Mathematically Modelling Microbial Interactions Between *Staphylococcus aureus* and *Pseudomonas aeruginosa*

MTHM005: Mathematical Sciences Project

Student: Connor Peter John Tynan
Supervisor: Prof Krasimira Tsaneva-Atanasova

Abstract

Increasingly quantitative descriptions of systems and processes in the life sciences require highly innovative mathematical tools and techniques in order to successfully address biological and biomedical challenges. The aim of this project is to apply mathematical tools and techniques to investigate interactions between *Staphylococcus aureus* and *Pseudomonas aeruginosa*, two important pathogenic bacteria with implications in antimicrobial resistance and cystic fibrosis to name a couple of areas. We used the generalised Lotka-Volterra equations to explain the change in dynamics between novel mono and co-culture data. We found a transcritical bifurcation to underpin the switch between a state of species co-existence and a state in which one species out-competes the other. We infer biological meaning from our results and discussed any relevant literature.

Acknowledgements

I would like to acknowledge Jayanth Kumar Narayana for fruitful conversation about microbial interactions and providing some references regarding use of Stan, and Dr Stefano Pagliara for providing the novel mono and co-culture experimental data. I would also like to thank Prof Krasimira Tsaneva-Atanasova for her continuous support over the duration of this project.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background and Motivation

Microorganisms, or microbes for short, are organisms of microscopic size (e.g., bacteria) associated with all life forms and can be found in almost all of Earth's environments. In addition to their ubiquitous nature, microbes play a key role in the Earth's ecosystems, natural and engineered for that matter [1]. When groups of microbes occupy a common living space they are called a 'microbial community'. The term 'microbiome' refers to a characteristic microbial community occupying a well-defined environment, such as the human gut microbiome [2]. In recent history, the human microbiome has been the subject of increasing focus because of its close association with health and disease. Whilst many microbes and microbial communities bring about beneficial contributions to human health [3], some can cause disease and harm to humans and animals. These harmful microbes are known as 'pathogens' and eliminating these microbes has been the primary focus of antimicrobial research. Antimicrobials are agents that act to kill, or prevent the growth of, microorganisms with the aim of preventing disease. However, widespread use of antimicrobials has resulted in the problem of antimicrobial resistance (AMR). AMR is best described as a pathogen's ability to resist antimicrobial treatments and has become one of the largest public health threats facing humanity to date, with a current estimated death toll of 5 million annually [4] and a predicted death toll of 10 million per year by 2050 [5]. With the worsening consequences of AMR, there is plenty of motivation to understand the processes by which AMR arises and to develop methods to combat such.

Staphylococcus aureus (*S. aureus*) and *Pseudomonas aeruginosa* (*P. aeruginosa*) are two examples of important pathogens which are known to cause disease in humans. Being opportunistic pathogens, they cause infection under conditions not usually met, such as the host having a weakened immune system or having had alterations occur in their microbiomes. Both species can be commonly found on the skin of healthy people [6] and are notorious for their ability to resist a range of antimicrobial treatments [7,8]. For example, methicillin-resistant *S. aureus* (MRSA) is a distinct strain of *S. aureus* that has been coined the 'MRSA Super Bug' for its extraordinary ability to resist a great range of antimicrobial solutions. Similarly, multidrug-resistant (MDR) strains of *P. aeruginosa* have been identified and are of particular concern due to *P. aeruginosa's* remarkable capacity for spreading AMR within living organisms (*in vivo*) [9].

S. aureus is a prevalent pathogen, existing in approximately 20% of the human population always and transiently/intermittently in another 60% [10]. The pathogen is most commonly found on human skin and within the nose, but well-adapted strains can also be found surviving in health-care settings, like hospitals where antimicrobials/antibiotics are most typically employed. *S. aureus* can cause infection by colonising the human skin or mucosal surface it is located on [11] and then invading the host's body via routes such as implanted medical devices (e.g., catheters) or damage to the skin barrier (e.g., cut wounds, burn wounds) [12]. The effects of host invasion (entering of the bloodstream and/or internal tissues) can range from mild skin and soft tissue infections to life threatening conditions such as infective endocarditis [13,14]. Typical human immune response to *S. aureus* invasion is hindered by its armamentarium of immune evasion and virulence factors [14]. These factors and AMR attained by various means make *S. aureus* and resultant infections complicated to treat.

P. aeruginosa, similar to *S. aureus*, has become an increasingly prevalent pathogen in recent times. Most commonly found in moist environments, *P. aeruginosa* can be transferred to individuals via shared spaces, like swimming pools or hot tubs, and equipment, such as taps and soap dispensers [15]. Being the primary cause of death in cystic fibrosis patients and largely associated with infections acquired in health-care settings, solutions to *P. aeruginosa* are actively being sought out [16]. Devising treatments for *P. aeruginosa* is

a task made increasingly complex for reasons similar to the case of *S. aureus*. *P. aeruginosa* is well-adapted to avoid stresses from the host's immune response and prescribed antimicrobials; strains found within health-care settings are typically the most resistant (MDR strains) and feature the fewest available treatment options. Currently, novel therapeutic strategies are concerned with disarming or circumventing *P. aeruginosa's* developed resistance. Some examples of these strategies include inhibition of quorum sensing (a method of communication between bacteria cells) and phage therapy (combating undesired bacteria using bacteria-specific viruses) [8,17,18].

Whilst there is a fair amount of research concerning either *S. aureus* or *P. aeruginosa*, studies investigating both of the pathogens are rarer. This lack of literature may be explained by the recency of research into microbial interactions, which has gained traction within the last 30 or so years [19]. Interactions between microbes occur via various mechanisms and play an essential role in establishing and maintaining single and multi-species populations [20]. Quorum sensing is an example of a microbial interaction and features the production and release of signalling molecules, enabling communication and coordinated response between bacteria cells [17,20,21]. As previously mentioned, infections can arise from the formation and invasion of pathogenic colonies. Furthermore, it has been previously demonstrated that AMR can be transferred between microbes via a mechanism known as 'horizontal gene transfer' (HGT) [22,23]. Additionally, there is an increasing amount of evidence to suggest that intraspecies interactions (interactions between one microbial species) and interspecies interactions (interactions between multiple microbial species) affect microbial response to antimicrobial agents [23]. Given the importance of microbial interactions in establishing pathogenics colonies and its relations to AMR, it is not surprising that a great deal of microbial interaction research is concerned with human health and the human microbiome. Furthering the current understanding of interaction mechanisms is essential to developing novel and improved ways of disrupting microbial communication and fighting AMR.

The co-inhabitance of *S. aureus* and *P. aeruginosa* has long been observed in many kinds of polymicrobial infection, including cystic fibrosis (CF), where *S. aureus* and *P. aeruginosa* are the two most prevalent organisms [24-28]. Because of this, much of the research conducted into *S. aureus* and *P. aeruginosa* interactions is in the context of CF. Initially, *S. aureus* is the most prevalent pathogen within younger CF patients, with almost 80% of patients having some trace of *S. aureus* in their airways before adulthood [25,26]. Colonisation of *P. aeruginosa* typically follows during a patient's mid-to-late teen years and usually dominates, becoming the most prevalent pathogen in a patient's system. It has been recorded that around 60-75% of adult CF patients are persistently infected by *P. aeruginosa* [25-27]. In 2016, Limoli and colleagues reported that co-infection of *S. aureus* and *P. aeruginosa* can be observed in $\sim$20% of CF patients and 46% of patients suffering with CF-related diabetes [28]. They found this co-infection to be associated with reduced lung function and increased frequency of pulmonary exacerbation, suggesting a synergy between the two pathogens [28,29]. This synergism and its enhanced symptoms in CF patients is an additional motivator for understanding the mechanisms and fundamental processes underlying *S. aureus* and *P. aeruginosa* interactions.

The relationship between *S. aureus* and *P. aeruginosa* is complicated, with many published results contradicting each other [30,31]. Whilst the general consensus suggests that, when grown together, *P. aeruginosa* outcompetes *S. aureus* for environmental resources and dominates, some recent publications suggest otherwise [30]. For example, DeLeon et al. (2014) demonstrated how *S. aureus* and *P. aeruginosa* are able to mutualistically co-exist using a chronic wound biofilm model [32]. Furthermore, DeLeon et al. observed an improved ability of both *S. aureus* and *P. aeruginosa* to resist antimicrobial treatments, when grown together in co-culture as a result of the cooperative nature of their co-existence. Yung. et al., (2021) published a comprehensive review discussing the contrasting results surrounding *S. aureus* and *P. aeruginosa* interspecies interactions [31]. They conclude

that the wide-range of possible interactions (i.e., competitive, cooperative, etc) are due to variations in genetic composition and environmental adaptations acquired during infection. Depending on the type of interaction observed, these interactions then go on to influence virulence factors, antimicrobial susceptibility and possibly host immunity, making polymicrobial infections formidable to treat. It is also important to note that certain strains of *S. aureus* and *P. aeruginosa* interacting can result in multiple types of behaviour. For example, Zhao. et al., (2018) was able to capture a transition from competitive behaviour to cooperative behaviour and narrowed this dynamic down to a downregulation in *P. aeruginosa* quorum sensing systems [33].

Overall, interactions between *S. aureus* and *P. aeruginosa* are complicated and remain controversial. Reasons behind such include contrasting results and difficulty growing the two species together in co-culture. Interactions are most commonly competitive or mutualistic; the type of interaction is determined by the genetic background and adaptations acquired by each of the pathogens. There is evidence to support synergies between the two microbes that can negatively affect patient health. This combined with *S. aureus* and *P. aeruginosa* being the two most prevalent species in polymicrobial infections [19] makes understanding interaction mechanisms and dynamics of great importance, as such will provide insight into future effect treatments.

1.2 Mathematically Modelling Microbial Interactions

Having outlined some of the motivations and background information concerning study of the interactions between *S. aureus* and *P. aeruginosa*, we turn out attention to mathematical modelling and touch on how it is being used to understand and decipher microbial interactions.

Mathematics has long been employed to model and understand the world around us. Pierre-Francois Verhulst was first among many to develop a mathematical model for predicting population dynamics. His work would lay down the foundations for mathematicians like Alfred Lotka and Vito Volterra who are best known for proposing the famous Lotka-Volterra model which has since shaped the realm of ecological modelling as we know it. With the ecological modelling scene growing, Jacques Monod (1942) would be the first person to model bacteria growth using a mathematical approach [34,35]. His work would later inspire what has become microbial ecological modelling today, an expansive interdisciplinary domain concerned with understanding microbial communities and ecosystems. In an effort to understand microbial dynamics, mathematical approaches have been used in recent times to model microbial interactions, direct and indirect. We will discuss some existing models and discuss what approaches, if any, have been used to study the interactions between *S. aureus* and *P. aeruginosa* in co-culture.

Population-based modelling approaches are the most common way in which microbial community dynamics have been investigated/simulated. Within these models homogeneity is often assumed. This is done for the sake of maintaining simplicity and emphasising the focus on entire populations of microbes as opposed to individual cells [1]. Interactions between microbial species are quantified based on the overall effect they have on each other. For example, mutualism describes a relationship between microbes in which both species benefit from each other's presence. Table X outlines the different possible interaction dynamics that can occur within microbial communities. With a list of interaction types established, basic microbial interactions can then be investigated between two species by comparing rates of growth in a mono and co-culture settings. The variations in growth rate between the two settings then categorises the type of interaction shared between the two species.

Type	Species A	Species B	Notes
Neutralism	None	None	Rarely observed in natural systems [1]
Mutualism	+	+	Synergism can be considered mutualistic
Competition	-	-	One species usually dominates
Antagonism	+	-	Encompasses parasitism and predation
Commensalism	+	None	Relatively rare compared to other types
Amensalism	-	None	Relatively rare compared to other types

Table 1: Different types of microbial interaction and their respective effects on the engaging microbial species. Interactions between *S. aureus* and *P. aeruginosa* are most commonly categorised as mutualistic or competitive, with *P. aeruginosa* usually out-competing *S. aureus* in the case of competition.

Two ways of mathematically modelling microbial community behaviour are using species-metabolite interaction (SMI) models and species-species interaction (SSI) models. SMI models aim to describe the interactions between species and environmentally available metabolites, signalling molecules which offer a mechanism of communication/interaction between microbes. On the other hand, SSI models quantify direct interactions between microbial species. SSI models may be tailored to study a specific type of community behaviour, such as competition or mutualism. The generalised Lotka-Volttera (gLV) equations are an example of a very general and very popular SSI model that is able to describe a variety of behaviour. In 2019, Brunner and Chia [36] published an article claiming that SMI models were more effective in capturing the behaviour of microbial communities, demonstrating areas in which the gLV model fell down. Brunner and Chia are not alone in their critique of the gLV model either. Momeni, Xie and Shou [37] published an article all about the pitfalls of the gLV method in capturing diverse pairwise microbial interactions. The paper proposes proper use of a mechanistic model, an approach which accounts for the medium in which microbes interact, over traditional pairwise approaches like the pairwise Lotka-Volterra (which is essential the gLV model for 2 dimensions). Mechanistic models are considerably more complicated than pairwise models to construct, but are more rich in their description of microbial interactions.

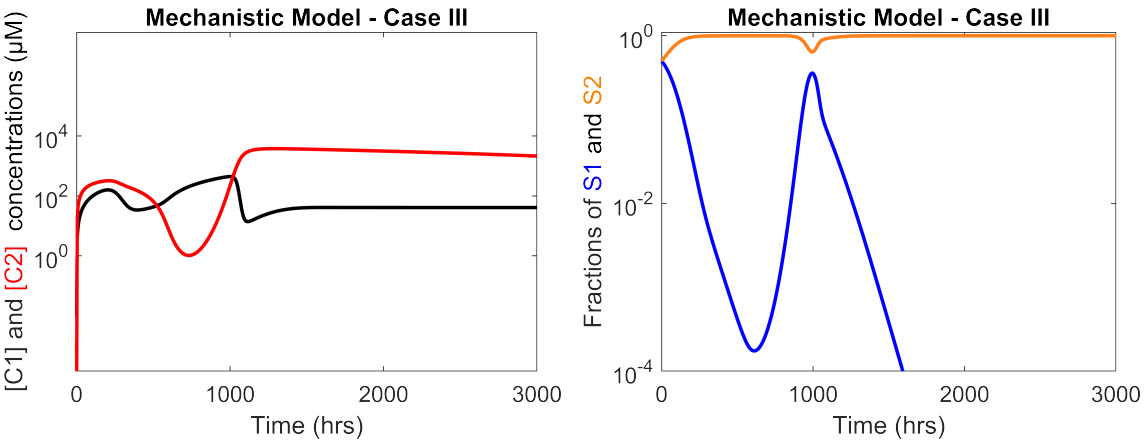

Figure 1: Reproduction of a result from Momeni et al. demonstrating how a mechanistic model can be used to model the population densities of two species (S1 and S2) as well as the chemical concentration of any environmentally available metabolites.

Despite critique of the famous gLV model, its flexible nature and low computational cost

makes it a popular choice for microbial ecology modelling. Take Stein et al. (2013) for example, they introduced an additional perturbation term to the original gLV equations which can be used to simulate the effect of antimicrobial agents, etc [38]. The Stein model has been discussed in the context of *Clostridium difficile*, where it has been able to successfully identify healthy and diseased states based on the model's parameter values. This is a great example of how mathematical models are being used to investigate microbial interactions and combat issues such as antimicrobial resistance and infections.

1.3 Filling The Gap - Aims of This Project

Having searched for relevant literature in the area, there does not appear to exist any literature that describes or explains the interactions between *S. aureus* and *P. aeruginosa* in co-culture using a mathematical model. This is where the novelty of this report lies. Whilst there are several publications that investigate the interactions between these two species using biological models, no paper has specifically considered the insights that mathematical models might provide into the processes underlying *S. aureus* and *P. aeruginosa* interactions. The aim of this project is therefore this, to explain novel *S. aureus* and *P. aeruginosa* mono and co-culture data using mathematical methods and approaches. This involves mathematically explaining the transition in dynamics from a mono-culture setting to a co-culture setting and inferring biological meaning from such.

2 Methodology

2.1 Preprocessing The Data

For this project we will be working with 4 sets of novel data:

1. *S. aureus* Mono-Culture Data
2. *S. aureus* Co-Culture Data
3. *P. aeruginosa* Mono-Culture Data
4. *P. aeruginosa* Co-Culture Data

'Mono-culture' refers to the pathogen being grown in absence of other microbial species, whereas 'co-culture' refers to both species being grown in proximity of each other, enabling for interspecies interactivity. Each set of data features 6 replicates (unique experiments) and each replicate includes 14 discrete hourly recordings of bacteria count starting from $t = 0$ hours ($0 \leq t \leq 13$, and $t \in \mathbb{Z}$). Summaries of the experimental data can be found in the appendix.

2.1.1 Averaging

To obtain a general idea of how each species evolves in mono and co-culture settings we can average across the 6 replicates for each time point.

$$y_{i,Mean}(t) = \frac{y_{i,R1}(t) + y_{i,R2}(t) + y_{i,R3}(t) + y_{1,R4}(t) + y_{i,R5}(t) + y_{i,R6}(t)}{6}, \tag{1}$$

where $y_i(t)$ is the bacteria count for species i at time t and $_{R1-R6}$ refers to the individual replicates, 1 through 6. The mean-averaged results for each set of experimental data are summarised in table 10/11 of the appendix and visualised in Figure 2 . Averaging the data also helps us leverage accuracy and computational cost as computing parameter estimates for each replicate individually will not only take considerably longer, but will also be subject to missing values, hindering the accuracy.

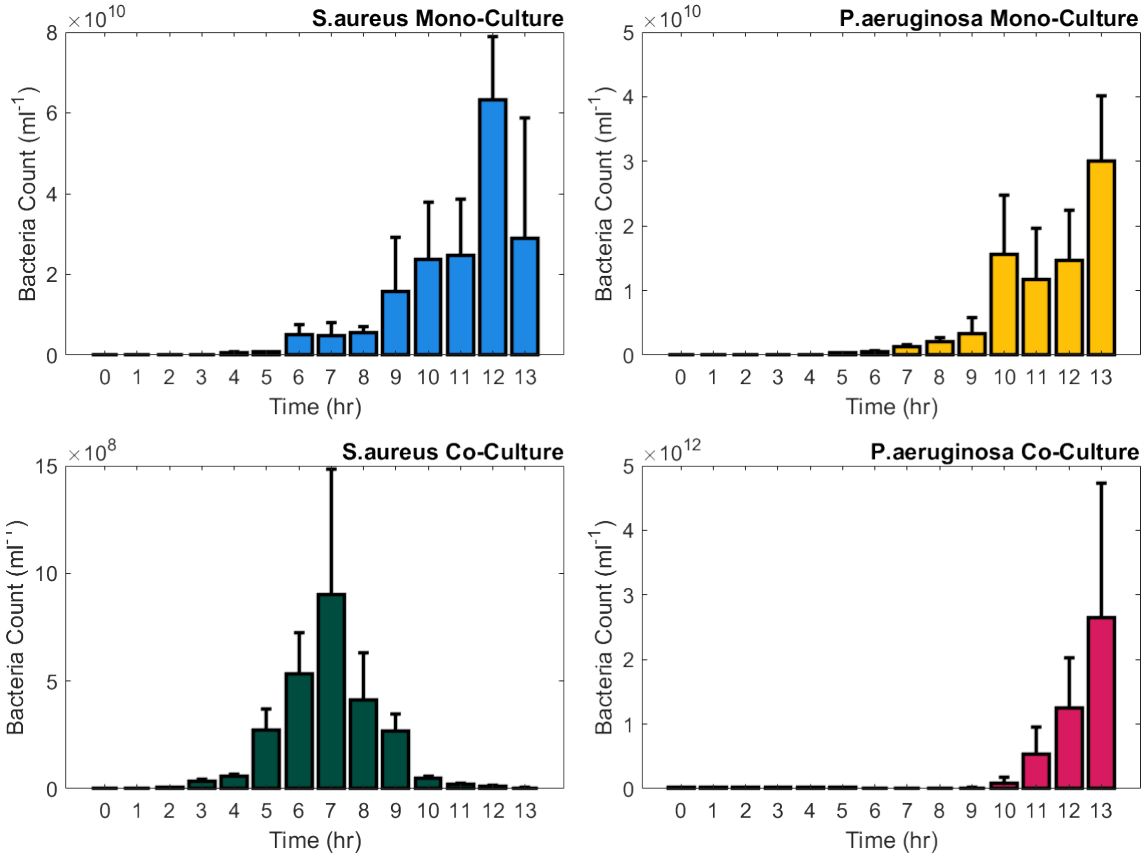

Figure 2: Mean-averaged experimental data. Mean values were calculated across all replicates at each time point $0 \leq t \leq 13$, $t \in \mathbb{Z}$. Missing values (N/A) were accounted for and omitted.

2.1.2 Normalisation

In addition to averaging, we employ normalisation to further organise the experimental data for future use. By dividing each mean-averaged data set through by its initial value we achieve a notionally common scale among the data, making the trends and scale of the results more interpretable.

$$y_{i,Norm}(t) = \frac{y_{i,Mean}(t)}{y_{i,Mean}(1)} \tag{2}$$

This method does not distort the essence of the data but instead makes it easier to work with, in particular the normalisation will come in handy during parameter estimation. The normalised mean-averaged data assumes the same patterns as observed in Figure 2, but for massively reduced values. This data is available in the appendix (Table 11).

2.2 The generalised Lotka-Volterra Model

The model of choice for this investigation is the generalised Lotka-Volterra (gLV) model. The gLV equations extend on the classical Lotka-Volttera equations (predator-prey and competition types) enabling for more wide-spread modelling of dynamics between an arbitrary number of species. Direct interactions on species growth are encoded through interaction coefficients. Whilst the model fails to describe indirect interactions, it provides a fantastic foundation for more advanced community modelling and understanding interactions in the basic sense. To an extent, the equations can be analysed analytically. This will prove useful down the line for steady-state analysis.

Assuming closed populations with only two-way interactions between microbes, we can model the change in bacteria count over time using the gLV equations [39]:

$$\frac{d}{dt}(y_i(t)) = y_i(t)\left(\mu_i + \sum_{j=1}^{n} M_{ij}y_j(t)\right), \tag{3}$$

where n is the number of microbes, $y_i(t)$, for $i \in \{1, \ldots, n\}$, is the population size of species i at time t, μ_i is the growth rate of species i and M_{ij} is the interactivity parameter describing the effect species j has on species i. Given that we have two species, set $n = 2$ and therefore $i \in \{1, 2\}$, hence we get a system of 2 ODEs,

$$\frac{d}{dt}(y_1(t)) = y_1(t)\left(\mu_1 + \sum_{j=1}^{2} M_{1j}y_j(t)\right), \tag{4}$$

$$\frac{d}{dt}(y_2(t)) = y_2(t)\left(\mu_2 + \sum_{j=1}^{2} M_{2j}y_j(t)\right), \tag{5}$$

where $i = 1$ and $i = 2$ represent the species *S. aureus* and *P. aeruginosa*, respectively. Expanding the summation notation and writing $y_i(t)$ as y_i we get,

$$f(y_1, y_2) = \frac{d}{dt}(y_1) = y_1\left(\mu_1 + M_{11}y_1 + M_{12}y_2\right), \tag{6}$$

$$g(y_1, y_2) = \frac{d}{dt}(y_2) = y_2\left(\mu_2 + M_{22}y_2 + M_{21}y_1\right). \tag{7}$$

Note that in the case of mono-culture growth interspecies interactions cannot occur and therefor $M_{12} = M_{21} = 0$. In this case the gLV equations can be expressed as

$$\left.\begin{aligned} f(y_1, y_2) = \frac{d}{dt}(y_1) = y_1\left(\mu_1 + M_{11}y_1\right) \\ g(y_1, y_2) = \frac{d}{dt}(y_2) = y_2\left(\mu_2 + M_{22}y_2\right) \end{aligned}\right\} \text{ Mono-culture equations} \tag{8}$$

The mono-culture equations now form a linear system which is fitting considering the species cannot interact.

2.2.1 Reparameterisation

As it stands, given that the interaction terms are coefficients of y_1y_2, y_1^2 and y_2^2, and noting the large values of bacteria count in the averaged and normalised data (values can be as high as $\sim 1\times10^6\text{ml}^{-1}$), we can expect the parameter estimate values for the interactivity parameters to be very small ($M_{ii}, M_{ij} << 1$). Hence, we reparameterise our gLV equations to make future estimates more interpretable. One way of reparameterising the equations is through converting to a logistic-like form, which can be achieved by first taking out a factor of the growth rate and reparameterising accordingly.

$$f(y_1, y_2) = \frac{d}{dt}(y_1) = y_1 \left(\mu_1 + M_{11} y_1 + M_{12} y_2\right),$$

$$\implies \frac{d}{dt}(y_1) = \mu_1 y_1 \left(1 + \frac{M_{11}}{\mu_1} y_1 + \frac{M_{12}}{\mu_1} y_2\right), \tag{9}$$

$$g(y_1, y_2) = \frac{d}{dt}(y_2) = y_2 \left(\mu_2 + M_{22} y_2 + M_{21} y_1\right)$$

$$\implies \frac{d}{dt}(y_2) = \mu_2 y_2 \left(1 + \frac{M_{22}}{\mu_2} y_2 + \frac{M_{21}}{\mu_2} y_1\right). \tag{10}$$

Now, by making the substitution $\Lambda_{ij} = -\mu_i / M_{ij}$ we obtain the following set of equations,

$$f(y_1, y_2) = \frac{d}{dt}(y_1) = \mu_1 y_1 \left(1 - \frac{y_1}{\Lambda_{11}} - \frac{y_2}{\Lambda_{12}}\right), \tag{11}$$

$$g(y_1, y_2) = \frac{d}{dt}(y_2) = \mu_2 y_2 \left(1 - \frac{y_2}{\Lambda_{22}} - \frac{y_1}{\Lambda_{21}}\right). \tag{12}$$

Recall that in the case of mono-culture populations $M_{12} = M_{21} = 0$, well, in the reparameterised case $-1/\Lambda_{12} = 0/\mu_1 = 0$ and $-1/\Lambda_{21} = 0/\mu_1 = 0$ leading to the following mono-culture equations.

$$\left.\begin{aligned} f(y_1, y_2) = \frac{d}{dt}(y_1) = \mu_1 y_1 \left(1 - \frac{y_1}{\Lambda_{11}}\right) \\ g(y_1, y_2) = \frac{d}{dt}(y_2) = \mu_2 y_2 \left(1 - \frac{y_2}{\Lambda_{22}}\right) \end{aligned}\right\} \text{ Mono-culture equations} \tag{13}$$

2.2.2 Parameter Constraints

Given the trends observed in Figure 1 and biological theory, we can make some assumptions about the values that our parameter values will take. Firstly, we assume the growth rate μ_i to be strictly positive ($0 < \mu_i$). A growth rate of zero suggests that the microbial population size never increases. This cannot possibly be the cases given that the mono-culture microbial populations continue to grow over the entire 13 hour duration. Furthermore, a negative growth rate is just not biologically feasible. We can make the same assumption about intraspecies interactions which describe how a microbial species interacts with itself. Assuming homogeneity we can rule out any intraspecies competition for resources. Furthermore it is unlikely for microbial interactions to neutral, so we can hence assume that intraspecies interactivity parameters are strictly positive ($0 < \Lambda_{ii}$). There is no evidence to suggest otherwise so interspecies interactivity terms remain unconstrained. The parameter values and their respective bounds are summarised in Table 2.

Parameter	**Meaning**	**Constraints**
μ_i	Growth Rate	$0 < \mu_i$
Λ_{ii}	intraspecies Interactions	$0 < \Lambda_{ii}$
Λ_{ij}	interspecies Interaction	$\Lambda_{ij} \neq 0$ (Co-culture) $\Lambda_{ij} = 0$ (Mono-culture)

Table 2: Assumed parameter constraints for different model parameters. Assumptions are justified using the experimental data and through consideration of biological feasibility.

2.3 Parameter Estimation

In order to simulate microbial population dynamics using the gLV model we must determine parameter values that best fit the experimental data. There exist several methods of parameter estimation with some of the best known approaches being 'least-squares regression' and 'maximum likelihood estimation'. In general, approaches aim to minimise the discrepancy between the observed data and model-predicted values. This is typically achieved by first establishing a function (objective, cost or discrepancy function) which condenses (measures) the difference between observed and predicted data. Once established, the function can be used to evaluate how well the predicted data describes the observed data, with the best-fitting models having a small value. So, in order to achieve an optimum fit one minimises the difference function by systematically adjusting the parameter value inputs in an iterative manner. Different parameter estimation techniques use different types of difference functions with their own subtle differences, but ultimately share a common goal.

For the duration of this report, we employ Bayesian inference as our statistical inferencing tool of choice.

2.3.1 Bayesian Inferencing

Bayesian inference is a powerful framework for making probabilistic inferences using prior knowledge and updates itself as more information becomes available. The approach is more robust than other methods such as least-squares regression, but more computationally expensive in return. To utilise Bayesian inference, we must first outline Bayes' theorem, a method of calculating conditional probability, which lies at the heart of the approach. Let $P(H)$ and $P(D)$ denote the probabilities of events H (the hypothesis / unknown quantities) and D (the data) occurring irrespective of each other (marginal probability). Additionally, denote the probability of H occurring given D occurring (conditional probability), $P(H \mid D)$ and vice versa. Bayes' theorem is hence stated by the following,

$$P(H \mid D) = \frac{P(D \mid H)P(H)}{P(D)}. \tag{14}$$

The terms of Bayes' theorem can be interpreted as,

1. $P(H \mid D)$: Posterior Probability
2. $P(D \mid H) = L(H \mid D)$: The likelihood (of H given D)
3. $P(H)$: Prior Information
4. $P(D)$: Observed Data / Evidence

In Bayesian inference prior information is expressed as a probability distribution called the 'prior distribution' and reflects the uncertainty surrounding it. Incorporation of prior knowledge is one of the Bayesian approach's strong suits and is particularly useful when there is lots of available knowledge surrounding the hypothesis in question. In the case of limited domain knowledge/large uncertainty one can use 'non-informative/vague priors' which often span a large parameter space. Vague priors retract from the value of Bayesian inference, but are still an improvement over no available data. In context to parameter estimation, knowledge surrounding the unknown parameter quantities serves as the prior information and this knowledge is updated/built upon with the introduction of new information. An example of prior information is the biological significance and the constraints we devised for our unknown parameters.

Remaining terms include the posterior, likelihood and the 'normalising' term ($P(B)$). The posterior term describes the probability that the hypothesis/parameter estimation is correct given the observed data and is proportional to the product of the likelihood and prior information (this is supposing that the normalising term is some constant).

$$Posterior \propto Likelihood \times Prior \tag{15}$$

After determining a posterior given some initial likelihood and prior information, the posterior (new information) becomes the new prior belief. This prior can then be used to calculate a new posterior and hence we have a systematic method of Bayesian updating. In context to parameter estimation, an initial prior distribution is specified and refined based on how well the parameter estimates fit the experimental data. The distribution of parameter estimates is continually refined until posterior describes the observed data well. The final posterior can then be summarised in a number of ways, the most common of which are through interval estimates (confidence intervals) and point estimates (posterior mean, median, mode).

In utilising Bayesian inferencing several computational problems can arise especially in higher dimensions. There exist several approaches to overcome the computational difficulties associated with a Bayesian approach, one of which is the 'Markov Chain Monte Carlo' (MCMC) method. In practice the normalisation factor $P(D)$ is not always straight forward to compute, as a result, calculating the posterior distribution can be challenging or even infeasible,

$$P(D) = \int_H P(D \mid H)P(H)dH. \tag{16}$$

MCMC sampling methods offer a solution to this dilemma. The Monte Carlo element on the method involves repeated random sampling to obtain numerical approximations/results. This is particularly useful in generating approximations to quantities that cannot be calculated analytically. Markov Chains are memoryless stochastic processes that describe a sequence of probabilistically related events. Based on a fixed set of probabilities (the stationary distribution), each event outcome is determined by the outcome before it. By setting the stationary distribution to the distribution we want to sample from (i.e., the posterior distribution), we can establish a Markov Chain to simulate and obtain results from. A combination of both of these concepts ultimately form the MCMC method.

In order to put Bayesian inference and MCMC methods into practice, we turn to 'Stan', a state-of-the-art program for statistical modelling and computation. Stan offers two methods of MCMC sampling, namely the No-U-Turn Sample (NUTS) and the Hamiltonian Monte Carlo (HMC) method. For this project we will be using the latter. As expected, begin by specifying prior distributions for the unknown quantities θ. Now, having specified a number of iterations, an auxiliary momentum variable ρ is generated and draws are taken from a newly formed joint probability,

$$P(\rho, \theta) = P(\theta)P(\rho \mid \theta), \tag{17}$$

where $P(\rho, \theta)$ is the probability of ρ and θ occurring. The purpose of the auxiliary momentum variable is to enable more effective exploration of the posterior distribution. It does this by helping avoid inefficient random walk behaviour which can be observed in other sample methods. Hamiltonian dynamics are unavailable without the introduction of this parameter, for which we can use any distribution to describe such. Stan describes the auxiliary density as being some multivariate normal density that is independent of θ,

$$\rho \sim \mathbf{MultiNorm}(0, M), \tag{18}$$

where M represents the Euclidean distance. Now, Stan uses the newly established joint

density to define a Hamiltonian, an operator commonly used in mechanics to describe the total energy of a system. The Hamiltonian operator is as follows,

$$H(\rho, \theta) = -\log(P(\rho, \theta)) = -\log P(\theta) - \log P(\rho \mid \theta), \tag{19}$$

where we define the potential energy as,

$$V(\theta) = -\log P(\theta), \tag{20}$$

and the kinetic energy as,

$$T(\rho \mid \theta) = -\log P(\rho \mid \theta). \tag{21}$$

We now discuss generating Markov transitions between states. After defining a distribution for auxiliary momentum and equations for kinetic and potential energy, the joint system (ρ, θ) is updated/evolved using Hamilton's equations. Firstly, θ evolves in time according to,

$$\frac{d\theta}{dt} = \frac{\partial}{\partial \rho} H(\rho, \theta) = \frac{\partial}{\partial \rho} T(\rho \mid \theta), \tag{22}$$

we only needed to take the partial derivative of T. To determine how ρ changes in time,

$$\frac{d\rho}{dt} = -\frac{\partial}{\partial \theta} H(\rho, \theta) = \frac{\partial}{\partial \theta}(-V - T). \tag{23}$$

Having assumed that the auxiliary momentum is independent of θ, $P(\rho \mid \theta)$ is simply $P(\rho)$ and so $T(\rho \mid \theta) = T(\rho)$, therefore $\partial_\theta T = 0$ and the evolution of ρ is,

$$\frac{d\rho}{dt} = -\frac{\partial}{\partial \theta} V(\theta). \tag{24}$$

We now have a pair of time derivatives,

$$\left(\frac{d\theta}{dt}, \frac{d\rho}{dt}\right) = \left(\frac{\partial T}{\partial \rho}, -\frac{\partial V}{\partial \theta}\right). \tag{25}$$

To solve this system of ODEs, Stan uses the leapfrog integration method (this is typical of most HMC implementations). The leapfrog method is a second-order symplectic (conserves the nature of Hamilton's equations) numerical approximation method that alternates between half-step updating the auxiliary momentum and full-step updating the unknown quantities (position) of our system. Let Δt denote the time step size, then ρ and θ are approximated as follows,

$$1. \quad \rho_{i+1/2} = \rho_i - \frac{\Delta t}{2}\left(\frac{\partial V}{\partial \theta}\right) \tag{26}$$

$$2. \quad \theta_{i+1} = \theta_i + \Delta t \rho_{i+1/2} \tag{27}$$

$$3, \quad \rho_{i+1} = \rho_{i+1/2} - \frac{\Delta t}{2}\left(\frac{\partial V}{\partial \theta}\right) \tag{28}$$

After N repetitions of the three steps above, the resultant state is (ρ^*, θ^*). Finally, we apply a Metropolis acceptance step to account for the leapfrog method's numerical error. The acceptance step ultimately determines whether the resultant state (ρ^*, θ^*) is accepted or rejected. If the resultant state is accepted, it replaces the current (ρ, θ) and is used to initialise the next iteration of calculations. The probability of accepting the proposed resultant state is,

$$\min(1, \exp\left[H(\rho, \theta) - H(\rho^*, \theta^*)\right]). \tag{29}$$

To implement HMC using Stan we use RStudio and the RStan library. From here, we

describe our Bayesian statistical model as a Stan script (`.stan` file type) and call it via an R script (`.r` file type) where we input and process our experimental data, define initial conditions and specify simulation options. It is worth mentioning configuration of our HMC simulations as well as the priors used. Table 3 summarises the sampling options used in numerical computations and Table 4 discusses our choice of priors and justification.

Option	**Value**	**Meaning**
`seed`	$\mathbb{Z}$	Seed for Monte Carlo random number generation.
`chains`	15	Number of Markov chains to simulate.
`iter`	30000	Total number of leapfrog iterations for each Markov chain/process (includes warmup iterations).
`warmup`	$\frac{\texttt{iter}}{2}$	Number of burnin iterations. These iterations are not used for sampling, but instead helps the Markov chain reach its equilibrium distribution.
`init_r`	2 (*default*)	Controls range of randomly generated values.

Table 3: Configure options for HMC sampling. Note that the `seed` value is arbitrary and can be any integer value.

In order to guarantee convergence and reliable results we must use a sufficient number of iterations and Markov chains. 30000 iterations (default is 2000) and 15 chains (default is 4) seems to be a sensible choice for all future simulations. Running this amount of simulations is computationally expensive, but should limit the number of convergence issues we run into. Next the prior information. The choice of prior distribution used for each unknown quantity is just as important than the sampling configuration. A poor or overly vague choice of prior can make convergence infeasible and hence lead to poorly fitting results. Table 4 below discusses choice of priors and our accompanying thought process.

Unknown Quantity	**Distribution of Choice**	**Explanation**
μ_i	$\text{Cauchy}(1, 0.01)$	Growth rates are strictly positive and unlikely to assume large values
Λ_{ii}	$\text{Cauchy}(100000, 1000)$	intraspecies interactions are constraint to be positive.
Λ_{ij}	$\text{Cauchy}(0, 100000)$	interspecies interactivity parameters can assume any non-zero value

Table 4: Summary of different model priors. Cauchy priors have been a popular choice of prior for Bayesian inference [40]. We employ them for their heavy tails which compliment our lack of prior knowledge surrounding the parameter estimates. These heavy tails will help us explore the parameter space sufficiently and identifying converging results. Our choice of priors leverages computational efficiency and appropriate exploration of the parameter space.

Having discusses some of the theory behind Bayesian inference and its implementation using RStan, we can now talk about the manner in which we estimate the parameters.

2.3.2 Workflow

The availability of mono and co-culture data allows us to estimate parameter values for both cases and compare their differences. There are several different ways in which we can estimate the co-culture data, some ways utilise mono-culture parameter estimates and others do not. Depending on the approach we use, we can infer different biological meanings from our results as each features their own assumptions. In Table 5, we summarise the different cases of parameter estimation and their respective assumptions.

Case	**Assumptions**
$1a$	Growth rate, μ_i, and intraspecies interactions, Λ_{ii}, are intrinsic to microbial species and unaffected by interspecies interactions, Λ_{ij} where $i \neq j$.
$1b$	Interspecies interactions, Λ_{ij} where $i \neq j$, directly affect the growth rate, μ_i, but not the intraspecies interactions, Λ_{ii}, of neighbouring microbial species.
$1c$	Interspecies interactions, Λ_{ij} where $i \neq j$, directly affect the intraspecies interactions, Λ_{ii}, but not the growth rate, μ_i, of neighbouring microbial species.
$1d$	Interspecies interactions, Λ_{ij} where $i \neq j$, directly affect the growth rate, μ_i, and intraspecies interactions, Λ_{ii}, of neighbouring microbial species.

Table 5: Cases $1a-d$ and their respective assumptions. These cases represent different workflows during parameter estimation. The assumptions attached to each case provides biological context to any results.

Alternatively, we can visualise the different paths of parameter estimation using a flow chart diagram. Figure 3 outlines the methodology behind different cases of parameter estimation, but does not comment on the accompanying biological assumptions/meaning.

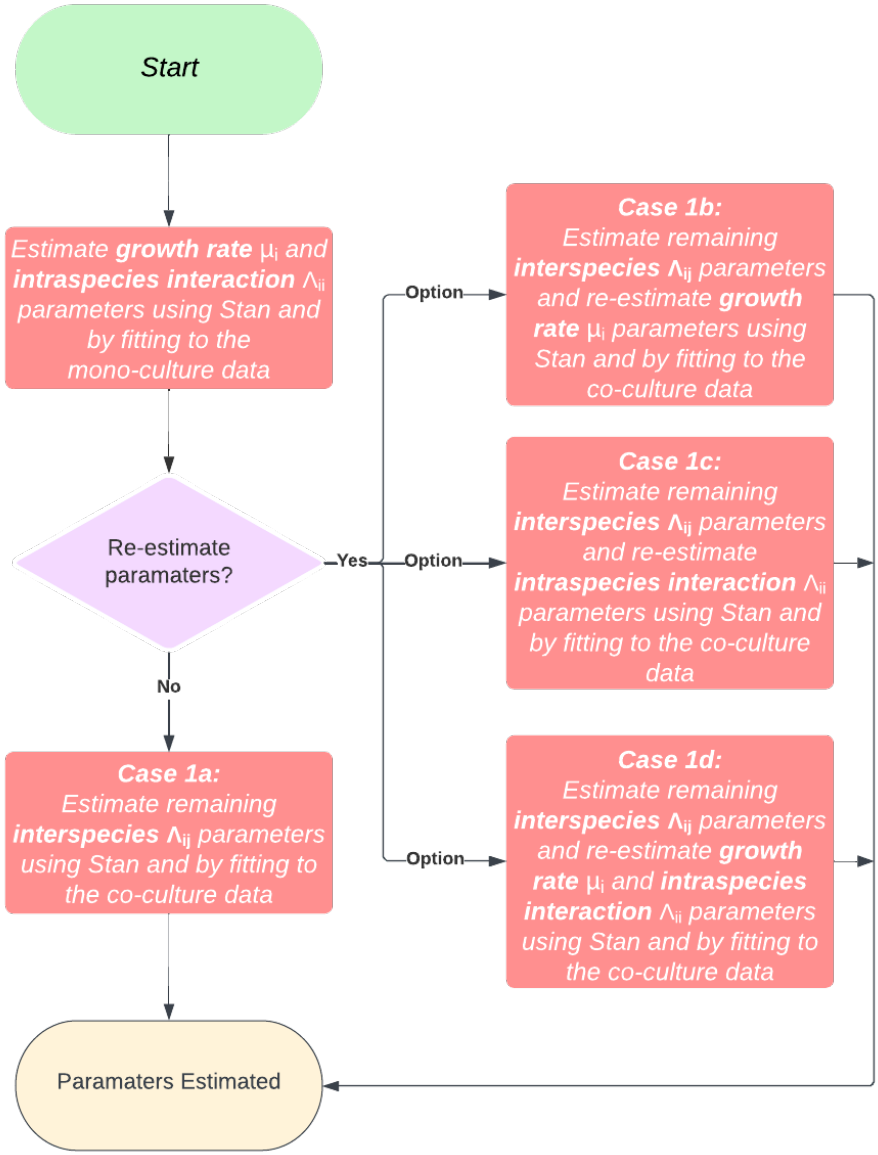

Figure 3: Workflow of parameter estimation.

2.3.3 Validation

Once we have generated some posterior distributions describing our parameter estimates, we can evaluate how well these results fit the experimental data. One way of doing this is by taking point averages of the posterior distributions and substituting these parameters into the original differential system. We can then integrate the equations to get a continuous curve of results which can then be compared to the observed data. To solve the ODE system we step each equation forward in time using MATLAB's built-in `ODE45` variable step size solver.

Additionally, we are concerned with the mixing of our Markov chains. Well mixed Markov chains imply that the parameter space has been explored efficiently and effectively. To diagnose convergence and investigate mixing we can consider trace plots and convergence diagnostics. Rhat is the convergence statistic that we will use to measure convergence. It is calculated by considering the variance within Markov chains and between them. An Rhat value of 1 means the solution has converged where as an Rhat greater than 1 suggests poor mixing and limited convergence.

2.4 Steady-State Analysis

Steady-state analysis is a useful tool for understanding the behaviour of a mathematical system over time. Thanks to the low dimensionality of our system, we are able to investigate the steady-state dynamics analytically using nullclines. Determining the steady states will aid us in explaining the differences between the mono and co-culture experiments, leading on nicely to stability analysis and bifurcation analysis.

2.4.1 Nullclines

Nullclines are sets of points (curves) in which the rate of change of a system state variable is 0. In the context of our model, this translates to the points at which y_1 and y_2 are constant in time. So, to find the nullclines of the system first consider the time invariant state of our equations:

$$f(y_1, y_2) = \frac{d}{dt}(y_1) = \mu_1 y_1 \left(1 - \frac{y_1}{\Lambda_{11}} - \frac{y_2}{\Lambda_{12}}\right) = 0, \tag{30}$$

$$g(y_1, y_2) = \frac{d}{dt}(y_2) = \mu_2 y_2 \left(1 - \frac{y_2}{\Lambda_{22}} - \frac{y_1}{\Lambda_{21}}\right) = 0, \tag{31}$$

solution of these equations yield the nullclines. Let the solutions of $\frac{dy_1}{dt} = 0$ be the y_1-nullclines and $\frac{dy_2}{dt} = 0$ be the y_2-nullclines, respectively. Firstly, solving Equation (X) gives the following y_1-nullclines,

$$\mu_1 y_1(t) = 0 \quad \text{or} \quad 1 - \frac{y_1}{\Lambda_{11}} - \frac{y_2}{\Lambda_{12}} = 0,$$

inferring that,

$$\implies \boldsymbol{y_1 = 0} \quad \text{or} \quad \boldsymbol{y_1 = \Lambda_{11}\left(1 - \frac{y_2}{\Lambda_{12}}\right)} \tag{32}$$

Here we have a y_1-nullcline along the $y_1 = 0$ axis and a straight line equation dependent on Λ_{11} and Λ_{12}. Similarly, we calculate the y_2-nullclines by solving Equation (Y),

$$\mu_2 y_2(t) = 0 \quad \text{or} \quad 1 - \frac{y_2}{\Lambda_{22}} - \frac{y_1}{\Lambda_{21}} = 0,$$

hence, leading to

$$\implies \boldsymbol{y_2 = 0} \quad \text{or} \quad \boldsymbol{y_2 = \Lambda_{22}\left(1 - \frac{y_1}{\Lambda_{21}}\right)} \tag{33}$$

Similar to the case of the y_1-nullclines, the y_2-nullclines take the form of two straight lines, one running along the $y_2 = 0$ axis and the other dependent on Λ_{22} and Λ_{21}. Having derived equations for the system's nullclines, the position of any fixed points (equilibria) can now be established using the intersections of between y_1-nullclines and y_2-nullclines (self-intersection yields no equilibria).

2.4.2 Equilibrium Points

As mentioned, the position of any equilibrium points can be found by considering intersections between y_1 and y_2-nullclines. Due to the nature of the nullclines being straight lines we can expect a single intersection point between each combination of y_1 and y_2-nullclines. Let (y_1^*, y_2^*) denote the position of an equilibrium point (point of intersection). Equilibrium points involving the intersection of 0-axis nullclines are trivial to compute:

$$
\begin{aligned}
y_1 = 0 \;\text{ and }\; y_2 = 0 \quad &\implies \quad (y_1^*, y_2^*) = (0,0), \\
y_1 = \Lambda_{11}\left(1 - \frac{y_2(t)}{\Lambda_{12}}\right) \;\text{ and }\; y_2 = 0 \quad &\implies \quad (y_1^*, y_2^*) = (\Lambda_{11}, 0), \\
y_1 = 0 \;\text{ and }\; y_2 = \Lambda_{22}\left(1 - \frac{y_1(t)}{\Lambda_{21}}\right) \quad &\implies \quad (y_1^*, y_2^*) = (0, \Lambda_{22}).
\end{aligned}
$$

Three of the system's equilibria are situated on axis in the y_1y_2-plane. Given the constraints made on our reparameterised system, these equilibrium points are contained to first quadrant of the y_1y_2-plane which are biologically feasible values. Additionally, consider the intersection between non-zero nullclines,

$$
\begin{aligned}
& y_1 = \Lambda_{11}\left(1 - \frac{y_2(t)}{\Lambda_{12}}\right) \;\text{ and }\; y_2 = \Lambda_{22}\left(1 - \frac{y_1(t)}{\Lambda_{21}}\right), \\
\implies \quad & (y_1^*, y_2^*) = \left(\frac{\left(\Lambda_{11} - \dfrac{\Lambda_{11}\Lambda_{22}}{\Lambda_{12}}\right)}{\left(1 - \dfrac{\Lambda_{11}\Lambda_{22}}{\Lambda_{12}\Lambda_{21}}\right)}, \frac{\left(\Lambda_{22} - \dfrac{\Lambda_{11}\Lambda_{22}}{\Lambda_{21}}\right)}{\left(1 - \dfrac{\Lambda_{11}\Lambda_{22}}{\Lambda_{12}\Lambda_{21}}\right)} \right), \\
\implies \quad & (y_1^*, y_2^*) = \left(\frac{(\Lambda_{12} - \Lambda_{22})\Lambda_{11}\Lambda_{21}}{\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22}}, \frac{(\Lambda_{21} - \Lambda_{11})\Lambda_{12}\Lambda_{22}}{\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22}} \right).
\end{aligned}
$$

In the case of mono-culture growth $\Lambda_{12} = \Lambda_{21} = 0$ and the equilbrium point can be found at $(y_1^*, y_2^*) = (\Lambda_{11}, \Lambda_{22})$ (the intersecting nullclines are horizontal and vertical lines). Positions of the system's fixed points are outlined in Table 6.

Equilibrium Point - (y_1^*, y_2^*)	**Biological Meaning (Stable)**
$(0,0)$	Extinction of y_1 and y_2
$(\Lambda_{11}, 0)$	y_1 out-competes y_2
$(0, \Lambda_{22})$	y_2 out-competes y_1
$\left(\frac{(\Lambda_{12}-\Lambda_{22})\Lambda_{11}\Lambda_{21}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}, \frac{(\Lambda_{21}-\Lambda_{11})\Lambda_{12}\Lambda_{22}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)$	Co-existence of y_1 and y_2

Table 6: Steady states and respective biological meaning (if classified as a stable node). Depending on parameter estimate values of Λ_{ij} some equilibria may occur at biologically-unfeasible values. It is important to still investigate how said equilibria may affect the dynamics of our system.

It remains to investigate the stability of our newly found equilibrium points. Stability depends on the parameter values of the system so it makes sense to discuss parameter estimation before diving into classifying stability.

2.5 Dynamical Systems Analysis

Having analytically derived the existence and location of the system's fixed points and outlined estimation of system parameters, we can now discuss further means of understanding system dynamics using stability and bifurcation analysis, two methods which describe how system dynamics change in time.

2.5.1 Stability Analysis

Stability analysis investigates the response of a dynamical system to small perturbations, which is very useful in understanding how the system might change over time. We begin our stability analysis by calculating the Jacobian, a matrix of partial derivatives that, when evaluated at an equilibrium point, describes the nature of that equilibrium. The Jacobian is calculated using the following,

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} \end{bmatrix}, \tag{34}$$

where $f_{1,2}$ describe the ODE functions and $y_{1,2}$ the state variables. In the case of our system, the Jacobian will be slightly different based on whether one is considering the two species in mono or co-culture. However, these changes are also reflected in calculations later on, so we only derive the more inclusive Jacobian. Substituting in Equations (X-Y) yields,

$$\boldsymbol{J} = \begin{bmatrix} \mu_1 \left(1 - \frac{2y_1}{\Lambda_{11}} - \frac{y_2}{\Lambda_{12}}\right) & \mu_1 y_1 \left(-\frac{1}{\Lambda_{12}}\right) \\ \mu_2 y_2 \left(-\frac{1}{\Lambda_{21}}\right) & \mu_2 \left(1 - \frac{2y_2}{\Lambda_{22}} - \frac{y_1}{\Lambda_{21}}\right) \end{bmatrix}. \tag{35}$$

We have successfully derived the Jacobian matrix for our system. We must now calculate the eigenvalues of the Jacobian for which the following holds:

$$\boldsymbol{J}\mathbf{x} = \lambda \boldsymbol{I}\mathbf{x} \tag{36}$$

Here, $\boldsymbol{J}$ is the Jacobian matrix, $\boldsymbol{I}$ is the 2×2 identity matrix, λ is some scalar eigenvalue and $\mathbf{x}$ represents a corresponding non-zero eigenvector. Bringing both terms to one side,

$$\boldsymbol{J}\mathbf{x} - \lambda \boldsymbol{I}\mathbf{x} = 0, \tag{37}$$

we can now find λ by taking the determinant,

$$p(\lambda) = |\boldsymbol{J} - \lambda \boldsymbol{I}| = 0. \tag{38}$$

This is the characteristic polynomial equation, the roots of which provide us with the eigenvalues we are interested in. Substituting in our Jacobian and identity matrices,

$$\det \begin{bmatrix} \mu_1 \left(1 - \frac{2y_1}{\Lambda_{11}} - \frac{y_2}{\Lambda_{12}}\right) - \lambda & \mu_1 y_1 \left(-\frac{1}{\Lambda_{12}}\right) \\ \mu_2 y_2 \left(-\frac{1}{\Lambda_{21}}\right) & \mu_2 \left(1 - \frac{2y_2}{\Lambda_{22}} - \frac{y_1}{\Lambda_{21}}\right) - \lambda \end{bmatrix} = 0. \tag{39}$$

Taking the determinant of the characteristic matrix we get the following polynomial,

$$p(\lambda) = \left(\lambda - \mu_1 \left(1 - \frac{2y_1}{\Lambda_{11}} - \frac{y_2}{\Lambda_{12}}\right)\right)\left(\lambda - \mu_2 \left(1 - \frac{2y_2}{\Lambda_{22}} - \frac{y_1}{\Lambda_{21}}\right)\right) - \left(\frac{\mu_1 \mu_2 y_1 y_2}{\Lambda_{12}\Lambda_{21}}\right) = 0. \tag{40}$$

Using Equation (X) we can now classify the stability of each equilibrium point. From inspection, the characteristic polynomial is of order 2 meaning we can use the quadratic formula to find the roots/eigenvalues,

$$\lambda = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \tag{41}$$

where $a = 1$. Hence, the eigenvalues are solutions of the following form,

$$\lambda = \frac{1}{2}\left(-b \pm \sqrt{b^2 - 4c}\right). \tag{42}$$

Once the eigenvalues have been computed, stability can be determined by using each of the eigenvalues sign. Different possible stability types and respective eigenvalue sign are given in Table 7.

$\mathrm{Re}(\boldsymbol{\lambda_1})$	$\mathrm{Im}(\boldsymbol{\lambda_1})$	$\mathrm{Re}(\boldsymbol{\lambda_2})$	$\mathrm{Im}(\boldsymbol{\lambda_2})$	**Stability and Notation**	**Oscillations?**
$+$	0	$+$	0	Unstable Node (Source)	None
$-$	0	$-$	0	Stable Node (Sink)	None
$+$	0	$-$	0	Unstable Saddle Node	None
$+$	$\pm$	$+$	$\pm$	Unstable Spiral	Undamped
$-$	$\pm$	$-$	$\pm$	Stable Spiral	Damped
0	$\pm$	0	$\pm$	Neutrally Stable Center	Undamped

Table 7: Summary of possible behaviour at equilibrium points. Behaviour types are differentiated by the real and complex components of respective eigenvalues. In the case of purely real eigenvalues (zero-imaginary part) oscillatory behaviour is not observed. Note that only stable points have any biological significance. However, unstable equilibria help make sense of the system dynamics.

We now systematically devise stability criterion for each of the system's equilibrium points. Once this criterion has been established and parameter estimates have been made, we can discern the stability of each fixed point and work towards understanding system dynamics.

2.5.2 Classification $(y_1^*, y_2^*) = (0, 0)$

The stability for $(y_1^*, y_2^*) = (0, 0)$ is trivial to find. Substituting $(y_1^*, y_2^*) = (0, 0)$ into the characteristic polynomial gives us,

$$p(\lambda) = (\lambda - \mu_1)(\lambda - \mu_2) = 0. \tag{43}$$

The eigenvalues are simply $\lambda_1 = \mu_1$ and $\lambda_2 = \mu_2$. Given the previously devised parameter constraints both eigenvalues are strictly positive, $\lambda_{1,2} > 0$. Hence, $(y_1^*, y_2^*) = (0, 0)$ is an unconditionally unstable node, driving population trajectories away from itself. Under our current assumptions, in both the mono and co-culture, simultaneous extinction of *S. aureus* and *P. aeruginosa* is not feasible.

2.5.3 Classification $(y_1^*, y_2^*) = (\Lambda_{11}, 0)$

Stability classification for $(y_1^*, y_2^*) = (\Lambda_{11}, 0)$ and $(y_1^*, y_2^*) = (0, \Lambda_{22})$ follows a similar routine. Substituting $(y_1^*, y_2^*) = (\Lambda_{11}, 0)$ into the characteristic polynomial yields,

$$p(\lambda) = \left(\lambda - \mu_1\left(1 - \frac{2\Lambda_{11}}{\Lambda_{11}} - 0\right)\right)\left(\lambda - \mu_2\left(1 - 0 - \frac{\Lambda_{11}}{\Lambda_{21}}\right)\right) - 0 = 0. \tag{44}$$

This simplifies down to,

$$p(\lambda) = (\lambda + \mu_1)\left(\lambda - \mu_2\left(1 - \frac{\Lambda_{11}}{\Lambda_{21}}\right)\right) = 0. \tag{45}$$

The eigenvalues are therefore $\lambda = -\mu_1$ and $\lambda_2 = \mu_2\left(1 - \frac{\Lambda_{11}}{\Lambda_{21}}\right)$. λ_1 is strictly negative, therefore the stability depends on the value of λ_2, which itself depends on the values of Λ_{11} and Λ_{21}. Stability is outlined in the following piecewise,

$$\lambda_2 = \begin{cases} < 0 & \text{(Stable Node)} & \text{if } \Lambda_{11} > \Lambda_{21} > 0 \\ 0 & \text{(Bifurcation)} & \text{if } \Lambda_{11} = \Lambda_{21} > 0 \\ > 0 & \text{(Unstable Saddle Node)} & \text{if } \Lambda_{21} > \Lambda_{11} > 0 \text{ or } \Lambda_{21} < 0 \end{cases} \tag{46}$$

The stability of $(y_1^*, y_2^*) = (\Lambda_{11}, 0)$ is conditional on the effects y_1 has on itself (intraspecies interactions) and on y_2 (interspecies interactions). In the case of an equivalent effect, $(y_1^*, y_2^*) = (\Lambda_{11}, 0)$ is classified as a zero-eigenvalue bifurcation which will be covered more in the next section. In the case of stability, y_1 converges to Λ_{11} whilst y_2 heads towards extinction. This can be interpreted as a matter of competition and y_1 out-competing y_2.

2.5.4 Classification $(y_1^*, y_2^*) = (0, \Lambda_{22})$

Similarly, substitute $(y_1^*, y_2^*) = (0, \Lambda_{22})$ into the characteristic polynomial and recover the following,

$$p(\lambda) = \left(\lambda - \mu_1\left(1 - \frac{\Lambda_{22}}{\Lambda_{12}}\right)\right)(\lambda + \mu_2) = 0. \tag{47}$$

Once again, $\lambda_1 = -\mu_2$ is strictly negative always and so, stability depends on $\lambda_2 = \mu_1\left(1 - \frac{\Lambda_{22}}{\Lambda_{12}}\right)$. Similar to previous, the sign of λ_2 changes based on the outgoing interactions of y_2. This in turn affects the stability of the equilibrium point as summarised in the following piecewise,

$$\lambda_1 = \begin{cases} < 0 & \text{(Stable Node)} & \text{if } \Lambda_{22} > \Lambda_{12} > 0 \\ 0 & \text{(Bifurcation)} & \text{if } \Lambda_{22} = \Lambda_{12} > 0 \\ > 0 & \text{(Unstable Saddle Node)} & \text{if } \Lambda_{12} > \Lambda_{22} > 0 \text{ or } \Lambda_{12} < 0 \end{cases} \tag{48}$$

2.5.5 Classification $(y_1^*, y_2^*) = \left(\frac{(\Lambda_{12} - \Lambda_{22})\Lambda_{11}\Lambda_{21}}{\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22}}, \frac{(\Lambda_{21} - \Lambda_{11})\Lambda_{12}\Lambda_{22}}{\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22}}\right)$

It remains to determine stability criterion for $(y_1^*, y_2^*) = \left(\frac{(\Lambda_{12}-\Lambda_{22})\Lambda_{11}\Lambda_{21}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}, \frac{(\Lambda_{21}-\Lambda_{11})\Lambda_{12}\Lambda_{22}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)$. Recall that in the case of mono-culture growth the non-zero equilibrium occurs at $(y_1^*, y_2^*) = (\Lambda_{11}, \Lambda_{22})$. We will first classify equilibrium stability in absence of any interspecies interactions. Using the characteristic polynomial and assuming $1/\Lambda_{12} = 1/\Lambda_{21} = 0$ and $(y_1, y_2) = (\Lambda_{11}, \Lambda_{22})$:

$$p(\lambda) = \big(\lambda - \mu_1(1 - 2 - 0)\big)\big(\lambda - \mu_2(1 - 2 - 0)\big) - 0 = (\lambda + \mu_1)(\lambda + \mu_2) = 0. \tag{49}$$

The eigenvalues are equal to the species growth rates, $\lambda_{1,2} = \mu_{1,2}$, and therefore the equilibrium point is always a stable sink node. This seems sensible considering there are

no interspecies interactions that might encourage competition or have negative effects on either of the species growth. In such a case equilibria situated at $(\Lambda_{11}, 0)$ and $(0, \Lambda_{22})$ are unstable saddle nodes and direct trajectories towards $(\Lambda_{11}, \Lambda_{22})$.

Devising stability criterion in the co-culture case is significantly more complicated. Because of such, attempts to analytically derive stability criterion will be left for the appendix. With trouble deriving the criterion analytically, we can substitute our parameter estimates directly into the equilibrium point and into the characteristic polynomial to find the stability.

We now discuss some of bifurcation theory to aid identification of our system's bifurcations later on.

2.5.6 Bifurcation Theory and Analysis

Qualitative changes in system dynamics are named 'bifurcations'. These changes in dynamics occur as system parameters are varied. Examples of such changes include equilibrium points colliding, exchanging stability, or even annihilating, which in turn influences the behaviour of the system. Bifurcations possess great scientific importance and are studied in a variety of contexts, including finance, engineering and biology. As part of our effort to investigate the mechanisms underlying interactions between *S. aureus* and *P. aeruginosa* we perform bifurcation analysis to decipher any sensitivity the system might have to parameter changes.

In order to perform bifurcation analysis we first choose a control parameter (or in the case of higher dimensional bifurcation analysis, parameters) and fix the remaining parameter terms. We can then systematically vary the parameter(s) of interest whilst noting any changes in the total number of equilibrium points and any variations in their respective eigenvalues. One instance in which bifurcations occur is when an equilibrium point's eigenvalue changes sign, namely when $\lambda = 0$. These types of bifurcations points are called 'zero-eigenvalue bifurcations' and are probably the most common type of bifurcation to occur.

For any bifurcations we encounter during our analysis, we can classify the type of bifurcation based on genericity conditions and how the number of equilibrium points changes at the bifurcation point. In this context, genericity conditions are bifurcation-specific criteria that enable us to verify whether its possible for a specific type of bifurcation to occur. For example, one genericity condition for the zero-eigenvalue pitchfork bifurcation is that the system contains a cubic term, namely

$$f_{xxx}(x_0, \mu^*) \neq 0 \quad \textbf{Cubic term does not vanish}, \tag{50}$$

where (x_0, μ^*) describes the reference point / parameter values at which the bifurcation occurs. Clearly from our gLV equations we do not have a cubic term and so the criterion above could never be satisfied. From this we know that a pitchfork bifurcation cannot occur for our current system. The remaining zero-eigenvalue bifurcations, saddle-node and transcritical bifurcation, are certainly more feasible. Conditions for these bifurcations are outlined below.

Saddle-Node Bifurcation

The saddle-node bifurcation is a mechanism in which fixed points are created and annihilated. It features the collision of two fixed points, one stable and the other unstable. Their collision brings about their destruction and the qualitative behaviour of the system changes as a result. The co-dimension 1 bifurcation has a variety of important meanings in the field of biology and are is of the more common bifurcation types to occur.

Let $\dot{x} = f(x, \mu)$ where $x, \mu \in \mathbb{R}$. If the following conditions are satisfied,

$$\begin{aligned} f(x_0, \mu^*) &= 0 \quad \text{(Equilibrium at } (x_0, \mu^*)\text{)} \textbf{ [E1]} \\ f_x(x_0, \mu^*) &= 0 \quad \text{(Single zero eigenvalue at } (x_0, \mu^*) = 0\text{)} \textbf{ [B1]} \\ f_{xx}(x_0, \mu^*) &\neq 0 \quad \text{(Quadratic term does not vanish)} \textbf{ [G1]} \\ f_{x\mu}(x_0, \mu^*) &\neq 0 \quad \text{(Positive speed in } \mu\text{)} \textbf{ [G2]} \end{aligned}$$

then there is a saddle-node bifurcation at (x_0, μ^*). Certainly our system contains quadratic terms and fulfils the second genericity condition meaning that we could potentially have a saddle-node bifurcation.

Transcritical Bifurcation

Transcritical bifurcations, unlike saddle-node bifurcations, do not involve the destruction or creation of any equilibria. Instead, two equilibria collide and exchange stability. Transcritical bifurcations occur at fixed points which exist for all values. We might expect a transcritical bifurcation to occur at $(y_1^*, y_2^*) = (\Lambda_{11}, 0)$ or $(y_1^*, y_2^*) = (0, \Lambda_{22})$, which exist for all values of Λ_{ii}.

Like in the saddle-node's case, let $\dot{x} = f(x, \mu)$ where $x, \mu \in \mathbb{R}$. If the following conditions are satisfied,

$$\begin{aligned} f(x_0, \mu^*) &= 0 \quad \textbf{for all } \mu \in \mathbb{R} \textbf{ [E1]} \\ f_x(x_0, \mu^*) &= 0 \quad \text{(Single zero eigenvalue at } (x_0, \mu^*) = 0\text{)} \textbf{ [B1]} \\ f_{xx}(x_0, \mu^*) &\neq 0 \quad \text{(Quadratic term does not vanish)} \textbf{ [G1]} \\ f_{x\mu}(x_0, \mu^*) &\neq 0 \quad \text{(Positive speed in } \mu\text{)} \textbf{ [G2]} \end{aligned}$$

then there is a transcritical bifurcation at (x_0, μ^*). Here the conditions are almost identical to that of a saddle-node bifurcation with the exception that no equilibria are created or destroyed at a transcritical bifurcation. With our system satisfying both sets of criteria for saddle-node and transcritical bifurcations, to differentiate between the two we will have to examine how the number of equilibria changes at any potential zero-eigenvalue bifurcation points.

3 Results

Using the approaches and tools featured in our methodology section, we calculated parameter estimates for the 6 unknown quantities in our gLV model, analyse the stability of the system's equilibria and perform bifurcation analysis.

3.1 Review and Comparison of Parameter Estimates

3.1.1 Mono-Culture Parameter Estimation

We begin with computing parameter estimates for the mono-culture case. Here there is only one approach we can take which is to estimate both the growth rate μ_i and the intraspecies term Λ_{ii} using the mono-culture data. By feeding the mean-averaged, normalised mono-culture data into RStudio and appropriately framing the problem as a Stan model, we run HMC simulations and make parameter estimates.

In all cases, we find that our results successfully converged indicated by an Rhat score of 1 (see Table X). Point averages of our parameter estimates are also inline with our prior expectations, with growth rates being roughly equal to 1 ($\mu_i \sim 1$) and intraspecies interactions being in the ten thousands. Table X summarises the posterior distributions, describing convergence (Rhat), any variation in results (SD), point estimates (Mean and

Median (50% confidence interval)) and interval estimates (2.5%, 50% = – and 97.5% confidence intervals). In the mono-culture case $1/\Lambda_{ij} = 0$, therefore, no parameter estimates are recorded for any interspecies parameters.

Table 8: Summary of parameter estimation for mean-averaged and normalised mono-culture data using Bayesian inference and HMC sampling.

Parameter	**Rhat**	**Mean**	**SD**	**2.5%**	**50%**	**97.5%**
μ_1	1.00	1.01	0.06	0.91	1.01	1.14
μ_2	1.00	0.88	0.06	0.77	0.87	1.01
Λ_{11}	1.00	16905.37	3464.62	10853.09	16724.89	23979.05
Λ_{22}	1.00	18185.56	142280.55	8439.40	13790.53	38250.25
Λ_{12}	-	-	-	-	-	-
Λ_{21}	-	-	-	-	-	-
σ_1	1.00	3814.94	919.30	2496.38	3658.98	6050.01
σ_2	1.00	1754.41	446.56	1125.11	1675.17	2844.01

Firstly, recall that in mono-culture the system has one stable root at $(y_1^*, y_2^*) = (\Lambda_{11}, \Lambda_{22})$, meaning that the y_1 and y_2 populations converge to Λ_{11} and Λ_{22}, respectively. The parameter estimates reveal *S. aureus* to have, on average, a larger intrinsic growth rate and a larger median intraspecies interaction value. From this, one might suppose that, most of the time, *S. aureus* grows more rapidly and to larger population sizes than *P. aeruginosa* when grown alone. The large standard deviation of *P. aeruginosa* infers that there is a lot of discrepancy in the population sizes it converges to and sustains. This variation is also reflected in the mean of Λ_{22} being greater than the mean of Λ_{11}. To better illustrate the dynamics, we plot the y_1y_2-phase plane and time-series solutions. With the stability of the mono-culture system being fixed and having reviewed such analytically, there is no need to compute any eigenvalues.

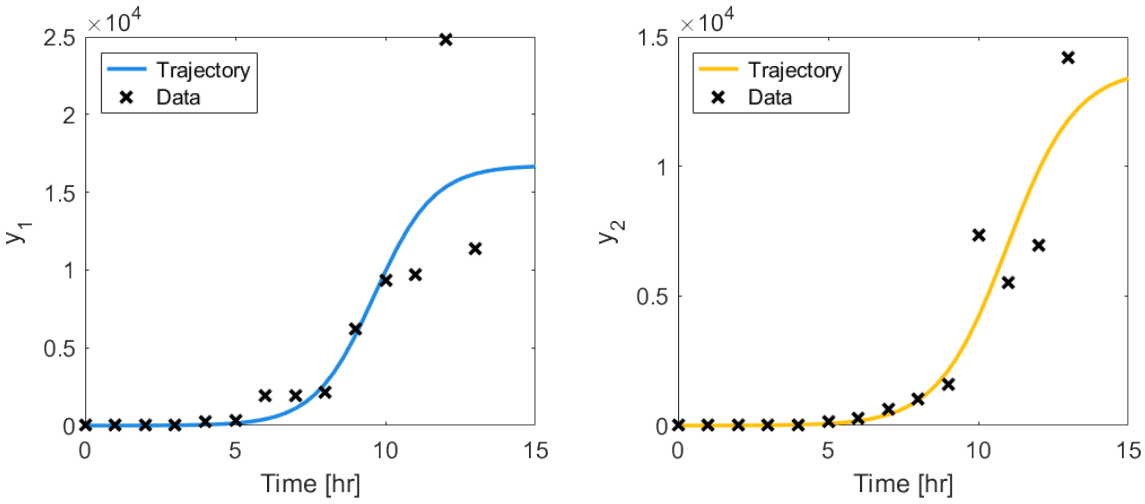

Figure 4: Time-Series solutions using median values of the posterior distributions. The results fit the y_2 observed data better than y_1 data.

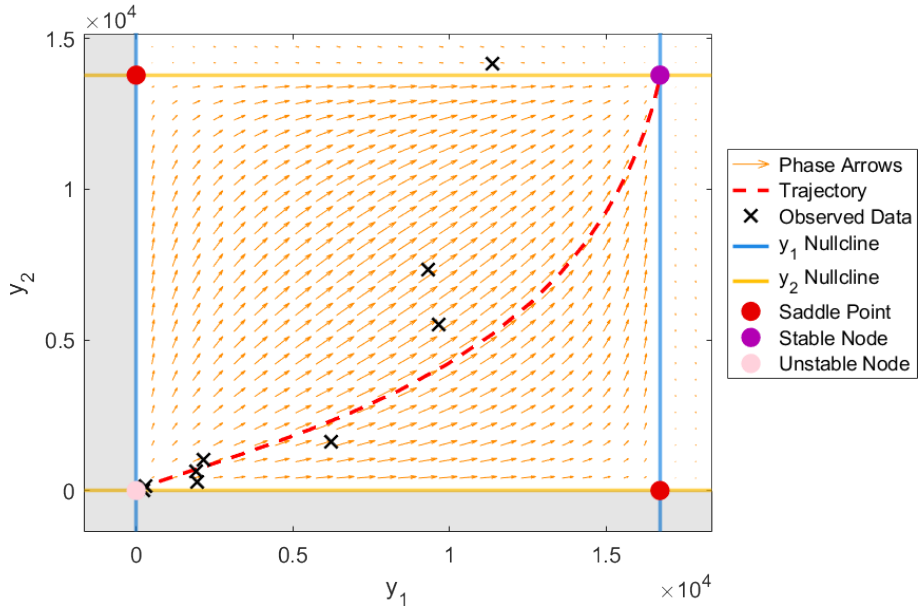

Figure 5: y_1y_2-phase plane demonstrating dynamics of the mono-culture system. The dynamics are relatively straight forward with the equilibrium points being as described previously. Populations originate close to the origin and are projected towards the stable node, indicated by the direction of the flow vectors/phase arrows. Grey shaded regions indicate biologically infeasible population values and parameter values are summarised by their posterior's median (which is why y_1 grows to a higher value than y_2).

3.1.2 Co-Culture Parameter Estimation

From here, we are interested in comparing the mono-culture parameter estimate results to that of any co-culture results, and ultimately answer whether the gLV model explains the observable data. To do this, we first calculate results for each of the 4 co-culture cases.

Using the same methods as for the mono-culture parameter estimation and following the different workflows outlined in Figure 3, we generate posterior distributions based on the mean-averaged, normalised co-culture data. For brevity we put the summarised posteriors of each case in the appendix section. Instead we create a table outlining the posterior medians for each parameter value in each case.

Case	$\boldsymbol{\mu_1}$	$\boldsymbol{\mu_2}$	$\boldsymbol{\Lambda_{11}}$	$\boldsymbol{\Lambda_{22}}$	$\boldsymbol{\Lambda_{12}}$	$\boldsymbol{\Lambda_{21}}$
Case 1a	1.01	0.87	16724.89	13790.53	460.64	105897.27
Case 1b	0.99	1.28	16724.89	13790.53	5002.28	-321331.69
Case 1c	1.01	0.87	97130.57	121839.35	460.18	106665.19
Case 1d	0.97	1.12	94155.04	2394258.29	2813.77	103731.30

Table 9: Parameter estimates (median values) for each of the four cases. Convergence

Each case has a unique set of parameter estimates, but share some similarities with each other. For example, cases 1a and 1c share identical growth rates and near-identical values for interspecies interactions, but have completely different intraspecies values. Given the assumptions of cases 1a and 1c, this suggests that intraspecies interactions are highly sensitive to interspecies interactions, but not the other way round. In the context of our system, higher intraspecies values correspond to larger stable population sizes (in the case that $(\Lambda_{11}, 0)$ and/or $(0, \Lambda_{22})$ are stable). This means that the presence of other species

(in co-culture) improves a microbes potential growth and population size in the case of competition, that is, if one species out-competes the other, it will grow to sizes much greater than if it were to be grown in isolation (mono-culture).

Furthermore, it is demonstrated that interaction terms are particularly sensitive to changes in a microbe's growth rate (which makes sense given the formula of Λ_{ij}). In cases 1b and 1d the mono-culture growth rates are re-estimated and a change in their values is observed. This implies that the presence of other microbial species directly affects the growth rate of a given species. This might be interpreted as the microbial species adapting themselves to the introduction of neighbouring species. A changing growth rate is the result of interspecies interactions, but also influences them! Take 1a and 1b. Both cases have identical intraspecies terms, but some variations to μ_1 and μ_2 in 1b result in significant changes to the interspecies values, namely Λ_{21}. Overall, due to the non-linear nature of our reparameterised gLV equations, parameter estimates in co-culture are sensitive to each other with some terms being more influenced by certain terms than others.

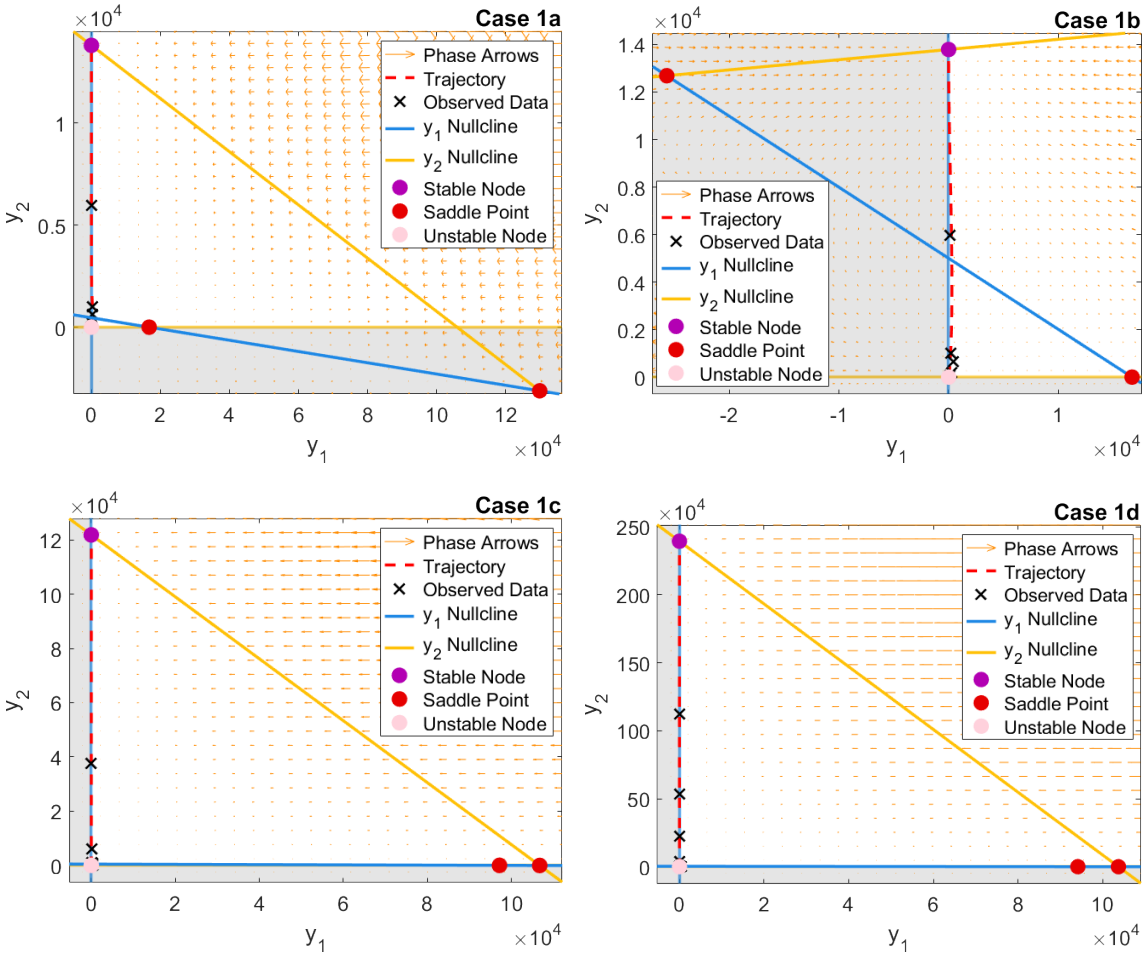

Figure 6: Phase planes illustrating the dynamics behind co-culture behaviour. In all 4 of the cases, the trajectory begins at $(1, 1)$ and converges towards the steady state $(0, \Lambda_{22})$. The primary difference in dynamics is the values for which the trajectory converges to. This is controlled by the parameter estimate of Λ_{22}. Additionally, the position of the non-zero equilibrium changes case-to-case, sometimes occurring at biologically infeasible values (Case 1b).

Visualising the dynamics using phase planes, we find each co-culture case to possess 2 unstable saddle points, an unstable node (at the origin, does not change) and a stable fixed point. What is interesting is that the stable node is no longer located at the non-zero intersection of y_1 and y_2 nullclines, but is rather situated at $(0, \Lambda_{22})$. Depending on the

case of interest, the position of the non-zero equilibrium point varies, but regardless of position remains an unstable saddle point. This switch in stability leads us to believe that a transcritical bifurcation has occurred, that is, two equilibria have collided and exchanged stability. Furthermore, no equilibria were created or destroyed, which is uncharacteristic of a saddle-node bifurcation. We can verify of the identity of the bifurcation that occurs at $\Lambda_{22} = \Lambda_{12}$ using bifurcation analysis later on. Now consider the time series solution of the gLV system against observed co-culture data.

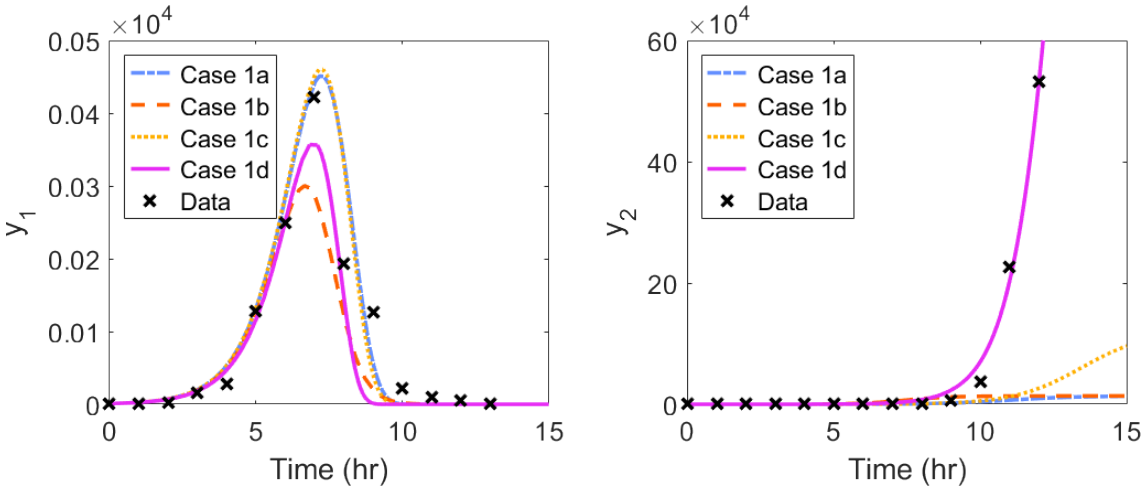

Figure 7: Time-Series solutions using median values of the posterior distributions.

Figure 6 displays the time series solutions $y_1(t)$ and $y_2(t)$ against the observed y_1 and y_2 co-culture, respectively. In general, the y_1 (*S. aureus* co-culture) data was better described, with all 4 of the cases capturing the general essence of the data. The y_2 (*P. aeruginosa* co-culture) data is equally well described until around $t = 10$ hours, where 3 of the 4 cases fail to capture the sudden increase in population size. The sudden decrease in y_1 and increase in y_2 is the result of a change in dynamics. This scenario corresponds closely with the observations commonly made in CF patients, in which patients often experience a co-existence of *S. aureus* and *P. aeruginosa* before the latter dominates. In the case of our observational data, *P. aeruginosa* outcompetes *S. aureus* leaving it with too few resources to survive and hence it goes extinct.

Comparing the mono-culture parameter estimates to case 1d in which all estimate are re-estimated, we find that both the species growth rate μ_i and intraspecies interactions Λ_{ii} change with the introduction of an interacting microbial species. In most of the cases, *S. aureus* had a greater effect on *P.aeruginosa* (Λ_{21} term) than it did on its own species (Λ_{11}). It is also worth noting that when the growth rate did change it decreased in the case of *S. aureus*, but increased in the case of *P. aeruginosa*. This illustrates *P. aeruginosa's* excellent ability to adapt to its surrounding environment.

3.2 Dynamical Systems Analysis - Transcritical Bifurcation

By implementing our gLV system into MATCONT, a numerical continuation software for MATLAB, we can decipher what bifurcations are occurring in the system. First, suppose a state similar to that of our mono-culture system, where $(\Lambda_{11}, 0)$ and $(0, \Lambda_{22})$ are saddle nodes (i.e., $\Lambda_{12} > \Lambda_{22}$ and $\Lambda_{21} > \Lambda_{11}$). We can select the non zero equilibria to be our initial point. Having initialised the model at a fixed point, we can now vary our parameter of interest in time (e.g. let Λ_{12} be the parameter we are varying). Decreasing Λ_{12} until $\Lambda_{12} = \Lambda_{22}$ we encounter a branch point which is another term for a zero-equilibrium bifurcation. From here the stability of the non-zero equilibrium goes from stable-to-unstable, inferring that the stability has switched with the equilibria situated at $(0, \Lambda_{22})$. This is a transcritical

bifurcation. These bifurcations can be found anywhere along the y_1 and y_2 axis as long as the correct conditions are satisfied (see Figure 8).

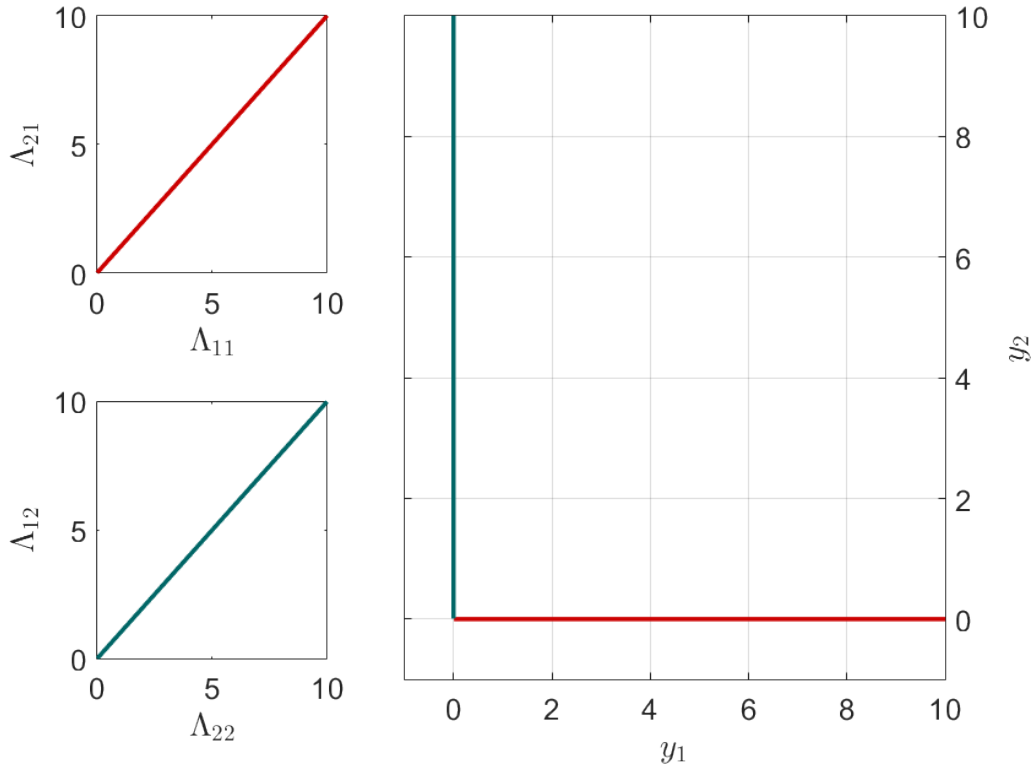

Figure 8: Curves of transcritical bifurcations. As demonstrated by the smaller subplots, these transcritical bifurcations occur along the y_1 and y_2 axis when $\Lambda_{11} = \Lambda_{21}$ and $\Lambda_{22} = \Lambda_{12}$, respectively. Transcritical bifurcations describe a change in system dynamics in which two fixed points exchange stability. If one of the aforementioned conditions is met, the non-zero equilibrium point will collide with $(\Lambda_{11}, 0)$ or $(0, \Lambda_{22})$.

Further variation of system parameters (incl. μ_i and Λ_{ii}) did not reveal any additional bifurcations. So overall, given the constraints on our reparameterised system the mechanisms underlying our system dynamics are relatively straight forward.

4 Discussion

Based on the findings outlined in the previous section and despite the critiques outlined in our introduction, the gLV equations successfully explain the experimental data, which undergoes a change in dynamics due to a transcritical bifurcation occurring at some critical value. Our results directly quantify the effects of interspecies interactions on a microbe's growth rate and intraspecies interactions, finding that *P. aeruginosa's* growth is boosted by the interspecies interactions from *S. aureus* (Λ_{21}). Similarly, the dynamics of the system are dependant on the values of Λ_{11} relative to Λ_{21} and Λ_{22} relative to Λ_{11}. Given that the experimental data is in line with the common consensus that *P. aeruginosa* dominates, these results are of some significance. For example, *P. aeruginosa* dominating is a major cause of morbidity in CF patients. Now understanding that a transcritical bifurcation occurs which leads *P. aeruginosa* to dominate, one can research methods to prevent this switch in dynamics from occurring. This might include limiting *P. aeruginosa's* outgoing and internal interactions or attempting to isolate the two species from each other.

Limitations of this study include our model being an oversimplification of real life dynamics. In reality, scenarios in which *S. aureus* and *P. aeruginosa* only interact with each are rare, with microbial communities usually being very diverse in different microbial species.

Furthermore, the gLV model, as pointed out by various scholars including Momeni et al., does not account for indirect interactions (e.g. interactions via metabolites). Updating our model to account for this (e.g. mechanistic model) may reveal completely different mechanisms underpinning the system dynamics. This said, the gLV model is definitely a great start in understanding the relationship between *S. aureus* and *P. aeruginosa*. The model is lightweight to compute, some components can be investigated analytically and successfully quantifies direct interactions between species.

4.1 Future Work

Much can be done to build on this existing work. Firstly, one should aim to validate the results of this report by investigating interaction dynamics using another model (one that isn't the gLV model). Given the low dimensions, this could be a pairwise model or a simple mechanistic model. Such a model should be fit to the experimental data and dynamical systems analysis run for comparison to this report's results. Secondly, the introduction of a perturbation term, like is featured in Stein et al., would enable us to understand how *S. aureus* and *P. aeruginosa* interact in and react to small dosages of different antimicrobial. Given the already apparent resistance of these two pathogens (more formidable in the case of co-culture), understanding how to combat their growth in mono-culture and co-culture is desirable. A perturbation term would most likely introduce new dynamics to the system so would warrant some sophisticated bifurcation analysis. These are just a couple of directions in which this line of work can be taken.

4.2 Summary of Key Points/Results

- Interspecies interactions between *S. aureus* and *P. aeruginosa* result in modifications to their rate of growth and the manner in which they interact with their own species (intraspecies interactions).
- The switch in dynamics is explained by the occurrence of a transcritical bifurcation. The bifurcation occurs when *P. aeruginosa* has an equal effect on itself and on *S. aureus*. The point marks the transition from both species co-existing, to *P. aeruginosa* out-competing *S. aureus* and dominating.
- Understanding that a transcritical bifurcation underlies the switch in model dynamics is potentially useful for many biomedical applications, including polymicrobial infections in CF patients, where *P. aeruginosa* often times dominates as the most prevalent pathogen.

5 References

[1] Song HS, Cannon WR, Beliaev AS, Kanopka A. Mathematical Modeling of Microbial Community Dynamics: A Methodological Review. Processes. 2014; 2(4): 711-752

[2] Berg G, et al. Microbiome definition re-visited: old concepts and new challenges. Microbiome. 2020; 8(103)

[3] Stark LA. Beneficial Microorganisms: Countering Microbephobia. CBE Life Sci Educ. 2010; 9(4): 387-389

[4] Murray CJL, et al. Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis. Lancet. 2022; 399(10325): 629–655

[5] Neely AN, Holder IA. Antimicrobial resistance. Burns. 1999; 25(1): 17-24

[6] Iglewski BH. Pseudomonas. Medical Microbiology. 4th edition. Galveston (TX): University of Texas Medical Branch at Galveston; 1996. Chapter 27. Available from: https://www.ncbi.nlm.nih.gov/books/NBK8326/

[7] Chambers HF, DeLeo FR. Waves of Resistance: Staphylococcus aureus in the Antibiotic Era. Nat Rev Microbiol. 2009; 7(9): 629–641

[8] Pang Z, et al. Antibiotic resistance in Pseudomonas aeruginosa: mechanisms and alternative therapeutic strategies. Biotechnol Adv. 2019; 37(1): 177-192

[9] Horcajada JP, et al. Epidemiology and Treatment of Multidrug-Resistant and Extensively Drug-Resistant Pseudomonas aeruginosa Infections. Clinical Microbiology Reviews. 2019; 32(9)

[10] Kluytmans J, Belkrum AV, Verbrugh H. Nasal carriage of Staphylococcus aureus: epidemiology, underlying mechanisms, and associated risks. Clin Microbiol Rev. 1997; 10(3): 505-520

[11] Liu GY. Molecular Pathogenesis of Staphylococcus aureus Infection. Pediatr Res. 2009; 65(5): 71-77

[12] O'Gara JP. Into the storm: Chasing the opportunistic pathogen Staphylococcus aureus from skin colonisation to life-threatening infections. Environmental Ecology. 2019; 19(10): 3823-3833

[13] Tong SYC, et al. Staphylococcus aureus Infections: Epidemiology, Pathophysiology, Clinical Manifestations, and Management. Clin Microbiol Rev. 2015; 28(3): 603–661

[14] Howden BP, et al. Staphylococcus aureus host interactions and adaptation. Nat Rev Microbiol. 2023.

[15] Wilson MG, Pandey S. Pseudomonas Aeruginosa. Treasure Island (FL): StatPearls Publishing; 2023. Available from: https://www.ncbi.nlm.nih.gov/books/NBK557831/

[16] Moradali MF, Ghods S, Rehm BHA. Pseudomonas aeruginosa Lifestyle: A Paradigm for Adaptation, Survival, and Persistence. Front Cell Infect Microbiol. 2017; 7: 39

[17] Rutherford ST, Bassler BL. Bacterial Quorum Sensing: Its Role in Virulence and Possibilities for Its Control. Cold Spring Harb Perspect Med. 2012; 2(11)

[18] Loc-Carrillo C, Abedon ST. Pros and cons of phage therapy. Bacteriophage. 2011; 1(2): 111–114

[19] Limoli DH, Hoffman LR. Help, hinder, hide and harm: what can we learn from the interactions between Pseudomonas aeruginosa and Staphylococcus aureus during respiratory infections? Thorax. 2019; 74: 684-692

[20] Braga RM, Dourado MN, Araujo WL. Microbial interactions: ecology in a molecular perspective. Braz J Microbiol. 2016; 47(1): 86–98

[21] Papenfort K, Bassler BL. Quorum-Sensing Signal-Response Systems in Gram-Negative Bacteria. Nat Rev Microbiol. 2016; 14(9): 576–588

[22] Cooper RM, Hasty J, Tsimring L. interspecies population dynamics enhance microbial horizontal gene transfer and spread of antibiotic resistance. eLife 6:e25950.

[23] Bottery MJ, Pitchford JW, Friman VP. Ecology and evolution of antimicrobial resistance in bacterial communities. ISME J. 2021; 15: 939–948

[24] Nguyen AT, Oglesby-Sherrouse AG. Interactions between Pseudomonas aeruginosa and Staphylococcus aureus during co-cultivations and polymicrobial infections. Appl Microbiol Biotechnol. 2016; 100(14): 6141–6148

[25] Alves PM, et al. Interaction between Staphylococcus aureus and Pseudomonas aeruginosa is beneficial for colonisation and pathogenicity in a mixed biofilm. Pathogens and Disease. 2018; 76(1)

[26] Rumpf C, et al. Staphylococcus aureus and Cystic Fibrosis—A Close Relationship. What Can We Learn from Sequencing Studies? Pathogens. 2021 Sep; 10(9): 1177

[27] Ahlgren HG, et al. Clinical outcomes associated with Staphylococcus aureus and Pseudomonas aeruginosa airway infections in adult cystic fibrosis patients. BMC Pulm Med. 2015; 15(67)

[28] Limoli DH, et al. Staphylococcus aureus and Pseudomonas aeruginosa co-infection is associated with cystic fibrosis-related diabetes and poor clinical outcomes. Eur J Clin Microbiol Infect Dis. 2016; 35: 947–953

[29] Filkins LM, O'Toole GA. Cystic Fibrosis Lung Infections: Polymicrobial, Complex, and Hard to Treat. PLOS Pathogens. 2015; 11(12)

[30] Yung DBY, Sircombe KJ, Pletzler D. Friends or enemies? The complicated relationship between Pseudomonas aeruginosa and Staphylococcus aureus. Molecular Biology. 2021; 116(1): 1-15

[31] Briaud P, et al. Most of Staphylococcus aureus and Pseudomonas aeruginosa co-infecting isolates coexist, a condition that may impact clinical outcomes in Cystic Fibrosis patients. medRxiv. 2020

[32] DeLeon S, et al. Synergistic Interactions of Pseudomonas aeruginosa and Staphylococcus aureus in an In Vitro Wound Model. Infect Immun. 2014; 82(11): 4718–4728

[33] Zhao K, et al. Pseudomonas aeruginosa Quorum-Sensing and Type VI Secretion System Can Direct Interspecific Coexistence During Evolution. Front. Microbiol. 2018; 9

[34] Wade MJ, et al. Perspectives in mathematical modelling for microbial ecology. Ecological Modelling. 2016; 321: 64-74

[35] Monod J. Recherches Sur La Croissance Des Cultures Bactériennes. 1st ed. Paris: Hermann, 1942. Print. Actualités Scientifiques Et Industrielles; 911. Available at: https://solo.bodleian.ox.ac.uk/permalink/f/n28kah/oxfaleph015108733

[36] Brunner JD, Chia N. Metabolite-mediated modelling of microbial community dynamics captures emergent behaviour more effectively than species–species modelling. J R Soc Interface. 2019; 16(159)

[37] Momeni B, Xie L, Shou W. Lotka-Volterra pairwise modelling fails to capture diverse pairwise microbial interactions. eLife 6:e25051.

[38] Stein RR, et al. Ecological Modeling from Time-Series Inference: Insight into Dynamics and Stability of Intestinal Microbiota. PLoS Comput Biol. 2013; 9(12)

[39] Steady-state reduction of generalized Lotka-Volterra systems in the microbiome Eric W. Jones* and Jean M. Carlson

[40] On the Use of Cauchy Prior Distributions for Bayesian Logistic Regression Joyee Ghosh, Yingbo Li, and Robin Mitra

6 Appendix

6.1 Experimental Data

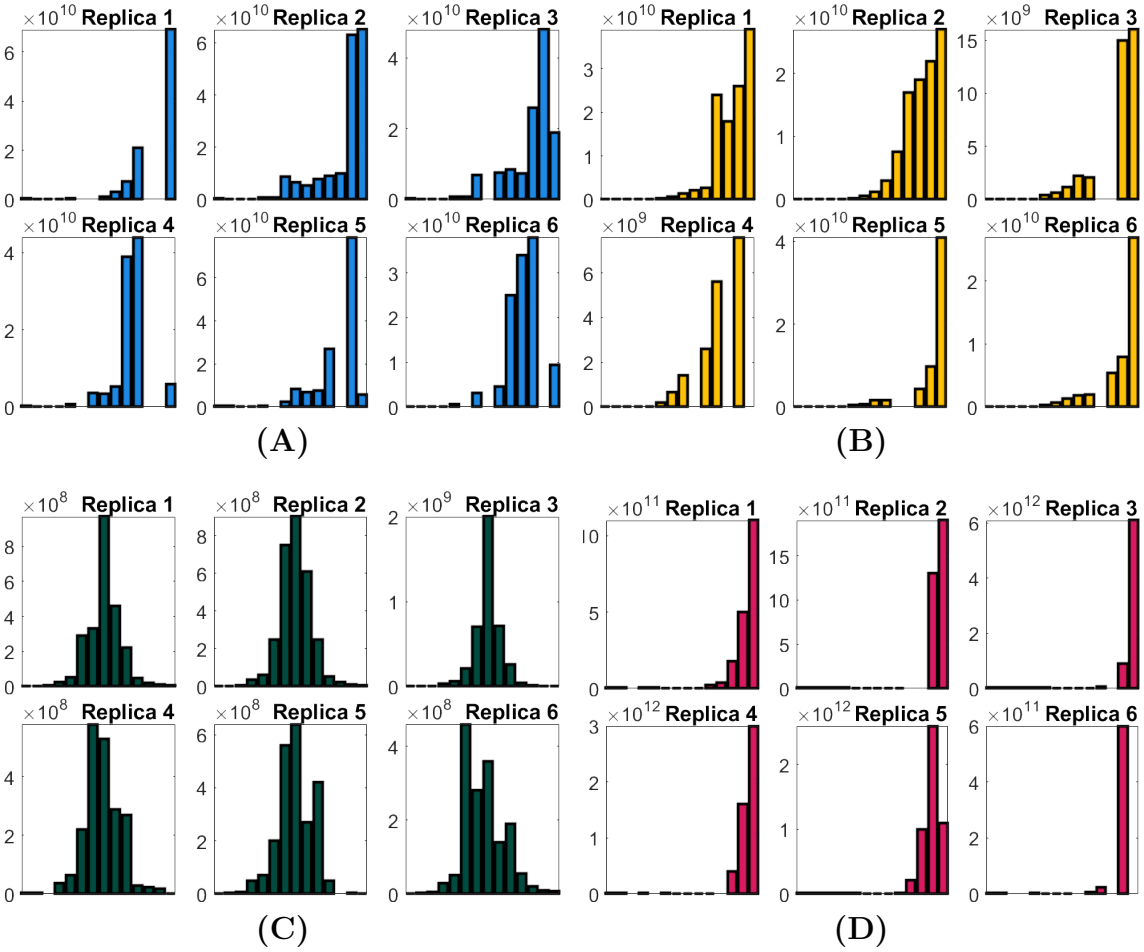

Figure 9: Raw experimental data for each species, culture-type and replicate, $0 \leq t \leq 13$, $t \in \mathbb{Z}$. **(A)**: *S. aureus* mono-culture, **(B)**: *P. aeruginosa* mono-culture, **(C)**: *S. aureus* co-culture, **(D)**: *P. aeruginosa* co-culture.

Table 10: Mean-averaged (non-normalised) experimental mono and co-culture data for *S. aureus* and *P. aeruginosa*.

Time (hr)	**S.A. Mono**	**P.A. Mono**	**S.A. Co**	**P.A. Co**
	Bacteria Count (ml^{-1})			
0	2550000	2116667	2133333	2350000
1	3400000	1883333	3166667	2416667
2	12750000	8550000	6260000	6733333
3	58666667	18666667	34500000	17400000
4	6.17×10^{08}	42166667	60333333	22400000
5	7.95×10^{08}	3.20×10^{08}	2.72×10^{08}	1.45×10^{08}
6	4.98×10^{09}	6.20×10^{08}	5.33×10^{08}	4.47×10^{08}
7	4.85×10^{09}	1.37×10^{09}	9.00×10^{08}	1.50×10^{09}
8	5.50×10^{09}	2.16×10^{09}	4.13×10^{08}	2.34×10^{09}
9	1.58×10^{10}	3.40×10^{09}	2.68×10^{08}	1.40×10^{10}
10	2.38×10^{10}	1.55×10^{10}	47500000	8.83×10^{10}
11	2.47×10^{10}	1.17×10^{10}	21000000	5.30×10^{11}
12	6.33×10^{10}	1.47×10^{10}	9833333	1.25×10^{12}
13	2.90×10^{10}	3.00×10^{10}	3166667	2.64×10^{12}

Table 11: Normalised, mean-averaged experimental mono and co-culture data for *S. aureus* and *P. aeruginosa*.

Time (hr)	**S.A. Mono**	**P.A. Mono**	**S.A. Co**	**P.A. Co**
	Bacteria Count (ml^{-1})			
0	1.00	1.00	1.00	1.00
1	1.33	0.89	1.48	1.03
2	5.00	4.04	2.93	2.87
3	23.01	8.82	16.17	7.40
4	241.83	19.92	28.28	9.53
5	311.76	151.18	127.34	61.70
6	1952.94	292.91	250.00	190.07
7	1901.96	645.67	421.86	639.01
8	2156.86	1020.47	193.75	995.74
9	6209.15	1606.30	125.78	5957.45
10	9313.73	7338.58	22.27	37553.19
11	9673.20	5515.75	9.84	225531.90
12	24836.60	6960.63	4.60	531914.90
13	11379.08	14173.23	1.48	1123404.00

6.2 Results of Parameter Estimation

Parameter	Rhat	Mean	SD	2.5%	50%	97.5%
μ_1	1.00	1.01	0.06	0.91	1.01	1.14
μ_2	1.00	0.88	0.06	0.77	0.87	1.01
Λ_{11}	1.00	16905.37	3464.62	10853.09	16724.89	23979.05
Λ_{22}	1.00	18185.56	142280.55	8439.40	13790.53	38250.25
Λ_{12}	1.00	460.10	27.57	402.98	460.64	513.59
Λ_{21}	1.00	810952.98	31754398.78	6914.74	105897.27	2783341.11
σ_1	1.00	3814.94	919.30	2496.38	3658.98	6050.01
σ_2	1.00	1754.41	446.56	1125.11	1675.17	2844.01
$\sigma_{1,2}$	1.00	40.15	9.63	26.41	38.50	63.47
$\sigma_{2,2}$	1.00	381953.52	84650.98	258560.39	368267.57	584338.09

Table 12: Case 1a parameter estimates.

Parameter	Rhat	Mean	SD	2.5%	50%	97.5%
μ_1	1.00	0.99	0.02	0.95	0.99	1.03
μ_2	1.00	1.25	0.11	0.99	1.28	1.41
Λ_{11}	1.00	16905.37	3464.62	10853.09	16724.89	23979.05
Λ_{22}	1.00	18185.56	142280.55	8439.40	13790.53	38250.25
Λ_{12}	1.00	4685.05	1687.61	1054.99	5002.28	7287.78
Λ_{21}	1.01	-290738.60	8032270.34	-9286424.86	-321331.69	8859114.87
σ_1	1.00	3814.94	919.30	2496.38	3658.98	6050.01
σ_2	1.00	1754.41	446.56	1125.11	1675.17	2844.01
$\sigma_{1,2}$	1.00	33.32	9.28	20.51	31.53	56.13
$\sigma_{2,2}$	1.00	379506.37	83577.08	255055.29	366188.93	576885.23

Table 13: Case 1b parameter estimates.

Parameter	Rhat	Mean	SD	2.5%	50%	97.5%
μ_1	1.00	1.01	0.06	0.91	1.01	1.14
μ_2	1.00	0.88	0.06	0.77	0.87	1.01
Λ_{11}	1.00	801659.59	42867042.76	4080.88	97130.57	2422735.67
Λ_{22}	1.00	880337.56	42024528.02	3838.21	121839.35	3314129.03
Λ_{12}	1.00	461.29	36.48	393.29	460.18	534.71
Λ_{21}	1.00	1069752.45	102022049.43	7376.16	106665.19	2578864.48
σ_1	1.00	3814.94	919.30	2496.38	3658.98	6050.01
σ_2	1.00	1754.41	446.56	1125.11	1675.17	2844.01
$\sigma_{1,2}$	1.00	40.73	9.84	26.62	39.08	64.56
$\sigma_{2,2}$	1.00	369997.86	81805.40	249323.32	357074.59	565971.85

Table 14: Case 1c parameter estimates.

Parameter	**Rhat**	**Mean**	**SD**	**2.5%**	**50%**	**97.5%**
μ_1	1.00	0.97	0.03	0.92	0.97	1.02
μ_2	1.01	1.12	0.01	1.11	1.12	1.14
Λ_{11}	1.00	664456.92	18279139.75	2075.67	94155.04	2646016.54
Λ_{22}	1.00	2409325.10	183388.64	2098628.41	2394258.29	2818050.92
Λ_{12}	1.00	2908.99	688.49	2174.94	2813.77	4132.61
Λ_{21}	1.00	884655.09	54540160.68	7606.93	103731.30	2582665.03
σ_1	1.00	3814.94	919.30	2496.38	3658.98	6050.01
σ_2	1.00	1754.41	446.56	1125.11	1675.17	2844.01
$\sigma_{1,2}$	1.00	44.29	11.19	28.61	42.29	71.83
$\sigma_{2,2}$	1.00	14920.15	3772.20	9648.37	14235.07	24141.48

Table 15: Case 1d parameter estimates.

6.3 Miscellaneous Simulation Results

Parameter	Rhat	Mean	SD	2.5%	50%	97.5%
μ_1	1.00	0.91	0.01	0.89	0.91	0.93
μ_2	1.00	0.94	0.04	0.86	0.94	1.03
Λ_{11}	1.00	28835.40	938.43	27086.31	28800.88	30802.31
Λ_{22}	1.00	18175.46	6015.18	12859.57	17663.69	26147.10
Λ_{12}	-	-	-	-	-	-
Λ_{21}	-	-	-	-	-	-
σ_1	1.00	404.93	148.94	226.42	372.33	776.91
σ_2	1.00	2174.91	542.23	1406.48	2079.73	3492.03

Table 16: Replica 1 Mono-Culture Parameter Estimates

Parameter	Rhat	Mean	SD	2.5%	50%	97.5%
μ_1	1.00	0.86	0.04	0.80	0.86	0.95
μ_2	1.00	0.96	0.02	0.93	0.96	1.00
Λ_{11}	1.00	61407.55	4205698.08	18170.05	33287.12	93328.72
Λ_{22}	1.00	12637.21	600.97	11477.23	12626.88	13872.62
Λ_{12}	-	-	-	-	-	-
Λ_{21}	-	-	-	-	-	-
σ_1	1.00	3013.36	784.87	1928.72	2870.15	4941.73
σ_2	1.00	626.45	157.51	404.12	598.56	1008.88

Table 17: Replica 2 Mono-Culture Parameter Estimates.

Parameter	Rhat	Mean	SD	2.5%	50%	97.5%
μ_1	1.00	1.00	0.09	0.88	0.99	1.17
μ_2	1.00	0.85	0.02	0.81	0.85	0.89
Λ_{11}	1.00	16878.29	10393.67	8920.19	16604.13	26101.17
Λ_{22}	1.00	9162.08	710.90	7973.57	9098.26	10735.09
Λ_{12}	-	-	-	-	-	-
Λ_{21}	-	-	-	-	-	-
σ_1	1.00	4756.32	1203.83	3047.31	4543.57	7694.02
σ_2	1.00	358.20	105.95	218.22	337.16	621.18

Table 18: Replica 3 Mono-Culture Parameter Estimates.

Parameter	Rhat	Mean	SD	2.5%	50%	97.5%
μ_1	1.00	1.23	1.47	0.88	1.17	1.79
μ_2	1.00	0.85	0.02	0.81	0.85	0.90
Λ_{11}	1.00	16058.61	23166.15	2809.40	15629.94	30336.55
Λ_{22}	1.00	3318.82	304.51	2757.60	3310.00	3929.10
Λ_{12}	-	-	-	-	-	-
Λ_{21}	-	-	-	-	-	-
σ_1	1.00	10141.04	2666.19	6166.74	9722.90	16459.51
σ_2	1.00	172.62	57.36	100.35	160.48	317.69

Table 19: Replica 4 Mono-Culture Parameter Estimates.

Parameter	Rhat	Mean	SD	2.5%	50%	97.5%
μ_1	1.00	1.04	0.50	0.82	1.01	1.35
μ_2	1.00	0.77	0.03	0.74	0.76	0.83
Λ_{11}	1.00	13125.21	46734.84	3153.39	12193.25	24099.04
Λ_{22}	1.00	1224539.73	54444317.50	19123.90	152428.87	4180103.57
Λ_{12}	-	-	-	-	-	-
Λ_{21}	-	-	-	-	-	-
σ_1	1.00	6828.93	1715.99	4355.96	6536.96	10975.74
σ_2	1.00	2073.82	729.06	1199.64	1908.15	3980.73

Table 20: Replica 5 Mono-Culture Parameter Estimates.

Parameter	Rhat	Mean	SD	2.5%	50%	97.5%
μ_1	1.00	1.10	0.27	0.93	1.08	1.40
μ_2	1.00	0.74	0.01	0.73	0.74	0.77
Λ_{11}	1.00	8490.99	4216.86	4216.18	8416.49	12956.78
Λ_{22}	1.00	1153731.53	23063339.45	28761.63	163773.34	4562906.83
Λ_{12}	-	-	-	-	-	-
Λ_{21}	-	-	-	-	-	-
σ_1	1.00	3287.91	984.21	1965.72	3096.01	5748.73
σ_2	1.00	1042.79	307.34	638.11	982.53	1801.04

Table 21: Replica 6 Mono-Culture Parameter Estimates.

6.4 Stability Criterion for $(y_1^*, y_2^*) = \left(\frac{(\Lambda_{12}-\Lambda_{22})\Lambda_{11}\Lambda_{21}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}, \frac{(\Lambda_{21}-\Lambda_{11})\Lambda_{12}\Lambda_{22}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)$

Substituting $(y_1, y_2) = \left(\frac{(\Lambda_{12}-\Lambda_{22})\Lambda_{11}\Lambda_{21}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}, \frac{(\Lambda_{21}-\Lambda_{11})\Lambda_{12}\Lambda_{22}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)$ into the Jacobian gives:

$$J = \begin{bmatrix} \mu_1\left(1 - 2\left(\frac{(\Lambda_{12}-\Lambda_{22})\Lambda_{21}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right) - \left(\frac{(\Lambda_{21}-\Lambda_{11})\Lambda_{22}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)\right) & -\mu_1\left(\frac{(\Lambda_{12}-\Lambda_{22})\Lambda_{11}\Lambda_{21}}{\Lambda_{12}(\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22})}\right) \\ -\mu_2\left(\frac{(\Lambda_{21}-\Lambda_{11})\Lambda_{12}\Lambda_{22}}{\Lambda_{21}(\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22})}\right) & \mu_2\left(1 - 2\left(\frac{(\Lambda_{21}-\Lambda_{11})\Lambda_{12}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right) - \left(\frac{(\Lambda_{12}-\Lambda_{22})\Lambda_{11}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)\right) \end{bmatrix} \tag{51}$$

By expanding each entry of the Jacobian, we find terms to cancel and the characteristic matrix to be:

$$\begin{bmatrix} \mu_1\left(\frac{\Lambda_{21}(\Lambda_{22}-\Lambda_{12})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right) - \lambda & \mu_1\left(\frac{\Lambda_{21}(\Lambda_{22}-\Lambda_{12})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)\left(\frac{\Lambda_{11}}{\Lambda_{12}}\right) \\ \mu_2\left(\frac{\Lambda_{12}(\Lambda_{11}-\Lambda_{21})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)\left(\frac{\Lambda_{22}}{\Lambda_{21}}\right) & \mu_2\left(\frac{\Lambda_{12}(\Lambda_{11}-\Lambda_{21})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right) - \lambda \end{bmatrix} \tag{52}$$

Now, taking the determinant of the characteristic matrix and simplifying yields the following characteristic polynomial,

$$\lambda^2 + \lambda\underbrace{\left(\frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22}) + \mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)}_{b} +$$
$$\underbrace{\mu_1\mu_2\left(\frac{\Lambda_{12}\Lambda_{21}(\Lambda_{21}-\Lambda_{11})(\Lambda_{12}-\Lambda_{22})}{(\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22})^2}\right)\left(1 - \frac{\Lambda_{11}\Lambda_{22}}{\Lambda_{12}\Lambda_{21}}\right)}_{c} = 0 \tag{53}$$

Cleaning up the c component,

$$\lambda^2 + \lambda\underbrace{\left(\frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22}) + \mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)}_{b} +$$
$$\underbrace{\mu_1\mu_2\left(\frac{\Lambda_{12}\Lambda_{21}(\Lambda_{21}-\Lambda_{11})(\Lambda_{12}-\Lambda_{22})}{(\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22})^2}\right)\left(\frac{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}{\Lambda_{12}\Lambda_{21}}\right)}_{c} = 0 \tag{54}$$

The final bracket in the third term cancels out and we are left with the following characteristic polynomial,

$$\lambda^2 + \lambda\underbrace{\left(\frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22}) + \mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)}_{b} + \underbrace{\mu_1\mu_2\left(\frac{(\Lambda_{21}-\Lambda_{11})(\Lambda_{12}-\Lambda_{22})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)}_{c} = 0 \tag{55}$$

The characteristic polynomial takes the form of a quadratic function. We can therefore use the quadratic formula to help diagnose the stability of this equilibrium point,

$$\lambda = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \tag{56}$$

where $a = 1$. Hence, the eigenvalues are solutions of the following form,

$$\lambda = \frac{1}{2}\left(-b \pm \sqrt{b^2 - 4c}\right). \tag{57}$$

We can consider three separate cases: $b^2 > 4c$, $b^2 = 4c$ and $b^2 < 4c$. Both $b^2 > 4c$ and $b^2 = 4c$ have strictly real solutions to Eq. (24). However, when $b^2 < 4c$, we can expect our eigenvalues to have an imaginary (complex) component. We can write b^2 and $4c$ in terms of Λ_{ij} and μ_i; firstly b^2,

$$b^2 = \frac{\mu_1^2\Lambda_{21}^2(\Lambda_{12} - \Lambda_{22})^2 + 2\mu_1\mu_2\Lambda_{12}\Lambda_{21}(\Lambda_{12} - \Lambda_{22})(\Lambda_{21} - \Lambda_{11}) + \mu_2^2\Lambda_{12}^2(\Lambda_{21} - \Lambda_{11})^2}{(\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22})^2}, \tag{58}$$

and now $4c$,

$$4c = 4\mu_1\mu_2\left(\frac{(\Lambda_{21} - \Lambda_{11})(\Lambda_{12} - \Lambda_{22})}{\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22}}\right). \tag{59}$$

Furthermore, we consider $\sqrt{b^2 - 4c}$,

$$\sqrt{b^2 - 4c} = \left[\frac{\mu_1^2\Lambda_{21}^2(\Lambda_{12} - \Lambda_{22})^2 + 2\mu_1\mu_2\Lambda_{12}\Lambda_{21}(\Lambda_{12} - \Lambda_{22})(\Lambda_{21} - \Lambda_{11})}{(\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22})^2}\right.$$
$$\left. + \frac{\mu_2^2\Lambda_{12}^2(\Lambda_{21} - \Lambda_{11})^2 - 4\mu_1\mu_2(\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22})(\Lambda_{21} - \Lambda_{11})(\Lambda_{12} - \Lambda_{22})}{(\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22})^2}\right]^{\frac{1}{2}}. \tag{60}$$

We can simplify by cancelling one of the terms for,

$$\sqrt{b^2 - 4c} = \left[\frac{\mu_1^2\Lambda_{21}^2(\Lambda_{12} - \Lambda_{22})^2 + \mu_2^2\Lambda_{12}^2(\Lambda_{21} - \Lambda_{11})^2}{(\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22})^2} + \dots\right.$$
$$\left. - \frac{2\mu_1\mu_2(\Lambda_{12}\Lambda_{21} - 2\Lambda_{11}\Lambda_{22})(\Lambda_{21} - \Lambda_{11})(\Lambda_{12} - \Lambda_{22})}{(\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22})^2}\right]^{\frac{1}{2}}. \tag{61}$$

We can also consider the cases of $b^2 > 4c$, $b^2 = 4c$ and $b^2 < 4c$ in terms of Λ_{ij} and μ_i. Take $b^2 > 4c$ for example, we substitute in our values for b and c for,

$$\frac{\mu_1^2\Lambda_{21}^2(\Lambda_{12} - \Lambda_{22})^2 + 2\mu_1\mu_2\Lambda_{12}\Lambda_{21}(\Lambda_{12} - \Lambda_{22})(\Lambda_{21} - \Lambda_{11}) + \mu_2^2\Lambda_{12}^2(\Lambda_{21} - \Lambda_{11})^2}{(\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22})^2}$$
$$> 4\mu_1\mu_2\left(\frac{(\Lambda_{21} - \Lambda_{11})(\Lambda_{12} - \Lambda_{22})}{\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22}}\right). \tag{62}$$

Multiplying both sides by $(\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22})^2$,

$$\mu_1^2\Lambda_{21}^2(\Lambda_{12} - \Lambda_{22})^2 + 2\mu_1\mu_2\Lambda_{12}\Lambda_{21}(\Lambda_{12} - \Lambda_{22})(\Lambda_{21} - \Lambda_{11}) + \mu_2^2\Lambda_{12}^2(\Lambda_{21} - \Lambda_{11})^2$$
$$> 4\mu_1\mu_2(\Lambda_{12}\Lambda_{21} - \Lambda_{11}\Lambda_{22})(\Lambda_{21} - \Lambda_{11})(\Lambda_{12} - \Lambda_{22}) \tag{63}$$

Cancelling common terms we get,

$$\mu_1^2\Lambda_{21}^2(\Lambda_{12}-\Lambda_{22})^2+\mu_2^2\Lambda_{12}^2(\Lambda_{21}-\Lambda_{11})^2-\mu_1\mu_2(2\Lambda_{12}\Lambda_{21}>4\Lambda_{11}\Lambda_{22})(\Lambda_{21}-\Lambda_{11})(\Lambda_{12}-\Lambda_{22}). \tag{64}$$

Dividing both sides through by $(\Lambda_{12}-\Lambda_{22})$ and $(\Lambda_{21}-\Lambda_{11})$,

$$\mu_1^2\Lambda_{21}^2\frac{(\Lambda_{12}-\Lambda_{22})}{(\Lambda_{21}-\Lambda_{11})}+\mu_2^2\Lambda_{12}^2\frac{(\Lambda_{21}-\Lambda_{11})}{(\Lambda_{12}-\Lambda_{22})}>2\mu_1\mu_2(\Lambda_{12}\Lambda_{21}-2\Lambda_{11}\Lambda_{22}). \tag{65}$$

Factorise for the following inequality,

$$\left(\mu_1^2\Lambda_{21}^2+\frac{1}{\mu_2^2\Lambda_{12}^2}\right)\left(\frac{\Lambda_{12}-\Lambda_{22}}{\Lambda_{21}-\Lambda_{11}}\right)>2\mu_1\mu_2(\Lambda_{12}\Lambda_{21}-2\Lambda_{11}\Lambda_{22}). \tag{66}$$

Some further tinkering brings about the following,

$$\frac{\alpha(\Lambda_{12}-\Lambda_{22})}{\Lambda_{12}\Lambda_{21}-2\Lambda_{11}\Lambda_{22}}>2\beta(\Lambda_{21}-\Lambda_{11}), \tag{67}$$

where $\alpha=\left(\mu_1^2\mu_2^2\Lambda_{12}^2\Lambda_{21}^2+1\right)>0$ and $\beta=\mu_1\mu_2^3\Lambda_{12}^2>0,\ \ \forall\Lambda_{12},\Lambda_{21}\in\mathbb{R}\setminus\{0\}$.
We can begin classifying the stability of $(y_1,y_2)=\left(\frac{(\Lambda_{12}-\Lambda_{22})\Lambda_{11}\Lambda_{21}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}},\frac{(\Lambda_{21}-\Lambda_{11})\Lambda_{12}\Lambda_{22}}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)$ with the three possible cases mentioned above,

$$\text{Stability}=\begin{cases}\frac{\alpha(\Lambda_{12}-\Lambda_{22})}{\Lambda_{12}\Lambda_{21}-2\Lambda_{11}\Lambda_{22}}>2\beta(\Lambda_{21}-\Lambda_{11}) & \lambda_1\neq\lambda_2 \text{ and } \lambda\in\mathbb{R}\\ \frac{\alpha(\Lambda_{12}-\Lambda_{22})}{\Lambda_{12}\Lambda_{21}-2\Lambda_{11}\Lambda_{22}}=2\beta(\Lambda_{21}-\Lambda_{11}) & \lambda_1=\lambda_2 \text{ and } \lambda\in\mathbb{R}\\ \frac{\alpha(\Lambda_{12}-\Lambda_{22})}{\Lambda_{12}\Lambda_{21}-2\Lambda_{11}\Lambda_{22}}<2\beta(\Lambda_{21}-\Lambda_{11}) & \lambda_1\neq\lambda_2 \text{ and } \lambda\in\mathbb{C}\end{cases} \tag{68}$$

Recalling that the eigenvalues of this equilibria assume the following form,

$$\lambda=\frac{1}{2}\left(-b\pm\sqrt{b^2-4c}\right), \tag{69}$$

we can now investigate the b of the characteristic polynomial and how it affects stability. Like with there being three distinct cases for b^2-4c, there are also three distinct cases for b, namely, $b>0$, $b=0$ and $b<0$. For example $b>0$,

$$b=\left(\frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}}\right)>0. \tag{70}$$

So, we now have three more conditions to classify stability with,

$$\text{Stability} = \begin{cases} \frac{\alpha(\Lambda_{12}-\Lambda_{22})}{\Lambda_{12}\Lambda_{21}-2\Lambda_{11}\Lambda_{22}} > 2\beta(\Lambda_{21}-\Lambda_{11}) & \begin{cases} \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} > 0 \\ \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} = 0 \\ \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} < 0 \end{cases} \\ \frac{\alpha(\Lambda_{12}-\Lambda_{22})}{\Lambda_{12}\Lambda_{21}-2\Lambda_{11}\Lambda_{22}} = 2\beta(\Lambda_{21}-\Lambda_{11}) & \begin{cases} \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} > 0 \\ \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} = 0 \\ \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} < 0 \end{cases} \\ \frac{\alpha(\Lambda_{12}-\Lambda_{22})}{\Lambda_{12}\Lambda_{21}-2\Lambda_{11}\Lambda_{22}} < 2\beta(\Lambda_{21}-\Lambda_{11}) & \begin{cases} \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} > 0 \\ \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} = 0 \\ \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} < 0 \end{cases} \end{cases} \tag{71}$$

We are able to exclude the possibility of the some stability types due to the constraints on our system's parameters, $\mu_1, \mu_2, \Lambda_{11}, \Lambda_{22} > 0$ and $\Lambda_{12}, \Lambda_{21} \in \mathbb{R} \setminus \{0\}$. Firstly, the only known instance for of a bifurcation occurring within our system thus far is when $\Lambda_{11} = \Lambda_{21}$ or $\Lambda_{12} = \Lambda_{22}$, in which a transcritical bifurcation occurs. In order for a Bogdanov-Taken (BT) bifurcation (co-dimension 2) to occur, two different curves of co-dimension 1 bifurcations must intersect. This would mean, in this case, a curve of transcritical bifurcations intersecting with a curve of some other co-dimension 1 bifurcation. So, consider the case when $\Lambda_{11} = \Lambda_{21}$, in order for a BT point to occur, $\lambda_1 = \lambda_2 = 0$, meaning that $b = b^2 - 4c = 0$. When $\Lambda_{11} = \Lambda_{21}$, b becomes,

$$b(\Lambda_{11} = \Lambda_{21}) = \frac{\mu_1\Lambda_{21}(\Lambda_{12} - \Lambda_{22}) + \mu_2\Lambda_{12}(\Lambda_{21} - \Lambda_{21})}{\Lambda_{12}\Lambda_{21} - \Lambda_{21}\Lambda_{22}}, \tag{72}$$

which simplifies down to,

$$b(\Lambda_{11} = \Lambda_{21}) = \frac{\mu_1\Lambda_{21}(\Lambda_{12} - \Lambda_{22})}{\Lambda_{21}(\Lambda_{12} - \Lambda_{22})} = \mu_1 \neq 0. \tag{73}$$

Therefore, in the case of $b^2 - 4c = 0$, $b \neq 0 \implies \lambda_1 = \lambda_2 = \frac{1}{2}(-b) \neq 0$, and so a BT point cannot occur for this system.

$$\text{Stability} = \begin{cases} \frac{\alpha(\Lambda_{12}-\Lambda_{22})}{\Lambda_{12}\Lambda_{21}-2\Lambda_{11}\Lambda_{22}} > 2\beta(\Lambda_{21}-\Lambda_{11}) & \begin{cases} \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} > 0 \\ \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} = 0 \\ \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} < 0 \end{cases} \\ \frac{\alpha(\Lambda_{12}-\Lambda_{22})}{\Lambda_{12}\Lambda_{21}-2\Lambda_{11}\Lambda_{22}} = 2\beta(\Lambda_{21}-\Lambda_{11}) & \begin{cases} \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} > 0 \\ \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} < 0 \end{cases} \\ \frac{\alpha(\Lambda_{12}-\Lambda_{22})}{\Lambda_{12}\Lambda_{21}-2\Lambda_{11}\Lambda_{22}} < 2\beta(\Lambda_{21}-\Lambda_{11}) & \begin{cases} \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} > 0 \\ \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} = 0 \\ \frac{\mu_1\Lambda_{21}(\Lambda_{12}-\Lambda_{22})+\mu_2\Lambda_{12}(\Lambda_{21}-\Lambda_{11})}{\Lambda_{12}\Lambda_{21}-\Lambda_{11}\Lambda_{22}} < 0 \end{cases} \end{cases} \tag{74}$$

In the case of $b^2 > 4c$ both values of λ are real and unique, and can be classified into stable, unstable, (unstable) saddle and bifurcation points.

6.5 Code - RStudio

```
1
2  # Connor Tynan - 2023
3
4  # Bayesian methods account for the uncertainty that is present in our ...
       observed data.
5  # Stan enables one to call powerful sampling methods. Create Stan ...
       script -> Call in R/Python
6  # Key ingredients for Bayesian models:
7  # Process Model:     Modelling the underlying process
8  # Measurement Model: Want to model/measure how error in our model is ...
       distributed
9  # Parameter Model:   The prior distribution. Specifying our beliefs ...
       about the variables.
10 #                    Beliefs are updated from sampling the posterior
11
12 #### Preamble ####
13
14 rm(list = ls())  # Creating fresh workspace (Deletes current workspace)
15
16 # Libraries
17 library(dplyr)
18 library(tidyr)
19 library(ggplot2)
20 library(ggmcmc)
21
22 # Initializing Stan packages
23 library(rstan)
24 options(mc.cores = parallel::detectCores()) # Options for RStan
25 rstan_options(auto_write = TRUE)            # Options for RStan
26 library(bayesplot)
27 library("shinystan")
28
29 # Miscellaneous / QOL
30 library(formatR) # Formatting code
31
32 # Importing data
33 library(readxl)
34
35 #### Data ####
36
37 # Importing Excel Data
38 # This script is only interested in the co-culture data
39
40 # Staphylococcus Aureus Co-Culture
41 sa_co_df <-
42   read_excel(
43     "Staph-Pseudomonas-Data.xlsx",
44     sheet = "S. aureus co-culture",
45     skip = 1
46   )
47
48 # Pseudomonas Aeruginosa Co-Culture
49 pa_co_df <-
50   read_excel(
51     "Staph-Pseudomonas-Data.xlsx",
52     sheet = "P. aeruginosa co-culture",
53     skip = 1
54   )
55
56 # Parsing the data
57 # Renaming columns for clarity
58 col_names = c('time', 'rep1', 'rep2', 'rep3', 'rep4', 'rep5', 'rep6')
```

```
colnames(sa_co_df)[which(names(sa_co_df) == "Time (h)")] <-
  "time"
names(sa_co_df)[1:7] <- col_names
colnames(pa_co_df)[which(names(pa_co_df) == "Time (h)")] <-
  "time"
names(pa_co_df)[1:7] <- col_names

## Using averaged replicate data (across all 6 replicates)
sa_co_df$mean <- rowMeans(sa_co_df[,2:7], na.rm=TRUE)
sa_co_df[,2:7] <- NULL

## Using averaged replicate data (across all 6 replicates)
pa_co_df$mean <- rowMeans(pa_co_df[,2:7], na.rm=TRUE)
pa_co_df[,2:7] <- NULL

####

# Normalising the data - Dividing through by initial time population
pa_co_df[, 2] = pa_co_df[, 2] / (filter(pa_co_df, time == 0) %>% ...
    select(-time) %>% unlist)
sa_co_df[, 2] = sa_co_df[, 2] / (filter(sa_co_df, time == 0) %>% ...
    select(-time) %>% unlist)

#### Parameter Estimation ####

# Preparing information for parameter estimation

# Initial time
t0 = 0.0
# Remaining times
ts = filter(sa_co_df, time > 0) %>% select(time) %>% unlist
# Growth data
z = filter(sa_co_df,time>0) %>% select(-time)
z = cbind(z,filter(pa_co_df,time>0) %>% select(-time))
# Initial population size
y0 = c(sa_co_df[1,2],pa_co_df[1,2])
y0 <- as.numeric(unlist(y0))

# Number of samples
nSamples = nrow(sa_co_df) - 1
# Number of replicates
nReplicates = 2

# Initialising Stan Model
gLV_co <-
  stan_model("gLV_Case_1c.stan", model_name = "ConnorModel")

# MCMC estimation
# 50/50 Train/test split
estimates <- sampling(
  object = gLV_co,
  data = list (
    T  = nSamples,
    nReplicates = nReplicates,
    y0 = y0,
    z  = z[, 1:nReplicates],
    t0 = t0,
    ts = ts
  ),
  seed = 13371,
  chains = 15,
  iter = 25000,
  warmup = 12500,
  init_r = 2,
  show_messages = TRUE
)
```

```
123
124 #### Results ####
125
126 # Specifying type of and printing results
127 parametersToPlot = c("theta", "sigma", "lp__")
128 print(estimates, pars = parametersToPlot)
129
130 # Initialising ShinyStan visualisation library
131 launch_shinystan(estimates)
```

6.6 Code - RStan

6.6.1 Mono-Culture Case

```
// Stan file generating gLV Stan model
// Case 1 - For generating growth rate and inter-interactivity ... paramaters using Mono-Culture data

// Function block:
functions {
  real[] logisticgrowth(real t,
                        real[] y,
                        real[] theta,
                        real[] x_r,
                        int[] x_i
  ) {
    real dydt[x_i[1]];
    for (i in 1:x_i[1]){
      dydt[i] = theta[1] * y[i] * (1-y[i]/theta[2]);
    }
    return dydt;
  }
}

// The input data is a vector 'y' of length 'N'.
// Data block: Information conditioned upon. Lower = Lower bound.

data {
  int<lower=1> T;           // Number of total replicates
  int<lower=0> nReplicates; // Number of replicates to sample
  real y0[nReplicates];     // Initial Population size for replicates
  real z[T,nReplicates];    // Averaging across replicates?
  real t0;                  // Initial time
  real ts[T];               // Time step
}

// Transformed data
transformed data {
  real x_r[0];
  int x_i[1];
  x_i[1] = nReplicates;
}

// Accepted parameters to estimate (Interactivity terms [theta] and ... error est. [sigma])
parameters {
  real<lower=0> theta[2];
  real<lower=0> sigma;
}

// The model to be estimated. We model the output
// 'y' to be normally distributed with mean 'mu'
// and standard deviation 'sigma'.
model {
  real y_hat[T,nReplicates];
  theta[1] ¬ cauchy(1,0.01);   // Vague choice of priors
  theta[2] ¬ cauchy(100000,1000); // Vague choice of priors
  sigma ¬ normal(0,10000);    // Vague choice of priors
  y_hat = integrate_ode_rk45(logisticgrowth, y0, t0, ts, theta, x_r, x_i);
  for (t in 1:T) {
    for (i in 1:nReplicates) {
      z[t,i] ¬ normal(y_hat[t,i], sigma);
    }
  }
```

```
59 }
60
61 // Generated Quantities
62 generated quantities{
63   real y_pred[T,nReplicates];
64   real z_pred[T,nReplicates];
65   y_pred = integrate_ode_rk45(logisticgrowth, y0, t0, ts, theta, x_r, ...
         x_i );
66   for (t in 1:T) {
67     for(i in 1:nReplicates){
68       z_pred[t,i] = y_pred[t,i] + normal_rng(0,sigma);
69     }
70   }
71 }
```

6.6.2 Case 1a

```
1      // Stan file generating gLV Stan model
2 // Case 1a - For generating intra-interactivity parameters
3
4 // Function block:
5 functions {
6   real[] logisticgrowth(real t,
7                         real[] y,
8                         real[] theta,
9                         real[] x_r,
10                        int[] x_i
11  ) {
12
13    real dudt = (1 - y[1]/16724.89 - y[2]/theta[1]) * y[1] * 1.01 ;
14    real dvdt =  (1 - y[2]/13790.53 - y[1]/theta[2]) * y[2] * 0.87 ;
15
16    return { dudt, dvdt };
17  }
18 }
19
20 // The input data is a vector 'y' of length 'N'.
21 // Data block: Information conditioned upon. Lower = Lower bound.
22
23 data {
24   int<lower=1> T;           // Number of total replicates
25   int<lower=0> nReplicates; // Number of replicates to sample
26   real y0[nReplicates];     // Initial Population size for replicates
27   real z[T,nReplicates];    // Averaging across replicates?
28   real t0;                  // Initial time
29   real ts[T];               // Time step
30 }
31
32 // Transformed data
33 transformed data {
34   real x_r[0];
35   int x_i[1];
36   x_i[1] = nReplicates;
37 }
38
39 // Accepted parameters to estimate (Interactivity terms [theta] and ...
       error est. [sigma])
40 parameters {
41   real theta[2];
42   real<lower=0> sigma[2];
43 }
44
45 // The model to be estimated. We model the output
```

```
// 'y' to be normally distributed with mean 'mu'
// and standard deviation 'sigma'.
model {
  real y_hat[T,nReplicates];
  theta[1] ¬ cauchy(0,100000); // Vague choice of priors
  theta[2] ¬ cauchy(0,100000); // Vague choice of priors
  sigma[1] ¬ normal(0,1000000);
  sigma[2] ¬ normal(0,1000000);
  y_hat = integrate_ode_rk45(logisticgrowth, y0, t0, ts, theta, x_r, x_i);
  for (t in 1:T) {
    for (i in 1:nReplicates) {
      z[t,i] ¬ normal(y_hat[t,i], sigma[i]);
    }
  }
}

//
generated quantities{
  real y_pred[T,nReplicates];
  real z_pred[T,nReplicates];
  y_pred = integrate_ode_rk45(logisticgrowth, y0, t0, ts, theta, x_r, ...
      x_i );
  for (t in 1:T) {
    for(i in 1:nReplicates){
      z_pred[t,i] = y_pred[t,i] + normal_rng(0,sigma[i]);
    }
  }
}
```

6.6.3 Case 1b

```
    // Stan file generating gLV Stan model
// Case 1b - For generating growth intra-interactivity and ...
    re-estimating growth parameters

// Function block:
functions {
  real[] logisticgrowth(real t,
                        real[] y,
                        real[] theta,
                        real[] x_r,
                        int[] x_i
  ) {

    real dudt = (1 - y[1]/16724.89 - y[2]/theta[3]) * y[1] * theta[1] ;
    real dvdt =  (1 - y[2]/13790.53 - y[1]/theta[4]) * y[2] * theta[2] ;

    return { dudt, dvdt };
  }
}

// The input data is a vector 'y' of length 'N'.
// Data block: Information conditioned upon. Lower = Lower bound.

data {
  int<lower=1> T;           // Number of total replicates
  int<lower=0> nReplicates; // Number of replicates to sample
  real y0[nReplicates];     // Initial Population size for replicates
  real z[T,nReplicates];    // Averaging across replicates?
  real t0;                  // Initial time
  real ts[T];               // Time step
}

```

```
// Transformed data
transformed data {
  real x_r[0];
  int x_i[1];
  x_i[1] = nReplicates;
}

// Accepted parameters to estimate (Interactivity terms [theta] and ...
    error est. [sigma])
parameters {
  real theta[4];
  real<lower=0> sigma[2];
}

// The model to be estimated. We model the output
// 'y' to be normally distributed with mean 'mu'
// and standard deviation 'sigma'.
model {
  real y_hat[T,nReplicates];
  theta[1] ¬ cauchy(1,0.01);     // Vague choice of priors
  theta[2] ¬ cauchy(1,0.01);     // Vague choice of priors
  theta[3] ¬ cauchy(0,100000); // Vague choice of priors
  theta[4] ¬ cauchy(0,100000); // Vague choice of priors
  sigma[1] ¬ normal(0,1000000);
  sigma[2] ¬ normal(0,1000000);
  y_hat = integrate_ode_rk45(logisticgrowth, y0, t0, ts, theta, x_r, x_i);
  for (t in 1:T) {
    for (i in 1:nReplicates) {
      z[t,i] ¬ normal(y_hat[t,i], sigma[i]);
    }
  }
}

//
  generated quantities{
    real y_pred[T,nReplicates];
    real z_pred[T,nReplicates];
    y_pred = integrate_ode_rk45(logisticgrowth, y0, t0, ts, theta, x_r, ...
        x_i );
    for (t in 1:T) {
      for(i in 1:nReplicates){
        z_pred[t,i] = y_pred[t,i] + normal_rng(0,sigma[i]);
      }
    }
  }
```

6.6.4 Case 1c

```
    // Stan file generating gLV Stan model
// Case 1c - For generating intra-interactivity and re-estimating ...
    inter-interactivity parameters

// Function block:
functions {
  real[] logisticgrowth(real t,
                        real[] y,
                        real[] theta,
                        real[] x_r,
                        int[] x_i
  ) {

    real dudt = (1 - y[1]/theta[1] - y[2]/theta[3]) * y[1] * 1.01 ;
    real dvdt =  (1 - y[2]/theta[2] - y[1]/theta[4]) * y[2] * 0.87 ;
```

```
15
16      return { dudt, dvdt };
17    }
18  }
19
20  // The input data is a vector 'y' of length 'N'.
21  // Data block: Information conditioned upon. Lower = Lower bound.
22
23  data {
24    int<lower=1> T;           // Number of total replicates
25    int<lower=0> nReplicates; // Number of replicates to sample
26    real y0[nReplicates];     // Initial Population size for replicates
27    real z[T,nReplicates];    // Averaging across replicates?
28    real t0;                  // Initial time
29    real ts[T];               // Time step
30  }
31
32  // Transformed data
33  transformed data {
34    real x_r[0];
35    int x_i[1];
36    x_i[1] = nReplicates;
37  }
38
39  // Accepted parameters to estimate (Interactivity terms [theta] and ...
        error est. [sigma])
40  parameters {
41    real theta[4];
42    real<lower=0> sigma[2];
43  }
44
45  // The model to be estimated. We model the output
46  // 'y' to be normally distributed with mean 'mu'
47  // and standard deviation 'sigma'.
48  model {
49    real y_hat[T,nReplicates];
50    theta[1] ¬ cauchy(100000,1000); // Vague choice of priors
51    theta[2] ¬ cauchy(100000,1000); // Vague choice of priors
52    theta[3] ¬ cauchy(0,100000); // Vague choice of priors
53    theta[4] ¬ cauchy(0,100000); // Vague choice of priors
54    sigma[1] ¬ normal(0,1000000);
55    sigma[2] ¬ normal(0,1000000);
56    y_hat = integrate_ode_rk45(logisticgrowth, y0, t0, ts, theta, x_r, x_i);
57    for (t in 1:T) {
58      for (i in 1:nReplicates) {
59        z[t,i] ¬ normal(y_hat[t,i], sigma[i]);
60      }
61    }
62  }
63
64  //
65    generated quantities{
66      real y_pred[T,nReplicates];
67      real z_pred[T,nReplicates];
68      y_pred = integrate_ode_rk45(logisticgrowth, y0, t0, ts, theta, x_r, ...
            x_i );
69      for (t in 1:T) {
70        for(i in 1:nReplicates){
71          z_pred[t,i] = y_pred[t,i] + normal_rng(0,sigma[i]);
72        }
73      }
74    }
```

6.6.5 Case 1d

```
    // Stan file generating gLV Stan model
// Case 1d - Estimating all parameters simultaneously using co-culture

// Function block:
functions {
  real[] logisticgrowth(real t,
                        real[] y,
                        real[] theta,
                        real[] x_r,
                        int[] x_i
  ) {

    real dudt = (1 - y[1]/theta[2] - y[2]/theta[3]) * y[1] * theta[1] ;
    real dvdt =  (1 - y[2]/theta[5] - y[1]/theta[6]) * y[2] * theta[4] ;

    return { dudt, dvdt };
  }
}

// The input data is a vector 'y' of length 'N'.
// Data block: Information conditioned upon. Lower = Lower bound.

data {
  int<lower=1> T;           // Number of total replicates
  int<lower=0> nReplicates; // Number of replicates to sample
  real y0[nReplicates];     // Initial Population size for replicates
  real z[T,nReplicates];    // Averaging across replicates?
  real t0;                  // Initial time
  real ts[T];               // Time step
}

// Transformed data
transformed data {
  real x_r[0];
  int x_i[1];
  x_i[1] = nReplicates;
}

// Accepted parameters to estimate (Interactivity terms [theta] and ...
    error est. [sigma])
parameters {
  real theta[6];
  real<lower=0> sigma[2];
}

// The model to be estimated. We model the output
// 'y' to be normally distributed with mean 'mu'
// and standard deviation 'sigma'.
model {
  real y_hat[T,nReplicates];
  theta[1] ¬ cauchy(1,0.01);    // Vague choice of priors
  theta[2] ¬ cauchy(100000,1000); // Vague choice of priors
  theta[3] ¬ cauchy(0,100000); // Vague choice of priors
  theta[4] ¬ cauchy(1,0.01);    // Vague choice of priors
  theta[5] ¬ cauchy(100000,1000); // Vague choice of priors
  theta[6] ¬ cauchy(0,100000); // Vague choice of priors
  sigma[1] ¬ normal(0,1000000);
  sigma[2] ¬ normal(0,1000000);
  y_hat = integrate_ode_rk45(logisticgrowth, y0, t0, ts, theta, x_r, x_i);
  for (t in 1:T) {
    for (i in 1:nReplicates) {
      z[t,i] ¬ normal(y_hat[t,i], sigma[i]);
    }
  }
}
```

```
65
66  //
67  generated quantities{
68    real y_pred[T,nReplicates];
69    real z_pred[T,nReplicates];
70    y_pred = integrate_ode_rk45(logisticgrowth, y0, t0, ts, theta, x_r, ...
          x_i );
71    for (t in 1:T) {
72      for(i in 1:nReplicates){
73        z_pred[t,i] = y_pred[t,i] + normal_rng(0,sigma[i]);
74      }
75    }
76  }
```

6.7 Code - MATLAB

6.7.1 Plotting Data Barchart

```
% Connor Tynan - Visualising S.aureus and P.aerugniosa mono and co-culture data.

% NOTE: Requires tight_subplot.m and barwitherr.m functions from the public mathworks file exchange

% Miscellaneous
clc; clf; close all; clear

% Importing the data
[mean_mono_sa, mean_mono_pa, mean_co_sa, mean_co_pa] = importmeandata;
[std_mono_sa, std_mono_pa, std_co_sa, std_co_pa] = importmeanstddata;
[mono_sa, mono_pa, co_sa, co_pa] = importrawdata;

% Figure 1 - Mean-averaged with error bars
fig = figure(1);

% Initialising 2x2 subplot
set(gca,'Visible','off')
ha = tight_subplot(2,2,[.125 .065],[.075 .075],[.05 .05]);

% S.aureus mono-culture
axes(ha(1));
% Bar chart with error bars
barwitherr(cat(3,zeros(14,1),std_mono_sa),0:1:13,mean_mono_sa,...
    'linewidth',2,'FaceColor','#1E88E5');
% Plot aesthetics
xlabel('Time (hr)'); ylabel('Bacteria Count (ml^{-1})')
LimitsX = xlim; LimitsY = ylim;
title('S.aureus Mono-Culture','HorizontAlalignment', 'right', ...
    'position', [LimitsX(2), LimitsY(2)])
set(gca,'FontSize',12)

% P.aeruginosa mono-culture
axes(ha(2));
% Bar chart with error bars
barwitherr(cat(3,zeros(14,1),std_mono_pa),0:1:13,mean_mono_pa,...
    'linewidth',2,'FaceColor','#FFC107');
% Plot aesthetics
xlabel('Time (hr)'); ylabel('Bacteria Count (ml^{-1})')
LimitsX = xlim; LimitsY = ylim;
title('P.aeruginosa Mono-Culture','HorizontAlalignment', 'right', ...
    'position', [LimitsX(2), LimitsY(2)])
set(gca,'FontSize',12)

% S.aureus co-culture
axes(ha(3));
% Bar chart with error bars
barwitherr(cat(3,zeros(14,1),std_co_sa),0:1:13,mean_co_sa,...
    'linewidth',2,'FaceColor','#004D40');
% Plot aesthetics
xlabel('Time (hr)'); ylabel('Bacteria Count (ml^{-1})')
LimitsX = xlim; LimitsY = ylim;
title('S.aureus Co-Culture','HorizontAlalignment', 'right', 'position', ...
    [LimitsX(2), LimitsY(2)])
set(gca,'FontSize',12)

% P.aeruginosa co-culture
axes(ha(4));
```

```
% Bar chart with error bars
barwitherr(cat(3,zeros(14,1),std_co_pa),0:1:13,mean_co_pa,...
    'linewidth',2,'FaceColor','#D81B60');
% Plot aesthetics
xlabel('Time (hr)'); ylabel('Bacteria Count (ml^{-1})')
LimitsX = xlim; LimitsY = ylim;
title('P.aeruginosa Co-Culture','HorizontAlalignment', 'right', ...
    'position', [LimitsX(2), LimitsY(2)])
set(gca,'FontSize',12)

% Export Settings
set(gcf, 'units', 'centimeters', 'position', [1 1 12.1*2 9*2]);


% Figure 2-5 - Visualising the raw data

% S.aureus mono-culture
figure(2)

% Initialising Subplot
set(gca,'Visible','off')
ha = tight_subplot(2,3,[.085 .065],[.07 .07],[.04 .04]);

% Plotting raw data
for i = 1:6
    axes(ha(i));
    % Bar chart with error bars
    bar(0:1:13,mono_sa(:,i),'LineWidth',2,'FaceColor','#1E88E5');
    axis tight
    % Plot aesthetics
    LimitsX = xlim; LimitsY = ylim;
    title(sprintf('Replica %d',i),'HorizontAlalignment', 'right', ...
        'position', [LimitsX(2), LimitsY(2)])
    xticks([])
    set(gca,'FontSize',14)
end

% P.aeruginosa mono-culture
figure(3)

% Initialising Subplot
set(gca,'Visible','off')
ha = tight_subplot(2,3,[.085 .065],[.07 .07],[.04 .04]);

% Plotting raw data
for i = 1:6
    axes(ha(i));
    % Bar chart with error bars
    bar(0:1:13,mono_pa(:,i),'LineWidth',2,'FaceColor','#FFC107');
    axis tight
    % Plot aesthetics
    LimitsX = xlim; LimitsY = ylim;
    title(sprintf('Replica %d',i),'HorizontAlalignment', 'right', ...
        'position', [LimitsX(2), LimitsY(2)])
    xticks([])
    set(gca,'FontSize',14)
end

% S.aureus co-culture
figure(4)

% Initialising Subplot
set(gca,'Visible','off')
ha = tight_subplot(2,3,[.085 .065],[.07 .07],[.04 .04]);

for i = 1:6
```

```
120     axes(ha(i));
121     % Bar chart with error bars
122     bar(0:1:13,co_sa(:,i),'LineWidth',2,'FaceColor','#004D40');
123     axis tight
124     % Plot aesthetics
125     LimitsX = xlim; LimitsY = ylim;
126     title(sprintf('Replica %d',i),'HorizontAlalignment', 'right', ...
            'position', [LimitsX(2), LimitsY(2)])
127     xticks([])
128     set(gca,'FontSize',14)
129 end
130
131 % P.aeruginosa co-culture
132 figure(5)
133
134 % Initialising Subplot
135 set(gca,'Visible','off')
136 ha = tight_subplot(2,3,[.085 .065],[.07 .07],[.04 .04]);
137
138 % Plotting raw data
139 for i = 1:6
140     axes(ha(i));
141     % Bar chart with error bars
142     bar(0:1:13,co_pa(:,i),'LineWidth',2,'FaceColor','#D81B60');
143     axis tight
144     % Plot aesthetics
145     LimitsX = xlim; LimitsY = ylim;
146     title(sprintf('Replica %d',i),'HorizontAlalignment', 'right', ...
            'position', [LimitsX(2), LimitsY(2)])
147     xticks([])
148     set(gca,'FontSize',14)
149 end
```

6.7.2 Mono-Culture Phase Planes + Time Series

```
% Connor Tynan - Visualising Mean-average Parameter Estimates
% We solve the gLV equations using MATLAB's in-built ODE45 solver.
% We can then visualise the time series and phase plane

% Miscellaneous
clc; clf; close all; clear;

% Importing mono-culture data
[mono_sa, mono_pa] = importrawdata;
[mono_sa_mean, mono_pa_mean] = importmeandata;

% Time parameters
t0 = 0; t1 = 25;

% Arrays for parameter estimates (median values)
mu1 = [0.91, 0.86, 0.99, 1.17, 1.01, 1.08];
mu2 = [0.94, 0.96, 0.85, 0.85, 0.76, 0.74];
L11 = [28800.88, 33287.12, 16604.13, 15629.94, 12193.25, 8416.49];
L22 = [17663.69, 12626.88, 9098.26, 3310.00, 152428.87, 163773.34];
L12 = zeros(1,6);
L21 = zeros(1,6);

% Parameters fit to mean data
mu1m = 1.01;
mu2m = 0.87;
L11m = 16724.89;
L22m = 13790.53;
L12m = 0;
L21m = 0;

% Parameter Estimates using Mean-averaged Mono-Culture Data
figure(1)

% Customising Figure Layout
ha = tight_subplot(1,1,[.125 .065],[.15 .06],[.25 .25]);
axes(ha);

hold on

% Highlighting biologically infeasible region
p1 = patch([1e5 -1e5 -1e5 1e5], [-1e5 -1e5, 0 0], [0.8 0.8 0.8], ...
    'FaceAlpha', 0.1, 'EdgeColor', 'none', 'FaceColor', [0 0 0]);
p2 = patch([0 -1e5 -1e5 0], [0 0, 1e5 1e5], [0.8 0.8 0.8], 'FaceAlpha', ...
    0.1, 'EdgeColor', 'none', 'FaceColor', [0 0 0]);

% ODEs to solve
Fm = @(t,y) [mu1m.*y(1)-mu1m.*y(1).*y(1)./L11m;
             mu2m.*y(2)-mu2m.*y(2).*y(2)./L22m];

% Solving the ODEs in time
[t,xts] = ode45(Fm, [t0 t1], [1 1]);

% Quiver Initial Conditions
X=linspace(L11m*0.03,L11m*0.97,25);
Y=linspace(L22m*0.03,L22m*0.97,25);

% Plotting Phase Arrows (See bottom of script)
pp_plot(X,Y,mu1m,mu2m,L11m,L22m)

% Quiver Initial Conditions - Lower Right Quadrant
X=linspace(L11m*1.03,L11m*1.97,25);
Y=linspace(L22m*0.03,L22m*0.97,25);
```

```
61
62 % Plotting Phase Arrows (See bottom of script)
63 pp_plot(X,Y,mu1m,mu2m,L11m,L22m)
64
65 % Quiver Initial Conditions - Upper Left Quadrant
66 X=linspace(L11m*0.03,L11m*0.97,25);
67 Y=linspace(L22m*1.03,L22m*1.97,25);
68
69 % Plotting Phase Arrows (See bottom of script)
70 pp_plot(X,Y,mu1m,mu2m,L11m,L22m)
71
72 % Quiver Initial Conditions - Upper Right Quadrant
73 X=linspace(L11m*1.03,L11m*1.97,25);
74 Y=linspace(L22m*1.03,L22m*1.97,25);
75
76 % Plotting Phase Arrows (See bottom of script)
77 pp_plot(X,Y,mu1m,mu2m,L11m,L22m)
78
79 % Plotting Phase Diagram
80 plot(xts(:,1),xts(:,2),'r--','linewidth',2);    % Trajectory
81 plot(mono_sa_mean,mono_pa_mean,'xk','MarkerSize',10,'LineWidth',1.5) % ...
       Observed Data
82 plot(0,0,'color','#1E88E5','linewidth',2)       % Dummy Nullcline for ...
       Legend
83 plot(0,0,'color','#FFC107','linewidth',2)       % Dummy Nullcline for ...
       Legend
84
85 % Equilibrium points
86 plot(0,L22m,'o','color',[0.9 0 0],'MarkerFaceColor',[0.9 0 ...
       0],'MarkerSize',10)
87 plot(L11m,0,'o','color',[0.9 0 0],'MarkerFaceColor',[0.9 0 ...
       0],'MarkerSize',10)
88 plot(L11m,L22m,'o','color',[0.7 0 0.7],'MarkerFaceColor',[0.7 0 ...
       0.7],'MarkerSize',10)
89 plot(0,0,'o','color','#FFD1DC','MarkerFaceColor','#FFD1DC','MarkerSize',10)
90
91 % y1 and y2 Nullclines
92 xh = xline(0,'color','#1E88E5','linewidth',2,'DisplayName','Nullcline');
93 xh2 = xline(L11m,'color','#1E88E5','linewidth',2);
94 xh3 = yline(0,'color','#FFC107','linewidth',2);
95 xh4 = yline(L22m,'color','#FFC107','linewidth',2);
96
97 % Formatting of Nullclines
98 Sxh = struct(xh);
99 Sxh.Edge.Layer = 'back';
100 Sxh = struct(xh2);
101 Sxh.Edge.Layer = 'back';
102 Sxh = struct(xh3);
103 Sxh.Edge.Layer = 'back';
104 Sxh = struct(xh4);
105 Sxh.Edge.Layer = 'back';
106
107 % Axis Aesthetics
108 Xlimit = xlim; Ylimit = ylim;
109 xlim([-L11m*0.1 L11m*1.1])
110 ylim([-L22m*0.1 L22m*1.1])
111
112 ax=gca; ax.XAxis.Exponent = 4; ax.YAxis.Exponent = 4;
113
114 % Legend
115 leg = legend('','','','','',' Phase Arrows',' Trajectory',' Observed ...
       Data',' y_{1} Nullcline',' y_{2} Nullcline', ...
116     ' Saddle Point','',' Stable Node',' Unstable Node','location','west');
117 leg.ItemTokenSize = [20,18];
118
119 % Plot aesthetics
```

```
LimitsX = xlim; LimitsY = ylim;
box on; xlabel('y_{1}'); ylabel('y_{2}')
set(gca,'FontSize',12)
set(gcf, 'units', 'centimeters', 'position', [1 1 12.1*2 9*1.35]);
% export_fig('Mean_Avg_Param_Est_Phase_Plane', gcf,...
    '-transparent', '-png', '-r125')

% Solving gLV equations in time with parameter estimated coefficients
% 1x2 Subplot showcasing S.aureus and P.aeruginosa trajectories against
% observed data

% Initialising new figure
fig2 = figure(2);

% Subplot 1: S.aureus
subplot(1,2,1); hold on; box on
% Plotting trajectory
plot(t,xts(:,1),'LineWidth',2,'color','#1E88E5')
% Plotting data for comparison/reference
plot(0:1:13,mono_sa_mean/mono_sa_mean(1),'kx','MarkerSize',8,'LineWidth',2)
% Axis
xlim([0 15])
xlabel('Time [hr]'); ylabel('y_{1}')
% Legend
leg = legend(' Trajectory',' Data','location','northwest');
leg.ItemTokenSize = [20,18];
% Plot Aesthetics
set(gca,'FontSize',12)
ax=gca; ax.YAxis.Exponent = 4;

% Subplot 2: P.aeruginosa
subplot(1,2,2); hold on; box on
% Plotting trajectory
plot(t,xts(:,2),'LineWidth',2,'color','#FFC107')
% Plotting data for comparison/reference
plot(0:1:13,mono_pa_mean/mono_pa_mean(1),'kx','MarkerSize',8,'LineWidth',2)
% Axis
xlim([0 15])
xlabel('Time [hr]'); ylabel('y_{2}')
% Legend
leg = legend(' Trajectory',' Data','location','northwest');
leg.ItemTokenSize = [20,18];
% Plot Aesthetics
set(gca,'FontSize',12)
ax=gca; ax.YAxis.Exponent = 4;
% Legend
leg = legend(' Trajectory',' Data','location','northwest');
leg.ItemTokenSize = [20,18];

% Figure Aesthetics
set(gcf, 'units', 'centimeters', 'position', [1 1 12.1*2 9]);
% export_fig('Mean_Avg_Param_Est_Time_Series', gcf, '-transparent', ...
    '-png', '-r125')

%%

% Function to plot phase plane
function q = pp_plot(X,Y,mu1m,mu2m,L11m,L22m)

    % ODEs to solve
    Fm = @(t,y) [mu1m.*y(1)-mu1m.*y(1).*y(1)./L11m;
                 mu2m.*y(2)-mu2m.*y(2).*y(2)./L22m];

    % Solving the ODEs in time
    [t,xts] = ode45(Fm, [0 13], [1 1]);

```

```
185     % Initialising Grid for Quiver - Lower Left Quadrant
186     [m,n] = meshgrid(X,Y);
187 
188     % Initialising Quiver Solution Vectors
189     u = zeros(size(m));
190     v = zeros(size(n));
191 
192     % Quiver Computation - Lower Left Quadrant
193 
194     for j = 1:numel(m)
195 
196         % Solving System of Equations
197         Yprime = Fm(t,[m(j); n(j)]);
198 
199         % Storing the solutions
200         u(j) = Yprime(1);
201         v(j) = Yprime(2);
202     end
203 
204     % Generating Quiver Plot --- Grid of points & Grid of solutions
205     q = quiver(m,n,u,v,'color',[255/255,140/255,0,0.6]);
206 
207 end
```

6.7.3 Co-Culture Phase Planes + Time Series

```
    % Connor Tynan - Visualising Mean-average Parameter Estimates
% Plotting the results of cases 1a-d

% Parameter Estimates using Mean-averaged Normalised Co-Culture Data

% Miscellaneous
clc; clf; close all; clear; hold on

% Importing mono and co-culture data
[mono_sa, mono_pa, co_sa, co_pa] = importrawdata;
[mono_sa_mean, mono_pa_mean, co_sa_mean, co_pa_mean] = importmeandata;

% Time parameters
t0 = 0;     % Initial time
t1 = 25;    % Final time

% Arrays for parameter estimates (median values)
mu1 = [1.01, 0.99, 1.01, 0.97];
mu2 = [0.87, 1.28, 0.87, 1.12];
L11 = [16724.89, 16724.89, 97130.57, 94155.04];
L22 = [13790.53, 13790.53, 121839.35, 2394258.29];
L12 = [460.64, 5002.28, 460.18, 2813.77];
L21 = [105897.27, -321331.69, 106665.19, 103731.30];

% Main computation
% - Plotting phase plane and time series of each population for each case

for i = 1:4

    % Initialising figure
    figure(i); hold on

    % Highlighting biologically infeasible region
    p1 = patch([1e8 -1e8 -1e8 1e8], [-1e8 -1e8, 0 0], [0.8 0.8 0.8], ...
        'FaceAlpha', 0.1, 'EdgeColor', 'none', 'FaceColor', [0 0 0]);
    p2 = patch([0 -1e8 -1e8 0], [0 0, 1e8 1e8], [0.8 0.8 0.8], ...
        'FaceAlpha', 0.1, 'EdgeColor', 'none', 'FaceColor', [0 0 0]);

    % ODEs to solve
    Fm = @(t,y) ...
        [mu1(i).*y(1)-mu1(i).*y(1).*y(1)./L11(i)-mu2(i).*y(1).*y(2)./L12(i);
                mu2(i).*y(2)- ...
                   mu2(i).*y(2).*y(2)./L22(i)-mu1(i).*y(1).*y(2)./L21(i)];

    % Solving the ODEs in time
    [t,xts] = ode45(Fm, [t0 t1], [1 1]);

    % Calculating non-zero equilibrium position
    xinter = (L12(i)-L22(i))*L11(i)*L21(i)/(L12(i)*L21(i)-L11(i)*L22(i));
    yinter = (L21(i)-L11(i))*L22(i)*L12(i)/(L12(i)*L21(i)-L11(i)*L22(i));

    % Quiver Initial Conditions
    X=linspace(max([L11(i) abs(xinter)])*0.02,max([L11(i) ...
        abs(xinter)])*1.05,25);
    Y=linspace(max([L22(i) yinter])*0.02,max([L22(i) yinter])*1.05,25);

    % Plotting Phase Arrows (See bottom of script)
    pp_plot(X,Y,mu1(i),mu2(i),L11(i),L22(i),L12(i),L21(i),i)

    % Quiver Initial Conditions - Lower Right Quadrant
    X=linspace(max([L11(i) xinter])*0.02,max([L11(i) xinter])*1.05,25);
    Y=-linspace(max([L22(i) yinter])*0.02,max([L22(i) yinter])*1.05,25);
```

```
59
60      % Plotting Phase Arrows (See bottom of script)
61      pp_plot(X,Y,mu1(i),mu2(i),L11(i),L22(i),L12(i),L21(i),i)
62
63      % Quiver Initial Conditions - Upper Left Quadrant
64      X=-linspace(max([L11(i) abs(xinter)])*0.02,max([L11(i) ...
            abs(xinter)])*1.05,25);
65      Y=linspace(max([L22(i) yinter])*0.02,max([L22(i) yinter])*1.05,25);
66
67      % Plotting Phase Arrows (See bottom of script)
68      pp_plot(X,Y,mu1(i),mu2(i),L11(i),L22(i),L12(i),L21(i),i)
69
70      % Quiver Initial Conditions - Upper Right Quadrant
71      X=-linspace(max([L11(i) xinter])*0.02,max([L11(i) xinter])*1.05,25);
72      Y=-linspace(max([L22(i) yinter])*0.02,max([L22(i) yinter])*1.05,25);
73
74      % Plotting Phase Arrows (See bottom of script)
75      pp_plot(X,Y,mu1(i),mu2(i),L11(i),L22(i),L12(i),L21(i),i)
76
77      % Plotting Phase Diagram
78      plot(xts(:,1),xts(:,2),'r--','linewidth',2);     % Trajectory
79      plot(co_sa_mean/co_sa_mean(1),co_pa_mean/co_pa_mean(1),'xk', ...
80          'MarkerSize',10,'LineWidth',1.5)             % Observed Data
81      plot(0,0,'color','#1E88E5','linewidth',2)        % Dummy Nullcline ...
            for Legend
82      plot(0,0,'color','#FFC107','linewidth',2)        % Dummy Nullcline ...
            for Legend
83
84      % Plotting nullclines
85      x = linspace(-1e5,1e6,10000);   % Dummy array
86
87      % Nullclines
88      plot(L11(i)*(1-x/L12(i)),x,'k','LineWidth',2,'color','#1E88E5');
89      plot(x,L22(i)*(1-x/L21(i)),'k','LineWidth',2,'color','#FFC107');
90
91      % Equilibrium points
92      plot(0,L22(i),'o','color',[0.7 0 0.7],'MarkerFaceColor',[0.7 0 ...
            0.7],'MarkerSize',10)
93      plot(L11(i),0,'o','color',[0.9 0 0],'MarkerFaceColor',[0.9 0 ...
            0],'MarkerSize',10)
94      plot(0,0,'o','color','#FFD1DC','MarkerFaceColor','#FFD1DC', ...
95          'MarkerSize',10)
96
97      % Non-zero equilibria
98      plot((L12(i)-L22(i))*L11(i)*L21(i)/(L12(i)*L21(i)-L11(i)*L22(i)), ...
99          (L21(i)-L11(i))*L22(i)*L12(i)/(L12(i)*L21(i)-L11(i)*L22(i)),'o', ...
                ...
100         'color',[0.9 0 0],'MarkerFaceColor',[0.9 0 0],'MarkerSize',10)
101
102     % Zero-axis nullclines
103     xh  = ...
            xline(0,'color','#1E88E5','linewidth',2,'DisplayName','Nullcline');
104     xh2 = yline(0,'color','#FFC107','linewidth',2);
105
106     % Formatting of Nullclines
107     Sxh = struct(xh);
108     Sxh.Edge.Layer = 'back';
109     Sxh = struct(xh2);
110     Sxh.Edge.Layer = 'back';
111
112     % Axis Aesthetics
113     Xlimit = xlim; Ylimit = ylim;
114     xlim([min([-0.5*10^4 xinter])*1.05 max([L11(i) xinter])*1.05])
115     ylim([min([L22(i)*-0.05 yinter])*1.05 max([L22(i) yinter])*1.05])
116
117     ax=gca; ax.XAxis.Exponent = 4; ax.YAxis.Exponent = 4;
```

```
118
119     % Legend
120     leg = legend('','','','','',' Phase Arrows',' Trajectory',' ...
            Observed Data',' y_{1} Nullcline',' y_{2} Nullcline', ...
121         '','',' Stable Node',' Saddle Point',' Unstable ...
                Node','location','best');
122     leg.ItemTokenSize = [20,18];
123
124     % Plot aesthetics
125     % Title
126     LimitsX = xlim; LimitsY = ylim;
127     cases = {'Case 1a', 'Case 1b', 'Case 1c', 'Case 1d'};
128     title(cases(i),'HorizontAlalignment', 'right', 'position', ...
            [LimitsX(2), LimitsY(2)])
129
130     box on; xlabel('y_{1}'); ylabel('y_{2}')
131     set(gca,'FontSize',14)
132     set(gcf, 'units', 'centimeters', 'position', [1 1 12.1*1.25 9*1.25]);
133     % export_fig('Mean_Avg_Param_Est_Phase_Plane', gcf, '-transparent', ...
            '-png', '-r125')
134
135     fig5 = figure(5); hold on
136
137     % Subplot 1: S.aureus
138     subplot(1,2,1); hold on; box on
139     % Plotting trajectory
140     if i == 1
141         pp1 = plot(t,xts(:,1),'-.','LineWidth',2,'color','#648FFF');
142     elseif i == 2
143         pp1 = plot(t,xts(:,1),'--','LineWidth',2,'color','#FE6100');
144     elseif i == 3
145         pp1 = plot(t,xts(:,1),':','LineWidth',2,'color','#FFB000');
146     else
147         pp1 = plot(t,xts(:,1),'-','LineWidth',2,'color','#E72BF5');
148     end
149     % Plotting data for comparison/reference
150     plot(0:1:13,co_sa_mean/co_sa_mean(1),'kx','MarkerSize',8,'LineWidth',2)
151     % Axis
152     xlim([0 15]); ylim([0 0.05*10^4])
153     xlabel('Time (hr)'); ylabel('y_{1}')
154     % Legend
155     leg = legend(' Case 1a','',' Case 1b','',' Case 1c','',' Case 1d',' ...
            Data', ...
156         'location','NorthWest');
157     leg.ItemTokenSize = [20,18];
158     % Plot Aesthetics
159     set(gca,'FontSize',14)
160     ax=gca; ax.YAxis.Exponent = 4;
161
162     % Subplot 2: P.aeruginosa
163     subplot(1,2,2); hold on; box on
164     % Plotting trajectory
165     if i == 1
166         pp2 = plot(t,xts(:,2),'-.','LineWidth',2,'color','#648FFF');
167     elseif i == 2
168         pp2 = plot(t,xts(:,2),'--','LineWidth',2,'color','#FE6100');
169     elseif i == 3
170         pp2 = plot(t,xts(:,2),':','LineWidth',2,'color','#FFB000');
171     else
172         pp2 = plot(t,xts(:,2),'-','LineWidth',2,'color','#E72BF5');
173     end
174     % Plotting data for comparison/reference
175     plot(0:1:13,co_pa_mean/co_pa_mean(1),'kx','MarkerSize',8,'LineWidth',2)
176     % Axis
177     xlim([0 15]); ylim([0 60*10^4])
178     xlabel('Time (hr)'); ylabel('y_{2}')
```

```
    % Legend
    leg = legend(' Trajectory',' Data','location','northwest');
    leg.ItemTokenSize = [20,18];
    % Plot Aesthetics
    set(gca,'FontSize',14)
    ax=gca; ax.YAxis.Exponent = 4;
    % Legend
    leg = legend(' Case 1a','',' Case 1b','',' Case 1c','',' Case 1d',' ...
        Data', ...
        'location','NorthWest');
    leg.ItemTokenSize = [20,18];

    % Figure Aesthetics
    set(gcf, 'units', 'centimeters', 'position', [1 1 12.1*2 9]);

end

% Solving gLV equations in time with parameter estimated coefficients
% 1x2 Subplot showcasing S.aureus and P.aeruginosa trajectories against
% observed data

% Initialising new figure
% export_fig('Mean_Avg_Param_Est_Time_Series', gcf, '-transparent', ...
    '-png', '-r125')

%%

% Function to plot phase plane
function q = pp_plot(X,Y,mu1,mu2,L11,L22,L12,L21,i)

    % ODEs to solve
    Fm = @(t,y) [mu1.*y(1)-mu1.*y(1).*y(1)./L11-mu2.*y(1).*y(2)./L12;
                 mu2.*y(2)-mu2.*y(2).*y(2)./L22-mu1.*y(1).*y(2)./L21];

    % Solving the ODEs in time
    [t,¬] = ode45(Fm, [0 13], [1 1]);

    % Initialising Grid for Quiver - Lower Left Quadrant
    [m,n] = meshgrid(X,Y);

    % Initialising Quiver Solution Vectors
    u = zeros(size(m));
    v = zeros(size(n));

    % Quiver Computation - Lower Left Quadrant

    for j = 1:numel(m)

        % Solving System of Equations
        Yprime = Fm(t,[m(j); n(j)]);

        % Storing the solutions
        u(j) = Yprime(1);
        v(j) = Yprime(2);
    end

    if i ≠ 4
        % Generating Quiver Plot --- Grid of points & Grid of solutions
        q = quiver(m,n,u,v,'color',[255/255,140/255,0,0.6], ...
            'AutoScaleFactor',1.2);
    elseif i == 4
        q = quiver(m,n,u,v,'color',[255/255,140/255,0,0.6], ...
            'AutoScaleFactor',0.1, ...
            'ShowArrowHead','on','AutoScale','on');
    end
end
```