

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 fulfilment of the requirements
for the degree of Master of Science in Computational Science*

in the

Computational Science Lab
Informatics Institute

May 2025



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: May 13, 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

Include your abstract here Abstracts must include sufficient information for reviewers to judge the nature and significance of the topic, the adequacy of the investigative strategy, the nature of the results, and the conclusions. The abstract should summarize the substantive results of the work and not merely list topics to be discussed. Length 200-400 words.

Acknowledgements

I would like to thank my parents for eternally loving me and for financially supporting me through my Bachelor and Master studies. Thank you to Dr. Matti Gralka for the weekly meetings and teaching me everything about phages and bacteria. Every meeting was always insightful, productive, and informative. I will forever be amazed at how he can remember which paper talks about which topic, and how he always had a paper for every topic. Thank you to Sofia Blaszczyk for finding this opening and suggesting that I email Dr. Gralka for an introductory meeting, and for acting as my rubber duck, and watching my cringe screen recordings that I sent her at 2am showcasing various demos of my code. Thank you to Dr. Rik Kaasschieter and Dr. Martijn Anthonissen for introducing me to Computational Sciences in my Bachelors program, as for without following their courses “Introduction Computational Sciences” and “Numerical Linear Algebra”, I would not have been interested in Computational Sciences and would not have found the MSc Computational Sciences program. Thank you to Sarah Flickinger for showing me the research that she has been doing in the lab. She allowed me to really connect my research and models to real life, reminding me that what I am doing has greater implications than just a purely theoretical or programming challenge. And finally, thank you to all of my friends, Indian computer science youtubers, and online resources for teachign me and pulling me through both of my programs.

Contents

Declaration of Authorship	i
Abstract	iii
Acknowledgements	iv
Contents	v
List of Figures	viii
List of Tables	xi
List of Algorithms	xii
Abbreviations	xiii
1 Introduction	1
1.1 Biological Background	2
2 Literature review	4
2.1 Methods of Modelling Phages and Bacteria	4
2.1.1 Generalized Lotka-Volterra Model	5
2.1.2 Generalized Consumer-Resource Model	5
2.1.3 Trait-Based Model	5
2.1.4 Agent-Based Models	5
2.2 Biology of Phages	7
2.2.1 What Are Phages?	7
2.2.2 How Does the Phage Cycle Work?	7
2.2.2.1 Infection Stage	7
Attachment	7
Entry of Phage Into Bacteria	7
2.2.2.2 Lysogenic Cycle	7
Repression of DNA	7
Integration of Phage DNA Into Bacteria DNA	8
Cellular Replication	8

	Induction of Phage	8
2.2.2.3	Lytic Cycle	8
	Replication, Transcription, and Translation of the Phage DNA	8
	Assembly of Phage Parts	8
	Lysis of the Bacterial Cell	8
2.3	Bacterial Defense Against Phages	8
2.3.1	Mutations in Bacterial DNA (Genetic (Co-)Evolution)	8
2.3.2	Phage Inactivation and Decoys	9
2.3.3	CRISPR-Cas Methods	10
2.3.4	Phenotype Resistance	10
2.3.5	Spatial Refuge/Biofilms	10
2.3.6	Phage Counter Defense Against Bacteria	10
2.3.7	Genetic mutations	11
2.3.8	Viral recombination	11
2.4	Phage Defense Against Phages	11
2.4.1	Implications of Phage Against Phage Defense	12
2.4.2	Software Mathematically Modelling Phages, Bacteria, and Resources	12
2.4.2.1	Cocktail	12
2.4.2.2	PhageDyn	12
3	Methods	13
3.1	Project Overview	13
3.1.1	Part 1: Graphical User Interface (GUI) Tool	13
3.1.2	Part 2: Simulation Framework	13
3.1.3	Part 3: Analysis and Visualization	14
3.2	Network Topography of Interactions	14
3.3	Dashboard	15
3.3.1	Editing Network and Parameter Values	16
3.3.1.1	Initial Condition	16
3.3.2	Advanced Visualization and Analysis	18
3.3.2.1	Serial Transfer	18
3.3.2.2	Parameter Analysis	19
3.3.2.3	Initial Value Analysis	20
3.3.2.4	Parameter Space Slot	21
3.3.2.5	SOBOL Analysis	21
3.3.2.6	Ultimate Analysis	26
4	Experiments and results	29
5	Discussion	30
6	Conclusion and future work	31
6.1	Conclusion	31
6.2	Future Work	31
6.2.1	Other Models	31
6.2.1.1	Spatial simulations	32
PDE	33	

Discretization	33
7 Ethics and Data Management	34
8 Appendix A: Equation Parameters	35
9 Appendix B: Industrial and Real Life Applications of Phages	37
9.1 Controlling Foodborne Bacteria	38
9.1.1 Current Applications	38
9.2 Phage Therapy and Antibiotics	40
9.2.1 Current Applications: Bacterial Infection Control	40
9.3 Environmental Protection	41
9.3.1 Current Applications	42
Bibliography	45

List of Figures

1.1	Life cycle of a phage, inside and outside a bacteria cell.	3
2.1	The three main ways that a (dead) bacterium can transfer DNA over to another bacterium [1].	9
3.1	The GUI tool when you start up the program. There are numerous tools that you can use to edit the graph. By default, an environment node holding parameters such as pH and temperature is added. A settings node is added as well, holding settings data to be used for the solver like the type of solver (RK23 or RK45).	15
3.2	An arbitrary $3 \times 2 \times 3$ network. Phage 0 (P0) interacts with bacteria 0 (B0). P1 interacts with B0 and B1, etc. Finally, resource 0 (R0) interacts with R1 (an example of this would be a complex sugar degrading into a simple sugar over time). The nature of the interaction needs to be defined and captured in the parameter names, values, and ODE equations. It is up to the user to correctly define the interaction in the ODE model. Nothing can interact with the environment and setting node, they are simply there to hold data about the environment and network solver.	16
3.3	The panel where a user can edit the initial conditions of the agents. The columns of each table show for which agent the value corresponds to. A copy of this data is sent to the solver. This instance of the model models a 3 resource, 3 bacteria, and 2 phage system. The bacteria are split into an uninfected and an infected stage. Once infected by a phage, a bacterium will go through four intermediary infection stages before lysing. The infected stages are represented as a matrix, with each row representing a bacterial agent while each column represents the stage of infection.	17
3.4	The panel where the user can edit the attributes for a given attribute name. This version is specifically used represent the attributes that are associated with nodes, and are represented as a vector. The columns of each table show for which agent the value corresponds to. A copy of this data is sent to the solver.	17
3.5	The panel where the user can edit the attributes for a given attribute name. This version is specifically used to represent the attributes that are associated with edges between agents, and are represented as a matrix. The rows and columns of each table show for which agent the value corresponds to. Due to limitations with Dash datatables, the row names can't explicitly be labelled, however text above the table shows the row names. A copy of this data is sent to the solver.	17

3.6	The panel where a user can edit the edge attributes the environment settings. Each column represents a single environment parameter. A copy of this data is sent to the solver.	17
3.7	The panel where a user can edit the settings of the solver. Various options exist, such as solver type, cutoff value for small values, or tolerances of the solver. A copy of this data is not sent to the solver. The user needs to save the settings, while for the other panels the user does not need to save the changes.	18
3.8	The section where the user can set up the serial transfer. To adjust the values added to the agent populations, the user would need to edit the initial condition values.	20
3.9	Running the serial transfer updates the plot at the top of the page. The simulation initially ran for $t = 10$, and completed two serial transfers with a dilution rate of 10. The first transfer was completed without adding new phages and bacteria. The second transfer was done with adding new phages and bacteria. The R1 (red) and R0 (purple) resource died out at $t = 1.5$ and $t = 2.8$ but are reintroduced at $t = 10$ as part of the serial transfer process. All bacteria become infected and die out before or at $t = 10$. At $t = 10$ the phages have a value of roughly 818 and 1078 pre-serial transfer, and have a value of 81.8 and 107.8 post serial transfer. At $t = 20$, a second serial transfer occurs where the bacteria were reintroduced to the system, and more phages were added to the existing population. The stacked line plots show the absolute and relative distribution of different population groups. The stacked bar chart at the bottom shows the final population count of each agent group type at the end of each serial transfer. It might not always be feasible to experimentally determine the population count at each timestep in a lab, hence this graph can be used as a replacement to show the population count at serial transfer time.	21
3.10	The user can choose two parameters to run, along with the values they want to test. There is an option for a serial transfer, along with interpolating the values between the slider option.	22
3.11	An example run of the parameter analysis. The point in time to be shown in the heatmap can be changed by moving the slider above the heatmaps. This in turn affects the value shown in the heatmap.	22
3.12	A unique setting featured in the initial value analysis is the "run name" and the "percentage from the max value reached".	23
3.13	23
3.14	The user can select two starting values for the initial condition, but they can't choose vector, matrix, or environment settings due to the plot showing the development of agent populations against other agent populations. As typical, the user can select their own values or auto-generate values between two values, as well as use a serial transfer option. There is also an option to take the logarithm of the x and/or y axis.	24

3.15	An example run of a parameter space plot. Resources are plotted against phages, with a high washout rate of 0.05 explaining why the phage population is decreasing. For initial phage population of 60 and less, the phages “die out” due to the washout rate, while for an initial population count of 70 and higher the phages are able to proliferate. There is an initial population decrease due to the washout, but after the infected bacteria start to die out and release new phages, the population can grow.	24
3.16	For each parameter, the user can choose a range of values that SOBOL will use to test and analyze. If no values for a parameter are inputted, then the parameter is not included in the analysis. As each run can produce different time data of varying length, the number of timesteps has to be fixed by the user. Due to the randomness of the sampling method, a seed value can be provided to ensure repeatable results.	26
3.17	27
3.18	The section where a user can start an ultimate analysis, where they provide the values they want to test, and run the simulations. As an output, the user receives a <i>.parquet</i> file containing the parameter values that were changed for that specific and the <i>t</i> and <i>y</i> values for that specific run. A pickle file of a dictionary containing information on that run such as the parameter names and values chosen, the original graph data, and original parameter values.	28
6.1	Exponential growth curve vs logistic growth	33
9.1	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 [2].	39
9.2	<i>Salmonella</i> count in a mixture of 5 <i>Salmonella</i> 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. [3]	39
9.3	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). [4]	42

List of Tables

List of Algorithms

Abbreviations

CSL	Computational Sceince Lab
UvA	Universitiet van Amsterdam
ODE	Ordinary Differential Equation
DDE	Delay Differential Equation
PDE	Partial Differential Equation
BVP	Boundary Value Problem
ABM	Agent Based Modelling
ARD	Arms Race Dynamic
FSD	Fluctuating Selection Dynamics
SNP	Single Nucleotide Polymorphism
DNA	DeoxyriboNucleic Acid
CRISPR	Clustered Regularly Interspaced Short Palindromic Repeats
SIE	Superinfection Exclusion

Chapter 1

Introduction

Phages, small viruses that infect and lyse (kill) bacteria, are nature's natural anti-microbial defense. Researchers are trying to determine phage applications in controlling bacterial infections and spread. But first the interactions between phages and bacteria need to be known. There is particular interest in phage applications due to their applications in human and animal health. Phage cocktails are a medicine for sick patients with bacterial diseases, such as *E. coli*. A patient can intake a pill filled with specific phages that target *E. coli*. The phages will target the specific *E. coli* bacteria, but it will not affect the other bacteria and will not have any side effects on the body. There are 100 trillion microbes across 5,000 different types of bacteria strains in the human gut. Using medicine such as antibiotics can disrupt the intricate ecosystem of the gut microbiome, acting as a scorched-earth mechanism. Phages on the other hand specifically target a specific bacterial strain, acting as a sniper, with minimal to no effects to other bacteria. This can be used to control bacterial infections and cure people, or to prevent the spread of common bacteria in livestock. Farmers often raise livestock in tight spaces with a lack of sanitation facilities, increasing the risk of a disease spreading.

Phages have many uses in an industrial setting. Phages can be used to control the growth of bacteria like *Salmonella* while producing food in a factory [2, 5].

In an ecosystem like the ocean, the gut, or in soil, there are thousands of different microbes all interacting with one another or the surrounding environment. The interactions are complex, with many factors affecting the growth of bacteria, fungi, phages, and more. Not every interaction can be identified, and if an interaction has been identified, the associated parameter values are unknown and need to be experimentally derived. External factors, such as flooding, droughts, chemical spills, or introduction of new agents have a massive impact on the ecosystem. These events can add or remove nutrients from the

system, change environmental parameters such as the surrounding temperature, introduce competition, or create an imbalance in the population by killing agents. These effects can affect the larger ecosystem and food chain as a whole.

Not much is known about phages in large and complex communities between other phages, bacteria, resources, and the environment. There have been previous attempts to model the complex dynamics of the populations between phages, bacteria, and resources, with the environment using Ordinary Differential Equations (ODE) and Delay Differential Equations (DDE). However, these methods have mainly stayed with 1-to-1-to-1 models, meaning 1 phage, 1 bacteria, and 1 resource. Other methods such as Partial Differential Equations (PDE) or cellular models have been created in an attempt to model these types of dynamics. There are two main ways to model phage-bacteria dynamics: a spatial model or a non-spatial one. A spatial model means that phages and bacteria can move through space and interact with their neighbors, whereas in non-spatial models, the bacteria and phages are assumed to be in a well-mixed solution and no distinction is made in regard to neighbors or distances to other agents. Special considerations have to be accounted for with spatial models. Bacteria and phages can only interact when they are in proximity to each other. Only a percentage p of bacteria and phages interact with one another at time step t . Spatial models can potentially lead to more interesting and complex results but are limited to smaller populations and harder to develop, while non-spatial models are easier to develop and are more effective in modeling large populations. PDE and cellular models are types of spatial models, while ODEs and DDEs are types of non-spatial models.

1.1 Biological Background

Phages are small viruses on the order of 27-190 nm that infect and lyse (kill) specific bacteria. The phage cycle process starts with a phage coming into contact with a bacterium. Once it has identified an injection site, the phage can inject a strain of DNA into the bacteria. The DNA strand has two options: it can either merge into the bacterial DNA, allowing the phage's DNA strand to replicate alongside the bacteria as they reproduce. This process defines the Lysogenic Cycle. After a set amount of time, the DNA of the phage can unmerge and hijack the DNA replicating mechanism, creating multiple copies of itself, using the transcription, translation, and replication process to create multiple copies of itself. The phages begin to self-assemble inside the bacteria

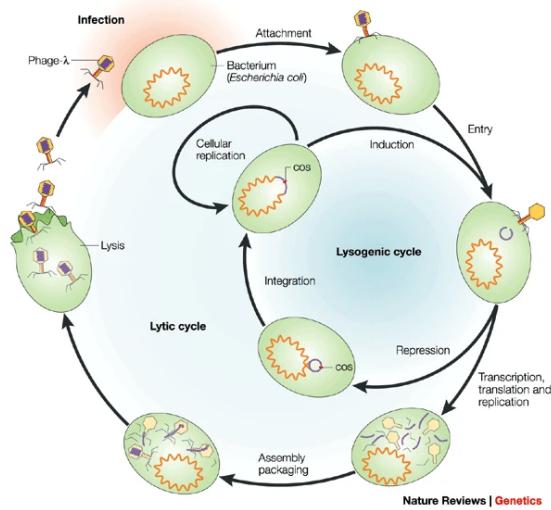


Figure 1.1: Life cycle of a phage, inside and outside a bacteria cell.

until the bacteria is full of phages and explodes, the lysis stage, releasing the phages into the environment, ready to repeat the process again.

This process can be visualized in Figure 1.1 [6].

Chapter 2

Literature review

2.1 Methods of Modelling Phages and Bacteria

There are numerous ways to model the interactions between phages and bacteria. Models can be built at a molecular level, where the model simulates the mechanical and chemical behavior of a phage as it interacts with the surface of a bacterium using computational chemistry methods. On the other end of the spectrum, a different type of model can be built where populations of phages, bacteria, and resources can be modeled using Ordinary Differential Equations (ODEs) or Delay Differential Equations (DDEs). DDEs are similar to ODEs, except where when ODEs are calculating the values of the equations at time t using time $t - 1$, DDEs can, but don't have to, use the value of the equation at time $t - \tau$, where $1 \leq \tau \leq t$. DDEs are a generalized version of ODEs.

Each type of system has its pros and cons. With the molecular level model, the model is more complex and needs significantly more startup time, simulation time, and is in general much more complex. However, more information can be gained from the simulations and can guide research in creating phages for a certain type of bacteria. The ODE method is simpler and easier to set up, however it can only capture large population dynamics. Certain assumptions about the community interactions have to be made. For example, ω percent of the bacteria population is washed out. The model can be made more complicated, by modelling each stage of the phage replication and lysis process, or instead of assuming exponential growth, there is a maximum carrying capacity of the population. The model can be further altered by using a normally distributed variable $\mathbf{N}(\mu = \omega, \sigma = 1)$ to account for noise when measuring the data. Ensuring the use of a seed value will ensure that each run of the model results in the same output.

2.1.1 Generalized Lotka-Voltera Model

The Lotka-Voltera model, a first-order non-linear differential model, is a model that captures the dynamics between predators and prey, with phages being the predator and bacteria being the prey. Any population can be modelled as such:

$$\frac{dB_i}{dt} = B_i \left(\left(r_i + \sum_j^N \alpha_{ij} B_j \right) - m_i \right)$$

where \dots .

2.1.2 Generalized Consumer-Resource Model

The generalized Consumer-Resource Model models the growth of a population and resource dynamics between a population of bacteria B_i and a resource R_i .

$$\frac{dB_i}{dt} = r_i B_i \left(\sum_{\alpha} \Delta w_{i\alpha} C_{i\alpha} R_{\alpha} \right) - m_i B_i \quad (2.1)$$

$$\frac{R_{\beta}}{dt} = - \sum_i C_{i\beta} R_{\beta} B_i + \sum_{\alpha,i} D_{\beta\alpha}^i C_{i\alpha} R_{\beta} B_i \quad (2.2)$$

$$\Delta w_{i\alpha} = \sum_{\beta} D_{\beta\alpha}^i w_{\beta} \quad (2.3)$$

2.1.3 Trait-Based Model

The Trait-Based Model is a model that takes into account external factors such as the temperature or pH of the system and can be modeled as follows:

$$\frac{dB_i}{dt} = (r_i - m_i) B_i \quad (2.4)$$

$$r_i = \frac{r_{i\alpha}^{max} R_{\alpha}}{R_{\alpha} + K_{i\alpha}} e^{S_i(T - T_{ref})} \quad (2.5)$$

where S_i is the sensitivity to B_i to factor T , and with trade off if $r_i^{max} > \text{mean } r^{max}$ then $S_i > \text{mean } S$.

2.1.4 Agent-Based Models

Agent-based Models (ABM) model the system through space and time. An $x \times y \times z$ grid (often z is left out for a 2D system) is created and split into smaller subcells containing resources and microbes. Each cell acts as its own tiny environment, where resources and

microbes interact within the environment, but not with the neighboring cells. Resources diffuse through the system using a PDE solver for a Boundary Value Problem (BVP). Agents can move into neighboring grids with a probability p , where p can depend on any number of parameters such as nutrient density, microbe density, or stochastic chance. ABMs are useful when simulating many individual elements interacting in a system. Chaotic or emergent behavior can arise from these interactions. Chaotic behavior refers to the irregular and unpredictable evolution of a system's behavior due to nonlinear equations, exhibiting sensitive dependence on initial conditions [7].

Emergent behavior is behavior that arises from the interactions of various agents in a system, that was not explicitly programmed into the system. The behavior can be beneficial, neutral, or harmful, but it can not be predicted until it arises, *if* it arises. Agents can have simple rules, but when interacting with other agents, behavior that hasn't been programmed can arise. Sometimes, people consider systems with emergent behaviors more complex than the sum of their parts.

$$\frac{\delta R_\alpha(r, t)}{\delta t} = \nabla [D(R_\alpha, r) \nabla R_\alpha(r, t)], r = (x, y) \quad (2.6)$$

, where r is a function of cell position (x, y) , and t represents time. The cellular agents rules are as follows:

$$\frac{di}{dt} = r_i \left(\sum_{\alpha} \Delta w_{i\alpha} C_{i\alpha} R_{\alpha} \right) \quad (2.7)$$

, where if $i >$ threshold, $\frac{i}{2}$ expands into the neighboring grid cell with a probability p . The system consumes resources and converts them into new sub-resource types with the following equation:

$$\frac{dR_{\alpha}}{dt} = - \sum_i C_{i\alpha} R_{\alpha} I \quad (2.8)$$

$$\frac{dR_{\beta}}{dt} = \sum_i C_{i\beta} R_{\beta} I + \sum_{\alpha, i} D_{\beta\alpha}^i C_{i\alpha} R_{\alpha} i \quad (2.9)$$

2.2 Biology of Phages

2.2.1 What Are Phages?

2.2.2 How Does the Phage Cycle Work?

There are 3 main parts to the phage-bacteria host cycle, the Infection stage, the Lysogenic cycle, and the Lytic cycle. In the infection stage, a phage floating through the environment detects and attaches to the surface of a bacteria cell. Once injected, the phage can control of the cell directly goes into the Lysogenic cycle or into the Lytic cycle. The phage proceeds to inject its DNA into the bacteria.

In the Lysogenic cycle, the phage DNA injects integrates into the genome of the bacteria. As the bacteria undergoes cellular replication, the DNA of the phage will be copied with the cell. After a set amount of time, the phage DNA can cut itself from the genome and enters the Lytic cycle.

In the Lytic cycle, the phage hijacks the cellular process of the bacteria. The phage DNA hijacks the replication, transcription, and replication process of the cell, making more and more copies of phage. The phage parts build together to make a full part. Eventually the cellwall bursts releasing the phages into the environment ready to infect more bacteria.

2.2.2.1 Infection Stage

The infection stage is characterized as the searching for a bacterium, detection, and subsequent attachment and injection of DNA into the bacteria.

Attachment

Entry of Phage Into Bacteria Injection of DNA into bacteria.

2.2.2.2 Lysogenic Cycle

Repression of DNA Represses DNA.

Integration of Phage DNA Into Bacteria DNA Integrate DNA.

Cellular Replication Cellular replication of the bacteria

Induction of Phage Phage DNA leaves bacteria DNA.

2.2.2.3 Lytic Cycle

Replication, Transcription, and Translation of the Phage DNA Replication of phage DNA.

Assembly of Phage Parts Assembly into a phage.

Lysis of the Bacterial Cell The lysis of the bacterial cell wall.

2.3 Bacterial Defense Against Phages

There is a constant battle between phages and bacteria. The bacteria don't want to be killed by the phages, so they adapt defenses such as thickening of the cell wall, or once the viral DNA has integrated with the bacteria's DNA, the bacteria will cut the viral DNA out of their DNA using CRISPR. [8]

2.3.1 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 infect the bacteria cell. Mutations might not always work, however. They can be partially effective if full effectiveness requires multiple steps to achieve, which can occasionally fail [9] or the mutation brings a cost to the bacteria cell by losing receptors on the cell wall.

Bacteria can horizontally transfer DNA to other bacteria on contact. There are three primary ways of this happening, which can be visualized in detail in Figure 2.1.

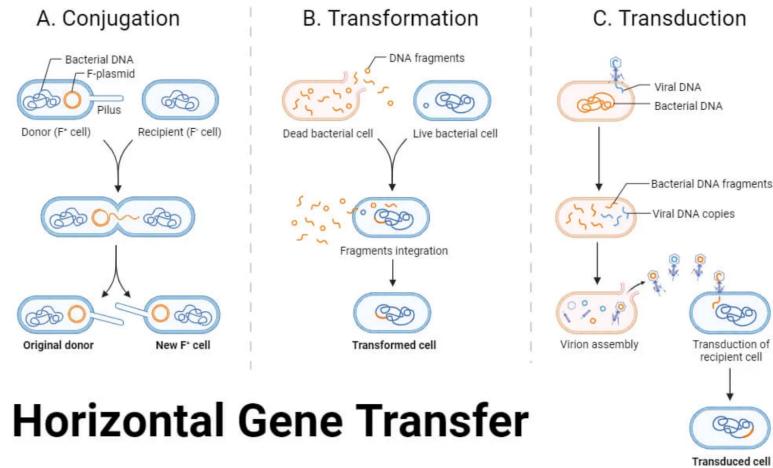


Figure 2.1: The three main ways that a (dead) bacterium can transfer DNA over to another bacterium [1].

The first method is via conjugation, where a donor cell donates DNA fragments using a mechanism called the F-factor or plasmid with a pilus. The pilus on the donor cell connects to the recipient cell and the DNA can be transferred to the receiver cell.

The second method, called transformation, occurs when a cell takes up DNA fragments released by a dead or degrading cell. Once inside the receiver cell, the donor DNA can integrate itself with the receiver DNA.

The third method is via transduction. When a phage is assembling in the cell just before lysis, the phage can collect a piece of the host's DNA instead of its own DNA. The dying bacterium proceeds to lyse, releasing the phages. The phage with the now dead host's DNA can infect the next bacteria, injecting the DNA strand of the now dead cell into the new host cell. The old bacterial DNA will proceed to integrate with the new host cell's DNA [1].

2.3.2 Phage Inactivation and Decoys

Bacteria can further protect themselves by producing decoys that the phage will attach to instead of themselves, inactivating the phage. Freshly lysed bacteria can still contain biomarkers that phages use to detect the bacteria, but upon injection, nothing happens as the cell doesn't function anymore. Bacteria can also produce proteolytic enzymes that will damage the proteins found in a phage [10]. Some bacteria can produce outer membrane vesicles that phages can absorb to, and later detach the vesicle with the phage [11]. The vesicle will proceed to float away with the phage attached/injected inside, posing no risk to the source bacteria or to other bacteria. It is suspected that the impact of these vesicles acting as a sink is minor [12], but helpful nonetheless.

2.3.3 CRISPR-Cas Methods

CRISPR is a gene editing tool that cells can use to cut out specified/unwanted parts of a DNA strand. Researchers are commonly using CRISPR to genetically engineer plants and animals to have specific features. Strands of DNA can be selectively added or removed from a DNA strand to achieve a better, more desired DNA strand. Specialized defenses in the bacteria can detect unwanted strands and remove the strand, acting as a line of defense against phages.

2.3.4 Phenotype Resistance

2.3.5 Spatial Refuge/Biofilms

Usually bacteria and phages coexist in well mixed environments such as the ocean, however some environments offer natural structure for bacteria to hide in. These structures can range from physical structure, like reeds in a lake, where the water is stagnant and harder for the phages to diffuse through, to biochemical structures like biofilms, where phages can't diffuse through the biofilm.

Circular bacterial colonies on an agar plate protect the inner bacteria from external phage infections [13]. Phages can not swim and do not contain any parts to move. They rather rely on passive forms of movement, such as diffusion through the environment or by mixing from environmental factors, such as changes in pressure or heat gradients [14]. Phages rely on Brownian motion, the seemingly random movement of small particles throughout a medium due to other microscopic particles interacting and bouncing off of one another [15]. Unlike phages, bacteria have the ability to actively move through the environment, and can use this to their advantage by swimming away. Bacteria and other microbial communities create biofilms, a layer of mucus containing various microbes. The thick mucus, microbes, and other spatial effects help protect the bacteria in the biofilm from external phages by making it hard for the phages to penetrate and diffuse through the mucus [16].

2.3.6 Phage Counter Defense Against Bacteria

With some of the defenses that bacteria have developed, phages are always mutating to counter their defenses. If phages don't adapt to the ever-changing bacterial defenses, the phages will experience an extinction event due to their inability to infect and grow their population count. It essentially becomes a race to the bottom, seeing who can out-adapt the other. However, if the phages out-adapt the bacteria too much, the bacteria die out,

then eventually the phages die out due to not having any bacteria left to infect. This can be avoided if the phages can adapt to target a second strain of bacteria, but this is unlikely. On the other hand, if bacteria out-adapt the phages, that is no problem for the bacteria because they don't need the phages to survive, and can keep on growing, limited only by the available space and nutrients.

This is a problem intrinsic to predator-prey systems, namely that the predators are dependent on the prey. Once the prey disappear, the predators also disappear. If the prey population goes down, and as a result the predator population goes down and extincts itself, the prey can come back without the threat of predators.

Phages face this exact same problem: the complete removal of either the bacteria or phages will lead to the removal of the phages from the system unless reintroduced.

2.3.7 Genetic mutations

2.3.8 Viral recombination

<https://www.sciencedirect.com/science/article/pii/S1931312821004170> <https://pmc.ncbi.nlm.nih.gov/>

Multiple phages can infect a cell and replicate itself using the cells internal replication process. Each phage has its own building blocks. Phage 1 could have long legs, a long neck, and a small head, while phage 2 can have long legs, a long neck, and a medium-sized head. When the phages are building copies of themselves, they could accidentally use the body parts of other phages. The primary method for proteins to bond with other proteins and molecules is via hydrogen bonds. These attractive forces hold proteins and other molecules in defined positions, and a change in molecule shape will change the bonds, which will force the other molecule to undergo changes in shape. If the proteins that build the subparts of each phage have similar chemical properties, they can be swapped between phages. This allows for biological diversity to spread throughout a phage population. Each phage body part can have unique characteristics such as better attachment rate, larger DNA storage capsule, or better probability of injection.

Coexistence between phages and bacteria via genetic co-evolution seems unlikely due to trade-offs imposed by the new mutations [17].

2.4 Phage Defense Against Phages

Some phages can employ defenses against other phages from infecting the bacterial cell. This is called superinfection exclusion (SIE), where a phage that has integrated with

the bacterial DNA, called a prophage, prevents a secondary infection from a similar or closely related phage [18]. There are various methods of preventing further infections. The phage 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 [19]. Other phages like the T4 phage can use proteins like the Spackle protein. The protein inhibits the lysozyme activity used in the process of DNA injection by other phages [19, 20]. Finally, 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 [21].

2.4.1 Implications of Phage Against Phage Defense

SIE can affect the speed and development of phage and bacterial populations. A phage restricting other phages from infecting the bacteria creates a competitive environment and can outcompete and dominate the other population. This is commonly seen in wildlife populations, where invading species can outcompete other species by eating more food/other species faster, breeding at a faster rate than other species, and having no natural predators.

2.4.2 Software Mathematically Modelling Phages, Bacteria, and Resources

Some software currently exists with the intended goal of modelling phage-bacteria-resource dynamics. Cocktail [22] and PhageDyn [23], a Java applet that interacts with existing files in GPS-X [24] to incorporate phage dynamics into models of wastewater treatment plants. PhageDyn itself does not model phage dynamics.

2.4.2.1 Cocktail

[22]

2.4.2.2 PhageDyn

[23]

Chapter 3

Methods

3.1 Project Overview

To help complete this Master thesis, I created various tools that would help create the final model outputs. The project is divided into three logical parts.

3.1.1 Part 1: Graphical User Interface (GUI) Tool

The first part involves the development of a GUI tool to create and edit the network topography of interactions. This tool allows users to quickly and intuitively define agents, interaction parameters, environmental parameters and setting parameters. It provides functionalities for adding, editing, and visualizing nodes and edges, as well as importing and exporting the network structure. The tool created a network graph which is then used for part 2 and part 3.

3.1.2 Part 2: Simulation Framework

The second part focuses on the simulation framework. The user provides an ODE model and the network topography as an input to the framework. The framework uses numerical solvers to simulate the interactions of the network over time. As output, the user receives two outputs. 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 of every agent.

3.1.3 Part 3: Analysis and Visualization

The third part involves analyzing and visualizing the simulation results. The user can use a dashboard built using Plotly Dash to interact with the solver and network. The user can change parameter, environment values, and setting values on the fly. This allows for the user to quickly change parameter values and test different situations. The dashboard includes various starter plots that allows for the user to test the model.

3.2 Network Topography of Interactions

In a microbial environment, there are numerous interactions occurring between agents. However not every agent can and will interact with one another. Based on which agents interact with one another, a network topography can be created, capturing the dynamics of the interactions. Every node represents a unique agent. An edge links agents together if there is an interaction occurring between the agents. The network allows for self-loops.

Each node contains attributes and properties intrinsic to that agent. For example, this would include the starting population or concentration, reproduction speed (if any), or death rate (if any). Each edge likewise also contains attributes to capture the unique dynamic interactions between the agents. This could be the probability for a successful interaction, the burst size of a specific phage-bacteria pair, or the bacteria consumption rate of resources. Adding the attributes to the nodes and edges allow for the capture of various interaction dynamics within the context of the community. The interactions between the agents can be visualized and edited using a GUI tool.

A GUI tool has been developed using Python and NetworkX to help aid in the development of this network topography. With this tool, a network topography can be created by adding any number of agents of varying type, such as bacteria, phages, or resources. There is an environment node that is used to store global environmental data, for example the temperature of the system, the pH of the system, washout rate, etc. There is a settings node that holds information such as simulation length, max timestep, and type of ODE solver to use. The attributes of the agents, interactions, and environment can easily be edited using the GUI tool.

Figure 3.1 shows the layout of the very simple tool build using tkinter, matplotlib, and NetworkX. Although the button labels are self-explanatory, the buttons allow the user to add exactly 1 node of either type “P” for phage, “B” for bacteria, or “R” for

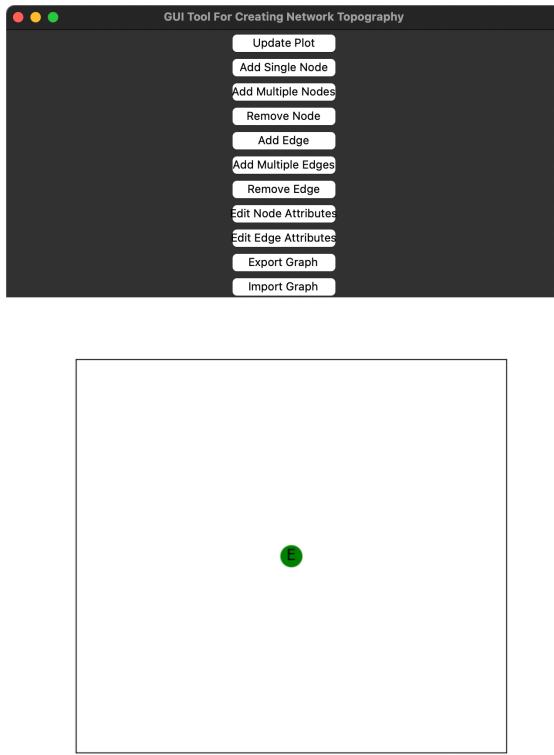


Figure 3.1: The GUI tool when you start up the program. There are numerous tools that you can use to edit the graph. By default, an environment node holding parameters such as pH and temperature is added. A settings node is added as well, holding settings data to be used for the solver like the type of solver (RK23 or RK45).

resource and provide a name. That is however tedious for large graphs, so the user can add multiple nodes at the same time. The newly created node is provided with default parameter values that the user has to provide beforehand. This can be done by importing the tool as one would import any other Python package. The user provides a base class, extending the class of the tool. The user can proceed to override the default implementation of the method that gives the default value. It is of course possible to add single or multiple edges at once, remove edges, edit node and edge attributes, and import or export the graph.

3.3 Dashboard

The dashboard allows for the user to interact with the network, the model, and some prebuilt visualizations, and is built into three logical sections. The first section allows for the user to edit the network parameters and setting values on the fly to quickly iterate through different conditions and to fine tune parameter selection without having to rebuild the network using the GUI tool. The second section allows for the user to see

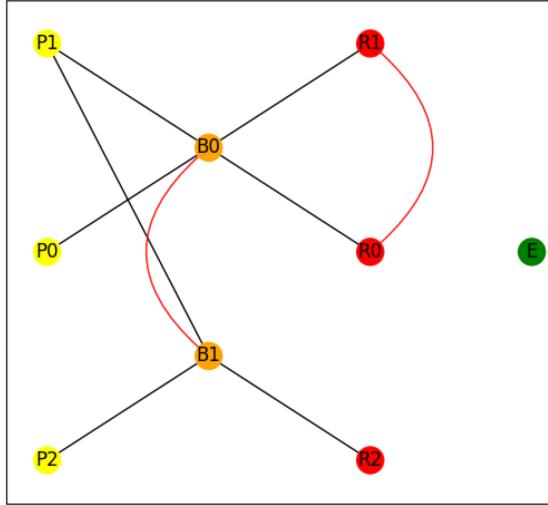


Figure 3.2: An arbitrary $3 \times 2 \times 3$ network. Phage 0 (P0) interacts with bacteria 0 (B0). P1 interacts with B0 and B1, etc. Finally, resource 0 (R0) interacts with R1 (an example of this would be a complex sugar degrading into a simple sugar over time). The nature of the interaction needs to be defined and captured in the parameter names, values, and ODE equations. It is up to the user to correctly define the interaction in the ODE model. Nothing can interact with the environment and setting node, they are simply there to hold data about the environment and network solver.

how the population count evolves over time for a given initial condition and parameter values, allowing to quickly test the network input. The final section allows for the user to run more advanced analysis on the network, for example by changing multiple parameter values and visualizing the output.

3.3.1 Editing Network and Parameter Values

The editing network and parameter value section five sections.

3.3.1.1 Initial Condition

The initial condition settings panel (Figure 3.3) allows for the user to edit the initial starting values of the agents. Each agent type has their own table containing the initial population count. Data that can be represented as a vector, for example the data attributed to an agent type have their own section, Figure 3.4. Data that is stored as a matrix, the data stored on edges between agents, is stored in the matrix tab Figure 3.5. The environment data and settings data also have their own tab, Figure 3.6 and Figure 3.7 respectively.

Graphing Data (Initial Conditions)	Non Graphing Data (Parameter Values): Vectors	Non Graphing Data (Parameter Values): Matrices	Environment Parameters	Settings
DataTable for Resources				
	80 200		R1 250	R2 100
DataTable for Uninfected Bacteria				
	80 40		I0 50	I2 55
DataTable for Infected Bacteria Row names: ['B0', 'B1', 'B2']				
Infected B0		Infected B1	Infected B2	Infected B3
0 0 0		0 0 0	0 0 0	0 0 0
DataTable for Phages				
		P0 10		P1 6

Figure 3.3: The panel where a user can edit the initial conditions of the agents. The columns of each table show for which agent the value corresponds to. A copy of this data is sent to the solver. This instance of the model models a 3 resource, 3 bacteria, and 2 phage system. The bacteria are split into an uninfected and an infected stage. Once infected by a phage, a bacterium will go through four intermediary infection stages before lysing. The infected stages are represented as a matrix, with each row representing a bacterial agent while each column represents the stage of infection.

Graphing Data (Initial Conditions)	Non Graphing Data (Parameter Values): Vectors	Non Graphing Data (Parameter Values): Matrices	Environment Parameters	Settings
DataTable for c_vector				
	80 0.10		R1 0.22	R2 0.15
DataTable for tau_vector				
	80 0.7		I0 0.92	I2 3.2

Figure 3.4: The panel where the user can edit the attributes for a given attribute name. This version is specifically used represent the attributes that are associated with nodes, and are represented as a vector. The columns of each table show for which agent the value corresponds to. A copy of this data is sent to the solver.

Graphing Data (Initial Conditions)	Non Graphing Data (Parameter Values): Vectors	Non Graphing Data (Parameter Values): Matrices	Environment Parameters	Settings
DataTable for v_matrix Row Names: ['B0', 'B1', 'B2']				
	80 1 0 1,1		I0 1,3 1,7 1,4	R2 0 1,43 0,9
DataTable for K_matrix Row Names: ['B0', 'B1', 'B2']				
	80 30 0 4,0		I0 1,1 0,9 2,0	R2 0 0,99 36
DataTable for r_matrix Row Names: ['P0', 'P1']				
	80 0,45 0,41		I0 0,49 0	R2 0,44 0
DataTable for B_matrix Row Names: ['P0', 'P1']				
	80 17 11		I0 14 8	R2 12 0

Figure 3.5: The panel where the user can edit the attributes for a given attribute name. This version is specifically used to represent the attributes that are associated with edges between agents, and are represented as a matrix. The rows and columns of each table show for which agent the value corresponds to. Due to limitations with Dash datatables, the row names can't explicitly be labelled, however text above the table shows the row names. A copy of this data is sent to the solver.

Graphing Data (Initial Conditions)	Non Graphing Data (Parameter Values): Vectors	Non Graphing Data (Parameter Values): Matrices	Environment Parameters	Settings
DataTable for Environment Parameters				
Note: Some parameters won't influence the simulation. For example, changing M won't affect the number of steps in the lysis process, but overall should have an immediate effect on the simulation.	Temperature 25	pH 7	M 4	washout 0

Figure 3.6: The panel where a user can edit the edge attributes the environment settings. Each column represents a single environment parameter. A copy of this data is sent to the solver.

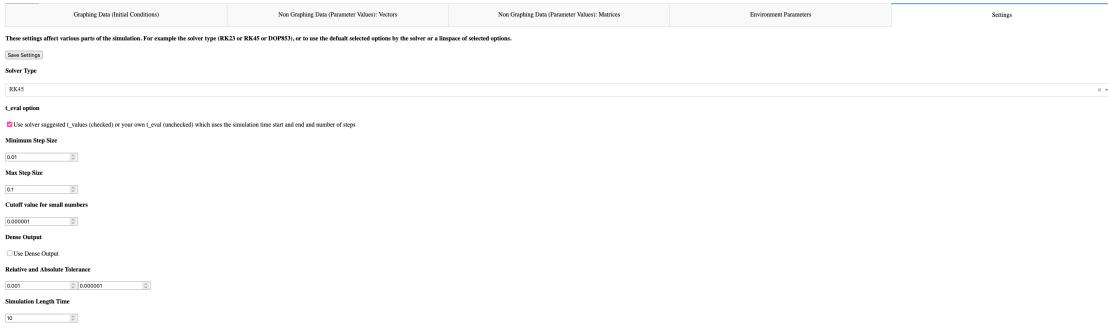


Figure 3.7: The panel where a user can edit the settings of the solver. Various options exist, such as solver type, cutoff value for small values, or tolerances of the solver. A copy of this data is not sent to the solver. The user needs to save the settings, while for the other panels the user does not need to save the changes.

3.3.2 Advanced Visualization and Analysis

In the advanced analysis section, the user can run different analysis methods to gain a greater understanding of the model. The visualizations only support a $1 \times 1 \times 1$ model, in order to make the analysis easier for the user, and to make it easier to analyze the visualization. These advanced visualizations were created with the mind of understanding a simple network. There are five different analysis and visualization methods, and one system where the user can run a large simulation on the whole network and receive an output file containing the raw simulation file data. The raw data is stored as a *parquet* file, a tabular-like data format, which when combined with Dask (not Dash), allows for querying of the data similarly to Pandas. Parquet with Dask offers superior performance and data storage solutions that Pandas can't offer. Once queried, the user can create their own graphs and plots as they have access to the parameter values used and the raw simulation data.

3.3.2.1 Serial Transfer

Serial transfer is a method employed by bacteriologist where after a set amount of time, the bacteriologist pipettes a specified amount of media (for example 10ml of liquid) containing bacteria and nutrients, possibly with phages, and transfers the old media into a solution containing new media. At this stage, the bacteriologist can introduce new agents, or re-introduce agents if the agent population or concentration has died out. However, usually only resources are added during the transfer process. An example would be an experiment starts with 50ml of solution. The experiment runs for 24 hours before 5ml is removed. Researchers can run various tests, such as using optical density measurements to assess bacterial density in the solution or employing a mass spectrometer to determine the concentration of the resources. The 5ml is then re-added to a new

solution of 45ml containing fresh resources. The effect that this has is it creates a sort of artificial stable point. As the bacteria grow, they consume the resources found in the solution. However eventually the resources run out, and the bacteria die out due to a lack of nutrients. By introducing new nutrients at set time intervals, the bacteria can regrow and exhibit a semi-stationary behavior.

The implementation of serial transfer is slightly different. A user can select a number which will divide the population count of the agents by that number (Figure 3.8). Then the program takes the initial condition values defined for the nutrients the initial condition in 3.3.1 and adds those values to the nutrients respectively. By selecting a checkbox, the values as defined in the initial condition box for phages and bacteria in 3.3.1 can optionally be added as well. As an example, if at the end of a simulation, there are 120 resources, 5000 bacteria, and 1000 phages remaining and the chosen serial transfer value is 15, then the resource, bacteria, and phage values would be decreased to 8, 333.33, and 66.66 respectively. Then, if the initial condition for the resources, bacteria, and phages in 3.3.1 are 500, 80, and 10 respectively, and the checkbox is unchecked, the new population count will be 508, 333.33 and 66.66 respectively. If the checkbox is checked, the new population count will be 508, 413.33, and 76.66 respectively. These new values would be used as the new starting initial condition for a new simulation, and the run results will be appended to the previous run. As output, new graphs are created showing the runs appended to one another, with an examplee output shown in Figure 3.9.

3.3.2.2 Parameter Analysis

The parameter analysis settings tab as shown in Figure 3.10 allows the user to choose two parameters and individually run the model with the varying input values. The values that can be tested and changed include all initial condition values, vector and matrix data, and environmental data. As input, the user can select 2 parameters of choice. After the parameter name selection, 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 serial transfer, where the serial transfer uses the settings found on the Serial Transfer tab.

Figure 3.11 shows the heatmap that the user can expect, one heatmap for each agent type. Each heatmap has cells that unique model input, and contains the value of A heatmap matrix is created for each agent type, with dimension of the input values. Each box corresponds to each pair of parameter inputs, and shows the population count

Serial Transfer	Parameter Analysis	Initial Value Analysis	Phase Portrait	SOBOL Analysis	Ultimate Analysis
-----------------	--------------------	------------------------	----------------	----------------	-------------------

Note: Using the serial transfer function will take the final iteration values of the current simulation as shown above and divide it by the value shown below. So if the final value for a bacteria is 100 and the serial transfer value is 10, the new simulation will start with a bacteria value of 10. There is a special case for nutrients where the new value is added to the value associated in the Graphing Data (Initial Conditions) section. So for nutrients, if the final value is 100 and the serial transfer value is 10, and the "Initial" Condition is 50, the new simulation will start with a nutrient value of $100/10 + 50 = 60$. If the checkbox is selected, the unique process will also apply to the phages and bacteria. If the checkbox is not selected, the unique process will only apply to the resources/nutrients. Change the value below to 1 if you want to simply add phages/bacteria/resources without removing substances.

Note 2: These settings are also used in Parameter Analysis, Initial Value, and Phase Portrait if the associated checkbox is selected

Serial Transfer Dilution rate:

Option to add phages and uninfected bacteria to serial transfer, uses value in Initial Condition

 Add Phages and Bacteria

Number of times to automatically run serial transfer

Use log y axis (checked) or linear y axis (unchecked)

Run Serial Transfer

The plots above in the Line Chart section will update with the new values after the serial transfer is complete, ensure that a model has already been run before running the serial transfer. To reset the chart, run the 'Rerun model' button above.

Figure 3.8: The section where the user can set up the serial transfer. To adjust the values added to the agent populations, the user would need to edit the initial condition values.

of the agent at that time. The value inside the cell can be changed by sliding the slider. Note that the heatmap color range resets for each heatmap, so similar colors across heatmaps will not correspond to the same values.

3.3.2.3 Initial Value Analysis

The initial value analysis settings tab as shown in Figure 3.12 allows for the user to choose a single parameter and vary the value of that parameter. This allows for the user to quickly iterate over a range of values for a parameter and visualize how the values affect the population count of the agents.

Figure 3.13 shows the plots that the user receives. For each agent type, there are three plots made. 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 of the peak. This value is considered the time of peak, and is used to fix some issues that can arise where the population plateaus or only keeps on rising. The initial value is plotted on the x-axis, with the time at which the max value is reached on the y-axis. Using the plotted data, a linear or log fit can be created. Using this data can be useful for understanding how a change in parameter value affects the time at which the population count reaches a maximum. The slope, intercept and R^2 value is stored and saved in the third plot,

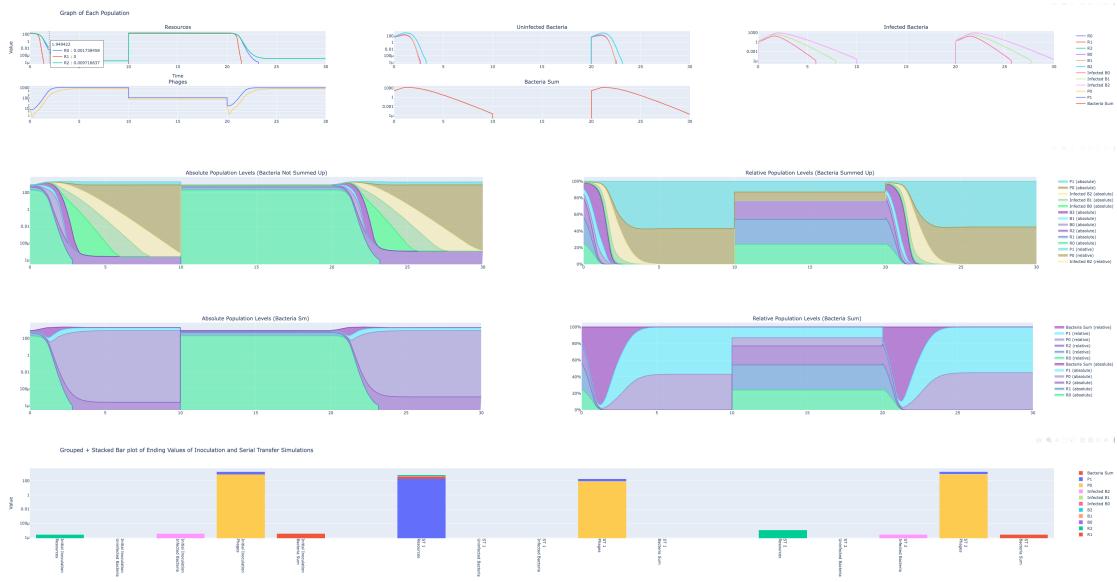


Figure 3.9: Running the serial transfer updates the plot at the top of the page. The simulation initially ran for $t = 10$, and completed two serial transfers with a dilution rate of 10. The first transfer was completed without adding new phages and bacteria. The second transfer was done with adding new phages and bacteria. The R1 (red) and R0 (purple) resource died out at $t = 1.5$ and $t = 2.8$ but are reintroduced at $t = 10$ as part of the serial transfer process. All bacteria become infected and die out before or at $t = 10$. At $t = 10$ the phages have a value of roughly 818 and 1078 pre-serial transfer, and have a value of 81.8 and 107.8 post serial transfer. At $t = 20$, a second serial transfer occurs where the bacteria were reintroduced to the system, and more phages were added to the existing population. The stacked line plots show the absolute and relative distribution of different population groups. The stacked bar chart at the bottom shows the final population count of each agent group type at the end of each serial transfer. It might not always be feasible to experimentally determine the population count at each timestep in a lab, hence this graph can be used as a replacement to show the population count at serial transfer time.

a bar chart, with an editable name. For every re-run of the initial value analysis, the slope, intercept and R^2 value is stored in the bar chart, allowing comparison of the slope-intercept data.

3.3.2.4 Parameter Space Slot

The parameter settings panel as to be seen in Figure 3.14 allows for the

3.3.2.5 SOBOL Analysis

SOBOL analysis, a variance-based sensitivity analysis, is a method that allows a user to quantify how important an input parameter has on a measured aspect of the output by changing the parameter values of the model and measuring the change in model output. SOBOL quantifies how much variance in the output can be attributed to a specific

Serial Transfer	Parameter Analysis	Initial Value Analysis	Phase Portrait	SOBOL Analysis	Ultimate Analysis
-----------------	--------------------	------------------------	----------------	----------------	-------------------

Note: Choose 2 parameters of choice. Input the values you want to test separated by commas. The program will run the simulation for each combination of the two parameters and display the results in a heatmap. The heatmap represents the end value of the simulation for each combination of the two parameters. Make sure you choose an appropriate range of values to test and end simulation lenght before everything drops to 0!

Choose two parameters to analyze

Resources	X ▾
Uninfected Bacteria	X ▾

Option 1: Input the values you want to test separated by commas

0.01, 0.1, 1, 5, 50 | 0.02, 0.2, 2, 6, 10, 20, 50

Option 2: Choose a start value and end value for each parameter separated by a '-' sign

0.01-0.8 | 0.01-0.8

And choose the values to test between the two values for uniform spaced intervals (including the end values)

10 | ▲ | 10 | ▼

- Checked for running Option 1, unchecked for Option 2
- Use Serial Transfer
- Extrapolate value in case slider value is in between 2 calculated time intervals

Run Parameter Analysis



Figure 3.10: The user can choose two parameters to run, along with the values they want to test. There is an option for a serial transfer, along with interpolating the values between the slider option.

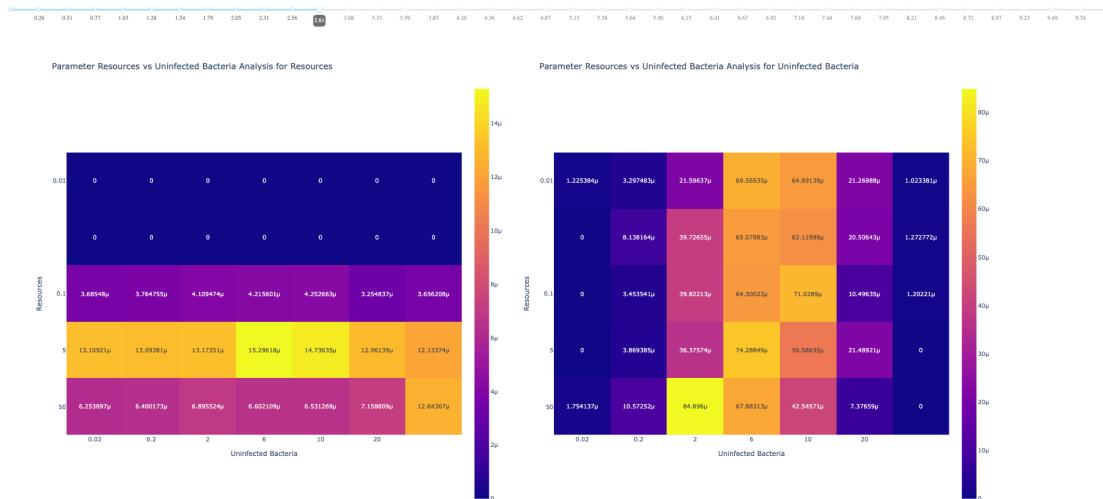


Figure 3.11: An example run of the parameter analysis. The point in time to be shown in the heatmap can be changed by moving the slider above the heatmaps. This in turn affects the value shown in the heatmap.

Serial Transfer	Parameter Analysis	Initial Value Analysis	Phase Portrait	SOBOL Analysis	Ultimate Analysis
-----------------	--------------------	------------------------	----------------	----------------	-------------------

Note: Choose a parameter of choice. Input the values you want to test separated by commas, or use a uniform separated list. The program will run the simulation for each initial value and display the results on a graph.

Resources

Option 1: Input the values you want to test separated by commas

Option 2: Choose a start value and end value for each parameter separated by a '-' sign

And choose the number of uniformly spaced values to test (including the end values)

Checked for running Option 1, unchecked for Option 2
 Use Serial Transfer

Name for the run (optional, default = 'Run [number]'):

Choose a scale for the graph (linear or log graph)

log-linear (log)

0.95

Log y axis IVA plot
 Run Initial Value Analysis

Clear Bar Chart

Clear Bar Chart

Figure 3.12: A unique setting featured in the initial value analysis is the "run name" and the "percentage from the max value reached".

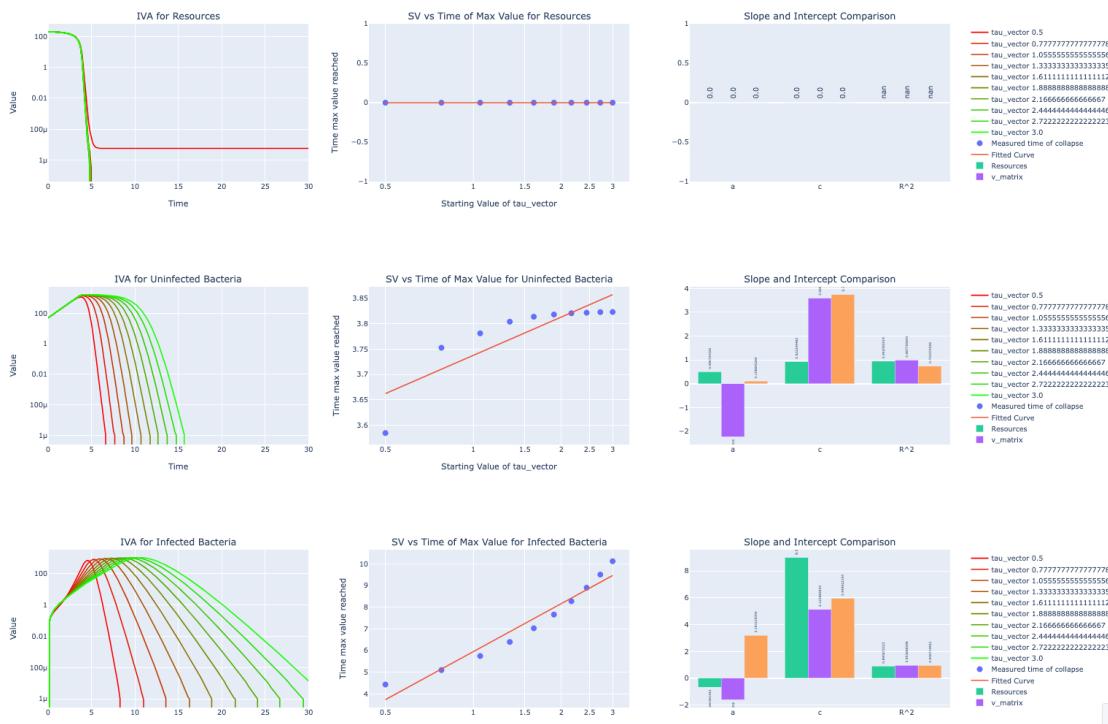


Figure 3.13

Serial Transfer	Parameter Analysis	Initial Value Analysis	Phase Portrait	SOBOL Analysis	Ultimate Analysis
-----------------	--------------------	------------------------	----------------	----------------	-------------------

Note: Choose 2 parameters of choice. The program will run a simulation and plot a phase portrait of the two parameters. The phase portrait will show the relationship between the two parameters over time.

Resources	x ▾
Uninfected Bacteria	x ▾

Option 1: Input the values you want to test separated by commas

48, 49, 50	10, 50, 100
------------	-------------

Option 2: Choose a start value and end value for each parameter separated by a '-' sign

Parameter 1:

48-50	15
-------	----

Parameter 2:

0.01-60	15
---------	----

Use option 1 (checked) or option 2 (unchecked)
 Log x graph
 Log y graph
 Use Serial Transfer

Figure 3.14: The user can select two starting values for the initial condition, but they can't choose vector, matrix, or environment settings due to the plot showing the development of agent populations against other agent populations. As typical, the user can select their own values or auto-generate values between two values, as well as use a serial transfer option. There is also an option to take the logarithm of the x and/or y axis.

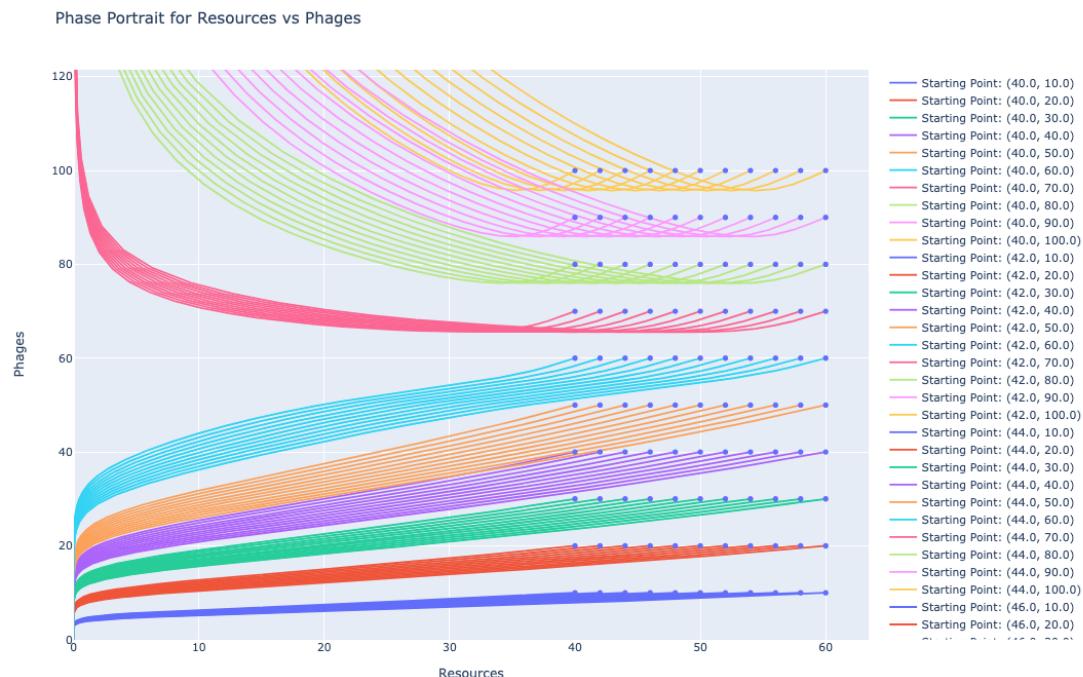


Figure 3.15: An example run of a parameter space plot. Resources are plotted against phages, with a high washout rate of 0.05 explaining why the phage population is decreasing. For initial phage population of 60 and less, the phages “die out” due to the washout rate, while for an initial population count of 70 and higher the phages are able to proliferate. There is an initial population decrease due to the washout, but after the infected bacteria start to die out and release new phages, the population can grow.

parameter and cna measure the effect of global, first, and second order sensitivity. When a model is viewed as a black-box model, the model can be seen as a function $Y = f(X)$, where X is an input vector of d elements, and Y is a univariate model output. X is assumed to be independently and uniformly distributed within a hypercube $X_i \in [0, 1]$ for $i = 1, \dots, d$. The first order sensitivity measures the output variance of the main affect 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 first order sensitivity is described as

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

where $V_i = Var_{X_i}(EX_{\sim i}(Y|X_i))$ and where $X_{\sim i}$ represents all the parameters that are not X_i .

The second order index measures the impact of input X_i interacting with 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, and measures the contribution to the output variance of X_i , including all variance due to X_i 's interaction with other variables.

$$S_{T_i} = \frac{EX_{\sim i}(Var_{X_i}(Y|X_{\sim i}))}{Var(Y)} = 1 - \frac{Var_{X_i}(EX_i(Y|X_{\sim i}))}{Var(Y)}$$

SOBOL can analyze various univariate outputs. This could be either the average value of an agent population, the variance in population count, the time at the peak of an agent count, the final population value, etc.

SOBOL accepts a list of parameter names and a list of range of values to sample from, which the user can input in the SOBOL settings tab Figure 3.16. 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 would need to be run.

If the user wants to analyze the second order interactions, then the model will run the system $N(2D + 2)$ times with the randomly sampled input values, where N is a multiple of 2, and D is the number of parameters being tested. Otherwise, if 2nd order is not chosen, the model is run $N(D + 2)$ times.

Three SOBOL analyses are included by default in the dashboard, as shown in Figure 3.17. An analysis of the final value of the simulation, the average population count, and the variance in population count. The global and first sensitivity are shown next to one another, and each subrow within a plot row represents each agent type.

Serial Transfer	Parameter Analysis	Initial Value Analysis	Phase Portrait	SOBOL Analysis	Ultimate Analysis
-----------------	--------------------	------------------------	----------------	-----------------------	-------------------

Runs a Sobol Analysis. Select the range of values to test for each parameter. The program will run the simulation for a random selection of values for each parameter in the range. The program will then plot the results of the Sobol Analysis. Done in parallel. If you don't want to include a parameter in the stability analysis, don't include the parameter range in the field. You can choose the number of samples to test, if it tests a 2nd order, and the seed value for the random number generator. It uses the sobol sampler and sobol analyzer read more on the SALib Sobol documentation.

Resources

0.01-10

Uninfected Bacteria

0.01-10

Infected Bacteria

0.01-10

-
-
-

washout

0.01-10

Number of samples, 2^x , where x is the number inputted below

2

Number of timesteps

100

Seed value

0

Calculate 2nd order

Run SOBOL Analysis

Figure 3.16: For each parameter, the user can choose a range of values that SOBOL will use to test and analyze. If no values for a parameter are inputted, then the parameter is not included in the analysis. As each run can produce different time data of varying length, the number of timesteps has to be fixed by the user. Due to the randomness of the sampling method, a seed value can be provided to ensure repeatable results.

3.3.2.6 Ultimate Analysis

The Ultimate Analysis section does not produce any visualizations or analysis, but instead allows for the user to define which initial conditions and parameter values they want to run a simulation on. The solver will iterate over every single parameter input possibility and save the results in a *.parquet* file. Similarly to the other sections, the user can specify a start and end value, along with the number of values to generate evenly spaced within that range, including both the start and end values.

Using Dask and the saved *.parquet*, 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 3.17

Serial Transfer	Parameter Analysis	Initial Value Analysis	Phase Portrait	SOBOL Analysis	Ultimate Analysis
-----------------	--------------------	------------------------	----------------	----------------	-------------------

Choose values you want to test for the ultimate analysis. The program runs the simulation for each combination of the parameters (so watch out for exponential explosion!). It overwrites all values in the associated vector/matrix. Then it saves a pickle file with the combinations, and other data, and saves a parquet file with the results of the full simulation (time and y values), without any processing to it. The system periodically updates the parquet file with the results of the simulation to prevent old data from using up ram. Read the documentation on Dask to load the data into your own program for later processing. Partitioning the data allows for faster querying on the data, so select a small subsection of data where you will want to do frequent queries on.

Option 1: Input the values you want to test separated by commas

Resources

Opt 1: your selected values	Opt 2: range of values	Opt 2: number of steps
-----------------------------	------------------------	------------------------

Use Opt 1 or 2
 Include parameter in simulation
 Partition data on this attribute

Uninfected Bacteria

Opt 1: your selected values	Opt 2: range of values	Opt 2: number of steps
-----------------------------	------------------------	------------------------

Use Opt 1 or 2
 Include parameter in simulation
 Partition data on this attribute

Infected Bacteria

Opt 1: your selected values	Opt 2: range of values	Opt 2: number of steps
-----------------------------	------------------------	------------------------

Use Opt 1 or 2
 Include parameter in simulation
 Partition data on this attribute

■
■
■

washout

Opt 1: your selected values	Opt 2: range of values	Opt 2: number of steps
-----------------------------	------------------------	------------------------

Use Opt 1 or 2
 Include parameter in simulation
 Partition data on this attribute

Figure 3.18: The section where a user can start an ultimate analysis, where they provide the values they want to test, and run the simulations. As an output, the user receives a *.parquet* file containing the parameter values that were changed for that specific and the *t* and *y* values for that specific run. A pickle file of a dictionary containing information on that run such as the parameter names and values chosen, the original graph data, and original parameter values.

Chapter 4

Experiments and results

Chapter 5

Discussion

Chapter 6

Conclusion and future work

6.1 Conclusion

6.2 Future Work

Next steps would be to give the model to the lab technicians running 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 review to see if their method is good. They might have accidentally not added enough resources, or accidentally miscalculated the initial concentration of bacteria.

6.2.1 Other Models

“All models are wrong, but some are useful” — George E. P. Box

Each model has its pros and cons. Take the exponential population growth model

$$\frac{dP}{dt} = rP \quad (6.1)$$

$$P(t) = P_0 e^{rt} \quad (6.2)$$

where $P(t)$ is the population at time t , P_0 is the initial population, and r is the growth rate. This model acts as a nice introduction to population modelling. It can accurately fit the exponential growth bacteria experience in a petri dish. However, this basic model does not account for a spatial and resource consumption. Eventually the bacteria run out of space and resources, and start to die out. A population can not grow exponentially forever, the resources can only support a maximum population, the carrying capacity. The model can be adapted to include a carrying capacity (the max population level that can be reached), where the new updated model is

$$\frac{dP}{dt} = rP\left(1 - \frac{P}{K}\right) \quad (6.3)$$

$$P = \frac{K}{1 + \left(\frac{K-N_0}{N_0}\right)e^{-rt}} \quad (6.4)$$

where K is the carrying capacity. This adapted model, the logistic growth model better accounts for the eventual restriction of population growth.

Figure 6.1 shows how the carrying capacity has a large influence on the speed and growth trajectory of a population. The logistic curve initially follows the exponential curve before the maximum growth rate is reached and starts to slow down and taper off as the population asymptotically approaches the carrying capacity $K = 200$.

A further step would be to introduce competition between other bacteria. For example, a $p_{0,1} \cdot P_0 \cdot P_1$ term can be subtracted from the logistic growth curve. This term accounts for competition between Populace 0 and Populace 1, with $p_{0,1}$ being the interaction factor between P_0 and P_1 . Assuming P_0 is being looked at and P_0 has a high value, if P_1 is high, then a lot of P_0 is going to die out due to the competition with P_1 . If P_1 has a low population, then not many P_0 are going to die out due to less competition with P_1 .

The model can be further extended by accounting for temperature, pH, more interactions between other agents, thermal constant addition and removal of other agents, and other considerations.

6.2.1.1 Spatial simulations

<https://www.sciencedirect.com/science/article/abs/pii/S0022519318305368> The ODE models work very nicely when there is no consideration for space and 2D/3D-space dimensionality. Spatial models complicate the simulation, making it harder to analyze. Data collection and analysis becomes harder. Unique and novel analysis and visualization methods have to be created to be able to represent and visualize the data through space and time.

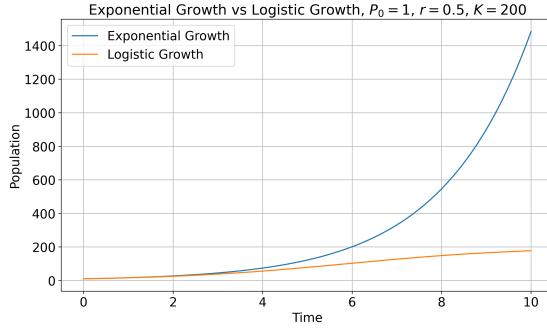


Figure 6.1: Exponential growth curve vs logistic growth

PDE PDE are the next logical step to add space to an ODE model. The general formula, as given by

$$\frac{\partial u}{\partial t} = D \nabla^2 u + f(u, x, y, \dots, t)$$

where $u(x, y, \dots, t)$ is the population density of interest, D is the diffusion constant, ∇^2 is the derivative of each spatial direction, and $f(x, y, \dots, t)$ is the function encapsulating growth, death, and interactions dynamics.

Discretization The dimensions can be discretized into boxes of dimensions $\delta x, \delta y, \dots$. This transforms the PDE into a system of difference equations, which can be solved numerically. For example, the Laplacian term $\nabla^2 u$ in 2D can be approximated using finite differences as:

$$\nabla^2 u \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\delta y^2}$$

where $u_{i,j}$ represents the value of u at the grid point (i, j) . This discretization allows the PDE to be solved iteratively over a grid, enabling spatial simulations of population dynamics. Each box can be represented by a matrix, and the population value can be displayed as a heatmap using visualization software.

Chapter 7

Ethics and Data Management

A new requirement for the thesis is that there must be a short section in which you reflect on the ethical aspects of your project. This requirement is related to one of the final objectives that a graduated student of the Master of Computational Science must meet: “The graduate of the program has insight into the social significance of Computational Science and the responsibilities of experts in this field within science and in society”. You don’t need to devote an entire chapter to this; a short section or paragraph is sufficient.

I acknowledge that the thesis adheres to the ethical code (<https://student.uva.nl/en/topics/ethics-in-research>) and research data management policies (<https://rdm.uva.nl/en>) of UvA and IvlI.

The following table lists the data used in this thesis (including source codes). I confirm that the list is complete and the listed data are sufficient to reproduce the results of the thesis. If a prohibitive non-disclosure agreement is in effect at the time of submission ”NDA” is written under ”Availability” and ”License” for the concerned data items.

Short description (max. 10 words)	Availability (e.g., URL, DOI)	License (e.g., MIT, GPL, Creative Commons)
Example dataset 1	<github url>or Figshare	GPL
Example source code	DOI (from Zenodo)	MIT
Example sensitive data	NDA	NDA

Chapter 8

Appendix A: Equation Parameters

Parameters used in equations.

Parameter	Parameter Full Name	Description	Default Value	Alternatives	Notes
P	Phage Parameter	Phage population count			
U	Uninfected Parameter	Uninfected bacteria population count			
I	Infected Parameter	Infected bacteria population count			
R	Resource Parameter	Resource concentration			
B	Bacteria Parameter	Bacteria population			Some n
ω	Washout Rate	Rate of parameter washing or flowing out of the system			
β	Burst Size	Number of phages created when bacteria cell bursts			
t	Time	Time step during simulation			
μ	Mean	Mean			
σ	Standard Deviation	Standard deviation			
T_{min}	Minimum Temperature	Minimum operating temperature for a microbe			
T_{opt}	Optimal Temperature	Optimal operating temperature for a microbe			
T_{max}	Maximum Temepra-ture	Maximum operating temperature for a microbe			
pH_{min}	Minimum pH	Minimum operating pH for a microbe			
pH_{opt}	Optimal pH	Optimal operating pH for a microbe			
pH_{max}	Maximum pH	Maximum operating pH for a microbe			

Chapter 9

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.

9.1 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 agents such as bleach is not desirable due to leaving chemicals on the food, which can be fatal if ingested. Physical agents like heat or radiation can kill bacteria, but at the cost of altering the food quality [25].

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.

9.1.1 Current Applications

Phage cocktails like SalmoFresh™ have been proven to safely reduce *Salmonella* contamination in pet food and raw pet food ingredients [2], as well as in romaine lettuce and bean sprouts [3]. 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 [5]. Figure 9.1 [2] and Figure 9.2 [3] show how applications of phages have reduced the count of *Salmonella* in ingredients used in pet food as well as romaine lettuce and bean sprouts. In Figure 9.1, 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 9.2, 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.

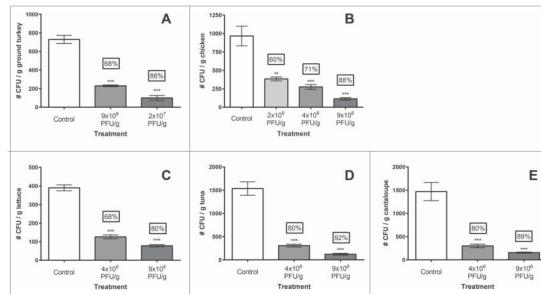


Figure 9.1: SalmoLyse® reduces *Salmonella* contamination on various food surfaces: Mean and standard error bars shown. Statistical analyses were carried out for each food group independently. Asterisks denote significant reduction from corresponding controls based on one-way ANOVA with Tukey's post-hoc 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 [2].

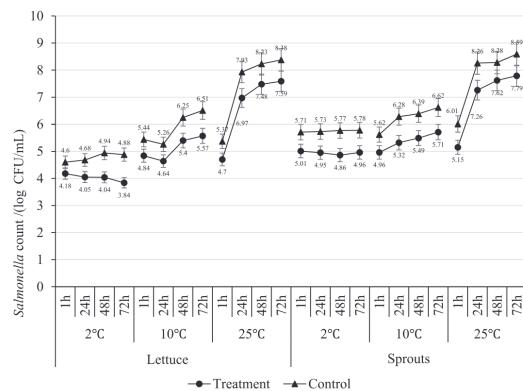


Figure 9.2: *Salmonella* count in a mixture of 5 *Salmonella* strains spot-inoculated (CFU/g) onto a) lettuce and b) sprouts after spraying with a mixture of bacteriophage (SalmoFresh™) relative to positive controls at 2, 10 and 25°C and stored for 1, 24, 48 and 72 h. [3]

9.2 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) [26]. 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.

9.2.1 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 [27], by disrupting the structural integrity of the cell wall [28], or by inhibiting protein synthesis [29].

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 [30]. 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 [31]. 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 [32].

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.

Nana Nguefang Laure et al. 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 [33].

Yuanyang Zhao et al. 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 re-introduce antibiotic susceptibility to previous insusceptible bacteria, preventing costly and lengthy research in new antibiotics [34].

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 [34].

9.3 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 nutrients like nitrogen and phosphorus, can occur in freshwater, coastal, and marine environment.

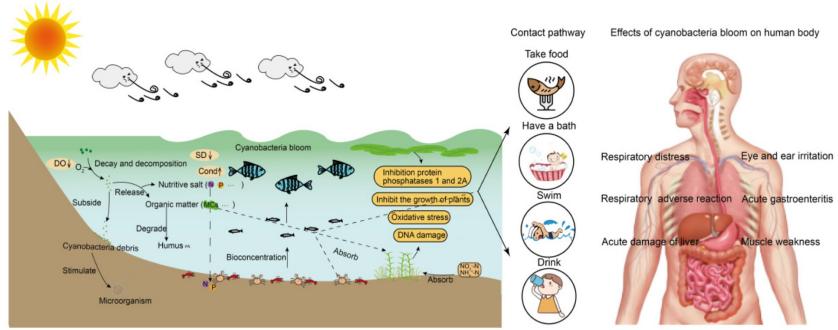


Figure 9.3: Cyanobacteria degradation cycle, main hazards of cyanobacteria bloom to water bodies, aquatic organisms, and the human body. (DO: dissolved oxygen; SD: water transparency; Cond: conductivity; N: nitrogen; P: phosphorus; MCs: microcystins).

[4]

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 with oxygen, outpacing other aquatic growth, and killing aquatic marine life. Toxins can make their way into the food and water consumed by humans, causing muscle fatigue, respiratory issues, liver damage, and gastrointestinal issues [4]. Figure 9.3 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.

9.3.1 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 [35].

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 nutrients like nitrogen. This allows for other bacterial strains to absorb the nutrients and dominate the bacterial community. Phages and the lysis of bacterial strains can have a dramatic effect on community dynamics and composition of other agents like phages, bacteria, and resources [36]. 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 [37?].

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[38]. 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 [39, 40], with a respiratory illness visit costing between \$0.5 and \$4 million [41].

Bibliography

- [1] Sanju Tamang. Horizontal Gene Transfer in Prokaryotes and Eukaryotes. <https://microbenotes.com/horizontal-gene-transfer-prokaryotes-eukaryotes/>, September 2023.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] Meyers Robert A. Encyclopedia of Physical Science and Technology. <http://www.sciencedirect.com:5070/referencework/9780122274107/encyclopedia-of-physical-science-and-technology>.

- [8] Claudia Igler. Phenotypic flux: The role of physiology in explaining the conundrum of bacterial persistence amid phage attack. *Virus Evolution*, 8(2):veac086, July 2022. ISSN 2057-1577. doi: 10.1093/ve/veac086.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] S. Moineau. Bacteriophage. In Stanley Maloy and Kelly Hughes, editors, *Brenner's Encyclopedia of Genetics (Second Edition)*, pages 280–283. Academic Press, San Diego, January 2013. ISBN 978-0-08-096156-9. doi: 10.1016/B978-0-12-374984-0.00131-5.
- [16] 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.
- [17] J. J. Bull. Optimality models of phage life history and parallels in disease evolution. *Journal of Theoretical Biology*, 241(4):928–938, August 2006. ISSN 0022-5193. doi: 10.1016/j.jtbi.2006.01.027.

- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] Advanced Wastewater Modelling | GPS-X - Hydromantis. <https://www.hydromantis.com/GPSX-innovative.html>.
- [25] 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.
- [26] Stephen Odonkor and Kennedy Addo. Bacteria Resistance to Antibiotics: Recent Trends and Challenges. *International Journal of Biological & Medical Research*, pages 1204–1210, January 2011.
- [27] 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.

- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] Global action plan on antimicrobial resistance. <https://www.who.int/publications/i/item/9789241509763>.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] 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.
- [37] Christopher R. Grasso, Kaytee L. Pokrzewski, 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.
- [38] (PDF) Economic Impacts of Red Tide Events on Restaurant Sales. https://www.researchgate.net/publication/23515658_Economic_Impacts_of_Red_Tide_Events_on_R
- [39] 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 Breve-toxins. *Environmental Health Perspectives*, 113(5):638–643, May 2005. ISSN 0091-6765. doi: 10.1289/ehp.7496.
- [40] 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.
- [41] 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.