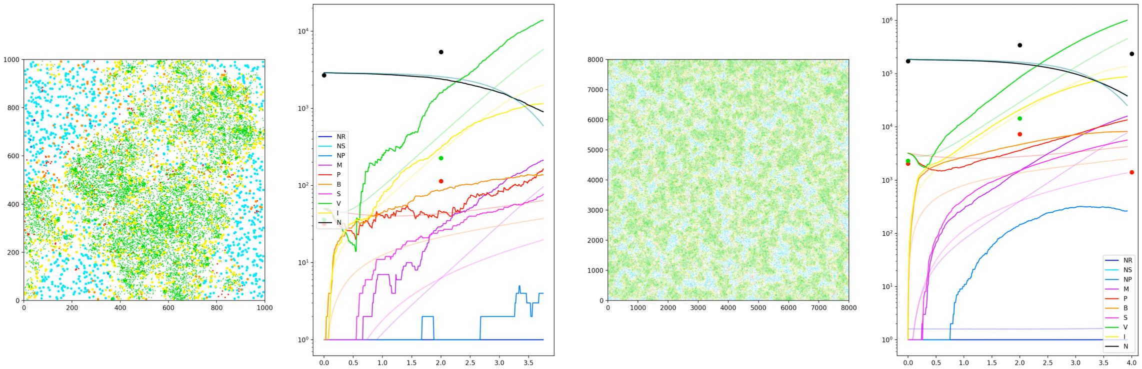

Figure 10. Frames from the visualization of the particles in the two scales of simulation. Left is primary scale; right is secondary scale. The timestamp for both is at ~4 hours.

Parameters

To fit the data, we adjusted several parameters included in the model, which include reaction probabilities, yield values, and particle size. These parameters are as such:

Reaction Probabilities

(1)	Bdellovibrio Attacks Sensitive E. coli:	0.6
(2)	Bdellovibrio Attacks Phage-Resistant E. coli:	0.6
(3)	Phage Attacks Sensitive E. coli:	0.8
(4)	Phage Attacks Plastic-Resistant E. coli:	0.8
(5)	Bdelloplast Maturation and Lysis:	0.004
(6)	Infected Prey Maturation and Lysis:	0.005
(7)	Sensitive Prey Growth:	0.99999
(8)	Phage-Resistant Prey Growth:	0.99999
(9)	Plastic Resistant Prey Growth:	1
(10)	Phage Resistance:	0.00001
(11)	Phage Resistance Reversion:	0.00001
(12)	Plastic Resistance:	0.1
(13)	Predator Mortality:	0.0011

Particle Size	
Sensitive E. coli (N_S):	5
Bacteriophage Resistant E. coli (N_R):	5
Plastic-Resistant E. coli (N_P):	5
Bdellovibrio bacteriovorus (P):	3
Bdelloplast (B):	5
Bacteriophage (V):	2
Phage-Infected Prey (I):	5
Medium (M):	1
Signal (S):	1

Yield Values	
$Y_{P/B}$ (fixed):	4.17
$Y_{V/I}$:	24
$Y_{M/B}$:	0.3
$Y_{M/I}$:	0.35

Gray-Scott Simulation Results

In our Gray-Scott simulation, we averaged all concentrations in each cell to give us the total population to compare the results with the data. Here are the graphs produced from this compared to the differential equation model and experimental data.

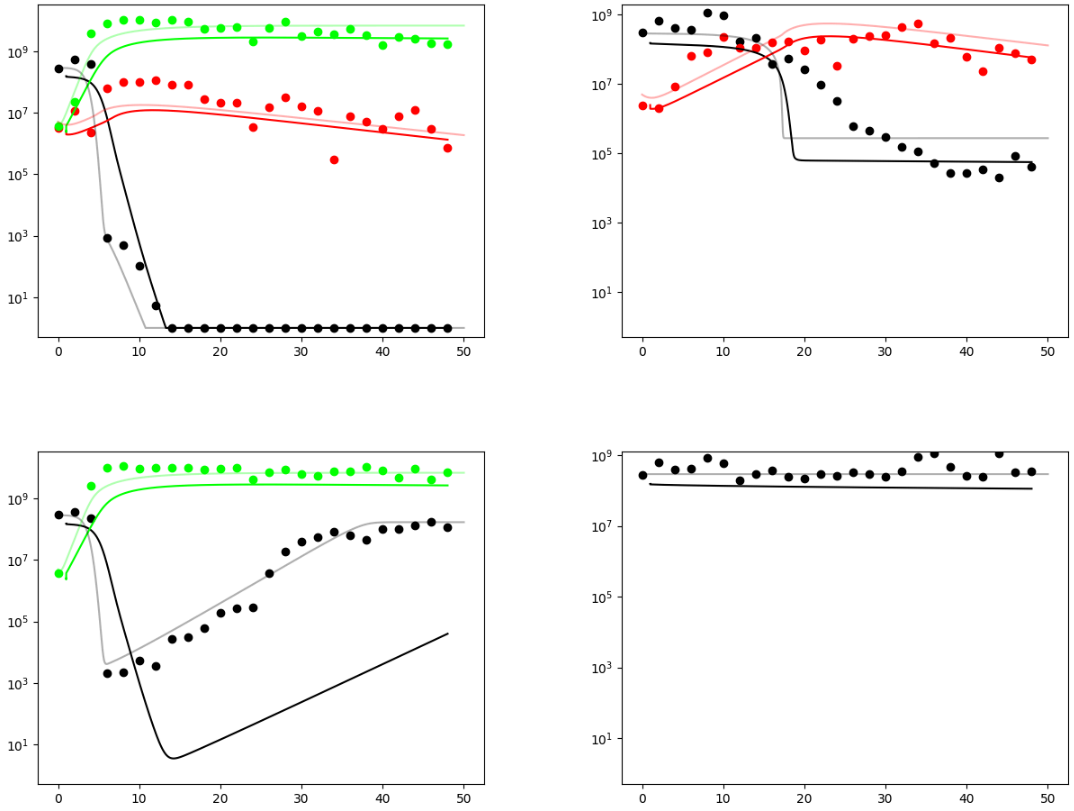

Figure 11. A visual, log-scale comparison of the Gray-Scott model and differential equation model with the experimental results. For all graphs, black represents the total population of E. coli, red represents the population of *Bdellovibrio bacteriovorus*, and green represents the population of bacteriophages. The darker-colored lines represent the particle simulation results, and the lighter-colored lines represent the differential equation results. Top left is the simulation of dual predation; top right is single predation by *Bdellovibrio bacteriovorus*; bottom left is single

predation by bacteriophages; bottom right is no predation. Note that the data from the Gray-Scott model represents the average concentrations of each particle in the cells.

We can see from the graphs that the data closely resembles the differential equation model, as expected since the model is based on them. However, while the model captured the general trends, it seems to perform worse except with the population of prey in the dual predation test, which is likely due to it being able to average and smooth out the rapid drop in population. The larger errors are likely due to the concentration inconsistencies due to the noise generation, which we added to account for differences in concentrations in the original experimental sample and to make sure that organisms in one area of can only interact with the organisms nearby. However, it seems like the model ended up averaging out all the concentrations in the cells throughout the system due to diffusion, even if the diffusion constant is set to be very low (in our experiment, we ended up with D = 1, as it seemed to make not much of a difference).

Gray-Scott Visualization

Similar to the particle simulation, we also have animations of the Gray-Scott simulations. However, they ended up not being as interesting as the particle simulations, since the diffusion caused all the concentrations to even out. Nonetheless, here are two frames from simulations that were somewhat interesting, where the majority population is changing from E. coli to one of the predators:

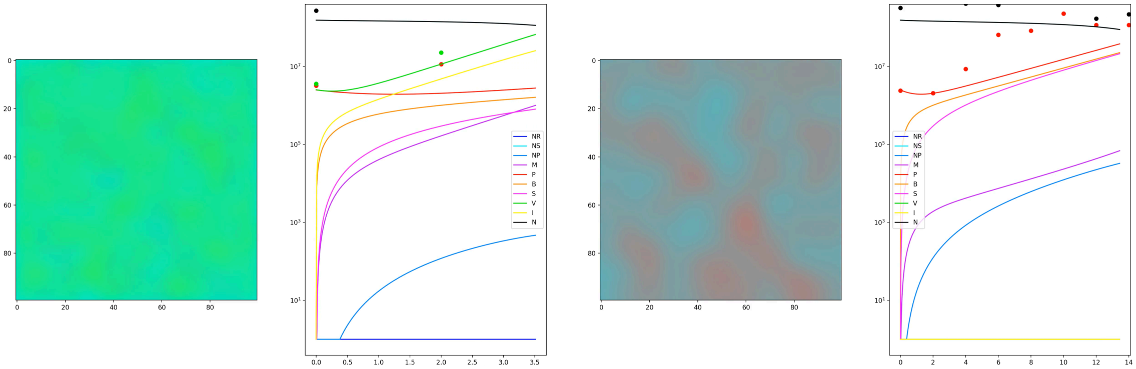

Figure 12. Two interesting frames from the Gray-Scott simulation visualizations. Left: dual predation at ~3.5 hours; Right: single predation by *Bdellovibrio bacteriovorus* at ~14 hours.

Numerical Evaluation

To score our models numerically (instead of just pure visual comparison), we used a distance function provided by Hobley et al., which they used to fit their parameters to the experimental data. So, we calculate the distance by the formula:

$$Average\ Distance\ =\ \frac{\sqrt{sum[(point\ distance)^2]}}{number\ of\ data\ points}$$

where the point distance is calculated by:

$$point\ distance\ =\ log(simulation\ datum)\ -\ log(experimental\ datum)$$

For each of the simulation tests and the differential equation model results, we evaluated the distance with the data. Note that for the primary scale particle simulation and Gray-Scott model, we were able to use all

48 hours of data points from the lab results for calculating the average distance, but for the secondary scale model, only the first 10 hours of data were compared for the dual predation and virus predation tests; the predator and no-predation tests used all 48 hours as we were able to simulate the entire 4800 frames. Furthermore, the distance for the differential equation model changes for the two scales of particle simulation and the Gray-Scott model due to scaling and clipping (the minimum to 1 to prevent log 0 errors), which affects the final score.

	Primary Scale Particle Simulation				Differential Equation Model			
Particle	Dual	Predator	Virus	None	Dual	Predator	Virus	None
N	0.08215	0.15619	0.26022	0.04955	0.04479	0.15755	0.06700	0.04957
P	0.08699	0.09252	–	–	0.10770	0.10118	–	–
V	0.06791	–	0.03203	–	0.07349	–	0.04799	–

	Secondary Scale Particle Simulation				Differential Equation Model			
Particle	Dual	Predator	Virus	None	Dual	Predator	Virus	None
N	0.42919	0.17484	0.46339	0.04957	0.18664	0.17794	0.18828	0.04957
P	0.19126	0.09231	–	–	0.23953	0.10118	–	–
V	0.16310	–	0.08701	–	0.18189	–	0.16500	–

	Gray-Scott Simulation				Differential Equation Model			
Particle	Dual	Predator	Virus	None	Dual	Predator	Virus	None
N	0.18320	0.19119	0.76015	0.10629	0.08929	0.17794	0.08488	0.04957
P	0.13004	0.10046	–	–	0.10770	0.10118	–	–
V	0.10552	–	0.11777	–	0.07349	–	0.04799	–

Figure 9. Tables of distance scores for each particle type in each test for the two scales of particle simulations and the Gray-Scott model, with the differential equation model. The top table compares the primary scale particle simulation with the differential equations; the middle compares the secondary scale simulation with the differential equations; the bottom compares the average cell concentrations in the Gray-Scott model with the differential equations. Note that a smaller distance is a better fit for the data.

Something to note is that for the dual predation and the virus-only tests for the secondary scale particle simulation, we see that the distance score is very high; this is because we are calculating the average distance, and since we have a high error at time = 8 hours when the rapid drop in E. coli population occurs, which usually is averaged out by the entire 48 hour time span, but now is only spread out to the 10-hour limited time span. The scores for the Gray-Scott simulation are also very high, though this is because of a bad fit to the data. Furthermore, since we can visually see these errors in the graph, these values don’t give us any new information—we provided these just for completeness.

4. Conclusion

The key takeaway from our research is as follows: by taking advantage of the dual predators in the system, we were able to observe complete and total lysis of our E. coli population within a reasonable time frame of 48 hours. Observing the data more closely reveals that this is no coincidence. As both predators proceed with their proliferation as usual, the prey is able to form resistance to their respective aggressors. However, at this point, we theorize that the subsequent generation of prey is still susceptible to the alternate predator, and is consumed regardless. By overlaying the results of both the single predation trials, we arrive at the same conclusion as our dual predation trial, implying there are no significant interactions between the predators influencing the result. As a result, we can conclude that simply the presence of an alternate predator is an effective combatant to individual resistance.

This conclusion opens up many options for potential therapeutics, with further investment and development, taking advantage of dual predation could lead to very robust methods of antibacterial treatment. As discussed earlier, the results of this trial suggest no special interaction between our selected bacteriophage and *B. bacteriovorus*, a natural next step would be trials with other bacteria, and their corresponding bacteriophages. Verifying this would allow us to develop new treatments for bacterial infections that are, ironically, resistant to resistance. That said, only further trials can confirm the effectiveness of this theory and any potential drawbacks that may arise with its implementation.

5. Data and Code Availability

Data

The data, graphs, and simulation videos produced from our simulations, including those from the experimental results and solved differential equations are located here. Note that the experimental data included come from the paper by Hobley et al.

Code

The Python code used for our implementations of the above simulations is located here.

6. References

Hasan, Mahadi, and Juhee Ahn. “Evolutionary Dynamics between Phages and Bacteria as a Possible Approach for Designing Effective Phage Therapies against Antibiotic-Resistant Bacteria.” *Antibiotics*, vol. 11, no. 7, July 2022, p. 915. https://doi.org/10.3390/antibiotics11070915.

Hatfull, Graham F., et al. “Phage therapy for Antibiotic-Resistant bacterial infections.” *Annual Review of Medicine*, vol. 73, no. 1, Jan. 2022, pp. 197–211. https://doi.org/10.1146/annurev-med-080219-122208.

Hobley, Laura, et al. “Dual Predation by Bacteriophage and Bdellovibrio bacteriovorus Can Eradicate Escherichia coli Prey in Situations where Single Predation Cannot.” *Journal of Bacteriology*, vol. 202, no. 6, Feb. 2020, https://doi.org/10.1128/jb.00629-19.

Lovering, Andrew L., and R. Elizabeth Sockett. “Microbe Profile: Bdellovibrio bacteriovorus: a specialized bacterial predator of bacteria.” *Microbiology*, vol. 167, no. 4, Apr. 2021, https://doi.org/10.1099/mic.0.001043.

World Health Organization: WHO. *Antimicrobial resistance*. 23 July 2019, www.who.int/health-topics/antimicrobial-resistance#:~:text=AMR%20occurs%20when%20bacteria%2C%20viruses,spread%2C%20severe%20illness%20and%20death.

Yair, Shemesh, and Édouard Jurkevitch. “Plastic phenotypic resistance to predation by Bdellovibrio and like organisms in bacterial prey.” *Environmental Microbiology*, vol. 6, no. 1, Dec. 2003, pp. 12–18. https://doi.org/10.1046/j.1462-2920.2003.00530.x.