2020

Exploring Heterogeneous Phenotypes in Response to Stress

Heather S. Deter
South Dakota State University

EXPLORING HETEROGENEOUS PHENOTYPES IN RESPONSE TO STRESS

BY

HEATHER S. DETER

A dissertation submitted in partial fulfillment of the requirements for the

Doctor of Philosophy

Major in Biological Sciences

Specialization in Microbiology

South Dakota State University

2020

DISSERTATION ACCEPTANCE PAGE

Heather S. Deter

This dissertation is approved as a creditable and independent investigation by a candidate for the Doctor of Philosophy degree and is acceptable for meeting the dissertation requirements for this degree. Acceptance of this does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Nicholas C. Butzin
Advisor Date

Heike Bucking
Department Head Date

Dean, Graduate School Date

Ἰησοῦ, ὅτι ἐξ αὐτοῦ καὶ δι’ αὐτοῦ καὶ εἰς αὐτὸν τὰ πάντα· αὐτῷ ἡ δόξα
εἰς τοὺς αἰῶνας.

ACKNOWLEDGEMENTS

This work was supported by funds from the National Science Foundation Division of Molecular and Cellular Biosciences, MCB-1330180 (Chapter 2) and the National Science Foundation award no. 1922542 (Chapter 4). As well as, the Hatch project grant no. SD00H653-18/project accession no. 1015687 from the USDA National Institute of Food and Agriculture (Chapters 4 and 4).

First, thank you to my parents (Paul and Donna), my sister Amber and my extended family for all their support. And thanks to all my brothers and sisters in Christ for their love and prayer.

Thank you to all of my friends and colleagues at SDSU and Virginia Tech who have supported me throughout this journey. With a special thanks to my lab mates Udaya, Anamul, Alawiah, Praj, Tahmina, Tanim and our undergraduate students Elise, Courtney, Chase and Mitch.

Thanks to all the professors who have taught and mentored me over the last several years. Especially, Dr. Will Mather for always answering my questions and all the kind advice.

A heartfelt thank you to my mentor and advisor, Dr. Nicholas Butzin for always believing in me and for being there when I needed him. Every single mile from Virginia to South Dakota was worth the journey.

And ultimately, thanks and praise to my Lord and Savior Jesus Christ.

CONTENTS

ABBREVIATIONS

Abbreviation	**Description**
ADEP4	acyldepsipeptide
Amp	ampicillin
AU	Arbitrary Units
CDC	Center for Disease Control
CDS	coding sequence
CFP	cyan fluorescent protein
CFU	colony forming units
Cip	ciprofloxacin
Cm	chloramphenicol
csv	Comma-separated values file format
FFT	Fast Fourier Transform
GUI	Graphic User Interface
IPTG	Isopropyl β-D-1-thiogalactopyranoside
LB	Lysogeny Broth-Miller’s media
Km	kanamycin
OD	optical density
RBS	ribosome binding site
Ribo-Seq	ribosome profiling
RNA-seq	RNA-sequencing
ROI	Region Of Interest
RPKM	reads per kilobase per million
TA	toxin-antitoxin
TIR	translation initiation rate

ABSTRACT

EXPLORING HETEROGENEOUS PHENOTYPES IN RESPONSE TO STRESS

HEATHER S. DETER

2020

This work combines traditional microbiology with bioinformatic and synthetic biology approaches to study antibiotic tolerance. Antibiotic tolerance is a widespread phenomenon that facilitates antibiotic resistance and decreases the effectiveness of antibiotic treatment. Tolerance is distinct from antibiotic resistance, because tolerance is short term survival and typically results from phenotypic variations rather than genetic variation.

The molecular mechanisms underlying tolerance are varied and debated in the literature. I have explored two intracellular processes related to tolerance, toxin-antitoxin (TA) systems (Chapter 2) and proteases (Chapter 4). Specifically, I focus on the ratio of antitoxin-to-toxin in type II TA systems, because type II TA systems must be regulated in such a way that antitoxins are more prevalent than their toxins. Our analysis of RNA-sequencing and ribosome profiling data demonstrates that most type II TA systems in *E. coli* are regulated at the translational level, while others rely on various combinations of transcriptional and post-transcriptional regulation. Before publishing this article, researchers often cited transcriptional regulation as the primary method of regulating TA systems.

Studying antibiotic tolerance and other subpopulations necessitates the ability to study single-cell dynamics in the context of the whole population. To facilitate single-cell analysis, we have developed single-cell tracking software that leverages machine learning to identify cells. The software then tracks the cell based on this classification and returns

data on cell size, location, division and fluorescence. The software provides the means of quantifying cell behavior before and after antibiotic treatment.

One such system we would like to apply this software to is our work on proteolytic queueing and antibiotic tolerance. Proteases are responsible for protein degradation and, as such, regulate many cellular functions. To better identify the role proteases play in persistence, we used proteolytic queueing to interfere with proteolytic activity. We found that interfering with degradation at the protease ClpXP increases antibiotic tolerance ~80 and ~60 fold in an *E. coli* population treated with ampicillin and ciprofloxacin, respectively. I used stochastic modeling to support our results, and we have experimentally determined that altering the expression of the synthetic system affects the level of tolerance in the population. I am currently using next-generation sequencing to identify the systems being affected by the queue.

1. INTRODUCTION

Microbial survival is a challenge in the face of constantly changing and stressful environments. As a result, microbes have developed countless, robust mechanisms to survive harsh environments, and these mechanisms can both benefit and challenge humanity. On one hand, areas of industry that rely on microbes take advantage of population robustness for bioproduction, bioremediation and other processes. On the other hand, industries that need to control pathogens and invasive microbes face a constant challenge of developing new antimicrobial drugs. As biotechnology develops, the application of synthetic biology to study these phenomena and solve the challenges microbes present has opened up opportunities for further exploration of microbial systems. In this work, we examine the regulation of genetic systems related to stress responses and develop new techniques to study these systems at the single-cell level.

1.1. Toxin-antitoxin (TA) Systems

Toxin-antitoxin systems are two-part systems consisting of a toxin and an antitoxin. Toxins in these systems slow growth by affecting metabolic processes (e.g. replication, transcription and translation) and can even lead to cell death (e.g. artificial overexpression of toxin can kill cells); antitoxins neutralize their cognate toxins through a variety of mechanisms. TA systems were first characterized as “addiction modules” located on plasmids. The genes for toxins and antitoxins often overlap, and the dependence on the antitoxin to prevent toxin activity is what led to their classification as addiction modules[1,2]. In the intervening decades, TA systems have been found in abundance throughout prokaryotic genomes and studied in the contexts of programmed cell death, phage infection, biofilm formation, virulence, stress response and persistence[3]. In *Salmonella,* TA systems have been shown to play a role in intracellular survival[4] and genomic analysis shows that TA systems are more prevalent in pathogens[5]. Currently, TA systems are thought to be present in many genomes because of horizontal gene transfer[6], and most free-living prokaryotes have multiple TA systems[7] with approximately 93% of known TA systems are chromosomal[3].

Genomic analyses have found that pathogens have higher numbers of TA systems than closely related non-pathogenic strains[5,8] and free-living prokaryotes (both bacteria and archaea) have TA systems. The ubiquity of TA systems in free-living prokaryotes strongly suggests that TA systems are related to survival, particularly cell survival in the face of stressful environments. Furthermore, TA systems are regulated in response to stress and are largely affected by proteases. Proteases are responsible for the degradation of the antitoxin proteins and affecting protease activity (which often occurs in response to stress) can increase toxin activity[1,2,9-12]. As an example, YoeB toxin activity is dependent on Lon degradation of its cognate antitoxin, YefM[9], and YoeB activity increases in response to heat shock as a result of increased degradation by Lon[13]. TA systems also respond to oxidative and nitrosative stresses, nutrient deprivation, acid/alkaline pH and bile acids[8]. The effects of TA systems on cell physiology thus emerge from an intricate network triggered by toxin levels and other factors, and activation of this network leads to altered cell metabolism and increased odds of cell survival under stressful conditions[8,14].

1.2. Regulation of TA Systems

It is well-established that TA systems are coordinated in a network, meaning that an individual TA system does not operate independently within a cell[15-18]. This is in part due to the regulatory activities of toxin-antitoxin proteins; TA systems commonly regulate their own transcription (autoregulation). A particular type of autoregulation has been shown in several TA systems, referred to as *conditional cooperativity*, wherein the ratio of toxin-to-antitoxin affects auto-repression resulting in decreased transcription when toxin levels are higher than antitoxin[19-21]. However, conditional cooperativity is not universal in TA systems[22,23]. In addition to autoregulation, some TA systems have been shown to regulate other TA systems[18] and even other genes, such as the stress response protein *cspD*[24]. In turn, TA systems are often upregulated in response to stress, including conditions that are commonly caused by the host response to infection (e.g. heat shock)[8]. Expression analysis of TA systems in *Mycobacterium tuberculosis* (contains 79 TA systems) showed that the systems responded to stress differently and that the level of toxin and antitoxin transcripts changed under certain conditions[25]. While it is clear that TA systems respond to stress, the role these systems play in response to antibiotic stress is heavily debated[26-28].

1.3. Antibiotic Tolerance, Persistence and Resistance

The discovery of penicillin in 1941 was one of the most momentous medical advances of the 20th century, in my opinion. In the intervening decades, the race to develop new antibiotics is slowly being outpaced by the spread of resistance[29]. As this trend continues, antibiotic resistance is impacting both human healthcare[30,31] and livestock production[32]. For example, *Salmonella* infections in the US alone lead to annual economic losses of about 1.3 billion dollars, and in 2005 the poultry industry lost approximately 1,000,000 birds mainly due to bacterial infections[32]. The US Department of Agriculture Economic Research Service estimates that foodborne illnesses annually cost over $15 billion[33]. The economic impact of bacterial infections and the increasing cost of treatments as antibiotic resistance lend to the importance of understanding this phenomenon. Antibiotic resistance results from genetic changes that enable a cell to grow in the presence of antibiotics.

In 2019, the Center for Disease Control (CDC) estimated that over 2.8 million antibiotic-resistant infections occur in the United States each year, which resulted in over 35,000 deaths[34]. Due to the wide impact of antibiotic resistance, the phenomenon is a growing concern worldwide, as antibiotic-resistant organisms continue to arise and propagate[30]. While antibiotic resistance in of itself is a concern, other methods of antibiotic survival (e.g. tolerance) that result from behavioral (phenotypic) changes also contribute to the steady increase in antibiotic-resistant organisms[35]. A major factor in the growing numbers of antibiotic-resistant organisms is the multidrug tolerant persister population[36]. In terms of human health, persisters play a key role in antibiotic survival and antibiotic treatment failure, particularly in chronic and reoccurring infections[36-39]. The contribution of persisters to antibiotic-resistant infections is likely a result of persister survival increasing the likelihood that a population develops antibiotic resistance[40,41].

The relationship between frequent antibiotic treatments, persistence and antibiotic resistance is a key aspect of why tolerance and persistence are so important to study and understand. One study examined the evolution of antibiotic resistance in *E. coli* populations undergoing frequent antibiotic treatments concluded that antibiotic tolerance increases over time and precedes antibiotic resistance[40]. Another study also found that repeated treatment of *E. coli* with antibiotics leads to the evolution of populations with higher (20-100%) persister levels[41], which is likely because persisters have high mutation rates and increased persister fractions correlates with increased antibiotic resistance[42]. These studies show that a high persister population provides a larger pool of cells present after subsequent antibiotic treatment to either mutate or acquire resistance genes from their environment. The relationship between persistence and resistance illustrates the need for reducing the survival of persister cells during antibiotic treatments to decrease the propagation of antibiotic resistance effectively.

1.4. Single-cell techniques

Understanding the dynamics of bacterial subpopulations requires the advent of single-cell technologies. In fact, there are many single-cell studies on tolerance and persistence[43-46]. Scaling up our ability to screen and track cells in microscopy images will improve our ability to quantify small subpopulations and even has the potential to identify cell behavior before a selection event (i.e. antibiotic treatment). Towards this end and for other work, we have developed a single-cell tracking software that incorporates machine learning to identify cells and mask them for tracking. As this software is designed to work with fluorescence imaging, cell tracking could be combined with molecular biology to design reporter strains and study cell behavior before and after antibiotic treatment. In the future, we intend to use this technique to study proteolytic activity in the antibiotic tolerant population.

1.5. Proteases, Chaperones, and Tolerance

Proteases and related chaperones are consistently identified as persister related genes in gene knockout experiments[47,48] and transcriptome analysis[49]. Proteases, such as Lon and ClpP, are globally responsible for protein degradation and cell maintenance[50,51]. They provide an important level of protein regulation throughout the cell, including degradation of RpoS (a transcription factor that responds to stress)[52] and polypeptides (incomplete proteins) synthesized by a stalled ribosome that has been rescued by the trans-translation (ribosome recovery) system[53]. As many persister studies incidentally study antibiotic tolerance[54,55], it follows that some of these mechanisms may also play a role in antibiotic tolerance. Indeed, one of the few drugs that target persisters directly, acyldepsipeptide (ADEP4), hyperactivates the protease ClpP and lowers persister levels[56]. However, studying and quantifying the role of proteases during cellular stress is made difficult by the fact that traditional techniques, e.g. knockouts and overexpression, cause stress and can lead to changes in growth rates. Herein, we use an alternative method of studying proteolytic activity in stress conditions called proteolytic queueing.

1.6. Queueing theory

Queueing theory has traditionally been applied to manufactured systems, such as computer networks and call centers, wherein one type of customer competes for processing by servers. Recently, queueing theory has been applied to biological systems, particularly intracellular processing pathways that contain bottlenecks due to limited resources. Mathematical modeling has demonstrated that traditional queueing regimes neatly organize the qualitative statistical properties of a system in which one or more proteins compete for an enzyme[57]. Theoretical predictions from these models have been explored in synthetic systems, particularly in the context of a synthetic oscillator with a particular focus on protein degradation tags for protein turnover[58]. Queueing theory can be applied to these systems at the proteolytic level due to the overproduction of proteins competing for a specific protease, thus forming a proteolytic queue. In *E. coli*, synthetic circuits often target protein degradation to the native ClpXP[58]; the Stricker oscillator (also called the dual-feedback oscillator or degrade-and-fire oscillator) is one such system[59]. Studies that co-expressed a seemingly unrelated fluorescent molecule (CFP) tagged for

ClpXP degradation with a synthetic oscillator demonstrated that the competition for ClpXP degradation between the oscillator and CFP was sufficient to propagate the oscillator signal to CFP, thus verifying that an intracellular signal can be propagated via queueing competition.

1.7. Layman's Summary

Antibiotics save lives. Antibiotic resistance means the antibiotics are less effective, which presents a problem. One way bacteria gain resistance is by surviving antibiotic treatments for a relatively short period, also known as antibiotic tolerance, and these bacteria and their descendants later become resistant. What I have done is studied how bacteria regulate (control) systems related to surviving antibiotics. First, I cover type II toxin-antitoxin (TA) systems, which are particularly interesting because the toxin and antitoxin should be produced at the same rate in the cell to prevent the toxin from disaffecting the cell (i.e. making the cell "sick"). Our results show that the cell has different methods of balancing the production rates, which an important step towards understanding how such a balance might get disrupted and lead to increased survival of stresses like antibiotics. Next, I improved a method to study single cells for long periods under a microscope. By establishing a robust method to track single cells, future studies can use this technique to study cell behavior before and after antibiotic treatment. Lastly, I used a genetically engineered system to interfere with proteases, the protein recycling centers of the cell, and showed that antibiotic tolerance increases when the proteases are less able to function. Figuring out the specific protein(s) being affected will take us one step closer to identifying a mechanism by which cells can become tolerant to antibiotics.

2. MECHANISMS FOR DIFFERENTIAL PROTEIN PRODUCTION IN TOXIN–ANTITOXIN SYSTEMS

This chapter was modified from Deter *et al.* 2017 published in *Toxins*[60].

Authors: Heather S. Deter[1,2], Roderick V. Jensen[3], William H. Mather[4], and Nicholas C. Butzin [5,*]

[1] Department of Physics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0435, USA
[2] Center for Soft Matter and Biological Physics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0435, USA
[3] Department of Biology, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0435, USA
[4] Quantitative Biosciences, Inc., Solana Beach, CA 92075, USA
[5] Department of Biology and Microbiology, South Dakota State University, Brookings, SD 57006, USA

2.1. Abstract

Toxin–antitoxin (TA) systems are key regulators of bacterial persistence, a multidrug-tolerant state found in bacterial species that is a major contributing factor to the growing human health crisis of antibiotic resistance. Type II TA systems consist of two proteins, a toxin and an antitoxin; the toxin is neutralized when they form a complex. The ratio of antitoxin to toxin is significantly greater than 1.0 in the susceptible population (non-persister state), but this ratio is expected to become smaller during persistence. Analysis of multiple datasets (RNA-seq, ribosome profiling) and results from translation initiation rate calculators reveal multiple mechanisms that ensure a high antitoxin to toxin ratio in the non-persister state. The regulation mechanisms include both translational and transcriptional regulation. We classified *E. coli* type II TA systems into four distinct classes based on the mechanism of differential protein production between toxin and antitoxin. We find that the most common regulation mechanism is translational regulation. This classification scheme further refines our understanding of one of the fundamental

mechanisms underlying bacterial persistence, especially regarding maintenance of the antitoxin to toxin ratio.

2.2. Introduction

Persistence is a metabolically inactive state that enables many bacterial species to maintain a subpopulation of cells that can survive harsh changes in the environment [61,62]. From a human health standpoint, persister cells are a growing problem since the metabolic dormancy that characterizes the persister state results in the persister population being multidrug-tolerant and a major contributing factor to ineffective antibiotic treatments. In addition, it has been suggested that persisters indirectly lead to antibiotic resistance; persister cells survive antibiotic treatment and are then able to acquire antibiotic resistant genes from their neighbors [38,39]. Investigations have revealed that a central regulator of bacterial persistence is a network of multiple toxin–antitoxin (TA) systems [17,63,64]. Evidence suggests that TA systems trigger persistence when rare events allow active toxins to accumulate and affect metabolic dormancy by slowing processes such as translation and transcription. A variety of bacterial species have TA systems, which are classified into types based on the mechanism the antitoxin uses to neutralize its cognate toxin [11,65-67]. Types I and III have RNA antitoxins that inhibit toxin synthesis or sequester their toxin, while types II, IV and V have protein antitoxins. Type II TA systems are the only type where the antitoxin protein directly binds to the toxin to form a protein–protein complex which sequesters the toxin and effectively neutralizes it [67]. In *Escherichia coli* alone, there at least 36 known TA systems, most of which are type two [17,68]. In this work, we were able to classify 10 out of 16 type II TA systems (there was insufficient data to classify the remaining six) found in *E. coli* K12 str. MG1655.

Type II TA systems are operons that encode genes for two proteins, a stable toxin and an unstable antitoxin. Since the antitoxin protein is rapidly degraded by proteases, it must be continually produced to prevent a buildup of free toxin protein and to maintain the susceptible population (non-persister state) (Figure 1) [6,11,19]. Thus, antitoxin is expected to be produced at a sufficiently higher rate than toxin in non-persister cells [65,69]. However, there is disagreement in the literature as to how this ratio is ensured across type II TA systems. It is often cited that transcriptional regulation is responsible for the higher production rate of antitoxin [21,65,70-74]. Research shows that the RnlAB system contains an internal promoter that is independently regulated and allows for the possibility of differential transcriptional regulation [75], but differential transcriptional regulation has not been confirmed to regulate the production ratio of antitoxin to toxin for most type II TA systems.

A model of TA systems that includes transcriptional regulation, conditional cooperativity, proposes that the antitoxin when complexed with its cognate toxin autoregulates the whole operon and could lead to some control of the antitoxin-to-toxin ratio [19,21], but this theory depends on antitoxin translation rates being higher than toxin [74]. Another model of TA systems also depends on the translation rate of antitoxin being higher than that of toxin, but does not require conditional cooperativity [15]. Many other operons are known to use translational regulation (different translation rates) to maintain differential protein production within an operon, including the ATPase operon [76,77]. As we will support, many type II TA systems use translational regulation, and transcriptional regulation is not the only mechanism that can explain the higher production rate of the antitoxin protein.

This study examines the possibilities of both transcriptional and translational regulation using bioinformatics approaches that combine diverse datasets and analyses. We analyzed type II TA systems in *E. coli* K12 str. MG1655 using RNA sequencing (RNA-seq) data from multiple studies and growth conditions, data from ribosome profiling analysis (Ribo-Seq), identified promoters and terminators with experimental data and

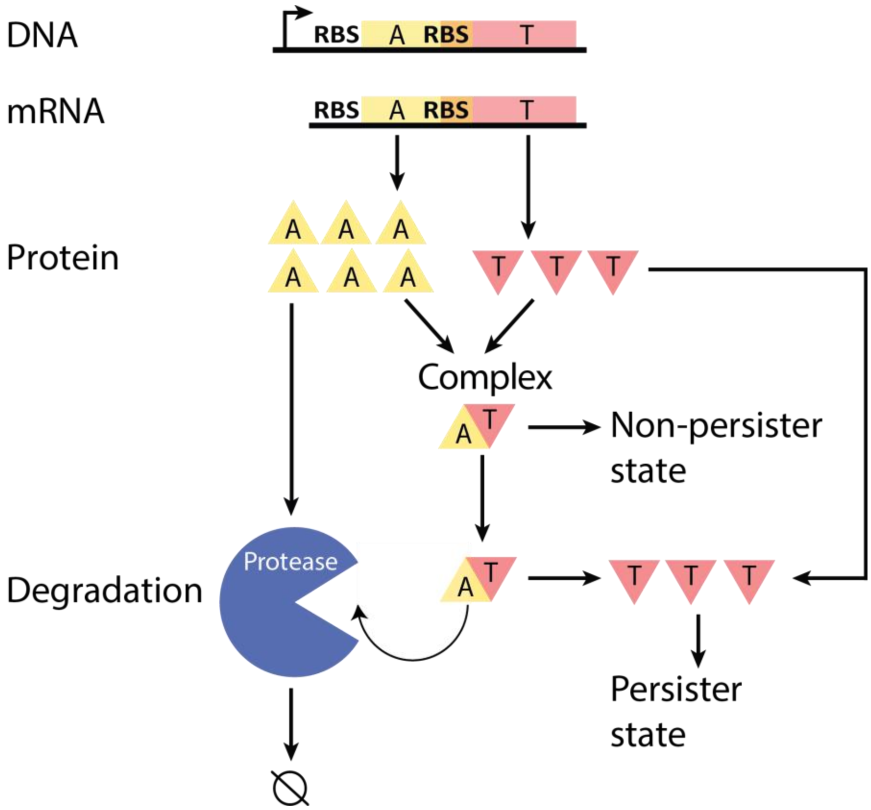


Figure 1. Schematic of a typical Type II toxin–antitoxin (TA) system. A TA system operon is transcribed to produce a corresponding mRNA, which is then translated to produce toxin and antitoxin proteins. With sufficient concentration of anti-toxin protein, toxin protein can be primarily neutralized in a complex, which allows the cell to maintain a non-persister state, or else the toxin can exist as a free and active protein in the cell, which may lead to persistence. The antitoxin-to-toxin protein ratio, which is expected to be sufficiently greater than 1.0 in the non-persister state, controls these scenarios. Antitoxin is actively degraded by proteases (at a greater rate than the toxin), which requires the excess production of antitoxin to ensure the non-persister state. T: toxin. A: antitoxin. Not to scale.

predictions annotated on EcoCyc [78], and calculated translation initiation rates with prediction algorithms (TIR calculators). By combining these analyses, we classified *E. coli* type II TA systems into four classes based on the mechanisms of differential protein production. This classification scheme further refines our understanding of how TA systems maintain differential antitoxin-to-toxin expression and one of the fundamental mechanisms underlying bacterial persistence.

A major result of our investigation is that differential transcriptional regulation of antitoxin and toxin expression is unlikely in many type II TA systems. The key pieces of evidence are a less than two-fold difference of antitoxin-to-toxin mRNA for several systems, and a lack of obvious internal promoters or terminators within the operon sequence that could explain excess antitoxin mRNA. We predict that TA systems with less than a two-fold difference in mRNA expression leverage translational regulation to maintain higher production rates of antitoxin. Our results extend the pattern found for many operonic genes that use translational regulation to maintain differential protein production from a single transcript [76,77,79].

2.3. RESULTS AND DISCUSSION

We analyzed 10 different type II TA systems in *E. coli* (Table 1) using data informing their operon organization, mRNA concentration (RNA-seq), protein synthesis rates (Ribo-Seq), and predicted translation initiation rates (TIR's). Six type II TA systems lacking substantial RNA-Seq data were not included. Understanding of the operon organization and mRNA products for each TA system led to a compelling classification scheme that includes at least four major classes. We find that these classes of TA systems all have mechanisms in place to ensure sufficient production of antitoxin protein relative to toxin protein, though the details vary from class to class. In this section, we present the major findings of our study. We first present the details concerning our TA system classification scheme; then we analyze trends in associated quantitative data. We finally discuss our interpretation of these results in terms of the four identified classes of TA systems.

2.3.1. Classification Scheme for Type II Toxin–Antitoxin Systems Based on DNA Sequence and mRNA Products

A survey of 10 TA systems (Table 1) using the online database EcoCyc [78], revealed consistent patterns based on operon organization, which refers to the order of the genes (whether the toxin is upstream at the 5’ end or downstream at the 3’ end of the operon), and whether there is an additional promoter or transcriptional termination mechanism that can produce multiple distinct mRNA products.

Further analysis of our representative set of TA systems led to one major class and three other classes (Figure 2). The major class (Class 1: FicAT, MazEF, MqsAR, PrlF-YhaV, RelBE, YefM-YoeB) found in our study includes TA systems that produce a single transcript, which results in the condition $A \approx T$ for mRNA (the concentration of antitoxin coding region is approximately equal to the concentration of toxin coding region). Because these systems are apparently not differentially regulated at the transcriptional level, we predict that different translation rates of the two proteins are responsible for higher antitoxin protein production than toxin. These predictions are confirmed later in this article.

Table 1. Type II TA systems examined in this study, ordered alphabetically. Ten out of sixteen type II TA systems (there was insufficient data to classify the remaining six) in E. coli were considered. The toxin is underlined.

TA System	Toxin Function	Toxin Family
DinJ-YafQ	Endoribonuclease that act 5' to adenine between the codon second and third nucleotides [80]	RelE [6]
FicAT	Mediates post-translational protein modification [81]	Unknown
HicAB	mRNase [82]	Unknown [83]
MazEF	mRNA interferase that cleaves mRNA at ACA sites [84]	CcdB/MazF [6]
MqsAR	Ribosome-independent RNase [85]	RelE [83]
PrlF-YhaV	Ribonuclease [86]	RelE [83]
RelBE	mRNA interferase that cleaves mRNA in the ribosome A site [87]	RelE [6]
RnlAB	RNase [75]	Unknown [83]
YafNO	Ribosome-dependent mRNA interferase [88]	YafO [6]
YefM-YoeB	Ribosome-dependent mRNase [89]	RelE [6]

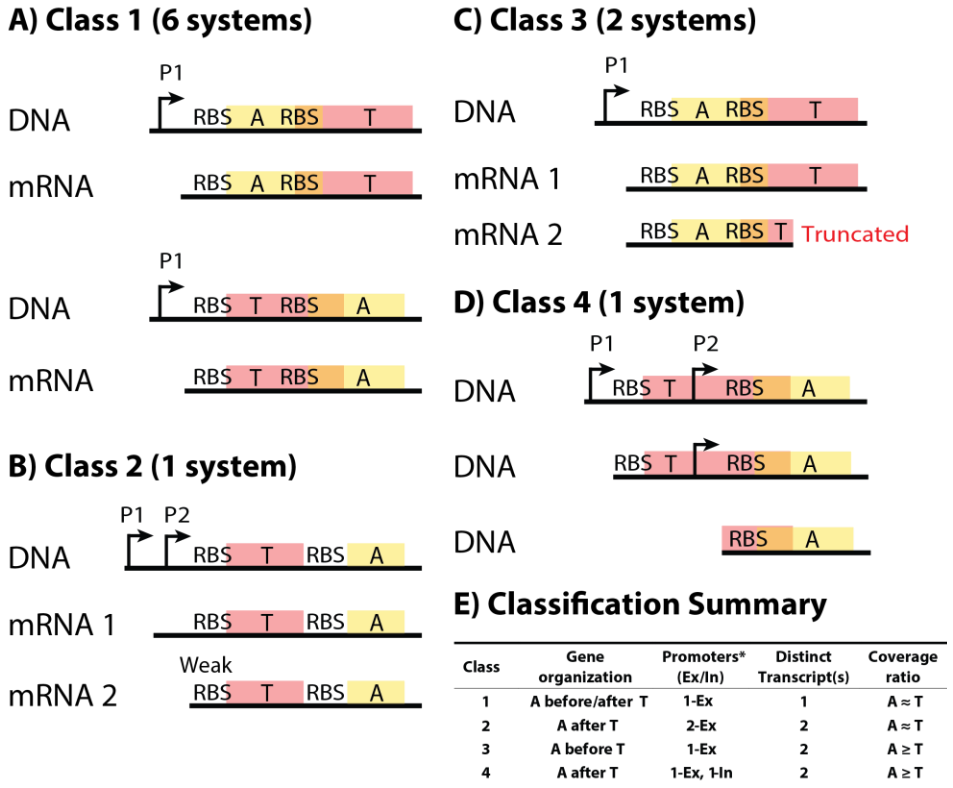
The three other classes (Class 2: HicAB; Class 3: DinJ-YafQ, YafNO; and Class 4: RnlAB) are inspired by a few systems that have similar gene organization to Class 1, but other features deviate from Class 1 that allow for transcriptional or post-transcriptional regulation of the antitoxin to toxin protein production ratio. Interestingly, we find the ribosome-binding site (RBS) of the downstream gene is embedded in the upstream gene for nine out of 10 TA systems (only HicAB in Class 2 does not).

Many of these classes were identified by examining mRNA expression from RNA-seq data in combination with promoter identification (see methods, section 3.1). Class 2 is similar to Class 1, but an additional external promoter (located before the coding regions) is near the toxin RBS. Transcription from this promoter results in a truncated mRNA that has a weakened RBS for toxin translation. Thus, antitoxin is produced at a greater rate than toxin [23]. Class 3 likely has a truncated mRNA according to our analysis of RNA-seq data, but due to the gene organization of this class, it probably has either an early transcriptional terminator or post-transcriptional mRNA degradation to truncate the toxin-coding region. Class 4 has an internal promoter that produces excess antitoxin mRNA. Classes 3 and 4 rely on increased abundance of functional antitoxin-coding mRNA relative to toxin. Thus, a higher TIR for antitoxin relative to toxin is perhaps then not as necessary, in contrast to Class 1.

2.3.2. Antitoxin and Toxin mRNA Coverage by RNA-seq

Our classification scheme (Figure 2) was supported using a diverse-data analysis approach. As part of this analysis, we quantitatively estimated the mRNA abundance for antitoxin and toxin coding regions using publicly available RNA-seq data. A total of 13 different whole transcriptome *E. coli* K12 str. MG1655 RNA-seq datasets were derived from two different studies with a total of six different experimental conditions that include different growth densities and media (Tables

Table **A1**).



Class	Gene organization	Promoters* (Ex/In)	Distinct Transcript(s)	Coverage ratio
1	A before/after T	1-Ex	1	A ≈ T
2	A after T	2-Ex	2	A ≈ T
3	A before T	1-Ex	2	A ≥ T
4	A after T	1-Ex, 1-In	2	A ≥ T

Figure 2. TA (toxin–antitoxin) systems were classified into four different classes based on DNA sequence and measured mRNA products. (A–D) Representative wire diagrams of the operon organization and mRNA products of each class (not to scale). (A) Class 1 TA systems (FicAT, MazEF, MqsAR, PrlF-YhaV, RelBE, YefM-YoeB), the most abundant class in our study, have a single transcript for the operon and should rely on translational regulation to ensure higher antitoxin production relative to toxin production; (B) Class 2 is characterized by a second promoter that produces a slightly shorter transcript, which is predicted by our work to have a lower toxin translation rate than the transcript of the first promoter. The one example (HicAB) available has non-overlapping coding regions with the toxin upstream from the antitoxin; (C) Class 3 TA systems (DinJ-YafQ, YafNO) have a truncated transcript in addition to the whole transcript for the operon, due to some unknown mechanism (perhaps a terminator or post-transcriptional degradation); (D) Class 4 TA systems (RnlAB) produce two transcripts: a transcript of the whole operon (both toxin and antitoxin mRNA), and a transcript that can only be translated to antitoxin; (E) A summary of the classification of TA systems in this study. For the promoter column, the number indicates the number of promoters that are located upstream the coding regions (external, Ex), or within the coding regions (internal, In). Each class has a different A to T RNA ratio (see Table 2) based on analysis of 13 different RNA-seq datasets from a variety of conditions, such as growth in rich and minimal media, and cell densities (Table S1). *Some genes may have additional promoters, but they did not affect the ratio of mRNA expression. P: Promoter. RBS: Ribosome Binding Site. A: Antitoxin. T: Toxin.

RNA-seq analysis shows that the sequencing coverage (number of reads aligned to the gene normalized by dividing the length of the gene) for seven TA systems had less than a two-fold difference in expression at the mRNA level for a variety of conditions (Table 2, Figure 3), as anticipated for Class 1 and Class 2 TA systems. The two TA systems in Class 3 (DinJ-YafQ, YafNO) consistently had more antitoxin mRNA than toxin, which is also as anticipated, though the two-fold difference is modest. Our one example of a Class 4 system (RnlAB) also had less than a two-fold difference between toxin and antitoxin mRNA expression, but the coverage of functional antitoxin mRNA is higher than functional toxin mRNA (Figure A1).

Table 2. Median of the ratios for antitoxin to toxin coverage based on RNA-seq. The median (n = 13) of the antitoxin to toxin coverage ratios calculated for each of the 13 datasets (see Methods). All TA systems exhibited nearly equal (less than a two-fold difference) or higher antitoxin mRNA abundance. TA systems are ordered by their class. * Although near the two-fold cut off, these systems are placed in Class 1 from additional analysis (Table **3**). ** A truncated mRNA results in differing levels of functional mRNA (Figure A1). The toxin is underlined.

TA System	A/T	St. Dev	Class
FicAT *	1.99	0.53	1
YefM-YoeB *	1.89	0.42	1
MazEF	1.42	0.25	1
PrlF-YhaV	1.11	0.27	1
MqsAR	0.69	0.16	1
RelBE	0.95	0.10	1
HicAB	0.93	0.20	2
DinJ-YafQ	2.89	0.35	3
YafNO	2.08	0.39	3
RnlAB **	1.01	0.25	4

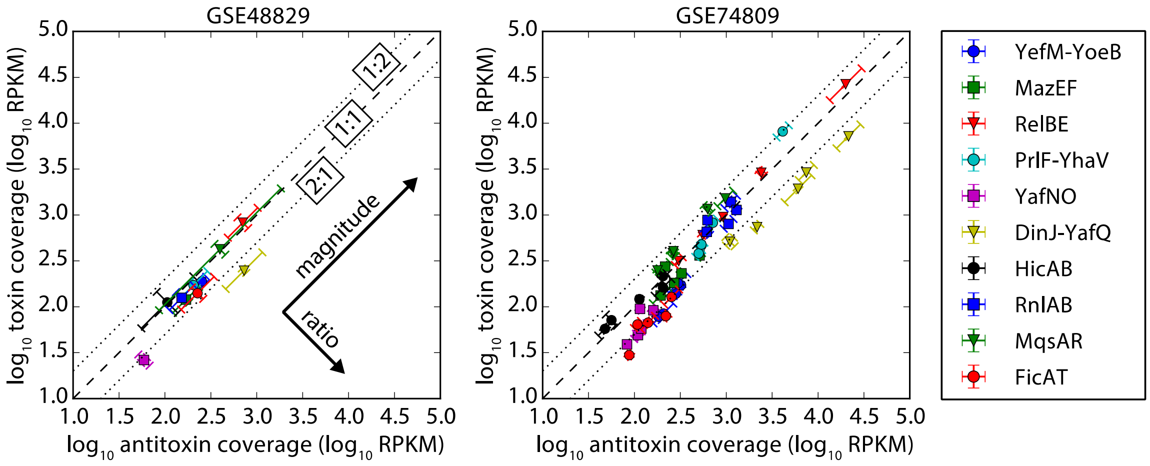


Figure 3. RNA-seq coverage of antitoxin vs. toxin open reading frames. Left: dataset GSE48829 [26] triplicate biological replicates sampled during exponential growth in minimal media. Right: dataset GSE74809 [27] duplicate biological replicates sampled from five different stages of growth in M9 (glucose) media. Both plot the quantitative analysis of sequence coverage of antitoxin and toxin (see methods, section 3.1) on a common axis for a variety of TA systems. Most TA systems in the conditions considered have less than a two-fold difference in coverage (1:1 coverage is indicated by a dashed line) between antitoxin to toxin mRNA, suggesting expression of the mRNA as a single transcript. TA systems that fall within the dotted lines had a 1:2 to 2:1 ratio of antitoxin to toxin coverage. TA systems were not included if either toxin or antitoxin had an average of less than one read per base for more than half of the datasets (the minimum read count for a gene is 168 reads). The error was calculated in two different directions (ratio and magnitude, see methods, section 3.1), and error bars are aligned to these primary directions to illustrate the low error of the ratio. The individual replicates had similar groupings (Figure A2). Our results are also supported by a global error analysis (Figure A3), which shows a typically small error for the replicates. Units of coverage are Reads Per Kilobase Million (RPKM).

2.3.3. Protein Synthesis Rates Determined by Ribo-Seq

Sequencing coverage of mRNA was suggestive, in that most TA systems had less than a two-fold difference in coverage of antitoxin and toxin. To test our classification of type II TA systems into four classes based on gene organization and RNA-seq data, we analyzed protein synthesis rates from publicly available Ribo-Seq data. Indeed, the Ribo-Seq data supports our classification. Protein synthesis rates were determined using the Li et al. (2014) open source database that gives protein synthesis rates for *E. coli* based on Ribo-Seq data [90] (See methods, section 3.1). Protein synthesis rates were calculated from Ribo-

Seq based on the coverage (counts normalized by gene length) of the ribosomally protected mRNA present after the mRNA has been extracted and treated with RNase [91]. From this analysis, seven out of eight of the TA systems with sufficient coverage (HicAB and FicAT had low confidence, less than 128 mapped reads) had a higher protein synthesis rate for the antitoxin than the toxin (Figure 4), as anticipated. We hypothesize that the systems with less than a two-fold difference in toxin and antitoxin mRNA expression likely use translational regulation to maintain the higher antitoxin production rate. The major exception we found was the synthesis ratio for the RnlAB TA system (Class 4), which we hypothesize may be due to third protein that interacts with this system (see comments in section 2.3.5).

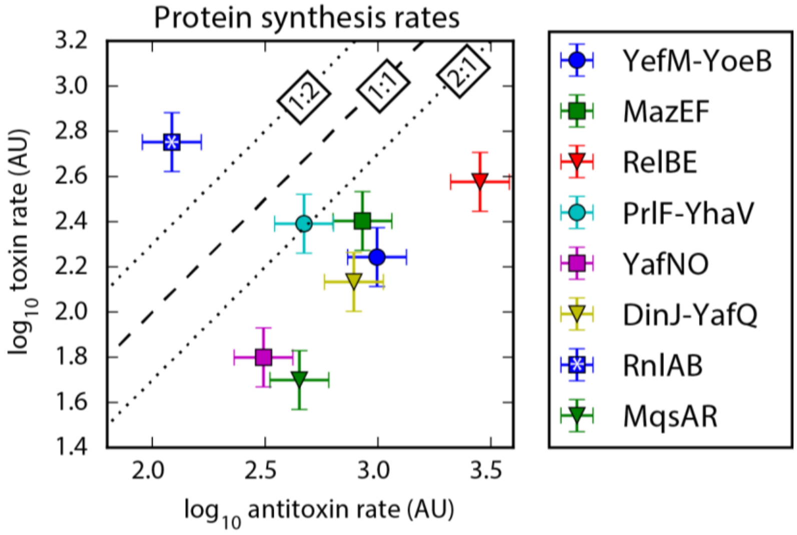


Figure 4. Protein synthesis rates plotted for each TA system based on Ribo-Seq data. A systematic bias towards higher antitoxin protein synthesis rate relative to toxin is evident in all but one case, and the one outlier (RnlAB), which we explain in Section 2.3.5. The dashed line represents a 1:1 ratio of antitoxin to toxin, and the dotted lines mark a two-fold difference. TA systems with values of low confidence (less than 128 reads) in protein synthesis data [90] were not included in the figure. Error bars assume a 30% error, as estimated in Li et al. 2014 [90]. Arbitrary units: AU.

2.3.4. Analysis of Differential Protein Expression Using Translation Initiation Calculators (TIRs)

An independent theoretical investigation of TIRs for the 10 TA systems was done to assess the robustness of our findings based on Ribo-Seq data. We predicted TIRs using three experimentally-conditioned TIR calculators, of which two are based on detailed thermodynamic models (the RBS Calculator [92,93] and the UTR Designer [94]), and a third is based on a log-linear regression of *E. coli* gene expression (Barrick Calculator [95]). The results from the TIR calculators vary greatly due to the inherent differences between the calculation methods, but for six of the seven systems that have less than a two-fold difference in mRNA expression, at least two out of three calculation methods qualitatively agree that for each system the antitoxin TIR is higher than toxin (

Table **3** and Table A2). In all cases, the calculators support our classification of the TA systems and provide a method independent of Ribo-Seq.

2.3.5. Summary and Discussion of Major Trends and Exceptions for TA System Classes

Our picture from a diverse-data analysis of 10 TA systems has thus led to the following key interpretations of their behavior: Class 1 is the most abundant class with six TA systems that use translational regulation to maintain a high antitoxin-to-toxin protein synthesis rate, as seen in the Ribo-Seq analysis (Table 2, Figure 4). The TA systems in this class have less than a two-fold difference in mRNA expression (Table 2, Figure 3) and likely use translation regulation to maintain the antitoxin-to-toxin ratio. The RBS site of the downstream gene in these systems overlaps with the upstream gene's coding region, and they are not dependent upon the order of those genes to maintain the antitoxin-to-toxin production ratio (Figure 2A). Included in this class is RelBE, which previous researchers have shown to use translational regulation to maintain a high antitoxin-to-toxin ratio [70]. The only TA system in this study to clearly use translational regulation that is not in Class 1 is HicAB, which we have placed in Class 2.

Table 3. Classification of each TA system. Based on the classification scheme outlined in Figure 2, we used qualitative conclusions from our analysis to classify 11 different TA systems in *E. coli*. Data from RNA-seq was used to determine mRNA ratio. The toxin is underlined.

TA System	Class	mRNA Ratio	RBS Calculator	UTR Designer	Barrick Calculator	Synthesis Rates
FicAT	1	<2 *	+	+	+	LC
MazEF	1	<2	+	+	+	+
MqsAR	1	<2	−	+	+	+
PrlF-YhaV	1	<2	+	+	+	+
RelBE	1	<2	+	+	+	+
YefM-YoeB	1	<2 *	+	+	+	+
HicAB P1	2	<2	−	−	+	LC
HicAB P2			+	−	+	
DinJ-YafQ	3	>2	+	−	−	+
YafNO	3	>2	+	−	−	+
RnlAB	4	<2 **	+	+	−	−

+ Antitoxin is higher than toxin. -: toxin is higher than antitoxin. LC: low confidence; * mRNA ratio ranges broadly (sometimes >2) but calculator results place these systems in Class 1; ** A truncated mRNA results in differing levels of functional mRNA (Figure S1).

The protein synthesis rates of HicAB could not be determined by the Li et al. (2014) data due to low expression of this system, but our RNA-seq analysis shows that the HicAB system has less than a two-fold difference in mRNA expression, like Class 1. Unlike Class 1, HicAB does not have overlapping genes and the antitoxin is located downstream of the toxin (Figure 2B). Our first analysis showed that translational regulation is unlikely because two out of three TIR calculators predict that the antitoxin TIR is lower than the toxin. Interestingly, a recent study showed that HicAB combines transcriptional and translational regulation by using two different promoters at the beginning of the operon to produce two different transcripts with different toxin translation rates [23]. When using the transcription start site of the second promoter, the results of the RBS Calculator change to predict the antitoxin TIR to be higher than the toxin. This entails in two of the three TIR calculators predicting antitoxin TIR is higher than toxin (Table A2), supporting that this system uses translational regulation and supporting the effectiveness of the TIR calculators when using a consensus of two out of three.

Class 3 contains the remaining two systems that had higher antitoxin protein synthesis rates (Figure 4), but these systems also have two-fold higher antitoxin mRNA expression than toxin (Table 2, Figure 3). The organization of these systems has the antitoxin upstream of the toxin with the RBS of the toxin overlapping the end of the antitoxin-coding region (Figure 2C). While the difference in mRNA would indicate that these systems are transcriptionally regulated, the regulation mechanism apparently does not use a promoter; the antitoxin location upstream of the toxin would result in an additional promoter for the antitoxin reading through the toxin as well. We hypothesize that the lower toxin mRNA level is a result of a truncated mRNA due to an unidentified transcriptional terminator or RNA degradation specific to the toxin mRNA sequence. Several type II toxins are endoribonucleases (including those in Class 3) and they could possibly degrade their own message with a bias toward toxin mRNA degradation (Table 1). Regardless of the mechanism, Class 3 TA systems likely use differential mRNA expression to regulate antitoxin-to-toxin ratios either transcriptionally with terminators or post-transcriptionally through RNA degradation.

Class 4 contains RnlAB, the only TA system that does not have a higher antitoxin protein synthesis rate than toxin (Figure 4). The gene organization of this system has the antitoxin downstream of the toxin with the antitoxin RBS overlapping the end of the toxin-coding region (Figure 2D). This organization means that the RnlAB system can use transcriptional regulation of an internal promoter to express antitoxin mRNA higher than toxin, and analysis of the RNA-seq data supports the presence of an internal promoter at approximately 280 nucleotides upstream from the antitoxin (Figure A1). The RNA-seq analysis is supported by a previous study on the regulation of RnlAB, which experimentally determined an internal promoter to be near this location. The same study establishes that the two promoters are independent and transcriptionally regulated, which would allow for upregulation of antitoxin expression [75]. The internal promoter in this system provides a possible mechanism to upregulate antitoxin production, but the protein synthesis rates indicate that toxin production is higher than antitoxin production.

The toxin homologs RnlA and LsoA (not found in the *E. coli* strain used in this study) are unlike other type II TA systems because the protein structure of these toxins is different from established structures of others. Additionally, a third protein interacts with the RnlAB system, RNase HI. Recent studies have shown that RNase HI acts as a corepressor of the toxin RnlA in the presence of its antitoxin RnlB [96]. However, when antitoxin is not present in the system (in the absence of RnlB), RNase HI stimulates RnlA activity [97]. This system also acts as an anti-phage response, because during infection RnlA degrades the infecting phage’s mRNA and RnlB concentration decreases allowing RNase HI to activate RnlA and strengthen its activity [96,97]. The third component of the RnlAB system changes the dynamics and is likely the reason that the ratio of antitoxin protein synthesis rate to the toxin is lower than expected.

2.3.6. Incorporation of Our Results into Current Models

Currently, many studies on regulation of TA systems focus on conditional cooperativity, which depends on transcriptional regulation by TA autoregulation complexes. Numerous studies on TA autoregulation complexes focus on RelBE, a Class 1 system. The RelBE system is known to produce the antitoxin (RelB) approximately 10-fold faster than the toxin (RelE) [70], which is further supported by the protein synthesis rate data, which predicts RelB to be produced 7.5 times greater than RelE [90]. Studies on RelBE show that DNA binding of the $RelB_2E$ complex is stronger than the $RelB_2$ alone [20,98], and studies suggest that the stronger autoregulation activity of the complex is important to the control of the antitoxin-to-toxin ratio [74,98]. However, in these models the translation rate of antitoxin being higher than that of the toxin is critical to prevent a constant overabundance of free toxin, and therefore the system is still dependent on translational regulation [70,74]. In contrast to the TA autoregulation complex of the RelBE system, the DinJ-YafQ complex does not have increased DNA binding affinity when compared to DNA binding affinity of the antitoxin (DinJ) alone [22]. Interestingly, the DinJ-YafQ system was placed into Class 3, which our results suggest relies on increased abundance of antitoxin mRNA rather than translation rates. Further studies are needed to fully understand the variety of regulation mechanisms that play a role in the maintenance and control of the antitoxin-to-toxin ratio.

3. CONCLUSIONS

Both translational and transcriptional regulation play important roles in the maintenance of the production ratio of antitoxin to toxin in type II TA systems in the non-persister state. Analysis of RNA-seq data reveals that most TA systems have less than a two-fold difference between antitoxin mRNA and toxin under the conditions studied in these datasets (Table A2). The antitoxin-to-toxin production ratio in Class 1 TA systems, which represent the most abundant class in our study, is not differentially regulated transcriptionally or post-transcriptionally, but it is regulated at the translational level. Class 2 uses a combination of both transcriptional and translational regulation to ensure a higher rate of antitoxin production. Class 3 is likely not regulated translationally but by some mechanism that results in a truncated mRNA and overall greater expression of antitoxin mRNA than toxin. Class 4 contains RnlAB, which is regulated by internal promoters, but the interaction of RNase HI with RnlAB lessens its dependency on a higher production rate of antitoxin.

Our classification of TA systems emphasizes their diversity with respect to gene organization and regulation mechanisms, while other classification systems are based on protein structures and functions [6,67]. The diversity of regulation mechanisms explains the disagreement in the literature as to whether TA systems use transcriptional regulation or some other form of regulation to maintain an antitoxin production rate higher than the cognate toxin. Various studies have focused on specific TA systems and their conclusions have been applied to type II TA systems as a group, but our results emphasize the diversity found within these systems. Further applications of the methods used in this study would be to classify and expand the knowledge of TA systems in other organisms and other differentially expressed operons in bacteria. Using bioinformatics methods alone, the methods in this study can be applied to classify other type II TA systems. Classification of different type II TA systems into these classes should allow future researchers to predict regulation (transcriptional, post-transcriptional, or translational) without the expense and time of RNA-seq and Ribo-Seq experiments.

3.1. Materials and Methods

3.1.1. DNA Sequence and mRNA Sequence Analysis

The annotated *E. coli* K12 str. MG1655 reference sequence NC_000913.3 was used for all gene sequences in this study.

3.1.2. RNA-seq Analysis

Selected RNA-seq datasets (Tables

Table **A1**) from GEO NCBI (accession numbers GSE48829 and GSE74809) [99,100] were aligned to the *E. coli* K12 str. MG1655 reference sequence NC_000913.3 using Geneious v. 10.0.9 [101]. Gene expression was analyzed in Geneious using the calculate expression tool to generate a list of reads mapped to each gene. Coverage for any specific antitoxin or toxin was calculated in Reads Per Kilobase Million (RPKM). Defining n as the raw number of reads mapped to the coding sequence (CDS) (with partial reads, reads mapping to more than one region of the genome, counted as 0.5 reads), T as the total number of mapped reads for a particular RNA-seq dataset, and L as the number of bases in the CDS, then RPKM = $10^9\ n\ /\ LT$. Results were exported to a CSV file, which was then processed by custom Python scripts using the NumPy library [102].

The ratio of antitoxin to toxin was calculated for each TA system for all 13 datasets, and the median of those 13 ratios was used to determine the A/T ratio. The standard deviations were calculated using the 13 ratios from the datasets (Table 1).

Error estimates for gene coverage were computed as follows. Each RNA-seq dataset contained biological replicates, and we analyzed error by comparing gene expression between replicates (see Figure A2 for an illustration of replicates). Dataset GSE48829 contained triplicate data for one experimental condition, while dataset GSE74809 contained duplicate data for five experimental conditions. For each dataset separately, we ran an error analysis of the log-error (standard error of logarithmic quantities) in two major directions, ratio and magnitude. We choose log-error since this is the most natural representation of the error for features plotted in log-space, as with Figure 3. If c_A and c_T are gene coverages for antitoxin and toxin, respectively, then the error of the log-ratio is the standard error for the quantity $\ln c_A - \ln c_T \, , = \ln(c_A/c_T)$. Since $\ln(c_A/c_T) \approx (c_A/c_T) - 1$ when c_A and c_T are close in value, this error also approximates the error of the ratio (the constant -1 does not contribute to the standard error). This error is plotted as error bars along the ratio direction in Figure 3 after scaling by the factor $\ln(10)$ to plot in logarithm base 10 space. A complementary error estimate is that of the log-magnitude, which we define as the standard error of the quantity $\ln c_A + \ln c_T \, , = \ln(c_A \cdot c_T)$, and which we plot as error bars along the magnitude direction in Figure 3 after scaling by $\ln(10)$. Notice that the log-ratio error is zero for quantities with zero uncertainty in the ratio (c_A/c_T), even though the log-magnitude error may be significant.

3.1.3. Protein Synthesis Rates Based on Ribosome Profiling (Ribo-Seq)

We used protein synthesis rates for *E. coli* calculated by Li et al. 2014 [90] from their Ribo-Seq data according to their open source database: http://ecoliwiki.net/tools/proteome/. Genes with less than 128 mapped reads were given values of low confidence in this database.

3.1.4. Translation Initation Rate (TIR) Calculators

TIRs were predicted using three different translation rate initiation (TIR) calculators with the reference sequence, NC_000913.3. The reverse engineering feature of the Ribosome Binding Site Calculator v2.0 was used [92,93]. Any difference in sequence length using the RBS Calculator v2.0 changed the translation initiation rates because this software accounts for secondary structure. Therefore, the input sequence we used included the entire 5' UTR region as determined by the transcription start site annotated in the EcoCyc database [78] to the end of the TA system. The reverse engineering feature of the UTR Designer was used with the required input of −25 to +35 from the start codon for each gene [94]. The third method analyzed the sequence −11 to +1 from the start codon using an equation based on empirical data developed by Barrick et al. in their study of *E. coli* ribosome binding sites (Barrick Calculator) [95]. We used the agreement of two out of three calculators to determine whether antitoxin TIR is predicted to be higher than toxin.

3.2. Author Contributions

H.S.D performed RNA-seq data and translation initiation calculator analysis to classify the TA systems and wrote the manuscript. N.C.B. initiated and directed the project. R.V.J. was essential to understanding the methods of RNA-seq data analysis. W.H.M. directed the project and performed statistical analyses for the RNA-seq data. All the authors took an active part in writing and editing the manuscript.

4. A CELL SEGMENTATION/TRACKING TOOL BASED ON MACHINE LEARNING

This chapter contains material from: Deter *et al.*, "A Cell Segmentation/Tracking Tool Based on Machine Learning". *Methods Mol Biol*[103].

4.1. Introduction

During the last decade, there has been a transition in the analysis of cellular physiology from the batch level (population-scale) to the single cell level. This transition has been stimulated by the development of quantitative and high-throughput techniques that require computer-aided methods to extract information at the single cell level. Using these approaches researchers have been able to show, for example, the relevance of stochastic sources in gene expression[104] or the logic underlying cell size homeostasis[105]. Thus, single-cell analysis is less subject to averaging effects (Figure 5) and offers a level of discrete detection that is unobtainable with traditional techniques[58,106-108]. In this context, the adoption of single-cell microscopy techniques has been limited because identifying, tracking, and quantifying single cells within a population of cells is usually difficult, time-consuming, and prone to errors that require manual corrections. Indeed, the identification stage implies a methodology able to recognize and outline the domain of individual entities (segmentation) and is particularly critical since tracking and quantification depend on it.

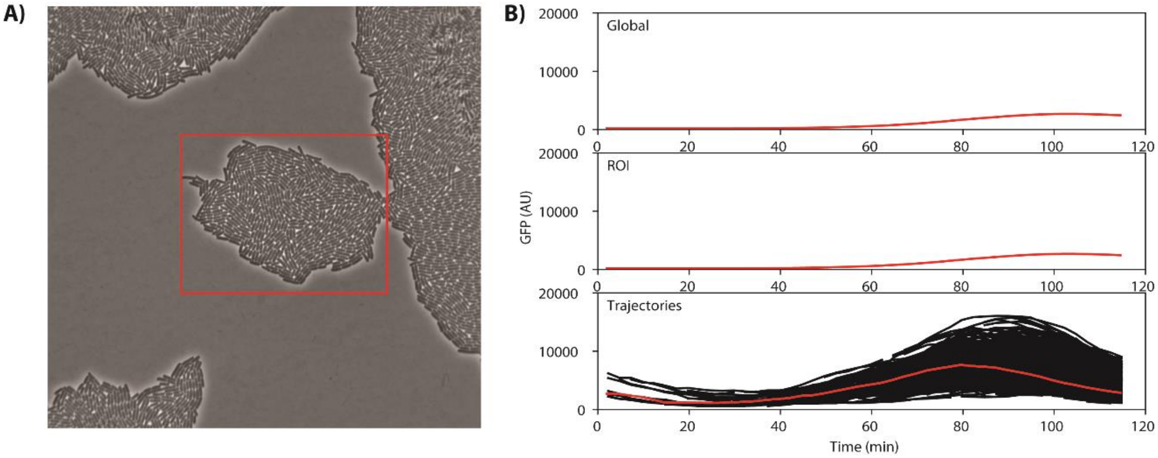


Figure 5. A) The final frame of the image dataset. The region of interest (ROI) is outlined in red. **B)** Median fluorescence for given selections over time (below). Global: the median fluorescence over time for the whole image. ROI: the median fluorescence over time for the ROI is outlined in red in A). Trajectories: the fluorescence over time for each trajectory (black) and the median fluorescence for all cells (red).

Traditional cell segmentation algorithms are based on image processing techniques that ultimately compute gradients and use thresholding to measure the intensity and spatial relationships of pixels in order to detect cell boundaries. The latter is especially challenging in dense cell populations, e.g. bacterial colonies, and, while some edge detectors have been proven to be more effective than others, e.g. Marr-Hildreth vs. Canny[109], small changes in the microscopy illumination conditions require, more often than not, non-trivial adjustments of the segmentation parameters. In that regard, during the last years, a number of segmentation/tracking software suites have been publicly released[110-112]. Here we highlight three examples that, while essentially based on, and consequently constrained by, the aforementioned methodology, stand out because of their additional features and reliability. MicrobeJ[113,114] is a plugin available through Fiji ImageJ[115] that has a wide variety of tools available to analyze cell morphology and track cells in their user-friendly interface. Oufti[116] offers a friendly user interface and a number of functionalities for quantitative analysis that include subpixel resolution for “reading” fluorescent signals within single cells. On the other hand, CellX[117] uses a novel approach for cell segmentation based on membrane patterning that is versatile in terms of cell shapes and robust to image noise.

More recently, the advent of artificial intelligence and machine learning techniques into the field has made possible the development of segmentation/tracking tools able to learn from training datasets and improve from experience without the need of explicit programming or parameter tweaking, e.g. CellProfiler[118] or more recently SuperSegger[119]. Here, following these ideas, we present a detailed protocol that utilizes an open-source ImageJ (Fiji) plugin[115], the Trainable Weka Segmentation Tool[120], complemented by custom-made open-source Python scripts. The computational methods herein can be used to count and track objects in any series of 16-bit tiff images. We have used these methods to count colonies on agar plates[121] and track cells in microscope images. Here we detail one method of obtaining microscope images, which aims to reduce the training queue and improve the segmentation/tracking process. To give a hands-on experience, we provide a dataset in the context of bacterial growth that was obtained using this method https://osf.io/75avy (DOI: 10.17605/OSF.IO/75AVY).

4.2. Operating System, Software, and Data Repository

All of the following are open-source, and downloads are available online (see Table 4 and Table 5).

1. Ubuntu 16.04 LTS, a Linux operating system. Alternatively, Mac OS X can also be used to run the pipeline. To run the pipeline on other operating systems a virtual machine[122] can be installed to use Ubuntu.
2. Fiji ImageJ, an open source Java image-processing program[115].

3. Anaconda 2.7 is an open source distribution of Python, a programming language that has a wide range of tools and libraries for image analysis, including SciPy and NumPy (see Note 1). Our scripts have exclusively been tested with Python 2.7.

Table 4. List of all software and operating systems.

Software	**Function**	**Website**
Ubuntu 16.04 LTS	Linux operating system	https://www.ubuntu.com/download [123]
VirtualBox (optional)	Virtual machine to run Linux environment	https://www.virtualbox.org/wiki/Downloads [122]
Fiji ImageJ (includes Weka segmentation tool)	Classification of images using machine learning	https://imagej.net/Fiji/Downloads [115,124]
Anaconda (Python 2.7)	Open source distribution of Python and related packages (including NumPy and SciPy)	https://www.anaconda.com/download/ [125]
OpenCV	A Python package with tools for image analysis (not included in Anaconda). On a Linux machine, install through Anaconda using the command “conda install opencv” in the terminal.	https://pypi.python.org/pypi/opencv-python [126]
avconv	Software package for handling videos	https://libav.org/avconv.html#Description **[127]**

4. OpenCV is a Python package required for the pipeline that is not included in the initial Anaconda download (Table 5). Only OpenCV downloaded through Anaconda using the command “conda install opencv” in the terminal has been tested to work with our scripts (see Note 2).

5. Avconv, a library for video and audio conversion. This library can be installed in Ubuntu using the following command in the terminal: sudo apt install libav-tools. To install Avconv for Mac OS X the command is "brew install libav" (Homebrew must be installed prior to installing Avconv).

6. Scripts (Table 5) and 16-bit tiff images. All scripts and sample imaging datasets are available at: https://osf.io/75avy (DOI: 10.17605/OSF.IO/75AVY).

Table 5. List of custom scripts. All scripts include comments to facilitate modification. Scripts can be downloaded at GitHub: https://github.com/hdeter/CellTracking [128] or at the public repository http://osf.io/gdxen/ [129].

Script	Section	Language	Brief Description
SegmentandTrack.py	4.3	Python	The master script to run the pipeline based on user input
Image_alignment.py	4.3.2	Python	Aligns images based on differences calculated through FFT
Segmentation.ijm	4.3.3	Fiji macro (ijm)	Calls Trainable Weka Segmentation tool and can be used to train or apply classifiers
Batch_segment.bsh	4.3.3	BeanShell	Called by RunWeka.py to segment a batch of images
RunWeka.py	4.3.3	Python	Calls Segmentation.ijm and Batch_segment.bsh
TrackCellLineages.py	4.3.4	Python	Labels a binary mask and calculates the differences between a given cell to cells within a given area in the previous image. Saves single-cell data, finds cell trajectories and identifies cell lineages. Labels lineages from the first frame and outputs lineage data.
Lineage_analysis.py	4.3.4	Python	Outputs csv files with frame-by-frame data for lineages tracked in TrackCellLineages.py
Image_analysis_stack.py	4.3.5 & 4.3.6	Python	Analyzes global or ROI data

*CSV files output by other scripts will also be output when running SegmentandTrack.py

4.3. Methods

The processing pipeline consists of a series of scripts that were designed to analyze a swath of single-cell datasets (Table 5). These scripts are designed to work on 16-bit tiff images with filenames in the following format: name, 6 digit number, xy, 1 digit number, c, 1 digit number; e.g. **name**000001**xy**1**c**1**.tif**. Phase and fluorescence images must be separate (not stacked), and are differentiated by the number following ‘c’ (e.g. name000001xy1c**1**.tif). Modification of the scripts is required to use alternative naming systems.

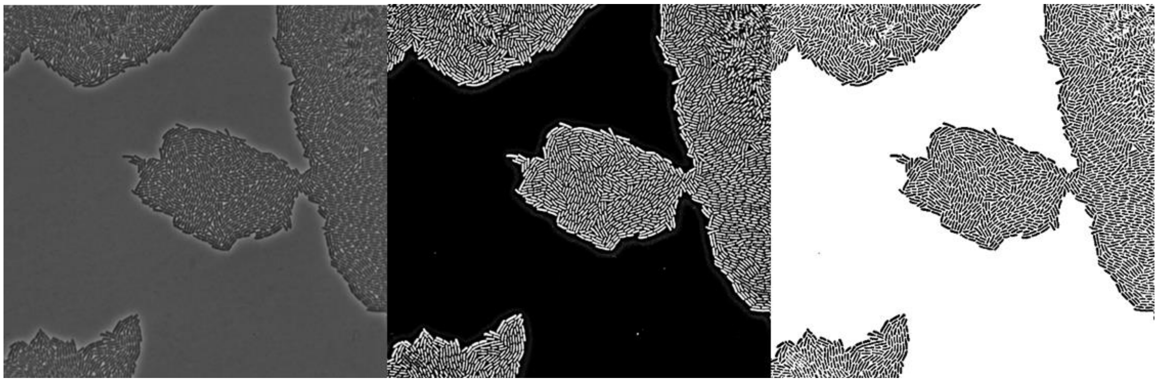


Figure 6. An example of cell segmentation using our method. Left: phase image. Center: Mask 1, a probability mask based on the classification of the phase image. Right: Mask 2, a binary mask based on the classification of Mask 1.

We have developed specific custom scripts that utilize open-source software for cell segmentation (Figure 6) and lineage tracking (Figure 7). To facilitate use, we provide SegmentandTrack.py, a Master Script to run the entire pipeline based on user input (Table 5). The pipeline (see Table 5) has been tested using an Ubuntu 16.04 LTS operating system (recommended) and Mac OSX. For convenience, place the scripts in a folder that contains the images to be analyzed in a subfolder. It is essential to read the entire protocol before running the analysis pipeline for optimal operation.

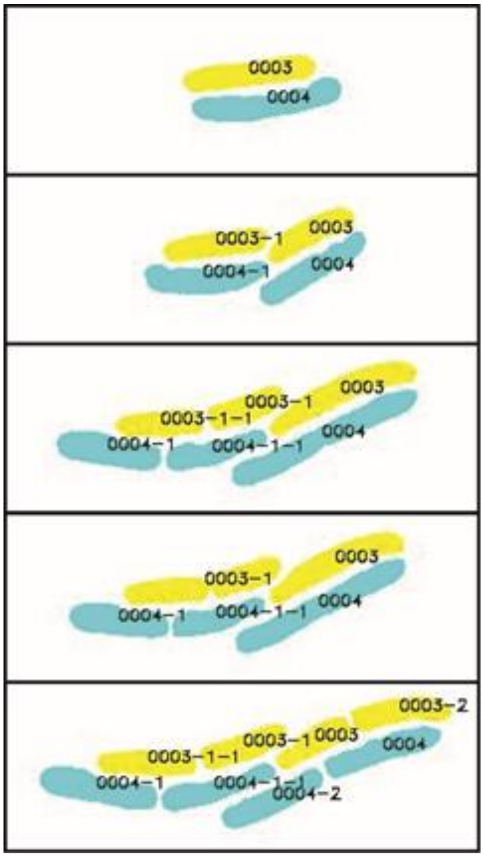


Figure 7. Cell lineages are kept track of by renaming trajectories that overlap with another trajectory and therefore have a common ancestor (see Video 2). Cells in the top images were tracked, labeled, and the output quantified as the cells divide (bottom images) using this method.

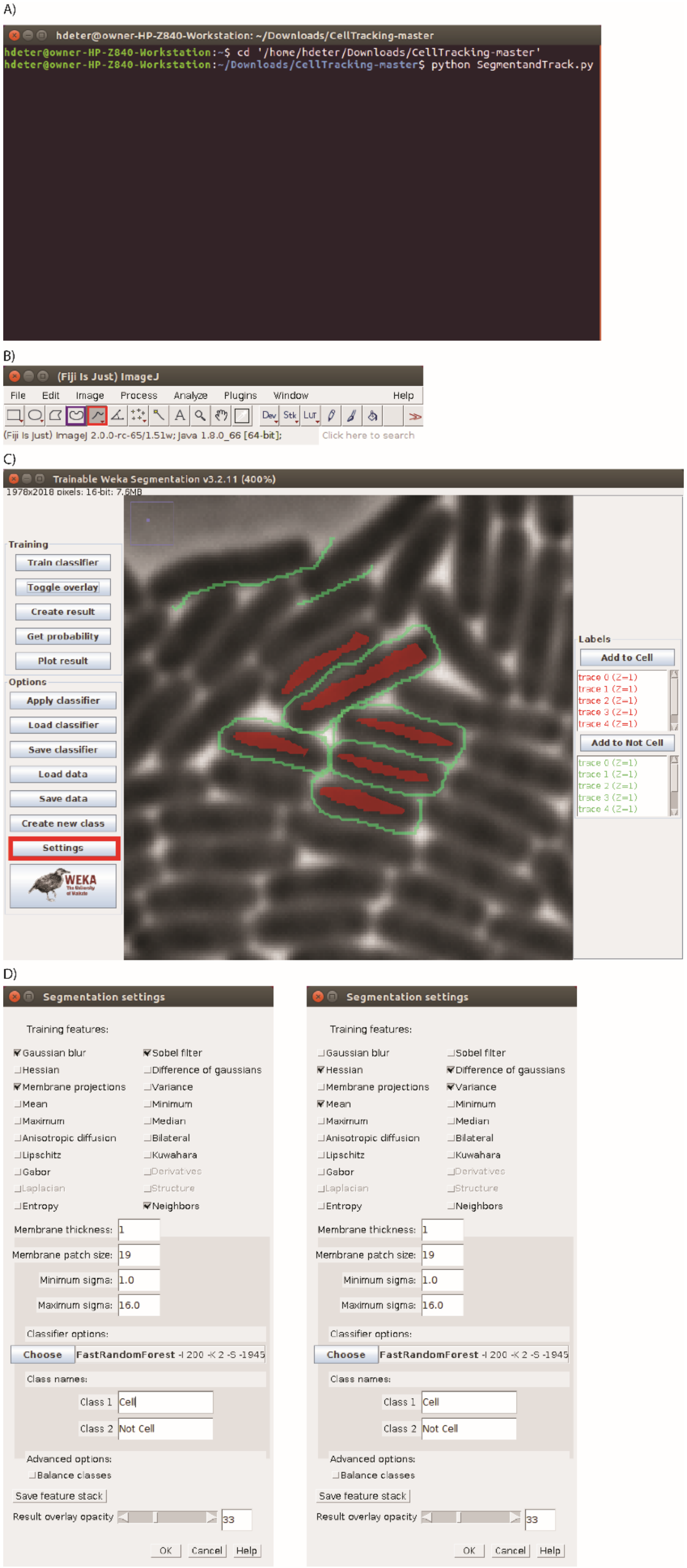
To run SegmentandTrack.py (see Video A1 for a video tutorial):

1. Open a terminal. Keep in mind that the terminal is case-sensitive.

2. Change directories to the folder that contains the scripts and the images that are to be analyzed (Figure 8A). Note that directories will differ based on individual machine and file locations (see Note 4).

3. Type “python SegmentandTrack.py” (Figure 8A).

4. Answer user prompts. For yes or no (Y/N) questions, answer “y” for yes or “n” for no. As an alternative to using the terminal for user prompts, see section 4.3.7 Option to use comma separated values (csv) file instead of terminal prompts. However, we recommend that first time users do not use a csv file.



A)

```
hdeter@owner-HP-Z840-Workstation: ~/Downloads/CellTracking-master
hdeter@owner-HP-Z840-Workstation:~$ cd '/home/hdeter/Downloads/CellTracking-master'
hdeter@owner-HP-Z840-Workstation:~/Downloads/CellTracking-master$ python SegmentandTrack.py
```

B)

(Fiji Is Just) ImageJ 2.0.0-rc-65/1.51w; Java 1.8.0_66 [64-bit];

C)

1978x2018 pixels; 16-bit; 7.6MB

trace 0 (Z=1)
trace 1 (Z=1)
trace 2 (Z=1)
trace 3 (Z=1)
trace 4 (Z=1)

trace 0 (Z=1)
trace 1 (Z=1)
trace 2 (Z=1)
trace 3 (Z=1)
trace 4 (Z=1)

D)

Training features:

☑ Gaussian blur ☑ Sobel filter
☐ Hessian ☐ Difference of gaussians
☑ Membrane projections ☐ Variance
☐ Mean ☐ Minimum
☐ Maximum ☐ Median
☐ Anisotropic diffusion ☐ Bilateral
☐ Lipschitz ☐ Kuwahara
☐ Gabor ☐ Derivatives
☐ Laplacian ☐ Structure
☐ Entropy ☑ Neighbors

Membrane thickness: 1
Membrane patch size: 19
Minimum sigma: 1.0
Maximum sigma: 16.0

Classifier options:
FastRandomForest -I 200 -K 2 -S -1945

Class names:
Class 1 Cell
Class 2 Not Cell

Advanced options:
☐ Balance classes

Result overlay opacity 33

Training features:

☐ Gaussian blur ☐ Sobel filter
☑ Hessian ☑ Difference of gaussians
☐ Membrane projections ☑ Variance
☑ Mean ☐ Minimum
☐ Maximum ☐ Median
☐ Anisotropic diffusion ☐ Bilateral
☐ Lipschitz ☐ Kuwahara
☐ Gabor ☐ Derivatives
☐ Laplacian ☐ Structure
☐ Entropy ☐ Neighbors

Membrane thickness: 1
Membrane patch size: 19
Minimum sigma: 1.0
Maximum sigma: 16.0

Classifier options:
FastRandomForest -I 200 -K 2 -S -1945

Class names:
Class 1 Cell
Class 2 Not Cell

Advanced options:
☐ Balance classes

Result overlay opacity 33

Figure 8. The Weka settings for our classifiers. Here we highlight default settings that have worked well in the past, but these parameters can be altered depending on the user requirements. **(A)** An example of commands to run a python script in the terminal. Note that the directory the script is running from can be different. **(B)** The freehand selection tool (Purple) is useful for selecting cells for cell segmentation. The line tool (Red) is useful for outlining cells and dividing cells to add to the "Not cell" label (Note 13). **(C)** An example image loaded in Fiji for training in Weka by going to "Plugins" tab, then to "Segmentation" and then to "Trainable Weka Segmentation." To adjust the parameter and to make labels click on "Settings" (Red). The image loaded into Weka has magnified to demonstrate cell selection (see Note 13) **(D)** Left: The settings used to classify the phase images to a probability mask. Right: The settings used to classify the phase images to a binary mask (see Note 9). To follow our example, we change the Weka default names from "Class 1" and "Class 2" to "Cell" and "Not cell" (see Note 13).

4.3.1. Selecting a Region of Interest (ROI) (optional)

Regions of interest (ROIs) can be used to crop images or select a specific region for a particular type of analysis. SegmentandTrack.py offers two options that use an ROI (see sections 3.5 and 3.9), which relies on a csv (comma-separated values) file created using the Fiji Measure tool as follows (see Video 1 for a video tutorial):

1. Open the desired image in Fiji. To observe the same region throughout multiple images import an image sequence using “File” ➔ “Import” ➔ “Image Sequence.”
2. Remove any scale associated with the images using the “Set Scale” feature under the Analyze tab (see Note 5. It is essential that the scale of the bounding rectangle be in pixels because the scripts imports the values as pixel values and therefore any other scale would lead to the ROI being different from intended. In the case of image alignment (see section 0), this could unintentionally result in a region that includes moving objects, which will hinder the alignment (see Note).
3. Set measurements (under the Analyze tab) to only measure the bounding rectangle (“Bounding rectangle” should be the only checked box) with 0 decimal places (the results need to be whole, even numbers; see Note 6).
4. Select the desired region with the rectangular selection tool and use the Measure tool, located in the Analyze tab (Figure **8**B). Save the results as a csv file (e.g. filename.csv) in the same directory as the scripts.

4.3.2. Image Alignment (optional)

Image alignment is an optional image pre-processing step, yet it is essential if there are significant shifts between phase images, because cell tracking is accomplished by comparing cell locations between consecutive frames. Alignment is not required if the image registration does not shift significantly. If running the alignment on an image with a large number of moving objects we suggest using an ROI during the alignment (see Note 7). Align the images as follows (see Video A1 for a video tutorial):

1. Answer “y” when asked, “Do you wish to align images?”
2. If using an ROI to base the alignment upon (optional), make an ROI file (see section 0) that indicates a stationary area of the image, (the largest possible background region that contains a minimal number of moving objects throughout the images of the image set; see Note 7).
3. If using an ROI, Answer “y” when asked, “Do you have a ROI file for a stationary area?” If aligning based on the whole image answer “n” for the same prompt.
4. If using an ROI, Enter the path to the csv file, relative to the working directory.
5. Enter the name of the directory into which images will be saved.

4.3.3. Cell Segmentation

Single cell analysis is primarily dependent upon cell segmentation; we use the Trainable Weka Segmentation tool in Fiji. Weka uses machine learning to train a classifier based on training data selected by the user[130]. Section 4.3.3 covers how to use Weka to classify images, and we have had success in the past using two rounds of classification because this method improves segmentation of neighboring cells (see Note 9). Our custom script make the process of training and applying a classifier faster and more direct (see Note 8; see Video A1 for a video tutorial).

1. To use Trainable Weka Segmentation to classify images through SegmentandTrack.py. Answer “y” when asked, “Do you wish to train and/or apply a classifier?”

2. Input the full path to the Fiji executable file located in the Fiji.app directory when prompted (see Note 4).
3. Enter how many rounds of classification you are running (1 or 2); we recommend two rounds (see Note 9).
4. If using a Linux machine, answer “y” if you would like to classify the images in the background (it is faster). Then enter how many processes are available for multiprocessing. The number of threads available for these processes will depend on the number of processors available to the individual machine (less if using a virtual machine). Do not use more than half of the available threads for multiprocessing.
5. To train a classifier on a subset of phase images answer “y” when asked by SegmentandTrack.py, “Do you have a trained classifier? (Y/N)”. Fiji will then open and run user prompts in the GUI.
6. Answer the Fiji prompts. If you click “Yes” in response to “Are you classifying a phase image?” the script uses a Bandpass filter to subtract background and outputs a probability mask (first round of classification; see Note 10). When you click “No” (i.e. when classifying the probability mask for the second round of classification), the script does not run a Bandpass filter on the image and outputs a binary mask (which the pipeline uses for cell tracking).
7. In the “Trainable Weka Segmentation” window (also located under “Plugins” ➔ “Segmentation”), change the settings as desired (Figure **8**C). Although there are a few different classifiers available, we have had success in the past using the Fast Random Forest classifier, which is the Weka default setting. The training features

in Weka should be adjusted based on the image dataset to improve classification (see Note 11). Figure **8**D shows the settings we used with this dataset (see Note 12).

8. Select regions of your image and add them to the appropriate label (e.g. “Cell” or “Not cell”) then train the classifier (see Note 13). For cell segmentation, the freehand selection tool (Figure **8**B) is useful for selecting cells. To indicate the separation of cells that are recently divided, draw a line using the freehand line tool and add the line to the “Not cell” label (Figure **8**C; see Note 14).
9. Repeat step 8, selecting data based on the results of the classifier (see Note 14).
10. Save the data and classifier (see Note 14).
11. Press OK in the dialog box. This causes the script to prompt the user for filenames and then open another image in Trainable Weka Segmentation and load the previously saved classifier and data.
12. Continue to train the classifier on multiple images by repeating steps 8 through 12 (see Note 14). Be sure to save (step 10) both the classifier and data after every round of training.
13. To finish training the classifier select “No” when asked, “Do you wish to continue training the classifier?” This will trigger the script to close the open windows in ImageJ and continue in the terminal.
14. In the terminal, enter the path to the classifier, relative to the working directory.
15. Enter the name of the directory within your image directory into which the masks will be saved.
16. Based on this input the script segments the images. If using two rounds of classification, you will need to repeat steps 5-13 using the probability masks. The

first round of classification will produce a probability mask and the second will produce the binary mask (Figure **6**, see Note 9).

4.3.4. Obtaining Single Cell Data and Cell Lineages

Once images have been segmented cells can be tracked based on overlapping regions across images. Tracking is primarily limited by cell segmentation, so improving cell tracking typically requires further training of the classifier (see section 4.3.3). Cell division events are also tracked, enabling both individual cell movements and cell movement to be tracked throughout the image dataset (Figure 7).

1. Answer “y” when asked, “Do you wish to track cells?”
2. Enter the minimum and maximum cell areas (in pixels) for tracking. If you would like to determine values that work well for your data set, you can measure the area of the cells, background artifacts or features with the measure tool in Fiji (see Note 15).
3. Enter the minimum number of frames through which a trajectory must be tracked to be included in the analysis (see Note 16).
4. SegmentandTrack.py will then run the analysis. This analysis uses the binary masks created during cell segmentation (see section 4.3.3; Note 9) and outputs csv and pickle files containing single cell and lineage data (Table 5; see Note 18). The lineage data includes the mean cell doubling time for a given lineage (see Note 19). Each lineage is numbered, with dashes to indicate a branch (e.g. Lineage 0001-1 divided from lineage 0001, see Video A2 and Video A3).
5. If desired, our script can also output csv files with individual lineage information. Enter “y” when asked, “Do you wish to output csv files detailing data for individual

lineages?" to obtain frame-by-frame data. Then either answer "y" when asked, "Do you wish to get data for all of the lineages?" or answer "n" and you will be prompted to input the number of lineages you wish to analyze and then input each lineage name separately (see Note 20).

6. Advanced users can further analyze the data using the pickle files (see Note 20).

4.3.5. Measuring fluorescence

Fluorescence is reported in arbitrary units (AU) and based on the mean or the median pixel intensity of the fluorescence image over the area. Pixel intensity of 16-bit images can be measured using Fiji (the Measure tool located under the Analyze tab; shortcut key: Ctrl/Cmd+M), or by converting the image to a NumPy array and getting the value of the desired pixels. Our custom scripts utilize the latter and report raw fluorescence data (no background subtraction; see Note 21. There are a few options for filtering the data and subtracting background available within the script (see). Single cell fluorescence is analyzed when tracking cells (see section 4.3.4).

To analyze the whole image and ROI fluorescence:

1. Answer "y" when asked, "Do you wish to analyze images?"
2. If you wish to analyze an ROI, answer "y" when asked, "Do you wish to analyze a region of interest?"
3. Enter the path to the ROI file to analyze (see section 0), relative to the working directory. A different ROI than the one used for image alignment is required.
4. Fluorescence data will be output as a csv file into the working directory.

4.3.6. Video Rendering

The pipeline run by SegmentandTrack.py uses avconv to render videos from an image directory (see Note 30).

1. Answer “y” when asked, “Do you wish and analyze images?” and “Do you wish to render videos?” to combine the phase contrast and fluorescent channels (see Note 22. Any processing of the images (e.g. background subtraction) must be done before rendering the videos when using these methods. Background subtraction is included in the “filtered” results when analyzing whole image fluorescence (see Note).
2. To crop the video to an ROI answer “y” when asked, “Do you want to crop the images based on an ROI?” Enter the path to the ROI file to analyze (see section 0), relative to the working. A different ROI than the one used for image alignment is required, but the same ROI must be used for analyzing fluorescence of an ROI (see section 4.3.5).
3. Cells can be numbered based on lineage (see Video A2) if you answer “y” when asked, “Do you want to number the cells in the images based on lineage tracking?”
4. Binary masks can also be utilized to contour (outline) the cells (Video A4) if you answer “y” when asked, “Do you want to outline cells based on masks?”

4.3.7. Option to use comma separated values (csv) file instead of terminal prompts

To facilitate frequent and rapid use of the software, we have included an option to use a comma separated values (csv) file to answer user prompts. The csv file is included with the files available in the repository https://osf.io/75avy (DOI: 10.17605/OSF.IO/75AVY).

4.4. Notes

1. To install the Anaconda shell file in Linux (see Table 4) type the following commands in the terminal and replace “path/to/script.sh” with the path to your shell file: 1) sudo chmod +x path/to/script.sh, 2) /path/to/script.sh. Then you must add Anaconda to the path in Linux (so that conda commands can be run through the terminal) enter the command “export PATH=~/anaconda2/bin:$PATH” into the terminal.

2. The scripts have only been tested to work using OpenCV (version 3.1.0) downloaded through Anaconda. Using a different version of OpenCV may result in an error reading “ImportError: No module named cv2.”

3. The scripts are hardcoded to be specific to the file naming system herein described (see section 4.3) and to analyze data without fluorescence or containing a single fluorescence channel. The code is capable of analyzing more than one fluorescence channels, but requires some editing to output data.

4. To find the directory in which a file is located in Ubuntu, right click on the file and select “Properties.” The directory the file can be found in “Location” under the “Basic” tab (e.g. /home/user/Downloads). You can copy and paste this path into the terminal using the mouse.

5. It is essential that the scale of the bounding rectangle be in pixels because the scripts imports the values as pixel values and therefore any other scale would lead to the ROI being different from intended. In the case of image alignment (see section 0), this could unintentionally result in a region that includes moving objects, which will hinder the alignment (see Note 7).

6. The python scripts Image_alignment.py and Image_analysis_stack.py require bounding box input to be whole number integers. Additionally, avconv cannot render a video using images with an odd length or width, and therefore any ROI that is used to crop the images in Image_analysis_stack.py must have an even length and width in pixels to output videos.

7. Fast Fourier transform (FFT) alignment works best on regions that have limited change between images, and therefore moving objects (such as cells) should be minimally included in the region to be aligned. When selecting an ROI in Fiji (see section 0), use an image stack to scroll through multiple images and ensure that there is minimal movement within the selected area.

8. SegmentandTrack.py provides the option to run classification without training a new classifier, which allows a classifier to be used across multiple experiments. Furthermore, Segmentation.ijm can be run in Fiji, independently of the python scripts, to call Weka and load images and classifiers for training.

9. The purpose of the segmentation is to create a binary mask that can then be used to label and identify single cells. However, an initial probability mask gives a more nuanced picture of the classification results than a binary mask and allows for a second round of classification (Figure **6**). We have empirically determined that two rounds of classification can result in a final segmentation that is more accurate and sometimes faster than when classifying with a single, larger classifier. However, one round of classification may be sufficient for certain datasets.

10. Subtracting Background: In Fiji, we use the Bandpass filter plugin, which removes high and low spatial frequencies, to regularize the image and subtracts the background.

The plugin is located under “Process” ➔ “FFT”. We have had previous success with large structures filtered to 100 pixels, small structures filtered up to 0 pixels, no suppression and a 5% tolerance. The Bandpass filter is included in the custom Fiji scripts Segmentation.ijm and Batch_segment.bsh for the first round of classification.

11. Weka settings are saved with the data and therefore can only be set when training the classifier on the first image. Further information on Weka settings is on the Fiji ImageJ website: https://imagej.net/Trainable_Weka_Segmentation [124,130]. Additionally, the Weka Explorer is a tool provided by Weka to aid in determining which classifiers or training features should be used for a given dataset. It can be accessed by clicking on the Weka logo in Trainable Weka Segmentation.

12. The training features we used for our classification were different for each round of classification due to the differences between the images being classified. For the first classifier (phase to probability mask) we used Gaussian blur, Sobel filter, Membrane projections and Neighbors. For the second classifier (probability mask to binary mask) we used Hessian, Difference of Gaussians, Variance, and Mean (Figure 8D).

13. In the “Settings” of Trainable Weka Segmentation, the labels can be named to help guide the user. We call label 1 “Cell” and label 2 “Not cell” wherein “Not cell” includes anything that is not a cell, including background, features, etc. (Figure 8CD). The same method can be used for identifying other objects in an image; for example, labeled organelles could be identified and tracked in eukaryotes.

14. For good-quality segmentation results, it helps to outline just outside the edges of a cell and add it to the “Not cell” label (Figure 8C). When training the classifier, it is important not to overfit the data (i.e. when the classifier matches a training set closely but

is no longer applicable to the more extensive dataset). Furthermore, extensively training the classifier can slow the classification process with minimal returns on efficacy. To prevent this, it is important to save classifiers and data intermittently (with sequential names); in case a later classifier results in a decrease in efficacy. Our classifiers were trained on 5-15 cells in every 20-30 images.

15. To measure an area in Fiji, first set measurements ("Analyze" ➔ "Set Measurements") then make a selection and measure it ("Analyze" ➔ "Measure"; Ctrl+"M"). Our results were generated using a minimum value of 100 and a maximum value of 2500 (Video A1 and Video A2).

16. The maximum number of frames is two less than the total number of frames (the first and last frames are not included in the analysis). Currently, the script is hardcoded so that the overlap requirement to track cells between frames is at least half and to end a trajectory if the cell decreases in area by more than 40%. Changing these values (see comments in TrackCellLineages.py) can increase or decrease the fidelity of the tracking.

17. We track cells from the last frame to the first frame, because the process of cells merging are more apparent than cells dividing. Currently, the distance radius to test for cell overlap (THRESHOLD in TrackCellLineages.py) is set to 150, but this can be adjusted based on the dataset by editing the script. A smaller radius can speed up the process, while a larger one may be necessary for larger or faster moving cells.

18. The csv files can be opened using Microsoft Excel, LibreOffice Calc, and most other spreadsheet software. Pickle files are used to pack and unpack Python objects. The pickle files produced by TrackCellLineages.py (lineagetracking.pkl and

lineagetrackingsummary.pkl) can be unpacked and used for further analysis by advanced users; see Lineage_analysis.py for an example.

19. Our method determines doubling time based on two observed divisions, and therefore a lineage must divide at least twice in the course of the experiment to determine doubling time. The doubling time output by Track-cell-lineage.py is the mean of the amount of time between each division over the entire lineage.

20. Lineage names are output in lineagedata.csv, and visual output (masks with cells colored and labeled according to their lineage) is available in the Lineages subfolder, created within the folder containing the analyzed images (Video A2). Outputting all of the files at once (answering "y" when asked, "Do you wish to get data for all of the lineages?") is often faster than typing in multiple individual names.

21. There are a few options for filtering the data and subtracting background available within the script (see Note 22. Any processing of the images (e.g. background subtraction) must be done before rendering the videos when using these methods. Background subtraction is included in the "filtered" results when analyzing whole image fluorescence (see Note 22. Any processing of the images (e.g. background subtraction) must be done before rendering the videos when using these methods. Background subtraction is included in the "filtered" results when analyzing whole image fluorescence (see Note), but modifications are required to use them for single cell tracking. Global fluorescence data reports unfiltered and filtered data, wherein filtered refers to data that has been normalized so that the maximum fluorescence measurement within the frame is 1 and the minimum is 0.

22. Any processing of the images (e.g. background subtraction) must be done before rendering the videos when using these methods. Background subtraction is included in the “filtered” results when analyzing whole image fluorescence (see Note 23). There are a few different filters to choose from in Image_analysis_stack.py to subtract the background. These can be changed by adjusting plot_filterIndex with Image_analysis_stack.py.

23. Image_analysis_stack.py includes functions for adjusting image brightness, scaling the image values and adjusting color. The parameters for these adjustments can be modified based on the dataset as described by the script comments.

5. PROTEOLYTIC QUEUES AT CLPXP INCREASE ANTIBIOTIC TOLERANCE

Reproduced with permission from Deter *et al.* 2020 published in *ACS Synthetic Biology*[121]. Copyright 2020 American Chemical Society.

Authors: Heather S. Deter[1], Alawiah H. Abualrah[1], Prajakta Jadhav[1], Elise K. Schweer[1], Curtis T. Ogle[2], and Nicholas C. Butzin[1]
[1] Department of Biology and Microbiology. South Dakota State University. Brookings, SD. 57006. USA.
[2] Independent researcher. Prosser, WA, 99350. USA.

5.1. Abstract

Antibiotic tolerance is a widespread phenomenon that renders antibiotic treatments less effective and facilitates antibiotic resistance. Here we explore the role of proteases in antibiotic tolerance, short-term population survival of antibiotics, using queueing theory (i.e. the study of waiting lines), computational models, and a synthetic biology approach. Proteases are key cellular components that degrade proteins and play an important role in a multi-drug tolerant subpopulation of cells, called persisters. We found that queueing at the protease ClpXP increases antibiotic tolerance ~80 and ~60 fold in an E. coli population treated with ampicillin and ciprofloxacin, respectively. There does not appear to be an effect on antibiotic persistence, which we distinguish from tolerance based on population decay. These results demonstrate that proteolytic queueing is a practical method to probe proteolytic activity in bacterial tolerance and related genes, while limiting the unintended consequences frequently caused by gene knockout and overexpression.

5.2. Introduction

The discovery of penicillin in the 1920s led to a new age of human and animal medicine as many antibiotics were quickly identified and developed, but the subsequent explosion of antibiotic treatments and applications has simultaneously driven microbial evolution and the development of widespread resistance[131,132]. A significant contributing factor to the abundance of antibiotic-resistant microorganisms are subpopulations of cells that survive of antibiotic treatment without a genetic mutation, antibiotic tolerant and persistent cells[36,38] . Persistence is a physiological state that enables cells to survive antibiotic treatment via temporary changes in phenotype, such as slowed growth and biosynthesis, rather than genotype (e.g. antibiotic resistance)[55]. Although persistence has been studied for over 70 years, there has been a lack of specificity in the literature between antibiotic tolerance and persistence[54,55]. Recently, a consensus statement that was released after a discussion panel with 121 researchers defined antibiotic persistence as a tolerant subpopulation of cells that result in a distinct phase of population decay[55]. We use population decay to differentiate between tolerance and persistence in this work (Figure 9A).

The widespread nature of persistence suggests that similar mechanisms exist to trigger the persistent state in prokaryotes. These

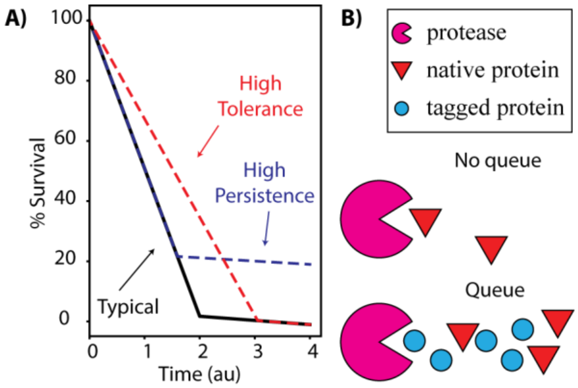


Figure 9 A) Examples of population decay in typical (black), high persistence (blue) and high tolerance (red) populations. A shift in tolerance can be distinguished from a change in the number of persisters. For example, a high persistence population can initially have the same decay rate as a typical population but have higher survival because of more persisters (dotted blue line). A high tolerance population can have the same persister level as a typical population but have a shift in the initial decay rate (dotted red line). **B)** A simple model of proteolytic queueing. When native proteins have low competition for the protease, there is no queue. Induction of synthetic tagged proteins competes with the native proteins for the protease and overloads the protease, which results in a proteolytic queue (bottleneck).

mechanisms include many common systems, e.g. toxin-antitoxin (TA) systems and proteases. Although the precise role of TA systems in persistence is unclear due to the complications of knocking out all TA systems (*E. coli* has >45 known and predicted TA systems[78,133,134]) and their interrelated role in cellular responses to stress[28], toxins in TA systems can trigger persistence when at a higher level than their cognate antitoxin[28,135,136]. Within the cell, the ratio of toxin to antitoxin is regulated during protein production[18,20,60] and through degradation by proteases[10,13]. Proteases, such as Lon and ClpP, are largely responsible for protein degradation and cell maintenance[50,51]. They provide an essential level of protein regulation throughout the cell, including degradation of RpoS (a transcription factor that responds to stress)[52] and tagged polypeptides (incomplete proteins) synthesized by stalled ribosomes that have been rescued by the trans-translation system[53]. In *E. coli*, *ssrA* (tmRNA) and *smpB* are the primary genes responsible for trans-translation, a cellular mechanism for recovering stalled ribosomes. A tmRNA molecule acts as a tRNA by binding to the A-site of a stalled ribosome. The ribosome then translates the protein-coding region of the tmRNA, which adds an amino acid tag to target the polypeptide for degradation by ClpXP[53]. While *ssrA* is not essential in *E. coli*, *ssrA* knockouts cause growth defects, increase susceptibility to certain antibiotics[137], and affect persistence[28,138-140]. Proteases and related chaperones are also consistently identified as persister-related genes in gene knockout experiments[47,48] and transcriptome analysis[49]. Indeed, a drug that targets persisters, acyldepsipeptide (ADEP4), activates the protease ClpP and lowers persister levels[56]. While most published articles focus on methods that reduce persister levels, conditions that increase their levels are integral to understanding the causative mechanisms of action and developing new drugs. As many persister studies incidentally examine antibiotic tolerance[54,55], it follows that some of the above mechanisms may play a role in antibiotic tolerance.

Synthetic biology takes advantage of these mechanisms to develop new cellular circuits. For example, synthetic oscillators require rapid degradation of proteins, which is accomplished using the *ssrA* degradation tag[59,141,142]; the ssrA degradation tag is the amino acid sequence AANDENYALAA[53], which we abbreviate to LAA throughout. Previous work establishes that multiple circuits can be coordinated by overproduction of a common degradation tag to target proteins to a protease[143,144]. When a protease is overloaded, protein species compete for degradation; the enzyme is unable to keep up with the influx of new proteins[57]. This phenomenon can be explained by queueing theory, in which one type of customer competes for processing by servers, which has traditionally been applied to systems such as computer networks and call centers. Limited processing resources in a cell (e.g. proteases) cause biological queues[141,145] (Figure 9B). The queueing effect at the protease ClpXP is essential in allowing for oscillation of the highly used synthetic oscillator (often called Stricker oscillator or dual-feedback oscillator)[59,142]. Variations of this oscillator have been used in different strains of *E. coli*[59,143,144,146], and in *Salmonella* ser. Typhimurium[147], indicating that queueing at ClpXP is not specific to one strain or species. The coupling of otherwise independent synthetic systems via proteolytic queueing demonstrates that queueing affects protein degradation and thus provides a tunable method of studying proteolytic degradation with little effect on cell growth[141,143-145] compared to gene knockouts and overexpression of proteases[50,148,149].

We set out to test the hypothesis that proteolytic queueing at the ClpXP complex affects survival of *E. coli* during antibiotic treatment. Previous studies have used knockout mutants to disrupt activity of specific proteases in *E. coli,* but these studies yielded mixed results[48,139,150,151]. The variability between results of knockout mutations could be due to differences in growth rates and metabolism, which would modulate antibiotic efficacy[152,153]. Proteases are essential to regulating many biological networks and simply removing them likely has downstream effects. For example, ClpXP is known to degrade at least 50 proteins in *E. coli*[154], and many of them are transcription factors like RpoS, the global regulator of stationary phase[155,156]. Many proteins are regulated at the proteolytic level by ClpXP[51], including RpoS[157], and simply removing ClpXP disrupts this regulation and any quantification of persistence or tolerance is indirectly measuring an alteration in the levels of proteins regulated by ClpXP degradation.

Proteolytic queueing is preferred over protease knockouts when probing antibiotic efficacy because while protease knockouts often result in growth defects[50,148], proteolytic queueing does not noticeably affect cell growth or death[141,143-145] (Figure A4). Our results show that during antibiotic treatment, degradation plays a role in cell survival and the effect is tunable using queue formation. Proteolytic queueing at ClpXP increases antibiotic survival and analysis of population decay with and without a queue demonstrates that queueing specifically increases antibiotic tolerance. We hypothesize that the queue is affecting the degradation of one or many regulatory molecules within the cell that cause downstream effects and enhance antibiotic tolerance. These results demonstrate that proteolytic queueing provides a new method to probe proteolytic activity in antibiotic tolerance and persistence.

5.3. Results

5.3.1. Proteolytic queueing affects tolerance

Cultures were grown to stationary phase and incubated for 24 hours prior to dilution into fresh media containing ampicillin to quantify persistence (see Methods). A proteolytic queue was induced via the production of a ssrA tagged fluorescent protein, CFP-LAA, expressed under an IPTG inducible promoter, $P_{lac/ara-1}$. No apparent change in growth was observed by induction (Figure A4) as reported previously[143,144]. The effects of queue formation on antibiotic survival are shown as the percentage of the population that survived ampicillin treatment (Figure 10). When CFP alone (no degradation tag control) was overexpressed during ampicillin treatment, there was no significant effect on persister levels ($p > 0.2$, Figure 10A). Queue formation (overexpression of CFP-LAA) during ampicillin treatment led to a 25-fold increase in survival after three hours in a concentration-dependent manner (Figure 10B; $p<0.0001$, $n \geq 12$).

When a queue was induced for 24 hours prior to ampicillin treatment the surviving population at three hours was over 80-fold higher than the uninduced population, only if induction was maintained during ampicillin treatment. However, if the inducer was removed during ampicillin treatment, the initial 24 hours of queueing had a minimal effect on survival at three hours ($p>0.01$, Figure 10C). These results indicate that survival was affected by queue formation rather than CFP itself, and that the size of the queue (level and length of induction) determines the effect.

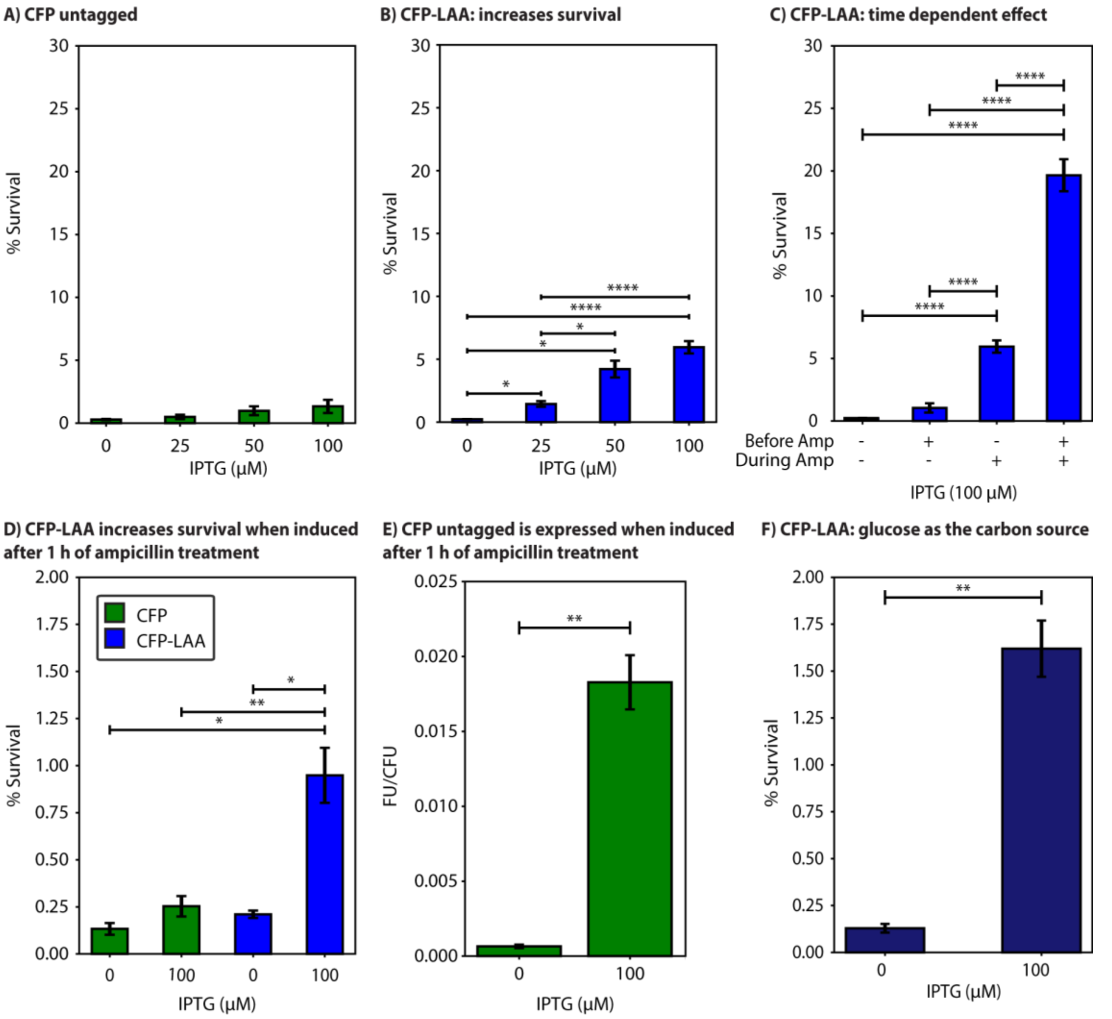


Figure 10. Proteolytic queueing affects survival of cells treated with the antibiotic ampicillin. a. Induction of untagged CFP during antibiotic treatment has no significant effect on survival (p>0.2). b. Induction of CFP-LAA during antibiotic treatment causes an increase in survival. c. CFP-LAA was induced (+) with 100 µM of IPTG or not induced (-). Induction before ampicillin lasted 24 h in stationary phase prior to antibiotic treatment. Queueing affects survival if the queue is maintained during ampicillin treatment. d-e. Expression of CFP or CFP-LAA was induced with IPTG one hour into the three-hour antibiotic treatment. Induction of CFP alone (no queue) had no significant effects on survival. Induction of CFP-LAA increased survival (d). Population fluorescence was measured for untagged CFP after antibiotic treatment, demonstrating that CFP is being produced via induction (e). f. Induction of CFP-LAA during antibiotic treatment causes an increase in survival with glucose as a carbon source rather than glycerol, demonstrating that it is not a solely a carbon-specific phenomenon. Cultures were treated with ampicillin (100 µg/ml). Error bars represent SEM. n ≥ 3. *p<0.05. **p<0.01. ***p<0.001. ****p<0.0001.

To confirm that these results are due to induction during antibiotic treatment, we waited one hour into ampicillin treatment before inducing expression of the fluorescent protein. As we previously observed, induction of untagged CFP had no apparent effect on persister levels (Figure 10D), while quantification of fluorescence after ampicillin treatment confirmed that CFP was produced (Figure 10E). Overexpression of CFP-LAA for two hours of ampicillin treatment still increased cell survival compared to the uninduced and untagged CFP populations (Figure 10D).

We did further testing to confirm this effect is not specific to glycerol as a carbon source or ampicillin as the antibiotic. When glucose was the carbon source rather than glycerol, survival still increased due to CFP-LAA induction (Figure 10F), which demonstrates that the effect is not directly related to the carbon source. We then tested the effects of queueing against the antibiotic ciprofloxacin, because ciprofloxacin targets DNA gyrase[158] while ampicillin targets the cell wall[159]. CFP alone caused a slight increase in survival (Figure 11A), however the CFP-LAA tag led to a 60-fold increase in survival (Figure 11B).

5.3.2. Chloramphenicol inhibits the synthetic queue

Neither ampicillin nor ciprofloxacin directly affect production of the fluorescent protein (i.e. target transcription or translation) and thus should not prevent queue formation. On the other hand, an antibiotic that affects protein production should prevent queue formation, and therefore CFP-LAA induction would not affect survival in the presence of such an antibiotic. We found this to be the case when testing the effects of queueing on the survival of cells treated with chloramphenicol. Chloramphenicol is an antibiotic that inhibits protein translation by binding to bacterial ribosomes and inhibiting protein synthesis, thereby inhibiting bacterial growth[160]. Induction of CFP-LAA does not increase survival of antibiotic treatment when treated with chloramphenicol alone (Figure A5), but chloramphenicol is not bactericidal, so we co-treated cultures with both ampicillin and chloramphenicol. The overall percent survival with chloramphenicol is much higher than with ampicillin alone, which is consistent with the literature[161]. As expected, co-treatment with ampicillin and chloramphenicol had no apparent effect on cell survival, supporting that even when CFP-LAA was induced the queue could not form if translation was blocked (Figure 11C).

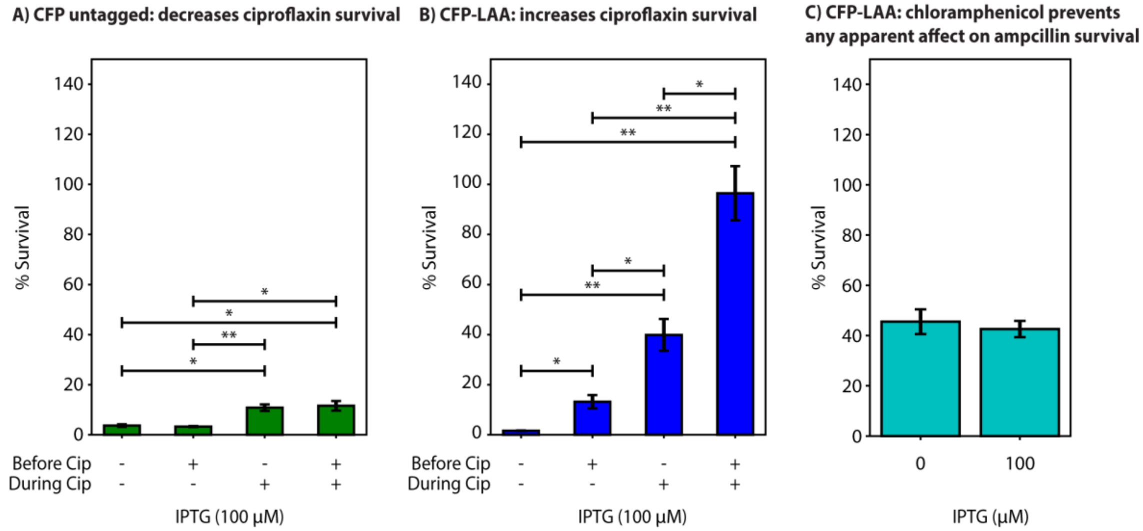


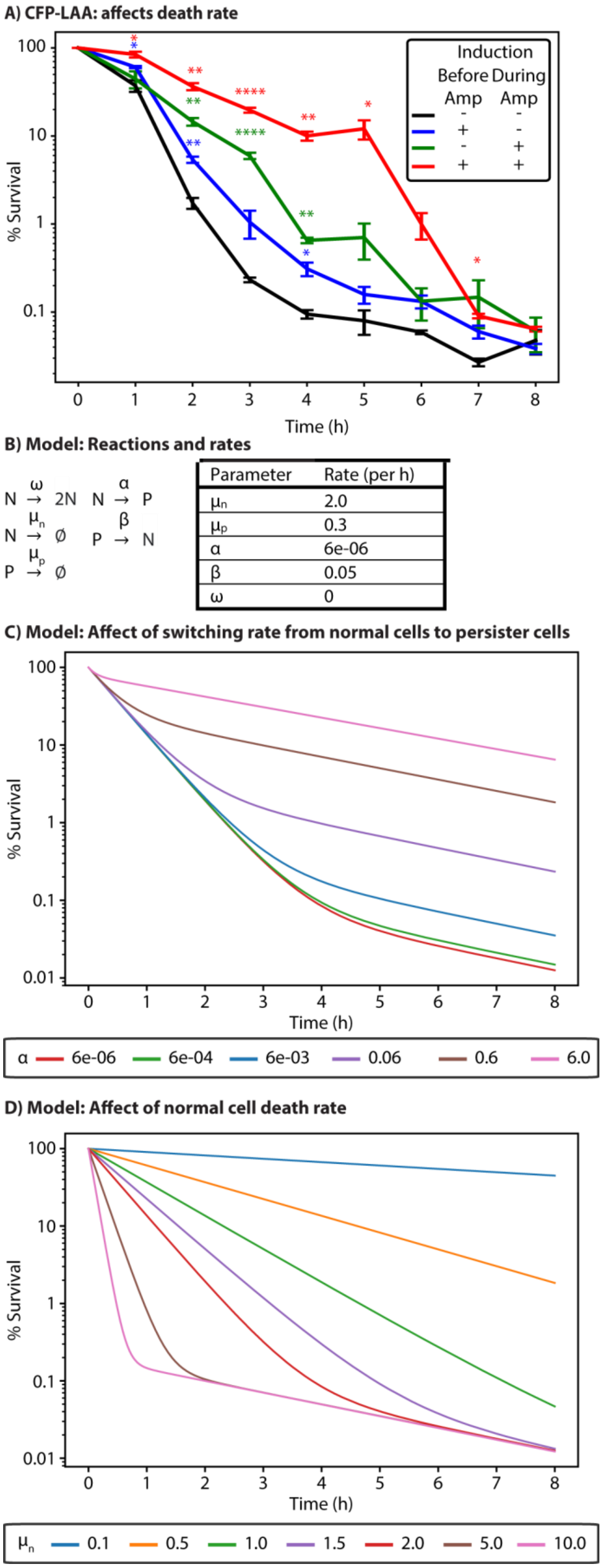
Figure 11. Proteolytic queueing effects in the presence of ciprofloxacin and chloramphenicol. a. Induction of untagged CFP during ciprofloxacin treatment increases survival less than 4-fold. **b.** Induction of CFP-LAA during ciprofloxacin treatment increases survival ~60-fold. **c.** Induction of CFP-LAA during ampicillin and chloramphenicol treatment has no apparent effect on survival ($p>0.7$). X-axis labels correspond to Fig. 2. Cultures were treated with ciprofloxacin (1 μg/ml) or chloramphenicol (5 μg/ml) respectively. Error bars represent SEM. $n \geq 3$. *$p<0.05$. **$p<0.01$.

5.3.3. Proteolytic queueing affects population decay

To gain further insight into the relationship between proteolytic queueing, tolerance and persistence, we measured how a proteolytic queue affects population decay by measuring survival for up to 8 hours of ampicillin treatment. Our results show a typical biphasic curve indicative of persister cells in the uninduced population. When the population is induced 24 hours prior to and during antibiotic treatment this curve shifts as the rate of population decay slows compared to uninduced cultures. The addition of the inducer exclusively during antibiotic treatment takes a similar effect between two and three hours into treatment. If the queue is induced 24 hours prior to antibiotic treatment, but the queue is not maintained (i.e. the inducer is removed during antibiotic treatment) the effect of the queue dissipates between one to two hours. There is no apparent difference between induced and uninduced cultures after 8 hours, which suggests there is little to no effect on persistence (Figure 12A).

5.3.4. Computational modeling supports queueing-tolerance

Based on the *in vivo* results, we considered a simple computational model of population decay during antibiotic treatment modified from Kussel *et al.*[162]. In our model, the persister population (P) has a lower death rate than the susceptible population (N), where the death rates are represented by μ_p and μ_n respectively. We estimated μ_p and μ_n based on the experimentally determined decay rate of the uninduced population before and after two hours, and set the initial persister population to 0.2% of the total population (Figure 12B). Normal (susceptible) cells enter persistence at rate α, and persister cells return to the normal state at rate β. The rates α and β were set relative to μ_n based on the relationship between these values in Kussel *et al*[162]. Our base model resembles population decay as measured in experimental tests. We use the model to determine whether the increase in overall population survival due to queue formation can be attributed to an increased rate of entering persistence (α) or increased tolerance (i.e.



$$N \xrightarrow{\omega} 2N \qquad N \xrightarrow{\alpha} P$$
$$N \xrightarrow{\mu_n} \emptyset \qquad P \xrightarrow{\beta} N$$
$$P \xrightarrow{\mu_p} \emptyset$$

Parameter	Rate (per h)
μ_n	2.0
μ_p	0.3
α	6e-06
β	0.05
ω	0

Figure 12. Time of queue formation influences survival. a. Stationary phase cells were diluted 1/100 into fresh media containing ampicillin (100 μg/ml) and sampled every hour for 8 h (n ≥ 3). Symbols (-/+) correspond to Fig. 2c. Error bars represent SEM. Asterisks indicate p-value (compared to no induction (black)) *p<0.05, **p<0.01, ***p<0.001, ****p<0.0001. There is 100% survival at time zero, because percent survival is determined based on the surviving CFU/ml compared to the CFU/ml at time zero. **b-d.** Stochastic model of population decay with antibiotic treatment. **b.** Reactions for the model (left) and baseline rates used for the simulations (right) unless stated otherwise (red lines below). Normal cell division (ω) was set to zero as dividing cells die during ampicillin treatment. **c.** Increasing the rate of entering persistence (α) increases cell number during the second phase of population decay. **d.** Decreasing the rate of normal cell death (μ_n) causes the first phase of population decay to lengthen. Y-axes are in logarithmic scale for **a.**, **c.**, and **d**.

decreased μ_n). Exploration of these parameters using stochastic simulations shows that increasing the rate at which normal cells become persisters (α) shortens the first phase of population decay and increases the number of persisters (Figure 12C). Decreasing the rate of normal cell death (μ_n) lengthens the first phase of population decay but has little to no effect on the number of persisters (Figure 12D).

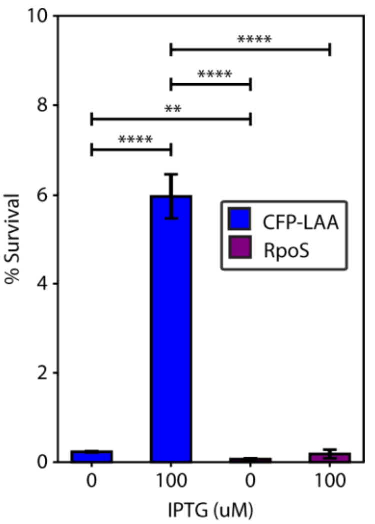


Figure 13. Induction of RpoS during antibiotic treatment has no significant effect on survival after 3 hours of ampicillin treatment (p>0.4). The same CFP-LAA data was used in Figure 10F.

5.3.5. Overexpression of RpoS does not reproduce queueing-tolerance

An increase in tolerance in response to proteolytic queueing at ClpXP is likely due to an increase in the number of one or many proteins. A good candidate is the transcription factor RpoS, a persister related gene[48] that is responsible for regulation of stationary phase, affected by the level of *ssrA*, and regulated by proteolytic degradation[52]. As such, we tested the effects of increasing RpoS levels by gene overexpression using the same vector, promoter and ribosome binding sites as used to overexpress CFP and CFP-LAA. We found that overexpression of RpoS does not cause a significant increase in tolerance, especially when compared to proteolytic queueing (Figure 13).

6. DISCUSSION

Proteolytic queueing is an integral component of native systems, and synthetic queues have great potential for studying systems at the proteolytic level. Here we show that queueing provides a tunable method to interfere with protease degradation and affect antibiotic tolerance. Increased antibiotic tolerance in response to queueing was independent of the carbon source (glycerol or glucose) and antibiotic class (β-lactam or fluoroquinolone). When we prevented queue formation using chloramphenicol, adding the inducer did not affect cell survival under the treatment of ampicillin. While CFP production alone slightly increased survival for ciprofloxacin, we suspect that high production of CFP with no apparent method of removal (besides cell division; minimal degradation) causes cell stress and affects survival, especially since high levels of fluorescent proteins can cause oxidative stress[163,164], which is known to increase persistence[165-167]. However, because CFP-LAA is removed via degradation (indicated by lower fluorescence than CFP-untagged), the effects seen via overexpression of CFP should be less prominent during CFP-LAA overexpression. The results we describe here would not have been identified in a *clpP* knockout, because *clpP* knockouts break cellular systems and detrimentally affect cellular processes[168], as evidence by growth defects[149]. Similarly, studies of TA systems and their role in antibiotic survival are confounded by the fact that TA systems make up highly interconnected networks with built in redundancy so that removal of several TA systems does not fully disclude activities by the others[25,28,169] and can even affect growth[170]. As such, changes in tolerance are difficult to differentiate from affects caused by permanent alterations in system dynamics resulting from genetic mutations. Here we demonstrate the utility of proteolytic queueing to study antibiotic survival, while minimizing negative effects of protease knockouts that could obfuscate the phenomenon of interest.

In some cases, the change in survival at three hours might be interpreted as a change in persistence; however, the shift in decay rates (as described in Figure 12A) clearly demonstrates that queueing increases antibiotic tolerance rather than persistence. Furthermore, the effects caused by adding or removing the inducer during antibiotic treatment suggest that the change in antibiotic tolerance is due to an active response to the queue, which must be maintained to affect survival. Although persistence does not appear to be affected by the proteolytic queue at ClpXP, further overloading ClpXP is possible and we simply may not be able to measure an affect at this level. However, persisters are considered metabolically dormant and an active response to the queue could explain why tolerance is affected but not persistence. Alternatively, the synthetic queue may not actually form in persister cells due to slowed translation and transcription. Our model supports that antibiotic tolerance is being affected by queueing rather than persistence, as altering survival of the 'normal' population (i.e. tolerance) more closely resembles the effects of proteolytic queueing than altering the rate of switching into persistence. While these results are specific to queueing at ClpXP, tags are available to test the effects of queueing at other proteases (e.g. Lon and ClpAP)[145].

Queueing at ClpXP is likely affecting the proteome of the cell, either directly or indirectly, and pleiotropic effects on protein content and gene regulation could be limiting antibiotic efficacy. We suspect that queue formation increases the intracellular concentration of one or multiple protein species causing a regulatory cascade. When considering proteins both degraded by ClpXP and related to persistence, TA systems are unlikely to be the causative factor, because decreasing degradation should increase antitoxin levels and decrease survival rather than increase survival as we observe. Instead, we consider regulatory proteins as candidates for the causative factor in queueing effects on tolerance. Several regulatory proteins are degraded by ClpXP[154] including

RpoS and DksA, proteins that have been implicated in persistence[139,167,171] and may be involved in tolerance.

We have tested the effect of overexpressing RpoS under the same conditions as proteolytic queueing to see if we could replicate the queueing-tolerance phenotype. Our results show that overexpression of RpoS does not significantly affect antibiotic tolerance. These results do not confirm that RpoS alone is responsible for changes in tolerance, especially considering that RpoS levels range broadly under different stress conditions and that over 23% of the *E. coli* genome is regulated by RpoS[156]. However, several other regulatory and stress response proteins are degraded by ClpXP[154], and increased concentrations of one or several of these proteins due to slowed degradation could be causing the downstream effects that lead to increased tolerance. In a similar vein, computational modeling has shown that altering degradation of MarA (a regulatory protein degraded by Lon that is related to antibiotic tolerance) leads to increased coordination of downstream genes[172].

The increase in antibiotic tolerance due to queue formation at ClpXP may be specific to overexpression of the LAA-tag, especially when considering that the number of LAA tagged proteins naturally increases during stress. The number of proteins with LAA tags increase during heat shock[173], and queue formation at the proteases is likely a consequence of the increasing cellular traffic. If the native LAA tag is removed from SsrA while maintaining the ribosome rescue function, the survival of ampicillin treatment decreases in *E. coli*[139]. As the LAA tag could be a measurement of environmental stress, cells may have evolved to increase tolerance in response to increased queueing via LAA. Since ribosome rescue and proteolytic queueing are common across species, stress signaling via proteolytic queueing could be a general mechanism to regulate survival related genes. Considering that proteolytic queueing is a natural phenomenon and synthetic queues have fewer negative effects compared to protease knockouts, our work demonstrates that proteolytic queueing is a viable alternative method to study proteolytic degradation by specific proteases. In the case of antibiotic tolerance, identifying the key proteins affected by the queue during bacterial tolerance and then understanding how these proteins interact has the potential to determine new drug targets for killing bacterial pathogens.

6.1. Materials and Methods

6.1.1. Strains and Plasmids

All strains are derived from *E. coli* DH5αZ1, and contain plasmids with the synthetic circuits, p24KmNB82 (CFP-LAA) and p24KmNB83 (untagged CFP) as described in REF[145] and p24KmAA01, which contains RpoS cloned downstream of $P_{lac/ara}$ promoter of p24Km (kanamycin 25 µg/mL) as in REF[145]. As such, CFP, CFP-LAA and RpoS are all expressed under identical promoters and ribosome binding sites. DH5αZ1 was derived from *E. coli* K12 (arguably the most studied bacteria strain[174]), it is used by many in synthetic biology and outside the field[175-178], this strain has previously been used to study persistence/tolerance or mechanisms related to them (e.g. toxin-antitoxin systems)[179-181], and our previous queueing experiments used these derivitives[145].

The cultures were grown in modified MMA media[182], which we will refer to as MMB. MMB media consists of the following: K_2HPO_4 (10.5 mg/ml), KH_2PO_4 (4.5 mg/ml), $(NH_4)_2SO_4$ (2.0 mg/ml), $C_6H_5Na_3O_7$ (0.5 mg/ml) and NaCl (1.0 mg/ml). Additionally, MMB+ consists of MMB and the following: 2 mM $MgSO_4$ x $7H_2O$, 100 µM $CaCl_2$, thiamine (10 µg/ml), 0.5% glycerol and amino acids (40 µg/ml). Cultures grown on glucose as the carbon source included 0.5% glucose instead of glycerol. Strains containing the plasmid p24Km and derivatives were grown in MMB+ kanamycin (Km, 25 µg/ml) or on Miller’s Lysogeny broth (LB) agar plates + Km (25 µg/ml). All cultures were incubated at 37° C and broth cultures were shaken at 250 rpm.

6.1.2. Quantification of persistence

Persisters were quantified by comparing colony-forming units per milliliter (CFU/ml) before antibiotic treatment to CFU/ml after antibiotic treatment. The procedure for quantifying persister levels is based on previous research[39,179,183] (Figure A6). Briefly, overnight cultures were diluted 1/100 into fresh media and grown until they reach between OD_{600} 0.2-0.3. A reduced volume of culture (20 ml) was aliquoted into a 125 ml flask, and grown for 16 hours to enter stationary phase. Once in stationary phase, cultures were divided into two flasks with 0.2% arabinose, one flask of each replicate was also treated with 100 µM IPTG to induce expression under $P_{lac/ara-1}$.

Arabinose was added to both induced and uninduced cultures to maintain consistency (Figure A7). All flasks were incubated for 24 hours before taking samples for plating and antibiotic treatment; cells were diluted 1/100[179,183] into glass tubes, treated with 10X the MIC of ampicillin (100 µg/ml; Figure A8) or 100X MIC of ciprofloxacin (1 µg/ml) at 37° C and shaken at 250 rpm for select time periods, 3 hours unless otherwise stated. Ampicillin solutions were stored at -80°C and only thawed once to reduce variability[137,184]. When indicated, samples were treated with chloramphenicol (5 µg/ml); cultures treated with chloramphenicol alone were diluted 1/10. Samples for quantification of CFU/ml were kept on ice and diluted using cold MMB before plating on LB/Km (25 µg/ml) agar plates. Cultures treated with ciprofloxacin were centrifuged at 16,000 *x g* for 3 minutes then washed with cold MMB to dilute ciprofloxacin before taking samples for quantification. LB agar plates were incubated at 37ºC for 40-48 hours, then scanned using a flatbed scanner[40,185]. Custom scripts were used to identify and count bacterial colonies[103] then used to calculate CFU/ml and persister frequency. Colonies were tested periodically for resistance, and we found no resistance in >350 colonies tested.

6.1.3. Quantification of CFP

Cells were grown and treated with ampicillin as described in quantification of persistence above. After antibiotic treatment, 300 µl of cell culture was added to individual wells in a 96-Well Optical-Bottom Plate with Polymer Base (ThermoFisher) for fluorescence measurement using FLUOstar Omega microplate reader. The excitation and emission (Ex/Em) used for CFP measurement was 440/480. Readings were measured after four minutes of shaking to decrease variability between wells. Background fluorescence (mean fluorescence of MMB media) was subtracted from the raw reads. Fluorescence values were normalized by CFUs as determined by quantification of persistence, which was carried out simultaneously. Mean and SEM for fluorescence was determined across four biological replicates and three technical replicates.

Computational modeling

Our model is modified from Kussel *et al.*[162] where P is the persister population and N is the susceptible population (Figure 12B). Initial species counts P and N were set to 99800 and 200 respectively for all simulations, which we based on the percent survival of uninduced cultures. The death rate of N (μ_n) and P (μ_p) and the rate of entering (α) and exiting (β) persistence were set as shown in Fig. 4b unless otherwise stated. The rate of susceptible cell division (ω) was set to zero, as normal cells cannot divide without lysis during ampicillin treatment[46]. All simulations were performed using a custom implementation of the Gillespie algorithm[186] in Python leveraging optimizations made possible by the Cython library[187]. Libraries from the SciPy stack[188] were used for analysis.

6.1.4. Statistics

All data is presented as mean ± SD or SEM of at least 3 biological replicates as appropriate[189]. Statistical significance for populations with the same number of replicates (n) was determined using one-way f-test to determine variance ($p<0.001$ was considered to have significant variance) followed by a Student's t-test (no variance) or a Welch's t-test (significant variance). Populations with different n values were compared using a Welch's t-test. All statistical tests were run in Python using libraries from SciPy on groups with at least three biological replicates.

6.1.5. Calculation of doubling times

Optical density (OD) was measured at 600 nm in a microplate reader (see Quantification of CFP). Doubling time (t_d) was determined as described in REF[190]. Briefly, we calculated the linear regression of the natural logarithm (ln) of the OD over time (t). The equation of the line can thus be derived from a logarithmic growth curve and solved for t_d (Eq. 1-2).

$$OD = OD_0 e^{\frac{\ln(2)}{t_d}t} \qquad \text{Eq. 1}$$

$$\ln(OD) = \frac{\ln(2)}{t_d}t + \ln(OD_0) \qquad \text{Eq. 2}$$

6.2. Author Contributions

H.S.D wrote the manuscript, developed custom code for colony counting, measured bacterial growth rates and ran statistical analyses. A.A. performed ampicillin and ciprofloxacin persister assays. P.J. performed plate reader fluorescence assays. E.S. performed chloramphenicol persister assay. C.T.O. and H.S.D. adapted the persister model and ran stochastic simulations. N.C.B. initiated and directed the project. All authors contributed to discussing and editing the manuscript.

7. CONCLUSIONS

One of the reasons synthetic biology has developed rapidly over the past two decades is because the field is, by definition, interdisciplinary. In this work, I show how bioinformatics, molecular biology, computer science and traditional microbiology can work in combination to inform our understanding of how biological cells function, with a particular focus on how microbes respond to stress. Research to understand how cells respond to stress and survive is critical to improving antimicrobial treatments, but also for maximizing bioproduction and optimizing other processes dependent on living microbes. Herein we show that toxin-antitoxin systems, which are associated with stress responses, have diverse mechanisms of regulation (Chapter 2). I believe that further understanding the role TA systems play in cellular stress responses will require a network-based approach.

Another area that will benefit from a network-based analysis is quantifying the molecular mechanisms responsible for the Queueing-Tolerance phenomenon (Chapter 5). While we have clearly demonstrated using queueing theory that proteases are a key aspect of tolerance, we do not yet know whether multiple genes/proteins are responsible for the effects. Holistic approaches to study this phenomenon could include RNA-sequencing or mass spectroscopy. These types of holistic approaches will also benefit from complementary targeted approaches. In the case of antibiotic tolerance, the challenges of studying the dynamics of individual systems are compounded by the fact that the tolerant population is rather small compared to the general population. Single-cell tracking provides the means to study small populations, and thus cell tracking is a necessary step in understanding the dynamics of tolerant cells. The single-cell tracking software we have developed (Chapter 4) is designed to be implemented in future studies towards this end.

8. APPENDIX

8.1. Figures

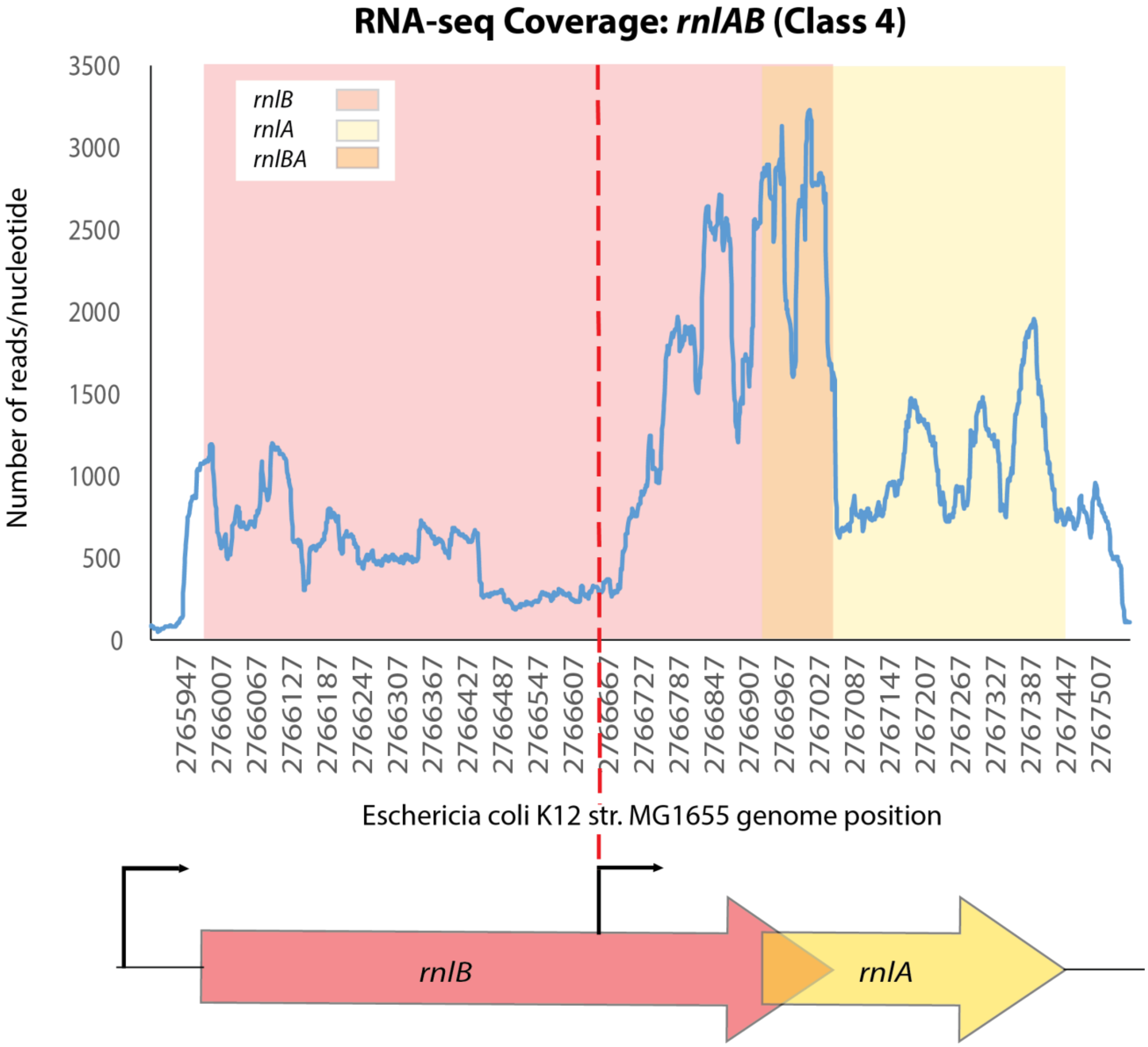


Figure A1. RNA-seq reads from experiment SRX1424838 (GSE74809, [100]) mapped to *rnl*AB (Class 4). An increase of transcription occurs at the transcriptional start site for the internal promoter [75] (indicated as a dashed line) approximately 280 nt upstream from the antitoxin start codon. Comparison of the coverage between toxin (coding region red and orange) and antitoxin (coding region yellow and orange) using number of reads mapped to each gene would misrepresent the ratio of mRNA due to the transcriptional start site location within the rnl*B* gene. The coverage between the non-overlapping transcribed regions of *rnl*B to *rnl*A to show that antitoxin coverage is over two-fold greater than toxin.

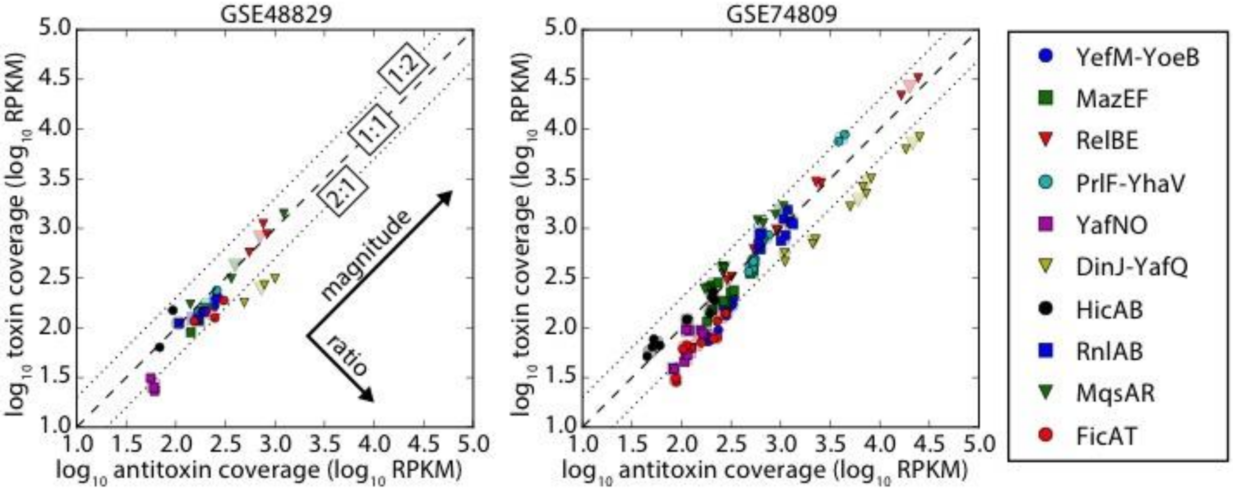


Figure A2. Biological replicate values for antitoxin and toxin coverage across RNA-seq datasets. Shown is a comparison of coverage for biological replicates from the datasets GSE48829 [99] (left) and GSE74809 [100] (right), the former of which contains triplicate data in one growth condition, and the latter of which contains duplicate data across five growth conditions. Replicates are shown as small symbols, while the mean of their $\log_{10}$ coverage is shown as a corresponding larger transparent symbol. The dashed line represents antitoxin to toxin coverage ratio 1:1 (equal coverage), while the dotted lines represent antitoxin to toxin coverage ratios equal to 1:2 and 2:1. Units of coverage are RPKM, and the major directions of ratio and magnitude are also included (see Methods).

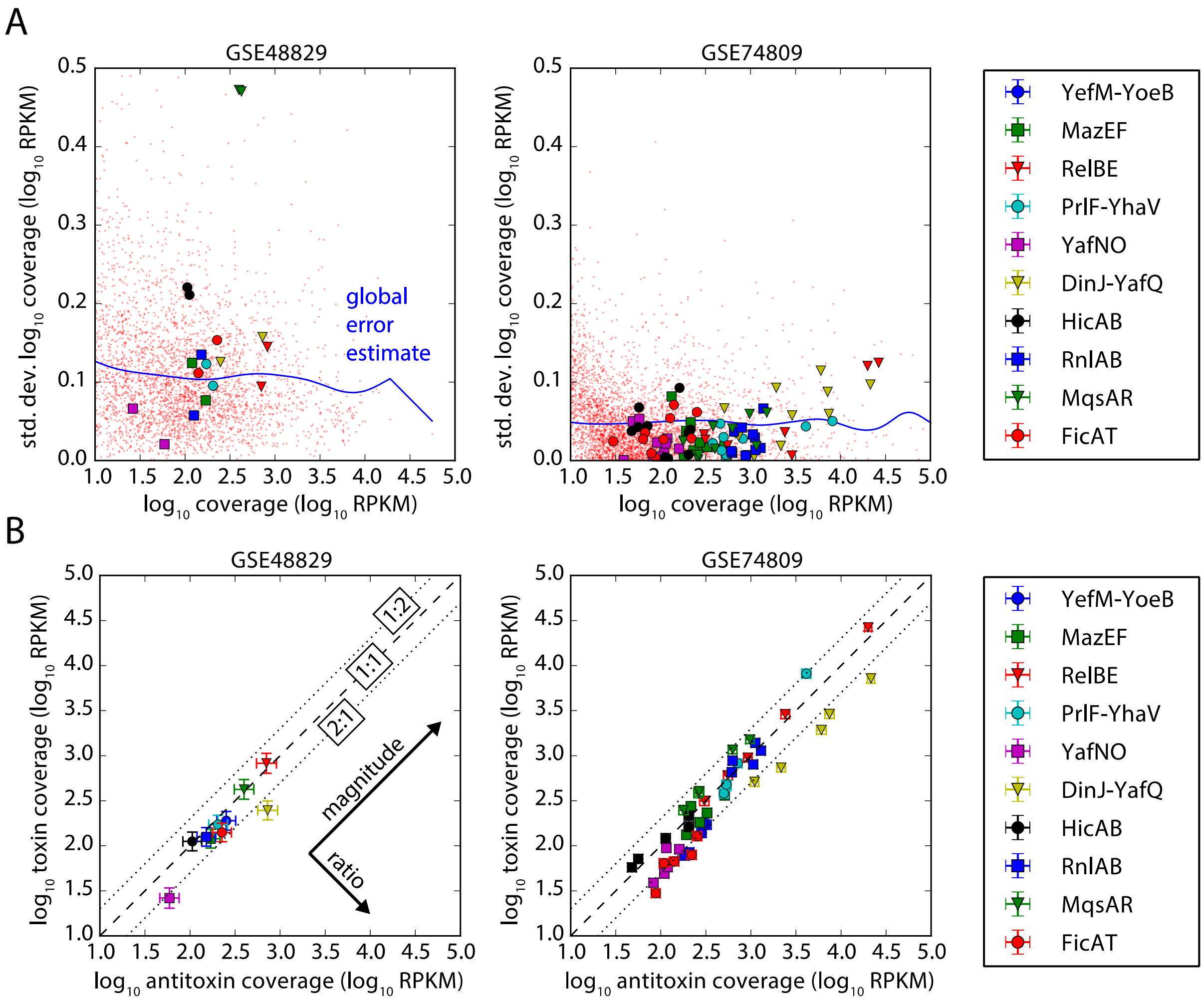
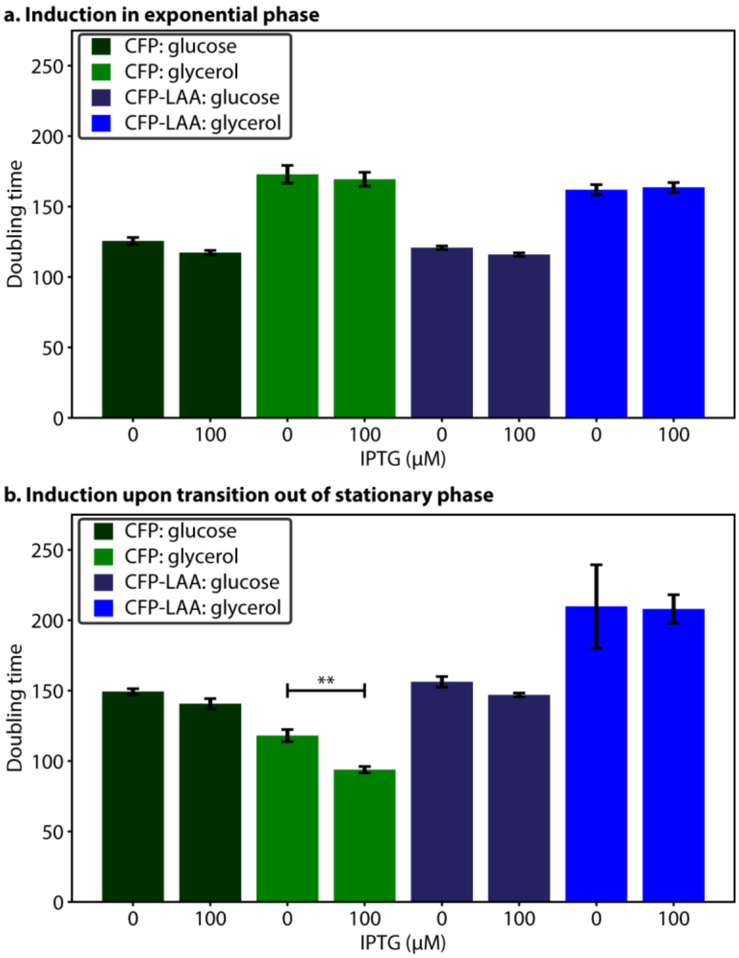


Figure A3. Representative error estimates for RNA-seq datasets. (A) Biological replicates in the datasets GSE48829 [99] (left) and GSE74809 [100] (right) were used to estimate the standard error of the log-coverage (natural logarithm of the coverage) for each gene in the dataset (red dots). This logarithmic error measurement is natural for data represented in log-log coordinates. Antitoxin and toxin genes belonging to the TA systems listed in the legend are represented using their own symbols. A smooth global error estimate is plotted as a blue line. This global error estimate $\langle\sigma\rangle_i$ for a gene with index i is derived from the formula $\langle\sigma\rangle_i = \sum_j \sigma_j \rho_{ji}$ (summation over all indices j), where ρ_{ji} is a normalized weighting factor proportional to $\exp\left(-2\left(x_i - x_j\right)^2\right)$, with x_i the log-coverage for a gene with index i. (B) The mean log-coverage for each condition from panel A are plotted, with error bars corresponding to their global error estimates.



a. Induction in exponential phase

b. Induction upon transition out of stationary phase

Figure A4. Induction of untagged CFP and CFP-LAA tag has no apparent effect on growth in MMB+ media. a. Cultures were induced during exponential phase. Induction of CFP-untagged or CFP-LAA had no apparent effect on growth when glucose or glycerol were the sole carbon source. b. Cultures were induced after 1/100 dilution into fresh media (the same dilution used during persister quantification, see Methods). Induction of CFP-untagged or CFP-LAA had no apparent effect on growth when glucose was the sole carbon source. Induction of CFP-untagged or CFP-LAA had no apparent effect on growth when glycerol was the sole carbon source. CFP-untagged grew slightly faster when induced by IPTG in glycerol media. However, if this difference in growth were to affect antibiotic survival, we would expect a decrease rather than no significant change in survival (see Results). Doubling time was calculated based on OD_{600} readings over time in a microplate reader (see Methods). n = 4. Error bars represent SEM.

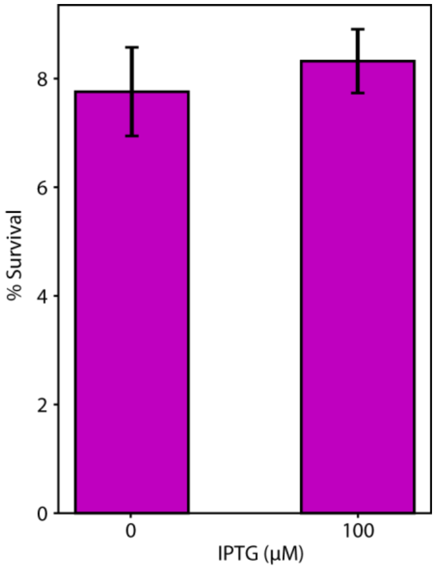
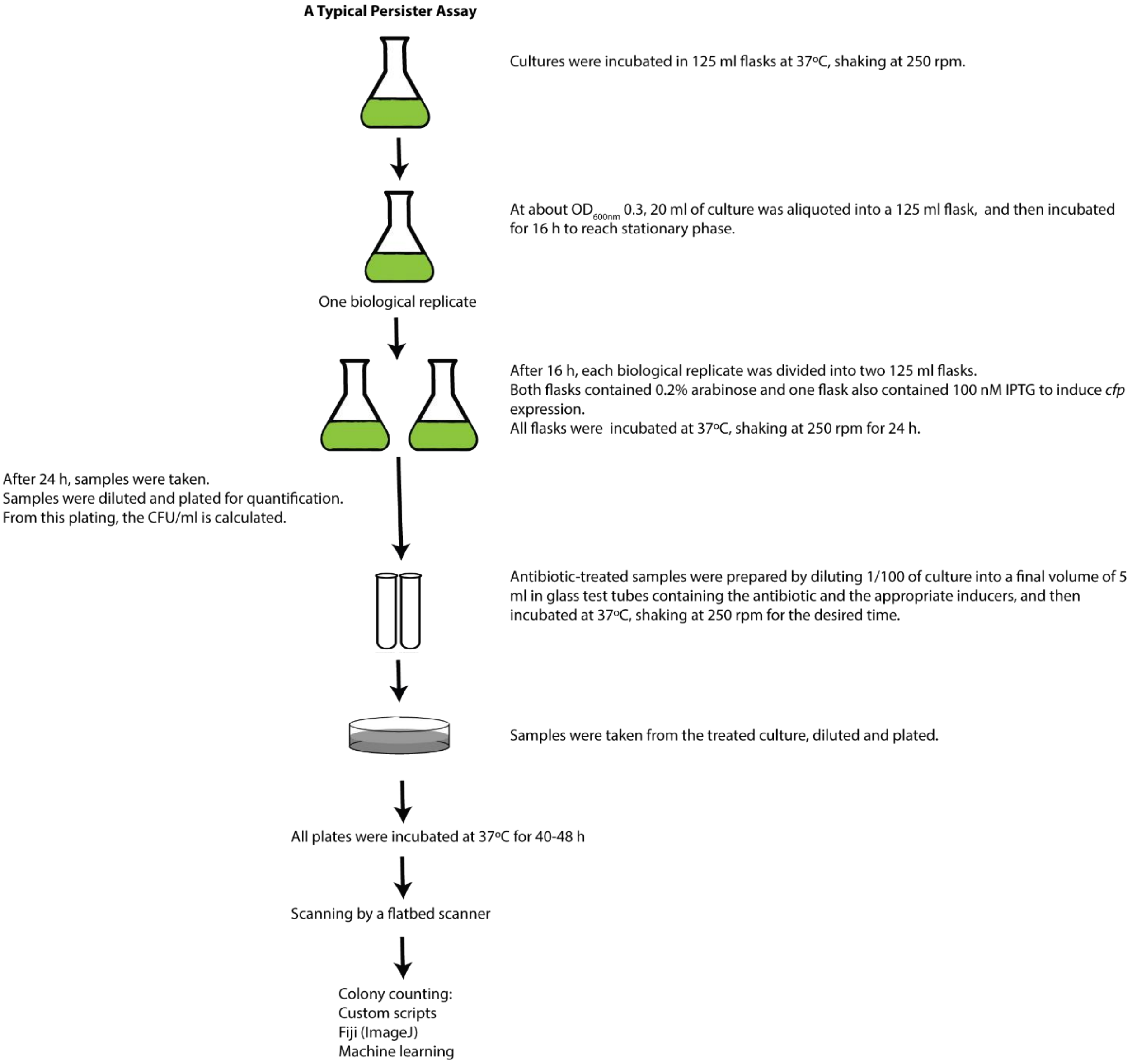


Figure A5. Induction of CFP-LAA does not increase survival of cells treated with chloramphenicol. Cultures were treated with chloramphenicol, an antibiotic that inhibits translation, after a 1/10 dilution into fresh media from stationary phase. Induction of CFP-LAA via IPTG had no significant change in persistence compared to the uninduced cultures ($p > 0.7$; $n \geq 3$). Error bars represent SEM.



A Typical Persister Assay

Cultures were incubated in 125 ml flasks at 37°C, shaking at 250 rpm.

At about OD_{600nm} 0.3, 20 ml of culture was aliquoted into a 125 ml flask, and then incubated for 16 h to reach stationary phase.

After 16 h, each biological replicate was divided into two 125 ml flasks.
Both flasks contained 0.2% arabinose and one flask also contained 100 nM IPTG to induce *cfp* expression.
All flasks were incubated at 37°C, shaking at 250 rpm for 24 h.

After 24 h, samples were taken.
Samples were diluted and plated for quantification.
From this plating, the CFU/ml is calculated.

Antibiotic-treated samples were prepared by diluting 1/100 of culture into a final volume of 5 ml in glass test tubes containing the antibiotic and the appropriate inducers, and then incubated at 37°C, shaking at 250 rpm for the desired time.

Samples were taken from the treated culture, diluted and plated.

All plates were incubated at 37°C for 40-48 h

Scanning by a flatbed scanner

Colony counting:
Custom scripts
Fiji (ImageJ)
Machine learning

Figure A6. Persister assay flow chart of a typical assay. See Methods for details.

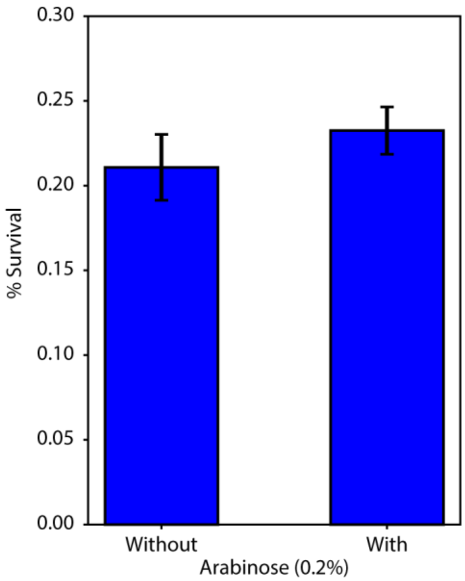


Figure A7. The addition of arabinose had no apparent effect on the tolerance/persister level during ampicillin treatment. Both IPTG and arabinose are inducers for CFP untagged and CFP-LAA tagged proteins. IPTG induces expression, arabinose alone does not induce expression, but arabinose can enhance expression when used in combination with IPTG. The effect of adding arabinose (0.2%) on tolerance/persistence to ampicillin was tested with CFP-LAA. Adding arabinose does not have a significant effect on survival of cells after 3 hours of ampicillin treatment ($p > 0.3$). Error bars represent SEM. $N \geq 3$.

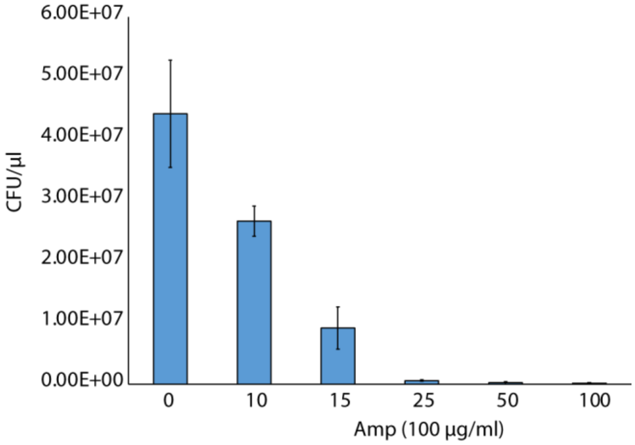


Figure A8. Determination of Minimal Inhibitory Concentration (MIC) for ampicillin. Exponential phase cultures were treated with different concentrations of ampicillin. The MIC was determined to be 10 µg/ml ($p < 0.03$ compared to zero). Error bars represent the standard deviation. The ciprofloxacin MIC was determined in a similar manner.

Tables

Table A1. Selected RNA-seq experiment numbers and conditions for GSE48829 (grey background) and GSE74809

Experiment	Media	Growth phase
SRX322083	Minimal media	Exponential
SRX322084	Minimal media	Exponential
SRX322085	Minimal media	Exponential
SRX1424798	M9 (glucose)	Early Exponential
SRX1424799	M9 (glucose)	Early Exponential
SRX1424808	M9 (glucose)	Mid-Exponential
SRX1424809	M9 (glucose)	Mid-Exponential
SRX1424818	M9 (glucose)	Transition to Stationary
SRX1424819	M9 (glucose)	Transition to Stationary
SRX1424828	M9 (glucose)	Stationary
SRX1424829	M9 (glucose)	Stationary
SRX1424838	M9 (glucose)	Late Stationary
SRX1424839	M9 (glucose)	Late Stationary

Table A2. Ratios of the calculated antitoxin to toxin translation initiation rates. All translation initiation rates (TIR) were calculated using translation rate calculators, as outlined in the main text (see Methods). TIR is in arbitrary units (AU).

TA System	RBS Calculator	UTR Designer	Barrick Calculator
MazEF	8.93	1.94	6.05
PrlF-YhaV	262	4.97	1.39
RelBE	17.8	2.84	1.45
MqsAR	0.01	1.34	15.1
YefM-YoeB	1.48	3.83	72.9
DinJ-YafQ	1.36	0.46	0.23
YafNO	10.4	0.97	0.32
RnlAB	119	NA	0.15
FicAT	31.4	18.9	1.20
HicAB P1	0.10	0.18	7.61
HicAB P2	8.93	1.94	6.05

N/A: Not available because the UTR Designer calculator cannot calculate with a TTG start codons.

8.2. Videos

All videos are available in the public repository https://osf.io/75avy (DOI: 10.17605/OSF.IO/75AVY).

Video A1. A tutorial video following sections 3.3 through 3.9 on a Linux computer. Filename: Tutorial_full.mp4 (https://osf.io/5d3sm/). Also available on Youtube: .

Video A2. *E. coli* cells (of the data repository) after being labeled according to cell lineage. Filename: Video1.mp4 (https://osf.io/hbe34/).

Video A3. Image masks with cells colors corresponding to cell lineage. Filename: Video2.mp4 (https://osf.io/pkxje/).

Video A4. *E. coli* cells (of the data repository) outlined in red based on segmentation. Filename Video3.mp4 (https://osf.io/ytbrx/).

LITERATURE CITED

1 Aizenman, E., Engelberg-Kulka, H. & Glaser, G. An Escherichia coli chromosomal "addiction module" regulated by guanosine [corrected] 3',5'-bispyrophosphate: a model for programmed bacterial cell death. *Proc Natl Acad Sci U S A* **93**, 6059-6063 (1996).

2 Engelberg-Kulka, H. & Glaser, G. Addiction modules and programmed cell death and antideath in bacterial cultures. *Annu. Rev. Microbiol.* **53**, 43-70, doi:10.1146/annurev.micro.53.1.43 (1999).

3 Helaine, S. & Kugelberg, E. Bacterial persisters: formation, eradication, and experimental systems. *Trends Microbiol.* **22**, 417-424, doi:10.1016/j.tim.2014.03.008 (2014).

4 Lobato-Marquez, D., Moreno-Cordoba, I., Figueroa, V., Diaz-Orejas, R. & Garcia-del Portillo, F. Distinct type I and type II toxin-antitoxin modules control Salmonella lifestyle inside eukaryotic cells. *Sci Rep* **5**, 9374, doi:10.1038/srep09374 (2015).

5 Georgiades, K. & Raoult, D. Genomes of the most dangerous epidemic bacteria have a virulence repertoire characterized by fewer genes but more toxin-antitoxin modules. *PLoS One* **6**, e17962, doi:10.1371/journal.pone.0017962 (2011).

6 Leplae, R. *et al.* Diversity of bacterial type II toxin-antitoxin systems: a comprehensive search and functional analysis of novel families. *Nucleic Acids Res.* **39**, 5513-5525, doi:10.1093/nar/gkr131 (2011).

7 Pandey, D. P. & Gerdes, K. Toxin-antitoxin loci are highly abundant in free-living but lost from host-associated prokaryotes. *Nucleic Acids Res.* **33**, 966-976, doi:10.1093/nar/gki201 (2005).

8 Lobato-Marquez, D., Diaz-Orejas, R. & Garcia-Del Portillo, F. Toxin-antitoxins and bacterial virulence. *FEMS Microbiol. Rev.* **40**, 592-609, doi:10.1093/femsre/fuw022 (2016).

9 Christensen, S. K. *et al.* Overproduction of the Lon protease triggers inhibition of translation in Escherichia coli: involvement of the yefM-yoeB toxin-antitoxin system. *Mol. Microbiol.* **51**, 1705-1717, doi:10.1046/j.1365-2958.2003.03941.x (2004).

10 Muthuramalingam, M., White, J. C. & Bourne, C. R. Toxin-Antitoxin Modules Are Pliable Switches Activated by Multiple Protease Pathways. *Toxins (Basel)* **8**, doi:10.3390/toxins8070214 (2016).

11 Page, R. & Peti, W. Toxin-antitoxin systems in bacterial growth arrest and persistence. *Nat. Chem. Biol.* **12**, 208-214, doi:10.1038/nchembio.2044 (2016).

12 Donegan, N. P., Thompson, E. T., Fu, Z. & Cheung, A. L. Proteolytic regulation of toxin-antitoxin systems by ClpPC in *Staphylococcus aureus*. *J. Bacteriol.* **192**, 1416-1422, doi:10.1128/JB.00233-09 (2010).

13 Janssen, B. D., Garza-Sanchez, F. & Hayes, C. S. YoeB toxin is activated during thermal stress. *Microbiologyopen* **4**, 682-697, doi:10.1002/mbo3.272 (2015).

14 Prax, M. & Bertram, R. Metabolic aspects of bacterial persisters. *Front Cell Infect Microbiol* **4**, 148, doi:10.3389/fcimb.2014.00148 (2014).

15 Ogle, C. T. & Mather, W. H. Proteolytically coordinated activation of toxin-antitoxin modules. 1-24, doi:https://doi.org/10.1101/146027 (2017).

16 Fasani, R. A. & Savageau, M. A. Molecular mechanisms of multiple toxin-antitoxin systems are coordinated to govern the persister phenotype. *Proc Natl Acad Sci U S A* **110**, E2528-2537, doi:10.1073/pnas.1301023110 (2013).

17 Fasani, R. A. & Savageau, M. A. Unrelated toxin-antitoxin systems cooperate to induce persistence. *J R Soc Interface* **12**, 20150130: 20150131-20150113, doi:10.1098/rsif.2015.0130 (2015).

18 Kasari, V., Mets, T., Tenson, T. & Kaldalu, N. Transcriptional cross-activation between toxin-antitoxin systems of Escherichia coli. *BMC Microbiol.* **13**, 45, doi:10.1186/1471-2180-13-45 (2013).

19 Rocker, A. & Meinhart, A. Type II toxin: antitoxin systems. More than small selfish entities? *Curr. Genet.* **62**, 287-290, doi:10.1007/s00294-015-0541-7 (2016).

20 Overgaard, M., Borch, J., Jorgensen, M. G. & Gerdes, K. Messenger RNA interferase RelE controls *relBE* transcription by conditional cooperativity. *Mol. Microbiol.* **69**, 841-857, doi:10.1111/j.1365-2958.2008.06313.x (2008).

21 Gelens, L., Hill, L., Vandervelde, A., Danckaert, J. & Loris, R. A general model for toxin-antitoxin module dynamics can explain persister cell formation in *E. coli*. *PLoS Comput Biol* **9**, e1003190: 1003191-1003117, doi:10.1371/journal.pcbi.1003190 (2013).

22 Ruangprasert, A. *et al.* Mechanisms of toxin inhibition and transcriptional repression by *Escherichia coli* DinJ-YafQ. *J. Biol. Chem.* **289**, 20559-20569, doi:10.1074/jbc.M114.573006 (2014).

23 Turnbull, K. J. & Gerdes, K. HicA toxin of *Escherichia coli* derepresses *hicAB* transcription to selectively produce HicB antitoxin. *Mol. Microbiol.* **104**, 781-792, doi:10.1111/mmi.13662 (2017).

24 Kim, Y. *et al.* Escherichia coli toxin/antitoxin pair MqsR/MqsA regulate toxin CspD. *Environ. Microbiol.* **12**, 1105-1121, doi:10.1111/j.1462-2920.2009.02147.x (2010).

25 Gupta, A., Venkataraman, B., Vasudevan, M. & Gopinath Bankar, K. Co-expression network analysis of toxin-antitoxin loci in *Mycobacterium tuberculosis* reveals key modulators of cellular stress. *Sci Rep* **7**, 5868, doi:10.1038/s41598-017-06003-7 (2017).

26 Wood, T. K. & Song, S. Forming and Waking Dormant Cells: The ppGpp Ribosome Dimerization Persister Model. *Biofilm*, doi:10.1016/j.bioflm.2019.100018 (2020).

27 Fraikin, N., Goormaghtigh, F. & Van Melderen, L. in *Persister Cells and Infectious Disease* Ch. Chapter 8, 181-202 (2019).

28 Ronneau, S. & Helaine, S. Clarifying the Link between Toxin-Antitoxin Modules and Bacterial Persistence. *J. Mol. Biol.*, doi:10.1016/j.jmb.2019.03.019 (2019).

29 Durand, G. A., Raoult, D. & Dubourg, G. Antibiotic discovery: History, methods and perspectives. *Int. J. Antimicrob. Agents*, doi:10.1016/j.ijantimicag.2018.11.010 (2018).

30 Pires, D., de Kraker, M. E. A., Tartari, E., Abbas, M. & Pittet, D. ‘Fight Antibiotic Resistance—It’s in Your Hands’: Call From the World Health Organization for 5th May 2017. *Clin. Infect. Dis.* **64**, 1780-1783, doi:10.1093/cid/cix226 (2017).

31 Martinez, J. L. & Baquero, F. Emergence and spread of antibiotic resistance: setting a parameter space. *Ups J Med Sci* **119**, 68-77, doi:10.3109/03009734.2014.901444 (2014).

32 Wernicki, A., Nowaczek, A. & Urban-Chmiel, R. Bacteriophage therapy to combat bacterial infections in poultry. *Virology journal* **14**, 179, doi:10.1186/s12985-017-0849-7 (2017).

33 Economic Research Service (ERS), U. S. D. o. A. U. *Cost Estimates of Foodborne Illnesses*, <https://www.ers.usda.gov/data-products/cost-estimates-of-foodborne-illnesses> (2018).

34 CDC. (ed CDC U.S. Department of Health and Human Services) (Atlanta, GA, 2019).

35 Windels, E. M., Michiels, J. E., Van den Bergh, B., Fauvart, M. & Michiels, J. Antibiotics: Combatting Tolerance To Stop Resistance. *MBio* **10**, doi:10.1128/mBio.02095-19 (2019).

36 Fisher, R. A., Gollan, B. & Helaine, S. Persistent bacterial infections and persister cells. *Nat. Rev. Microbiol.* **15**, 453-464, doi:10.1038/nrmicro.2017.42 (2017).

37 Fauvart, M., De Groote, V. N. & Michiels, J. Role of persister cells in chronic infections: clinical relevance and perspectives on anti-persister therapies. *J. Med. Microbiol.* **60**, 699-709, doi:10.1099/jmm.0.030932-0 (2011).

38 Michiels, J. E., Van den Bergh, B., Verstraeten, N. & Michiels, J. Molecular mechanisms and clinical implications of bacterial persistence. *Drug Resist Updat* **29**, 76-89, doi:10.1016/j.drup.2016.10.002 (2016).

39 Rowe, S. E., Conlon, B. P., Keren, I. & Lewis, K. Persisters: Methods for Isolation and Identifying Contributing Factors--A Review. *Methods Mol Biol* **1333**, 17-28, doi:10.1007/978-1-4939-2854-5_2 (2016).

40 Levin-Reisman, I. *et al.* Antibiotic tolerance facilitates the evolution of resistance. *Science* **355**, 826-830, doi:10.1126/science.aaj2191 (2017).

41 Van den Bergh, B. *et al.* Frequency of antibiotic application drives rapid evolutionary adaptation of *Escherichia coli* persistence. *Nat Microbiol* **1**, 16020, doi:10.1038/nmicrobiol.2016.20 (2016).

42 Windels, E. M. *et al.* Bacterial persistence promotes the evolution of antibiotic resistance by increasing survival and mutation rates. *ISME J*, doi:10.1038/s41396-019-0344-9 (2019).

43 Kim, J. S., Yamasaki, R., Song, S., Zhang, W. & Wood, T. K. Single cell observations show persister cells wake based on ribosome content. *Environ. Microbiol.*, doi:10.1111/1462-2920.14093 (2018).

44 Goormaghtigh, F. *et al.* Reassessing the Role of Type II Toxin-Antitoxin Systems in Formation of Escherichia coli Type II Persister Cells. *MBio* **9**, doi:10.1128/mBio.00640-18 (2018).

45 El Meouche, I. & Dunlop, M. J. Heterogeneity in efflux pump expression predisposes antibiotic-resistant cells to mutation. *Science* **362**, 686-690, doi:10.1126/science.aar7981 (2018).

46 Balaban, N. Q., Merrin, J., Chait, R., Kowalik, L. & Leibler, S. Bacterial persistence as a phenotypic switch. *Science* **305**, 1622-1625, doi:10.1126/science.1099390 (2004).

47 Wang, W. *et al.* Transposon Mutagenesis Identifies Novel Genes Associated with *Staphylococcus aureus* Persister Formation. *Front Microbiol* **6**, 1437, doi:10.3389/fmicb.2015.01437 (2015).

48 Wu, N. *et al.* Ranking of persister genes in the same *Escherichia coli* genetic background demonstrates varying importance of individual persister genes in tolerance to different antibiotics. *Front Microbiol* **6**, 1003, doi:10.3389/fmicb.2015.01003 (2015).

49 Wu, S., Yu, P. L., Wheeler, D. & Flint, S. Transcriptomic study on persistence and survival of Listeria monocytogenes following lethal treatment with nisin. *J Glob Antimicrob Resist* **15**, 25-31, doi:10.1016/j.jgar.2018.06.003 (2018).

50 Gottesman, S. Proteases and their targets in Escherichia coli. *Annu. Rev. Genet.* **30**, 465-506, doi:10.1146/annurev.genet.30.1.465 (1996).

51 Baker, T. A. & Sauer, R. T. ClpXP, an ATP-powered unfolding and protein-degradation machine. *Biochim. Biophys. Acta* **1823**, 15-28, doi:10.1016/j.bbamcr.2011.06.007 (2012).

52 Ranquet, C. & Gottesman, S. Translational regulation of the *Escherichia coli* stress factor RpoS: a role for SsrA and Lon. *J. Bacteriol.* **189**, 4872-4879, doi:10.1128/JB.01838-06 (2007).

53 Janssen, B. D. & Hayes, C. S. The tmRNA ribosome-rescue system. *Adv Protein Chem Struct Biol* **86**, 151-191, doi:10.1016/B978-0-12-386497-0.00005-0 (2012).

54 Kim, J. S. & Wood, T. K. Persistent Persister Misperceptions. *Front Microbiol* **7**, 2134, doi:10.3389/fmicb.2016.02134 (2016).

55 Balaban, N. Q. *et al.* Definitions and guidelines for research on antibiotic persistence. *Nat. Rev. Microbiol.*, doi:10.1038/s41579-019-0196-3 (2019).

56 Conlon, B. P. *et al.* Activated ClpP kills persisters and eradicates a chronic biofilm infection. *Nature* **503**, 365-370, doi:10.1038/nature12790 (2013).

57 Mather, W. H., Cookson, N. A., Hasty, J., Tsimring, L. S. & Williams, R. J. Correlation resonance generated by coupled enzymatic processing. *Biophys. J.* **99**, 3172-3181, doi:10.1016/j.bpj.2010.09.057 (2010).

58 Butzin, N. C. & Mather, W. H. Synthetic genetic oscillators. *Reviews in Cell Biology and Molecular Medicine* (2015).

59 Stricker, J. *et al.* A fast, robust and tunable synthetic gene oscillator. *Nature* **456**, 516-519, doi:10.1038/nature07389 (2008).

60 Deter, H. S., Jensen, R. V., Mather, W. H. & Butzin, N. C. Mechanisms for Differential Protein Production in Toxin-Antitoxin Systems. *Toxins (Basel)* **9**, doi:10.3390/toxins9070211 (2017).

61 Wood, T. K., Knabel, S. J. & Kwan, B. W. Bacterial persister cell formation and dormancy. *Appl. Environ. Microbiol.* **79**, 7116-7121, doi:10.1128/AEM.02636-13 (2013).

62 Harms, A., Maisonneuve, E. & Gerdes, K. Mechanisms of bacterial persistence during stress and antibiotic exposure. *Science* **354**, aaf4268: 4261-4269, doi:10.1126/science.aaf4268 (2016).

63 Korch, S. B. & Hill, T. M. Ectopic overexpression of wild-type and mutant *hipA* genes in *Escherichia coli*: effects on macromolecular synthesis and persister formation. *J. Bacteriol.* **188**, 3826-3836, doi:10.1128/JB.01740-05 (2006).

64 Maisonneuve, E., Shakespeare, L. J., Jorgensen, M. G. & Gerdes, K. Bacterial persistence by RNA endonucleases. *Proc Natl Acad Sci U S A* **108**, 13206-13211, doi:10.1073/pnas.1100186108 (2011).

65 Gerdes, K. & Maisonneuve, E. Bacterial persistence and toxin-antitoxin loci. *Annu. Rev. Microbiol.* **66**, 103-123, doi:10.1146/annurev-micro-092611-150159 (2012).

66 Maisonneuve, E. & Gerdes, K. Molecular mechanisms underlying bacterial persisters. *Cell* **157**, 539-548, doi:10.1016/j.cell.2014.02.050 (2014).

67 Goeders, N. & Van Melderen, L. Toxin-antitoxin systems as multilevel interaction systems. *Toxins (Basel)* **6**, 304-324, doi:10.3390/toxins6010304 (2014).

68 Yamaguchi, Y. & Inouye, M. Regulation of growth and death in *Escherichia coli* by toxin-antitoxin systems. *Nat. Rev. Microbiol.* **9**, 779-790, doi:10.1038/nrmicro2651 (2011).

69 Chan, W. T., Espinosa, M. & Yeo, C. C. Keeping the Wolves at Bay: Antitoxins of Prokaryotic Type II Toxin-Antitoxin Systems. *Front Mol Biosci* **3**, 9: 1-20, doi:10.3389/fmolb.2016.00009 (2016).

70 Overgaard, M., Borch, J. & Gerdes, K. RelB and RelE of Escherichia coli form a tight complex that represses transcription via the ribbon-helix-helix motif in RelB. *J. Mol. Biol.* **394**, 183-196, doi:10.1016/j.jmb.2009.09.006 (2009).

71 Vandervelde, A. *et al.* Molecular mechanism governing ratio-dependent transcription regulation in the ccdAB operon. *Nucleic Acids Res.* **45**, 2937-2950, doi:10.1093/nar/gkx108 (2017).

72 Hayes, F. & Kedzierska, B. Regulating toxin-antitoxin expression: controlled detonation of intracellular molecular timebombs. *Toxins (Basel)* **6**, 337-358, doi:10.3390/toxins6010337 (2014).

73 Unterholzner, S. J., Poppenberger, B. & Rozhon, W. Toxin-antitoxin systems: Biology, identification, and application. *Mob Genet Elements* **3**, e26219, doi:10.4161/mge.26219 (2013).

74 Cataudella, I., Trusina, A., Sneppen, K., Gerdes, K. & Mitarai, N. Conditional cooperativity in toxin-antitoxin regulation prevents random toxin activation and promotes fast translational recovery. *Nucleic Acids Res.* **40**, 6424-6434, doi:10.1093/nar/gks297 (2012).

75 Otsuka, Y. *et al.* IscR regulates RNase LS activity by repressing *rnlA* transcription. *Genetics* **185**, 823-830, doi:10.1534/genetics.110.114462 (2010).

76 Quax, T. E. *et al.* Differential translation tunes uneven production of operon-encoded proteins. *Cell Rep* **4**, 938-944, doi:10.1016/j.celrep.2013.07.049 (2013).

77 Rex, G., Surin, B., Besse, G., Schneppe, B. & McCarthy, J. E. The mechanism of translational coupling in *Escherichia coli*. Higher order structure in the *atpHA* mRNA acts as a conformational switch regulating the access of de novo initiating ribosomes. *J. Biol. Chem.* **269**, 18118-181127 (1994).

78 Karp, P. D. *et al.* The EcoCyc Database. *EcoSal Plus* **6**, ESP-0009-2013: 0001-0013, doi:10.1128/ecosalplus.ESP-0009-2013 (2014).

79 Burkhardt, D. H. *et al.* Operon mRNAs are organized into ORF-centric structures that predict translation efficiency. *Elife* **6**, e22037: 22031-22023, doi:10.7554/eLife.22037 (2017).

80 Armalyte, J., Jurenaite, M., Beinoraviciute, G., Teiserskas, J. & Suziedeliene, E. Characterization of *Escherichia coli dinJ-yafQ* toxin-antitoxin system using

insights from mutagenesis data. *J. Bacteriol.* **194**, 1523-1532, doi:10.1128/JB.06104-11 (2012).

81 Stanger, F. V., Harms, A., Dehio, C. & Schirmer, T. Crystal Structure of the *Escherichia coli* Fic Toxin-Like Protein in Complex with Its Cognate Antitoxin. *PLoS One* **11**, e0163654: 0163651-0163621, doi:10.1371/journal.pone.0163654 (2016).

82 Jorgensen, M. G., Pandey, D. P., Jaskolska, M. & Gerdes, K. HicA of Escherichia coli defines a novel family of translation-independent mRNA interferases in bacteria and archaea. *J. Bacteriol.* **191**, 1191-1199, doi:10.1128/JB.01013-08 (2009).

83 Masuda, H. & Inouye, M. Toxins of Prokaryotic Toxin-Antitoxin Systems with Sequence-Specific Endoribonuclease Activity. *Toxins (Basel)* **9**, 140: 141-123, doi:10.3390/toxins9040140 (2017).

84 Zhang, Y., Zhang, J., Hara, H., Kato, I. & Inouye, M. Insights into the mRNA cleavage mechanism by MazF, an mRNA interferase. *J. Biol. Chem.* **280**, 3143-3150, doi:10.1074/jbc.M411811200 (2005).

85 Brown, B. L. *et al.* Three dimensional structure of the MqsR:MqsA complex: a novel TA pair comprised of a toxin homologous to RelE and an antitoxin with unique properties. *PLoS Pathog* **5**, e1000706: 1000701-1000715, doi:10.1371/journal.ppat.1000706 (2009).

86 Schmidt, O. *et al. prlF* and *yhaV* encode a new toxin-antitoxin system in *Escherichia coli*. *J. Mol. Biol.* **372**, 894-905, doi:10.1016/j.jmb.2007.07.016 (2007).

87 Pedersen, K. *et al.* The bacterial toxin RelE displays codon-specific cleavage of mRNAs in the ribosomal A site. *Cell* **112**, 131-140, doi:http://doi.org/10.1016/S0092-8674(02)01248-5 (2003).

88 Zhang, Y., Yamaguchi, Y. & Inouye, M. Characterization of YafO, an *Escherichia coli* toxin. *J. Biol. Chem.* **284**, 25522-25531, doi:10.1074/jbc.M109.036624 (2009).

89 Grady, R. & Hayes, F. Axe-Txe, a broad-spectrum proteic toxin-antitoxin system specified by a multidrug-resistant, clinical isolate of *Enterococcus faecium*. *Mol. Microbiol.* **47**, 1419-1432 (2003).

90 Li, G. W., Burkhardt, D., Gross, C. & Weissman, J. S. Quantifying absolute protein synthesis rates reveals principles underlying allocation of cellular resources. *Cell* **157**, 624-635, doi:10.1016/j.cell.2014.02.033 (2014).

91 Ingolia, N. T. Ribosome profiling: new views of translation, from single codons to genome scale. *Nat. Rev. Genet.* **15**, 205-213, doi:10.1038/nrg3645 (2014).

92 Salis, H. M., Mirsky, E. A. & Voigt, C. A. Automated design of synthetic ribosome binding sites to control protein expression. *Nat. Biotechnol.* **27**, 946-950, doi:10.1038/nbt.1568 (2009).

93 Espah Borujeni, A., Channarasappa, A. S. & Salis, H. M. Translation rate is controlled by coupled trade-offs between site accessibility, selective RNA unfolding and sliding at upstream standby sites. *Nucleic Acids Res.* **42**, 2646-2659, doi:10.1093/nar/gkt1139 (2014).

94 Seo, S. W. *et al.* Predictive design of mRNA translation initiation region to control prokaryotic translation efficiency. *Metab. Eng.* **15**, 67-74, doi:10.1016/j.ymben.2012.10.006 (2013).

95 Barrick, D. *et al.* Quantitative analysis of ribosome binding sites in *E. coli*. *Nucleic Acids Res.* **22**, 1287-1295 (1994).

96 Naka, K., Koga, M., Yonesaki, T. & Otsuka, Y. RNase HI stimulates the activity of RnlA toxin in *Escherichia coli*. *Mol. Microbiol.* **91**, 596-605, doi:10.1111/mmi.12479 (2014).

97 Naka, K., Qi, D., Yonesaki, T. & Otsuka, Y. RnlB Antitoxin of the *Escherichia coli* RnlA-RnlB Toxin-Antitoxin Module Requires RNase HI for Inhibition of RnlA Toxin Activity. *Toxins (Basel)* **9**, 29: 21-13, doi:10.3390/toxins9010029 (2017).

98 Li, G. Y., Zhang, Y., Inouye, M. & Ikura, M. Structural mechanism of transcriptional autorepression of the *Escherichia coli* RelB/RelE antitoxin/toxin module. *J. Mol. Biol.* **380**, 107-119, doi:10.1016/j.jmb.2008.04.039 (2008).

99 Meysman, P. *et al.* COLOMBOS v2.0: an ever expanding collection of bacterial expression compendia. *Nucleic Acids Res.* **42**, D649-653, doi:10.1093/nar/gkt1086 (2014).

100 Lal, A., Krishna, S. & Seshasayee, A. S. N. Regulation of global transcription in *E. coli* by Rsd and 6S RNA. 1-55, doi:http://dx.doi.org/10.1101/058339 (2016).

101 Kearse, M. *et al.* Geneious Basic: an integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics* **28**, 1647-1649, doi:10.1093/bioinformatics/bts199 (2012).

102 van der Walt, S., Colbert, S. C. & Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Comput Sci Eng* **13**, 22-30 (2011).

103 Deter, H. S., Dies, M., Cameron, C. C., Butzin, N. C. & Buceta, J. A Cell Segmentation/Tracking Tool Based on Machine Learning. *Methods Mol Biol* **2040**, 399-422, doi:10.1007/978-1-4939-9686-5_19 (2019).

104 Rosenfeld, N., Young, J. W., Alon, U., Swain, P. S. & Elowitz, M. B. Gene regulation at the single-cell level. *Science* **307**, 1962-1965, doi:10.1126/science.1106914 (2005).

105 Campos, M. *et al.* A constant size extension drives bacterial cell size homeostasis. *Cell* **159**, 1433-1446, doi:10.1016/j.cell.2014.11.022 (2014).

106 Brehm-Stecher, B. F. & Johnson, E. A. Single-cell microbiology: tools, technologies, and applications. *Microbiol. Mol. Biol. Rev.* **68**, 538-559, table of contents, doi:10.1128/MMBR.68.3.538-559.2004 (2004).

107 Ferry, M. S., Razinkov, I. A. & Hasty, J. Microfluidics for synthetic biology: from design to execution. *Methods Enzymol.* **497**, 295-372, doi:10.1016/B978-0-12-385075-1.00014-7 (2011).

108 Nketia, T. A., Sailem, H., Rohde, G., Machiraju, R. & Rittscher, J. Analysis of live cell images: Methods, tools and opportunities. *Methods* **115**, 65-79, doi:10.1016/j.ymeth.2017.02.007 (2017).

109 Vallotton, P., Turnbull, L., Whitchurch, C. B. & Mililli, L. Segmentation of Dense 2D Bacilli Populations. *2010 International Conference on Digital Image Computing: Techniques and Applications*, 82-86 (2010).

110 Chowdhury, S., Kandhavelu, M., Yli-Harja, O. & Ribeiro, A. S. Cell segmentation by multi-resolution analysis and maximum likelihood estimation (MAMLE). *BMC Bioinformatics* **14 Suppl 10**, S8, doi:10.1186/1471-2105-14-S10-S8 (2013).

111 Sadanandan, S. K. *et al.* Segmentation and Track-Analysis in Time-Lapse Imaging of Bacteria. *Ieee J-Stsp* **10**, 174-184, doi:10.1109/Jstsp.2015.2491304 (2016).

112 Hu, Y. *et al.* Trajectory energy minimization for cell growth tracking and genealogy analysis. *R Soc Open Sci* **4**, 170207, doi:10.1098/rsos.170207 (2017).

113 Ducret, A., Quardokus, E. M. & Brun, Y. V. MicrobeJ, a tool for high throughput bacterial cell detection and quantitative analysis. *Nat Microbiol* **1**, 16077, doi:10.1038/nmicrobiol.2016.77 (2016).

114 Boyle, E. A., Li, Y. I. & Pritchard, J. K. An Expanded View of Complex Traits: From Polygenic to Omnigenic. *Cell* **169**, 1177-1186, doi:10.1016/j.cell.2017.05.038 (2017).

115 Schindelin, J. *et al.* Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9**, 676-682, doi:10.1038/nmeth.2019 (2012).

116 Paintdakhi, A. *et al.* Oufti: an integrated software package for high-accuracy, high-throughput quantitative microscopy analysis. *Mol Microbiol* **99**, 767-777, doi:10.1111/mmi.13264 (2016).

117 Dimopoulos, S., Mayer, C. E., Rudolf, F. & Stelling, J. Accurate cell segmentation in microscopy images using membrane patterns. *Bioinformatics* **30**, 2644-2651, doi:10.1093/bioinformatics/btu302 (2014).

118 Kamentsky, L. *et al.* Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software. *Bioinformatics* **27**, 1179-1180, doi:10.1093/bioinformatics/btr095 (2011).

119 Stylianidou, S., Brennan, C., Nissen, S. B., Kuwada, N. J. & Wiggins, P. A. SuperSegger: robust image segmentation, analysis and lineage tracking of bacterial cells. *Mol Microbiol* **102**, 690-700, doi:10.1111/mmi.13486 (2016).

120 Arganda-Carreras, I. *et al.* Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics* **33**, 2424-2426, doi:10.1093/bioinformatics/btx180 (2017).

121 Deter, H. S. *et al.* Proteolytic Queues at ClpXP Increase Antibiotic Tolerance. *ACS Synth Biol* **9**, 95-103, doi:10.1021/acssynbio.9b00358 (2020).

122 Oracle. (2018).

123 Canonical Ltd. Ubuntu. (2018).

124 Rueden, C. T. *et al.* ImageJ2: ImageJ for the next generation of scientific image data. *BMC Bioinformatics* **18**, 529, doi:10.1186/s12859-017-1934-z (2017).

125 Anaconda, I. (2018).

126 Python Software Foundation. (Python Software Foundation, 2018).

127 (2018).

128 Deter, H. S. *CellTracking*, <https://github.com/hdeter/CellTracking > (2018).

129 Deter, H. S., Dies, M., Cameron, C. C., Butzin, N. C. & Buceta, J. A Bacteria Segmentation/Tracking Tool based on Machine Learning doi:http://doi.org/10.17605/OSF.IO/GDXEN (2018).

130 Arganda-Carreras, I. *et al.* Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics*, doi:10.1093/bioinformatics/btx180 (2017).

131 Baker, S., Thomson, N., Weill, F. X. & Holt, K. E. Genomic insights into the emergence and spread of antimicrobial-resistant bacterial pathogens. *Science* **360**, 733-738, doi:10.1126/science.aar3777 (2018).

132 Durao, P., Balbontin, R. & Gordo, I. Evolutionary Mechanisms Shaping the Maintenance of Antibiotic Resistance. *Trends Microbiol.*, doi:10.1016/j.tim.2018.01.005 (2018).

133 Xie, Y. *et al.* TADB 2.0: an updated database of bacterial type II toxin-antitoxin loci. *Nucleic Acids Res.*, doi:10.1093/nar/gkx1033 (2017).

134 Horesh, G. *et al.* SLING: a tool to search for linked genes in bacterial datasets. *Nucleic Acids Res.*, doi:10.1093/nar/gky738 (2018).

135 Moll, I. & Engelberg-Kulka, H. Selective translation during stress in Escherichia coli. *Trends Biochem. Sci* **37**, 493-498, doi:10.1016/j.tibs.2012.07.007 (2012).

136 Soo, V. W., Cheng, H. Y., Kwan, B. W. & Wood, T. K. de novo synthesis of a bacterial toxin/antitoxin system. *Sci Rep* **4**, 4807, doi:10.1038/srep04807 (2014).

137 Luidalepp, H., Hallier, M., Felden, B. & Tenson, T. tmRNA decreases the bactericidal activity of aminoglycosides and the susceptibility to inhibitors of cell wall synthesis. *RNA Biol* **2**, 70-74, doi:10.4161/rna.2.2.2020 (2005).

138 Liu, S. *et al.* Variable Persister Gene Interactions with (p)ppGpp for Persister Formation in *Escherichia coli*. *Front Microbiol* **8**, 1795, doi:10.3389/fmicb.2017.01795 (2017).

139 Amato, S. M. & Brynildsen, M. P. Persister Heterogeneity Arising from a Single Metabolic Stress. *Curr. Biol.* **25**, 2090-2098, doi:10.1016/j.cub.2015.06.034 (2015).

140 Wilmaerts, D., Windels, E. M., Verstraeten, N. & Michiels, J. General Mechanisms Leading to Persister Formation and Awakening. *Trends Genet.*, doi:10.1016/j.tig.2019.03.007 (2019).

141 Cookson, N. A. *et al.* Queueing up for enzymatic processing: correlated signaling through coupled degradation. *Mol. Syst. Biol.* **7**, 561, doi:10.1038/msb.2011.94 (2011).

142 Butzin, N. C. & Mather, W. H. in *Reviews in Cell Biology and Molecular Medicine* (Wiley-VCH Verlag GmbH & Co. KGaA, 2016).

143 Butzin, N. C., Hochendoner, P., Ogle, C. T. & Mather, W. H. Entrainment of a Bacterial Synthetic Gene Oscillator through Proteolytic Queueing. *ACS Synth Biol* **6**, 455-462, doi:10.1021/acssynbio.6b00157 (2017).

144 Butzin, N. C., Hochendoner, P., Ogle, C. T., Hill, P. & Mather, W. H. Marching along to an Offbeat Drum: Entrainment of Synthetic Gene Oscillators by a Noisy Stimulus. *ACS Synth Biol* **5**, 146-153, doi:10.1021/acssynbio.5b00127 (2016).

145 Butzin, N. C. & Mather, W. H. Crosstalk between Diverse Synthetic Protein Degradation Tags in Escherichia coli. *ACS Synth Biol* **7**, 54-62, doi:10.1021/acssynbio.7b00122 (2018).

146 Mondragon-Palomino, O., Danino, T., Selimkhanov, J., Tsimring, L. & Hasty, J. Entrainment of a population of synthetic genetic oscillators. *Science* **333**, 1315-1319, doi:10.1126/science.1205369 (2011).

147 Prindle, A. *et al.* Genetic Circuits in Salmonella typhimurium. *ACS Synth Biol* **1**, 458-464, doi:10.1021/sb300060e (2012).

148 Thomsen, L. E., Olsen, J. E., Foster, J. W. & Ingmer, H. ClpP is involved in the stress response and degradation of misfolded proteins in Salmonella enterica serovar Typhimurium. *Microbiology* **148**, 2727-2733, doi:10.1099/00221287-148-9-2727 (2002).

149 Weichart, D., Querfurth, N., Dreger, M. & Hengge-Aronis, R. Global role for ClpP-containing proteases in stationary-phase adaptation of Escherichia coli. *J. Bacteriol.* **185**, 115-125, doi:10.1128/jb.185.1.115-125.2003 (2003).

150 Harms, A., Fino, C., Sorensen, M. A., Semsey, S. & Gerdes, K. Prophages and Growth Dynamics Confound Experimental Results with Antibiotic-Tolerant Persister Cells. *MBio* **8**, doi:10.1128/mBio.01964-17 (2017).

151 Shan, Y. *et al.* ATP-Dependent Persister Formation in *Escherichia coli*. *MBio* **8**, doi:10.1128/mBio.02267-16 (2017).

152 Lopatkin, A. J. *et al.* Bacterial metabolic state more accurately predicts antibiotic lethality than growth rate. *Nat. Microbiol.*, doi:10.1038/s41564-019-0536-0 (2019).

153 Pontes, M. H. & Groisman, E. A. Slow growth determines nonheritable antibiotic resistance in *Salmonella enterica*. *Sci Signal* **12**, doi:10.1126/scisignal.aax3938 (2019).

154 Flynn, J. M., Neher, S. B., Kim, Y. I., Sauer, R. T. & Baker, T. A. Proteomic discovery of cellular substrates of the ClpXP protease reveals five classes of ClpX-recognition signals. *Mol. Cell* **11**, 671-683, doi:10.1016/s1097-2765(03)00060-1 (2003).

155 Shimizu, K. in *Bacterial Cellular Metabolic Systems* 95-213 (2013).

156 Wong, G. T. *et al.* Genome-Wide Transcriptional Response to Varying RpoS Levels in Escherichia coli K-12. *J. Bacteriol.* **199**, doi:10.1128/JB.00755-16 (2017).

157 Peterson, C. N., Levchenko, I., Rabinowitz, J. D., Baker, T. A. & Silhavy, T. J. RpoS proteolysis is controlled directly by ATP levels in Escherichia coli. *Genes Dev.* **26**, 548-553, doi:10.1101/gad.183517.111 (2012).

158 Hooper, D. C., Wolfson, J. S., Ng, E. Y. & Swartz, M. N. Mechanisms of action of and resistance to ciprofloxacin. *Am. J. Med.* **82**, 12-20 (1987).

159 Kohanski, M. A., Dwyer, D. J. & Collins, J. J. How antibiotics kill bacteria: from targets to networks. *Nat. Rev. Microbiol.* **8**, 423-435, doi:10.1038/nrmicro2333 (2010).

160 Hong, W., Zeng, J. & Xie, J. Antibiotic drugs targeting bacterial RNAs. *Acta Pharm Sin B* **4**, 258-265, doi:10.1016/j.apsb.2014.06.012 (2014).

161 Kwan, B. W., Valenta, J. A., Benedik, M. J. & Wood, T. K. Arrested protein synthesis increases persister-like cell formation. *Antimicrob. Agents Chemother.* **57**, 1468-1473, doi:10.1128/AAC.02135-12 (2013).

162 Kussell, E., Kishony, R., Balaban, N. Q. & Leibler, S. Bacterial persistence: a model of survival in changing environments. *Genetics* **169**, 1807-1814, doi:10.1534/genetics.104.035352 (2005).

163 Ganini, D. *et al.* Fluorescent proteins such as eGFP lead to catalytic oxidative stress in cells. *Redox Biol* **12**, 462-468, doi:10.1016/j.redox.2017.03.002 (2017).

164 Kalyanaraman, B. & Zielonka, J. Green fluorescent proteins induce oxidative stress in cells: A worrisome new wrinkle in the application of the GFP reporter system to biological systems? *Redox Biol* **12**, 755-757, doi:10.1016/j.redox.2017.03.019 (2017).

165 Cohen, N. R., Lobritz, M. A. & Collins, J. J. Microbial persistence and the road to drug resistance. *Cell Host Microbe* **13**, 632-642, doi:10.1016/j.chom.2013.05.009 (2013).

166 Wang, T., El Meouche, I. & Dunlop, M. J. Bacterial persistence induced by salicylate via reactive oxygen species. *Sci Rep* **7**, 43839, doi:10.1038/srep43839 (2017).

167 Trastoy, R. *et al.* Mechanisms of Bacterial Tolerance and Persistence in the Gastrointestinal and Respiratory Environments. *Clin. Microbiol. Rev.* **31**, doi:10.1128/CMR.00023-18 (2018).

168 Frees, D., Gerth, U. & Ingmer, H. Clp chaperones and proteases are central in stress survival, virulence and antibiotic resistance of Staphylococcus aureus. *Int. J. Med. Microbiol.* **304**, 142-149, doi:10.1016/j.ijmm.2013.11.009 (2014).

169 Nikolic, N. Autoregulation of bacterial gene expression: lessons from the MazEF toxin-antitoxin system. *Curr. Genet.*, doi:10.1007/s00294-018-0879-8 (2018).

170 Ramisetty, B. C., Ghosh, D., Roy Chowdhury, M. & Santhosh, R. S. What Is the Link between Stringent Response, Endoribonuclease Encoding Type II Toxin-Antitoxin Systems and Persistence? *Front Microbiol* **7**, 1882, doi:10.3389/fmicb.2016.01882 (2016).

171 Radzikowski, J. L. *et al.* Bacterial persistence is an active sigmaS stress response to metabolic flux limitation. *Mol. Syst. Biol.* **12**, 882, doi:10.15252/msb.20166998 (2016).

172 Rossi, N. A., Mora, T., Walczak, A. M. & Dunlop, M. J. Active degradation of MarA controls coordination of its downstream targets. *PLoS Comput Biol* **14**, e1006634, doi:10.1371/journal.pcbi.1006634 (2018).

173 Morgan, G. J., Burkhardt, D. H., Kelly, J. W. & Powers, E. T. Translation efficiency is maintained at elevated temperature in *Escherichia coli*. *J. Biol. Chem.* **293**, 777-793, doi:10.1074/jbc.RA117.000284 (2018).

174 Hobman, J. L., Penn, C. W. & Pallen, M. J. Laboratory strains of *Escherichia coli*: model citizens or deceitful delinquents growing old disgracefully? *Mol. Microbiol.* **64**, 881-885, doi:10.1111/j.1365-2958.2007.05710.x (2007).

175 Mazumder, M. *et al.* An *Escherichia coli* system for evolving improved light-controlled DNA-binding proteins. *Protein Eng. Des. Sel.* **28**, 293-302, doi:10.1093/protein/gzv033 (2015).

176 Forster, A. H., Beblawy, S., Golitsch, F. & Gescher, J. Electrode-assisted acetoin production in a metabolically engineered *Escherichia coli* strain. *Biotechnol Biofuels* **10**, 65, doi:10.1186/s13068-017-0745-9 (2017).

177 Wentzel, A., Christmann, A., Adams, T. & Kolmar, H. Display of passenger proteins on the surface of *Escherichia coli* K-12 by the enterohemorrhagic *E. coli* intimin EaeA. *J. Bacteriol.* **183**, 7273-7284, doi:10.1128/JB.183.24.7273-7284.2001 (2001).

178 O'Brien, S. P. & DeLisa, M. P. Functional reconstitution of a tunable E3-dependent sumoylation pathway in *Escherichia coli*. *PLoS One* **7**, e38671, doi:10.1371/journal.pone.0038671 (2012).

179 Gefen, O., Gabay, C., Mumcuoglu, M., Engel, G. & Balaban, N. Q. Single-cell protein induction dynamics reveals a period of vulnerability to antibiotics in persister bacteria. *Proc Natl Acad Sci U S A* **105**, 6145-6149, doi:10.1073/pnas.0711712105 (2008).

180 Mok, W. W., Patel, N. H. & Li, Y. Decoding toxicity: deducing the sequence requirements of IbsC, a type I toxin in Escherichia coli. *J. Biol. Chem.* **285**, 41627-41636, doi:10.1074/jbc.M110.149179 (2010).

181 Kaspy, I. *et al.* HipA-mediated antibiotic persistence via phosphorylation of the glutamyl-tRNA-synthetase. *Nat Commun* **4**, 3001, doi:10.1038/ncomms4001 (2013).

182 Miller, J. H. *Experiments in molecular genetics*. (Cold Spring Harbor Laboratory, 1972).

183 Joers, A., Kaldalu, N. & Tenson, T. The frequency of persisters in *Escherichia coli* reflects the kinetics of awakening from dormancy. *J. Bacteriol.* **192**, 3379-3384, doi:10.1128/JB.00056-10 (2010).

184 Nickolai, D. J. *et al.* Effects of storage temperature and pH on the stability of eleven beta-lactam antibiotics in MIC trays. *J. Clin. Microbiol.* **21**, 366-370 (1985).

185 Datla, U. S. *et al.* The spatiotemporal system dynamics of acquired resistance in an engineered microecology. *Sci Rep* **7**, 16071, doi:10.1038/s41598-017-16176-w (2017).

186 Gillespie, D. T. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry* **81**, 2340-2361, doi:10.1021/j100540a008 (1977).

187 Stefan Behnel, R. B., Dag Sverre Seljebotn, Greg Ewing, et al. *The Cython compiler*, <http://cython.org.> (2008).

188 E Jones, T. O., P Peterson, et al. Scipy: Open source scientic tools for python [software]. (2001).

189 Cumming, G., Fidler, F. & Vaux, D. L. Error bars in experimental biology. *J. Cell Biol.* **177**, 7-11, doi:10.1083/jcb.200611141 (2007).

190 Widdel, F. Theory and measurement of bacterial growth. *Di dalam Grundpraktikum Mikrobiologie* **4**, 1-11 (2007).