Simulating Bacterial Growth, Competition, and Resistance with Agent-Based Models and Laboratory Experiments

Anne E. Yust and Davida S. Smyth

Abstract In this chapter, we will introduce the concepts of growth, competition, and adaptation using bacteria. We will use both simulation and laboratory-based exercises and provide practice with critical skills to assess your understanding throughout the chapter. The chapter ends by focusing on the capacious and global problem of antibiotic resistance, considered to be one of the most important public health threats of the twenty-first century. We will review the basic biological concepts underlying the phenomenon, and introduce the mathematical content necessary to begin to create agent-based models in order to simulate and analyze antibiotic resistance and its effect on the planet. we will present a selection of research projects throughout the later portion of the chapter for you to further explore antibiotic resistance and other contexts of bacteria.

Suggested Prerequisites *This chapter is intended to be accessible to a wide range of undergraduates. The only necessary prerequisite is a strong interest and desire to learn about simulation, bacteria, and antibiotic resistance. We do recommended previous exposure to microbes and introductory biology, programming—especially with* NetLogo*, mathematical modeling, and data analysis.*

1 Introduction

Bacteria are microscopic, unicellular, prokaryotic organisms that can exist in the form of spheres, rods, and spirals. The major category which differentiates bacteria is the structure of their cell walls, which are called *gram-positive* or *gram-negative*. The prototypical gram-negative organism, *Escherichia coli*, has an inner and outer

A. E. Yust (✉) · D. S. Smyth
Department of Natural Sciences and Mathematics, Eugene Lang College of Liberal Arts at The New School, New York, NY, USA
e-mail: yusta@newschool.edu; smythd@newschool.edu

H. Callender Highlander et al. (eds.), *An Introduction to Undergraduate Research in Computational and Mathematical Biology*, Foundations for Undergraduate Research in Mathematics, https://doi.org/10.1007/978-3-030-33645-5_5

membrane, between which is found a periplasmic space with a thin layer of peptidoglycan. Gram-positive bacteria have a much thicker layer of peptidoglycan surrounding their inner membrane and have no outer membrane. The prototype gram-positive organism is *Staphylococcus aureus*.

Bacteria are noted for their ubiquity; they are some of the most adaptable and resilient organisms on the planet. Different species of bacteria can survive and grow at temperatures ranging from −10 to 100 °C. They can happily reside in cold salty ponds and in the frigid waters of the polar regions to the boiling water of hot springs. They can even be found growing in the vicinity of thermal volcanic vents at the bottom of the oceans where temperatures exceed 300 °C. Bacteria are also resilient to pH; they have been isolated in acid wastes from mines and in the alkaline waters of soda lakes [63]. They have been isolated in black anaerobic silts of estuaries and in the purest waters of biologically unproductive or oligotrophic lakes. This means that, in whatever situation, bacteria can find a way of adapting and surviving, regardless of the environment they find themselves [63].

Bacteria have proven a useful model system in which to investigate many cellular functions and processes. They have simple genomes, many of which are amenable to genetic modification. They can also be readily propagated in the laboratory, and they have fast generation times. Knowledge gained when studying bacterial systems can often be applied to homologous proteins in more complex higher organisms. Bacteria are used in industry and are critical for the production of yoghurt, cheese, sour cream, pickles, sauerkraut, and kombucha to name but a few. Bacteria can also be engineered to make useful products such as human proteins and drugs, and importantly they can be used in bioremediation and to detoxify poisonous substances. It is their adaptability and resilience that not only makes bacteria one of the world's greatest allies but also one of the world's greatest foes. Bacteria are responsible for many different diseases and a major cause of morbidity and mortality across the globe. One of the most important and clinically relevant bacterial adaptations is antibiotic resistance.

1.1 Introduction to Antibiotic Resistance

The World Health Organization has named antibiotic resistance as one of the most important public health threats of the twenty-first century [90]. Importantly, infections caused by antibiotic-resistant organisms are associated with significant mortality [3] and are an important economic burden, estimated to cost over $20 billion per year in the USA alone [35, 39, 113]. At least 23,000 people die from infections with antibiotic-resistant bacteria annually in the USA as estimated by the Centers for Disease Control and Prevention [3]. By 2050, it is estimated that antibiotic resistance will cause around 300 million premature deaths, and result in a loss of up to $100 trillion by the global economy [89]. More worryingly still, the World Health Organization has warned that there is a serious lack of new antibiotics

under development to combat the growing threat of antibiotic resistance [8], with only eight of the 51 new antibiotics and biological agents currently in clinical development to treat antibiotic-resistant pathogens adding value to the current drugs on offer [62].

Antibiotic resistance is an ancient phenomenon, a consequence of long-term biowarfare among organisms in their natural environments. Most antimicrobials are natural molecules. In fact, many are secreted by bacteria and other environmental organisms. When organisms growing in communities were exposed to the effects of antibiotics secreted by members of the community, they evolved counter-mechanisms to overcome their action in order to survive. Some organisms that are naturally resistant to antibiotics are called *intrinsically resistant*. A great example of intrinsic resistance occurs in multi-drug resistant gram-negative bacteria such as *E. coli*, which are insensitive to many types of antibiotics used to effectively treat gram-positive bacteria. Their resistance is due to the presence of the outer membrane, which is the differentiating factor between gram-negative and gram-positive bacteria. The outer membrane is impermeable to many molecules. Additionally, these strains possess a variety of efflux pumps that can effectively pump out the antimicrobial from the cells [84]. Many environmental organisms are prolifically and intrinsically resistant to many different classes of antibiotics, and in particular, those that dwell in the soil have many uncharacterized mechanisms of resistance [38]. What is more intriguing is that antibiotic resistance often predates the clinical use of antibiotics and has emerged independently of the selective pressure imposed by using antibiotics [25, 37]. Scientists consider environmental organisms, such as those found in soil or in the urban environment, to be important reservoirs of novel resistance genes that could be transferred to pathogens. This presents a major health concern [59].

While environmental organisms often have intrinsic mechanisms of resistance, antibiotic-resistant bacteria in the hospital or clinical setting often exhibit resistance that has been acquired. Acquired resistance emerges in a bacterial population that was originally susceptible to the antibiotic. In response to exposure, resistance is often acquired by mutations in the chromosome of the bacteria or the acquisition of external resistance-encoding genes, i.e., *horizontal gene transfer (HGT)*.

1.1.1 Genetic Basis of Antibiotic Resistance

Antibiotic resistance emerges genetically in two main ways:

- **Resistance by mutation.** Cells within a susceptible population of bacteria develop mutations in genes that ameliorate the activity of the drug. These cells thus survive the antimicrobial agent, multiply, and proliferate, while the susceptible cells succumb to the agent. Depending on the type of mutation, there can be a fitness cost, and as a consequence, these mutations are only selected for in the presence of the antibiotic. Interestingly, the use of antibiotics has

been shown to increase the mutation rate of bacteria [68] and even to select for mutants with higher mutation rates in the microbial flora of patients treated with antibiotics [53].

- **Resistance by horizontal gene transfer.** HGT is a major driver of bacterial evolution. This process involves the acquisition of foreign DNA, which could contain genetic sequences that transfer the antibiotic resistance. HGT is frequently responsible for acquired resistance to antibiotics and antimicrobials. The most common mechanisms used by bacteria to acquire external genetic material are transformation (incorporation of naked DNA), transduction (mediated by phages—viruses that infect bacteria), and conjugation (when bacteria have "sex" mediated by a pilus). The simplest type of HGT, transformation, is demonstrated by a small number of species of high clinical relevance including the pneumococcus or *Streptococcus pneumoniae* [86], and *Neisseria meningitidis* and *Neisseria gonorrhoeae* [116]. Transduction by phages is a very important mode of HGT and has been shown to be an important vehicle for resistance genes in the environment [19]. The most common and most efficient form of HGT is conjugation. This type of transfer needs cell-to-cell contact and is mediated by the presence of conjugative elements in the genome of the donor cell. Tetracycline resistance is readily transferred among *N. gonorrhoeae* and *Enterococcus faecalis* strains by means of a conjugative plasmids, circular forms of DNA [71, 116]. Other types of mobile DNA such as integrons and transposons also play important roles in the dissemination of antibiotic resistance genes, such as carbapenamases [97]. Genes encoding resistance to streptomycin, spectinomycin, and sulfonamides as well as metals such as mercury have been found on complex transposons and plasmids in members of the Enterobacteriaceae [23].

1.1.2 Mechanistic Basis of Antibiotic Resistance

There are several categorizations of antibiotic resistance mechanisms:

- **Modifying the antimicrobial molecule itself.** One mechanism found in both gram-positive and gram-negative bacteria is to produce enzymes that modify the chemical composition of the antimicrobial molecule by phosphorylation, acetylation, and adenylation. Chloramphenicol resistance is mediated by chloramphenicol acetyltransferases known as CATs, widespread among bacteria [103]. Alternatively, some bacteria produce enzymes that can destroy the antibiotic itself. One of the most famous examples is the family of beta-lactamases. Beta-lactamases were identified before the introduction of penicillin to the market [12] and are considered to be ancient. More than 1000 different beta-lactamases have been described to date.
- **Blocking the action of the antibiotic against its target.** The first line of defense used by gram-negative bacteria to prevent antimicrobials from reaching their intracellular or periplasmic targets is their outer membrane as we have

discussed. In addition, they can prevent hydrophilic (water soluble) molecules from traversing the membrane (which is not water soluble) using porins (channels in the membrane) by altering the types of porins present, the expression of the porin genes and by impeding porin function [85]. Efflux is another widespread mechanism to avoid antibiotic action. *E. coli* can actively pump the antibiotic tetracycline out of the cell using an efflux pump. Efflux pump mechanisms are found in both gram-positive and gram-negative bacteria and can be antibiotic specific such as *mef* which encodes macrolide resistance in pneumococci or can be broadly specific facilitating the multi-drug resistance (MDR) phenotype [98].

- **Changing the target site or bypassing it entirely.** This occurs through two main mechanisms, which include protection of the target and modifications of the target site, decreasing affinity for the antibiotic. Tet(M) first described in *Streptococcus* spp., interacts with the ribosome and actively dislodges tetracycline from the target site [33]. Linezolid resistance involves mutation of the binding site in the ribosome and results in decreased affinity of the drug for its ribosomal target [78]. Lastly, bacteria can evolve entirely new target structures that have the same function but bypass the antibiotic entirely, such as methicillin resistance in *S. aureus* due to the acquisition of an exogenous PBP (PBP2a) and vancomycin resistance in enterococci through modifications of the peptidoglycan structure mediated by the *van* gene clusters [17, 31].
- **Changing regulatory networks which control important metabolic pathways.** An important example of this type of resistance is resistance to daptomycin (DAP) and vancomycin (low level in *S. aureus*). In these cases, the bacteria make systematic changes to fundamental systems such as their cell wall structure to withstand the action of the drug. An example in both enterococci and *S. aureus*, YycFG (WalKR), an essential two-component regulatory system, has been implicated in cell wall synthesis and homeostasis, is important for resistance to daptomycin. The exact mechanism is unknown, but it appears to involve alteration in cell wall metabolism resulting in changes in surface charge which repulses the positively charged calcium-DAP complex from the cell envelope [22, 115]. High-level vancomycin resistance in *S. aureus* was the result of acquisition by a methicillin-resistant *S. aureus* (MRSA) strain of the *vanA* gene cluster from a vancomycin-resistant enterococcus (*E. faecalis*) isolate [107]. Thankfully such high-level resistance to the last available drug for treatment, vancomycin, is rare in Staphylococci. However, low level resistance, called vancomycin intermediate *S. aureus* (VISA), is much more prevalent and involves several systematic changes that reduce peptidoglycan cross-linking (in the cell wall) which results in a thicker cell wall. Additional changes in VISA cells include an increase in fructose utilization and fatty acid metabolism, as well as an increase in the expression of cell wall synthesis genes [56].

1.2 Spread and Severity of Antibiotic-Resistant Infections

Infections due to antibiotic-resistant bacteria are already widespread in the USA and across the planet [118]. In 2011, the Infectious Diseases Society of America (IDSA) Emerging Infections Network survey of national infectious disease specialists concluded that more than 60% of participants had seen a pan-resistant, untreatable bacterial infection within the prior year [109]. The gram-positive pathogens, *S. aureus* and Enterococcus species, are responsible for a global pandemic, which poses the biggest threat [3]. MRSA kills more Americans each year than HIV/AIDS, murder, Parkinson's, and emphysema combined [52]. Vancomycin-resistant enterococci are developing resistance to many common antibiotics [50]. Health care settings are seeing serious gram-negative infections due to resistant Enterobacteriaceae (mostly *Klebsiella pneumoniae*), *Pseudomonas aeruginosa*, and *Acinetobacter* [3], with multi-drug resistant gram-negative strains, including extended-spectrum beta-lactamase-producing *E. coli* and *N. gonorrhoeae* emerging in the community and non-health care settings [102]. A review in 2014 indicated that an estimated 700,000 deaths globally were caused by infections caused by antibiotic-resistant organisms, and predicted this number rise to 10 million per year by 2050 [4, 89].

Antibiotic-resistant bacteria and the infections they cause are having an impact on every field of medicine and have a significant impact on morbidity and mortality. It has been estimated that infections caused by antibiotic-resistant bacteria have two-fold higher rates of adverse outcomes compared with similar infections caused by susceptible strains [36]. The impacts of negative outcomes include treatment failure and/or death as well as economic impacts such as increased cost of care and length of stay due to treatment failure of the antibiotic [44]. Serious infections due to MRSA have a significantly higher case fatality rate when compared with methicillin-susceptible *S. aureus* infections [36]. Enterobacteriaceae that produce extended-spectrum beta-lactamases are associated with greater treatment failure and mortality than non-ESBL producing strains [77]. Infections due to *K. pneumoniae* with resistance to carbapenems demonstrate a two- to five-fold higher risk of death than infections caused by carbapenem-susceptible strains [29]. Forty-five percent of bacteremia cases due to carbapenem-resistant *Acinetobacter baumannii* are associated with a 14-day mortality [87].

1.3 The Economic, Social, and Civic Impacts of Antibiotic Resistance

It is also important to consider the impact that antibiotic-resistant bacteria have on local and global economies, individuals, communities, and populations and on policies, regulations, and future planning pertaining to healthcare, food production, and agriculture. The emergence of antibiotic-resistant bacteria has been called a "crisis" or "nightmare scenario" that could have "catastrophic consequences" and

in recent years has been recognized as a global threat. As a consequence it is now being recognized by governments and worldwide organizations as a target for policy generation and implementation [83]. The federal Interagency Task Force on Antimicrobial Resistance founded in 1999 succeeded in documenting collaboration and communication among the 11 agencies working on resistance issues, but it failed to set an agenda for federal response [1]. In 2013, the CDC declared that the human race is now in the "post-antibiotic era," and in 2014, the World Health Organization (WHO) warned that the antibiotic resistance crisis is becoming dire [80], stating that the problem "threatens the achievements of modern medicine. A post-antibiotic era—in which common infections and minor injuries can kill—is a very real possibility for the 21st century." Antibiotic resistance poses a substantial threat to US public health and national security according to the IDSA and the Institute of Medicine. In March 2015, the Obama administration released a National Action Plan for Combating Antibiotic-Resistant Bacteria [6] and the 2016 federal budget almost doubled the amount of federal funding for combating and preventing antibiotic resistance to more than $1.2 billion [1, 7].

1.3.1 Economic Burden of Antibiotic Resistance

Antibiotic-resistant infections pose an economic burden as well. Patients with antibiotic-resistant infections spend longer in the hospital, from 6.4 to 12.7 days, collectively adding an extra eight million hospital days [50]. The medical cost per patient infected with an antibiotic-resistant strain is estimated to be in the range of from $18,588 to $29,069 [21, 50]. The US economy faces a total economic burden estimated to be as high as $20 billion in health care costs and $35 billion a year in lost productivity due to antibiotic resistance [50]. Individual families and communities lose wages and have higher health care costs [80]. Staggeringly, the global gross domestic product could be reduced by 2–3.5% by 2050 due to the mortality from antibiotic-resistant infections, about $60 and $100 trillion [4, 89].

1.3.2 Increased Impact on Subpopulations

Many subpopulations are affected by the rise in antibiotic-resistant pathogens considerably more than others. We outline a few specific subpopulations below.

- **Developing populations.** For people in the developing world, a post-antibiotic era has already arrived. In parts of Africa, studies have shown that as many as 97% of *S. aureus* are caused by MRSA [11] and high levels of resistance to amoxicillin and penicillin in *S. pneumoniae* and *Haemophilus influenzae* have been observed, causing concern given that pneumonia is a leading cause of death in children [114]. In India and Pakistan, up to 95% of adults carry bacteria that are resistant to β-lactam antibiotics including carbapenems, where by comparison, only 10% of adults in the Queens area of New York carry such

bacteria [101]. Worryingly, not all countries are collecting data on the prevalence of these bacteria and their infections. According to the WHO only 129 of 194 member countries provided any national data on drug resistance in bacteria with only 22 countries tracking the organisms and resistance that pose the greatest threat including *S. aureus* and methicillin, *E. coli* and cephalosporins, and *K. pneumoniae* and carbapenems [101].

- **Underserved and impoverished communities.** According to 2016 US census data, the official poverty rate was 12.7%, down from 13.5% in 2015. Since 2014, the poverty rate has fallen 2.1 percentage points from 14.8 to 12.7%. This means that in 2016, 40.6 million people were living in poverty, 2.5 million fewer than in 2015 and 6.0 million fewer than in 2014 [105]. For most demographic groups, the number of people in poverty decreased from 2015. Adults aged 65 and older were the only population group to experience an increase in the number of people in poverty [105]. Many factors associated with poverty contribute to the development of antibiotic-resistant organisms, some of which impact affecting resistance in the USA [95]. Studies have shown that seniors and low-income patients obtain antimicrobials from other countries and may engage in the sharing of medications while others will save antibiotics from a regimen they did not complete and self-treat [95]. Self-treating can drive antimicrobial resistance because of the inappropriate use of antibiotics for viral illness, the antimicrobials may not work for the specific organism type and the dosage may be incorrect [95]. The high cost of healthcare and lack of access to healthcare for those who are uninsured, prevents many from seeking necessary and lifesaving treatment. The WHO has cited the provision of universal healthcare as a means to

  > improve access to appropriate and affordable treatment of infections, especially for the poor through enactment and enforcement of regulations, dissemination of treatment guidelines based on antibiotic resistance surveillance data, along with awareness raising on the responsible use of antimicrobials and the challenge of antibiotic-resistant bacteria [26].

  It is imperative to remove financial barriers and allow access to antimicrobial treatment of infections.
- **At-risk populations.** While antibiotic-resistant bacteria pose a threat to the population as a whole they are likely to cause illness in populations with greater overall risk of contracting infectious diseases. These at-risk populations include the military [32], the homeless [5], children attending daycare [9], immunocompromised persons [42], and the elderly [18]. Using prisoners as an example, community-associated MRSA outbreaks in the USA have been reported among persons incarcerated in prisons and jails with estimates of MRSA colonization in prisons as high as 80–90%. Crowding and sharing of contaminated personal items may contribute to MRSA spread among incarcerated persons [69]. Of considerable global concern, Russian prisons are said to be driving resistance among strains of TB [61].

1.3.3 Impact on the Food Supply and Agriculture

Antibiotics have been widely used in agriculture and in some countries for growth promotion [64, 89]. This practice was discontinued in the European Union in 2006 [2]. In the Americas and Asia this practice is still in use, where large scale husbandry systems contribute to infection with these bacteria. Treatment is generally delivered via the feed or water to all animals regardless of their infection status [73]. Data from the US Food and Drug Administration shows that in 2015, 74% of farm animal antibiotics were administered via feed and 21% in drinking water, for mass medication. It is estimated that the use of medically important antibiotics in food animals in the USA is approximately three times higher than human use [74]. As a consequence, antibiotic use in animals is thought to be an important selective pressure for antibiotic resistance globally [64]. Sales in the USA in 2015 of the critically important fluoroquinolones antibiotics was 20 tonnes, a 16% increase over 2014 and a 33% increase over 2013 [74]. However, in the USA since 2005, the use of fluoroquinolones has been banned in poultry due to scientific evidence that this use was leading to fluoroquinolone resistance in human Campylobacter infections. In 2016, the FDA showed that sales and distribution of all antimicrobial drugs approved for use in food-producing animals rose by 1% from 2014 to 2015 [75]. An FDA policy named FDA Guidance for Industry #213, asked that drug sponsors voluntarily remove growth promotion from the labels of all medically important antibiotics used in food animals from 2017 onwards [76]. Thankfully, major US food companies such as McDonald's and Tyson Foods have reduced and in some cases eliminated antibiotics in their products [79]. The success or failure of #213 will not be known for a number of years.

1.4 Introduction to Agent-Based Models of Bacteria

Agent-based models (ABMs) vary from differential equation (DE) models by differentiating individual agents acting within the world, instead of treating populations as homogeneous. Though it is common for DE models to capture heterogeneity in a population by partitioning it into subpopulations (compartment models), probabilistically perturbing parameters or rates of flow from one state to another (stochastic differential equations (SDEs)), or using spatial characteristics of the world to influence proportions of the population (partial differential equations (PDEs)), these techniques all maintain anonymity of the agents within the population. There are pros and cons to both types of modeling approaches. Used in conjunction, ABMs and DEs can help us better understand the dynamics of a complex system than if we used one method of modeling in isolation. There has been much discussion in the modeling community about the individual benefits of each [28, 92, 94, 99, 104] and the creation of hybrid models that incorporate both methods [27, 30, 119]. We recommend that you read through the portions of these papers that quite elegantly describe the utility of ABMs, often grounded in biological contexts.

Recently, ABMs have been used in addition to DEs to explore the spread of infectious disease throughout a population. Agents (e.g., bacteria, people, bears, sharks, coral, hurricanes, houses) act as identifiable entities that make a series of decisions, choosing between a prescribed set of options. For instance, the agents may move throughout the world by moving in a uniformly random direction, then potentially transmitting a disease to another agent with a probability that depends on the distance between the agents. Using ABMs allows us to more easily integrate spatial components and randomness into our model than formulating and analyzing PDE or SDE models.[1] Ultimately, we are interested in the behavior of the system that emerges when the agents continue to make stochastic or deterministic decisions based on their current state, which may be affected by other agents in the world or their environment. In the models of bacterial behavior studied throughout this chapter, we will often want to use an ABM to study the emergent behavior of the system. For instance, we may want to predict the proportion of a population that will be affected by an infectious disease or investigate the effects of an intervention (e.g., antibiotics).

Though we include a research project on modeling the spread of infectious disease, most of this chapter will be dedicated to creating, using, and interpreting ABMs that simulate bacterial growth, competition between bacteria within a system, and the ways that bacteria can gain resistance to an antibiotic. The ability to allow probabilistic interactions between various agents in space will help us mimic the biological mechanisms inherent in the processes to better understand the reasons for emergent behavior that we witness, predict trends we expect to see in the future, and reconcile the output of our models with the data collected through laboratory experiments.

To create our ABMs, we will be using NetLogo throughout this chapter—a common environment for programming ABMs. Created by Uri Wilensky in 1999, the platform is free to download with a plethora of texts to assist with the basics of model creation and is currently one of the standards for ABMs [120]. See [100] and [121] for thorough descriptions of ABMs, NetLogo, and their use and capabilities. Though we will spend time building a knowledge-base and familiarity with common techniques and commands with tutorial-style exercises while designing models of bacterial growth, we recommend that students who wish to pursue the challenge problems and research projects outlined later in this chapter use supplemental resources to gain additional experience and assistance with NetLogo. Working through Chapters 2, 4, and 5 of [100] would be particularly useful. If you feel comfortable creating simulations in NetLogo, you can likely move through the next section relatively quickly, focusing your attention on the biological content. If this is your first experience with programming and/or using NetLogo, we strongly recommend (at the least) completing the introductory exercises and three tutorials created by Wilensky that are available for free on the NetLogo website [120].

[1] This is especially notable for students that would like to research a complicated biological and/or social phenomenon without a mathematical background that includes advanced topics like PDEs and SDEs.

Throughout the following sections, the exercises, challenge problems, and projects are meant to prompt your own discovery of concepts associated with bacteria and ABMs through guided inquiry. There are few responses that require computations. Most will necessitate trial-and-error-type processes using the simulations you will create, followed by thoughtful reflection on why an action "worked" or "failed." For this reason, solutions are not provided—though practically all versions of the code are freely available on the QUBES (Quantitative Undergraduate Biology Education and Synthesis) website [41], with some complete code included in the Appendix.

2 Bacteria Growth

To gain some familiarity with modeling in NETLOGO, we will begin by creating a model of simple bacterial growth. However, before we begin to dive into the code, we need to distill down the microbiological background given in the previous section to the basic processes integral to bacterial growth. Bacteria reproduce asexually by a process called binary fission. Typically, bacteria divide into two identical daughter cells, containing identical genetic material. Depending on the strain of bacteria and the environmental conditions (e.g., temperature, nutrients), the rate of cell division, called generation time, can vary. In a laboratory, *E. coli*, a type of bacteria, divide every 15–20 min in nutrient-rich media. The same bacterium will divide every 12–24 h in the human intestine, where the environment is less friendly and nutrients are limited [49]. Certain disease-causing bacteria, or pathogens, have especially long generation times even when measured in the laboratory. *Mycobacterium tuberculosis*, the causative agent of TB, has a generation time of 15–20 h. Long generation times are thought to play an advantage in their capacity to cause disease or virulence [66].

Exercise 1 (Theory) The bacterium *E. coli* reproduce approximately once every 20 min in a near-optimal environment [106]. If you begin with one *E. coli* how many bacteria would you expect to see after

1. 20 min?
2. 1 h?
3. 2 h?
4. 1 day?
5. n divisions (i.e., generation times)?

If you were to plot the number of *E. coli* cells against time, what type of curve would you expect?

An initial amount of bacteria, N_0, with a generation time of t_g, will theoretically grow to

$$N(t) = N_0 2^{\frac{t}{t_g}}, \tag{1}$$

after t units of time.

Exercise 2 (Theory) Check your responses to Exercise 1 with the formula given in Eq. (1). In terms of the variables and parameters used in Eq. (1), how would you calculate the number of divisions or generation times n?

When recording and graphing data generated in a laboratory experiment, biologists plot log (logarithm) counts of the bacteria, due to the large numbers of bacteria.[2] By the time bacteria have saturated the culture, there can be as many as 8×10^8 cells per ml. You likely encountered this in the previous exercises when calculating the number of bacteria after relatively few generations, even when beginning with only one bacterium. Imagine trying to plot the numerical counts you calculated in the previous exercises using a linear scale. You would either have to use a very inaccurate scale, or you would run out of paper (or at least table space). Using a log scale makes the very large cell counts typically found easier to visualize. Since bacterial growth is exponential, the log transformation will appear linear. Additionally, a log transformation allows us to estimate the generation time of bacteria by simply finding the slope of the curve.

Exercise 3 (Theory) Show that the function $\log(N(t))$ is linear. Then, determine the generation time, t_g, by only using the slope the linear function, $\log(N(t))$.

Exercise 4 (Lab) Many of the following concepts can be demonstrated in the laboratory with the minimum of resources and equipment. They can be accomplished in a microbiology teaching lab under the supervision of your biology or microbiology instructor. We have provided examples of experiments (with commercial kits) that can reinforce your learning about the simulations, which can be found here: [108]. In addition virtual labs and online resources are supplied to aid with student learning.

Exercise 5 (Theory) Suppose we have discovered a new as of yet unidentified, deadly bacterial pathogen, *Morbum malum*. Through extremely careful laboratory experimentation, we closely monitored its growth. Beginning with a bacterial count of approximately 1.0×10^8 *M. malum* cells, we estimated approximately 4.2×10^9 cells 66 h and 6 min later by plating serial dilutions of the culture on agar and

[2]There is a debate over using log transformations with data that expresses counts when performing advanced statistical analyses [88]. However, most of the arguments against this type of transformation come from ecologists who frequently encounter zero counts.

counting the numbers of colonies that form (these are called colony-forming units or cfu). Find an estimate for the generation time (or doubling time) of *M. malum*. Compare this generation time with that of *E. coli*. Which bacteria would you expect to pose more of a threat to humanity? Why?

Now, in NETLOGO, let us begin with a single bacterium centered in the world, simulating a bacterium placed in the center of a nutrient-rich, continuous-culture solution in a flask. In continuous culture, nutrients are added and waste is removed continuously. In the Code tab, type the following in order to create a `setup` procedure:

```
to setup
  clear-all
  create-turtles 1 [set shape "circle 2" set color pink]
  reset-ticks
end
```

When this procedure is called, NETLOGO will clear any graphs and erase any variable values it once knew, create one pink circle at the center of the world, then reset the `ticks` to zero. A `tick` typically represents one iteration through the code. Notice the primitive command to generate an agent is `create-turtles`. Though we are creating bacteria, the default name for the agentset is turtles.[3] We set the shape as circles to approximate the form of the bacteria.[4] Below the `setup` procedure, create a `go` procedure. Use the `hatch` command in the code below to model binary fission. Every time you click the `go` button this procedure will "hatch" (create) a clone of each turtle in the agentset `turtles`. We will consider each tick to be the estimated generation time of the bacteria.

```
to go
  ask turtles [
    hatch 1 [right random 360 forward 1]
  ]
  tick
end
```

Do not forget to create a `setup` and `go` button in the interface area.

[3] In the NETLOGO language, all movable agents are called turtles. A parent programming language, LOGO, could be used to program robots that moved in the physical world based on the commands given. The original robots had shells that gave the appearance of a turtle. A pen could be attached to them so they would draw their path on paper. Visualizing the agents as these roaming robots may help you better understand the natural language programming conventions and syntax.

[4] If you are reading an electronic version of this text, we strongly recommend typing the sample code into the Code tab of NETLOGO instead of copying and pasting. This will help with your understanding of the flow and syntax. Also, and possibly more importantly, NETLOGO will not understand the formatted quotation marks, and you will receive an error. The naming conventions, spacing, hyphenation, and lack of capitalization used throughout the code are standard style choices for NETLOGO.

Exercise 6 (Code) The primitive procedures `right`, `random`, and `forward` take a single number as an input. Vary the numeric inputs. Explain what each of these primitives do.

Use the model to confirm your responses to Exercises 1 and 2. For now, leave the *Forever* checkbox unclicked. When the *Forever* checkbox is clicked, a circular arrow symbol appears in the bottom right-hand corner of the button. This makes the `go` button run *Forever*—meaning NetLogo will continue to loop through the code until you unclick the `go` button. By default, the *Forever* option is turned off. This allows you to view the evolution of the simulation after each tick by a manual click. You may notice that it may be difficult to count the number of bacteria even after a few ticks (and for your computer to generate such large numbers of circles). In order to better understand and analyze the results of our simulation, we will want a count of the bacteria. We can create a monitor that displays the exact number present at any given iteration.

In the Interface tab, choose *Monitor* from the drop-down menu to the right of the *Add* button. Click within the interface area to place your monitor. In the *Reporter* text box, we want to report the count of the turtles. Type `count turtles`, then click *Ok*. Now, click the `go` button a few times and note the count of bacteria.

Exercise 7 (Theory) Do the counts that appeared in the simulation match the number of bacteria you calculated in Exercises 1 and 2? You found a closed-form formula for the number of bacteria present after n generations. Can you find an iterative formula for the number of bacteria present after n generation times given the bacterial count at $n - 1$ generations?

It is difficult to understand the growth rate of the bacteria by just examining the counts at each step. A graphical display of the counts will provide more insight. In the Interface tab, choose *Plot* from the drop-down menu to the right of the *Add* button. Click within the interface area to place your graph. The default *Pen update commands* already contain the appropriate command, `plot count turtles`. A *pen* is a line created by plotting the reporter over time, in this case, the total count of bacteria over time. Following best practices, give the plot a more appropriate title (e.g., Bacteria Growth Curve) and label the axes (e.g., Number of Bacteria, Concentration of Bacteria (cfu/ml), Optical Density of the Bacteria (OD600) or Absorbance[5] versus Time[6]). Note, there are other modifications you can make to the plot, like changing the color of the pen or adding more pens. Then click *Ok*.

[5] In the laboratory setting, spectrophotometry is often used to estimate the concentration of bacteria (cells per ml or colony-forming units (cfu) per ml). In essence, light is shone through a bacterial suspension and the spectrophotometer records the amount of light that makes it all the way through to the sensor. This is called the optical density or OD. For instance an OD600 (600 refers to the wavelength of the light used) of 1.0 is roughly the same as 8×10^8 cells/ml depending on the strain. In the laboratory experiments provided in this chapter, this method is used to approximate the count of the bacteria cells present at a given time [108].

[6] Note that we are simplifying our model so that each unit on the horizontal is a generation time of the bacteria. This means the scale could have a wide range of standard time units. In the case of

Now when you run the simulation, you should see a curve, showing the total amount of bacteria over time. The slope of this curve is the growth rate. You can find the complete code up to this point at [125]: Model 1.0.

Exercise 8 (Theory) What type of curve do you see? Is this what you expected? Describe how the growth rate changes over time. Explain why this change in growth rate occurs.

Exercise 9 (Code) As previously mentioned, biologists often plot bacterial counts on a log scale. Create an additional plot in the interface area to show the growth curve on a log scale.[7] Does this curve appear as you expected? (See [125]: Model 1.1 for sample code.)[8]

At this point, our bacteria have not moved from their initial positions, with clusters grown around the original, single bacterium. Though there are non-motile bacteria (e.g., *S. aureus*), many types of bacteria are motile such as strains of *E. coli*. Though *E. coli* are known to move in a coordinated fashion (which is an entire area of study in itself),[9] we will simplify this motion to allow free and random movement of the bacteria [24].

We will take this opportunity to reorganize the model structure since we will be asking the bacteria to do two separate sub-procedures: move and divide. Our goal is to create a straightforward `go` procedure that will just ask the bacteria to move then divide. Often when designing a simulation and determining the order of actions, modelers will create flow diagrams to visually map an agent's path through a single tick. Figure 1 illustrates a simple example that we will use to build the code for Model 1.2.0.

Then, we can translate the visual map we created into the `go` procedure that will replace our previous `go` procedure:

```
to go
  ask turtles [
    move
    divide
  ]
  tick
end
```

E. coli, one tick could represent 20 min. However, if simulating *M. tuberculosis* growth, the same change in the horizontal axis could represent 20 h.

[7]The NetLogo Dictionary found on the NetLogo website is extremely useful for browsing available primitive commands and their necessary syntax. In this case, you may want to check the NetLogo Dictionary for help with the `log` primitive.

[8]It is good practice to save a model with a new name whenever you make a substantial addition or change in the code. Also, make comments with text following a semicolon or two in the code to help others understand what each line does. Often this will even be useful to remind yourself what you were thinking when you wrote the code.

[9]Though not the focus of this chapter, cell motility is a very interesting topic of study. The NetLogo Model Library contains some models that explore this (e.g., Bacteria Food Hunt, Bacteria Hunt Speeds). You could create a research project on this topic by performing a literature search and designing a model that simulates the current understanding of the way bacteria move on and in different media.

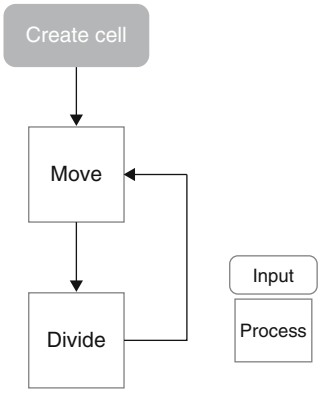

Fig. 1 This flow diagram shows a visual plan for Model 1.2.0. The initialization provided in the `setup` procedure to create one bacteria cell is the starting point of the diagram. Then the cell will move in a way prescribed by the `move` sub-procedure we will create. Then, the cell will divide in a way prescribed by the `divide` sub-procedure we will create

We must write a procedure for each of those actions. We have already written the `divide` procedure; we just wrote it straight into the `go` procedure (`hatch 1 [right random 360 forward 1]`). We can copy and paste that into its own procedure like so:

```
to divide
  hatch 1 [right random 360 forward 1]
end
```

Note, we called the `divide` procedure within the turtle context in the `go` procedure (`ask turtles [...]`). Therefore, we do not need to "ask" the turtles again in the `divide` procedure. Always be mindful of the context of each procedure. Who is being asked to act? Who is asking?

Now, we will insert a `move` procedure, giving the cells the same directions that we gave to the "hatched" cells:

```
to move
  right random 360
  forward 1
end
```

At this point, we could simplify the code even more and instruct the daughter cell that was "hatched" to follow the `move` procedure; however, we may want the flexibility to change the way that the bacteria move in further additions and revisions of this model. If you are interested in building a model that emulates the observed movement of particular motile bacteria, you will certainly need to enhance the `move` procedure. (See [125]: Model 1.2.0 and the Appendix for sample code.)

Exercise 10 (Theory) If we start with the same amount of bacteria in ten different nutrient-rich flasks, would you expect to count the exact same number of bacteria after 1 h in each of the flasks? Why?

Example 10 is referring to deterministic versus probabilistic or stochastic system dynamics. If we repeat a process over and over again, are we sure to get the same result (deterministic) or will there be some (or possibly a lot of) variability in the result (probabilistic)? For instance, our current model uses both deterministic and probabilistic processes. The number of cells exactly doubles with each click; this is deterministic. Each cell moves forward one step with each click (deterministic), but in a uniformly random direction (probabilistic). Though we will always create the same number of cells after n clicks, the cells will be located in different places because of the probabilistic `move` procedure.

Exercise 11 (Theory) In reality, would you expect to see bacteria to continue growing in this manner? Why?

One of the many reasons that you will witness variability in the growth (and not just placement) of bacteria in laboratory experiments (and in nature) is that bacterial cells die naturally, like all living organisms. Since bacterial growth rates are found from empirical counts seen in the lab or in patients, the generation times used in the model already account for cell death.[10] In future models, we will consider many other reasons for why the death rate may rise (or the growth rate will decrease), as well as other environmental conditions that lead to variability in bacterial growth.

Challenge Problem 1 (Code) Update the flow diagram and code to add a probabilistic natural death rate into Model 1.2.0. There are multiple ways this could be done. A simplified version of natural death could be coded by asking each bacterium at every tick to roll a (non-standard, many-sided) die to determine if they will die. In other words, you would be asking each bacterium at every tick to die with some probability p, where p is likely quite small. It is important to consider the order of the sub-procedures. For instance, what is the difference in the effect if you allow bacteria to die before versus after they reproduce? You will use the built-in `random` and `die` commands in NetLogo. (See [125]: Model 1.2.1 for sample code.)

When growing in a flask or on an agar plate in the laboratory, bacteria do not demonstrate unrestricted exponential growth *ad infinitum*. A flask or an agar plate would represent a closed system. In a closed system, the growth of an organism is limited by the available resources and many other possible factors. This is comforting, as otherwise the world would have been overrun by *E. coli* and many other species of bacteria by now.

Exercise 12 (Theory) Consider some environmental conditions that might lead to a reduction of the growth rate of bacteria. Why do you think these conditions would lead to a reduction in growth rate?

[10] We have included a project on quantifying bacteria death at the end of the chapter. The idea uses genetic sequencing techniques used more commonly when studying viruses, but could be applied to estimate changes in bacterial death rates to identify genetic mutations that affect fitness [124].

2.1 Bacterial Growth on Agar Plates

Let us assume the spatial aspects of the world is a limitation in the bacteria's growth. Suppose the bacteria is growing on a two-dimensional agar plate and that it is non-motile. If space is not available for the bacteria to divide, then the bacteria will not divide. We can model this in NETLOGO by using an if-statement. We check the bacteria's eight closest neighboring patches to see if any are empty. If so, we will allow the bacteria to divide and occupy one of the available neighboring patches. Model 1.1 simulated unrestricted growth of non-motile bacteria; alter the `go` procedure of that model in following way:

```
to go
  ask turtles [
    if any? neighbors with [count turtles-here = 0] [
      hatch 1 [
        move-to one-of neighbors with [count turtles-here = 0]
      ]
    ]
  ]
  tick
end
```

Now, the limitation on growth is the total number of patches in the simulated plate. No longer will each bacterium divide at each tick. Observing the plot, notice the growth now appears logistic. Instead of unlimited growth, we now see the total bacteria count not only approaching but also achieving a carrying capacity. Figure 2 shows a flow diagram of the potential paths for each bacteria cell for the code as it is written.[11] (See [125]: Model 2.0.0 for sample code.)

Exercise 13 (Theory) Examining the graph of the log of the counts of bacteria present at each generation time, why is the curve no longer linear?

Exercise 14 (Theory) Since the time it takes for each cell to divide now varies based on its environment, would you expect the average generation time to be higher or lower than when there were no spatial restrictions?

Exercise 15 (Theory) If you modify the necessary condition for cell division in Model 2.0.0 to be `count turtles-here <= 1`, how do you think the model output would change? Try these modifications in the code, making sure to change both instances of the logical expression. Were you right? Compare the carrying capacity to that of the previous model. Right-click on a patch. What do you notice in the menu? In terms of the bacterial growth on a plate, what is the difference in how the bacteria will form?

[11] If we were concerned about computational power, we may want to improve the efficiency of this code. In its current form, the loop that continually asks bacteria that had already failed the unoccupied-neighbors test is in vain; there is no mechanism in the current code to transform occupied patches into unoccupied patches. We always want to be mindful of efficiency by eliminating redundancy in the code and/or using structures that minimize computation time.

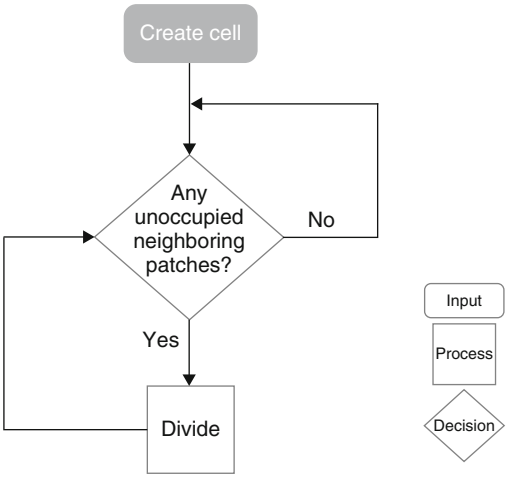

Fig. 2 This flow diagram shows a visual plan for Model 2.0.0. The initialization provided in the `setup` procedure to create one bacteria cell is the starting point of the diagram. Next, the cell must determine if it has any unoccupied neighboring patches. If so, the cell divides, and the newly formed daughter cell is placed in one of the unoccupied neighboring patches. If not, the cell will continue to look for open neighboring patches

Challenge Problem 2 (Code) Even on plates, some bacteria (like certain strains of *E. coli*) are motile on the thin film of fluid spanning the agar surface [34, 40, 60, 112]. Though they still express coordinated behavior (swimming, swarming, twitching, etc.) in their motion on an agar plate, we will simplify this to add movement in a uniformly random way like we used when modeling movement in a fluid. Create two different models that introduce movement on a plate. One can be created by inserting the move procedure from Model 1.2 into Model 2.0.0 ([125]: Model 2.0.1). The other method could restrict movement in the same way that we restricted the cell division, by only allowing a cell to divide if there is a neighboring patch that contains no bacteria cells ([125]: Model 2.0.2). Describe the difference in the bacterial growth between these two models. Why does the first model appear to grow without bound, though with a significantly reduced growth rate?

2.2 *Effects of Energy Source Availability on Bacterial Growth on an Agar Plate*

In the previous exercises, you explored the theoretical idea of the vertical, or three-dimensional, growth of bacteria on an agar plate by allowing multiple cells to exist on the same patch. This simulates stacked bacteria growing out from the plate. Experimental results combined with modeling have shown that the growth rate of bacteria is almost identical whether they are grown in broth or on agar with similar nutrients [45]. This leads us to posit that there are other environmental factors that cause bacteria growth to subside before spatial constraints could possibly cause a statistically significant effect.

You may have considered the absence of an energy source as a condition likely to reduce bacteria growth. Indeed, bacteria cannot survive forever on an unmodified surface; without food from which to obtain energy, they would die. Let us modify

our model of bacteria growth on a plate, Model 2.0.0, and introduce an initial food source (e.g., sugar, in this case, glucose) that will be consumed by the bacteria over time. First, we will add to the code a patch variable called sugar. At the very top of the code in Model 2.0.0 type,

```
patches-own [sugar]
```

Then, in the `setup` procedure, we need to specify the initial amount of sugar on each patch.

```
ask patches [set sugar 50]
```

We use 50 units here in order for the bacteria to grow to capacity before the food source becomes limited. At every tick, we will ask each bacteria to consume one unit of sugar from the patch they are on. Let us create a `consume` procedure in the following way:

```
to consume
  if [sugar] of patch-here = 0 [die]
  ask patch-here [
    if sugar > 0 [
      set sugar sugar - 1
    ]
  ]
end
```

This procedure first checks if the sugar on the patch is depleted. If so, the bacterium dies. In this case, the bacterium no longer exists to ask... anything. Therefore, the `consume` procedure would be exited if there is no sugar on the patch. However, if there is sugar left on the patch, the bacterium does not die and continues to consume one unit of sugar (simulated by the incremental decrease in the value of the sugar variable). Do not forget to add the new `consume` procedure into the `go` procedure, like so:

```
to go
  ask turtles [
    consume
    if any? neighbors with [count turtles-here = 0] [
      hatch 1 [
        move-to one-of neighbors with [count turtles-here = 0]
      ]
    ]
  ]
  tick
end
```

We want to ask the bacteria to consume the sugar prior to dividing, as energy is required for binary fission. Once we instruct the bacteria to consume the sugar, the simulation will produce a logistic growth curve with a relatively steep decay rate beginning after 52 iterations until all of the bacteria die. (See [125]: Model 2.1 for sample code.)

Exercise 16 (Code) When you run the simulation, the graph of the log growth curve stops plotting and the title turns red. Investigate this. What is the error? Edit the `go` procedure to force the simulation to terminate using the `stop` primitive command to correct the error, while also eliminating unnecessary iterations. You may find an alternative method than the solution provided in the sample code for Model 2.2 in the Appendix and in [125].

Exercise 17 (Lab + Code) Bacterial growth on different substrates supplemented with different nutrients can be easily demonstrated in the laboratory. We have provided ideas for experiments where students could use agar or broth, or vary the conditions including the pH, the sugars incorporated into the media, or vary the oxygenation [108]. Students would prepare serial dilutions of bacteria and plate for counts. The counted values could be logarithmically transformed and the log values graphed against units of time.

After collecting the data, vary the parameters in your model to best mimic the trends you witnessed in the bacteria counts. Think about adjusting numeric values like the initial amount of sugar, the maximum amount of sugar a single bacteria consumes in one tick, or the restrictions on how closely packed the bacteria can be. Devise methods for determining what makes your simulation best reflect the data. Additionally, as you learn more about environmental effects on bacteria growth, you can return to this exercise or the future lab protocols provided to continue to improve the replication of the trends you discovered by collecting data in the lab.

2.3 *Effects of Energy Source Availability on Bacterial Growth in a Flask*

Now, in a three-dimensional flask, we would not expect spatial restrictions to affect bacterial growth as much as the availability of an energy source. Additionally, the food would be relatively uniformly dispersed throughout the solution through mixing. We will return to our simulation that considers motile bacteria growing unrestricted by space in a flask (Model 1.2.0), but now we will add a restriction that they must have enough energy to perform binary fission and to generally survive. To the two sub-procedures we already included (`move` and `divide`), let us add two new sub-procedures to execute within the `go` procedure: `consume` and `expire`.[12] In Model 2.2, we added a `consume` procedure to ask the bacteria

[12] It seems that the obvious name for this sub-procedure would be "die." However, as you previously saw, `die` is a primitive command hard-coded into NETLOGO that performs the opposite action as `hatch`—`hatch` creates a turtle, and `die` removes a turtle.

to consume spatially fixed sugar on an agar plate. We will use a similar idea for the `consume` procedure in a flask, while keeping the idea in mind that the sugar will be well-mixed in the solution. We will begin by declaring sugar as a global variable (`globals [sugar]`). We will also need the bacteria to have an energy variable (`turtles-own [energy]`) that will determine whether the bacteria can divide or if it will expire. Both of those declarations will need to be added to the top of the code.

Exercise 18 (Code) In this new model, we will want to keep track of the amount of sugar that remains in the system as it will be depleted over time and not replenished (like agar in a plate or broth in a flask). Create a monitor in the interface area that reports the amount of sugar that has not yet been consumed. Note, sugar is a variable, not an agentset. The syntax is slightly different than the monitor that shows the number of bacteria.

In the `consume` procedure, we will want each bacterium to consume a sugar unit if it is available. This means the sugar will get depleted and the energy of the bacterium will increase. However, if the sugar has been completely consumed, the bacterium's energy will remain the same. We could use the following code:

```
to consume
  if sugar > 0 [
    set sugar sugar - 1
    set energy energy + 2
  ]
end
```

Now, we will alter the `move` procedure written in Model 1.2.0 to use energy:

```
to move
  right random 360
  forward 1
  set energy energy - 1
end
```

So for every movement, the bacteria have their energy reduced by one unit. However, while sugar is available, the bacteria will be steadily increasing their energy levels at a rate of one unit per tick.

The `divide` sub-procedure introduced in Model 1.2.0 will be transformed into a conditional process. For the purposes of our model, we will assume that in order for bacteria to perform binary fission, they require a sufficient amount of energy. Then, when the division occurs and the two daughter cells remain, they will each have half of the original cell's energy. Though this may not be the precise method of energy redistribution, the even split will be a reasonable proxy. Therefore, we could use the following code:

```
to divide
  if energy >= 20 [
    set energy energy / 2
    hatch 1 [right random 360 forward 1]
  ]
end
```

Note that the hatched bacterium is a clone of the original, and so has the same energy level. Also, we have semi-arbitrarily set the energy threshold for binary fission to be 20 energy units. This parameter was chosen in order to approximately reproduce the growth patterns we see in the scientific studies that plot bacteria amounts or concentrations over time. See [13, 45] for graphic and verbal explanations examples of slightly more sophisticated curve-fitting techniques with experimental bacteria growth data. Although the authors of these papers are using DE models, the basic ideas are the same—use an underlying set of relationships and behaviors, then find relative parameter values that best imitate the trends you see in the data.

The bacteria will die when their energy levels are completely depleted, and so we can create a simple expire procedure in the following way:

```
to expire
  if energy = 0 [die]
end
```

Exercise 19 (Code) The order that the sub-procedures are called within the `go` procedure matters. What order makes sense to you? Change the order of the sub-procedures in the code. Do you notice any changes in the emergent behavior? Repeat this multiple times.

In this model, we will also begin with a larger number of randomly placed initial bacteria to simulate a well-mixed solution. The energy level of each bacterium is determined by a uniform distribution. Therefore we will alter the setup procedure in this way:

```
to setup
  clear-all
  create-turtles 25 [
    set shape "circle 2"
    set color pink
    set energy random 20
    setxy random-xcor random-ycor
  ]
  set sugar 100000
  reset-ticks
end
```

The total amount of sugar is set to 100,000 units in order to visualize the exponential growth, a plateau, then a sharp decay. The simulation will need to run for approximately 100 ticks to witness the growth and decay. Just as we observed in Model 2.1 (bacteria growth on an agar plate), the log plot turns red when all of the bacteria have died. Therefore, we may fix this error in the same way by using the `stop` command.

Exercise 20 (Code) If you do not wish to click the go button 100 times, create another go button that runs the simulation until you unclick the button using the *Forever* option. Note, in order to witness the growth and decay at a visually processable speed, you may use the slider near the top of the Interface tab to slow down the tick rate. (See [125]: Model 3.0.0 and the Appendix for sample code.)

Exercise 21 (Code) What do you expect will happen in the simulation if the amount of energy required for the bacteria to divide is decreased? Increased? Try this in the model. Were you correct? What changed in the simulation output?

Challenge Problem 3 (Code) Instead of following the count of sugar by observing the monitor, use the color of the world to indicate the amount of sugar that remains in the solution. I suggest using the `pcolor` and `scale-color` primitive commands. (See [125]: Model 3.0.1 for sample code.)

Challenge Problem 4 (Code) Return to Model 3.0 once more to incorporate a spatial component of the nutrients in the way we used in the agar plate example in Model 2.2. Do this by attaching quantities of sugar to the patches and only allowing the bacteria to consume the sugar if it is on a patch that has remaining sugar. (See [125]: Model 3.0.2 for example code if you get stuck.)

You may notice in Model 3.0.0 the plateau signifying the stationary phase is quite abrupt and crudely models logistic growth. If you think about the nutrients available in the flask, there would certainly be a spatial component; once a substantial portion of the sugar has been broken down, not all bacteria would be in the proximity of an energy source. Let us consider a way to simulate a spatially dependent food source by using breeds in NETLOGO. Breeds allow us to designate classes of agents that can have their own variables and actions. We will alter Model 3.0.0 that introduced the four main sub-procedures: consume, move, divide, expire. The bacteria will be a class of agents (breed) that are required to be in the proximity of a sugar (another breed) in order to consume it and gain energy for binary fission. First, we must define our breeds at the top of the code:

```
breed [sugars sugar]
breed [bacteria bacterium]
```

Note, the `breed` primitive requires both a plural and singular form of the agentset. Instead of designating the energy variable to all agents by using `turtles-own`, we can restrict the assignment of energy to only the bacteria using `bacteria-own`. Essentially, we now use the plural name we gave to the agentset anywhere we would have previously used `turtles`. The singular form is still reserved for addressing a particular agent. Moreover, if we would like to address all agents (in this case bacteria and sugar), we would still use the entire agentset of `turtles`. Now, we must make some additions and slight alterations to the `setup` procedure. We must populate the solution with sugar, so we will add:

```
create-sugars 2000 [
    set shape "dot"
    set color white
    setxy random-xcor random-ycor
  ]
```

This creates 2000 randomly placed small white circles, representing sugar. Notice we use `create-sugars` instead of `create-turtles` since we have two distinct breeds. The initial amount of 2000 sugars was chosen in order to model

all phases of growth and decay. Next, we need to modify the bacteria initialization to designate the agents as bacteria:

```
create-bacteria 15 [
    set shape "circle 2"
    set color pink
    set energy random 21
    setxy random-xcor random-ycor
  ]
```

Again, the numbers were chosen to produce an accurate simulation. In the laboratory, bacteria growing in liquid are rotated quickly so as to agitate the contents, ensuring mixing of nutrients, the cells themselves and good aeration. For our simulation, we will assume the flask is shaken regularly to ensure the sugars are mixing evenly in the solution. Thus, our simulated sugars (and bacteria) will move randomly within the world. We add the movement of the sugar to the `go` procedure in the following way:

```
ask sugars [
    right random 360
    forward 5
  ]
```

A forward movement of 5 ensures the remaining sugars are well-mixed and not remaining far away from the bacteria clusters. In the `move` procedure for the bacteria, increase the forward movement to 5 as well, as it seems as if the mixing would have the same effect on the positional change of the sugar and bacteria.

The `divide` and `expire` sub-procedures for the bacteria will remain the same; however, we must modify the `consume` procedure. Just as we considered the spatial proximity of the bacteria and sugar in Challenge Problem 4, we will use a similar concept here. The bacteria can only consume the sugar if they are adjacent to it. So, we will use the built-in `in-radius` command to ask the bacteria to look with a certain range around themselves and consume a sugar if they find one. If there are no sugars within the designated area, the bacteria lose energy.

```
to consume
  if any? sugars in-radius 2 [
    ask one-of sugars in-radius 2 [die]
    set energy energy + 15
  ]
end
```

Note, we have increased the energy gain from a sugar unit to 15, enough for the bacteria to divide after consuming two sugars within a few ticks. This change simulates conditions where bacteria can easily and quickly perform binary fission. (See [125]: Model 4.0 for sample code.)

Exercise 22 (Code) Click the `go` button a few times and observe the growth curves. Why do they appear flat? And, why is the sugar monitor reporting an error (indicated by the red font color)? Compare to the number of bacteria. Does this make sense? Edit the *Pen update commands* within the plots in order to plot the

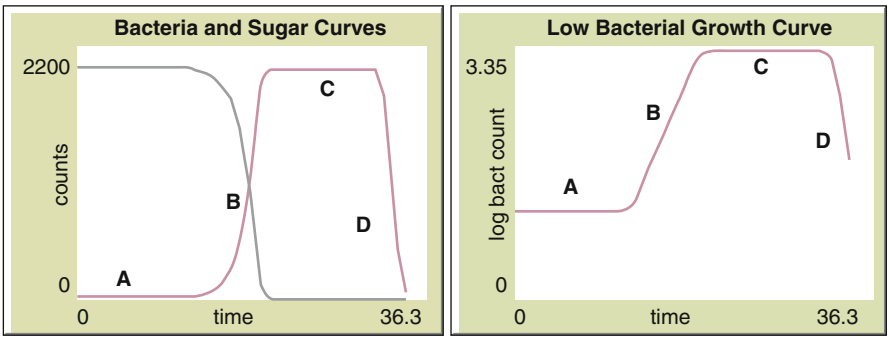

Fig. 3 The labeled curves show all four phases of bacterial growth: lag (A), log (B), stationary (C), and death (D). The increasing then decreasing bacterial counts (left) are displayed in pink and then shown on a log scale (right). The additional decreasing gray curve (left) displays the amount of sugar remaining at each time step. The plots were generated with Model 4.3 found in [125]

actual bacterial growth curves. Then add an additional pen to the plot to display the amount of sugar that remains. Modify the monitors for bacteria and sugar so that they show the proper totals. (See [125]: Model 4.1 for sample code.)

Exercise 23 (Code) Run the simulation for 50 ticks. What is not happening? Inspect a bacterium by right-clicking on it, choosing one of the bacteria on the patch, and clicking *Inspect* from the menu shown. Carefully consider the values of the variables associated with the bacterium. Find the logical error in the code that is not producing the desired results. Then, alter the code in order to fix the simulation. (See [125]: Model 4.2 for sample code.)

Model 4.2 simulates three of the four phases of bacterial growth: the log (or exponential) phase, the stationary phase, and the death phase, shown in Fig. 3 by B, C, and D, respectively. The remaining phase, indicated by A, occurs at the beginning of the growth process, called the lag phase. This is when the bacteria have just been placed in the growth medium and are preparing for the necessary processes that must take place in order to consume the energy source. The plots shown in Fig. 3 are copies of simulated bacteria growth trajectories that were generated in NETLOGO.[13] Though the resolution of the data visualizations is lacking, these images are shown here to validate your own output (with some variation, given the variability in the simulation).[14]

[13]Though the horizontal axis is labeled as time, recall, the time unit used in this simulation is arbitrary. The purpose of building this model was to replicate the phases of growth of an unspecified bacteria in a closed system; we did not try to use parameter values that would approximate growth with empirical cell counts or concentrations of a particular bacteria over time. Adjusting the generation time to fit data you collected through a laboratory experiment or using a known generation time to observe the emergent behavior of the system are components of many of the proposed research projects later in the chapter.

[14]The letter labels in Fig. 3 were added in post-production. You should not expect to see those when you create your plots in NETLOGO.

Exercise 24 (Code) Modify your model to include a lag phase. Note, even when the bacteria are adjusting to their new environment in the flask, they will continue to move and use energy. (See [125]: Model 4.3 and the Appendix for sample code.)

Not only does the depletion of energy sources cause bacterial decay, but paradoxically, the consumption of an energy source can contribute to bacteria death as well. For instance, when *E. coli* consume glucose , they are fermenting the glucose in a process called *mixed-acid fermentation*. This process results in the production of acetate as well as other acids and gas (carbon dioxide) which acidifies the medium, rendering the environment sub-optimal for the *E. coli* [16, 96]. Note, if the bacteria were inhabiting a continuous-culture solution, the acidic waste would be removed.

Let us add this unfortunate, yet natural, effect of consumption in a closed system into our model. We will begin by including a new global variable, acidity, defined at the very top of the code with `globals [acidity]`. Whenever we use a global variable, it is important to initialize its value in the `setup` procedure. Since we assume the initial medium is near-optimal, we will set the acidity to be zero, `set acidity 0`. Since acid is added to the solution when sugar is consumed by the bacteria, we need to incorporate an increase in acidity (or a decrease in pH) to the `consume` procedure. After a sugar is consumed, we can simulate the acidity level rising in this way:

```
to consume
  if any? sugars in-radius 2 [
    ask one-of sugars in-radius 2 [die]
    set energy energy + 15
    set acidity acidity + 1
  ]
end
```

In order to model the adverse effect of the acidity on the *E. coli*, we will add another cause of death on top of starvation. One way to simulate the continual decay in the quality of the solution due to acidity is to transform acidity into a death rate. We will use the following code as an example:

```
to expire
  if energy <= 0 [die]
  if random-float 1 < acidity / 200000 [die]
end
```

Here, `random-float` is introduced as an alternative way to model probabilistic behavior. This chooses a uniformly random non-negative floating point number that is strictly less than 1. Then, because 200,000 is 100 times greater than the initial amount of sugar specified in the `setup` procedure—and, therefore, much greater than the maximum acidity value—this creates a monotonically increasing death rate due to consumption with a minimum of 0 and a maximum of 0.01. In other words, the acidity will begin by killing 0% of the bacteria with every tick and slowly increases to killing 1% once all the sugar has been consumed. (See [125]: Model 5.0 and the Appendix for sample code.)

In the versions of Model 1, we used one tick as a proxy for generation time. In the versions of Models 2, 3, and 4, we restricted cell division by other environmental conditions, e.g., occupation of neighboring patches, availability of an energy source. We pose a research project to assess how environmental changes affect the average generation time.

Research Project 1 We noted early in Sect. 2 that biologists have witnessed significantly different generation times for *E. coli* when observed in a laboratory versus in a human body. Use the sources provided above, along with your own literature review, to determine potential causes of the variation of generation times, e.g., availability of efficient energy sources, ability to thrive in extreme temperatures. Create multiple models that incorporate those environmental differences. Within each model, track the number of ticks between every cell division. Record those disaggregated generation times in a list for export. You may want to investigate the use of lists in the Programming Guide found in the NETLOGO User Manual hosted on the NETLOGO website [120]. You will want to analyze the simulated data by considering measures of center and spread of generation times in a given environmental setting, ultimately, reconciling the emergent behavior with data you gathered by completing one of the laboratory protocols we provided or with experimentally produced data found in the literature.

In the following steps to design the next iteration of the model, we will create a slider so the user can input a bacteria's empirically estimated generation time. This will lay a framework that will allow simulated competition between two strains of bacteria. However, we should be mindful that we are hard-coding a minimum generation time into the simulation. Therefore, the average generation time that emerges will be longer than the one specified.

At this point, we will adjust the code in order to allow for user-defined variability in generation time. We will convert ticks to time units and create an internal counter to track the time required before the cell division is completed. In making these changes, we can also make some adjustments to improve the sample code given for Exercise 24, which simulates the lag phase. First, we need to add a counter variable to the bacteria. Within the variable declaration for the bacteria breed we will now include:

```
bacteria-own [
  energy
  generation-time-counter
]
```

The `generation-time-counter` will increase by one for every tick, triggering a division when the counter has reached the specified generation time. Of course, we still need to maintain the energy requirement for cell division as well. To keep

track of each bacterium's clock, we will insert a standard counter at the end of the `consume-move-expire-divide` cycle within the `go` procedure:

```
ifelse ticks > 10
  [
    ask bacteria [
      consume
      move
      divide
      expire
      set generation-time-counter generation-time-counter + 1
    ]
  ]
  [
    ask bacteria [ move ]
  ]
```

Note, when the `ifelse` conditional statement checks if the number of ticks is greater than 10, this forces the lag phase to continue for 10 ticks (a possible solution to Exercise 24 to include a lag phase in the growth curve). In the `divide` sub-procedure, we must restrict the cell division even further than before. Not only is an energy threshold required, but enough time must have passed for the bacteria to complete binary fission. As an example, we will use the 20-min average generation time of *E. coli*, where a tick represents 1 min in the following code:

```
to divide
  if energy >= 100 and generation-time-counter >= 20 [
    set energy energy / 2
    set generation-time-counter 0
    hatch 1 [right random 360 forward 1]
  ]
end
```

Here we remember to reset the `generation-time-counter` to 0 to begin the binary fission process again. Recall, the "hatched" bacterium will have the same variable values as its parent bacterium, resulting in the `generation-time-counter` for both daughter cells to be 0.[15] Notice that the energy threshold has increased. This is due to the increase in the number of sugars that the bacteria can consume between cell divisions and for the purpose of simulating all phases of bacterial growth. We will also change the amount of energy gained from the consumption of a sugar to 5 (`set energy energy + 5`) in the `consume` procedure, in order to maintain the relatively arbitrary choice to have the maximum amount of energy gained over the generation time to be equal to the threshold needed for division.

Finally, we need to initialize the bacteria with a `generation-time-counter` value in the `setup` procedure. We will allow this value to be a uniformly random value between 0 and 20 using `set generation-time-counter random 21`

[15] It is important to recognize that in this construction, the 20-min generation time used here is a minimum generation time; this will not result in an average generation time of 20 min.

and placing this in the `setup` procedure in the creation of the bacteria. This way, all bacteria will begin at different stages of preparedness for cell division. After the lag phase is over, some of the bacteria will be able to split immediately, while others will need a bit more time. This simulates the empirically driven concept that individual bacteria will adjust to their environment at different rates. The stochasticity of the cell division times is part of what gives us variability in the bacterial counts after a fixed time has passed. Much work has been done to model with stochastic differential equations and agent-based models backed with experimental evidence on methods to measure the variability caused by multiple stochastic processes during all phases of bacterial growth [15, 47].

Because of the additional ticks that result in additional consumed sugars, we will increase the initial number of sugars. Instead of hard-coding the amount of sugar added into the model, let us create a slider to allow us to change this value in the interface without needing to alter the code. In the Interface tab, find *slider* in the drop-down menu. Then click within the interface area to place the slider. Sliders are named with the global variable that you plan to alter. Conventionally, we use a descriptive noun or phrase with hyphens between words. In this case, we will name the variable `initial-sugars`. Then, we will set the range and granularity of the slider. In this case, we will allow the slider to go from 0 to 20,000 by 1000 sugar increments. We set the default value to be the center, 10,000, since this results in an outcome that displays all phases of bacterial growth.

Now, we need to incorporate this variable into our code. This user-determined value should replace the previous hard-coded numeric value for the initial number of sugars created in the `setup` procedure. Substitute `create-sugars 2000` with `create-sugars initial-sugars` in the code. Now, we can use the slider to change this value easily. Note, we did not need to define this global variable using `globals` within the code. In fact, if you do this, you will receive an error. In allowing the initial amount of sugar to be user-defined, we must alter the `death-by-acidity` conditional statement. We will make this adjustment:

```
if random-float 1 < acidity / (500 * initial-sugars) [die]
```

By multiplying the user-determined global variable, `initial-sugars`, by 500, we are restricting the death rate even more than in the previous version, now simulating an even slower monotonic increase from 0 to 0.5%.

Finally, all that remains is to change the axis titles on the graphs to reflect the appropriate units. Time in standard units (minutes, hours, etc.) is now measured by the horizontal axis instead of a unit of time indicating a generation time.[16] You may

[16]No units have been specified for time because any time unit could be used in the model as long as we are consistent. For instance, in the case of *E. coli*, we use 20 for our `generation-time` variable. The units on this variable are now minutes, and so the x-axis would show time in minutes. If we were modeling bacteria with a longer generation time, like, *M. tuberculosis*, we may choose to still use 20 for `generation-time`. However, the x-axis would be a measure of time in hours. Alternatively, we could stick with minutes and set `generation-time` to be 1200, which would mean the x-axis units would be minutes again. You may realize that such a large value for the

also wish to remove the pen that displays the amount of sugar that was added in Model 4.1, as we did in the sample code, so you can view the bacterial growth curve in more detail. (See [125]: Model 6.0 for sample code.)

Exercise 25 (Code) Vary the parameter determining the maximum death rate due to fermentation and increased acidity of the environment.

(a) What happens to the growth phases when the parameter is decreased? Why do you think this occurs?
(b) Use the methods described in Research Project 1 to find the average generation time for various values of the parameter. Is there an effect?

Exercise 26 (Code) Vary the slider for the initial value of added sugar.

(a) What do you notice in the time it takes to execute the simulation once? Why does this occur?
(b) What do you notice in the graphical output when you run the simulation to completion? Why does this occur?
(c) Use the methods described in Research Project 1 to find the average generation time for various values of the sugar-slider. Is there an effect?

Exercise 27 (Code) Vary the amount of energy gained from consuming a sugar and the threshold needed for cell division.

(a) What changes occur in the bacterial growth curve? Inspect the bacteria. Why does this occur?
(b) Use the methods described in Research Project 1 to find the average generation time for various values of the parameter. Is there an effect?

Exercise 28 (Code) At this point our simulation allows the bacteria to continue to consume sugar and amass stores of energy. In reality, the bacteria would genetically regulate this, so we will in our model as well. Include an additional condition in the `consume` procedure to only allow bacteria to consume sugar if their energy is below a fixed threshold, say 100 units. (See [125]: Model 6.1 for sample code.)

Exercise 29 (Theory) Instead of using the simulated data to find an average generation time, how could you estimate the generation (doubling) time of the bacteria from the graphs produced? You may want to consider Eq. (2). Use this method to estimate and confirm the expected generation time of the *E. coli* example. Now devise a method to find the generation time from the log bacteria growth curve. Estimate and confirm the expected generation time with this method.

Exercise 30 (Code) Change the code to model *Lactobacillus acidophilus*, a probiotic (i.e., "good bacteria") that is part of healthy gut flora. *L. acidophilus* has an average generation time of approximately 2.86 h when in a glucose solution [67].

---

generation time would lead to a lengthier simulation, as each tick or iteration through the go procedure is one time unit, likely leading to a preference for using a specific unit of time.

Use the methods from above to support that the changes you made in the code actually reflect the new generation time.

Exercise 31 (Code) Create a slider in the interface for generation time to more easily model different strains of bacteria. (See [125]: Model 6.2 and the Appendix for the complete sample code.)

Research Project 2 DIY Lab: Let's make kombucha!

We can witness the mixed-acid fermentation first-hand with a delicious experiment you can conduct at home. Kombucha is a carbonated acidic beverage made by fermenting sweet tea with a symbiotic culture of bacteria and yeast (SCOBY). The SCOBY contains multiple strains of bacteria and yeast. You can acquire this stiff, jelly-like culture by purchasing one online, getting one from a friend who makes kombucha, or growing your own with a plain, unflavored kombucha that you purchase at a store. The simplified process of fermentation that creates the finished kombucha product is as follows: the yeast consume the sugar, which produces carbon dioxide and ethanol. The bacteria then convert the ethanol into acid [72]. Note, because of the symbiotic nature of the yeast and bacteria, the ethanol that is produced is mostly metabolized by the bacteria, causing the alcohol content of the beverage to typically be kept to under 0.5% ABV (alcohol by volume).[17] The longer the SCOBY is thriving in the sweet tea solution, the more carbonated and acidic the beverage becomes. Additionally, the acid-producing bacteria are genetically predisposed to thrive in relatively high-acid environments—unlike many bacteria, like *E. coli*, that would die in these conditions. For this project, you will combine DIY scientific experimentation with computer simulation to better understand the fermentation process and predict the outcomes of your kombucha dependent on varied parameters.

1. Create a laboratory protocol to systematically measure the amount of sugar, acid, and SCOBY present in the solution.[18] If you are doing this at home—and not in a lab—you could qualitatively measure the sugar with a sweetness scale, use litmus strips to measure the pH, and either weigh the SCOBY with a kitchen scale or estimate its volume by using a ruler to find its dimensions. Note, temperature can

[17] So, for those of you under the legal age for drinking most of the fermented beverages currently on the market, you may legally consume the beverage produced after the project is complete (unless your kombucha was produced in the lab—never consume anything that was produced/modified/brought into the lab. There could be (and probably are) nasty hitchhiking pathogens swimming around in there!).

[18] See [81] for an example of a chemistry-focused laboratory experiment that uses technical equipment to perform measurements of more variables.

affect growth rates and the fermentation process, so be sure to consider this in your data collection. For a more advanced experiment, repeat your experiment multiple times to collect more data.[19]

2. Use the data collected to find relationships between the three variables you measured. Create a simulation that mimics the fermentation process. You may want to create a few more breeds to model the additional organisms and byproducts found in and necessary for the fermentation process. You should also do additional research on the types of yeast and bacteria present in a standard SCOBY to better model their interactions. Keep in mind that the bacteria used are not affected by the rise in acidity—unlike our previous model that was made with *E. coli* in mind.
3. Use the relationships you found to determine parameters and units for your model.
4. Compare the results of your experiment with the results of your simulation to assess your model's validity. Adjust parameter values to maximize your confidence in your model's ability to predict the quantities or concentrations of the model variables over time.
5. Consider several scenarios and use your model to predict the outcomes. Then, test your predictions by making kombucha under conditions that are as close to the hypothetical scenario you chose. For instance:

    a. You are fermenting your sweet tea in an apartment in New York City over the summer, and your air conditioner breaks. Alternatively, it is the winter, and your heat gets shut off.
    b. You order a SCOBY from a company online. You make kombucha as directed, but the end product has an acidity level close to that of apple cider.
    c. Oh no! Visible mold growth has appeared on the surface of your kombucha.[20]

Exercise 32 (Lab) The ability of bacteria to ferment sugars and to change the environment can be demonstrated several ways. The simplest would be to use solid media that changes color as the bacteria grow, ferment and produce acid, the acid changes the pH, thus the agar changes color. Fermentation tubes can also be used to demonstrate the ability of bacteria to ferment (this is indicated by a color change) and to produce gas (a bubble) and pH indicator strips could be inserted (cautiously) and the pH recorded. Triple sugar iron slants can also be used to demonstrate fermentation and gas production but

(continued)

[19] Each fermentation process requires significant time (on the order of a month). You could consider running multiple experiments simultaneously to produce more data in less time. You could also vary parameters (type of tea, incubation temperature, light exposure) and see what happens.

[20] You will likely want to read through the next section on competition before tackling this scenario.

Exercise 32 (continued)
also the production of H_2S gas. All these procedures would require access to the laboratory. You can find lab protocol examples here: [108].

3 Competition Between Bacteria Strains

Now that we have a simplified model of the growth of a single strain of bacteria, let us consider a simulation of the competition between two types of bacteria that consume the same energy source. There are many ways the bacteria could vary; they could have different generation times, death rates, energy consumption rates, etc. For instance, *E. coli* and *Salmonella enteritidis* both consume glucose for energy. However, *E. coli* has a much shorter generation time than *S. enteritidis*, ∼20 min and ∼30 min, respectively. Therefore, we would expect when competing for the same resource in a closed system, *E. coli* would grow at a faster rate than *S. enteritidis* initially, while the environment is still habitable. Since there will eventually be many *E. coli* to ferment the glucose and increase the acidity of the medium, this will cause the environment to be much too acidic for both strains of bacteria in the solution.

Exercise 33 (Lab) Competition between bacteria as well as their ability to adapt and evolve is one of the most fascinating labs to perform but it takes more time than a standard lab. This may be best suited to a long-term project or independent study. Find a mentor for this [108].

Challenge Problem 5 Use the main ideas from Model 6.2 to create a simulation of two strains of bacteria competing over the same energy source. When analyzing the results of the simulation, consider changes in the initial amount of sugar, initial amount of each bacteria strain, and generation time of each strain. How do these changes affect the amount of each bacteria strain over time?

Lactic acid bacteria, like the previously introduced *Lactobacillus acidophilus*, are characterized by their production of lactic acid and their propensity for thriving in high-acid environments that most bacteria cannot tolerate. However, *L. acidophilus*' 2.86 h generation time is significantly longer than that of *E. coli*'s—only 20 min. Both strains can thrive independently in milk and consume lactose for energy [13].

Exercise 34 (Theory) Suppose there are equal numbers of *L. acidophilus* and *E. coli* in milk. Formulate a hypothesis for the trajectory of the quantity of each strain over time.

Challenge Problem 6 Create a simulation to test the hypothesis you formulated in Exercise 34.

Now, suppose we want to simulate competition in a solution containing two energy sources: glucose and lactose. Let us return to competition between *E. coli* and *S. enteritidis*. *E. coli* can metabolize glucose and lactose. However, in media containing both glucose and lactose, *E. coli* favors glucose; it will metabolize all of the glucose first, ignoring the lactose until the glucose has been consumed. The genes needed for the breakdown of lactose are being repressed by the bacteria in a tightly regulated process called catabolite repression [82]. While the bacteria are breaking down the glucose, they produce a glucose breakdown product which inhibits an important enzyme called adenylate cyclase. This enzyme is responsible for the conversion of ATP (adenosine triphosphate—the energy currency of the cell) into cAMP. When the *E. coli* have metabolized all the glucose, the breakdown product is no longer created, and adenylate cyclase is activated, which forms cAMP. When this happens, the cell relieves the repression on the genes for lactose breakdown, the genes are transcribed, the proteins translated and the breakdown of lactose as a source of energy can begin. On the other hand, *S. enteritidis* can only metabolize glucose; it is a non-lactose fermenter.

Exercise 35 (Theory) Suppose we begin a culture with an equal number of cells of *E. coli* and *S. enteritidis*. Assume they consume all energy sources at roughly the same rate. Sketch the curves that represent the bacteria count of each type over time when

1. no food source is added;
2. only glucose is added;
3. only lactose is added; and
4. glucose and lactose are added.

Challenge Problem 7 Modify the competition simulation you previously made to include glucose and lactose, making sure to abide by the metabolic and fermentation constraints outlined above. Use your simulation to confirm your expectations for the curves in the previous exercise.

Lactose usually is fermented rapidly by *Escherichia*, *Klebsiella*, and some Enterobacter species and more slowly by *Citrobacter* and some *Serratia* species. In the clinical lab, this is the mechanism used to distinguish between pathogenic and non-pathogenic Enterobacteriaceae—non-lactose fermenters are usually pathogens, e.g., Salmonella and Shigella [14]. The ability of bacteria to ferment different sugars can be demonstrated using a variety of procedures in the laboratory (See [108].).

Exercise 36 (Lab) The ability of bacteria to ferment lactose is of particular importance in clinical microbiology where non-lactose fermenters are commonly pathogenic. Students can determine the ability of bacteria to ferment this sugar by simply streaking them onto MacConkey agar [14, 108].

4 Genetic Mutations

Every time a cell divides, there is a chance—albeit, typically quite a small chance—for a genetic mutation. For instance, researchers have used whole-genome sequencing to determine that *E. coli* have a mutation rate of approximately 2.2×10^{-10} mutations per nucleotide per generation or 1×10^{-3} mutations per genome per generation [65].[21] However, this method is impractical for determining mutation rates for bacteria with larger genomes due to the computationally expensive process.

Exercise 37 (Theory) Use the equivalent rates of mutation given above to approximate the number of nucleotides in the genome of the strain of *E. coli* studied in [65].

Challenge Problem 8 Use the approximate number of mutations per genome per generation to create a simulation of *E. coli* growth with mutations. Track the total number of mutations over time and the total number of mutated bacteria. Note, when a mutated bacterium divides, it replicates the mutation.

Some genetic mutations affect the fitness of the bacteria—the bacteria's ability to survive and divide. For example, a mutation could improve the fitness of a bacterium: it could enable the bacterium to consume a new energy source; it could increase the range of temperatures in which the bacterium would thrive; it could render a chemical compound ineffective whose purpose is to kill the bacteria (more on this later!). However, genetic mutations can also decrease the fitness of a bacterium: it could render the bacteria unable to complete the binary fission process; it could prevent the bacteria from consuming a vital energy source; it could decrease the range of pH levels that the bacteria could thrive. Another possibility is that the mutation could have no effect on the fitness level of the bacterium at all.

Challenge Problem 9 Create a simulation that allows mutations to affect the fitness level of bacteria in a positive, negative, or neutral direction. Incorporate effects on survival and division.

[21] Though, it does appear that not all nucleotides are equally likely to mutate under selective pressures due to evolutionary processes [65].

Research Project 3 Create a model that allows the user to change the environment (e.g., temperature, energy source, pH level). Simulate mutations that directly affect the fitness of a bacterium to the potential changes in the environment. Run your simulation many times with many different parameter settings to identify combinations of parameter values where certain types of mutated bacteria prevail. Read about the BehaviorSpace tool in NETLOGO to assist with running this type of experiment.

Recall from the laboratory experiments and simulation exercises on bacterial growth that biologists estimate the generation time of a bacteria strain in a fixed environment by determining the number of bacteria present over multiple time steps during the log growth phase. You may remember that we ignored natural cell death (for the most part) in our growth models. This is due to the difficulty in experimentally determining the number of bacteria that have died; our laboratory procedure only records living cells as we count them on the agar. We cannot count dead cells. Because there are no bacterial corpses to count, this makes estimating a natural death rate difficult. In the study of viruses (virology), researchers often use next generation DNA sequencing to determine viral death and genetic diversity distributions [124]. Simply put, this technology allows you to determine the proportions of a given DNA sequence as a fraction of all the DNA sequences in a sample at a given time. If you can determine the number of each type of mutated virus strains that exist in every sample over time, then when a strain no longer exists in the sample, you know that (at least) all of those viruses have been lost. The same method could conceivably be used in determining bacterial strain death, particularly for bacteria with higher mutation rates.

Research Project 4 Create a model that includes (significantly) reduced genome sequence of a bacteria and include a natural death rate. Force mutations to the genome at a fixed rate. Isolate the different sequences made by the mutations and track each strain's count over time. From the data you generate, attempt to recover the natural death rate you had hard-coded into the model. Then, incorporate the effect that mutations can have on fitness levels into your model. Could you use the data you generate to identify specific portions of the genome that have particular effects on fitness? If you allow changes in the environment, could you recover more information about the mutations? Even though your findings will only be recovering information you had to specify in the simulation, how could you use this process in a laboratory setting to better understand bacterial mutations?

5 Antibiotic Intervention and Resistance

Though most bacteria inside and outside of our bodies are harmless—or even helpful, there are strains of bacteria that are detrimental to the health of a human. When inside a human body, these disease-causing, or pathogenic, bacteria are often treated with antibiotics in order to hasten the elimination of the bacteria, thereby relieving the host of undesirable symptoms more quickly than their natural immune response. As mentioned in the introduction, antibiotics are not only used to treat humans but are widely used to promote growth in food-producing animals. Though legislative efforts have been successful in decreasing the overall use of "medically important" antibiotics in food-producing animals, FDA data has shown that in 2017, 62% of farm animal antibiotics were administered via feed, and 30% in drinking water, for mass medication [10]. The widespread and non-discriminate use of antibiotics in animal populations has been implicated in the rapid increase and spread of antibiotic-resistant bacteria.

Challenge Problem 10 Modify and add an antibiotics breed to Model 6.2 to create a simulation of the effect of antibiotics on bacterial growth. Determine a typical prescribed dosing schedule of an antibiotic and incorporate this into the simulation to track the bacterial growth over time. Consider alternative dosing schedules to witness the effectiveness of antibiotics when a non-standard dosing schedule is followed.[22]

Suppose you have a urinary tract infection (UTI), and your medical doctor prescribes you Cipro (ciprofloxacin)—a broad-spectrum antibiotic, which indiscriminately kills all strains of bacteria—based on your description of your symptoms. These broad-spectrum antibiotics will kill the pathogenic bacteria, but will also eliminate "good bacteria" such as lactobacilli found in a healthy digestive tract [55]. Alternatively, if your medical doctor first tests the bacteria present in a urine sample, they could prescribe you with a narrow-spectrum antibiotic, like Primsol (trimethoprim) that would only target the pathogenic bacteria causing the unwanted symptoms of a UTI [51].

Challenge Problem 11 Simulate the effect that antibiotics have on bacteria. Consider the varied affect that broad- versus narrow-spectrum antibiotics would have on multiple strains of bacteria contained in a system.

Research Project 5 Expand the simulation from above to specifically consider the use of broad-spectrum antibiotics for a UTI, which will kill the "good" and "bad" bacteria. One of the probiotics ("good" bacteria) found

(continued)

[22] See the existing model, Bacterial Infection, located in the NetLogo Model Library for an example of this.

Research Project 5 (continued)
in the urinary tract is *Lactobacillus acidophilus*, which was introduced in a previous section. *E. coli* are a common cause of urinary tract infections. *L. acidophilus* strains secrete antibacterial substances with activity against *E. coli*, along with other bacteria and yeast [48]. *L. acidophilus* can also thrive at much lower pH levels than *E. coli*. Use the ideas of bacterial growth, competition, and antibiotic intervention to simulate the trajectory of the populations of bacteria. In your model, consider the difference between antibiotics introduced in a closed system (e.g., an agar plate in a laboratory) or in the human (or livestock) body. For instance, how would the immune system contribute to the elimination of the pathogens? Additionally, what if narrow-spectrum antibiotics were used instead of broad-spectrum?

Under laboratory conditions, some antibiotics kill the targeted bacteria (bactericidal), while others prevent the bacteria from dividing (bacteriostatic) [91]. For first line therapy of patients, bactericidal rather than bacteriostatic agents are recommended because the eradication of microorganisms serves to limit, although not completely avoid, the development of bacterial resistance [111]. Resistance has been shown to occur more rapidly with bacteriostatic agents such as tetracyclines, sulfonamides, and macrolides than it does with bactericidal agents such as beta-lactams and aminoglycosides [111].

The use of antibiotics increases the chance of creating antibiotic-resistant bacteria.[23] Some bacteria gain resistance through mutations. Recall, mutations occur at some rate per nucleotide per genome per generation. When antibiotics are present, bacteria that are resistant will survive, while the others will die. This forces the selection of this mutation in the genome.

Research Project 6 Modify the previous simulation that tracked genetic mutations and their effect on fitness to focus only on mutations that result in antibiotic resistance. Perform a literature search to find an estimate for an average rate of acquiring resistance for a particular bacteria to a particular

(continued)

[23] Antibiotics can increase the mutation rate in bacteria [68]. Antibiotics not only impose a selective challenge to all bacteria but also accelerate the rate of adaptation by magnifying the rate at which advantageous mutations arise (any type of mutations, not just for resistance). In addition, mutation rates have been shown to increase in the bacterial flora of patients treated with antibiotics, not only for the targeted bacteria [53]. Strains that have acquired antibiotic resistance mutations often have a lower growth rate and are less invasive or transmissible initially than their susceptible counterparts [126]. The fitness costs of resistance mutations can be ameliorated by secondary site mutations. These so-called compensatory mutations may restore fitness in the absence and/or presence of antimicrobials.

Research Project 6 (continued)
antibiotic. Then model the introduction of the antibiotic to the bacteria. Run the simulation many times to analyze the outcomes. How frequently is resistance gained and selected for? Consider the differences between the effect of the antibiotic in a controlled laboratory setting and in the human body. What other environmental factors may influence the rate of selection for resistance?

Tuberculosis (TB) is often treated with a drug cocktail; in other words, a collection of narrow-spectrum antibiotics that target *Mycobacterium tuberculosis*—the causative agent of TB. Drug cocktails are prescribed in order to lessen the chance of creating a resistant strain. The cocktail is often made of four bactericidal drugs: isoniazid (INH), rifampin (RIF), streptomycin (STM) or ethambutol (EMB), and pyrazinamide (PZA). The individual use of any of these drugs to treat TB has a relatively high chance to cause resistance, but the risk is significantly reduced when taken together.

Exercise 38 (Theory) Suppose an individual with sensitive TB was prescribed a cocktail of INH, PZA, RIF, and EMB. The rate at which TB gains resistance to each antibiotic in the cocktail has been previously determined—when introduced in isolation. Resistance is acquired to INH at a rate of 2.56×10^{-8} mutations per bacterium per division, PZA at a rate of 10^{-5}, RIF at a rate of 3.32×10^{-9}, and EMB at a rate of 1.0×10^{-7} [57, 110]. Assuming each mutation is independent of the other, at what rate would we expect TB to mutate to gain resistance to all four antibiotics?

Research Project 7 Using the rates given above (and your own literature review), create a simulation that supports your calculation. Use the estimate for the generation time of *M. tuberculosis* (provided earlier in the chapter) and the simulation to determine the typical length of time resistance would emerge when taking any combination of the drugs in the cocktail. Find the standard dosing schedule used for sensitive strains of TB. Use your model to simulate the dosing schedule. Alter the schedule and/or dosages to change the chances of resistance. What if you were just given one antibiotic at a time and were only prescribed the next if the strain gained resistance to the current drug? What if you used the same cocktail for a strain of TB that is already resistant to at least one of the antibiotics?

6 Spread of Antibiotic-Resistant Bacteria in Humans

Another way bacteria gain resistance is through horizontal gene transfer (HGT). This is the process where bacteria acquire DNA from similarly-designed bacteria. For instance, suppose you have acquired an innocuous strain of *E. coli* that has gained resistance to carbapenem—antibiotics that are often considered to be the last stand against bacteria. However, you have a healthy immune system that prevents this strain of *E. coli* from reaching high enough levels to make you sick. Perhaps you visit your elderly, diabetic grandfather in a nursing home, and you pick up an antibiotic-sensitive strain of *K. pneumoniae*. The *K. pneumoniae* begins to make you quite ill. You go to a walk-in clinic, and they prescribe you a broad-spectrum antibiotic. The antibiotic begins to wipe out all of the bacteria in your body, but the stress causes *K. pneumoniae* to conjugate with *E. coli* and in so doing to acquire the carbapenamase plasmid that confers resistance [54]. After a few generations, there are enough carbapenem-resistant *K. pneumoniae* to take hold. A healthy immune system may be able to respond to this attack, but you did just wipe out your entire gut flora with the broad-spectrum antibiotic. Unfortunately, the resistant *K. pneumoniae* will not respond to another antibiotic now.

Exercise 39 (Lab) There are several commercially available kits to demonstrate the ways that bacteria can exchange genetic information. Though they require access to a microbiology lab, the resources needed are minimal and the experiments are simple to perform and interpret [108].

Research Project 8 Create a simulation of HGT. Formulate a model of the intracellular mechanisms and dynamics involved in the process, then consider a human population-scale model of the implications of HGT on planetary health.

Research Project 9 In the introduction, we discussed the societal causes and implication of antibiotic resistance. There are many agent-based models created in NetLogo that address the spread of infectious disease. In the NetLogo Model Library you can find the epiDEM models, which give basic models that simulate an epidemic [122, 123]. There are also NetLogo ABMs that consider the effect of antibiotic resistance [20], the spread of vector-borne diseases [46], and the effect of vaccinations [58]. Other ABMs use

(continued)

Research Project 9 (continued)
many of the above strategies and network structures to illustrate physical and social connections between individuals [43, 70]. Other researchers are using more advanced ABM systems that can utilize immense amounts of data. For instance, many are incorporating Geographic Information Systems (GIS) data to incorporate real city-structures and census data to better understand the spread of infectious disease [93, 117]. We challenge you to create an agent-based model that considers and analyzes the social implications of antibiotic resistance that were addressed in the introduction. For instance, incarceration of people with limited healthcare and physical space, regulation of antibiotic use on livestock, use of broad- versus narrow-spectrum antibiotics, changes in the prescription of antibiotics prompted by the regulation of healthcare, and increased mobility of people and populations. Standard infectious disease models could be adapted to incorporate these social effects. A thorough analysis of the simulation could help inform policy or support further research in the area you studied.

Appendix

Model 1.2.0

```
to setup
  clear-all
  create-turtles 1 [set shape "circle 2" set color pink]
  reset-ticks
end

to go
  ask turtles [
    move
    divide
  ]
  tick
end

to move
  right random 360
  forward 1
end


to divide
  hatch 1 [right random 360 forward 1]
end
```

Model 2.2

```
patches-own [sugar]

to setup
  clear-all
  create-turtles 1 [set shape "circle 2" set color pink]
  ask patches [set sugar 50]
  reset-ticks
end

to go
  ask turtles [
    consume
    if any? neighbors with [count turtles-here = 0] [
      hatch 1 [move-to one-of neighbors with [count turtles-here
          = 0]]
    ]
  ]
  if count turtles = 0 [stop]
  tick
end

to consume
  if [sugar] of patch-here = 0 [die]
  ask patch-here [
    if sugar > 0 [
      set sugar sugar - 1
    ]
  ]
end
```

Model 3.0.0

```
globals [sugar]

turtles-own [energy]

to setup
  clear-all
  create-turtles 25 [
    set shape "circle 2"
    set color pink
    set energy random 20
    setxy random-xcor random-ycor
  ]
  set sugar 100000
  reset-ticks
end

to go
  ask turtles [
```

```
    consume
    move
    divide
    expire
  ]
  if count turtles = 0 [stop]
  tick
end

to consume
  if sugar > 0 [
    set sugar sugar - 1
    set energy energy + 2
  ]
end

to move
  right random 360
  forward 1
  set energy energy - 1
end

to divide
  if energy >= 20 [
    set energy energy / 2
    hatch 1 [right random 360 forward 1]
  ]
end

to expire
  if energy = 0 [die]
end
```

Model 4.3

```
breed [sugars sugar]
breed [bacteria bacterium]

bacteria-own [energy]

to setup
  clear-all
  create-sugars 2000 [
    set shape "dot"
    set color white
    setxy random-xcor random-ycor
  ]
  create-bacteria 15 [
    set shape "circle 2"
    set color pink
    set energy random 21
    setxy random-xcor random-ycor
  ]
  reset-ticks
```

```
end

to go
  ifelse ticks > 10
  [
    ask bacteria [
      consume
      move
      divide
      expire
    ]
  ]
  [
    ask bacteria [ move ]
  ]
  if count bacteria = 0 [stop]
  ask sugars [
    right random 360
    forward 5
  ]
  tick
end

to consume
  if any? sugars in-radius 2 [
    ask one-of sugars in-radius 2 [die]
    set energy energy + 15
  ]
end

to move
  right random 360
  forward 5
  set energy energy - 1
end

to divide
  if energy >= 20 [
    set energy energy / 2
    hatch 1 [right random 360 forward 1]
  ]
end

to expire
  if energy <= 0 [die]
end
```

Model 5.0

```
globals [acidity]

breed [sugars sugar]
breed [bacteria bacterium]
```

```
bacteria-own [energy]

to setup
  clear-all
  create-sugars 2000 [
    set shape "dot"
    set color white
    setxy random-xcor random-ycor
  ]
  create-bacteria 15 [
    set shape "circle 2"
    set color pink
    set energy random 21
    setxy random-xcor random-ycor
  ]
  set acidity 0
  reset-ticks
end

to go
  ifelse ticks > 10
  [
    ask bacteria [
      consume
      move
      divide
      expire
    ]
  ]
  [
    ask bacteria [move]
  ]
  if count bacteria = 0 [stop]
  ask sugars [
    right random 360
    forward 5
  ]
  tick
end

to consume
  if any? sugars in-radius 2 [
    ask one-of sugars in-radius 2 [die]
    set energy energy + 15
    set acidity acidity + 1
  ]
end

to move
  right random 360
  forward 5
  set energy energy - 1
end
```

```
to divide
  if energy >= 20 [
    set energy energy / 2
    hatch 1 [right random 360 forward 1]
  ]
end

to expire
  if energy <= 0 [die]
  if random-float 1 < acidity / 200000 [die]
end
```

Model 6.2

```
globals [acidity]

breed [sugars sugar]
breed [bacteria bacterium]

bacteria-own [
  energy
  generation-time-counter
]

to setup
  clear-all
  create-sugars initial-sugars [
    set shape "dot"
    set color white
    setxy random-xcor random-ycor
  ]
  create-bacteria 15 [
    set shape "circle 2"
    set color pink
    set energy random 100
    setxy random-xcor random-ycor
    set generation-time-counter random generation-time + 1
  ]
  set acidity 0
  reset-ticks
end

to go
  ifelse ticks > 10
  [
    ask bacteria [
      consume
      move
      divide
      expire
      set generation-time-counter generation-time-counter + 1
    ]
```

```
  ]
  [
    ask bacteria [move]
  ]
  if count bacteria = 0 [stop]
  ask sugars [
    right random 360
    forward 5
  ]
  tick
end

to consume
  if any? sugars in-radius 2 and energy < 100 [
    ask one-of sugars in-radius 2 [die]
    set energy energy + 5
    set acidity acidity + 1
  ]
end

to move
  right random 360
  forward 5
  set energy energy - 1
end

to divide
  if energy >= 100 and generation-time-counter >=
      generation-time [
    set energy energy / 2
    set generation-time-counter 0
    hatch 1 [right random 360 forward 1]
  ]
end

to expire
  if energy <= 0 [die]
  if random-float 1 < acidity / (500 * initial-sugars) [die]
end
```

References

1. U.S. Action to Combat Antibiotic Resistance. https://www.cdc.gov/drugresistance/federal-engagement-in-ar/index.html.
2. Ban on antibiotics as growth promoters in animal feed enters into effect. Press Release, European Commission, December 2005.
3. Antibiotic Resistance Threats in the United States, 2013. Technical report, Centers for Disease Control and Prevention, April 2013.
4. The global economic impact of anti-microbial resistance. Technical report, KPMG LLP, December 2014.
5. Multidrug-resistant Shigellosis Spreading in the United States. Press Release, Centers for Disease Control and Prevention, April 2015.

6. National action plan for combating antibiotic-resistant bacteria. Technical report, The White House, March 2015.
7. President's 2016 Budget Proposes Historic Investment to Combat Antibiotic-Resistant Bacteria to Protect Public Health. Fact Sheet, The White House, January 2015.
8. Antibacterial Agents in Clinical Development: An analysis of the antibacterial clinical development pipeline, including tuberculosis. Technical report, World Health Organization, Geneva, Switzerland, 2017.
9. Antibiotic Resistance in Nursing Homes and Day Care Centers. https://www.cdc.gov/healthcommunication/toolstemplates/entertainmented/tips/AntibioticResistance.html, September 2017.
10. 2017 Summary Report On Antimicrobials Sold or Distributed for Use in Food-Producing Animals. Technical report, Food and Drug Administration, Center for Veterinary Medicine, December 2018.
11. Nasiru Abdullahi and Kenneth Chukwuemeka Iregbu. Methicillin-Resistant *Staphylococcus aureus* in a Central Nigeria Tertiary Hospital. *Ann. Trop. Pathol.*, 9(1):6, January 2018.
12. E. P. Abraham and E. Chain. An enzyme from bacteria able to destroy penicillin. 1940. *Rev. Infect. Dis.*, 10(4):677–678, 1988 Jul-Aug.
13. P Ačai, L' Valík, A Medved'ová, and F Rosskopf. Modelling and predicting the simultaneous growth of *Escherichia coli* and lactic acid bacteria in milk. *Food Sci. Technol. Int.*, 22(6):475–484, September 2016.
14. Mary E. Allen. MacConkey Agar Plates Protocols. http://www.asmscience.org/content/education/protocol/protocol.2855, September 2005.
15. Antonio A. Alonso, Ignacio Molina, and Constantinos Theodoropoulos. Modeling Bacterial Population Growth from Stochastic Single-Cell Dynamics. *Appl Environ Microbiol*, 80(17):5241–5253, September 2014.
16. Natalie Angier. A Population That Pollutes Itself Into Extinction (and It's Not Us). *The New York Times*, June 2018.
17. Cesar A. Arias and Barbara E. Murray. The rise of the Enterococcus: Beyond vancomycin resistance. *Nat. Rev. Microbiol.*, 10(4):266–278, March 2012.
18. S. Augustine and R. A. Bonomo. Taking stock of infections and antibiotic resistance in the elderly and long-term care facilities: A survey of existing and upcoming challenges. *Eur J Microbiol Immunol (Bp)*, 1(3):190–197, September 2011.
19. Jose Luis Balcazar. Bacteriophages as Vehicles for Antibiotic Resistance Genes in the Environment. *PLoS Pathog*, 10(7), July 2014.
20. Sean L. Barnes, Daniel J. Morgan, Anthony D. Harris, Phillip C. Carling, and Kerri A. Thom. Preventing the transmission of multidrug-resistant organisms (MDROs): Modeling the relative importance of hand hygiene and environmental cleaning interventions. *Infect Control Hosp Epidemiol*, 35(9):1156–1162, September 2014.
21. John G. Bartlett, David N. Gilbert, and Brad Spellberg. Seven ways to preserve the miracle of antibiotics. *Clin. Infect. Dis.*, 56(10):1445–1450, May 2013.
22. Arnold S. Bayer, Tanja Schneider, and Hans-Georg Sahl. Mechanisms of daptomycin resistance in *Staphylococcus aureus*: Role of the cell membrane and cell wall. *Ann. N. Y. Acad. Sci.*, 1277:139–158, January 2013.
23. P M Bennett. Plasmid encoded antibiotic resistance: Acquisition and transfer of antibiotic resistance genes in bacteria. *Br J Pharmacol*, 153(Suppl 1):S347–S357, March 2008.
24. Howard C. Berg. *E. coli in Motion*. Springer Science & Business Media, January 2008.
25. Kirandeep Bhullar, Nicholas Waglechner, Andrew Pawlowski, Kalinka Koteva, Eric D. Banks, Michael D. Johnston, Hazel A. Barton, and Gerard D. Wright. Antibiotic resistance is prevalent in an isolated cave microbiome. *PLoS ONE*, 7(4):e34953, 2012.
26. Gerald Bloom, Gemma Buckland Merrett, Annie Wilkinson, Vivian Lin, and Sarah Paulin. Antimicrobial resistance and universal health coverage. *BMJ Glob Health*, 2(4), October 2017.

27. Georgiy V. Bobashev, D. Michael Goedecke, Feng Yu, and Joshua M. Epstein. A Hybrid Epidemic Model: Combining The Advantages Of Agent-Based And Equation-Based Approaches. In *2007 Winter Simulation Conference*, pages 1532–1537, Washington, DC, USA, 2007. IEEE.
28. E. Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proc. Natl. Acad. Sci.*, 99(Supplement 3):7280–7287, May 2002.
29. Abraham Borer, Lisa Saidel-Odes, Klaris Riesenberg, Seada Eskira, Nejama Peled, Ronit Nativ, Francisc Schlaeffer, and Michael Sherf. Attributable mortality rate for carbapenem-resistant *Klebsiella pneumoniae* bacteremia. *Infect Control Hosp Epidemiol*, 30(10):972–976, October 2009.
30. Andrei Borshchev and Alexei Filippov. From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools. In *Proceedings of the 22nd International Conference of the System Dynamics Society*, volume 22. Citeseer, 2004.
31. Henry F. Chambers and Frank R. Deleo. Waves of resistance: *Staphylococcus aureus* in the antibiotic era. *Nat. Rev. Microbiol.*, 7(9):629–641, September 2009.
32. Ruvani M. Chandrasekera, Emil P. Lesho, Uzo Chukwuma, James F. Cummings, and Paige E. Waterman. The State of Antimicrobial Resistance Surveillance in the Military Health System: A Review of Improvements Made in the Last 10 Years and Remaining Surveillance Gaps. *Mil. Med.*, 180(2):145–150, February 2015.
33. Sean R. Connell, Dobryan M. Tracz, Knud H. Nierhaus, and Diane E. Taylor. Ribosomal protection proteins and their mechanism of tetracycline resistance. *Antimicrob. Agents Chemother.*, 47(12):3675–3681, December 2003.
34. Matthew F. Copeland, Shane T. Flickinger, Hannah H. Tuson, and Douglas B. Weibel. Studying the Dynamics of Flagella in Multicellular Communities of *Escherichia coli* by Using Biarsenical Dyes. *Appl. Environ. Microbiol.*, 76(4):1241–1250, February 2010.
35. Sara E. Cosgrove. The relationship between antimicrobial resistance and patient outcomes: Mortality, length of hospital stay, and health care costs. *Clin. Infect. Dis.*, 42 Suppl 2:S82–89, January 2006.
36. Sara E. Cosgrove, George Sakoulas, Eli N. Perencevich, Mitchell J. Schwaber, Adolf W. Karchmer, and Yehuda Carmeli. Comparison of mortality associated with methicillin-resistant and methicillin-susceptible *Staphylococcus aureus* bacteremia: A meta-analysis. *Clin. Infect. Dis.*, 36(1):53–59, January 2003.
37. Vanessa M. D'Costa, Christine E. King, Lindsay Kalan, Mariya Morar, Wilson W. L. Sung, Carsten Schwarz, Duane Froese, Grant Zazula, Fabrice Calmels, Regis Debruyne, G. Brian Golding, Hendrik N. Poinar, and Gerard D. Wright. Antibiotic resistance is ancient. *Nature*, 477(7365):457–461, August 2011.
38. Vanessa M. D'Costa, Katherine M. McGrann, Donald W. Hughes, and Gerard D. Wright. Sampling the antibiotic resistome. *Science*, 311(5759):374–377, January 2006.
39. Carlos A. DiazGranados, Shanta M. Zimmer, Mitchel Klein, and John A. Jernigan. Comparison of mortality associated with vancomycin-resistant and vancomycin-susceptible enterococcal bloodstream infections: A meta-analysis. *Clin. Infect. Dis.*, 41(3):327–333, August 2005.
40. Willow R. DiLuzio, Linda Turner, Michael Mayer, Piotr Garstecki, Douglas B. Weibel, Howard C. Berg, and George M. Whitesides. *Escherichia coli* swim on the right-hand side. *Nature*, 435(7046):1271–1274, June 2005.
41. Sam Donovan, Carrie Diaz Eaton, Stith T. Gower, Kristin P. Jenkins, M. Drew LaMar, DorothyBelle Poli, Robert Sheehy, and Jeremy M. Wojdak. QUBES: A community focused on supporting teaching and learning in quantitative biology. *Lett. Biomath.*, 2(1):46–55, January 2015.
42. Donald M. Dumford and Marion Skalweit. Antibiotic-Resistant Infections and Treatment Challenges in the Immunocompromised Host. *Infect. Dis. Clin. North Am.*, 30(2):465–489, June 2016.
43. Joshua M Epstein and Robert Axtell. *Growing Artificial Societies: Social Science from the Bottom Up*. Brookings Institution Press, 1996.

44. N. D. Friedman, E. Temkin, and Y. Carmeli. The negative impact of antibiotic resistance. *Clinical Microbiology and Infection*, 22(5):416–422, May 2016.
45. Hiroshi Fujikawa and Satoshi Morozumi. Modeling surface growth of *Escherichia coli* on agar plates. *Appl. Environ. Microbiol.*, 71(12):7920–7926, December 2005.
46. Holly Gaff. Preliminary analysis of an agent-based model for a tick-borne disease. *Math. Biosci. Eng.*, 8(2):463–473, April 2011.
47. Míriam R. García, José A. Vázquez, Isabel G. Teixeira, and Antonio A. Alonso. Stochastic Individual-Based Modeling of Bacterial Growth and Division Using Flow Cytometry. *Front Microbiol*, 8, January 2018.
48. Ralitsa Georgieva, Lyubomira Yocheva, Lilia Tserovska, Galina Zhelezova, Nina Stefanova, Akseniya Atanasova, Antonia Danguleva, Gergana Ivanova, Nikolay Karapetkov, Nevenka Rumyan, and Elena Karaivanova. Antimicrobial activity and antibiotic susceptibility of Lactobacillus and Bifidobacterium spp. intended for use as starter and probiotic cultures. *Biotechnol Biotechnol Equip*, 29(1):84–91, January 2015.
49. Beth Gibson, Daniel J. Wilson, Edward Feil, and Adam Eyre-Walker. The distribution of bacterial doubling times in the wild. *Proc Biol Sci*, 285(1880), June 2018.
50. Zhabiz Golkar, Omar Bagasra, and Donald Gene Pace. Bacteriophage therapy: A potential solution for the antibiotic resistance crisis. *J Infect Dev Ctries*, 8(2):129–136, February 2014.
51. G. Gopal Rao and Mehool Patel. Urinary tract infection in hospitalized elderly patients in the United Kingdom: The importance of making an accurate diagnosis in the post broad-spectrum antibiotic era. *J Antimicrob Chemother*, 63(1):5–6, January 2009.
52. Michael Gross. Antibiotics in crisis. *Curr. Biol.*, 23(24):R1063–1065, December 2013.
53. I. Gustafsson. Bacteria with increased mutation frequency and antibiotic resistance are enriched in the commensal flora of patients with high antibiotic usage. *J. Antimicrob. Chemother.*, 52(4):645–650, September 2003.
54. C. A. Hardiman, R. A. Weingarten, S. Conlan, P. Khil, J. P. Dekker, A. J. Mathers, A. E. Sheppard, J. A. Segre, and K. M. Frank. Horizontal Transfer of Carbapenemase-Encoding Plasmids and Comparison with Hospital Epidemiology Data. *Antimicrob. Agents Chemother.*, 60(8):4910–4919, August 2016.
55. Peera Hemarajata and James Versalovic. Effects of probiotics on gut microbiota: Mechanisms of intestinal immunomodulation and neuromodulation. *Therap Adv Gastroenterol*, 6(1):39–51, January 2013.
56. Benjamin P. Howden, John K. Davies, Paul D. R. Johnson, Timothy P. Stinear, and M. Lindsay Grayson. Reduced vancomycin susceptibility in *Staphylococcus aureus*, including vancomycin-intermediate and heterogeneous vancomycin-intermediate strains: Resistance mechanisms, laboratory detection, and clinical implications. *Clin. Microbiol. Rev.*, 23(1):99–139, January 2010.
57. David L. Hugo. Probability Distribution of Drug-Resistant Mutants in Unselected Populations of Mycobacterium tuberculosis. *APPL MICROBIOL*, 20:5, 1970.
58. Winfried Just and Hannah Callender Highlander. Vaccination Strategies for Small Worlds. In Aaron Wootton, Valerie Peterson, and Christopher Lee, editors, *A Primer for Undergraduate Research*, pages 223–264. Springer International Publishing, Cham, 2017.
59. Kang Kang, Yueqiong Ni, Jun Li, Lejla Imamovic, Chinmoy Sarkar, Marie Danielle Kobler, Yoshitaro Heshiki, Tingting Zheng, Sarika Kumari, Jane Ching Yan Wong, Anand Archna, Cheong Wai Martin Wong, Caroline Dingle, Seth Denizen, David Michael Baker, Morten Otto Alexander Sommer, Christopher John Webster, and Gianni Panagiotou. The Environmental Exposures and Inner- and Intercity Traffic Flows of the Metro System May Contribute to the Skin Microbiome and Resistome. *Cell Rep*, 24(5):1190–1202.e5, July 2018.
60. Daniel B. Kearns. A field guide to bacterial swarming motility. *Nat Rev Microbiol*, 8(9):634–644, September 2010.
61. Georgina Kenyon. Russia's prisons fuel drug-resistant tuberculosis. *The Lancet Infectious Diseases*, 9(10):594, October 2009.
62. Zosia Kmietowicz. Few novel antibiotics in the pipeline, WHO warns. *BMJ*, 358:j4339, September 2017.

63. A. A. A. Kwaasi. MICROBIOLOGY | Classification of Microorganisms. In Benjamin Caballero, editor, *Encyclopedia of Food Sciences and Nutrition (Second Edition)*, pages 3877–3885. Academic Press, Oxford, January 2003.
64. Timothy F. Landers, Bevin Cohen, Thomas E. Wittum, and Elaine L. Larson. A review of antibiotic use in food animals: Perspective, policy, and potential. *Public Health Rep*, 127(1):4–22, 2012 Jan-Feb.
65. H. Lee, E. Popodi, H. Tang, and P. L. Foster. Rate and molecular spectrum of spontaneous mutations in the bacterium *Escherichia coli* as determined by whole-genome sequencing. *Proc. Natl. Acad. Sci.*, 109(41):E2774–E2783, October 2012.
66. Helen C. Leggett, Charlie K. Cornwallis, Angus Buckling, and Stuart A. West. Growth rate, transmission mode and virulence in human pathogens. *Philos. Trans. R. Soc. B Biol. Sci.*, 372(1719):20160094, May 2017.
67. Jennifer A. P. Liu and Nancy J. Moon. Commensalistic Interaction Between *Lactobacillus acidophilus* and *Propionibacterium shermanii*. *Appl. Environ. Microbiol.*, 44(3):715–722, September 1982.
68. Hongan Long, Samuel F. Miller, Chloe Strauss, Chaoxian Zhao, Lei Cheng, Zhiqiang Ye, Katherine Griffin, Ronald Te, Heewook Lee, Chi-Chun Chen, and Michael Lynch. Antibiotic treatment enhances the genome-wide mutation rate of target cells. *Proc. Natl. Acad. Sci.*, 113(18):E2498–E2505, May 2016.
69. Bianca Malcolm. The Rise of Methicillin-Resistant Staphylococcus aureus in U.S. Correctional Populations. *J Correct Health Care*, 17(3):254–265, July 2011.
70. Carrie A. Manore, Kyle S. Hickmann, James M. Hyman, Ivo M. Foppa, Justin K. Davis, Dawn M. Wesson, and Christopher N. Mores. A network-patch methodology for adapting agent-based models for directly transmitted disease to mosquito-borne disease. *J. Biol. Dyn.*, 9(1):52–72, January 2015.
71. Janet M. Manson, Lynn E. Hancock, and Michael S. Gilmore. Mechanism of chromosomal transfer of *Enterococcus faecalis* pathogenicity island, capsule, antimicrobial resistance, and other traits. *Proc. Natl. Acad. Sci. U.S.A.*, 107(27):12269–12274, July 2010.
72. Alan J. Marsh, Orla O'Sullivan, Colin Hill, R. Paul Ross, and Paul D. Cotter. Sequence-based analysis of the bacterial and fungal compositions of multiple kombucha (tea fungus) samples. *Food Microbiol.*, 38:171–178, April 2014.
73. Scott A. McEwen and Paula J. Fedorka-Cray. Antimicrobial Use and Resistance in Animals. *Clin Infect Dis*, 34(Supplement_3):S93–S106, June 2002.
74. Center for Veterinary Medicine. CVM Updates - FDA Annual Summary Report on Antimicrobials Sold or Distributed in 2015 for Use in Food-Producing Animals. https://www.fda.gov/AnimalVeterinary/NewsEvents/CVMUpdates/ucm534244.htm.
75. Center for Veterinary Medicine. CVM Updates - FDA Releases Annual Summary Report on Antimicrobials Sold or Distributed in 2016 for Use in Food-Producing Animals. https://www.fda.gov/AnimalVeterinary/NewsEvents/CVMUpdates/ucm588086.htm.
76. Center for Veterinary Medicine. FDA Announces Implementation of GFI #213, Outlines Continuing Efforts to Address Antimicrobial Resistance. WebContent.
77. Mark Melzer and Irene Petersen. Mortality following bacteraemic infection caused by extended spectrum beta-lactamase (ESBL) producing *E. coli* compared to non-ESBL producing *E. coli*. *J. Infect.*, 55(3):254–259, September 2007.
78. Rodrigo E. Mendes, Lalitagauri M. Deshpande, and Ronald N. Jones. Linezolid update: Stable in vitro activity following more than a decade of clinical use and summary of associated resistance mechanisms. *Drug Resist. Updat.*, 17(1-2):1–12, April 2014.
79. Zlati Meyer. Tyson Foods will eliminate antibiotics in chicken. *USA TODAY*, May 2017.
80. Carolyn Anne Michael, Dale Dominey-Howes, and Maurizio Labbate. The antimicrobial resistance crisis: Causes, consequences, and management. *Front Public Health*, 2:145, 2014.
81. Breanna Miranda, Nicole M. Lawton, Sean R. Tachibana, Natasja A. Swartz, and W. Paige Hall. Titration and HPLC Characterization of Kombucha Fermentation: A Laboratory Experiment in Food Analysis. *J. Chem. Educ.*, 93(10):1770–1775, October 2016.

82. Annik Nanchen, Alexander Schicker, Olga Revelles, and Uwe Sauer. Cyclic AMP-Dependent Catabolite Repression Is the Dominant Control Mechanism of Metabolic Fluxes under Glucose Limitation in *Escherichia coli*. *J. Bacteriol.*, 190(7):2323–2330, April 2008.
83. Carl Nathan and Otto Cars. Antibiotic Resistance — Problems, Progress, and Prospects. https://www.nejm.org/doi/10.1056/NEJMp1408040?url_ver=Z39.88-2003&rfr_id=ori%3Arid%3Acrossref.org&rfr_dat=cr_pub%3Dwww.ncbi.nlm.nih.gov, November 2014.
84. H. Nikaido. Prevention of drug access to bacterial targets: Permeability barriers and active efflux. *Science*, 264(5157):382–388, April 1994.
85. Hiroshi Nikaido. Molecular basis of bacterial outer membrane permeability revisited. *Microbiol. Mol. Biol. Rev.*, 67(4):593–656, December 2003.
86. Eric L. Nuermberger and William R. Bishai. Antibiotic Resistance in *Streptococcus pneumoniae*: What Does the Future Hold? *Clin Infect Dis*, 38(Supplement_4):S363–S371, May 2004.
87. A. Nutman, R. Glick, E. Temkin, M. Hoshen, R. Edgar, T. Braun, and Y. Carmeli. A case-control study to identify predictors of 14-day mortality following carbapenem-resistant *Acinetobacter baumannii* bacteraemia. *Clin. Microbiol. Infect.*, 20(12):O1028–1034, December 2014.
88. Robert B. O'Hara and D. Johan Kotze. Do not log-transform count data. *Methods Ecol. Evol.*, 1(2):118–122, June 2010.
89. Jim O'Neill. Antimicrobial resistance: Tackling a crisis for the health and wealth of nations. *Rev. Antimicrob. Resist*, 20:1–16, 2014.
90. World Health Organization, editor. *Antimicrobial Resistance: Global Report on Surveillance*. World Health Organization, Geneva, Switzerland, 2014. OCLC: ocn880847527.
91. G. A. Pankey and L. D. Sabath. Clinical Relevance of Bacteriostatic versus Bactericidal Mechanisms of Action in the Treatment of Gram-Positive Bacterial Infections. *Clin Infect Dis*, 38(6):864–870, March 2004.
92. H Van Dyke Parunak, Robert Savit, and Rick L Riolo. Agent-based modeling vs. equation-based modeling: A case study and users' guide. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 10–25. Springer, 1998.
93. Liliana Perez and Suzana Dragicevic. An agent-based approach for modeling dynamics of contagious disease spread. *International Journal of Health Geographics*, 8(1):50, August 2009.
94. Gael Pérez-Rodríguez, Martín Pérez-Pérez, Daniel Glez-Peña, Florentino Fdez-Riverola, Nuno F. Azevedo, and Anália Lourenço. Agent-Based Spatiotemporal Simulation of Biomolecular Systems within the Open Source MASON Framework. *BioMed Res. Int.*, 2015:1–12, 2015.
95. Margaret B. Planta. The Role of Poverty in Antimicrobial Resistance. *J Am Board Fam Med*, 20(6):533–539, January 2007.
96. P. Pletnev, I. Osterman, P. Sergiev, A. Bogdanov, and O. Dontsova. Survival guide: Escherichia coli in the stationary phase. *Acta Naturae*, 7(4):22–33, 2015.
97. Laurent Poirel, Johann D. Pitout, and Patrice Nordmann. Carbapenemases: Molecular diversity and clinical consequences. *Future Microbiol*, 2(5):501–512, October 2007.
98. Keith Poole. Efflux-mediated antimicrobial resistance. *J. Antimicrob. Chemother.*, 56(1):20–51, July 2005.
99. Hazhir Rahmandad and John Sterman. Heterogeneity and Network Structure in the Dynamics of Diffusion: Comparing Agent-Based and Differential Equation Models. *Manag. Sci.*, 54(5):998–1014, May 2008.
100. Steven F. Railsback and Volker Grimm. *Agent-Based and Individual-Based Modeling: A Practical Introduction*. Princeton University Press, Princeton, 2012.
101. Sara Reardon. Antibiotic resistance sweeping developing world. *Nat. News*, 509(7499):141, May 2014.
102. Gian Maria Rossolini, Fabio Arena, Patrizia Pecile, and Simona Pollini. Update on the antibiotic resistance crisis. *Curr Opin Pharmacol*, 18:56–60, October 2014.

103. Stefan Schwarz, Corinna Kehrenberg, Benoît Doublet, and Axel Cloeckaert. Molecular basis of bacterial resistance to chloramphenicol and florfenicol. *FEMS Microbiol. Rev.*, 28(5):519–542, November 2004.
104. Shelby M. Scott, Casey E. Middleton, and Erin N. Bodine. An Agent-Based Model of the Spatial Distribution and Density of the Santa Cruz Island Fox. In *Handbook of Statistics*, volume 40, pages 3–32. Elsevier, 2019.
105. Jessica L Semega, Kayla R Fontenot, and Melissa A Kollar. Income and Poverty in the United States: 2016. Technical report, US Census Bureau, September 2017.
106. G. Sezonov, D. Joseleau-Petit, and R. D'Ari. Escherichia coli Physiology in Luria-Bertani Broth. *J. Bacteriol.*, 189(23):8746–8749, December 2007.
107. DM Sievert, ML Boulton, G Stoltman, D Johnson, MG Stobierski, FP Downes, PA Somsel, JT Rudrik, W Brown, W Hafeez, T Lundstrom, E Flanagan, R Johnson, J Mitchell, and S Chang. *Staphylococcus aureus* Resistant to Vancomycin. *Morb. Mortal. Wkly. Rep.*, 51(26):565–567, July 2002.
108. Davida S. Smyth. Laboratory exercises on microbial growth, horizontal gene transfer, competition and evolution. *QUBES Educ. Resour.*, 2018.
109. Brad Spellberg and David N. Gilbert. The future of antibiotics and resistance: A tribute to a career of leadership by John Bartlett. *Clin. Infect. Dis.*, 59 Suppl 2:S71–75, September 2014.
110. Karolien Stoffels, Vanessa Mathys, Maryse Fauville-Dufaux, René Wintjens, and Pablo Bifani. Systematic Analysis of Pyrazinamide-Resistant Spontaneous Mutants and Clinical Isolates of *Mycobacterium tuberculosis*. *Antimicrob Agents Chemother*, 56(10):5186–5193, October 2012.
111. Charles W. Stratton. Dead Bugs Don't Mutate: Susceptibility Issues in the Emergence of Bacterial Resistance. *Emerg Infect Dis*, 9(1):10–16, January 2003.
112. Jean-Marie Swiecicki, Olesksii Sliusarenko, and Douglas B. Weibel. From swimming to swarming: *Escherichia coli* cell motility in two-dimensions. *Integr Biol (Camb)*, 5(12):1490–1494, December 2013.
113. Emily R. M. Sydnor and Trish M. Perl. Hospital epidemiology and infection control in acute-care settings. *Clin. Microbiol. Rev.*, 24(1):141–173, January 2011.
114. Birkneh Tilahun Tadesse, Elizabeth A. Ashley, Stefano Ongarello, Joshua Havumaki, Miranga Wijegoonewardena, Iveth J. González, and Sabine Dittrich. Antimicrobial resistance in Africa: A systematic review. *BMC Infect Dis*, 17, September 2017.
115. Truc T. Tran, Diana Panesso, Hongyu Gao, Jung H. Roh, Jose M. Munita, Jinnethe Reyes, Lorena Diaz, Elizabeth A. Lobos, Yousif Shamoo, Nagendra N. Mishra, Arnold S. Bayer, Barbara E. Murray, George M. Weinstock, and Cesar A. Arias. Whole-genome analysis of a daptomycin-susceptible *enterococcus faecium* strain and its daptomycin-resistant variant arising during therapy. *Antimicrob. Agents Chemother.*, 57(1):261–268, January 2013.
116. Magnus Unemo and William M. Shafer. Antimicrobial resistance in *Neisseria gonorrhoeae* in the 21st century: Past, evolution, and future. *Clin. Microbiol. Rev.*, 27(3):587–613, July 2014.
117. Srinivasan Venkatramanan, Bryan Lewis, Jiangzhuo Chen, Dave Higdon, Anil Vullikanti, and Madhav Marathe. Using data-driven agent-based models for forecasting emerging infectious diseases. *Epidemics*, 22:43–49, March 2018.
118. C. Lee Ventola. The antibiotic resistance crisis: Part 1: Causes and threats. *P T*, 40(4):277–283, April 2015.
119. Gudrun Wallentin and Christian Neuwirth. Dynamic hybrid modelling: Switching between AB and SD designs of a predator-prey model. *Ecol. Model.*, 345:165–175, February 2017.
120. Uri Wilensky. NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, 1999.
121. Uri Wilensky and William Rand. *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. The MIT Press, Cambridge, Massachusetts, 2015.
122. C Yang and Uri Wilensky. NetLogo epiDEM Basic model, 2011.
123. C Yang and Uri Wilensky. NetLogo epiDEM Travel and Control model, 2011.

124. Fengjiao Yu, Yujie Wen, Jibao Wang, Yurong Gong, Kaidi Feng, Runhua Ye, Yan Jiang, Qi Zhao, Pinliang Pan, Hao Wu, Song Duan, Bin Su, and Maofeng Qiu. The Transmission and Evolution of HIV-1 Quasispecies within One Couple: A Follow-up Study based on Next-Generation Sequencing. *Sci. Rep.*, 8(1):1404, January 2018.
125. Anne E. Yust. Sample NetLogo Code of Bacterial Growth. *QUBES Educ. Resour.*, October 2018.
126. Pia Schulz zur Wiesch, Jan Engelstädter, and Sebastian Bonhoeffer. Compensation of Fitness Costs and Reversibility of Antibiotic Resistance Mutations. *Antimicrob. Agents Chemother.*, 54(5):2085–2095, May 2010.