University of Amsterdam

Masters Thesis

Mathematically Modeling the Interactions Between Phages, Bacteria, and the Environment

Examiner:
Dr. Jaap Kaandorp

Author:
Victor Piaskowski

Supervisor:
Dr. Matti Gralka

Assessor:
Dr. Yuval Mulla

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computational Science

in the

Computational Science Lab
Informatics Institute

June 2025

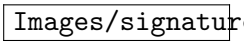
clslogo.png

Declaration of Authorship

I, Victor Piaskowski, declare that this thesis, entitled 'Mathematically Modeling the Interactions Between Phages, Bacteria, and the Environment' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at the University of Amsterdam.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:



Date: June 26, 2025

"All models are wrong, but some are useful"

George E. P. Box

UNIVERSITY OF AMSTERDAM

Abstract

Faculty of Science
Informatics Institute

Master of Science in Computational Science

Mathematically Modeling the Interactions Between Phages, Bacteria, and the Environment

by Victor Piaskowski

In any given microbial ecosystem, hundreds of different phages and bacteria interact with one another in a complex system. These individual interactions have a significant impact on the bacterial community and the overall ecosystem, with important ecological consequences. They force bacteria to mutate and evolve, changing the community's fitness level. Phages can control bacterial populations, ensuring that dangerous bacteria do not over-reproduce and harm the ecosystem. Phages control bacterial population dynamics using the "kill-the-winner" dynamic, preventing any one species from dominating the population. Upon lysis, bacteria release resources into the environment that other bacteria and plants can use.

There is a need to model complex systems mathematically; however, current models typically only model one or two phages and bacteria. It is, however, straightforward to extend the model to multiple species. Mathematically modeling large bacterial communities is important because running laboratory experiments is time-consuming and costly. These simulations can guide future lab work, for example, in combating antibiotic-resistant bacteria.

I created a program consisting of three parts that can model complex communities of p phages, b bacteria, and r resources. The first part uses a GUI tool to create and edit the interactions between the phages, bacteria, and resources using a graph network. The nodes and edges hold the interaction and environmental parameter values. The second part, the simulation framework itself, models and calculates the population levels of phage, bacteria, and resources using the user-provided network and ODE model. The third part allows the user to interact with the simulation framework with a custom-built dashboard. The user can, for example, modify the parameter values from the dashboard and immediately observe the impact of the change on the simulation. There are five pre-built plots that the user can interact with. It is possible to download the simulation results, allowing users to create custom analyses and plots.

I apply this process to the Golding model from Geng et al. [1]. I qualitatively analyze the model's behavior and use a Sobol analysis to quantitatively assign a value to the model's variance in output based on its input. Additionally, understanding the various restriction regimes in phages, as well as the factors that drive reaction rates and determine the rate-limiting steps, is crucial. By varying the initial population of uninfected bacteria, the model can interpolate between two limiting scenarios: adsorption and latency. It is essential to determine whether a phage population will proliferate in a laboratory setting under constant washout conditions to prevent the phage population from being inadvertently wiped out. I analyze whether phages will proliferate or die out, depending on the initial condition. Finally, I perform and analyze a co-existence analysis on a large 20-phage, 20-bacteria, and 10-resource community to determine whether phages and

bacteria can coexist under various conditions. In large communities, phages and bacteria can coexist. If a debris term is introduced, bacterial defenses randomly deactivate phage, increasing the bacteria's survivability rate.

Acknowledgements

I want to thank my parents for loving me despite some of my faults and for supporting me throughout my Bachelor's and Master's studies. Without them, I would not be where I am today, and I would certainly be a different person if it were not for them.

Thank you to Dr. Matti Gralka for the weekly meetings and for teaching me everything about phages and bacteria. Every meeting was always insightful, productive, engaging, and informative. I will forever be amazed at how he can recall which paper discusses which topic and how he always had a paper for every topic.

Thank you to Sofia Blaszczyk for finding the Master thesis opening and suggesting that I email Dr. Gralka for an introductory meeting. She acted as my rubber duck programming buddy and watched the cringe-worthy screen recordings that I sent her at 2 am, showcasing various demos of my project. If she had not found this opening, I would not know what I would be doing for my thesis.

If I had not followed Dr. Rik Kaasschieter's and Dr. Martijn Anthonissen's courses "Introduction Computational Sciences "and "Numerical Linear Algebra "in my Bachelor's, I would not have been interested in Computational Sciences. I would not have found the MSc Computational Sciences program, as Computational Sciences fits my interests and skill sets better than any other program I could have taken. Rik and Martijn have forever altered my career trajectory.

Thank you to Sarah Flickinger for showing me the research she has been working on in the lab. She allowed me to connect my research and models to real life, reminding me that what I am doing has real-life applications rather than just being a purely theoretical or programming challenge.

Moreover, I want to thank my friends for keeping me sane and supporting me throughout both of my programs.

Contents

List of Figures

List of Tables

Abbreviations

DDE	**D**elay **D**ifferential **E**quation
DNA	**D**eoxyribo**N**ucleic **A**cid
GUI	**G**raphical **U**ser **I**nterface
IC	**I**nitial **C**ondition
IVA	**I**nitial **V**alue **A**nalysis
OD	**O**ptical **D**ensity
ODE	**O**rdinary **D**ifferential **E**quation
PA	**P**arameter **A**nalysis
PDE	**P**artial **D**ifferential **E**quation
RNA	**R**ibo**N**ucleic **A**cid
ST	**S**erial **T**ransfer
UA	**U**ltimate **A**nalysis
UvA	**U**niversitiet **v**an **A**msterdam

Chapter 1

Introduction

Phages are small viruses, measuring 27-190 nm in diameter, that infect and lyse (kill) specific bacteria. Phages act as nature's antimicrobial defense, but they also impact bacterial evolution and resource turnover. There are various medical and industrial applications for phages to control bacterial growth; however, to realize these applications, it is essential to understand how phages interact with bacteria, enabling the implementation of a robust method to control bacterial growth.

1.1 Thesis Overview

To answer the question of how phages, bacteria, resources, and their interactions can be mathematically modeled, I developed a simulation framework that can model and interact with any $p \times b \times r$ system, where p represents the number of phages, b represents the number of bacteria and r represents the number of resources. Using the software, I answer how phages impact community dynamics in complex microbial communities.

First, there is a biological introduction to phages and bacteria. This introduction covers how phages infect bacteria, how bacteria defend against phages, how phages defeat bacterial defenses, and how phages defend against other phages. There is an introduction to different methods of modeling phage and bacteria dynamics. This thesis provides a brief overview of software that models phages, resources, bacteria, and their associated limitations. This thesis presents software I developed to support the research, demonstrating its capabilities using a representative model of phage, bacteria, and resource interactions. The section also provides an overview of its usage, including example outputs from demonstration runs. I use the software to analyze various scenarios, such as phage proliferation under a washout scenario, and analyze growth rate-limited regions.

1.2 The Environment

In ecosystems like the ocean, the gut, or soil, there are thousands of different microbes that interact with one another or with their surrounding environment. The interactions are complex, with many factors affecting the growth of bacteria and phages. The interactions between entities in the environment are often synergistic. When an animal dies, bacteria begin to digest and decompose the animal into simpler chemicals, such as carbon and nitrogen, which plants can use to grow. Animals then eat these plants, and the cycle continues. External factors, such as flooding, droughts, chemical spills, or the introduction of new entities, have a massive impact on the ecosystem. These events can add or remove resources from the system, alter environmental parameters such as the temperature and acidity of the soil, introduce competition, or create a population imbalance by killing microorganisms. These effects have a larger effect on the ecosystem and food chain as a whole, as bacteria are one of the fundamental foundations for resource recycling.

1.3 Biological Background

Phages are small viruses on the order of 27-190 nm (the average size of marine phages is 54 nm) that infect and lyse (kill) specific bacteria. The phage cycle process starts with a phage coming into contact with a bacterium Figure B.3. Once it has identified an injection site, the phage can inject a strain of DNA into the bacteria. The DNA strand has two options.

The first option is that the phage can enter the lysogenic cycle by integrating its DNA into the host cell's DNA. Prophages are phages that have integrated with the host cell's DNA. The new cell will contain the prophage's DNA as the bacteria replicate. The phage will exit the lysogenic cycle and enter the lytic cycle after receiving a signal, such as cell damage, and hijack the DNA replication mechanism.

The second option is that the phage immediately enters the lytic cycle by hijacking the DNA replication process to create copies of its DNA. The phage will create multiple copies of itself using the host's transcription and translation machinery to replicate itself. The phages self-assemble inside the bacteria until they lyse the cell (e.g., through chemically induced rupture of the cell wall), releasing the phages into the environment.

Figure 1.1: Life cycle of a phage, inside and outside a bacteria cell. Significant steps in the life cycle of a phage include the infection stage, integration, replication, and lysing process. Figure sourced from Campbell [2].

1.3.1 Phage's Role in the Environment

Phages play a significant role in the ecosystem. Phage-mediated death occurs when phages infect susceptible bacteria, resulting in cell lysis and the release of cellular contents and nutrients that other organisms, such as bacteria and plants, can utilize. This process not only reduces bacterial populations but also accelerates the turnover of resources, such as nitrogen and carbon, for other bacteria and plants to utilize. Phages also help mediate horizontal gene transfer, disperse pathogenic diseases, and spread antibiotic resistance [8]. Phages directly alter bacteria population diversity and population fitness by introducing new ways for bacteria to mutate [9].

There are about 10^6 bacteria cells/ml and 10^7 phages/ml of marine water. About 5% of any bacteria are currently infected, and phages account for about 15% of daily bacterial deaths [10]. Phage populations grow by infecting their hosts, but they can also be degraded, e.g., by UV radiation. UV is a significant factor in deactivating marine phages, causing up to a 5% reduction in phage infectivity per hour [10].

1.3.1.1 Phages and Controlling Bacterial Blooms

Cyanobacteria cause damage to aquatic life by consuming resources and oxygen, starving aquatic life, and negatively affecting human health. Scientists are investigating the potential use of phages to control cyanobacteria (also known as blue-green algae) blooms in the environment [11]. There is hope that phages can biologically control water quality in wastewater treatment plants and in the environment without the use of harsh

chemical processes that would otherwise pose environmental and health hazards [4, 12]. Appendix B.3 contains more information about controlling *Cyanobacteria*.

1.4 Phage Cocktails and Human Health

There is particular interest in phage applications in human and animal health, called phage therapy. There are approximately 100 trillion microbes, comprising about 5,000 different types of bacterial strains, in the human gut. The phages will target the specific bacteria of interest, for example, *E. coli*, but they will not affect the other bacteria found in the human gut. Antibiotics indiscriminately affect any bacteria, disrupting the intricate ecosystem of the gut microbiome, acting as a scorched-earth mechanism. Antibiotics have also faced the challenge that bacteria are growing resistant to them, making the antibiotics less effective in the future [13, 14].

Phages, on the other hand, specifically target a specific bacterial strain. Antibiotic-resistant bacteria are typically less resistant to phages. The bacteria cell typically has a trade-off between antibiotic resistance and phage resistance. So by designing phages to be highly infective, there is hope that the phage-resistant bacteria will lose the antibiotic resistance to counter the phages [15, 16]. Appendix B.2 offers a more in-depth look at how healthcare professionals can utilize phages.

1.5 Potential Applications of Phages

Phages have numerous applications in industrial settings. Phage therapies can prevent the spread of common bacteria in livestock by incorporating phage therapy into the animal feed. Farmers often raise livestock in cramped spaces with inadequate sanitation facilities, which increases the risk of disease spreading. Factories can utilize phages to control the growth of bacteria, such as *Salmonella* while producing food [5, 17]. Appendix B.1 in Appendix B goes into more detail about using phages to control foodborne bacteria.

1.6 Modelling Phages in a Complex Community

What we know about phages mechanistically often comes from well-researched bacteria in a laboratory, such as *E. coli* and its phages, and what we know about phages in the environment comes from metagenomic surveys. Making the connection between

the mechanistic and metagenomic models, which would enable us to develop and test different models, is the challenging part. Because of this, we need mathematical models to help bridge the gap between the lab and the environment. There have been previous attempts to model the complex dynamics of populations between phages, bacteria, and resources within the environment using Ordinary Differential Equations (ODEs) and Delay Differential Equations (DDEs). Researchers cannot identify every interaction in the complex community, and even if they identify an interaction, they must experimentally derive the associated parameter values.

There are two primary methods for modeling phage-bacteria dynamics: spatial and non-spatial. In a spatial model, phages and bacteria can move through space and interact with their neighbors. Researchers have used partial differential equations (PDE) and cellular agent-based models (ABM) to model spatial interactions. Spatial models lead to more computationally complex models but can result in more biologically realistic results. By contrast, ODE and DDE models describe non-spatial models. In a non-spatial model, a well-mixed solution contains the bacteria and phages, and researchers make no distinctions regarding neighbors or distances to other entities. They simplify interactions with a probabilistic approach. At time step t, a percentage p of entities interact with each other.

Non-spatial models are easier to develop and understand and are more effective in modeling large populations, albeit at the cost of losing spatial information. For this thesis, the focus will be on modeling resource, phage, and bacteria interactions using an ODE model. I describe a phage-bacteria-resource system as a $p \times b \times r$ system. Current modeling methods have mainly stayed with $1 \times 1 \times 1$ models, meaning one phage, one bacteria, and one resource. This thesis aims to develop a simulation framework that can model any $p \times b \times r$ ODE system.

1.7 Software Overview

The project contains three parts, with an optional fourth part. The first section is to create the network interaction. Here, the user can define the number of resources, phages, and bacteria, who interacts with whom, and the strength and type of interactions. See Section 3.1.1 for further information. In Section 3.1.2, the user uploads the network model and parameters and, as output, receives the time data and population data as an array. Section 3.1.3 allows the user to interact with Section 3.1.1 and Section 3.1.2 with a dashboard. The user can graphically edit the attribute values of the network's edges and nodes and run more advanced visualizations, such as changing a parameter value to observe its impact on the population count.

Several plots are included out of the box, allowing the user to test them. The plots offered in Part 3 of the program provide interactivity, including the ability to hide and show lines and dots, zoom in and out, and hover over lines and dots to display more detailed data.

Finally, the user can optionally run multiple simulations and download the data to their disk to create their own visualizations using Section 3.1.4. The visualizations created in Section 3.1.3 can be recreated in Section 3.1.4. The user can choose the same parameter values used for a specific plot in Section 3.1.3, run the simulation (as described in Section 3.1.3.2), download the data, and reimplement the graphs.

The user can use the tool themselves by importing the Python classes in their code, initializing the classes, and passing the appropriate data.

Chapter 2

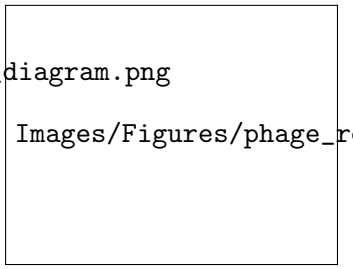
Literature review

2.1 Phage Biology

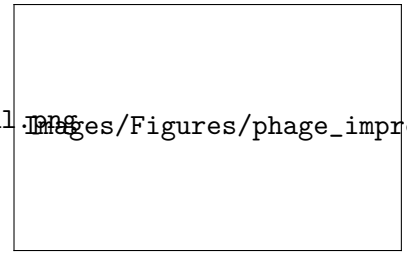
2.1.1 What Are Phages?

Phages are small bundles of proteins that contain viral DNA. Phages are composed of multiple parts, built like LEGO blocks, to complete the task of infecting a bacterium. Figure 2.1a shows the body parts of a phage. The phage aims to find a suitable bacterial host and infect the host with viral DNA. The DNA alters the host's metabolic pathways to its benefit and hijacks the cellular replication process to create new copies of the phage. Eventually, the cell lyses, releasing the newly created phages into the environment to infect more bacteria.

(a) Phage body structure. Image sourced from Fla [18].



(b) Phages infecting an *E. coli* bacteria. Image sourced from Twilley [19].



(c) Artist representation of phages infecting a bacterium. Image sourced from Vir [20]

Figure 2.1: Parts of a phage, a real-life picture of phages infecting an *E. coli* bacterium, and an artist's impression of phages infecting a bacterium.

2.1.2 How Does the Phage Cycle Work?

There are three main parts to the phage-bacteria host cycle: the infection stage, the lysogenic cycle, and the lytic cycle. Figure 1.1 shows a detailed overview of the phage cycle.

In the infection stage, a phage attaches to the surface of a bacteria cell. The infection stage involves the phage searching for, detecting, and attaching to a bacterium, followed by the injection of its DNA. Detection and attachment occur via phage receptor-binding proteins located at the tip of the phage tail that recognize specific receptors on the bacterial cell wall, triggering conformational changes that enable DNA injection [21]. The success of this process depends on the specificity and density of both phage and bacterial receptors [22]. Once attached, the phage injects its DNA into the host cytoplasm, where it can replicate independently. Once injected, the phage-cell pair enters either the lysogenic cycle or the lytic cycle.

The lysogenic cycle involves phage DNA integrating into the bacterial genome as a prophage, where it is replicated along with the host cell without causing immediate lysis. The phage evades host defenses such as CBASS and CRISPR-Cas systems, which can initiate programmed cell death, preventing phage replication or detect and degrade foreign DNA [23, 24]. Programmed cell death helps recycle resources for other bacteria [25]. Once integrated, the prophage can alter host fitness and provide resistance to other phages. During cell division, the prophage is copied into daughter cells but remains at risk of being excised by restriction enzymes [26]. Under certain stress conditions, such as DNA damage or activation of the SOS response, the prophage induces itself to exit the genome and enter the lytic cycle [22, 27, 28].

The lytic cycle is the process where a phage infects a bacterium, hijacks its replication machinery to produce new phage components, assembles these parts, and ultimately lyses the host cell to release new phages. This process involves hijacking the host's DNA replication to synthesize phage parts like the capsid, sheath, and tail Figure 2.1a. The phage does this by redirecting resources from internal cellular functions towards viral replication [25]. The phage parts self-assemble via protein-protein and protein-nucleic acid interactions [29]. Phages induce bacterial lysis by producing holin proteins that disrupt the cell membrane, releasing the phages and resources [30].

2.2 Bacterial Defense Against Phages

There is a constant battle between phages and bacteria. The bacteria do not want to be killed by the phages, so they develop defenses such as thickening their cell walls or

destroying the viral DNA.

2.2.1 Mutations in Bacterial DNA (Genetic (Co-)Evolution)

As bacteria cells grow and divide, random point mutations can occur in the DNA. These mutations can affect phage defenses, like thickening the cell wall or removing a receptor, making it harder for the phages to detect and infect the cell. Mutations can be partially effective if full effectiveness requires multiple steps to achieve. Random mutations can also fail to make the bacteria more resistant to phages by increasing phage susceptibility or by incurring a cost to the bacterial cell, such as losing receptors on the cell wall [31].

2.2.2 Horizontally Transferring DNA

Bacteria can horizontally transfer DNA to other bacteria on contact. A donor cell can donate DNA fragments using a mechanism called a pilus. The pilus acts as a tunnel between the donor cell and the recipient cell so that DNA can transfer from the donor cell to the receiver cell [32].

A phage can accidentally collect a piece of the host's DNA instead of its DNA during assembly. The phage, with the now dead host's DNA, can infect the next bacterium, injecting the new bacterium with the dead cell's DNA, thereby horizontally transferring the DNA [33, 34]. The transferred DNA can include natural phage defenses or significantly alter the genes and phenotype of the bacterium, making it undetectable to future phages.

2.2.3 Phage Inactivation and Decoys

Bacteria can further protect themselves by producing decoys that the phage will attach to rather than themselves. Freshly lysed bacteria may still have biomarkers that attract phages, leading phages to attach to non-viable cells where successful infection cannot occur. Bacteria can also produce proteolytic enzymes that will damage the proteins found in a phage [35]. Some bacteria can produce outer membrane vesicles that phages can absorb to, and later detach and float away with the phage [36]. These vesicles are suspected of having a minor impact as a sink [37].

2.2.4 Phenotype Resistance

Not all new phenotypes arise from genetic mutations. Resistance can result from phenotypic variation within a genetically identical population, allowing bacteria to express different resistance traits without altering their DNA. Gupta et al. [38] found that some *Bacteroides fragilis* bacteria were able to evade phage infection. The presence of combinatorial phenotypic states, where the differential expression of protective mechanisms created rare, super-resistant cells capable of withstanding phage attack. By acting together, these heterogeneously expressed anti-phage defense mechanisms created a phenotypic landscape where distinct protective combinations enabled the survival and regrowth of bacteria expressing these phenotypes without acquiring additional mutations [38].

2.2.5 Spatial Refuge/Biofilms

Bacteria can evade phages by forming spatial refuges, such as biofilms, or hide behind physical structures. Biofilms are dense microbial communities embedded in mucus, which impedes phage diffusion and protects resident bacteria [39]. On agar plates, spatial structure similarly limits phage spread [40]. While phages rely on passive diffusion, bacteria can actively move, further enhancing their survival [41].

2.3 Phage Counter Defense Against Bacteria

With some of the defenses that bacteria have developed, phages are constantly mutating to counter their defenses. If phages do not adapt to the ever-changing bacterial defenses, the phages will die out due to their inability to infect and multiply. It becomes an arms race, with each side trying to out-adapt the other. The bacteria and phages must have a delicate balance to ensure coexistence.

2.3.1 Genetic Mutations

Mutations in viral DNA will affect how the phage body parts are designed and built. These mutations will affect both external phage behavior, such as how it detects a bacterium, and internal behavior, including evading detection and integrating with the cell's DNA. The changes will lead to changes in overall phage fitness, i.e., the ability of the phage to infect, replicate, and lyse bacteria.

2.3.2 Viral Recombination

Multiple phages can infect a cell and replicate itself using the cell's internal replication process. Each phage has its own building blocks, for example, the capsid, tail, and sheath. If the proteins that build the subparts of each phage have similar chemical properties, the phage parts can be swapped between phages [29]. The swapping of parts allows for biological diversity to spread throughout a phage population. Each phage body part can have unique characteristics, such as a higher attachment rate, a larger DNA storage capsule, or a better probability of injection.

2.4 Phage Defense Against Phages

Some phages can employ defenses against other phages from infecting the bacterial cell, ensuring the host resources are all for itself. The act of preventing a secondary infection from a similar or closely related phage is called superinfection exclusion (SIE) [42]. The following are several methods for preventing further infections.

2.4.1 Altering Cell Structure

The prophage can alter the surface receptors of the bacteria, making it harder for other phages to detect the bacteria, reducing the chance of attachment and injection by other phages [43].

2.4.2 Protein Creation

Other phages, like the T4 phage, can create proteins like the Spackle protein, which inhibits the lysozyme activity used in the process of DNA injection by other phages [43, 44]. Some prophages can encode proteins that will interfere with the replication process of other phages. For example, the SieA protein encoded by phage P22 blocks infection from other phages [45].

Tail Assembly Blocker (TAB) is an anti-phage defense mechanism encoded by a *Pseudomonas aeruginosa* prophage. While TAB permits the invading phage to replicate its genome, it inhibits the assembly of the phage tail, thereby preventing the production of infectious virions. The prophage that encodes TAB is not affected by this inhibition, as it also expresses a protein that neutralizes TAB's blocking activity. Although the host cell still undergoes lysis, no infectious phages are released.

2.5 Bacteria and Phages in the Lab

Researchers worldwide are conducting laboratory experiments to gain a deeper understanding of the interactions between phages and bacteria. The aim is to gain a better understanding of how phages work and interact with bacteria at the molecular, host, and population levels.

2.5.1 Running Experiments

A researcher might run the experiment in a liquid medium containing water, carbon, and nitrogen sources. This liquid medium, often referred to as broth, facilitates the cultivation of bacteria in a well-mixed environment, allowing researchers to monitor bacterial growth and phage infection dynamics over time. By adjusting parameters such as resource concentration, temperature, and pH, researchers can simulate different environmental conditions and observe their effects on phage-bacteria interactions.

Samples are taken at set time points to measure bacterial density, phage titer, and resource concentration. These data enable researchers to fit mathematical models, estimate growth rates, and determine phage parameters, such as latent time and burst size, from one-step growth curves [1, 46].

2.5.2 Chemostats

Commonly used setups include liquids containing phages, bacteria, and resources in a chemostat and batch culture. Chemostats enable the continuous addition of resources and removal of waste, thereby maintaining steady-state conditions that are ideal for studying long-term dynamics.

2.5.3 Petri Dishes

Petri dishes are another common method used to grow bacterial colonies. Agar, a jelly-like substance derived from seaweed, is commonly used as a solid-growth medium in Petri dishes. Agar provides a stable surface for bacteria to grow on and form visible colonies. Clear zones, called plaques, appear where phages have infected and lysed the bacteria, allowing for the quantification and observation of phage activity. The phages can diffuse on the agar plate, infecting neighboring cells. Phage infection creates clear plaques (2-3 mm) where bacteria are absent. Figure 2.2 shows an example of a bacteria lawn with phage plaques.

2.5.4 Measuring Growth

Bacterial density in liquid media is usually measured by optical density (OD) using a spectrophotometer. For phages and resources, mass spectrometry can be used. OD measurements require calibration and are not directly comparable across devices [47]. Using cell counts (e.g., $cells/ml$) allows for more consistent comparisons across experiments and labs [48].

With Petri dishes, it is more challenging to measure bacterial growth and plaque size. It may be possible to wash the bacteria off into a test tube with water to measure the optical density, but the results are inconsistent. It may be possible to quantify the change in plaque size using an image analysis program; however, the results may be inaccurate and sensitive to variations in lighting conditions.

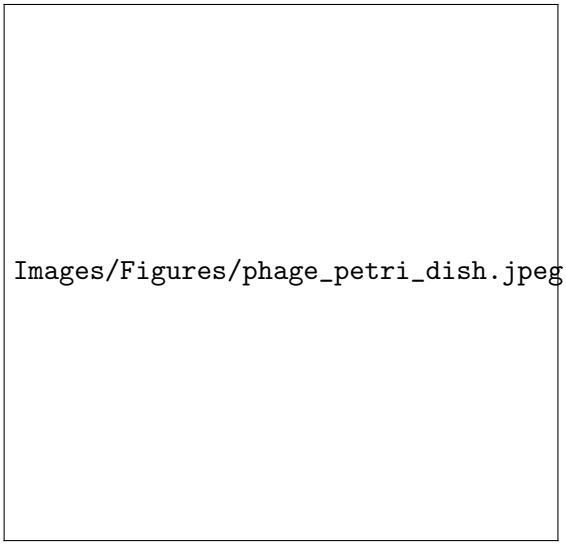


Figure 2.2: Bacteria lawn, the dots on the petri dish show no bacteria growth due to the presence of phages. Photo courtesy of S. Flickinger.

2.5.5 Serial Transfer

Serial transfer (ST) is a method employed by a bacteriologist, where, after a set amount of time, the bacteriologist pipettes medium containing phages, bacteria, and resources out of a test tube and adds the old medium to a new test tube with fresh medium. At this stage, the bacteriologist can add more bacteria or phages to the test tube. Only resources are typically added during the transfer process. Researchers can optically measure bacterial density using an optical density meter or employ a mass spectrometer to determine phage concentration at set time points during the experiment. As the bacteria grow, they consume the resources found in the medium. The resources will eventually run out, and the bacteria die out due to a lack of resources. By introducing

new resources at set time intervals, the bacteria can regrow and exhibit a semi-stationary behavior.

2.5.6 Growth Curves Typically Seen in a Lab

When choosing parameter values, it is essential to select values that can realistically be found in real-life systems and be replicated in the lab. There are various features that a researcher will look for in a growth curve produced in a lab. A combination of these features yields an ideal growth curve that replicates real-life bacterial growth.

The idealized dynamics of bacterial populations undergoing phage infection have several phases. First, there is an apparent exponential rise in bacteria growth, growing 40-100x in population in the span of a few hours. At a certain point in time, the bacterial population starts decreasing almost as fast as it was growing.

Phage populations also exhibit exponential growth but with a delay in growth. Initially, there is no growth in the phage population. After a set amount of time, the phage population will start to grow and peak a few hours after the bacteria population reaches its peak. If there is no phage death or removal, the phage population will eventually reach a plateau when every bacteria has died.

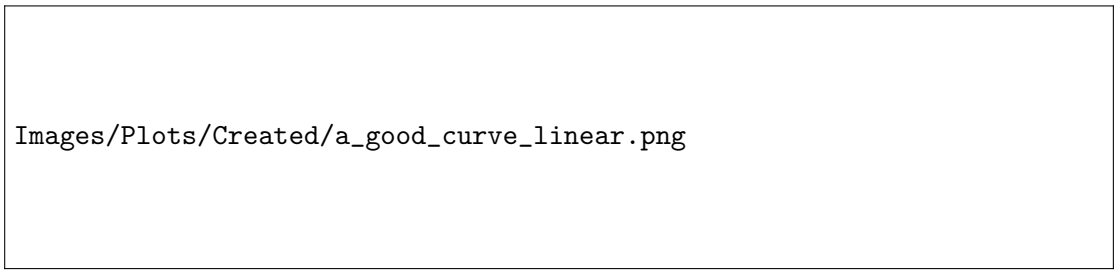
Figure 2.3a shows an example of a curve for a $1 \times 1 \times 1$ system that would typically be seen in a lab. Figure 2.3b is the same plot but with a logarithmic y-axis. These specific plots exhibit a clear growth, peak, delay, and death cycle.

2.6 Software Mathematically Modelling Phages, Bacteria, and Resources

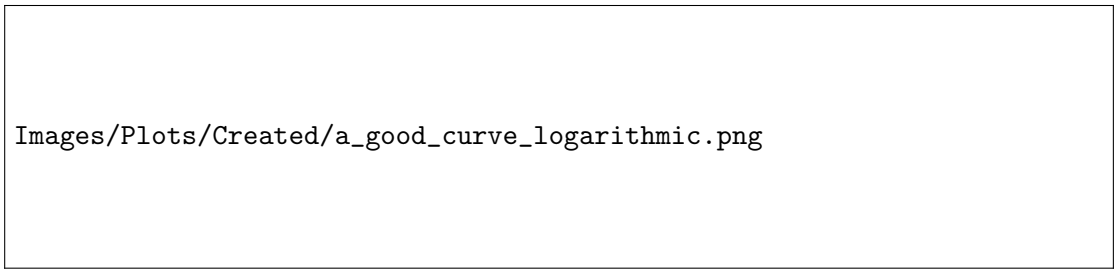
Some software programs modeling phage-bacteria-resource interactions already exist. Here I cover two software, Cocktail and PhageDyn, that have been created to model phages and bacteria, as well as their limitations.

2.6.1 Cocktail

Nilsson [3] developed Cocktail to model phage-bacteria-resource kinetics in a chemostat. The model assumes there is one bacterial strain that can be infected by phage A and phage B, and by both phages simultaneously, phage AB. The model simulates bacterial resistance to phage A and phage B, as well as a combination of phage A and B.



(a) An example linear y-axis for a curve that researchers aim to replicate.



(b) The equivalent logarithmic y-axis plot for a curve that researchers aim to replicate.

Figure 2.3: Growth of a population in a $1 \times 1 \times 1$ system. The log plot allows us to visualize behavior at values approaching zero and to plot data on a logarithmic scale. The parameters used for this plot can be found in Table E.1.

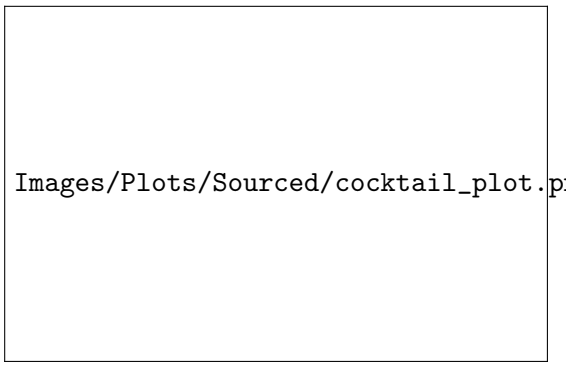
The user can control parameter values, such as resistance rates to A, B, and AB, resource concentration and outflow, and phage adsorption rate. The user can also control model settings, such as if the model is deterministic or stochastic, and the step size [3]. Figure 2.4a shows four sample output plots.

2.6.2 PhageDyn

PhageDyn is a Java applet that models phage dynamics in multi-reactor industrial wastewater treatment plant models. PhageDyn interacts with existing GPS-X [49] files to incorporate phage dynamics into models of industrial wastewater treatment plants [4]. Krysiak-Baltyn et al. [4] developed PhageDyn to determine how phages can reduce foaming caused by bacteria in wastewater treatment plants, another real-life application of phages [50]. PhageDyn does not simulate phage dynamics on its own; instead, it manipulates existing files in GPS-X to incorporate phage dynamics into wastewater treatment plant models. Figure 2.4b shows the output that PhageDyn provides.

2.6.3 Cocktail and PhageDyn Limitations

There are limitations to Cocktail and PhageDyn. Cocktail can model up to a $2 \times 1 \times 1$ system and in a chemostat environment. Chemostats receive a constant influx of new resources and a constant removal of medium from the chemostat. Cocktail's model



(a) Figure A) *E. coli* infected with phage T4 in a chemostat exhibiting an oscillating growth behavior, following the model of Bohannan and Lenski [51]. Figure B) Oscillations of bacteria and phages can exist at higher titers, dependent on low resource concentration, following the model of Lenski [52]. Figure C) As the concentration of resources change, this results in increasing oscillations, but not going extinct. Figure D) A system modeling the interactions with phage A and B.

(b) **Purple** is heterotrophic biomass, **Blue** is foaming biomass, **Red** is phages, **Light Blue** is total suspended solids. Figure A) Biomass concentration immediately post phage dosing. Figure B) Biomass concentration with low phage concentration and maintain low concentration post spike in population count. Figure C) Biomass concentration when phages are extinct. Figure D) Biomass concentration with a less virulent and low adsorption rate phage, indicating coexistence with biomass. A change in phage concentration shows a decrease in heterotrophic and foaming biomass [4].

Figure 2.4: Example output from Cocktail and PhageDyn, respectively. For PhageDyn, the concentration of heterotrophic biomass in an aerobic plug flows across four situations. See Nilsson [3] and Krysiak-Baltyn et al. [4] for more information on parameter values and supplementary resources.

can not be easily adapted to other models. The ODE model accepts inputs from a hardcoded GUI front end. Any changes to the front end or the ODE model will require corresponding changes to both the ODE model and the front end to accommodate the new inputs and outputs. The code for Cocktail is open source, so adding new buttons and changing the model should not pose a significant challenge, but it is still an undertaking.

PhageDyn works with GPS-X, a highly specialized wastewater treatment modeling software. PhageDyn is programmed for a specific task, with no flexibility in modifying the model or inputs. PhageDyn assumes biomass instead of individual bacteria populations. However, PhageDyn is no longer available for download.

2.7 Methods of Modeling Phages and Bacteria

There are various ways to model phage, bacterial, and resource populations and their interactions, but the most common approach is with Ordinary Differential Equations (ODEs) or Delay Differential Equations (DDEs).

The ODE method is simple to understand and easy to set up, but it can only capture large population dynamics. Certain assumptions about community interactions also need to be made, such as the assumption that everything is probabilistic in nature. For example, every timestep, α percentage of phages interact with bacteria DDEs are similar to ODEs, but DDEs incorporate time delays to account for processes that depend not only on the current state but also on past states, thereby incorporating behavior that has a delay, such as latent infection time.

One way to introduce a delay in the infected bacteria releasing phages is to force the infected bacteria to undergo stages. Once the infected bacterium has progressed through every stage of infection, it releases the phages into the system. For example, in the paper Geng et al. [1], infected bacteria go through M stages of infection before lysing. By decreasing τ (the latent period) in the model proposed by Geng et al. [1], more infected bacteria go from infected state i to infected state $i+1$ per timestep with rate $\frac{M}{\tau}$, causing the infected peak population count to peak earlier.

Each model can be further developed, for example, by adding temperature and pH dependence, bacteria releasing nutrients upon lysis, or phage resistance.

2.7.1 Campbell DDE Model for Chemostats

The Campbell [53] model is a DDE model that describes the growth of bacteria and phages in a chemostat, a bioreactor used to cultivate microorganisms, where fresh liquid medium is constantly added, and liquid culture is constantly removed at a constant rate to maintain a constant culture volume. The growth of bacteria B and phage P is described as follows.

$$\frac{dB_i}{dt} = k_B B\left(1 - \frac{B}{L}\right) - \alpha B - k_A P B \tag{2.1}$$

$$\frac{dP}{dt} = k_A N\left[B(t-\tau)P(t-\tau)\right] - k_A B P - k_I P - \alpha P \tag{2.2}$$

α is the inflow and outflow rate, k_A is the adsorption rate, k_I is the rate of spontaneous inactivation of phages, k_B is the bacterial growth rate following a logistic growth rate, with max level L, N is the burst size after τ time units, and t is the time. $B(t-\tau)$ and $P(t-\tau)$ is the population of the bacteria and phages at time $t-\tau$.

Steady states occur when $\frac{dB}{dt} = \frac{d0}{dt} = 0$, and there are four steady-state solutions.

1. $B = 0, P = 0$: Both bacteria and phages are absent.

2. $B = 0, k_I + \alpha = 0, P = P_0$: Bacteria are absent, and phages persist only if the inactivation and outflow rates sum to zero.
3. $B = L\left(1 - \frac{\alpha}{k_B}\right), P = 0$: Bacteria reach their carrying capacity in the absence of phages.
4. $B = \frac{k_I+\alpha}{k_A(N-1)}$, $P = \frac{k_B}{Lk_A}\left[L\left(1 - \frac{\alpha}{k_B}\right) - \frac{k_I+\alpha}{k_A(N-1)}\right]$: Coexistence of bacteria and phages at steady state.

Solutions 3 and 4 happen when the initial populations are not zero, and the flow rate or the spontaneous phage inactivation rate is greater than 0.

2.7.2 Weitz ODE Model for Coevolutionary Arms Race

The Weitz et al. [54] model describes how to model selective pressure and trait adaptations in phage and bacterial populations, and how these adaptations affect the dynamics of phage and bacterial populations. As phages and bacteria evolve, the phage's ability to adsorb to bacteria and the bacteria's ability to consume resources change. The receptors on the bacteria's cell wall and phage's tail act as a key-lock model, so any changes to the

$$\frac{dR}{dt} = -\omega(R - R_0) - \sum_i \epsilon\gamma(x_i)\frac{RB_i}{R+K} \tag{2.3}$$

$$\frac{dB_i}{dt} = -\omega B_i + \gamma(x_i)\frac{RB_i}{R+K} - \sum_j \phi(x_i, y_j)B_iP_j \tag{2.4}$$

$$\frac{dP_i}{dt} = -\omega P_i + \sum_j \beta\phi(x_j, y_i)B_jP_i \tag{2.5}$$

$$\gamma(x_i) = \gamma_0 e^{-\frac{(x-x_0)^2}{2\xi_n^2}} \tag{2.6}$$

$$\phi(x_i, y_j) = \phi^{-\frac{(x_i-y_j)^2}{2\xi_v^2}} \tag{2.7}$$

ω is the washin rate, R_0 is the initial resource concentration, x_i is the trait value for bacteria i, and y_j is the trait value for phage j, $\gamma(x_i)$ is the resource consumption rate dependent on trait x_i, $\phi(x_i, y_j)$ is the adsorption rate between trait x_i and trait y_j. K is the half saturation Monod constant, β is the burst size of a phage, ξ_n is the stable uptake range of hosts, and ξ_v is the host range of phages.

Specifically, ξ_n is the range of possible host phenotypes. whose maximal growth rate is within $e^{-\frac{1}{2}}$ of the maximum for all phenotypes. ξ_v is the range of possible host phenotypes for which any given phage has an adsorption rate within $e^{-\frac{1}{2}}$ of its maximal adsorption rate. K and β can also be considered traits of the bacteria and phage, but

the authors decided to hold them constant for mathematical tractability. $\gamma(x_i)$ implies that there is an optimal configuration for maximal resource uptake when $x_i = x_0$, and an opportunity for a trade-off between resource uptake and phage avoidance. $\phi(x_i, y_j)$ suggests that for every bacteria trait x_i, there is a phage trait y_j that maximizes the strain-specific adsorption rate.

2.7.3 The Golding Model

The "Golding model" will be the default ODE model used in this report, as described by Geng et al. [1]. The model describes the interactions between resources, uninfected bacteria, infected bacteria, and phages.

2.7.3.1 The Original Golding Model

The Golding model (Equation (2.13)) describes three biological processes: cell consumption of resources and growth, phage/cell encounters and infection, and cell lysis.

Once infected by a phage, the bacteria goes from U to I_1. The bacteria go through M stages of infection $I_1, \ldots, I_M$ before lysing, The bacteria goes from state I_k to state I_{k+1} with equal transition rate $\frac{M}{\tau}$. The infection rate of a cell is r. After a bacteria lyses after stage I_M, β phages are released, the burst size of the phage. $g(R, v, K)$ described the cell growth process, the instantaneous growth rate dependent on the Monod equation, where v is the maximal growth rate of the bacteria population, and K is the Monod constant. Bacteria consume a resource at a rate of e.

2.7.3.2 The Adapted Golding Model

The original Golding model simulates a $1 \times 1 \times 1$ system. To adapt this model to fit a $p \times b \times r$ model, it needs to be slightly modified. Other changes can be made to the model, for example, by adding a washin rate ω^i, where resources are constantly introduced, and a washout rate ω^o, where all phages, bacteria, and resources are washed out at a proportional rate. Equation (2.19) highlights the ω^i and ω^o in red. If ω^i and ω^o are zero, and there is only one phage, one bacteria, and one resource, then you get the original Golding model.

The adapted model accounts for the interactions among multiple phages, bacteria, and resources, assuming that these interactions occur independently of one another.

Equation 2.13 The Golding model was sourced from Geng et al. [1]. The text in red has been added to the model, incorporating (the washin) fresh resources (ω^i) and the removal (washout) of entities (ω^o). By default, ω^i and ω^o values are zero unless stated otherwise. The parameter values can be found in Table A.1.

$$\frac{dR}{dt} = -e \cdot g(R, v, K) \cdot (U + \sum_{k=1}^{M} I_k) \tag{2.8}$$

$$\frac{dU}{dt} = g(R, v, K) \cdot U - r \cdot U \cdot P \tag{2.9}$$

$$\frac{dI_1}{dt} = r \cdot U \cdot P - \frac{M}{\tau} \cdot I_1 \tag{2.10}$$

$$\frac{dI_k}{dt} = \frac{M}{\tau}(I_{k-1} - I_k) \text{ for } k = 2, \ldots, M \tag{2.11}$$

$$\frac{dP}{dt} = \beta \cdot \frac{M}{\tau} \cdot I_M - r \cdot (U + \sum_{k=1}^{M} I_k) \cdot P \tag{2.12}$$

$$g(R, v, K) = \frac{v \cdot R}{R + K} \tag{2.13}$$

Equation 2.19 The adapted Golding model. The probability of phage p infecting bacteria b is r_{pb} and is not to be confused with the resource concentration R_r. The interactions are the sum of all interactions as they occur simultaneously.

$$\frac{dR_r}{dt} = -\sum_{b \in B} e_{br} \cdot g(R_r, v_{br}, K_{br}) \cdot (U_b + \sum_{k=1}^{M} I_{b_k}) + w_r^i - w^o \cdot R_r \tag{2.14}$$

$$\frac{dU_b}{dt} = U_b \cdot \sum_{r \in R} g(R_r, v_{br}, K_{br}) - U_b \cdot \sum_{p \in P} r_{pb} \cdot P_p - -w^o \cdot U_b \tag{2.15}$$

$$\frac{dI_{b_1}}{dt} = U_b \cdot \sum_{p \in P} r_{pb} \cdot P_p - \frac{M}{\tau_b} \cdot I_{b_1} - w^o \cdot I_{b_1} \tag{2.16}$$

$$\frac{dI_{b_k}}{dt} = \frac{M}{\tau_b}(I_{b_{k-1}} - I_{b_k}) - w^o \cdot I_{b_k} \text{ for } k = 2, \ldots, M \tag{2.17}$$

$$\frac{dP_p}{dt} = \sum_{b \in B} \beta_{pb} \cdot \frac{M}{\tau_b} \cdot I_{b_M} - r_{pb} \cdot (U_b + \sum_{k=1}^{M} I_{b_k}) \cdot P_p - w^o \cdot P_p \tag{2.18}$$

$$g(R_r, v_{br}, K_{br}) = \frac{v_{br} \cdot R_r}{R_r + K_{br}} \tag{2.19}$$

2.7.4 Other Models

Here I discussed how DDEs and ODEs can be used to model populations of phages and bacteria. However differential equations can also be used to model sub-cellular molecular level, potentially to model individual bacteria and phages interacting with one another [55, 56], or in the case of partial differential equations, model the diffusion of bacteria, phages, and resources through the system [57, 58].

Chapter 3

Methods

3.1 Project Overview

To help complete this Master's thesis, I developed various tools to assist in creating the final model outputs. The project consists of four parts, with the final one being optional.

- A GUI network creation tool is used to define the number of phages, bacteria, and resources in the network and their interactions.
- The simulation framework handles the data, runs the ODE simulations and sends the data back to the dashboard.
- A dashboard that the user can interact with to run simulations, edit parameter values, and plot and interact with the visualizations.
- Optionally download the simulation data to create your own visualizations and analyses.

Appendix C contains a flowchart illustrating the user-system interactions.

3.1.1 Network Creation Tool

The user uses the GUI network creation tool to create and edit the network interactions. There are three types of variables in the simulations: phages, bacteria, and resources. Every node in the network represents either a unique phage population, bacterial population, or resource. A bacterium can be further divided into uninfected and infected bacteria. An edge links two nodes together if there is an arbitrary interaction occurring between them. Phages infect bacteria and consume resources. The ability of a phage to

infect a specific bacterium and the resources each bacterium can utilize describe a network of interactions. Finally, the user can export (and later import to edit) the network representation for use in Section 3.1.2, Section 3.1.3, and Section 3.1.4. Figure 3.1 shows the layout of the GUI tool and example networks. This report will use these network structures.

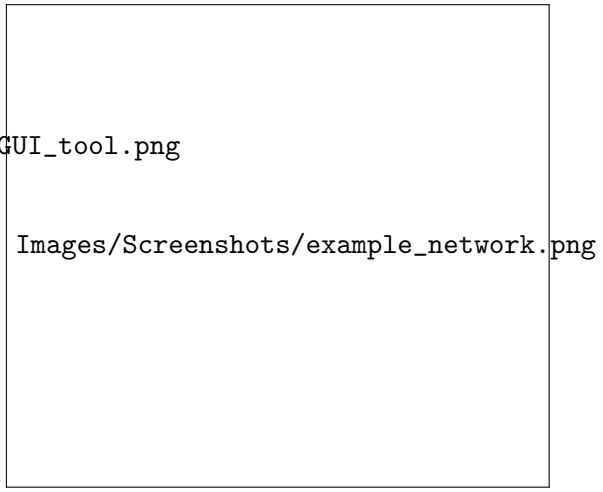
Every node represents a unique entity, and each entity has its intrinsic properties. The user can intuitively define phages, bacteria, resources, their interactions, environmental data, and model settings using the GUI tool. This tool allows users to quickly and intuitively define entities and their attributes, entity interactions and their attributes, environmental data, and model settings. An edge links two entities together if there is an arbitrary interaction occurring between the entities, with the properties exhibited in the interaction dependent on the interacting entities. Self-interactions are allowed in the network. There is an environment node used to store global environmental data, such as the system's temperature and pH. The settings node holds information such as simulation length, max time step, and the type of ODE solver to use. The tool provides functionalities for adding, editing, and visualizing nodes and edges, as well as importing and exporting the network structure.

Once the user is happy with the graph shape, they can export the network representation for use in Section 3.1.2, Section 3.1.3, and Section 3.1.4. The most important part is that the user defines the shape and the attributes of the network, as these cannot be edited in part 2 onwards. It is possible to return to the network creation tool and upload the graph to edit the network representation and default parameter values.

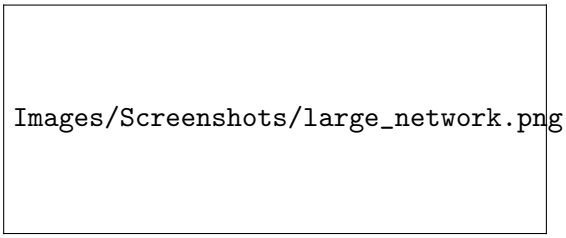
The user can edit the values of the attributes in Section 3.1.3, so the parameter values do not have to be perfect. As such, the user does not need to keep on using the GUI tool to edit parameter values.

Figure 3.1a shows the layout of the GUI tool. Figure 3.1b shows an example network. Users can edit the graph, including adding or removing nodes and edges, as well as editing parameter values by using the various buttons. Manually adding nodes and edges can become tedious and repetitive for large graphs, so users can add multiple nodes and edges simultaneously. The user can self-determine the default attribute names and values to assign, as well as, if applicable, how the parameter values are randomized. By default, there is already an environment node, "E", and a setting node, "S". These nodes store data such as pH, temperature, or the simulation length. Nothing can interact with the environment and setting node, as they hold data about the environment and network solver.

(a) The GUI tool, as seen upon program startup.



(b) A $1 \times 1 \times 1$ and $2 \times 2 \times 3$ network.



(c) A $20 \times 20 \times 10$ network.

Figure 3.1: This network topography, along with a $1 \times 1 \times 1$ network, will be used in the Chapter 3 and Chapter 4 sections. The parameter values for the networks can be found at Table E.1, Table E.4 and Table E.3 Each node represents a phage, bacteria, or resource, with arbitrary interactions occurring between them. Although not shown and used here, edges between the same entity types and self-loops are allowed.

3.1.2 Simulation Framework

The user provides an ODE model and the network topography as input to the framework. The simulation framework handles the input, output, collection, and storage of the simulation input and output. The framework uses SciPy's [59] *solve_ivp()* numerical solver [60] to simulate the provided ODE equations and calculate the population levels through time. The user receives two outputs from the framework. The first output is an array of time values that the solver used to calculate the population count. The second output is an array containing the population count at each time step for every entity.

To facilitate more complex model behavior, additional system variables can be added to the simulation. An example of this distinction is the difference between uninfected and infected bacteria. In the network model, you explicitly create a $3 \times 2 \times 3$ network with

three phages, two bacteria, and three resources. You additionally tell the solver to add $2 \cdot M$ additional states that represent the infected states. Therefore, the ODE solver would solve for three phages, two uninfected bacteria states, $2 \cdot M = 2 \cdot 4 = 8$ infected bacteria states, and three resources.

Adding a resource reservoir to the model would also be straightforward. Three extra resource variables would be added to the solver, where the ODE and solver would model the transfer of resources from the reservoir to the simulation environment. The provided ODE model must accurately model and transfer resources from $R_{r_{\text{reservoir}}}$ to $R_{r_{\text{chemostat}}}$ correctly. The bacteria would only consume from $R_{r_{\text{chemostat}}}$.

The user's ODE model must accurately represent each (extra) phage, bacterium, and resource and correctly handle changes in states.

3.1.3 Visualization Dashboard

The third part involves analyzing and visualizing the simulation results on an interactive Dash Plotly [61] dashboard. The user can use a dashboard built using Plotly Dash to interact with the solver and network. The user can quickly change parameter, environment, and setting values with the dashboard. As output, the dashboard displays interactive plots, enabling the user to analyze the system.

The dashboard enables users to interact with the network, model, and prebuilt visualizations. The dashboard contains three separate sections. The first section enables the user to edit parameter values and solver settings on the fly, allowing for quick iteration through different conditions and fine-tuning of parameter selection without needing to rebuild the network using the GUI tool. The second section enables users to visualize how the population count evolves for a given IC and parameter values, allowing them to test the network input quickly. The final section enables the user to conduct more advanced analyses on the network, for example, by modifying multiple parameter values and visualizing the resulting output.

3.1.3.1 Editing Network and Parameter Values

The editing network and parameter value contain five separate sections.

Initial Condition The IC settings panel (Figure 3.2a) allows the user to edit the initial starting values of the entities. Each entity type has a table containing the initial population count.

Vector Data Data stored as a vector, which includes data on the nodes of phages, bacteria, and resources, is displayed in the Vector tab. This data is typically associated with node data. That would be the washin rate of resources or the bacteria's latent time Figure 3.2c.

Matrix Data Data stored as a matrix, typically representing the edges between phages, bacteria, and resources, is stored in the matrix tab, Figure 3.2b.

Environment and Settings The environment data and settings data also have their own tab, Figure 3.2d and Figure 3.2e, respectively. The data stored in the environment act as global variables, like the pH of the system or the constant washout rate ω^o. The settings node contains the solver and simulation settings, including the simulation length and minimum time step.

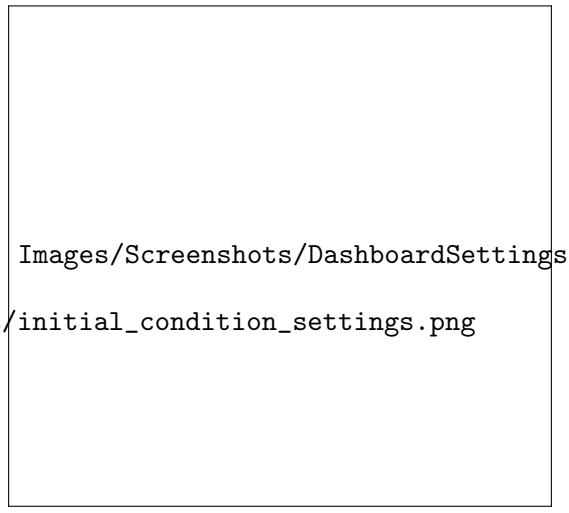
3.1.3.2 Visualization and Analysis

In the analysis section, the user can run different analysis methods to gain a greater understanding of the model. For simplicity, the visualizations on the dashboard only support a $1 \times 1 \times 1$ model. This makes it easier for the user to analyze the system. The goal of the dashboard is to investigate a simple system in order to gain a deeper understanding of the system. The next step is to simulate more complex models using the Ultimate Analysis section, where you implement your own visualizations. There are five prebuilt visualizations, which are described below. These five visualizations are called serial transfer, parameter analysis, initial value analysis, phase portrait, and Sobol analysis. The aim of these visualizations is to investigate how a simple system responds to varying inputs before moving on to more complex models.

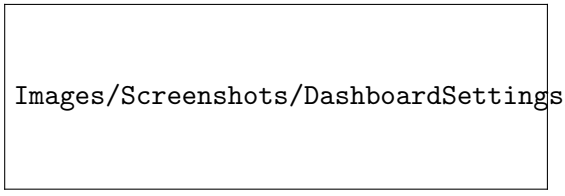
After the user has a deeper understanding of the system, they can run and download custom simulations to create their own custom visualizations in the Ultimate Analysis section. The saved simulation data is stored as a *.parquet* file, a tabular-like data format. Dask can query the simulation data, allowing users to find specific simulation results. Parquet with Dask offers superior performance and data storage solutions that Pandas does not offer.

Serial Transfer On the dashboard, a user can select the dilution factor, which divides the phage, bacteria, and resource population count by that number (Figure 3.3a). Then, the program takes the IC values defined in Section 3.1.3.1 and adds those values to the respective entity.

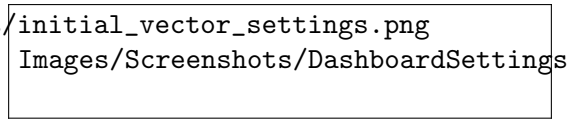
(a) The tab where users can edit the ICs of the entities.



(b) The tab where the user can edit the matrix attribute values.



(c) The tab where users can edit vector attribute values.



(d) The tab where users can edit environment values.

(e) The tab where a user can edit the settings of the solver and simulation.

Figure 3.2: The tabs where the user can edit the various parameter values and control the simulation parameters

As an example, if the simulation ended with 3500 phages, 100 (uninfected) bacteria, and 50 resources, the dilution factor is 10, and the user wants to add 350 new bacteria and 130 new resources for the following simulation, the new starting condition for the following simulation would be $\frac{3500}{10} = 350$ phages, $\frac{100}{10} + 350 = 660$ bacteria, and $\frac{50}{10} + 130 = 135$ resources.

As output, ST will display how the population evolves, as well as the final population value at the end of each serial transfer run. An example output is shown in Figure 3.3b.

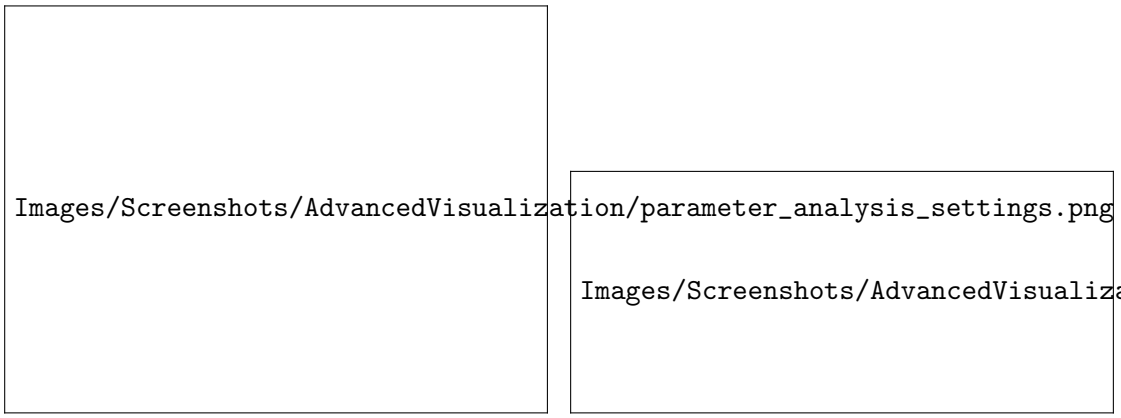
(a) The section where the user can set up the ST. **(b)** The ST output plots.

Figure 3.3: The ST settings and output.

Parameter Analysis The Parameter Analysis (PA) settings tab, as shown in Figure 3.4a, allows the user to select two parameters and individually run the model with varying input values. The values that can be tested and changed include all IC values, vector and matrix data, and environmental data. As input, the user can select two parameters of choice. The user can manually choose which parameter values they want to test or test a range of values equally spaced by selecting the number of values to test. Finally, the user can optionally run a ST, where the ST uses the settings found on the ST tab.

Figure 3.4b shows the heatmap that the user can expect, one heatmap for each entity type. Each heatmap cell represents the input of two unique parameter values and shows the population count for that parameter run at the time indicated by the slider. As the user slides the slider, the value inside the cell updates to correspond with the selected time. Note that the heatmap color range resets for each heatmap, so similar colors across heatmaps and across time will not correspond to the same values.

Initial Value Analysis The initial value analysis (IVA) settings tab, as shown in Figure 3.5a, allows the user to select a single parameter and adjust its value over a range of values, visualizing how a change in parameter value affects the population count of the entities.



(a) The available PA settings. **(b)** The expected PA output.

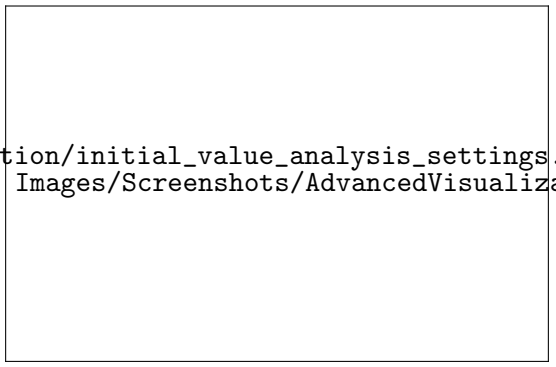
Figure 3.4: The PA settings and output.

Figure 3.5b shows the plots that the user receives. For each entity type, three plots are created. The left plot shows the population count through time, one line for each parameter value submitted. The middle plot takes each run and calculates the "percentage from the max value" (default value of $0.95 \rightarrow 95\%$) reached from the peak. This value is considered the time of peak and is used to address some issues that can arise when the population plateaus or continues to rise. For example, suppose the phage population starts to plateau early in the simulation. In that case, the program can calculate that the "peak" of the phage population happened at the beginning of the simulation.

The plotter places the initial value on the x-axis, while the time at which the max value is reached appears on the y-axis. Using the plotted data, one can create a linear or log fit. In Figure 1 of Mulla et al. [46], the authors vary the initial bacterial concentration and measure the time until bacterial collapse. The initial concentration and corresponding collapse time are plotted on a logarithmic x-axis, with a linear regression fitted to the log-transformed data. The observed logarithmic decrease suggests that the phage kinetics is adsorption-limited. Figure 4.2a replicates this graph.

Using the IVA tool can help understand how a change in parameter value affects the time at which the population count reaches a maximum. The slope, intercept, and R^2 value are stored and saved in the third plot, a bar chart with an editable name. For every rerun of the IVA, the bar chart stores the slope, intercept, and R^2 value and displays the linear regression next to previous runs. When executed with multiple parameters, this enables comparison of high-level results across various parameters and experimental conditions.

(a) The settings for the IVA tab.



(b) An example of an IVA output.

Figure 3.5: The IVA settings and output.

Phase Portrait The phase portrait plot enables users to analyze how a phage, bacterium, or resource population evolves in relation to one another. Phase portraits indicate how one population increases while the other decreases and vice versa. Steady states can be identified and classified as either stable, unstable, or saddle points. It is also possible to visually identify attractor and repeller points by observing where the population values tend to trend. By comparing different starting points, it is possible to see if the system is chaotic or not. Figure 3.6a and Figure 3.6b show the phase portrait setup and sample output.

(a) The user can select two starting values for the IC. However, they cannot choose vector, matrix, or environment settings, as the plot displays the development of phage, bacteria, or resource populations against other phage, bacteria, or resource populations. As typical, the user can select their own values or auto-generate values between two values, as well as use an ST option. There is also an option to take the logarithm of the x and/or y-axis.

(b) An example phase portrait run.

Figure 3.6: Phase portrait settings and output.

Sobol Sensitivity Analysis It is essential to understand how a change in parameter value impacts the output of a model. Models will have parameters that are more important and have a greater impact on the model's output than other parameters.

Sobol analysis [62], a variance-based sensitivity analysis, is a method that allows a user to quantify the importance of input parameters on a measured aspect of the output by changing the input parameter values of the model and measuring the resulting change in model output. Sobol can only measure a single univariate model output, for example, the final population value, the smallest or largest value reached, the time at which the largest value was reached, or any other univariate output. Sobol quantifies the variance in the output that can be attributed to a specific parameter and measures the effects of global/total (ST), first-order ($S1$), and second-order sensitivity ($S2$).

First order Si, or local sensitivity, is the measurement of the effect that parameter i has on the variance of the output. The second order is the measurement of the interaction between parameter i and parameter j and how this interaction contributes to the output variance. Etc. for third order and higher. Global, also known as total sensitivity, is the sum of all interactions. If $ST_i \gg S1_i$, then parameter i depends on higher-order interactions with other parameters, while when $ST_i \approx S1_i$, then i does not interact much with and depend on other parameters. $ST_i \geq S1_i$ and ST_i can be greater than 1, while $S1_i \leq 1$.

When a model is treated as a black-box model, the model acts as a function $Y = f(X)$, where X is an input vector of d parameter values, and Y is a univariate model output. The first-order sensitivity measures the output variance resulting from the primary effect of parameter X_i. Measuring the effect of varying X_i averaged over other input parameters and standardized to provide a fractional contribution to the overall output variance. The following equation describes the first-order sensitivity.

$$S1_i = \frac{V_i}{Var(Y)}$$

where $V_i = Var_{X_i}(\mathbb{E}_{X_{\sim i}}[Y|X_i])$ and where $X_{\sim i}$ represents all the parameters that are not X_i. All parameters are summarized in Table A.2

The second-order index measures the impact of the interaction between input X_i and X_j. For many inputs, this becomes unwieldy to analyze. The global sensitivity is used to analyze the global sensitivity without evaluating $2^d - 1$ indices. It measures the contribution to the output variance of X_i, including all variance due to X_i's interaction with other variables.

$$S1_i = \frac{\mathbb{E}_{X_{\sim i}}[Var_{X_i}(Y|X_{\sim i}))}{Var(Y)} = 1 - \frac{Var_{X_i}(\mathbb{E}_{X_i}[Y|X_{\sim i}])}{Var(Y)}$$

Sobol accepts a list of parameter names and an interval of values to sample from, which the user can input in the Sobol settings tab, Figure 3.7a. If you do not specify any

values, the simulation excludes the parameter and uses the default value instead. The user then needs to select the number of samples to run, using the formula 2^x, where x is the number they input, and 2^x is the number of samples that Sobol will create and run. The larger x is, the more accurate the Sobol analysis results will be, but the more simulations we will need to run. If the second order is not chosen, we run the model $N(D+2)$ times with the randomly sampled input values. If the user wants to analyze the second-order interactions, then the model will run the system $N(2D + 2)$ times, where N is a multiple of 2, and D is the number of input parameters. Due to the randomness of the sampling method, the user can, but does not need to, submit a seed value.

Figure 3.7b shows a sample Sobol output. The global and first-order sensitivities are displayed side by side, with each sub-row in a plot representing a specific entity type. We can see the proportion of the global and local sensitivity for each entity type and each parameter.

Upon completion of a Sobol analysis, the original simulation data is saved to the disk as a *.pickle* file so that the user can reuse the data and run their own Sobol analyses.

(a) The Sobol settings tab.

(b) Sobol's expected output.

Figure 3.7: Sobol variance analysis settings and output.

Ultimate Analysis Creating a dashboard that can accommodate various inputs is challenging. Predicting the type of plots that a user might be interested in and the type of behavior they want to analyze is impossible to predict. The Ultimate Analysis section does not produce any visualizations or analysis; instead, it allows the user to define which ICs and parameter values they want to simulate (Figure 3.8). The solver will iterate over every possible parameter input and save the results in a *.parquet* file..

Using Dask and the saved *.parquet* file, the user can query for specific runs, for example, runs where a parameter value was greater than 0.05, and use the simulation data to create their own plots.

Figure 3.8: The Ultimate Analysis setup tab.

3.1.4 Custom Visualizations and Analyses

The final part, an optional step, allows the user to define the parameters they want to simulate and download the simulation results. The user can use this data to create their own custom visualizations without having to rerun the simulations, especially if there are many simulations. The data can be further processed and visualized as desired by the user.

As the dashboard cannot create a graph for every situation or be easily adapted to analyze every situation, Section 3.1.3.2 can be used to run and download the simulation data to the disk to create your own custom visualizations later.

3.2 Software Used and Packages

I created the program with Python [63] and extensively use various packages, ranging from standard scientific packages such as NumPy [64] and SciPy to more niche packages, including pickle and SALib [65, 66].

The graphical tool utilizes Tkinter as the front end, handling user inputs, while NetworkX [67] stores the graph and contains the attribute data of the edges and nodes. The GUI tool displays a generated graphical representation of the graph. The creation of the picture utilizes Matplotlib [68] to generate the graph figure.

The simulation framework, the backend of the modeling, makes extensive usage of SciPy's *solve_ivp()* to create the ODE data. It also uses NetworkX to load the graph and parameter values.

The visualization component heavily utilizes Dash and Plotly. Dash hosts the web server and displays the HTML and visualizations while handling input and output. Upon choosing parameter values and clicking "Submit", Dash registers the activity and calls the function registered to the button, sending data such as parameter values and options, like "log x-axis" to the backend server. In the backend, various inputs are handled, such as converting the input string *"0.05, 0.1, 0.15, 0.2"* into an iterable list *[0.05, 0.1, 0.15, 0.2]* that the simulation framework can iterate over to vary the parameter value.

If there are many simulations to run, an intermediate call executes Joblib, a parallel computing program, as seen in Section 3.1.3.2 and Section 3.1.3.2. Joblib parallelizes computations across multiple CPUs to speed up computation time. The ultimate analysis uses Pandas to write the data to a *.parquet* file. Pandas Parquet offers efficient data compression, efficient memory usage, and, when combined with Dask, efficient querying functionalities in a DataFrame format that many data scientists are familiar with.

Sobol uses the SALib library to sample and analyze the parameter input. Both ultimate analysis and Sobol save a *.pickle* file containing a dictionary with the parameter values tested, setting values, and other important information regarding the simulation.

Section 3.1.3.2 uses SciPy's *curve_fit()* function to curve fit the points in the middle plot (Figure 3.5b).

Other packages used in the project include collections, copy, warnings, itertools, os, datetime, json, gc, and time.

Chapter 4

Experiments and Results

In the following section I will apply the software to demonstrate the predicted dynamics of the phages and bacteria under different conditions. I first begin with analyzing the model using a Sobol analysis on the simple Golding model, Equation (2.13). Using the most important parameters identified from the Sobol analysis, I run multiple IVAs to describe the change in graph behavior for different inputs. I use the information gained from this to describe how the graphs change for a wide range of inputs. Next, I analyze if phages will proliferate or not depending on the initial phage, uninfected bacteria, and resource concentration. I finally extend the analyses to a large community and analyze how changing parameter values will change the phage and bacteria growth, and if the phages and bacteria can coexist or not.

4.1 A Realistic Growth Curve

As the bacterial population grows, resource consumption accelerates until only trace amounts remain. The delay between the peaks of uninfected and infected bacteria is due to the infection stages and the latent period of phage infection. Each bacterium transitions from infection stage k to $k+1$ at a rate of $\frac{M}{\tau}$. Therefore, decreasing the number of infection steps M or increasing the latent period τ amplifies this delay. A longer latent period means it takes more time for bacteria to progress through the infection stages.

At $t = 4$, the infection rate surpasses the bacterial replication rate, causing the bacterial population to decline even though resources are still available. This moment coincides with the rise of the phage population. Observing the timing of these events and changes in the graph's behavior, as well as their relationships across different graphs, helps

clarify the complex population dynamics and the interdependence of the populations. The nature of the models being nonlinear makes the analysis harder to understand and conceptualize.

Understanding the system becomes more complicated as the model increases from a $1 \times 1 \times 1$ system to a $p \times b \times r$ system. Now up to any number of phages can interact with any number of bacteria, and any number of bacteria can interact with any number of resources, each with their own unique parameter values. These varying rates will significantly influence the dynamics of the system, and make it hard to determine what event caused what due to the rise in number of interactions. There are only 2 interactions that occur in a $1 \times 1 \times 1$ system. With a $p \times b \times r$ system, there are at most $p \cdot b + b \cdot r$ interactions that can occur. So many individual events occurring at the same time makes it harder to identify the cause of the event, and how the event's action will propagate through the network. A method to circumvent this issue is by knocking out certain nodes or edges and rerunning the simulation and compare the results. The knocked out node or edge won't contribute to the simulation anymore, so there will be a cascading effect on the other populations. Hsu et al. [69] found that predating a specific bacteria strain in a mouse's gut microbiome with a phage had a cascading effect in other bacteria populations due to interbacterial interactions.

4.2 Sobol Sensitivity Analysis Results

The Sobol method is a global sensitivity analysis technique that quantifies the contribution of each input parameter, as well as their interactions, to the variance of a model's univariate output. It decomposes the output variance into fractions attributed to individual parameters and their combinations, providing first-order and total-order sensitivity indices. A Sobol analysis identifies the most influential parameters affecting model output. The insights from this analysis inform the selection of key parameters for subsequent simulations, ensuring that further investigations focus on those with the greatest impact.

Figure 4.1a shows the impact that the parameter had on the final value of the population at $t = 15$ for a $1 \times 1 \times 1$ system, on the original Golding model, Equation (2.13). Figure 4.1b shows the impact that the parameter had on the peak population count, using the 95% rule. Figure 4.1c shows the impact that the parameter had on the time of the peak, using the 95% rule.

The parameters that were tested include all the parameters listed in the basic Golding model, except for the uninfected bacteria $(I_1, \ldots, I_4), M, \omega^i$, and ω^o. Infected bacteria

was not included as it doesn't make sense to start with infected bacteria to the system. M, the number of stages that the infection goes through, can not be tested with Sobol as M is an integer, while Sobol randomly chooses float values. While testing, the washin rate ω^i and washout rate ω^o consistently had the largest influence on the final, peak value, and time of peak value, using the 95% rule. Washin and washout are not part of the original model from Golding, and the addition of a washin and washout term significantly skews the results and analysis, so it was left out of the analysis. The results for a Sobol analysis with washin and washout can be found in Appendix F.1.

4.2.1 Resources

The final value for the resources depended heavily on the initial resource concentration. There weren't many interactions with other parameters because $ST \gg S1$. e had little influence on the system despite e acting as the link between the resources and bacteria and directly controlling the rate of resource consumption. τ had a larger influence in the final resource value than e.

The peak value and time of peak value graphs are empty for the Sobol indices for Resources as the resource concentration is always decreasing from different initial conditions, the initial resource always had a peak at $t = 0$.

4.2.2 Phages

The final phage population value depended on r the most, with β as the second most important parameter influencing the final population value. The other parameters had little to no influence on the final phage population levels.

The Sobol peak value plot is basically the same as the final population value. Similar to the final value, the phage max value is highly dependent on the value of r and β, and the other parameters had little to no influence on the peak value for the phages.

For the time of peak value, τ becomes the most important parameter for determining the time of peak value, while r isn't as important anymore. The initial phage population has a small influence on the final population value, about as equally important as r. β roughly maintains the same sensitivity value across the final, peak, and time of peak analyses.

4.2.3 Total Bacteria

The final total bacteria population depended mostly on β, the burst size of the phage, but via many second or there are many higher order interactions occurring as $ST \gg S1$. The final population depended heavily on many higher order interactions with the initial resource concentration, τ, and e.

β is still the most important parameter to the model, but instead of ST and $S1$ being equal to 1 and 0.28 like in the final value, the sensitivity value is only 0.54 and 0.16 respectively. Every parameter plays some influence on the output, but with higher order interactions as for all parameter inputs, $ST > S1$.

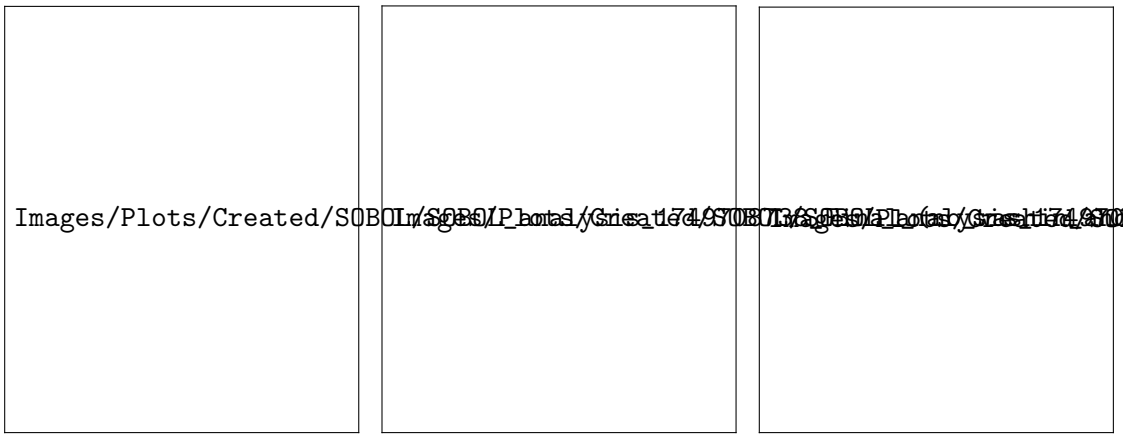
β and τ are the two most important factors in determining the time of peak for the total bacteria. The only parameter that does not influence the time of peak in some manner is e, otherwise every parameter has some sort of influence on the time at which the bacteria population peaks.

4.2.4 Results

Some results are surprising. e, v, and K are consistently the least important factor in determining the final value, peak value, and time of peak. It would be expected that changing the parameter values that directly affect the bacterial growth would have a large impact on the resources and phages. More bacteria mean more resources are consumed, and more phages are created. Knowing that e, v, and K are relatively unimportant compared to a parameter like r or τ, future analyses do not have to focus on e, v, and K. As β, r, and τ are relatively important, future analyses could focus on how those parameters influence the growth of phages and bacteria.

4.3 Graph Behavior with IVA

Knowing how a change in parameter values across a wide range of values affects the curve is important. The sections below quantitatively and qualitatively elaborate how a change in parameter value across multiple values changes the shape of the population curves. Only the most important parameters identified in the Sobol section, along with the initial resource, uninfected bacteria, and phage value will be analyzed. So initial resource, uninfected bacteria, phage, τ, r, and β will be analyzed. The default parameter values can be found in Table E.1, but are given in the text as well. The sections below use an IVA run to analyze the change in behavior across a wide range of values. Each figure from which the conclusions are made can be found in Appendix F.3. ω^i and ω^o



(a) Final value, no washin and washout.

(b) Peak population value, no washin and washout.

(c) Time of peak value, no washin and washout.

Figure 4.1: Sobol analyses for the final, peak, and time of peak value, without a washin and washout rate. The input value ranges to test each parameter used for this Sobol test can be found in Table E.3, except washin and washout is 0.

are not included despite having a large impact as seen in Appendix F.1 because ω^i and ω^o are not a part of the original Golding model.

4.3.1 *R*

The initial resource concentration values were tested across a range from 1 resource to 500 resources, with a default value of 200 resources. The IVA run and results can be found at Figure F.3. All resource values are depleted at around the same time, within 1 time unit of each other. As the resource concentration increases, the time it takes for the uninfected bacteria to peak goes from 0.5 time units to 4.12 time units. The infected bacteria face the opposite, where it as the resource concentration increases, the time to peak goes from 9.5 time units to 4.9 time units. The same behavior is exhibited for the phages, except the time goes from 14 time units to 6.3 time units. The phage population showed a huge variety in max population, ranging from a max of 464 phages to 17,200 phages, a 37x difference. The max bacteria sum population varied from 55 to 3106 total bacteria, a difference of 56.5 times. Figure F.3.

4.3.2 *U*

The initial uninfected bacteria population values were tested across a range from 1 uninfected bacteria to 100 uninfected bacteria, with a default value of 200 uninfected bacteria. The IVA run and results can be found at Figure F.4. As the initial uninfected bacteria population increases, the resources are consumed faster, the bacteria populations grow faster and peak earlier. The phages were able to grow faster. The phages peaked at

5.7 time units for 100 initial uninfected bacteria, while for small initial uninfected, the phages peaked at 10.3 time units. Varying the uninfected bacteria had little influence on the max value reached. There is a difference of 600 phages between the phages that started with 1 uninfected and 100 uninfected bacteria. The difference in total bacteria peak values is 173. Changing the uninfected bacteria had little influence on the peak values, but had a large influence on the time of peak value.

4.3.3 P

The initial phage population values were tested across a range from 1 phage to 50 phages, with a default value of 10 phages. The IVA run and results can be found at Figure F.5. The differing phages values had almost no influence on resource consumption, and limited impact on the peak value and time to peak value for the bacteria and phages. The difference between the peak and time of peak values is minor. The final phage population at $t = 15$ was 9436 for an initial phage value of 50, while for an initial phage value of 1, the final phage value reached 10,676. The time of peak difference is just 1.44 time units. As the phage value increases, the time for the uninfected, infected, and phages to reach their peak decreases in time.

4.3.4 τ

The τ parameter values were tested across a range from 0.5 to 3.5, with a default parameter value of 0.7. The IVA run and results can be found at Figure F.6. τ has no influence on how fast the resources are being consumed. But as τ increased, the time for the infected bacteria population to peak went from 4.43 to 11.21 time units, and the time it took for the phages to peak went from 5.49 to 14.70 time units, a 2.53 times and 2.68 times increase in time length. Smaller τ values resulted in larger final phage populations and smaller total bacteria populations. The peak value and time of peak value for varying τ values shows little difference for the bacteria sum, ranging from a peak of 1,444 to 1,702 total bacteria and time of peak of 3.74 to 3.82 time units, a difference of 0.08 time units. As τ increases, the infection process will take longer, and it will take longer for the bacteria to die. Dying later means the phage's population experiences a delay in growth, taking longer to grow and peak in population value. More bacteria can grow in the meantime, and consume resources. The bacteria will take longer to peak as there is less initial pressure from the phages.

4.3.5 r

The r parameter values were tested across a range from 0.001 to 0.2, with a default parameter value of 0.001. The IVA run and results can be found at Figure F.7. For small r values, all resources are consumed by $t = 5$, while for large r values only 13.12 resources were consumed as not enough bacteria were created throughout the simulation to consume all the resources. As r increases, the time to reach the max value decreases for the uninfected and infected bacteria, and phages. The delay in phage value went from 6.32 to 2.35 time units, and the total bacteria went from 3.81 to 0.59 time units. However, for large r values, there was little phage and bacteria growth. For large r values, there was a max of 154 phages and 79 bacteria, while for small r values there was a peak of 10,464 phages and 1,588 total bacteria. As r increases, the probability of a successful infection increases, increasing the adsorption rate of phages and causing the phages to grow and peak earlier. This higher adsorption rate causes the bacteria to be infected faster, causing them to die faster. Lower r values gives more time for the bacteria to grow, which gives the phages more bacteria to infect, leading to higher final phage populations at the end.

4.3.6 β

The β parameter values were tested across a range from 1 to 100, with a default parameter value of 10. The IVA run and results can be found at Figure F.8. For large β values, not every resource was consumed, similar to large r values, where for $\beta = 100$, 89 resources were consumed. As β increases, the time to peak for the infected bacteria and phages temporarily increases before decreasing. For the uninfected bacteria, the time to peak (does not change for $\beta = 1$ to $\beta = 10$) takes 3.8 time units, but for larger β values, the time to peak decreases to a minimum of 1.94 time units. There was a large difference in phage population values, ranging from 4.54 final phages to 42,570 phages, a 9,376 times increase in phage value. As β increases in value, more phages are being created upon lysis, which infects the bacteria. For β less than 10, the bacteria growth is restricted by latency, how fast a bacterium can go through the infection process, while for β greater than 10 the system changes to an adsorption limited regime, where the phages can't adsorb to the bacteria fast enough, allowing bacteria to grow faster and peak earlier. As more phages are being produced, more bacteria are being infected, so the bacteria population starts to die out earlier, with fewer bacteria being produced. With fewer bacteria, fewer resources are being consumed.

4.4 Initial Value Analysis Results

Figure 4.2a and Figure 4.2b illustrate how varying the initial uninfected bacteria population from 1 to 500 (using 100 different starting values) affects the dynamics and time of peak population of phage and total bacteria populations using the 95% rule.

Figure 4.2a perfectly replicates Figure 1 of Mulla et al. [46]. As the initial bacteria population increases, the time to reach the phage and bacteria sum peak decreases, following $y = -0.8648 \cdot ln(x) + 9.7911$ and $y = -1.0056 \cdot ln(x) + 7.7626$, with $R^2 = 0.9800, 0.9988$ respectively.

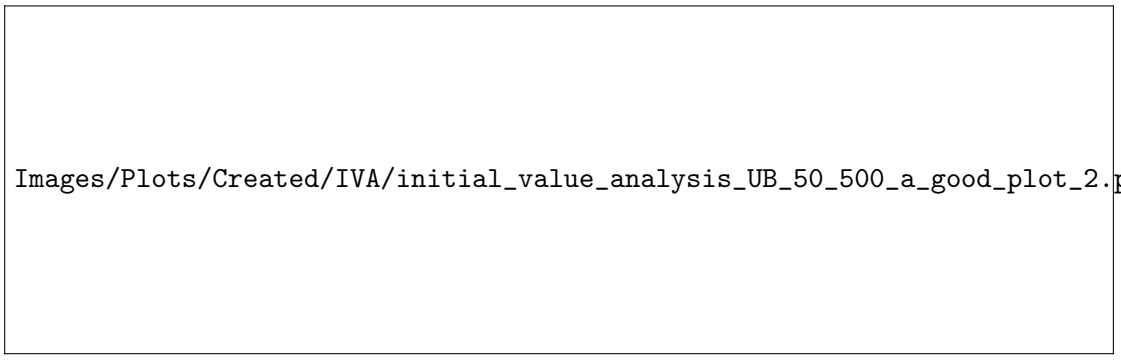
Figure 4.2b on the other hand shows different behavior. As the initial bacteria population decreases from 500 to 100, Figure 4.2a exhibits the same behavior. There is a change in behavior at 100 and less initial uninfected bacteria. Instead of following the predicted line like in Figure 4.2a, the curve for the phages decrease non-monotonically. The bacteria plateau before starting to increase again.

The fitted linear regression curves follow $y = -0.1292 \cdot ln(x) + 10.1462$ and $y = -0.6234 \cdot ln(x) + 6.9602$, with $R^2 = 0.5406, 0.9206$ respectively. The slope tells us how well the uninfected bacteria has an influence on the time of peak value. The larger (positive or negative) the slope is, the more impact the uninfected bacteria had on the time of peak value. The closer the R^2 value is to 1, the more proportion of the variance for the time to peak value is explained by the uninfected bacteria. The linear regression curve can not explain the change in limiting regions. So for the first curve, the uninfected bacteria had a small influence on the time of peak value, while with the second curve, changing the uninfected bacteria had a larger impact on the time of peak and is considerably more important in explaining the time of peak value.

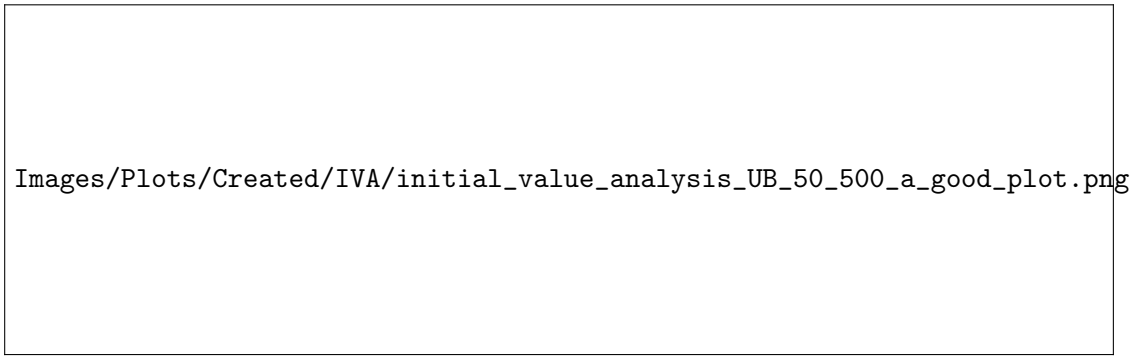
4.5 Phage Proliferation

4.5.1 Phase Portrait

Figure 4.3a shows a phase portrait varying the initial resource and phage concentration. The same initial phage values have the same color for the line. For phages that start above 25.98, the phage population can proliferate until the washout would eventually remove the phages. For phage populations that start below 25.98, the washout removes the phages before the phages had time to infect and kill the bacteria. Both regions of phages exhibit consistent behavior, of either going to 0 or proliferating. If the phage



(a) IVA for Table E.2. Replicates Figure 1 of Mulla et al. [46]. The system is adsorption limited [46].



(b) IVA for Table E.1. For uninfected bacteria less than 100, the phage-bacteria interaction is resource limited.

Figure 4.2: Varying the initial uninfected bacteria concentration, from 50 to 500, with 30 unique values tested. Varying the default parameter values a can have a large influence on how changing the initial bacteria concentration influences the dynamics of the system. The default values for Figure a) and b) can be found at Table E.1 and Table E.2.

population started at exactly 25.98, if the initial resources was 260 or above, the phages died out. If the initial resource value was 255 or below, the phages proliferated.

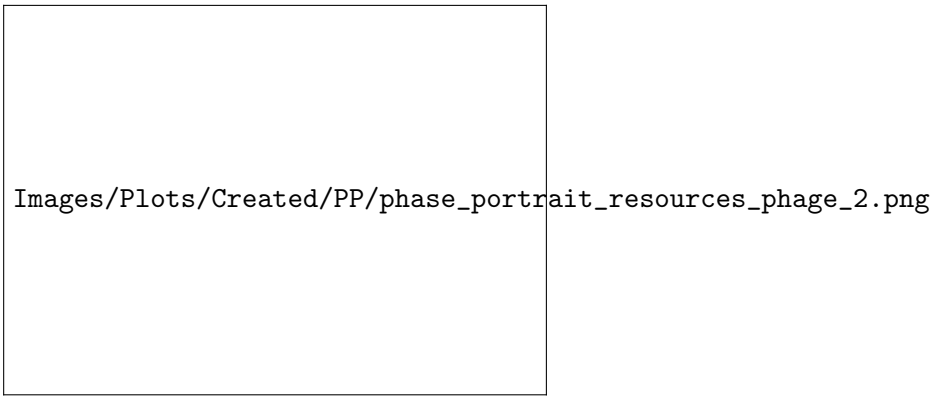
4.5.2 An Initial Phage and Resource Analysis for Phage Proliferation

Figure 4.3b expands on the phase portrait by simulating more values and coloring the square depending on if the phages proliferated or not. The initial resource values span from 1 to 500, and the initial phage values range from 25.5 to 26.5, each with 100 unique values sampled.

Figure 4.3 zooms into the range $(1-40, 24.2-25)$ for a high detailed view of the behavior happening around initial resources of 10. From 1 to around 7 initial resources, fewer phages are needed to ensure proliferation. At 7 initial resources, there is a minimum in the phage proliferation boundary. From 7 initial resources and upwards, as more resources are added to the system, more phages are needed to ensure proliferation. Despite this, the change is tiny, a difference of about 2 phages. Considering the range of

(a) Zoomed in plot of a phase portrait with varying resource and phage population from 245-265 and 25.5-26.5 respectively. Each row has its own line color. Diverging behavior can be seen for the orange line (phage=25.98).

(b) Phage population proliferation as a function of initial resource and phage concentrations. While the color appears uniform along the vertical axis, each cell is actually a slightly different value. The phage-resource proliferation boundary follows a fitted Monod equation.



(c) Zoomed in to analyze the regime of behavior change near resources= 10.

Figure 4.3: Varying initial resources and initial phages and the resulting proliferation and fitted proliferation curve. The box is colored red if the phages proliferated for that condition, and white if the phages died out. Phages proliferated if they reached 2 times their initial population at any point in time in the simulation. This simulation used the values from Table E.2, but with washout set to 0.02 instead of 0.

possible initial phage populations, the phage proliferation boundary is essentially flat. Under these parameter values, choosing an initial phage population of 27 or higher will ensure phage proliferation. The higher the initial resource concentration, the more final phages will appear. As there are more resources, more bacteria can grow from the resources, which in turn allows more phages to grow.

If there is a higher washout, similar behavior is observed where the phage proliferation boundary exhibits a similar shape to that of Figure 4.3b, except more phages are needed to proliferate. If K is increased to a larger value, the minimum in the proliferation boundary is shifted to the right.

4.5.3 An Initial Phage, Bacteria, and Resource Analysis for Phage Proliferation

The initial resource concentration had some, but very limited impact on if the initial phage concentration would affect if the phages proliferate. Within the context of the basic Golding model, the initial uninfected bacteria population is one of three parameters that a researcher can easily control, with the other two being the initial resource and initial phage. For low initial uninfected bacteria populations, it will be harder for the phages to proliferate. There is not enough bacteria to infect before the washout will remove the phages from the system. While for large initial uninfected bacteria populations it will be easier for the phages to proliferate such that the washout won't immediately wash the phages out. So I extend the initial resource and phage population analysis by adding a third dimension to the analysis, the initial uninfected bacteria population. The aim of adding the uninfected bacteria is to see how the initial uninfected bacteria will 1) affect if the phage can proliferate, and 2) affect the max population that the phages can reach.

Figure 4.4 has three axes, the initial phage, resource, and bacteria population value. The initial uninfected bacteria did not have a significant impact on the 1) if the phages would proliferate, and 2) by how much the phages would proliferate. If spliced along the bacteria axis, there is little difference in the shape of the curve. It was expected that as the uninfected bacteria count would increase there would be significantly fewer phages that would be needed to ensure phage proliferation. However, there is not a significant change in if the phages proliferated and by how much.

This suggests that under a washout situation, ensuring that there are enough phages is the most important factor to ensure that the phages are not washed out.

4.6 Plotting Parameter Change — $3 \times 2 \times 3$ Model

Now that we have identified the most important parameters in the Golding model, we can analyze how the curve shapes change across a range of parameters for a larger model. Knowing how multiple parameters influence the output is important. As the complexity of the model input increases, interactions, and their parameter values play an important role in determining which phage or bacteria can survive. The interactions (or lack thereof) will

The larger model will exhibit different behavior than a $1 \times 1 \times 1$ model due to the many interactions. The differing parameter values across each interaction will influence how

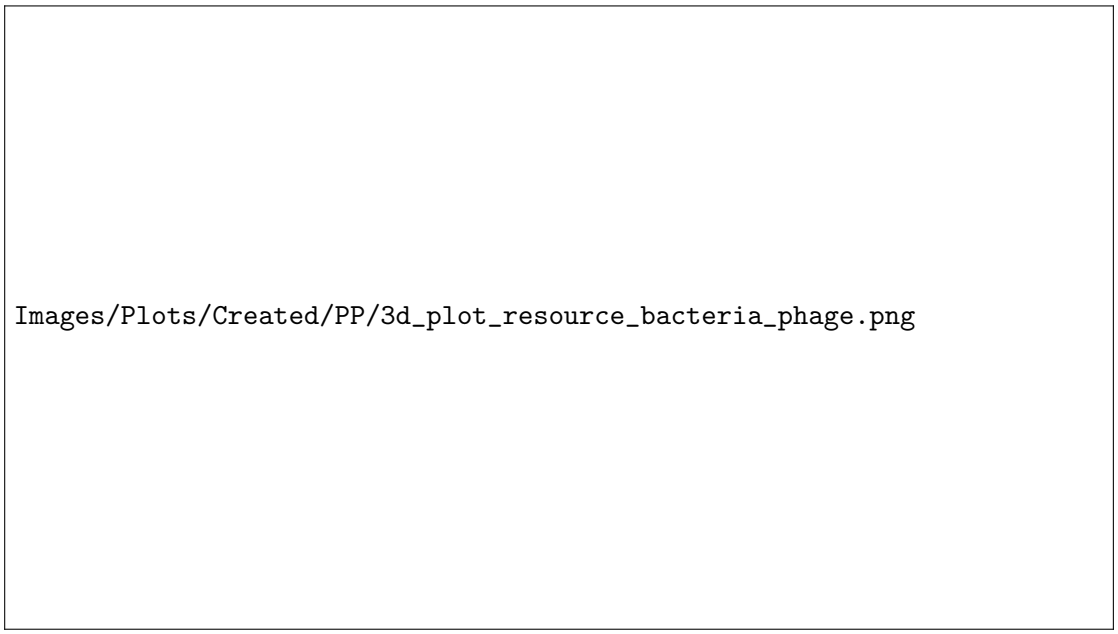


Figure 4.4: 3D plot of phage proliferation, dependent on initial resource, uninfected bacteria, and phage population. Color scaling from white to red, color is dependent on the max phage population reached.

fast each population can grow and die. A $3 \times 2 \times 3$ model was chosen as it is on the boundary between adding more phages, bacteria, or resources would clutter the plot with lines, while still offering behavior that can be compared against one another. The graph network that was used can be found at Figure 3.1b, with the default parameter values found at Table E.4. B_0 is infected by P_1 and P_2, and consumes R_0 and R_1. B_1 is infected by P_0 and P_2, while consuming R_2.

Figure 4.5a, Figure 4.5b, and Figure 4.5c, show a 7×7 matrix of subfigures across washout rates of 0, 0.02, and 0.05 Each subfigure uses a different combination of r and β parameter values.

All initial phage values started at 10. This was specifically chosen to show how even though the phage values all start the same, the different parameter values and interactions ultimately influence the population growth.

If r or β is equal to $Original$, then the simulation uses the original parameter values as defined in Table E.4, otherwise each r and each β parameter interaction has the value listed in the subfigure title.

The columns and rows of each figure show how a change in parameter value affects the curve, while keeping the other parameter the same. In $r, \beta, \omega^o = Original, Original, 0$, although not a realistic growth curve, shows how the different parameter values for each interaction uniquely affect the growth rate of each entity, especially the phage population (P_0=blue, P_1=green, and P_2=purple). Despite all phages starting at the

same population level, within the first two or so time units, P_1 has fewer phages than P_0 and P_2. P_2 has the fastest initial growth rate, as P_2 has the most phages until $t = 4$, at which point P_1 has a larger phage population. P_2 reaches its peak population count before P_0 or P_1, but despite the slower initial growth, P_0 and P_1 eventually overtake P_2 in total phage population. P_2 also reaches its peak before decreasing in population. Since the phage population is reduced by $r_{pb} \cdot (U_b + \sum_{k=1}^{M} I_{b_k})$, and by specifically choosing the parameter values as used in Table E.4, behavior that hasn't been seen in a $1 \times 1 \times 1$ system has been found. The complete extinction of the bacteria has been delayed long enough such that at trace amounts, there is phage reduction despite bacteria still existing. The peak times for P_0, P_1, and P_2 are $t = 6.33, 7.99, 4.52$, a difference of 3.47 time units.

Contrast that with the phage population dynamics of that with $r, \beta, \omega^o = Original, 100, 0,$ the phage populations do not show notable dynamics. The peak times are more similar and consistent to one another ($t = 5.50, 7.01, 6.78$, a difference of 1.51 time units). The phage population curve all appear the same, with slightly slower growth rates. There is no crossing of phage populations unlike with $r, \beta, \omega^o = Original, Original, 0$.

The highlighted example demonstrated the dynamics and influence that multiple agents can have on the final output.

The top row of Figure 4.5b shows how the phages and resources died out relative to the top row of Figure 4.5a. Even with a high burst value, the phages could not defeat the pressure from the washout. But by changing the r value from $r = 0.001$ to $r = 0.041$, the phages were able to save themselves and proliferate.

4.7 Phage and Bacteria Survivability Analysis For A $20 \times 20 \times 10$ System

In the context of large communities, it is important to understand how different conditions will influence if a phage or bacteria population will survive. Different interactions and environmental factors will influence how the community evolves. Sometimes populations of phages or bacteria will die out due to external factors, either that be a due to a lack of resources or bacteria to consume or infect, natural degradation, or external factors like UV light.

Using the simulation framework, I created and analyzed a $20 \times 20 \times 10$ system using the extended Golding model. I selected two parameters, τ and β, to run a survivability analysis using the extended Golding model. A phage survived if its final population is greater than 1 at the end of the simulation. A bacteria population survived if the final population for the uninfected bacteria was greater than 1.

(a) Washout $\omega^o = 0$.

(b) Washout $\omega^o = 0.02$.

(c) Washout $\omega^o = 0.05$.

Figure 4.5: Varying washout ω^o, r, and β. The default values for the parameters can be found in Table E.4. All initial phage population values were set to 10.

Each phage is guaranteed to interact with at least one bacterium but no more than two bacteria. Each bacterium interacts with at least one phage and one resource, but not more than two phages and two resources. Every resource interacts with at least one bacterium, and at most three bacteria. The parameter values were randomly selected from a uniform distribution in the Sobol analysis value ranges (Table E.3).

Figure 4.6 shows the phage and bacteria survivability matrix. It appears as if there is an inverse relationship between phage and bacteria survivability. If the phages survived, the bacteria died out. If the bacteria survived, the phages did not survive. The phages only died out with small burst sizes, β, while otherwise would consistently survive. The small burst size ensured that the phages could not proliferate in time. Once the burst

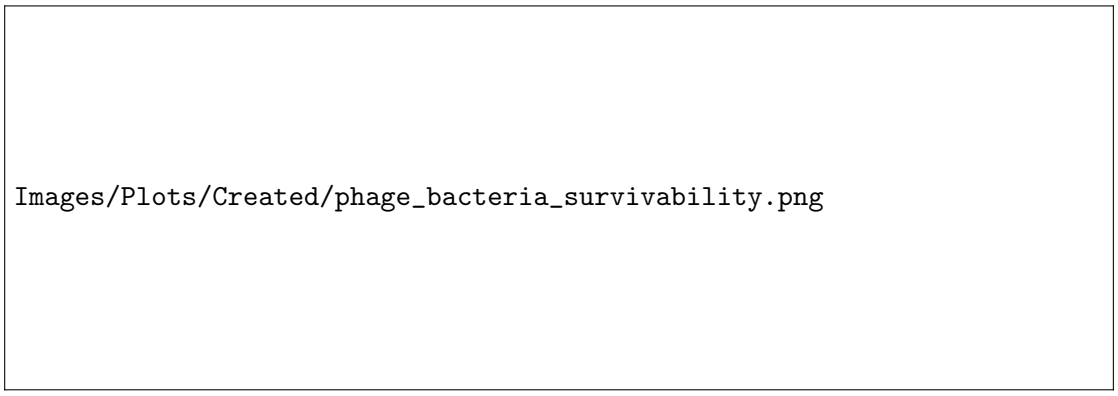


Figure 4.6: The output graph for the default parameter values for a large $20 \times 20 \times 10$ network. Parameter values were randomly chosen in the interval given by Table E.3.

size became large enough, around $\beta = 3$, more than half of the phages were able to survive. The bacteria only survived with small burst sizes β. For large burst sizes, the phages would grow too fast and infect the bacteria.

τ had less of an influence in survivability than β did, except for when the β values were less than 20. As τ increases, fewer phages survive and more bacteria survive. τ determines how fast a bacterium goes through the infection process, so as τ increases fewer phages will survive. The bacterium takes longer for the infection process to complete, so there is a greater chance for the phages and infected bacteria to be washed out before more phages can be created.

For the most part, the competitive exclusion principle is respected. The majority of the simulations resulted in less than 10 uninfected bacteria surviving. For small burst sizes, there were more than 10 surviving bacteria, however if the simulation length were to be extended it would be expected that more bacteria populations will start to die out.

Chapter 5

Discussion

This section presents an analysis and discussion of the results.

5.1 Graph Behavior

The behaviors shown represent typical trends observed when varying each parameter in the specified direction, but they may not apply to all possible values or reflect the magnitude of changes. Additionally, these results do not necessarily generalize to scenarios where two parameters are simultaneously varied. The graph behavior should be interpreted alongside the local $S1$ sensitivities from Section 4.2 to better understand how sensitive the output is to specific parameters and the potential impact of their variation.

A researcher would look at the growth curve and create their conclusions by visually looking at the curves and reasoning through the ODE model to explain the behavior. They can further supplement this analysis with a Sobol analysis to quantify how significant the change was, and compare a parameter's significance with other parameters.

5.2 Realistic Growth Curves

As the complexity of the network grows, biologically accurate curves might disappear. In a $1 \times 1 \times 1$ system, like that in Figure 2.3, there is a clear exponential growth, peak, and death cycle for the bacteria. The phages experience a delay in growth, but also exhibit exponential growth. These behaviors can disappear as the community size grows to a $p \times b \times r$ system. Figure 5.1 shows how these behaviors disappear. Some, but not every phage or bacteria in Figure 5.1 exhibits a clear growth and death cycle. Bacteria face increased competition for resources, and phages face increased competition for bacteria.

This increased competition decreases the pool of available resources for the bacteria and phages to grow. There is an external threat, the washout, so any bacterial and phage growth gets cancelled out by the washout, or even worse, if the growth is not fast enough, completely eliminated.

There are only 2 interactions that occur in a $1 \times 1 \times 1$ system. With a $p \times b \times r$ system, there are at most $p \cdot b + b \cdot r$ interactions that can occur. Any number of phages can interact with any number of bacteria, and any number of bacteria can interact with any number of resources, each with their own unique parameter values. These varying rates will influence the dynamics of the system, and make it hard to determine what event caused what due to the rise in number of interactions. With so many individual events occurring at the same time, it makes harder to identify the cause of the event, and how the event's action will propagate through the network.

There are ways to circumvent this issue. Simulations offer the ability for researchers to experiment with the system that normally might not be feasible. Knocking out certain nodes or edges and rerunning the simulation would allow the researcher to see how the knocked out node or edge influenced the simulation. The knocked out node or edge won't contribute to the simulation anymore, so there will be an impact on the population count. The larger the difference, the more important that that node or edge had on the simulation.

5.2.1 Knockout

Figure 5.2a and Figure 5.2b show the difference that a knockout has on the output of the graphs. P_2 and B_14 were knocked out of the network, using Figure 3.1c as the base network. On average, there are more resources, as there is less competition for the resources. In the knocked out graph, more bacteria were able to survive, and in total there were more bacteria in the system as evident by the bacteria sum. There are also less infected bacteria. With the phages, there is less of an obvious difference in behavior, but in Figure 5.2a, P_18, pink, suddenly increases in population at $t = 10$ until the end of the simulation, while this doesn't happen in Figure 5.2a. P_18 infects B_14 (as well as B_13), so by removing B_14 P_18 was negatively affected.

By removing P_2 and B_14, it was casually shown how the resource, bacteria, and phage dynamics were permanently altered. Figure 5.2b showed how knockouts have a lasting impact on the dynamics of the system, changing which phages and bacteria survive.

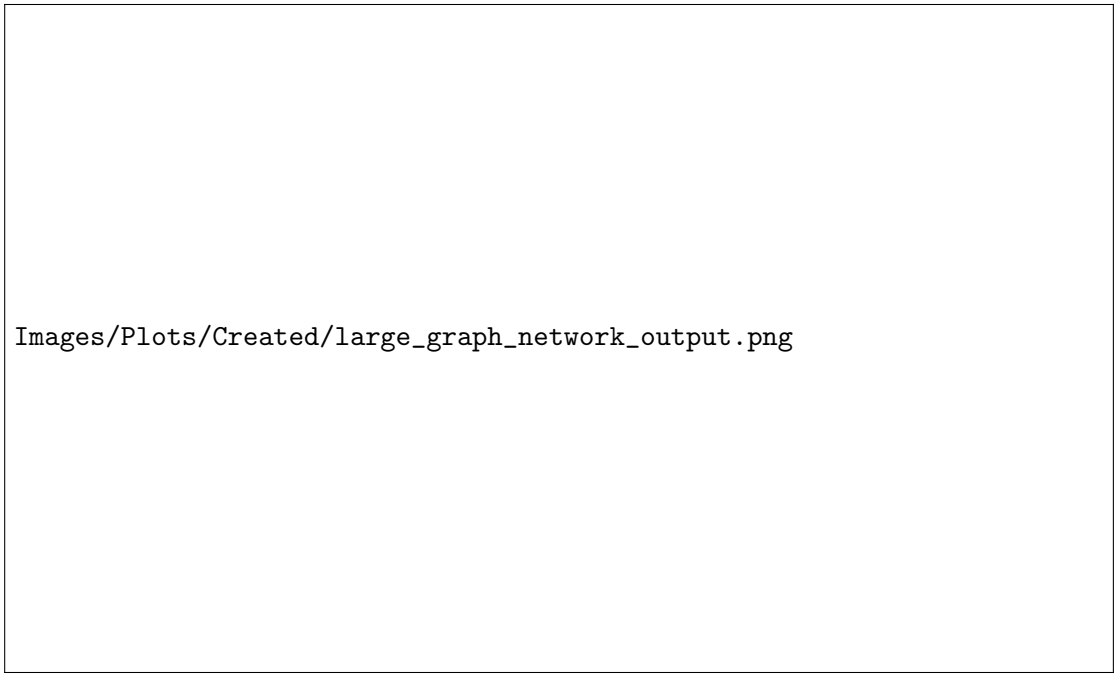
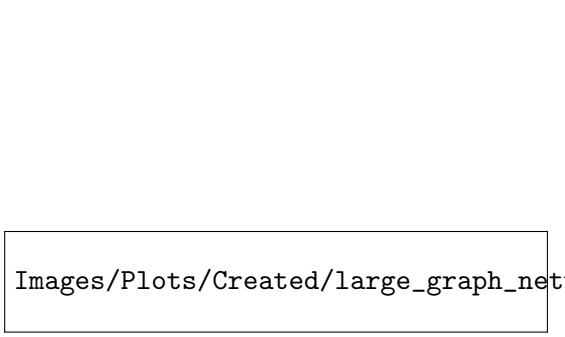
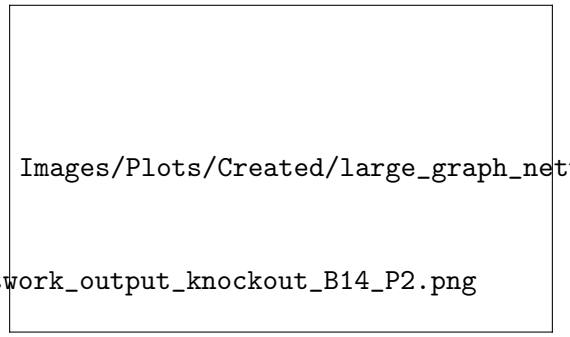


Figure 5.1: The output graph for the default parameter values for a large $20 \times 20 \times 10$ network. Parameter values were randomly chosen in the interval given by Table E.3.



(a) The same figure as Figure 5.1, but with phage P_2 and bacteria B_14 hidden from the graph.



(b) The large $20 \times 20 \times 10$ network, with phage P_2 and bacteria B_14 knocked out from the system.

Figure 5.2: Comparing the effect of knocking P_2 and B_14 out from the system. Figure a) has P_2 and B_14 hidden from the system, while Figure b) has the phage and bacteria knocked out from the system. The most important phage and bacteria as noted by the bacteria growth were knocked out. Figure 3.1c is used as the reference network model. The colors between a) and b) can not be directly compared as the color is not mapped to a specific phage or bacteria.

5.3 Sobol Sensitivity

A researcher might have an intuitive understanding of how a model works, but might be unsure of how important the parameters of the model has on the model results. A researcher can use Sobol to quantify how important a change in parameter value is in the context of the simulation results. By testing and measuring hundreds of simulation inputs and outputs, Sobol understands how changing a parameter value would influence the output of the parameter. By identifying the important parameters, a researcher could use that to their benefit in their research. Researchers can prioritize adjusting the most influential parameters in their simulations or consider engineering phages and bacteria to achieve the desired parameter values.

5.3.1 Resources

Every parameter affects how the bacteria and phage population grows in some way, which influences the consumption rate of resources, which determines the final resource value. Of the other parameters, τ has the biggest influence because τ determines how fast the bacteria will go through the infection process. The longer it takes, the longer it takes for the phage population to grow, allowing more bacteria to grow and consume resources.

5.3.2 Phages

The r value quantifies how fast the phages infect the uninfected bacteria. r, the adsorption rate of phages to bacteria, can be interpreted as the efficiency of infection. The smaller the value, the more efficient the infection process is, and fewer phages it requires to infect a bacterium. With a larger r value, more phages are used to infect a bacterium. β has an influence on the final phage population, as the infected bacterium will release β phages into the system. The more phages that are created for every lysed bacterium, the more phages are available in the system.

The barplot values for the peak value for phages using the 95% rule has the same values as in the final value plot for the phages. This makes sense as there is no removal of phages from the system, so any phages created by r or β will stay in the system. As the population is ever-increasing, the final and peak value will be occurring near one another and are tightly associated with one another.

The τ parameter influences how fast an infected bacterium goes through the infection process. Decreasing tau increases the speed of lysis, allowing more phages to be produced

faster. This in turn will increase the phage population faster than other parameters, meaning that the phages reached the peak population faster. r and β have similar effects, but rather have a larger effect on the final and peak phage population than the time it takes to reach the peak value. r and β add to the phage population, rather than specifically speed up a process which τ does. Decreasing r and increasing β lead to more phages being created, which can then infect more bacteria. This won't have nearly as large of an impact as shortening the infection period, literally decreasing the time until new phages are created, thus causing the phage population to reach its peak faster.

5.3.3 Total Bacteria

Resources, τ, e, and β all play a critical role in the final population value of bacteria. More resources allow bacteria to grow for a longer time, allowing more bacteria to be created. β has a very important role in determining the final bacteria population, but surprisingly r has no role in determining the final bacteria population. e determines the resource consumption rate. The larger e is, the faster the depletion rate of the resource. So by lowering e, more bacteria will be created, and the time at which the bacteria peak at occurs later in time. Of course as β changes value, it will have a large influence on the final bacteria population. More phages will cause more infections, slowing the spread of bacteria. However, r surprisingly does not have an influence on the final bacteria population level.

The total bacteria peak population is much more sensitive to the different parameter values. The bacteria act as a link between the phages and resources. Any changes in the resource consumption rate or initial condition will affect the bacteria population. Likewise, any change in phage adsorption rate will affect the bacteria growth, which in turn will affect the resource consumption rate. The bacteria have the ability to dampen the effect of certain parameters. For example, the initial resource value affects the final resource value. The initial resource value partially affects the final bacteria value, but it doesn't affect the phages. The parameters lose strength as it propagates through the system.

Since the bacteria exist in the middle between the phages and resources, the bacteria are exposed to many different parameters that will ultimately influence the peak value. Many of these parameters are interacting with one another hence why $ST > S1$ is true for many of the inputs.

Similar to the peak value, the bacteria interact with many parameters, who interact with other parameters. The time of the peak can only occur between $t = 0$ and the end of the simulation, limiting the values that can be measured, reducing the potential

variance seen in the output. The peak value on the other hand has no limit on the peak value, with a peak value occurring anywhere between 0 and ∞. e and K depend almost exclusively on higher order terms due to the nature of the bacteria growing at the Monod rate.

5.4 Initial Value Analysis

The behavior between Figure 4.2a and Figure 4.2b should be similar, however the change in parameter values altered the simulation to introduce a region in behavior change. It would be expected that for 100 initial uninfected bacteria and less the bacteria sum peak time would follow the linear regression line, but at around 100 uninfected bacteria and less, the peak curve deviated from the linear expression.

Between 100 and 500 uninfected bacteria, the system is adsorption limited. The adsorption of phages to bacteria depends on the bacterial concentration [46]. There are not enough phages relative to the population to infect the bacteria, so the phages slowly adsorb to the bacteria. The bacteria can grow without immediate pressure from the phages. There is also a lack of resources which is restricting the bacterial growth. For large initial bacteria concentrations, the resources run out. This severely limits the ability for the bacteria to grow and artificially limits the population cap of the bacteria.

The system is latency limited between 25 and 100 uninfected bacteria. The collapse time in a latency limited regime is independent of the initial bacteria population [46]. As the initial bacteria decreases, $r \cdot \beta$ eventually becomes smaller than $1/tau = \frac{1}{2.14} = 0.467$, that's around where the transition happens - see Yuval's paper supplement.the phage to bacteria ratio increases, and they can infect the bacteria faster. So the time to phage peak also decreases as the initial uninfected bacteria decreases.

As the uninfected bacteria decreases from 25 towards 1, it takes longer for the bacteria to grow and reach their peak population count. At these initial bacteria concentration levels, there is enough resources to fully sustain the bacteria through the whole simulation. For uninfected bacteria less than 25, the system enters a new sort of restriction, where the system experiences a delayed in infection due to low encounter rates.

The transition rate from U to I_1 is proportional to $U \cdot P$. Fewer bacteria are initially infected with a low initial starting bacteria. If the phages can't infect the bacteria, the later infection stages are delayed due to the slow infection process. There could be a threshold for phage to uninfected bacteria where a certain dilution rate will significantly affect the peak time. This value could be around $\frac{\text{initial phage}}{\text{initial bacteria}} = \frac{10}{25} = 0.4$ as at around 25 uninfected bacteria is when the system switches from latency limited to the new

limited regime. The bacteria have a longer amount of time to grow as there is a low infection rate and plenty of resources to consume. This means that it takes longer for the bacteria to grow, as noticed by the increase in peak time relative to larger initial uninfected bacteria populations. As the bacteria population drives the phage population, an increase in bacteria time of peak causes an increase in phage time of peak.

5.5 Phage Proliferation

5.5.1 Phase Portrait

There is non-linear trade-off between initial resources and initial phages when there is washout included. The washout non-linearly affects if the phages proliferate or not. The bigger the washout value, the harder it is for the phages and bacteria to proliferate. The bacteria couple the phage populations to resources, so changes in initial resource concentration will have an effect on the final phage value. Increasing the initial resource concentration creates more bacteria, while decreasing it results in fewer bacteria. More bacteria ultimately produce more phages, while fewer bacteria ultimately result in fewer phages. While Sobol with washin and washout, Figure F.1b, showed that the final value for phages due to changes in initial resource input values is limited (sensitivity ≈ 0.1), it still has an impact.

For low initial resource concentration values, values below 10 the resource, the Monod curve is below the half-velocity constant (the velocity v is 1 and K is 10). The lack of resources restricts the bacteria growth, and the bacteria can't grow quickly. As the resource concentration increases towards $K = 10$, the bacteria can grow faster. Since K is small, a small change in R causes a relatively large change in the Monod rate.

At around the minimum in the phage proliferation boundary, the behavior changes. The minimum happens at $R \approx 7$, the Monod rate reaches half of its max velocity. This behavior should be seen at around 10, but the washout has an effect on the Monod rate by artificially shifting the point where the half velocity occurs from $K = 10$ to $K \approx 7$. $10 \cdot 0.98^{15} = 7.386$

As R increases from K, the bacteria are not limited by the nutrients anymore and can grow faster. However, as R increases beyond K, each additional unit of R results in a diminishing increase in the Monod rate, which asymptotically approaches its maximum velocity. As bacteria grow according to the Monod equation $g(N, v, K)$, the phage population dynamics remain tightly coupled to bacterial growth.

Phage proliferation becomes a race against time under external pressure. The phages will not proliferate if the phage growth rate is not fast enough to initially beat the washout removal rate, or the infected bacteria are washed out before lysing.

5.5.2 3D Plot

It is difficult to see inside the matrix, but using the color on the outside can give some insights into the behavior happening inside the matrix. Even with the added bacteria, the phage proliferation boundary is still heavily dependent on the initial phages and initial resources, and less so on bacteria.

5.6 Plotting Parameter Change

5.7 Phage and Bacteria Survivability

There are so many interactions and parameters in large and complex systems that it becomes hard to analyze. The model network and parameter values are relatively random, and a figure of the network interactions and a copy of the parameter inputs is needed in order to have some sort of chance to analyze why a phage or bacteria is behaving in a certain way. Further, it is hard to simplify the model, for example by knocking out unimportant bacteria from the system, due to the interconnected nature of the system.

Since their discovery, bacteriophages have been traditionally regarded as the natural enemies of bacteria. However, recent advances in molecular biology techniques, especially data from "omics" analyses, have revealed that the interplay between bacterial viruses and their hosts is far more intricate than initially thought. On the one hand, we have become more aware of the impact of viral predation on the composition and genetic makeup of microbial communities thanks to genomic and metagenomic approaches. Moreover, data obtained from transcriptomic, proteomic, and metabolomic studies have shown that responses to phage predation are complex and diverse, varying greatly depending on the bacterial host, phage, and multiplicity of infection. Interestingly, phage exposure may alter different phenotypes, including virulence and biofilm formation. The complexity of the interactions between microbes and their viral predators is also evidenced by the link between quorum-sensing signaling pathways and bacteriophage resistance. Overall, new data increasingly suggests that both temperate and virulent phages have a positive effect on the evolution and adaptation of microbial populations. From this perspective, further research is still necessary to fully understand the interactions between phage

and host under conditions that allow co-existence of both populations, reflecting more accurately the dynamics in natural microbial communities.

Phages have traditionally been seed as the enemy of bacteria, but with new trancriptomic, proteomic, genomic, and metabolomic methods, we have a better understanding of hoe phage predation infleunces

Overall, new data increasingly suggests that both temperate and virulent phages have a positive effect on the evolution and adaptation of microbial populations. From this perspective, further research is still necessary to fully understand the interactions between phage and host under conditions that allow co-existence of both populations, reflecting more accurately the dynamics in natural microbial communities.

5.7.1 Ecology

With the competitive exclusion principle in ecology, there should be at most 10 surviving bacteria. Two or more species competing for the same resources can not coexist indefinitely.

The R* resource-ratio hypothesis rule is a rule that states that a species needs to be the best at consuming resources to persist. The growth rate of a species is proportional to the resources consumed with an energy factor, minus a maintenance and death rate. The bigger the resource overlap between competing bacteria, the smaller the realm of co-existence [70].

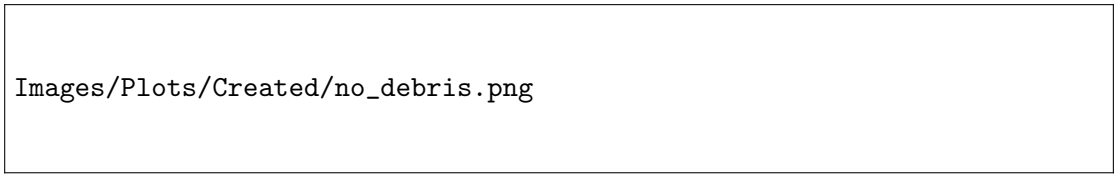
5.7.2 Debris Term

A recently published paper Dey et al. [71] uses the Golding model but adds a new term, debris. While running their simulations and experiments, the model's results would diverge from their experimental work. Their thinking is that freshly lysed bacteria still have biomarkers that phages can detect and attach to. Incorporating the debris term, which acts as an additional sink for phages, improved the model's alignment with the experimental data. The debris term can also encompass bacterial phage defenses (Section 2.2). The phage equation from the Golding model can be rewritten as

$$\frac{dP_p}{dt} = \sum_{b \in B} \beta_{pb} \cdot \frac{M}{\tau_b} \cdot I_{b_M} - r_{pb} \cdot (U_b + \sum_{k=1}^{M} I_{b_k}) \cdot P_p - w^o \cdot P_p - d_{p,b} \cdot P_p$$

Figure 5.3a shows how debris d had an effect on the growth curves of the phages and bacteria in comparison to no debris term, Figure 5.3b. Without the debris term, only

(a) The $20 \times 20 \times 10$ model with a debris factor included.



(b) The $20 \times 20 \times 10$ model without the debris factor included.

Figure 5.3: A large $20 \times 20 \times 10$ model with a debris model added. The debris parameter values were randomly and uniformly selected between 0.01 and 0.2.

one uninfected bacteria survived, and every phage survived except for one phage who died due to the washout. Three uninfected bacteria species reached more than 1000 uninfected bacteria at any point in the simulation time, and only two infected bacteria species reached more than 1000 infected. The maximum total bacteria population that was reached was a total of 6620 bacteria.

With the debris term included, different results are obvious. Three uninfected bacteria survived, four different uninfected bacteria strains ever reached a population value of greater than 1000 at any point in time during the simulation. Only one bacteria strain ever reached more than 1000 infected bacteria, The ending bacteria value is 12000, almost double the maximum bacteria population reached without debris. On average, there were significantly fewer phages at the end of the simulation than without debris. No phage population ended above 2000 with debris, while without debris two phage populations ended up with more than 2000 phages.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Understanding the relationship that phages have with bacteria and the environment is complex. Not including M and the initial infected bacteria population, the parameter input space is 12 dimensions large. Although quite small in comparison to other larger models, analyzing 12 unique parameters and their interactions takes times and an intricate understanding of the model. Finding parameter values that result in quality and noteworthy graphs is not the easiest task, although knowing the expected values and the biological relevance makes the task easier. Finding a set of parameters that results in behavior that is worth analyzing takes time.

To help aid myself with the task, I created a simulation framework that anyone can use to analyze their own custom model and interaction network. A user can visually create and edit their interaction network. Using the dashboard, they can edit the parameter values and run their own simulations. The dashboard comes with some prebuilt visualization tools that allows the user to interact with the simulation results. The tools allow the user to quickly iterate over and change parameter values to see how they influence the simulation results. Although the visualizations on the dashboard are specifically built for $1 \times 1 \times 1$ models, the user can download the full simulation data and implement their own visualization methods to visualize the data over a $p \times b \times r$ system. I use this tool to great effect, to introduce visualization methods like the IVA to identify growth bottlenecks. I take the analysis further by introducing new visualizations such as a phage proliferation analysis and a survivability matrix.

Using the default interactive graphs on the dashboard and their own custom visualizations, users can explore the parameter space to identify notable model behaviors and

gain a deeper understanding of the system's dynamics. As evident by Figure 4.2b and Figure 4.2a, different parameter values will lead to contrasting behavior, even if both parameter models replicate realistic growth curves.

Trying to expand an analysis into a $p \times b \times r$ system becomes even more complicated due to the interconnected nature of the entities. With larger and larger systems small changes in a single parameter will not have big of an influence on the final output. But if the parameter value does have an influence, it has a cascading effect on the whole network. As an example, with a $2 \times 2 \times 1$ system, where P_1 infects B_1, and P_2 infects B_1 and B_2, and both bacteria consume R_1. Increasing the infection rate of P_1 will slow the growth of the B_1 as B_1 is being infected faster. With slower B_1 growth, and less uninfected B_1, P_2 is affected as there are fewer B_1 to infect. With lower P_2 count, B_2 can grow, using more resources. As there are now less resources, it is harder for B_1 to grow, so B_2 can grow. Eventually P_2 starts to infect B_2, so B_2 start to die, which gives B_1 a chance to consume resources and grow. A self-reinforcing feedback loop starts, where a change in the infection rate has a cascading effect on the rest of the network. The coupled interactions will have a feedback loop on itself, causing non-obvious behavior to occur.

Parameterizing complex system is a hard task. In the context of a $p \times b \times 4$ system, with $p, b, r > 1$, if you want to for example change β to 35, how would you do that? There are various methods to solve this. For a given matrix of size $p \times b$, you change each parameter value to 35. You could alternatively randomize the parameter values in the matrix, with the random values averaging to 35. A third option would be to shift every parameter values up or down so that the average of the values equal 35.

Typically for bacterial communities there is a random parameter matrix, possible with a predetermined structure.

6.2 Future Work

Next steps would be to collaborate with the researchers running the lab experiments so that they can verify the results as seen in the output by comparing the lab results with the model output. With the lab results, the model can be adapted to better fit the lab results. This can be done by changing parameter values, or by changing the model equation. The user can decide to add the Monod microbial growth model to the growth of the bacteria, or adapt the Monod equation to being dependent on multiple sources. Using the model, the technicians can improve and validate their methods. If the empirical results significantly deviate from the model results, the technician can

theorize what might be happening and alter the model to account for the discrepancy. Dey et al. [71] was able to adapt their model to account for the discrepancy between the model results and the results seen in the lab. They theorized that phages were somehow being deactivated. By adding the debris term, they were better able to account for the phage deactivation, and achieved a better and more accurate curve fit.

6.2.1 Model Replication

Being able to replicate other models like that of Nilsson [3] would allow me to compare model outputs. A benefit of implementing Cocktail's model is that it would be possible to model multiple bacteria and phages at the same time, as noted as a limitation in Section 2.6.3. Cocktail limits itself to 2 phages, 1 bacteria, and 1 resource. Cocktail supports adding more phages at set times, but only at most three times. This arbitrary limitation can be removed with Cocktail's model implementation.

6.2.2 Debris

Looking further into the debris and its effects could be a next step in the project. It was casually shown how adding a debris term increased the survivability of the bacteria populations, on average showing a higher uninfected bacteria population, and lower uninfected and phage population count. The debris term influenced the phage and bacteria dynamics.

6.2.3 Lab Work

A next logical step would be to complete lab work to obtain curves that could be used to compare the simulation results with the lab results. If the curves are significantly different from that of the ODE model, a new ODE model can be created. Curve fitting algorithms can be used to numerically find the interaction parameters for the new ODE model. Using the simulation software would save time, money, and resources as fewer experiments would have to be run.

The lab work would act as an important model validation step. Dey et al. [71] showed how their ODE model would eventually diverge from the lab-produced ODE curve. They were able to achieve a better curve fit by adapting the model to include the debris term.

Future lab work can also include finding out which bacteria, phages, and resources are found in marine water via samples taken from the environment. The next step would be to build an interaction network, along with experimentally finding out the interaction

parameter values using experimental lab work. Researchers can predict how the system would behave under new untested conditions, saving money and time. It might also tell researchers if they made a mistake during testing. If the model says the system should behave in one way, but the system acts differently, the researcher can review their methods and maybe make changes to how they run the experiments. All in all, having a model that takes seconds to run will help aid researchers better understand the system.

6.2.3.1 Environmental Modelling

Many results in research papers come from controlled lab settings. As a next step, researchers can actively collect daily water samples and measure phage, bacteria, and resource concentrations. Collecting samples for over a year would create an ODE-like population curve of the entities. This approach would provide deeper insights into the dynamics of bacteria and phage populations in natural environments, at the loss of control over conditions. By continuously monitoring environmental factors such as hourly temperature, rainfall, and the concentrations of each entity, researchers can gain a deeper understanding of the causal relationships within the ecosystem. By conducting the experiment over the course of a year, short-term fluctuations in daily measurements are averaged out, resulting in a smoother overall curve.

The next step would be to use the model to fit and explain observed population dynamics. We would want to model how environmental variables (for example temperature, nutrient availability, rainfall) influence the interactions between phages, bacteria, and resources. Key phenomena could include year long seasonal cycles, such as a dry and wet season, rapid growth and decline in population count, resilience, and impact of random events such as storms or pollution spikes. By fitting the model to real-world data, we could identify which parameters or interactions are most sensitive to environmental changes, and potentially predict how the ecosystem might respond to future events and scenarios. Trying to isolate these communities and run different experiments could be a next step. Analyzing which phages interact with which bacteria, or a knockout experiment to analyze how a loss in a bacteria or resource node has a cascading effect in the population growth.

Clegg and Gross [72] created a $b \times r$ bacteria-resource model, so without phages, and identified which bacteria consumed which resources. By changing the number of resources needed to survive, the consumer requirements, the researchers were able to change the community diversity. By adding and removing bacteria-resource interactions to the adapted Golding model, these results could be replicated. Adding more

bacteria-resource edges would introduce more competition, so more bacteria would die out, decreasing the community diversity. Removing edges would remove competition for the resources, and the community diversity would increase.

6.3 Other Users

Even though I created the simulation framework to help me with running and analyzing the simulation results, the idea of this program is that anyone could download the framework source code and run their own simulations. A professor might be interested in using this tool in an "Introduction to Bacteriology" or "Biological Modelling" course, where the professor would instruct the students to design and implement an ODE model. They would instruct the students to interact with the model as an introduction to modelling phage and bacteria populations. A researcher with weaker programming skills could use this low code tool to better understand how the system they are analyzing in their lab would change under different conditions. The code has been designed such that a user with basic programming skills can create their own analysis, by copy-pasting boilerplate like code. The biggest challenge would be to program the ODE model, and possibly their own custom visualizations.

Other users and researchers could program their own subset of tools to work with the tool. The graph network can be programmatically created with NetworkX. As the network is graph based, they can run network analysis programs on the graph to identify the node with the most edges and remove that node. The user can create a tool would programmatically knock out nodes and edges. Since the simulation framework is object orientated, the user can interface with the framework from their code and skip the dashboard to automate every step along their simulation pipeline. This would help users programmatically run a wider range and more unique simulations.

Chapter 7

Ethics and Data Management

7.1 Ethical Considerations

There are some ethical considerations needed. Phages can be used to treat bacterial infections, and may need to be safely administered under the supervision of a doctor, despite little to no side effects on the patient. Trying to control phages either in food or in the environment by releasing a phage cocktail into waterways could potentially cause issues further down the line if the released solution contains unwanted chemicals. An extra step in the food production will increase food costs and make food production harder to control. The cost to create, maintain, and use phages at an industrial size can become costly and require a lot of energy. Dumping phages into the ecosystem could potentially cause issues if the phage concoction includes resources for the bacteria to use, and can become costly for taxpayers.

7.2 Data Management

All data and results can be found on GitHub. Some simulation data will have to be recreated as the *.parquet* datafiles are too big for GitHub to store. Measures have been taken to label and document the code, datasets, parameter configurations, and for example with Appendix E. Any qualified researcher should be able to replicate, audit, use, and edit the computational experiments and code if needed. This systematic approach to version control and storage aligns with best practices, ensuring that results are both traceable and verifiable.

7.3 Adherence to Codes and Principles

I acknowledge that the thesis adheres to the ethical code and research data management policies of UvA and IvI.

The following table lists the data used in this thesis, with the source code. I confirm that the list is complete and the listed data are sufficient to reproduce the results of the thesis.

Short description (max. 10 words)	**Availability (e.g., URL, DOI)**	**License**
Dataset	Simulation Results	MIT
Source Code	Source Code	MIT
Simulation Conditions	Appendix E and text under figures	

Appendix A

Appendix A: Equation Parameters

Parameters used in the various equations.

A.1 Simple/Advanced Golding Model Parameters

Variable	Name	Description
R/R_r	Resource entity	Resource r concentration
U/U_b	Uninfected Bacteria population	Uninfected population for bacteria b
I_i/I_{b_i}	Infected Bacteria population	Infected population for bacteria b at stage $1, \ldots, i, \ldots, M$
B/B_b	Bacteria population	Total bacteria population for bacteria b, assuming $B_b = U_b + \sum_{i=1}^{M} I_{b_i}$
P/P_p	Phages population	Phage population for phage p
e/e_{br}	Consumption rate	Rate at which resource r is consumed by bacteria b
β/β_{pb}	Burst size (B matrix)	Lytic burst size for phage p and bacteria b
r/r_{pb}	Successful phage/cell encounter	Probability of a phage p successfully infecting bacteria b
τ/τ_b	Latent period (tau vector)	Time it takes bacteria b to go through the infection stage
v/v_{br}	Maximal growth rate	Growth rate of bacteria b from resource r
K/K_{br}	Monod Constant	Monod constant representing at what resource r concentration at which bacteria b grows at half its maximal rate v
ω^i/ω_r^i	wash-in rate	Rate of resource r being added
ω^o	wash-out rate	Rate of phages, bacteria, and resources being removed, acts on everything proportionally
M	Number of infection stages	Number of infection stages that a bacteria goes through, all bacteria entities have the same value for M
d	Debris	A debris term used to deactivate phages. See Section 5.7.2 for more information.
t	time	time value through the simulation

Table A.1: Golding model parameters (Equation (2.13) and Equation (2.19)) with variables, names, and descriptions. Subscripts on parameters indicate relationships; for example, e_{br} is nonzero if there is an edge connecting bacteria b to resource r in the network, zero otherwise.

A.2 Sobol Parameters

Variable	Name	Description
Y	Univariate parameter output	univariate model output, such as mean μ or variance σ
X	Input vector	Vector of size d, input vector to f
i	Parameter input	Parameter i of input
X_i	Parameter value	Value of vector X at position $i = 1, \ldots, d$, the value of parameter i
d	Input size	Size of input vector X
$X_{\sim i}$	Parameter input	All values of X that are not X_i
f	Function f	Arbitrary black-box function describing model
N	Samples	Number of samples, power of 2, 2^x
D	Parameter input size	Number of parameters inputted into Sobol, $d = \|X\|$
ST_i	Global sensitivity	Contribution of parameter X_i to output variance of Y due to interactions with other variables
$S1_i$	First order sensitivity	Contribution of X_i to output variance of Y

Table A.2: Sobol parameter symbols, name, and description. See Section 3.1.3.2 for the equations.

A.3 Linear Regression Parameters

Variable	Name	Description
a	Slope	Slope of the linear regression line
c	Intercept	y-intercept of linear regression line
R^2	Regression Coefficient	Coefficient of determination of linear regression fit, quality of regression
x_i	Data point	Data point on the x-axis
y_i	Actual Value	Actual value of data for a given x_i
$\hat{y}_i$	Predicted Value	Value predicted of equation for a given x_i
$\bar{y}$	Average Value	Average y value
n	Number of samples	Number of samples being tested

Table A.3: Variable symbol, name, and description used for the linear regression.

Appendix B

Appendix B: Industrial and Real Life Applications of Phages

Due to the nature of killing bacteria, there are numerous applications where a researcher or an organization might be interested in controlling bacterial populations.

A Food Safety Specialist might be interested in introducing a solution containing a high concentration of phages during food production to prevent the spread and growth of *Salmonella* or *E. coli* in the pet food. Alternatively, the Food Safety Specialist might want to promote beneficial bacteria like *Streptococcus thermophilus* used in the production of Emmental cheese, which heat would kill when the milk undergoes the pasteurization process.

A doctor might be interested in providing swallowable pills, more commonly known as phage cocktails, to a patient with a bacterial infection. There is evidence that phage-resistant bacteria are more susceptible to antibiotics, so the doctor might prescribe both medicines to effectively deal with the infection.

An Environmental Protection Officer might be interested to see how they can use phages to stop the spread of *Cyanobacteria* blooms in waterways, more commonly known as blue-green algae, a photosynthetic microscopic organism that is technically a type of bacteria. This would keep waterways safe for boating and swimming activity, aquatic life, and water consumption in farms, factories, and homes.

When there are a few known bacterial strains, a targeted concoction of phages can be used to control the bacterial population growth in any setting, either be it food, healthcare, or environmental. Phages offer properties of microbial control that other methods do not, making them an ideal candidate for some applications.

B.1 Controlling Foodborne Bacteria

Foodborne diseases are one of the primary ways for bacteria to spread to humans and animals. Some bacteria use the food as a vector to infect hosts, while some bacteria will deposit toxins on the food that is then ingested. If consumed in large enough quantities, or further produced in the host, the toxins can be fatal to the host.

Methods exist to control bacterial growth, for example by storing food below 5°C or above 60°C. Bacteria need moisture to grow, so starches like rice will have minimal bacterial growth. Bacteria prefer to live in slightly acidic to neutral pH environments, so having an environment that is extremely acidic like vinegar will prevent bacterial growth. The use of chemical antibacterial entities such as bleach is not desirable due to leaving chemicals on the food, which can be fatal if ingested. Physical entities like heat or radiation can kill bacteria, but at the cost of altering the food quality [73].

For example, *Streptococcus thermophilus* is one of three different bacteria strains used to create Emmental cheese. However, Emmental cheese does not use pasteurized milk, increasing the risk of *E. coli*. Emmental cheese producers can add phages that target *E. coli* to the milk during the production stage, while not affecting the bacteria used to produce the cheese.

B.1.1 Current Applications

Phage cocktails like SalmoFresh^{TM} have been proven to safely reduce *Salmonella* contamination in pet food and raw pet food ingredients [5], as well as in romaine lettuce and bean sprouts [6]. Pet food contains meat and vegetables, where vegetables grown in or on the ground are at risk of *Salmonella* due to contact with soil, manure, compost, and other agricultural runoff from neighboring farms [17]. Figure B.1 and Figure B.2 show how application of phages have reduced the count of *Salmonella* in ingredients used in pet food as well as romaine lettuce and bean sprouts. In Figure B.1, each food group noticed at least a 68% reduction in CFU/g compared to the control when the 9×10^6 phage treatment was applied. There was at least an 80% reduction in CFU/g across all food groups when treated with a 9×10^6 or stronger phage solution. In Figure B.2, the lettuce and bean sprouts noticed a reduction of at least 0.6 log CFU/mL in *Salmonella* count across all temperature ranges. The smallest reduction in bacteria count in lettuce was noticed at 1 hour at 2°C with an absolute reduction in 62.0% between the control and treatment, while the largest reduction in bacteria of 90.0% was found at 72 hours at 2°C. For the bean sprouts, the lowest reduction in phages was found at 1 hour at 2°C with a reduction of 78.1%, and the largest reduction was 90.0% at 25°C after 48

hours. Although these values are still high above food safe, the ability to reduce the *Salmonella* population by at least 62% and up to 90% at different temperatures and incubation periods is impressive and can prolong shelf life, especially for foods that do not have long shelf lives before spoiling due to bacteria. As such, phages can be shown to control the spread of *Salmonella* in food sources and extend the potential shelf life of certain foods.

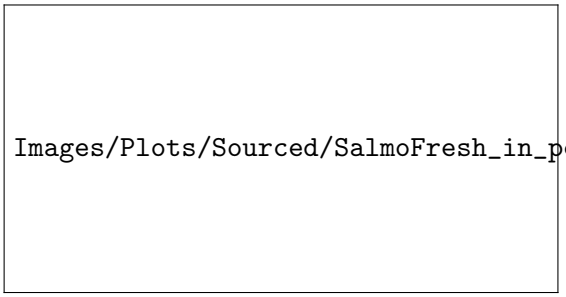


Figure B.1: SalmoLyse® reduces Salmonella contamination on various food surfaces: Mean and standard error bars shown. Statistical analyses were carried out for each food group independently. Asterisks denote significant reduction from corresponding controls based on one-way ANOVA with Tukey's post-hoc tests for multiple corrections: ** denotes $p < 0.01$, while *** denotes $p < 0.001$ compared to the corresponding controls. There was significant reduction in Salmonella on all food surfaces with the addition of SalmoLyse® compared to the controls; the mean percent reductions from the control are noted in the boxes above treatment bars. CFU/g D colony forming units per gram. Each letter denotes a food group that was tested with SalmoLyse® and compared to a control: A= chicken; B= lettuce; C= tuna; D= cantaloupe; E= ground turkey. Plot sourced from Soffer et al. [5].

Figure B.2: *Salmonella* count in a mixture of 5 *Salmonella* strains spot-inoculated (CFU/g) onto a) lettuce and b) sprouts after spraying with a mixture of bacteriophage (SalmoFresh™) relative to positive controls at 2, 10 and 25C and stored for 1, 24, 48 and 72 h. Plot sourced from Zhang et al. [6]

B.2 Phage Therapy and Antibiotics

Antibiotics are a common way to treat bacterial infections. However, antibiotics are not selective in the bacteria they kill, killing both harmful and beneficial bacteria. This can lead to the development of antibiotic-resistant bacteria, which makes it harder to combat that bacteria in the future. It has also been shown that antibiotics have a negative effect on the gut microbiome and brain development in mice. Phages are an alternative to antibiotics, as they are selective in the bacteria they kill and do not interact with cells or other important biological functions. The rise in antibiotic resistant bacteria can be attributed to the overuse and over-prescription of antibiotics and incorrect usage of antibiotics (for example prematurely stopping) [13]. These actions provide an evolutionary pressure on bacteria to mutate and gain resistance to the antibiotics. The phage therapy can contain any number of different phages that can target specific bacterial infections such as *Streptococcus pneumoniae* with minimal risk of side effects.

B.2.1 Current Applications: Bacterial Infection Control

One active area of research is the use of phages to control bacterial infections. Due to the specificity of phages, they can be used to target specific bacteria strains without affecting other beneficial bacteria. When sick with a bacterial infection, patients swallow antibiotic pills to help the body fight the infection. Antibiotics work by either interrupting intercellular processes like the synthesis of RNA [74], by disrupting the structural integrity of the cell wall [75], or by inhibiting protein synthesis [76].

However, antibiotics are not strain specific and indiscriminately kill gut and other bacteria. Common side effects of antibiotics, although usually not serious, include diarrhea, nausea, and headaches. It has also been shown that the effects of early-stage penicillin exposure in mice has found to have a long-lasting effect on the gut microbiome, frontal cortex gene expression, and amygdala gene expression [14]. Penicillin increases cytokine expression (small proteins used in cell signaling) in the frontal cortex of the brain, modifies the blood-brain barrier integrity, and alters behavior. The mice exhibited an increase in aggression and anxiety-like behavior [77]. Phages can be used as an alternative to antibiotics without the side effects and without affecting the gut biome.

With an increase in antibiotic usage, there has been an increase in antibiotic-resistant bacteria. The World Health Organization has stated that antibiotic resistance threatens the modern medicine and the sustainability of an effective, global public health response to the enduring threat from infectious diseases. Common infections, that previously

would have been easy to treat, are harder to treat, and can increase the risk of disease spread, severe illness, and death [78].

One area of research is exploring how bacteria can exchange traits such as phage resistance and antibiotic resistance. Some bacteria are multi-drug resistant, and don't react with the medicine anymore.

Laure and Ahn [15] showed evidence that *Salmonella Typhimurium* is more susceptible to ampicillin in the presence of phages, and phage-resistance can lead to reduced virulence and decreased antibiotic resistance.

Zhao et al. [16] showed that there exists an antagonist coevolution between the bacteria and phages, where the dynamics changed from an arms race dynamic (ARD) to a fluctuating selection dynamics (FSD). Due to phage selection and bacterial competition pressure, when the bacteria gained phage resistance, it lost antibiotic resistance. A genome analysis revealed mutations in the btuB gene of *Salmonella anatum*, with q higher mutation frequency in the ARD stage. A knockout experiment confirmed that the btuB gene is a receptor for the JNwz02 phage and resulted in reduced bacterial competitiveness. Further analysis detected multiple single nucleotide polymorphism (SNP) mutations in the phage-resistant strains. The SNPs potentially affected the membrane components, partially weakening the cell defense against antibiotics. These findings help advance our understanding of phage-host-antibiotics interactions and the impact of adaptations to antibiotic resistance. The research shows how phages can be used to reintroduce antibiotic susceptibility to previous insusceptible bacteria, preventing costly and lengthy research in new antibiotics [16].

Phage research is facing challenges due to bacterial strains evolving resistance to phages. Understanding the interplay between antibiotics and phages is essential for shaping future research [16].

B.3 Environmental Protection

Algae blooms, also called red tides, is the rapid spread of bacterial or algae organisms. Blooms are a growing environmental concern impacting water quality, aquatic ecosystems, and human health. These rapid increases in algae populations, often fueled by excess resources like nitrogen and phosphorus, can occur in freshwater, coastal, and marine environment.

Cyanobacteria blooms have major effects on the aquatic environment as well as human health. Cyanobacteria release nitrogen and phosphorous, which the bacteria use to grow

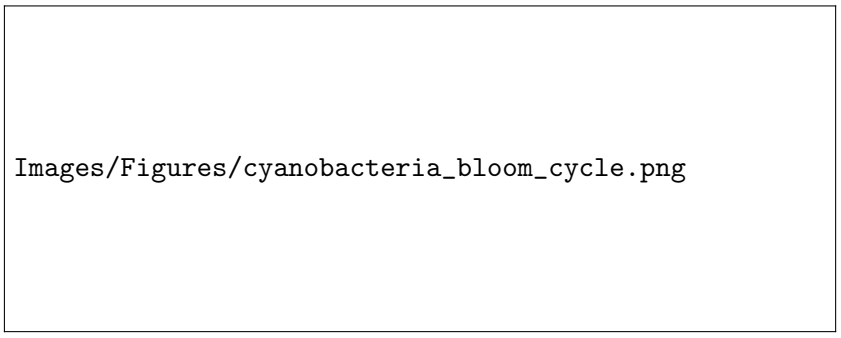


Figure B.3: Cyanobacteria degradation cycle, main hazards of cyanobacteria bloom to water bodies, aquatic organisms, and the human body. (DO: dissolved oxygen; SD: water transparency; Cond: conductivity; N: nitrogen; P: phosphorus; MCs: microcystins). [7]

with oxygen, outpacing other aquatic growth, and killing aquatic marine life. Bacterial toxins can make their way into the food and water consumed by humans, causing muscle fatigue, respiratory issues, liver damage, and gastrointestinal issues [7]. Figure B.3 shows the process of how cyanobacteria degrade and are absorbed into the environment, eventually making their way into the human body via various contact points.

B.3.1 Current Applications

There is interest in using phages to control cyanobacteria blooms. Phages can offer better and safer options than chemical options when trying to control bacterial blooms. Chemical options are indiscriminate, killing cyanobacteria, while also killing other beneficial bacteria and aquatic life, and can eventually seep into groundwater. Although not used to control bacteria blooms, some chemicals like PFAS, also called "Forever Chemicals", can last a long time in the environment and don't degrade and keep on negatively affecting the environment. Due to the specificity of phages, only the cyanobacteria will be targeted, and will not affect the surrounding environment.

Tucker and Pollard found that an isolated phage cocktail collected from Lake Baroon in Australia could decrease the abundance of *M. aeruginosa* by 95% within 6 days in a lab setting, before recovering within 3 weeks time [12].

There is evidence that phage-resistant bacteria can influence the population dynamics of other bacteria. It has been shown that the plankton level has been experimentally affected by the frequency of the phage-resistant *Nodularia* marine bacteria. Populations with high phage resistance ($> 50\%$) dominate the plankton communities despite a high phage count and eventually out compete other bacteria due to their slower loss in population count. Contrastingly, populations of bacteria with low phage resistance (between 0% and 5%) were lysed to extinction, releasing resources like nitrogen. This

allows for other bacterial strains to absorb the resources and dominate the bacterial community. Phages and the lysis of bacterial strains can have a dramatic effect on community dynamics and composition of other entities like phages, bacteria, and resources [11]. Phages have the potential to be used as a highly specific strategy for the control of cyanobacterial blooms, with minimal effects to the environment, and offer control of bacterial blooms, with limited impact to the environment. Usage should be relatively safe, novel, efficient, and sensitive.

However, there are issues with using phages to control bacterial blooms. Bacterial blooms can cover vast areas, or be in areas that would be hard to reach like marshlands, applying phages to combat the bloom might be infeasible. If the method of choice was to spray a solution of water containing phages, the solution needs to be shipped to the site and loaded onto special boats to spray the solution into the water, or the trucks need to drive along the shore and spray the solution into the water.

The phage density in the solution will have to be relatively high to quickly combat the bloom. These problems provide major logistical problems with creating the phages in a lab or factory, transporting the phages, and finally the administration of the phages to the waterways. Phages can only diffuse through the water, and can't actively swim, so they are dependent on the rate of diffusion and water currents. This will be difficult in marshlands, where the bacteria can "hide" in the grass and crevices created by aquatic life. If the bloom is in a high current area, like in a river or a bay, the water can wash the phages away.

Scientists have not yet fully understood the phage infection mechanism, and research into the artificial engineering of phages is limited, making it challenging to conduct studies in this area [79, 80].

Algae can produce toxins that threaten wildlife, contaminate drinking water, and disrupt local economies dependent on fishing and tourism. In the state of Florida, between the years 1995 and 2000, the restaurant and hotel industry lost an estimated $6.5 million to algae blooms. This accounts for about 25% of the average total monthly sales revenue in the region from June through October, the months that are most commonly affected by red tide[81]. During a red bloom event, hospital diagnoses in the county of Sarasota for pneumonia, gastrointestinal, and respiratory illness increased by 19%, 40% and 54% respectively [82, 83], with a respiratory illness visit costing between $0.5 and $4 million [84].

Appendix C

Appendix C: Flowchart of User and System Interactions

Figure C.1 shows how the user can interact with the system, the input and outputs for subsystems, and the systems working with one another. To read the flow chart, start from the top to the bottom. First the user creates a network using the GUI Network Creation Tool. After the graph is finished, the user provides an implementation of the network as an ODE model, using Python. Once finished, the user provides the network file and ODE model to the ODE solver. The solver uses information from the network file to determine the number of entities to create, parameter details (including names, values, and dimensions), and setting values. Then the user interacts with the Visualization Dashboard Tool, for example by clicking on buttons to run simulations, changing parameter values, (un)selecting checkboxes, and zooming in and out of plots, and hovering over plots to show data. Once a user has selected the parameter values, the parameter values are sent to the solver. The solver calculates the time and population values using the provided graph and ODE model and sends the data back to the Visualization Dashboard Tool, which then outputs the visualizations. If the user has run an Ultimate Analysis, then the user can query the saved data to make their own custom visualizations.

Figure C.1: The flowchart of user and system interactions. Read from top to bottom.

Appendix D

Appendix D: ODE Model Implementation

The code listed here is the implementation of Section 2.7.3.2.

Appendix E

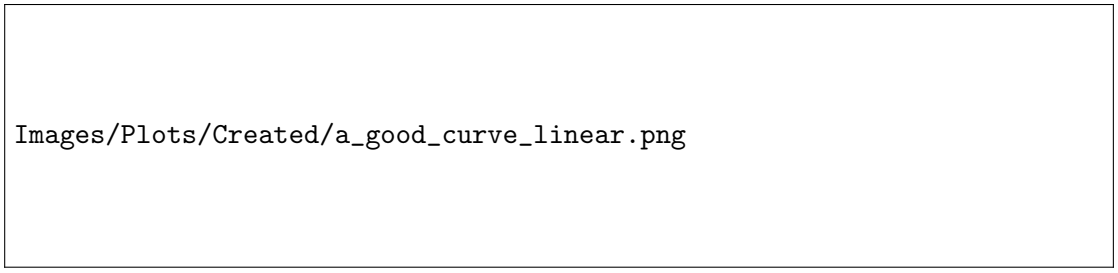
Appendix E: Parameter Values Used

E.1 Realistic Growth Curves

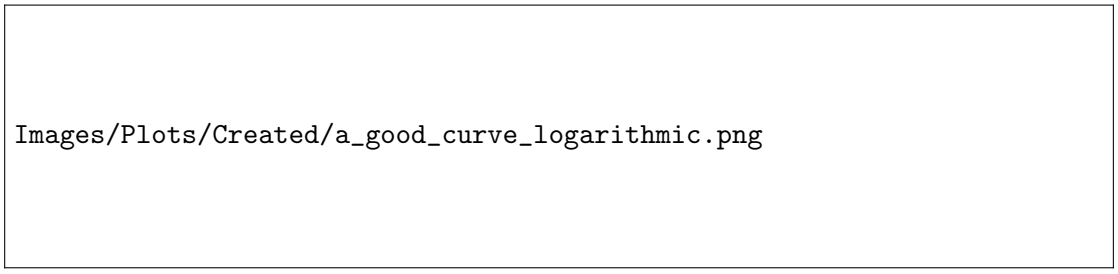
IC					
Resources	**Uninfected Bacteria**			**Infected Bacteria**	**Phages**
400	50			$\begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}$	10
Vector Data					
$\boldsymbol{\tau}$	$\boldsymbol{\omega^i}$				
2.14	0				
Matrix Data					
$\boldsymbol{e}$	$\boldsymbol{v}$	$\boldsymbol{K}$	$\boldsymbol{r}$	$\boldsymbol{\beta}$	
0.03	1.2	10	0.01	20	
Environment Data					
$\boldsymbol{M}$		$\boldsymbol{\omega^o}$			
4		0			

Table E.1: The parameter values used for Figure 2.3.

E.2 A Second Realistic Growth Curve



(a) A second realistic growth curve, linear y-axis



(b) A second realsitic growth curve, logarithmic y-axis.

Figure E.1: The parameters used for this plot can be found in Table E.1.

IC					
Resources	**Uninfected Bacteria**			**Infected Bacteria**	**Phages**
200	50			$\begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}$	10
Vector Data					
$\boldsymbol{\tau}$	$\boldsymbol{\omega^i}$				
0.7	0				
Matrix Data					
$\boldsymbol{e}$	$\boldsymbol{v}$	$\boldsymbol{K}$	$\boldsymbol{r}$	$\boldsymbol{\beta}$	
0.12	1	10	0.001	10	
Environment Data					
$\boldsymbol{M}$		$\boldsymbol{\omega^o}$			
4		0			

Table E.2: Another set of realistic growth curves. The linear and logarithmic plot of this data can be seen in Figure E.1.

E.3 Sobol Analysis

IC				
Resources	**Uninfected Bacteria**		**Phages**	
1-500	1-100		1-50	
Vector Data				
$\boldsymbol{\tau}$	$\boldsymbol{\omega^i}$			
0.5-3.5	0-100			
Matrix Data				
$\boldsymbol{e}$	$\boldsymbol{v}$	$\boldsymbol{K}$	$\boldsymbol{r}$	$\boldsymbol{\beta}$
0.05-0.25	0.8-1.9	10-250	0.001-0.2	1-100
Environment Data				
$\boldsymbol{\omega^o}$				
0-0.1				

Other Data				
Seed Value	**2nd Order**	**Number Samples**	**Simulations Run**	**Simulation Length**
0	False	15	$2^{15}(9+2) = 360448$	15

Table E.3: The parameter values used for the Sobol sensitivity analysis in Figure 4.1 (Sobol analysis without washin and washout) and Figure F.1 (Sobol analysis with washin and washout). For Sobol analysis with washin and washout, there are $2^{15}(9+2) = 425984$ unique simulations run.

E.4 Complex Model

IC

Resources	**Uninfected Bacteria**	**Infected Bacteria**	**Phages**
$\begin{bmatrix}236 & 287 & 270\end{bmatrix}$	$\begin{bmatrix}53 & 69\end{bmatrix}$	$\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}10 & 5 & 8\end{bmatrix}$

Vector Data

$\boldsymbol{\tau_b}$	$\boldsymbol{\omega_r^i}$
$\begin{bmatrix}2.73340 & 2.25015\end{bmatrix}$	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}$

Matrix Data

$\boldsymbol{e_{br}}$	$\boldsymbol{v_{br}}$
$\begin{bmatrix}0.15680 & 0.10871 & 0\\ 0 & 0 & 0.18009\end{bmatrix}$	$\begin{bmatrix}1.27601 & 0.86393 & 0\\ 0 & 0 & 1.22625\end{bmatrix}$

$\boldsymbol{K_{br}}$	$\boldsymbol{r_{pb}}$	$\boldsymbol{\beta_{pb}}$
$\begin{bmatrix}139.58353 & 12.83058 & 0\\ 0 & 0 & 82.86684\end{bmatrix}$	$\begin{bmatrix}0 & 0.11695\\ 0.144459 & 0\\ 0.11895 & 0.13065\end{bmatrix}$	$\begin{bmatrix}0 & 15\\ 34 & 0\\ 11 & 57\end{bmatrix}$

Environment Data

$\boldsymbol{M}$	$\boldsymbol{\omega^o}$
4	0

Table E.4: The parameter values used for the $3 \times 2 \times 3$ network model rounded to 5 decimal points. If there is no edge between a phage, bacteria, or resource, then in the matrix representation of the parameter, 0 is stored as the default value.

Appendix F

Appendix F: Extra Plots and Figures

F.1 Sobol Analysis With Washin and Washout

F.1.1 Final Value Analysis

F.1.1.1 Resources

Despite the resource consumption rate directly depending on e, v and K, the parameters had very little influence on the final value as evidence by the ST and S1 bar being near 0. The resource population was mainly driven by the washin and washout value. The peak resources are driven completely by the washin rate. Not many interactions between two or more parameters were occurring.

F.1.1.2 Phages

The most important factor for the final phage value is r, followed by β and ω^o. The other parameters had little to no effect on the final phage value.

F.1.1.3 Total Bacteria

The sum of uninfected and infected bacteria depended heavily on higher order interactions as $ST_i \gg S1_i$. Although not shown, the bar plots for the total bacteria resembled that of the bar-plots for the uninfected bacteria, and less that of the infected bacteria.

F.1.2 Peak Value and Time of peak

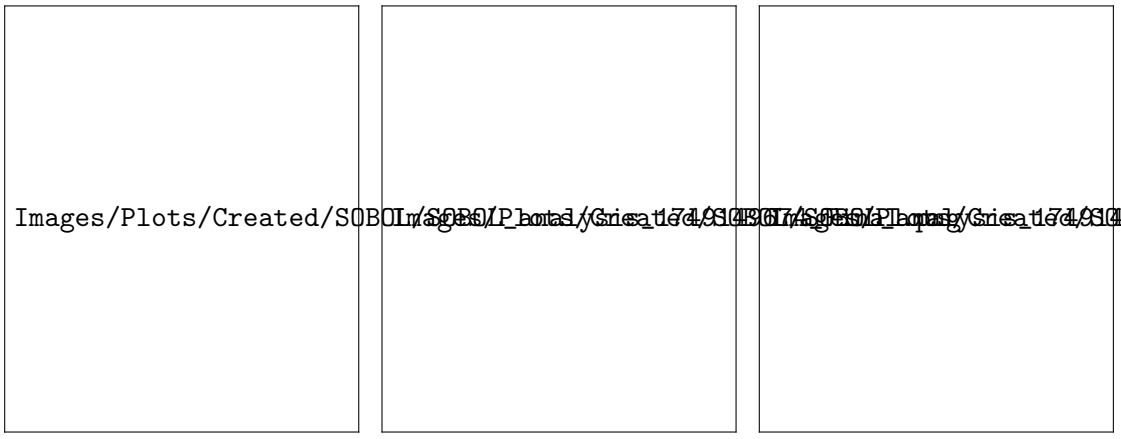
To create the custom Sobol analyses, the peak value and the time at the peak of the population is measured and analyzed. The peak is defined as the point where the population reaches 95% of its absolute maximum value. The time at peak is measured at the point in time that the population reaches 95% of the maximum value. This removes unintended side effects of the simulation. For populations that are only increasing in value, this prevents the measured peak from bunching up at the end of the simulation, skewing the data. As the peak is defined at 95% of the absolute maximum value, populations that have a faster increase on population count at the end will have a time value closer towards the end of the simulation. For populations that reach a plateau, the 95% rule will push the peak time towards the beginning of the simulation, while still "respecting" the absolute final value since $95\% \approx 100\%$. The 95% rule can fail under certain situations, such as when there is cyclic behavior. See Appendix F.2 for a more detailed explanation on why the 95% rule is used.

The results of the Sobol peak and time at peak analyses can be seen in Figure F.1b and Figure F.1c. Although some bars between the final, peak, and time at peak values are the same, some are different. But overall, similar values can be seen across the final, peak, and time at peak analyses. The peak infected values are more certain compared to the final infected values, which could be due to the 95% rule removing some of the noise of the simulation. The time at peak values have less error compared to the final and peak value. This is due to the restricted range of values. The time at peak value can only fall somewhere between 0 and 15, the start and end values of the simulation respectively. The final and peak values can fall anywhere between 0 and any value, depending on the IC and how high the population can rise, and how fast the population can fall, *if* the population count falls.

F.2 Why 95%?

The 95% rule helps in the IVA analysis. Due to the solver, when taking the absolute peak value, the same time value can occur. Or in an ever increasing value like phages, the peak values occur at the last time step of the simulation, or plateaus and doesn't grow anymore. However, as the parameter value is changing, each graph for every input change will change the growth rate of the entity, changing how fast the entity population grows.

Figure F.2 shows how using the 95% rule vs the 100% rule for finding the max value reached helps smooth out computational errors from the ODE solver and smooths out



(a) Final population value. **(b)** Peak population value. **(c)** Time at peak population.

Figure F.1: Sobol analyses for the final, peak, and time of peak value with a washin and washout rate. The data was saved from the dashboard and plotted using Matplotlib. The values used for this Sobol test can be found in Table E.3.

the shape. For the phages, using the 100% rule (Figure F.2b) shows that the population peaked at the end of the simulation, $t = 15$, for all e values. However, at $t = 15$, the population plateaued, as evident by the line graph. Plotting the same plot, but calculating the peak at 95% of the actual peak (Figure F.2a) shows that the green line ($e = 0.25$) "reached" its peak at $t = 8.4$ before the red line ($e = 0.05$) at $t = 9.4$, a full unit of time after $e = 0.25$. The user can thus conclude that for this instance, larger e values will cause the phage population to reach its "peak" faster than smaller e values.

Figure F.2c and Figure F.2d likewise show how the 95% rule can improve analysis of the change in peak time. Figure F.2d shows how apparently the peak is reached at set time values. Due to how *solve_ivp()* from SciPy works, it automatically chooses time values that it thinks would best capture the dynamics of the system without calculating too many steps. The user can control the step size by decreasing the absolute and relative error bounds, as well as by minimizing the time steps. The user can also provide their own time range with the number of steps to run, increasing the control of the time values chosen. It takes about 0.02321 seconds to run a simulation for 15 time units, where 200 time steps are selected and solved by the solver. Comparatively, a simulation with 1000 time units and 1000000 (a 5000x increase in samples) equidistant time samples takes about 1.71651 seconds to run, a 73.95562258x increase in time spent computing the simulation. The total time taken to run the whole method call, a call to the simple graph maker at the top of the dashboard took 1.76130 seconds vs 17.70634 seconds.

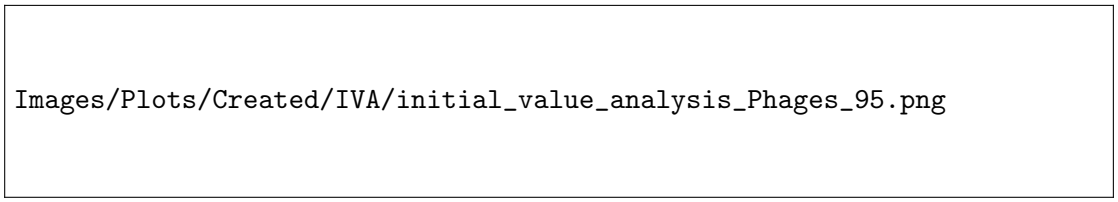
Alternatively instead of controlling the solver, the user can use the 95% rule. Although some accuracy is lost. Going from the 100% rule to the 95% rule, the solver still captures the peak values and the dynamics, but the accuracy is lost. The 100% rule shows that for $e = 0.25$, the time the uninfected population reached its peak occurred at $t = 3.2$.

But for the 95% rule, the time at which the peak occurred at is at $t = 3.05$. The slope (the a value) and the intercept (the c value) are somewhat similar, with very high and similar R^2 values (0.97), suggesting a good linear fit of the data.

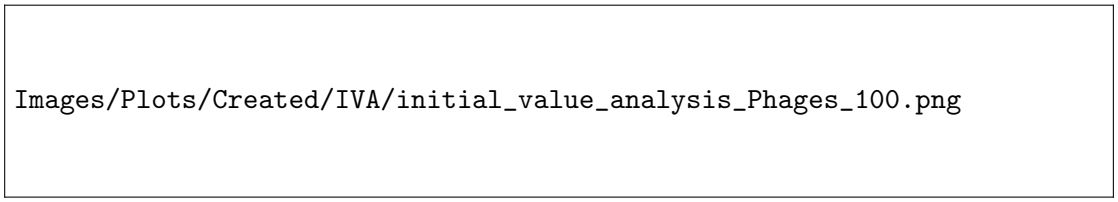
Figure F.2e shows how the by increasing the time sampling to more fine-grained results in a more accurate graph. Instead of having the solver choose the time values to test, 1000 equidistant time values were selected between 0 and 15. The solver can more accurately calculate the population values and calculate the proper peak time. Comparing the 100% rule without the custom time values with the 100% rule with the custom time values shows the same time values were calculated. In both, the $e = 0.25$ resulted in a time of peak at 3.2 and for $e = 0.05$, the time of peak occurred at $t = 3.95$. This is in stark comparison to the 95% rule vs 100% rule without the custom time, showing a difference of 0.15 time units. The custom time values also preserved the shape of the curve e-value vs time curve, being almost identical to that of the 95% rule as seen in Figure F.2c and Figure F.2e.

Another issue that arises with the custom time is that it doesn't solve the issue seen with the phages, where the time of peak is at $t = 15$.

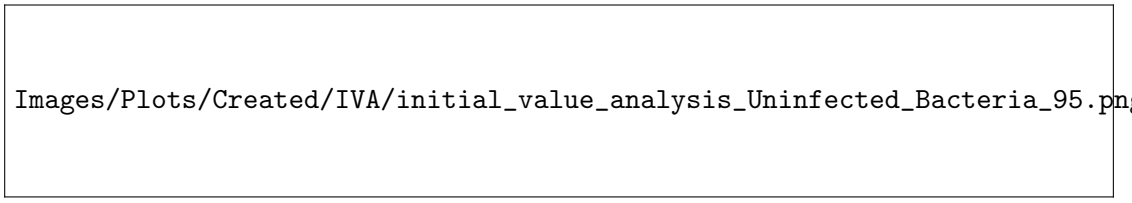
The user can control the % rule with a value input on the dashboard. They can select to use the 95% rule, or 100% rule, or even 83% rule if they want by changing the value they use. The user can use their own custom time values, to ensure that they get high quality curves.



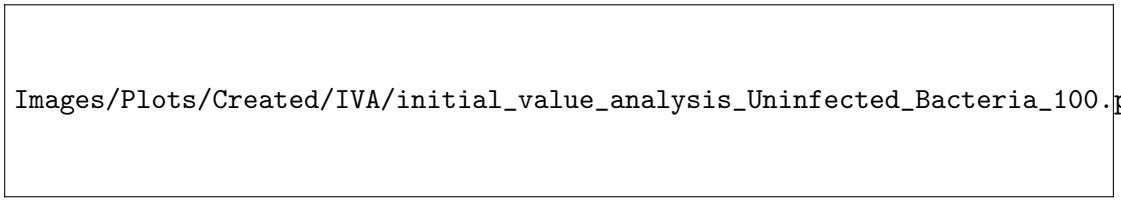
(a) IVA for phage population, 95% rule



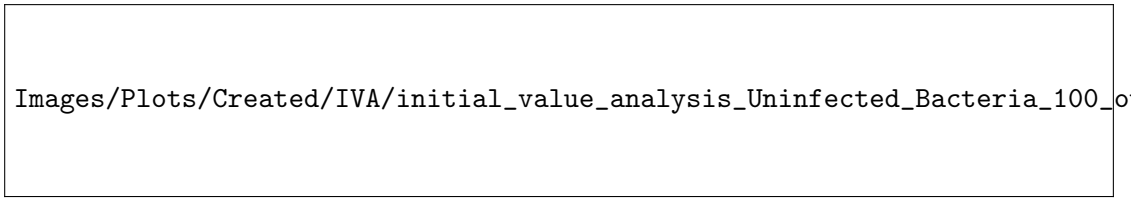
(b) IVA for phage population, 100% rule.



(c) IVA for uninfected population, 95% rule.



(d) IVA for uninfected population, 100% rule.



(e) IVA for uninfected population, 100% rule, 1000 equidistant time steps.

Figure F.2: Testing the 95% rule vs the 100% rule, where the time at the absolute peak is taken and plotted in the second plot. A comparison of phages and uninfected bacteria is shown. Verification of the graph shape between the 95% rule graph and a frequent time step with 100% rule can be seen between c) and e). The e value is changed, ranging from 0.05 to 0.25.

F.3 Graph Behavior with IVA

F.3.1 *R*

Figure F.3: Changing initial resource values.

F.3.2 *U*

Figure F.4: Changing initial uninfected bacteria values.

F.3.3 *P*

Figure F.5: Changing initial phage values.

F.3.4 τ

Figure F.6: Changing τ values.

F.3.5 r

Figure F.7: Changing initial r values.

F.3.6 β

Figure F.8: Changing β values.

Bibliography

[1] Yuncong Geng, Thu Vu Phuc Nguyen, Ehsan Homaee, and Ido Golding. Using bacterial population dynamics to count phages and their lysogens. *Nature Communications*, 15(1):7814, September 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-51913-6. URL https://www.nature.com/articles/s41467-024-51913-6.

[2] Allan Campbell. The future of bacteriophage biology. *Nature Reviews Genetics*, 4 (6):471–477, June 2003. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg1089. URL https://www.nature.com/articles/nrg1089.

[3] Anders S. Nilsson. Cocktail, a Computer Program for Modelling Bacteriophage Infection Kinetics. *Viruses*, 14(11):2483, November 2022. ISSN 1999-4915. doi: 10.3390/v14112483. URL https://www.mdpi.com/1999-4915/14/11/2483.

[4] Konrad Krysiak-Baltyn, Gregory J. O. Martin, Anthony D. Stickland, Peter J. Scales, and Sally L. Gras. Simulation of phage dynamics in multi-reactor models of complex wastewater treatment systems. *Biochemical Engineering Journal*, 122: 91–102, June 2017. ISSN 1369-703X. doi: 10.1016/j.bej.2016.10.011. URL https://www.sciencedirect.com/science/article/pii/S1369703X16302728.

[5] Nitzan Soffer, Tamar Abuladze, Joelle Woolston, Manrong Li, Leigh Farris Hanna, Serena Heyse, Duane Charbonneau, and Alexander Sulakvelidze. Bacteriophages safely reduce Salmonella contamination in pet food and raw pet food ingredients. *Bacteriophage*, 6(3):e1220347, July 2016. ISSN null. doi: 10.1080/21597081.2016.1220347. URL https://doi.org/10.1080/21597081.2016.1220347.

[6] Xuan Zhang, Yan Dong Niu, Yuchen Nan, Kim Stanford, Rick Holley, Tim McAllister, and Claudia Narváez-Bravo. SalmoFresh™ effectiveness in controlling Salmonella on romaine lettuce, mung bean sprouts and seeds. *International Journal of Food Microbiology*, 305:108250, September 2019. ISSN 0168-1605. doi: 10.1016/j.ijfoodmicro.2019.108250. URL https://www.sciencedirect.com/science/article/pii/S0168160519301709.

[7] Weizhen Zhang, Jing Liu, Yunxing Xiao, Yumiao Zhang, Yangjinzhi Yu, Zheng Zheng, Yafeng Liu, and Qi Li. The Impact of Cyanobacteria Blooms on the Aquatic Environment and Human Health. *Toxins*, 14(10):658, September 2022. ISSN 2072-6651. doi: 10.3390/toxins14100658. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9611879/.

[8] Basem Al-Shayeb, Rohan Sachdeva, Lin-Xing Chen, Fred Ward, Patrick Munk, Audra Devoto, Cindy J. Castelle, Matthew R. Olm, Keith Bouma-Gregson, Yuki Amano, Christine He, Raphaël Méheust, Brandon Brooks, Alex Thomas, Adi Lavy, Paula Matheus-Carnevali, Christine Sun, Daniela S. A. Goltsman, Mikayla A. Borton, Allison Sharrar, Alexander L. Jaffe, Tara C. Nelson, Rose Kantor, Ray Keren, Katherine R. Lane, Ibrahim F. Farag, Shufei Lei, Kari Finstad, Ronald Amundson, Karthik Anantharaman, Jinglie Zhou, Alexander J. Probst, Mary E. Power, Susannah G. Tringe, Wen-Jun Li, Kelly Wrighton, Sue Harrison, Michael Morowitz, David A. Relman, Jennifer A. Doudna, Anne-Catherine Lehours, Lesley Warren, Jamie H. D. Cate, Joanne M. Santini, and Jillian F. Banfield. Clades of huge phages from across Earth's ecosystems. *Nature*, 578(7795):425–431, February 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2007-4. URL https://www.nature.com/articles/s41586-020-2007-4.

[9] Teagan L Brown, Oliver J Charity, and Evelien M Adriaenssens. Ecological and functional roles of bacteriophages in contrasting environments: Marine, terrestrial and human gut. *Current Opinion in Microbiology*, 70:102229, December 2022. ISSN 1369-5274. doi: 10.1016/j.mib.2022.102229. URL https://www.sciencedirect.com/science/article/pii/S1369527422001138.

[10] Sandra Chibani-Chennoufi, Anne Bruttin, Marie-Lise Dillmann, and Harald Brüssow. Phage-Host Interaction: An Ecological Perspective. *Journal of Bacteriology*, 186(12):3677–3686, June 2004. ISSN 0021-9193. doi: 10.1128/JB.186.12.3677-3686.2004. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC419959/.

[11] Sebastián Coloma, Ursula Gaedke, Kaarina Sivonen, and Teppo Hiltunen. Frequency of virus-resistant hosts determines experimental community dynamics. *Ecology*, 100(1):e02554, 2019. ISSN 1939-9170. doi: 10.1002/ecy.2554. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ecy.2554.

[12] Stephen Tucker and Peter Pollard. Identification of Cyanophage Ma-LBP and Infection of the Cyanobacterium Microcystis aeruginosa from an Australian Subtropical Lake by the Virus. *Applied and Environmental Microbiology*, 71(2):629–635, February 2005. doi: 10.1128/AEM.71.2.629-635.2005. URL https://journals.asm.org/doi/10.1128/aem.71.2.629-635.2005.

[13] Stephen Odonkor and Kennedy Addo. Bacteria Resistance to Antibiotics: Recent Trends and Challenges. *International Journal of Biological & Medical Research*, pages 1204–1210, January 2011.

[14] Angelina Volkova, Kelly Ruggles, Anjelique Schulfer, Zhan Gao, Stephen D. Ginsberg, and Martin J. Blaser. Effects of early-life penicillin exposure on the gut microbiome and frontal cortex and amygdala gene expression. *iScience*, 24(7): 102797, July 2021. ISSN 2589-0042. doi: 10.1016/j.isci.2021.102797. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8324854/.

[15] Nana Nguefang Laure and Juhee Ahn. Phage resistance-mediated trade-offs with antibiotic resistance in Salmonella Typhimurium. *Microbial Pathogenesis*, 171: 105732, October 2022. ISSN 08824010. doi: 10.1016/j.micpath.2022.105732. URL https://linkinghub.elsevier.com/retrieve/pii/S088240102200345X.

[16] Yuanyang Zhao, Mei Shu, Ling Zhang, Chan Zhong, Ningbo Liao, and Guoping Wu. Phage-driven coevolution reveals trade-off between antibiotic and phage resistance in Salmonella anatum. *ISME Communications*, 4(1):ycae039, January 2024. ISSN 2730-6151. doi: 10.1093/ismeco/ycae039. URL https://doi.org/10.1093/ismeco/ycae039.

[17] Beata Kowalska. Fresh vegetables and fruit as a source of Salmonella bacteria. *Annals of agricultural and environmental medicine: AAEM*, 30(1):9–14, March 2023. ISSN 1898-2263. doi: 10.26444/aaem/156765.

[18] Flat illustration of bacteriophage structures and anatomy. | Premium Vector. URL https://www.freepik.com/premium-vector/flat-illustration-bacteriophage-structures-anatomy_18751587.htm.

[19] Nicola Twilley. Inside the World of Viral Dark Matter. *The New Yorker*, February 2015. ISSN 0028-792X. URL https://www.newyorker.com/tech/annals-of-technology/phage-killer-viral-dark-matter.

[20] Viruses may play an unexplored role in the gut-brain axis. URL https://www.science.org/content/article/viruses-may-play-unexplored-role-gut-brain-axis.

[21] A. A. Lindberg. Bacteriophage receptors. *Annual Review of Microbiology*, 27:205–241, 1973. ISSN 0066-4227. doi: 10.1146/annurev.mi.27.100173.001225.

[22] Edel Stone, Katrina Campbell, Irene Grant, and Olivia McAuliffe. Understanding and Exploiting Phage–Host Interactions. *Viruses*, 11(6):567, June 2019. ISSN 1999-4915. doi: 10.3390/v11060567. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6630733/.

[23] Dalton V. Banh, Cameron G. Roberts, Adrian Morales-Amador, Brandon A. Berryhill, Waqas Chaudhry, Bruce R. Levin, Sean F. Brady, and Luciano A. Marraffini. Bacterial cGAS senses a viral RNA to initiate immunity. *Nature*, 623(7989):1001–1008, November 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06743-9. URL https://www.nature.com/articles/s41586-023-06743-9.

[24] Asaf Levy, Moran G. Goren, Ido Yosef, Oren Auster, Miriam Manor, Gil Amitai, Rotem Edgar, Udi Qimron, and Rotem Sorek. CRISPR adaptation biases explain preference for acquisition of foreign DNA. *Nature*, 520(7548):505–510, April 2015. ISSN 1476-4687. doi: 10.1038/nature14302. URL https://www.nature.com/articles/nature14302.

[25] Joanna Warwick-Dugdale, Holger H. Buchholz, Michael J. Allen, and Ben Temperton. Host-hijacking and planktonic piracy: How phages command the microbial high seas. *Virology Journal*, 16(1):1–13, December 2019. ISSN 1743-422X. doi: 10.1186/s12985-019-1120-1. URL https://virologyj.biomedcentral.com/articles/10.1186/s12985-019-1120-1.

[26] P. M. Sharp. Molecular evolution of bacteriophages: Evidence of selection against the recognition sites of host restriction enzymes. *Molecular Biology and Evolution*, 3(1):75–83, January 1986. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a040377.

[27] Matthew K Waldor and David I Friedman. Phage regulatory circuits and virulence gene expression. *Current Opinion in Microbiology*, 8(4):459–465, August 2005. ISSN 1369-5274. doi: 10.1016/j.mib.2005.06.001. URL https://www.sciencedirect.com/science/article/pii/S1369527405000755.

[28] Louis-Charles Fortier and Ognjen Sekulovic. Importance of prophages to evolution and virulence of bacterial pathogens. *Virulence*, 4(5):354–365, July 2013. ISSN 2150-5594. doi: 10.4161/viru.24498. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3714127/.

[29] Anastasia A. Aksyuk and Michael G. Rossmann. Bacteriophage Assembly. *Viruses*, 3(3):172–203, February 2011. ISSN 1999-4915. doi: 10.3390/v3030172. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3185693/.

[30] I. N. Wang, D. L. Smith, and R. Young. Holins: The protein clocks of bacteriophage infections. *Annual Review of Microbiology*, 54:799–825, 2000. ISSN 0066-4227. doi: 10.1146/annurev.micro.54.1.799.

[31] Richard E Lenski. TWO-STEP RESISTANCE BY ESCHERICHIA COLI B TO BACTERIOPHAGE T2. *Genetics*, 107(1):1–7, May 1984. ISSN 1943-2631. doi: 10.1093/genetics/107.1.1. URL https://doi.org/10.1093/genetics/107.1.1.

[32] Laith Harb, Karthik Chamakura, Pratick Khara, Peter J. Christie, Ry Young, and Lanying Zeng. ssRNA phage penetration triggers detachment of the F-pilus. *Proceedings of the National Academy of Sciences of the United States of America*, 117 (41):25751–25758, October 2020. ISSN 0027-8424. doi: 10.1073/pnas.2011901117. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7568308/.

[33] Sanju Tamang. Horizontal Gene Transfer in Prokaryotes and Eukaryotes, September 2023. URL https://microbenotes.com/horizontal-gene-transfer-prokaryotes-eukaryotes/.

[34] Laura M. Kasman and La Donna Porter. Bacteriophages. In *StatPearls*. StatPearls Publishing, Treasure Island (FL), 2025. URL http://www.ncbi.nlm.nih.gov/books/NBK493185/.

[35] Demeng Tan, Sine Lo Svenningsen, and Mathias Middelboe. Quorum Sensing Determines the Choice of Antiphage Defense Strategy in Vibrio anguillarum. *mBio*, 6(3):10.1128/mbio.00627–15, June 2015. doi: 10.1128/mbio.00627-15. URL https://journals.asm.org/doi/10.1128/mbio.00627-15.

[36] Avinoam Rabinovitch, Ira Aviram, and Arieh Zaritsky. Bacterial debris—an ecological mechanism for coexistence of bacteria and their viruses. *Journal of Theoretical Biology*, 224(3):377–383, October 2003. ISSN 0022-5193. doi: 10.1016/S0022-5193(03)00174-7. URL https://www.sciencedirect.com/science/article/pii/S0022519303001747.

[37] James J. Bull, Kelly A. Christensen, Carly Scott, Benjamin R. Jack, Cameron J. Crandall, and Stephen M. Krone. Phage-Bacterial Dynamics with Spatial Structure: Self Organization around Phage Sinks Can Promote Increased Cell Densities. *Antibiotics*, 7(1):8, March 2018. ISSN 2079-6382. doi: 10.3390/antibiotics7010008. URL https://www.mdpi.com/2079-6382/7/1/8.

[38] Anika Gupta, Norma Morella, Dmitry Sutormin, Naisi Li, Karl Gaisser, Alexander Robertson, Yaroslav Ispolatov, Georg Seelig, Neelendu Dey, and Anna Kuchina. Combinatorial phenotypic landscape enables bacterial resistance to phage infection, January 2025. URL https://www.biorxiv.org/content/10.1101/2025.01.13.632860v1.

[39] Stephen T. Abedon. Phage "delay" towards enhancing bacterial escape from biofilms: A more comprehensive way of viewing resistance to bacteriophages. *AIMS*

Microbiology, 3(microbiol-03-00186):186–226, 2017. ISSN 2471-1888. doi: 10.3934/microbiol.2017.2.186. URL http://www.aimspress.com/article/doi/10.3934/microbiol.2017.2.186.

[40] Rasmus Skytte Eriksen, Sine L. Svenningsen, Kim Sneppen, and Namiko Mitarai. A growing microcolony can survive and support persistent propagation of virulent phages. *Proceedings of the National Academy of Sciences*, 115(2):337–342, January 2018. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1708954115. URL https://pnas.org/doi/full/10.1073/pnas.1708954115.

[41] Christoph Lohrmann, Christian Holm, and Sujit S. Datta. Influence of bacterial swimming and hydrodynamics on attachment of phages. *Soft Matter*, 20(24):4795–4805, June 2024. ISSN 1744-6848. doi: 10.1039/D4SM00060A. URL https://pubs.rsc.org/en/content/articlelanding/2024/sm/d4sm00060a.

[42] Pramalkumar H. Patel, Véronique L. Taylor, Chi Zhang, Landon J. Getz, Alexa D. Fitzpatrick, Alan R. Davidson, and Karen L. Maxwell. Anti-phage defence through inhibition of virion assembly. *Nature Communications*, 15(1):1644, February 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-45892-x. URL https://www.nature.com/articles/s41467-024-45892-x.

[43] Michael J. Bucher and Daniel M. Czyż. Phage against the Machine: The SIE-ence of Superinfection Exclusion. *Viruses*, 16(9):1348, August 2024. ISSN 1999-4915. doi: 10.3390/v16091348.

[44] Shuji Kanamaru, Kazuya Uchida, Mai Nemoto, Alec Fraser, Fumio Arisaka, and Petr G. Leiman. Structure and Function of the T4 Spackle Protein Gp61.3. *Viruses*, 12(10):1070, September 2020. ISSN 1999-4915. doi: 10.3390/v12101070. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7650644/.

[45] Justin C. Leavitt, Brianna M. Woodbury, Eddie B. Gilcrease, Charles M. Bridges, Carolyn M. Teschke, and Sherwood R. Casjens. Bacteriophage P22 SieA-mediated superinfection exclusion. *mBio*, 15(2):e02169–23, January 2024. doi: 10.1128/mbio.02169-23. URL https://journals.asm.org/doi/10.1128/mbio.02169-23.

[46] Yuval Mulla, Janina Müller, Denny Trimcev, and Tobias Bollenbach. Extreme diversity of phage amplification rates and phage-antibiotic interactions revealed by PHORCE, December 2024. URL https://www.biorxiv.org/content/10.1101/2024.06.07.597930v3.

[47] Jacob Beal, Natalie G. Farny, Traci Haddock-Angelli, Vinoo Selvarajah, Geoff S. Baldwin, Russell Buckley-Taylor, Markus Gershater, Daisuke Kiga, John Marken, Vishal Sanchania, Abigail Sison, and Christopher T. Workman. Robust estimation

of bacterial cell count from optical density. *Communications Biology*, 3(1):512, September 2020. ISSN 2399-3642. doi: 10.1038/s42003-020-01127-5. URL https://www.nature.com/articles/s42003-020-01127-5.

[48] Portia Mira, Pamela Yeh, and Barry G. Hall. Estimating microbial population data from optical density. *PLOS ONE*, 17(10):e0276040, October 2022. ISSN 1932-6203. doi: 10.1371/journal.pone.0276040. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0276040.

[49] Advanced Wastewater Modelling | GPS-X - Hydromantis. URL https://www.hydromantis.com/GPSX-innovative.html.

[50] Jacqueline Heard, Emma Harvey, Bruce B. Johnson, John D. Wells, and Michael J. Angove. The effect of filamentous bacteria on foam production and stability. *Colloids and Surfaces. B, Biointerfaces*, 63(1):21–26, May 2008. ISSN 0927-7765. doi: 10.1016/j.colsurfb.2007.10.011.

[51] Brendan J. M. Bohannan and Richard E. Lenski. Effect of Resource Enrichment on a Chemostat Community of Bacteria and Bacteriophage. *Ecology*, 78(8):2303–2315, 1997. ISSN 1939-9170. doi: 10.1890/0012-9658(1997)078[2303:EOREOA]2.0.CO;2. URL https://onlinelibrary.wiley.com/doi/abs/10.1890/0012-9658%281997%29078%5B2303%3AEOREOA%5D2.0.CO%3B2.

[52] Richard E. Lenski. Dynamics of Interactions between Bacteria and Virulent Bacteriophage. In K. C. Marshall, editor, *Advances in Microbial Ecology*, pages 1–44. Springer US, Boston, MA, 1988. ISBN 978-1-4684-5409-3. doi: 10.1007/978-1-4684-5409-3_1. URL https://doi.org/10.1007/978-1-4684-5409-3_1.

[53] Allan Campbell. Conditions for the Existence of Bacteriophage. *Evolution*, 15(2):153–165, 1961. ISSN 0014-3820. doi: 10.2307/2406076. URL https://www.jstor.org/stable/2406076.

[54] J. S. Weitz, H. Hartman, and S. A. Levin. Coevolutionary arms races between bacteria and bacteriophage. *Proceedings of the National Academy of Sciences*, 102(27):9535–9540, July 2005. doi: 10.1073/pnas.0504062102. URL https://www.pnas.org/doi/abs/10.1073/pnas.0504062102.

[55] Matthew Scott, Carl W. Gunderson, Eduard M. Mateescu, Zhongge Zhang, and Terence Hwa. Interdependence of Cell Growth and Gene Expression: Origins and Consequences. *Science*, 330(6007):1099–1102, November 2010. doi: 10.1126/science.1192588. URL https://www.science.org/doi/10.1126/science.1192588.

[56] Luis S. Mayorga, Ignacio Cebrian, Meghna Verma, Stefan Hoops, and Josep Bassaganya-Riera. Reconstruction of endosomal organization and function by a

combination of ODE and agent-based modeling strategies. *Biology Direct*, 13(1): 1–21, December 2018. ISSN 1745-6150. doi: 10.1186/s13062-018-0227-4. URL https://biologydirect.biomedcentral.com/articles/10.1186/s13062-018-0227-4.

[57] S. M. Krone. Spatial models: Stochastic and deterministic. *Mathematical and Computer Modelling*, 40(3):393–409, August 2004. ISSN 0895-7177. doi: 10.1016/j.mcm.2003.09.037. URL https://www.sciencedirect.com/science/article/pii/S089571770480270X.

[58] Michael Klann and Heinz Koeppl. Spatial Simulations in Systems Biology: From Molecules to Cells. *International Journal of Molecular Sciences*, 13(6):7798–7827, June 2012. ISSN 1422-0067. doi: 10.3390/ijms13067798. URL https://www.mdpi.com/1422-0067/13/6/7798.

[59] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020. ISSN 1548-7105. doi: 10.1038/s41592-019-0686-2. URL https://www.nature.com/articles/s41592-019-0686-2.

[60] J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, March 1980. ISSN 0377-0427. doi: 10.1016/0771-050X(80)90013-3. URL https://www.sciencedirect.com/science/article/pii/0771050X80900133.

[61] Dash Documentation & User Guide | Plotly. URL https://dash.plotly.com/.

[62] I. M Sobol′. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1):271–280, February 2001. ISSN 0378-4754. doi: 10.1016/S0378-4754(00)00270-6. URL https://www.sciencedirect.com/science/article/pii/S0378475400002706.

[63] Python Software Foundation. Python programming language, version 3.11.6, 2024. URL https://www.python.org. Accessed: 2025-05-19.

[64] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van

Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

[65] Takuya Iwanaga, William Usher, and Jonathan Herman. Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses. *Socio-Environmental Systems Modelling*, 4:18155–18155, May 2022. ISSN 2663-3027. doi: 10.18174/sesmo.18155. URL https://sesmo.org/article/view/18155.

[66] Jon Herman and Will Usher. SALib: An open-source Python library for Sensitivity Analysis. *Journal of Open Source Software*, 2(9):97, January 2017. ISSN 2475-9066. doi: 10.21105/joss.00097. URL https://joss.theoj.org/papers/10.21105/joss.00097.

[67] Aric Hagberg, Pieter J. Swart, and Daniel A. Schult. Exploring network structure, dynamics, and function using NetworkX. Technical Report LA-UR-08-05495; LA-UR-08-5495, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), January 2008. URL https://www.osti.gov/biblio/960616.

[68] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[69] Bryan B. Hsu, Travis E. Gibson, Vladimir Yeliseyev, Qing Liu, Lorena Lyon, Lynn Bry, Pamela A. Silver, and Georg K. Gerber. Dynamic Modulation of the Gut Microbiota and Metabolome by Bacteriophages in a Mouse Model. *Cell Host & Microbe*, 25(6):803–814.e5, June 2019. ISSN 1934-6069. doi: 10.1016/j.chom.2019.05.001.

[70] Naomi Iris van den Berg, Daniel Machado, Sophia Santos, Isabel Rocha, Jeremy Chacón, William Harcombe, Sara Mitri, and Kiran R. Patil. Ecological modelling approaches for predicting emergent properties in microbial communities. *Nature Ecology & Evolution*, 6(7):855–865, July 2022. ISSN 2397-334X. doi: 10.1038/s41559-022-01746-7. URL https://www.nature.com/articles/s41559-022-01746-7.

[71] Raunak Dey, Ashley R. Coenen, Natalie E. Solonenko, Marie N. Burris, Anna I. Mackey, Julia Galasso, Christine L. Sun, David Demory, Daniel Muratore, Stephen J. Beckett, Matthew B. Sullivan, and Joshua S. Weitz. Emergent higher-order interactions enable coexistence in phage-bacteria community dynamics, May 2025. URL https://www.biorxiv.org/content/10.1101/2025.05.15.651590v1.

[72] Tom Clegg and Thilo Gross. Cross-feeding creates tipping points in microbiome diversity. *Proceedings of the National Academy of Sciences*, 122(19):e2425603122,

May 2025. doi: 10.1073/pnas.2425603122. URL https://www.pnas.org/doi/10.1073/pnas.2425603122.

[73] Lars Fieseler and Steven Hagens. Food Safety. In David R. Harper, Stephen T. Abedon, Benjamin H. Burrowes, and Malcolm L. McConville, editors, *Bacteriophages: Biology, Technology, Therapy*, pages 857–890. Springer International Publishing, Cham, 2021. ISBN 978-3-319-41986-2. doi: 10.1007/978-3-319-41986-2_29. URL https://doi.org/10.1007/978-3-319-41986-2_29.

[74] Heinz G. Floss and Tin-Wein Yu. RifamycinMode of Action, Resistance, and Biosynthesis. *Chemical Reviews*, 105(2):621–632, February 2005. ISSN 0009-2665. doi: 10.1021/cr030112j. URL https://doi.org/10.1021/cr030112j.

[75] A. Tomasz. The Mechanism of the Irreversible Antimicrobial Effects of Penicillins: How the Beta-Lactam Antibiotics Kill and Lyse Bacteria. *Annual Review of Microbiology*, 33(Volume 33, 1979):113–137, October 1979. ISSN 0066-4227, 1545-3251. doi: 10.1146/annurev.mi.33.100179.000553. URL https://www.annualreviews.org/content/journals/10.1146/annurev.mi.33.100179.000553.

[76] Sergei B. Vakulenko and Shahriar Mobashery. Versatility of Aminoglycosides and Prospects for Their Future. *Clinical Microbiology Reviews*, 16(3):430–450, July 2003. doi: 10.1128/cmr.16.3.430-450.2003. URL https://journals.asm.org/doi/10.1128/cmr.16.3.430-450.2003.

[77] Sophie Leclercq, Firoz M. Mian, Andrew M. Stanisz, Laure B. Bindels, Emmanuel Cambier, Hila Ben-Amram, Omry Koren, Paul Forsythe, and John Bienenstock. Low-dose penicillin in early life induces long-term changes in murine gut microbiota, brain cytokines and behavior. *Nature Communications*, 8:15062, April 2017. ISSN 2041-1723. doi: 10.1038/ncomms15062. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5382287/.

[78] Global action plan on antimicrobial resistance. URL https://www.who.int/publications/i/item/9789241509763.

[79] Christopher R. Grasso, Kaytee L. Pokrzywinski, Christopher Waechter, Taylor Rycroft, Yanyan Zhang, Alyssa Aligata, Michael Kramer, and Anisha Lamsal. A Review of Cyanophage–Host Relationships: Highlighting Cyanophages as a Potential Cyanobacteria Control Strategy. *Toxins*, 14(6):385, June 2022. ISSN 2072-6651. doi: 10.3390/toxins14060385. URL https://www.mdpi.com/2072-6651/14/6/385.

[80] Katelyn M. McKindles, Makayla A. Manes, Jonathan R. DeMarco, Andrew McClure, R. Michael McKay, Timothy W. Davis, and George S. Bullerjahn. Dissolved Microcystin Release Coincident with Lysis of a Bloom Dominated by Microcystis

spp. in Western Lake Erie Attributed to a Novel Cyanophage. *Applied and Environmental Microbiology*, 86(22):e01397–20, October 2020. doi: 10.1128/AEM.01397-20. URL https://journals.asm.org/doi/10.1128/aem.01397-20.

[81] (PDF) Economic Impacts of Red Tide Events on Restaurant Sales. URL https://www.researchgate.net/publication/23515658_Economic_Impacts_of_Red_Tide_Events_on_Restaurant_Sales.

[82] Yung Sung Cheng, Yue Zhou, Clinton M. Irvin, Richard H. Pierce, Jerome Naar, Lorraine C. Backer, Lora E. Fleming, Barbara Kirkpatrick, and Dan G. Baden. Characterization of Marine Aerosol for Assessment of Human Exposure to Brevetoxins. *Environmental Health Perspectives*, 113(5):638–643, May 2005. ISSN 0091-6765. doi: 10.1289/ehp.7496. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1257561/.

[83] Barbara Kirkpatrick, Judy A Bean, Lora E Fleming, Gary Kirkpatrick, Lynne Grief, Kate Nierenberg, Andrew Reich, Sharon Watkins, and Jerome Naar. Gastrointestinal Emergency Room Admissions and Florida Red Tide Blooms. *Harmful algae*, 9(1):82–86, January 2010. ISSN 1568-9883. doi: 10.1016/j.hal.2009.08.005. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2786186/.

[84] Porter Hoagland, Di Jin, Lara Y. Polansky, Barbara Kirkpatrick, Gary Kirkpatrick, Lora E. Fleming, Andrew Reich, Sharon M. Watkins, Steven G. Ullmann, and Lorraine C. Backer. The costs of respiratory illnesses arising from Florida gulf coast Karenia brevis blooms. *Environmental Health Perspectives*, 117(8):1239–1243, August 2009. ISSN 1552-9924. doi: 10.1289/ehp.0900645.