STOCHASTIC MODELING OF STRUCTURED MICROBIAL ENVIRONMENTS

by

ISABEL RICHTER

A THESIS

Presented to the Department of Biochemistry
and the Robert D. Clark Honors College
in partial fulfillment of the requirements for the degree of
Bachelor of Science

May 2021

An Abstract of the Thesis of

Isabel Richter for the degree of Bachelor of Science
in the Department of Biochemistry to be taken June 2021

Title: Stochastic Modeling of Structured Microbial Environments

Approved: ____________*Tristan Ursell, Ph.D.*____________________
Primary Thesis Advisor

The survival of microbes depends on their ability to acquire space and nutrients as well as compete with other groups of microbes. Relatively fit microbes should completely outcompete their weaker counterparts, but such outcomes are not commonly observed in nature. In structured environments, such as soil or the mammalian gut, the structure itself may determine which microbes dominate and which are driven to exclusion. Our goal was to create a stochastic simulation that approximated the chance nature of ecological interactions to predict dynamics and timescales on which steric structure influences microbial competition. Future work will derive more data from this model and compare them with previous models from our lab.

Acknowledgements

I would like to first and foremost thank my Primary Thesis Advisor, Assistant Professor Tristan Ursell of the Physics Department for his dedication to my success, sharing his wisdom, and pushing me to do my best work. I cannot imagine going through the thesis process and my last two years at university without his support and guidance. I would also like to thank Assistant Professor Scott Hansen and Associate Professor Chris Sinclair of the Biochemistry and Mathematics departments, respectively, for serving on my thesis committee and for helping me craft a thesis I am proud to call my own. In addition to these faculty, The Clark Honors College has allowed me to embrace innumerable new perspectives that were otherwise out of reach and encouraged me to view things holistically whenever possible. I am greatly privileged to soon be an alumnus of the CHC.

I also want to take this opportunity to thank my mother, father, and grandmother whose love is inexhaustible. They have inspired me to get to where I am today and will be tomorrow. I also want to thank JB McIntosh and his family, for giving me the opportunity to come to the University of Oregon and to achieve my dreams. Because of you, I believe that anything is possible with enough love and determination.

Table of Contents

List of Figures

Introduction

Between species and/or within an isogenic population, groups of microbes frequently compete and cooperate with other groups of microbes. These interactions influence which microbes live, which microbes die, and which mix of species will persist in a particular environment[1]. The species composition of a microbial ecosystem alters environmental nutrients and their concentrations, which can have downstream effects on other organisms, such as plants, fungi, and animals (including humans)[2]. Ecological principles suggest that mutually competing organisms that occupy the same niche cannot stably coexist, due to a mechanism known as 'competitive exclusion'[3], yet natural ecosystems routinely contain numerous competing and coexisting microbial species at similar trophic levels[4]. Multiple mechanisms have been proposed to account for this unexpected multi-species stability[4–7]. Our lab is characterizing a potentially generic mechanism that robustly maintains competitive stability[8], by examining how steric structure within an environment impacts the stable representation of species that mutually compete for the same space and resources. Such mechanisms are likely relevant because most natural environments—like soils[9], sea water[10], and the mammalian gut[11] —are not isotropic, rather they contain varying degrees and length-scales of steric structure.

In the context of our work, competition primarily occurs for critical resources such as nutrients and space, which are themselves frequently linked. In response to the challenge of securing nutrients and space, microbes have evolved methods for actively targeting and killing competing species[12]. These methods frequently take the form of one species injecting proximal cells with a toxin or secreting a toxin into the local environment[13]. Not all such secreted or injected compounds are equally potent and thus the efficacies with which one species kills another are not necessarily equal. This competitive asymmetry suggests that if two mutual competitors initially grow free from competition in a uniform environment (e.g., a petri dish), they will each increase in population until they contact the other species and form an approximately one-dimensional competition interface. From there, both species would begin to secrete their respective toxins, and the species that produces the most effective toxin (either by potency or concentration) would eventually be the single dominant competitor to the exclusion of other species. The local extinction of a weaker competitor occurs when the interspecies boundary moves to reduce the territory of a weaker species, until none of the weaker species remains. This is a spatial example of the ecological principle of competitive exclusion, wherein different species competing for the same resources cannot coexist indefinitely as one will eventually outcompete the other, often to the point where one species is locally dominant[3,14]. In natural environments, however, multiple species are observed stably competing within a single niche or closely related niches[4,15], and thus ecosystems can maintain a level of species diversity that is incommensurate with the assertion of competitive exclusion. This type of competition is thought to occur in microbial ecosystems, including soils[9], sea water[10], and the mammalian gut[11].

Our work seeks to understand how the presence of solid ‘steric’ objects—from the size of a cell or larger—distributed throughout an environment affect ecosystem dynamics and ultimately the persistence of multiple competing species, by modulating the movement and stability of competitive interfaces between species. Previous work in our lab demonstrated that the presence of structure in the environment can maintain multiple competing species even with relatively large competitive asymmetries. This stabilization results from curved interspecies boundaries between steric objects in an environment. Geometrically, curved interfaces have more of one species on the outer side of the curve than the inner, allowing a weaker species to locally outnumber a stronger species, and thereby to compensate for deficiencies in competitive potency with increased numbers at the interface of interspecies competition.

While our previous work clearly indicates the potential for steric structure to alter population dynamics, natural systems are subject to multiple sources of stochasticity, including stochastic processes that control cell death at the hands of a competitor and acquisition of open territory by surrounding microbes[16–18]. My thesis work built on these previous continuum models of ecosystem dynamics to create a computational model of microbial competition in structured environments with the key addition of species-specific stochastic processes and discretization of space roughly on the length scale of a cell. Our model accounts for varying size and spacing between steric objects to approximate a range of particle sizes and densities, for instance, like those seen in power law distributions of particle size in soils[19]. We used a stochastic model to approximate the chance nature of interactions by discretizing interactions between microbes into a stepwise decision-tree with weighted random outcomes.

Our computational model quantitatively incorporates differences in the innate growth and expansion rates of each species and incorporates variations in the killing efficacy for each species -- for instance, species A may be more effective at killing species B than B is at killing A. Our 2D simulation environment includes steric objects as zones that exclude microbes and any chemical signals akin to the impermeable surfaces / particles frequently found in natural contexts, for example particles in soil.

Unlike earlier work from our lab that examined competition in structured environments, this model will approximate the ‘chance nature’ of interactions and competitive outcomes, and more accurately approximate the discrete, individual nature of microbes. This means that when two species meet in space, on average individuals with more potent toxins will advance and kill individuals of species presenting a less potent toxin – relevant parameters will be discussed in **Results** section. However, weaker species can gain territory and kill members of the fitter species during the course of dynamical fluctuations that result from underlying stochastic processes. In our model, fixed rate parameters establish the probability over a single time step that one species kills competitor species, and subsequently propagates into empty space. These chance outcomes alter the dynamics of competition in structured environments, and ultimately the fate of the species involved.

In previous work[8], our lab used a continuum and deterministic model to examine microbial interactions in model ecosystems with many competing species and on length scales of 10,000’s of thousands of cell, without implementing the stochastic framework. This work showed that regularly spaced, steric structure in an environment could preserve many competing species on long time scales across a range of geometric and competitive parameters; an example[8] is shown in **Figure 1**. This work also found that the competitive asymmetry between species could be compensated by the curvature of interspecies boundaries, with competitively weaker species residing on the outer curvature that provides a local numerical advantage. It was also observed that the greater the competitive asymmetry, the larger the difference in relative boundary area between species that could establish a stable interface.

Expanding on these findings, new work from our lab on multispecies competition using the continuum model revealed that these same patterns hold true for multi-species competitive communities[20], as exemplified in **Figure 2**. Work from this thesis will likely be expanded in to incorporate such multi-species communities in structured environments.

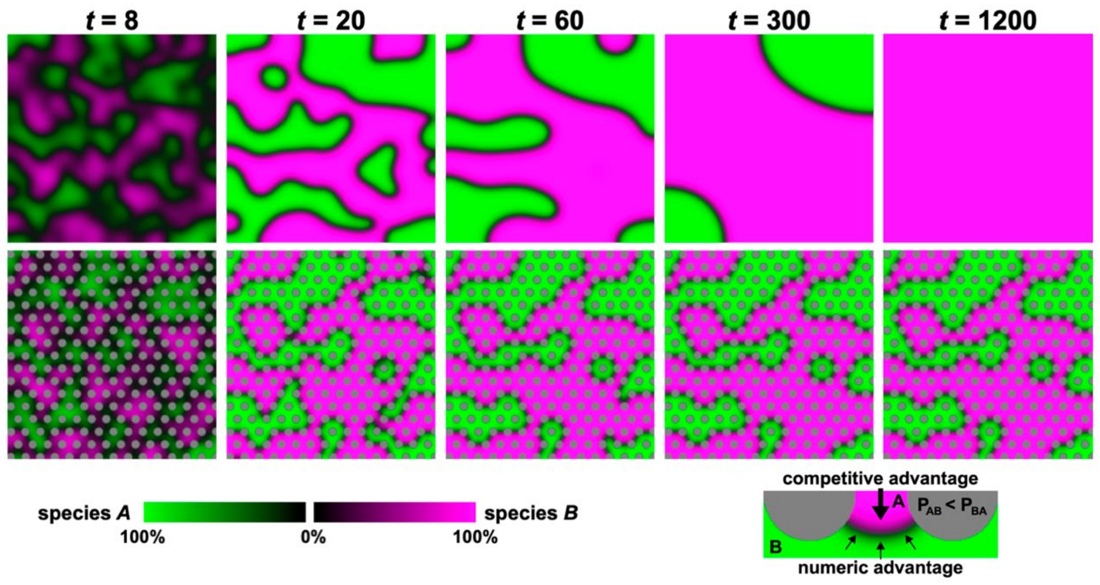

Figure 1: Structured environments halt competitive exclusion between two species. The top row is a selection of stills from competition between two species which ends in the magenta species eliminating the green species. The lower panels show competition in a structured environment that settles into a stable orientation relatively quickly and remains diverse over extremely long timescales. The diagram in the lower right shows competitive advantage and numeric advantage at a boundary between two pillars acting as equal and opposing 'forces' that keep the boundary stable.

Thus far, we have simulated hundreds of these situations, varying the placement and size of the solid objects, as well as the competition parameters that reflect toxin potency (i.e. the neighborhood size, and killing and filling rates; see Algorithm Design section). The outputs of these simulations are beginning to reveal how geometric and competitive parameters affect which species survive, how competing species are positioned in space, and how long it takes a given simulated ecosystem to reach a stable state or to become unstable--meaning a loss in biodiversity through competitive exclusion. These data will contribute to a mechanistic understanding of how relevant variations in the environment confer or hinder species stability in real-world ecosystems, and they will contribute to our future ability to engineer specific microbial ecosystems for use in industry, agriculture, and medicine.

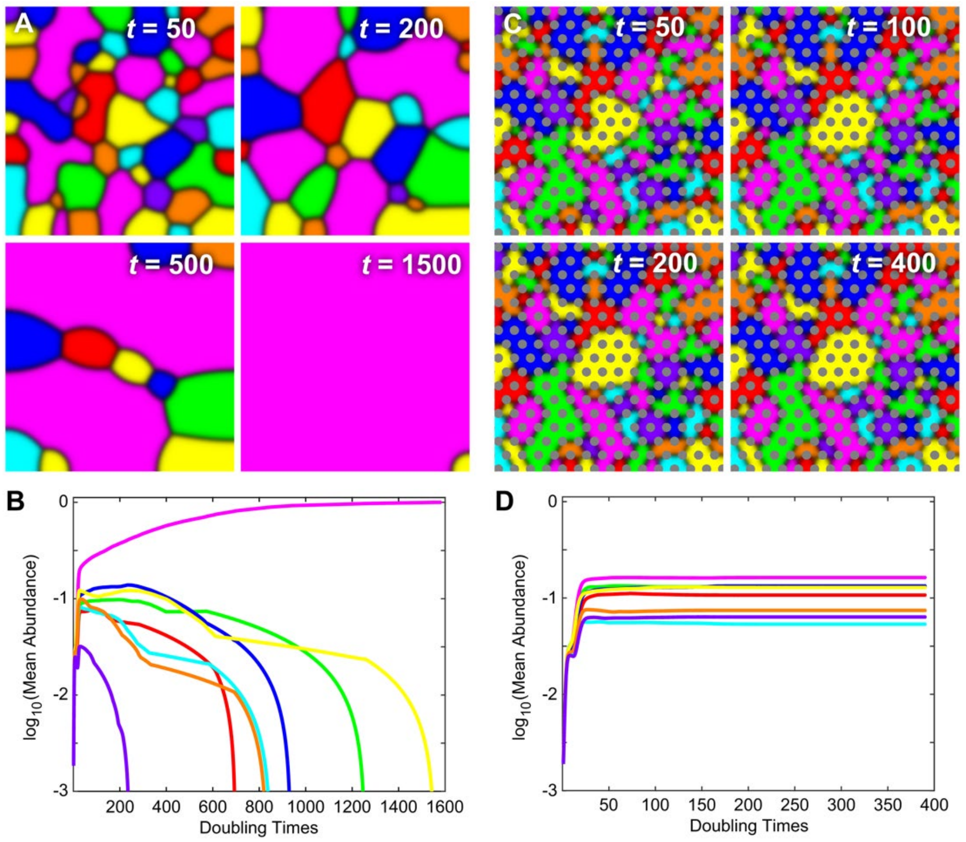

Figure 2: Structured environments halt competitive exclusion in competition between many species. Snapshots (A) and population levels (B) showing spatial competition between 8 species in an environment without steric structure – here eventually the magenta species dominates over all other species. Snapshots (C) and population levels (B) showing competition between 8 species in an environment that presents steric structure – here the model ecosystem rapidly reaches a population equilibrium in which all species are stably represented. This qualitatively different outcome is due to the present structure, and specifically the slowing effect of structure on interface movement.

Results

In the following sub-sections we discuss the design and underlying calculations of an algorithm that implements stochastic spatial competition. We explain the layout and control parameters of the simulations and display results from a subset of our first run of simulations that demonstrate stochastic fluctuations in this context. We present demonstrative examples of qualitatively distinct outcomes—not encompassed in the previous continuum work—that depend on parametric inputs and spatial stochasticity.

Algorithm Design

The design of our stochastic spatial algorithm begins with discretization of space into a square grid that can easily be represented and manipulated via matrix computations. The length-scale of a pixel corresponds to the size of a cell, though notably cells are not generally square, thus this correspondence sets a scale but should not be interpreted as a literal representation of a cell. Each point on this square grid has one of four identities (or states) and the design of the algorithm is an effort to describe the stochastic dynamics of transition between those states on each pixel. First, some pixels are immutable and do not host species nor do they contribute to the competitive dynamics, they are steric pixels whose positions construct solid objects in the space, represented computationally as not-a-number (NaN). Second, some pixels are 0, which indicates a free pixel that could, under suitable circumstances, be colonized by either species A (+1 pixel state) or species B (-1 pixel state). Lastly, colonized pixels have identities as either species A (+1 pixel state) or species B (-1 pixel state). We started with two species competition (+/- 1 pixel states), but the algorithm we designed has straightforward extensions to an arbitrary number of competing species via this state-space approach.

In our simulations, steric objects are circular (to within pixel resolution), immobile and do not change size or shape during the simulation; in other words, the number and position of NaNs in any particular simulation is fixed. All simulations had the same basic layout, as shown in **Figure 3**. Circular pillars were created with NaNs linked to the edge of the simulation box by NaN-barriers that constrained competition to occur in the space between the pillars. Each simulation was initialized with an interspecies interface exactly between the pillars; this symmetry ensured that each simulation started without a bias toward either species, regardless of their competitive fitness. The length and width of the simulated rectangular space span tens to hundreds of cells on each side, large enough that quasi-stable stochastic fluctuations do not contact the edge of the simulation box.

A second 'interaction' length scale is set by the radius of the pixel neighborhood that affects a given central pixel. Depending on the type of killing, cells can compete and influence the viability of proximal cells on varying length scales. For diffusible toxins secreted by cells[21], the effective length scale of competition may be several cells long, whereas contact-mediated competition[22,23] is restricted to one or two cell lengths away. Our model reflects these distinct biological mechanisms with a local pixel neighborhood whose size can vary across simulations; a range of isotropic neighborhoods are shown in **Fig 4**. Whichever the underlying mechanism of competition, cells of one species must be able to differentially compete with cells from another species, meaning our model must account for the possibility that one species is a stronger competitor than the other.

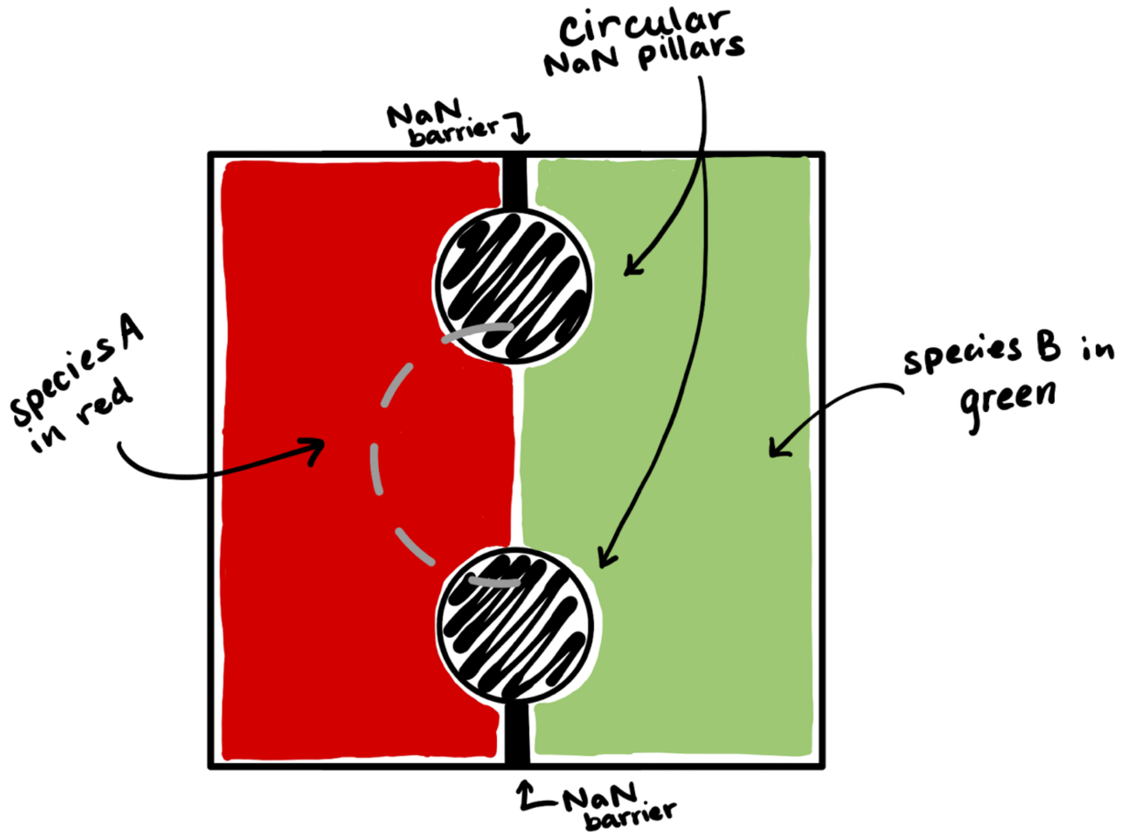

Figure 3: Basic layout of the simulation space. Each simulation is a rectangular box containing two circular NaN pillars, and corresponding NaN barriers that together restrict the competition interface to the space between the pillars. In visual display, we show steric pixels as gray, open pixels as black, species A as red, and species B as green. Each simulation is initialized with the species boundary exactly between the pillars, as shown above. This ensures that there is no bias in the boundary structure that could influence the stochastic dynamics. The center-to-center spacing of the pillars and their radius are the primary geometric, which ultimately set the height of the simulation space. The width of the simulation box is set to be large enough that a stable, fluctuating interface does not make contact with the edge of the simulation box, accomplished by setting the width to be roughly twice the distance set by the maximum curvature (gray dashed line).

Our model accounts for such differences through a pair of kinetic parameters k_{AB} and k_{BA} that characterize the rate at which an individual A pixel kills an individual B pixel, and vice versa. Further, each species has a rate at which it fills (grows into) a vacant pixel, k_{0B} and k_{0A}, respectively. Throughout this work, we set $k_{0A} = k_{0B} = 1$, which equalizes the spreading rate of each species and sets the natural timescale in the system, k_{0X}^{-1}, used as the unit of time in the current simulations; here X meaning either species.

The simulation space is structured so that competitive interactions are localized to the interfacial region between the pillars. Each pillar is specified by a radius R and a center-to-center separation D – these are the fundamental geometric parameters of the simulation. Within the model, steric pillars are represented as non-interacting not-a-number (NaN's), as are regions above and below each pillar to limit the interactions to the desired interface between the pillars. Each interaction neighborhood is characterized by a pixel radius (see **Fig. 4)**, which defines a particular neighborhood structure and corresponding number of pixels n_{hood}, excluding the central pixel. Within each neighborhood there is some number of vacant sites, some number of A's (n_A) and some number of B's (n_B). The central pixel in a neighborhood is the pixel whose identity has the potential to change. Two trivial cases cannot result in a state change of the central pixel: (i) if all of the pixels in a neighborhood (excluding the central pixel) are vacant and (ii) if the species identity of every non-vacant pixel in the neighborhood is identical, for instance, a central A pixel surrounded by some number of A's and no B's.

Outside those cases, the state of the central pixel can change according to a decision tree with probabilistic weights, as schematically described in **Figure 5**. The decision tree begins with determination of whether the central pixel is empty or occupied. If the central pixel is occupied by species X (i.e. either A or B) it will switch to a vacant site with probability

$$p_{X0} = 1 - e^{-\Delta t n_Y k_{YX}}$$

where Δt is the simulation time-step (see Coding Methods), and k_{YX} is the per-pixel rate at which Y kills X. Here we generalize the species labeling to X and Y, to indicate that there is an equivalent formula for each species (i.e. p_{A0} and p_{B0}). Note that if $n_Y = 0$ (no competitors in the neighborhood), then $p_{X0} = 0$ (no chance of state change). This probability is compared to a random number between 0 and 1, and if that number is less than p_{X0}, then the pixel changes state from occupied to vacant. This scheme implicitly assumes that the probability for neighboring X pixels to kill a central X pixel is

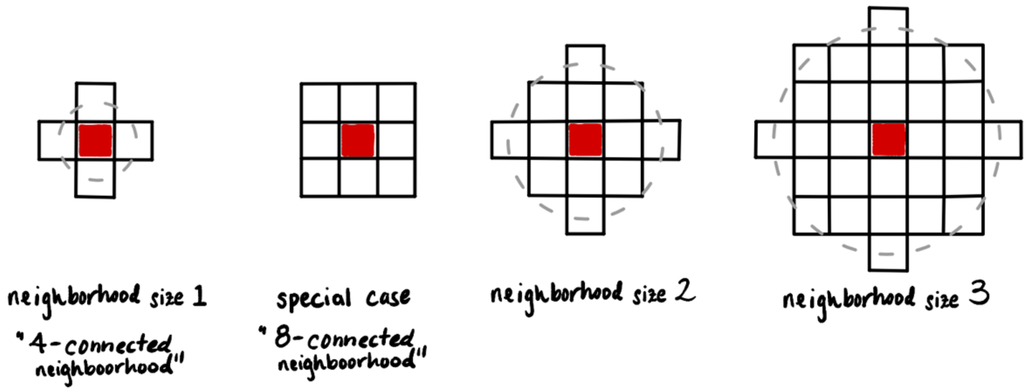

Figure 4: Examples of four different competition neighborhoods. The 4- and 8-connected neighborhoods approximate competition between proximal cells, as might be encountered in contact-mediate killing, whereas the larger 2 and 3 pixel-radius neighborhoods approximate the longer ranges of diffusible toxins.

identically zero (the 'no cannibalism' rule). Similarly, this scheme does not admit the possibility that other X's in the neighborhood have a protective effect on the central X (though this would not be difficult to incorporate). This scheme is applied to every occupied pixel that is not part of the trivial case mentioned above.

If a pixel is vacant and has occupied pixels of either species in its neighborhood, then the probability that its state changes from vacant to occupied is given by

$$p_{0X} = 1 - e^{-\Delta t(n_A k_{0A} + n_B k_{0B})}$$

and a random number between 0 and 1 determines whether this state change occurs. If the state changes to occupied then the probability that the pixel will be occupied by species A is

$$p_A = \frac{n_A k_{0A}}{n_A k_{0A} + n_B k_{0B}}$$

and we note that $p_A + p_B = 1$. These processes repeat over all pixels in a given time-step, and a single update to the ecosystem matrix happens after all pixel state changes are determined. Repeated over many time steps this evolves the state of the system as a whole according the four kinetic parameters, and subject to the two geometric parameters. The total number of simulation time steps is a 'soft' parameter that affects the actual time to simulate (approximately linearly) and, depending on the boundary dynamics and competitive asymmetry, affects the statistical view of boundary dynamics at a particular point in phase space and may allow the simulation to encompass dynamical transitions (e.g. from stable boundaries to unstable boundaries, see next section).

The dynamics that emerge from this algorithm, on average, follow the expected trends from the relative values of the killing rates. If $k_{AB} > k_{BA}$ then the boundary will tend, albeit with stochastic fluctuations, to move in the direction that favors species A, and vice versa if $k_{BA} > k_{AB}$. Even in the absence of explicit simulation, a few trends can be determined, given that $k_{0A} = k_{0B} = 1$. If the killing rates are fast in comparison to the spreading rates, then killing is essentially deterministic ($p_{X0} \rightarrow 1$), vacant sites are relatively common at the interface, and spreading stochasticity controls boundary dynamics. If killing is slow in comparison to the spreading rate, then vacant sites are rare and boundary dynamics slow. Finally, it is worth noting that as a stochastic simulation in a finite simulation space (i.e. a finite number of pixels), ‘anything can happen’. For instance, if $k_{AB} > k_{BA}$ there is a small, but non-zero probability that a series of stochastic fluctuations lead to species B dominating and excluding species A, whereas that would be impossible in a continuum simulation. In the next section we present first results of this stochastic model and discuss types of dynamics that emerged in our first run of simulations.

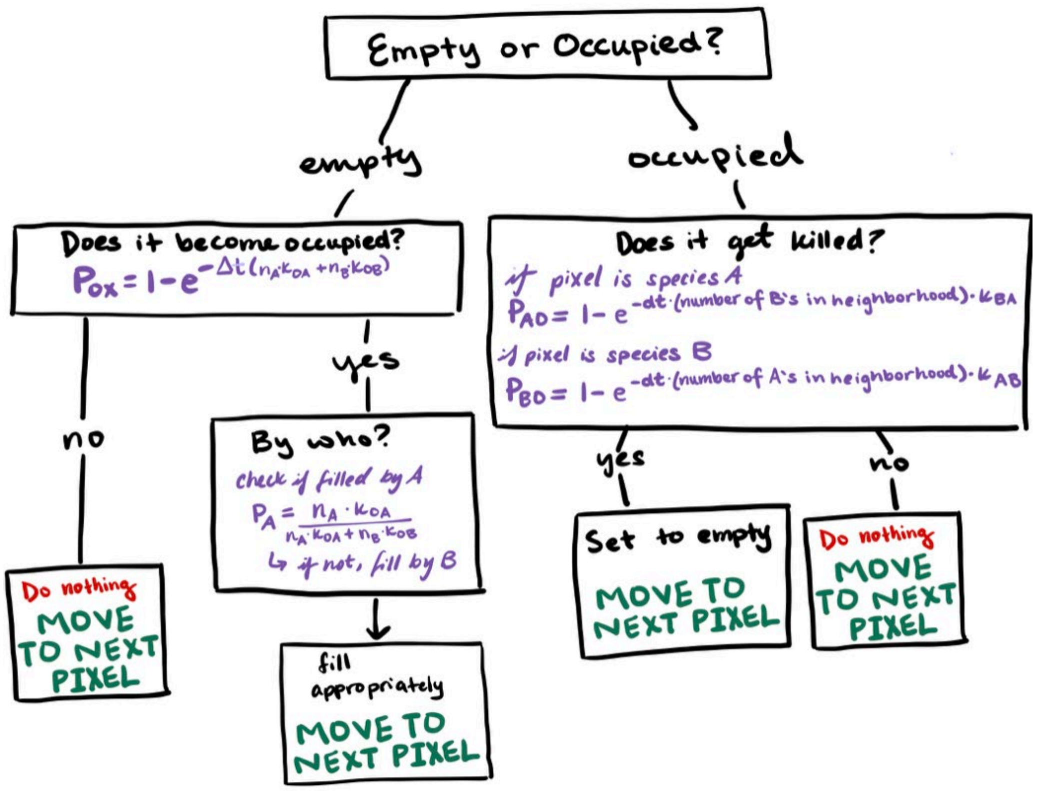

Figure 5: Probabilistic decision tree for spatial competition. At each time step, any pixel that is not a steric pixel (NaN) can change in accordance with the identities of pixels in its interaction neighborhood. The decision tree shown here indicates the series of steps that lead to a possible identity change with the inclusion of weighted stochastic decisions. For a given pixel that *is* a number and is *not* surrounded by identical pixels in its neighborhood, the tree is sampled. Every such pixel is either empty (0) or occupied (+/- 1). If a pixel is empty, a stochastic decision determines if the pixel is colonized, and in a subsequent step, by which species. If a pixel is occupied, a stochastic decision determines if it is killed by local competitors, and thus left vacant.

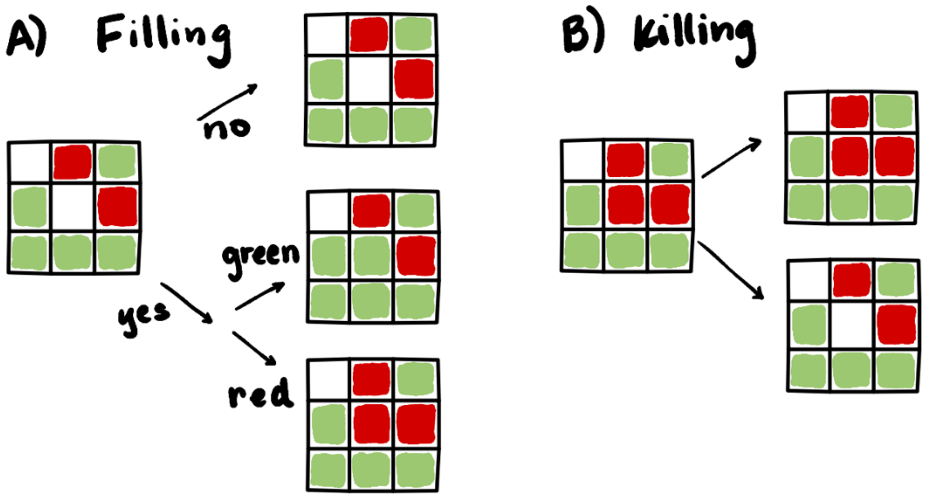

Figure 6: Visualizations of filling and killing operations in a hypothetical 8-connected neighborhood. (A) Upon identifying a vacant pixel, a stochastic binary decision is made to determine if that pixel becomes occupied. If that stochastic outcome indicates filling, a (hypothetical) filling operation ensues; the algorithm then rolls for which species takes over. Under the conditions of identical spreading rate, the probability of state change from vacant to occupied is set by the total number of occupied pixels in the neighborhood, and the identity of the possible occupant is set by the relative numbers of each species in the neighborhood. (B) The central pixel is occupied by a red cell. The green cells in the neighborhood each contribute to the cumulative rate at which the red cell will be killed and hence the site will be vacated. The probability that the central red cell will be killed depends on the killing rate of green on red, the number of green cells in the neighborhood, and the size of the simulation time-step.

Simulation Results

In our initial simulations, we explored neighborhood sizes of two and three, which approximate the biological scale of nearest-neighbor type VI contact-mediated killing[22,23] and near-field secreted toxins. Strictly speaking, stochastic dynamics within a finite simulation space always have a non-zero rate at which a single species dominates, but, relative to the natural time-scale set by the spreading rate, interface dynamics can be classified by long-time averages of the interfacial curvature and the time-scale on which dominance occurs. Preliminary simulations over different sets of the competition parameters k_{AB} and k_{BA} show four general cases: (1) symmetric and stable, (2) asymmetric and stable, (3) asymmetric and semi-stable, and (4) asymmetric and unstable. Inter-pillar distances and pillar radii geometrically dictate the largest interfacial curvature that steric objects can support under asymmetric competition (shown in the following figures as a dashed line)[8]. The combination of kinetic and geometric parameters then correspond to a point in the six-dimensional phase that belongs to one of these classes, with symmetric competition being strictly stable in the case of an arbitrarily large simulation space.

In a manner similar to previous work in the Ursell Lab, we observed interspecies interfaces that are formed between these steric pillars and exhibited stochastic fluctuations that are not observed in our previous continuum model of spatial competition. We selected kinetic parameters that correspond to similar levels of competitive asymmetry from previous continuum work and geometric parameters that might be found in environmental contexts with steric objects on the order of tens of cells (i.e. tens of microns).

In the case where competition is symmetric, the boundary, on average, remains exactly between the two pillars with a characteristic fluctuation width set by the kinetic parameters and the radius of the interaction neighborhood (**Fig 7**). The interface displays no net curvature as there is no overall bias in interface movement that would result from competitive asymmetry. Fluctuations away from a straight boundary occur, but curvature induced fluctuations in the killing and filling rates bring the boundary back to a straight configuration between the pillars. The asymmetric and stable case, while not strictly stable, are those cases that have asymmetric killing kinetics, but remain sufficiently far from the maximum curvature boundary that stochastic interface fluctuations do not push the interface into the unstable configuration on the (very long) time-scale of the simulation. The asymmetric killing rates do however cause the long-time average of the interface configuration to adopt a uniformly curved configuration (**Fig 8**), consistent with results from previous continuum simulations[8,20].

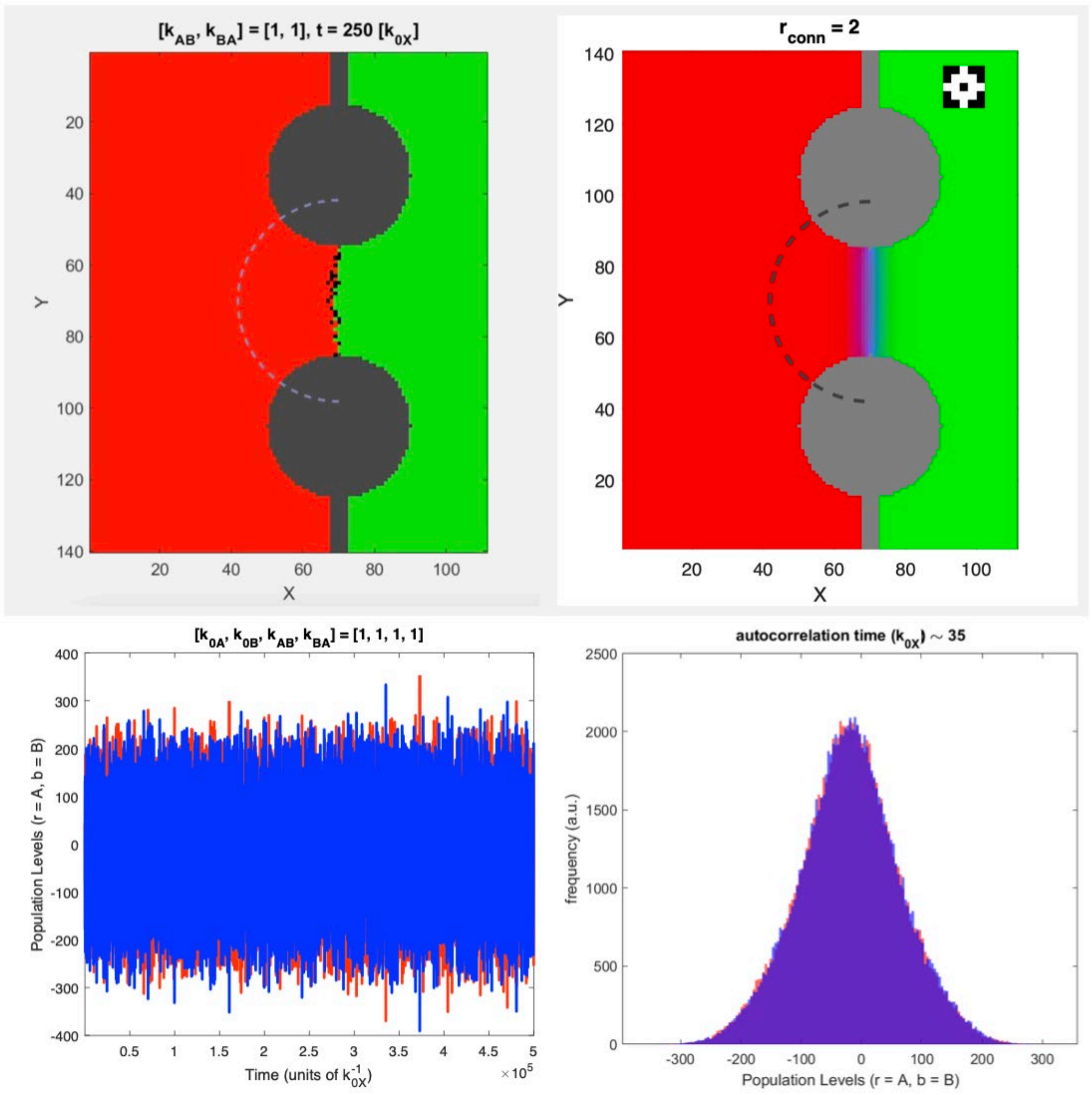

Figure 7: Interspecies average location over time and single time point for *symmetric and stable* case. The gray panel (top left) is a snapshot from the stochastic simulation. The image on the white panel (top right) is the time average across the whole simulation. The gray dashed lines demarcate the largest curve supported by these pillars. The relative population of the equal cases does not shift much, but it does shift because of the creation of empty spaces. These empty spaces cause the populations to not exchange exactly one-for-one. This can be seen by the graph of population differences over time (bottom left) as the differences in population linger around zero. The histogram on the bottom right is the population data for the two species. The apparent average is near zero as there is not competitive asymmetry between the species in this case.

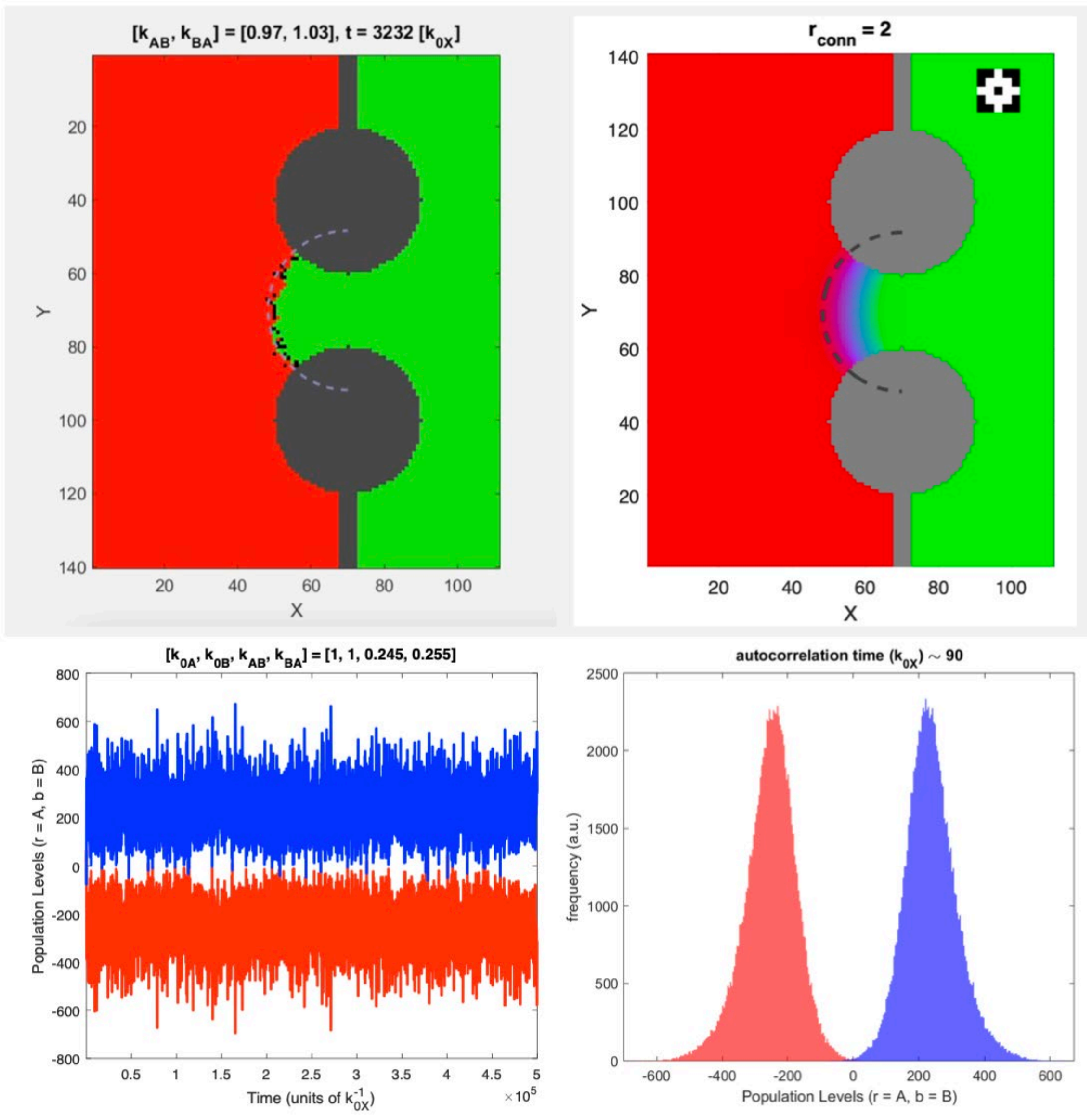

Figure 8: The boundary of the stable case remains curved on long time scales. The average arrangement of two asymmetrically competing species displays a curved interface that lies below the critical interface curvature. The top left panel shows a frame from the simulation and the top left represents location of the interspecies boundary over time. This boundary is curved to support more members of the weaker, red species on one side and fewer, stronger species on the other side. The population difference of the two species remains relatively stable throughout the course of the simulation, as seen in the lower left graph. This can also be seen in the lower right histogram: the histograms are roughly symmetric, but the presence of open spaces removes true symmetry.

Like stable cases, semi-stable cases adopt average uniform interface curvatures for long periods of time, but the asymmetry in competition parameters position the stochastic interface near the critical curvature boundary, and eventually push the interface past the critical curvature boundary well within the simulation time (**Fig 9**). Given enough time, semi-stable simulations will become unstable, and a single species will dominate. This phenomenon is similar to the gambler's dilemma in which there is no condition under which a gambler 'wins', rather stochastic fluctuations will always, eventually absorb the gambler into a state of money 'extinction'. Similar to the gambler's dilemma, the weaker species in our simulation is always attracted to the state of extinction, though that is not to say that it is impossible for the weaker species to dominate—stochasticity necessitates this possibility—but it is extremely unlikely.

Unlike the previous cases, the boundary of the unstable case does not linger in the zone defined by the maximum interface curvature, rather the interface continuously moves until the stronger competitor completely dominates and the simulation ends (**Fig 8**). Macroscopically, the movement is visually similar to a wildfire or tsunami in that the fitter species consistently and progressively dominates the competition space.

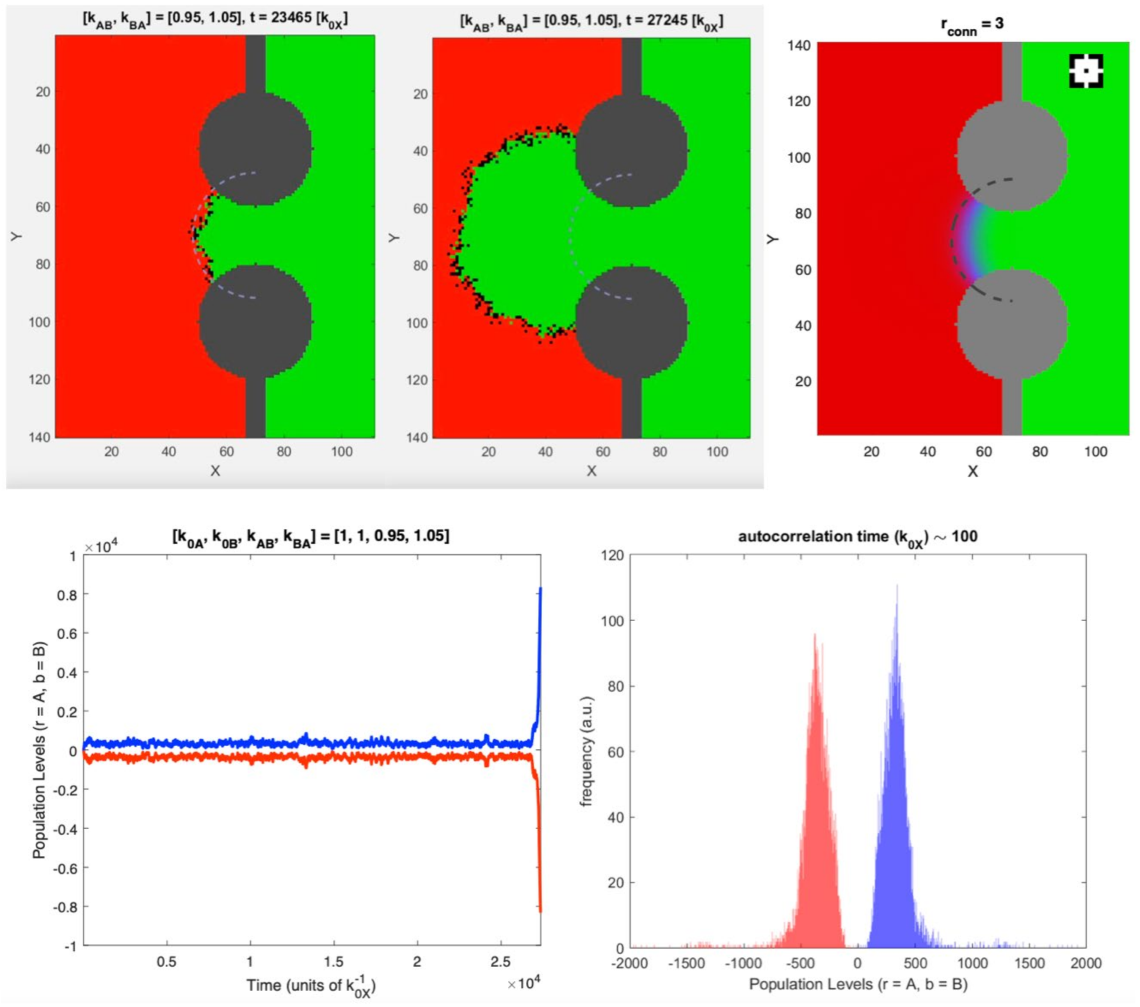

Figure 9: Semi-stable boundary temporal average and two snapshots Semi-stable simulations maintain a stable boundary near the critical curvature (upper left), but eventually cross that boundary and become unstable (upper middle), at which point the strong competitor dominates the simulation domain. This is accompanied by population levels remaining stable for some number of timesteps, then they diverge quickly as one species overtakes the other, as seen in the lower left graph. The histogram in the lower left shows a few time points where the population difference is very large. This would be near the end of the simulation as the strong competitor takes over. The location of the interspecies boundary (upper right) looks like an asymmetric, stable case at first glance as this simulation is mechanistically similar until it suddenly becomes unstable.

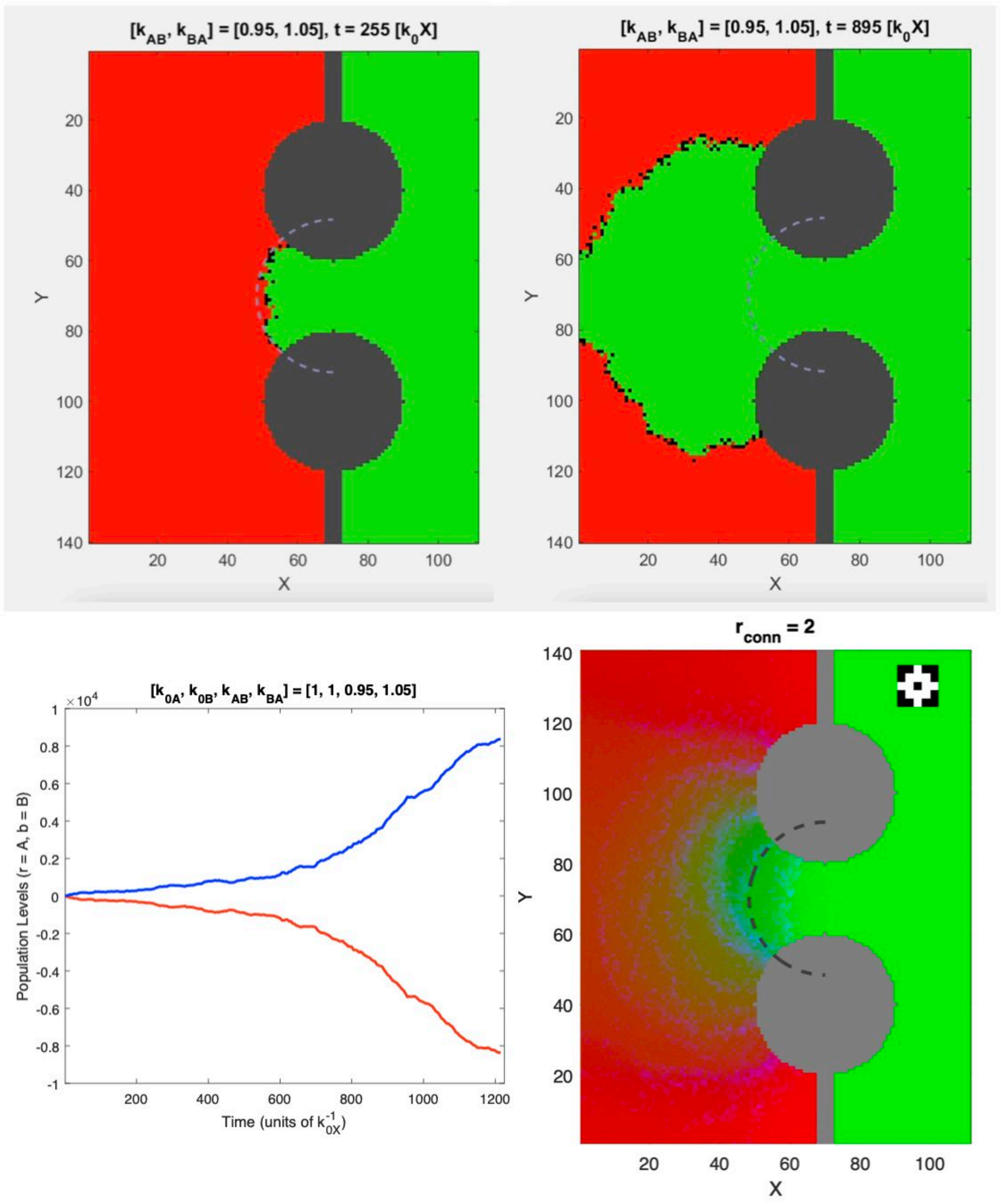

Figure 10: Unstable Boundaries continuously move until one species conquers the other. Unstable conditions result when the curvature that could potentially stabilize the boundary is greater than the maximum interface curvature set by pillar geometry (denoted by the dashed line). In this case, the system does not exhibit any stable dynamics, rather the interspecies interface consistently and progressively moves toward the dominance of the stronger competitor (lower right). Population levels diverge from the start of the simulation (lower left).

Discussion and Future Directions

This work lays the foundation for multiple rich directions of inquiry into the effects of stochasticity on competitive ecosystem dynamics. We identified a number of qualitatively distinct classes of dynamics that both expand our understanding of these systems and suggest crucial differences between continuum and discrete / stochastic models. Importantly, key takeaways thus far are: (i) that neighborhood size has significant effects, albeit not yet fully characterized, on the dynamics and stability of asymmetric competition that are not revealed by continuum simulations, and (ii) that stochasticity 'blurs the line' between stable and unstable systems, in particular suggesting that all finite-sized stochastic ecosystems have the possibility of single-species dominance. Within the language used, the current code is optimized in terms of computational efficiency and data handling, and four directions of inquiry immediately follow from our current code and results.

First, we seek to understand the structure of the wait-time distribution for single-species dominance, which requires many tens of thousands of simulations of the current code. Technically, because the simulation is stochastic, even symmetric competition has some (very long) timescale on which a finite-sized system is expected to reach a state of single-species dominance. In such stochastic processes it is difficult to pin-point when a system transitions from stable to unstable, but system geometry gives a clear delineator, specifically, the time at which all points along the competition interface lie beyond the line of maximum curvature between the pillars. Coupled with comparison to possible theoretical studies, the structure of the wait-time distribution may indicate different classes of expectation. In particular, if the wait time distribution decays with certain power-law exponents, which themselves may depend on the kinetic parameters, there may be finite or infinite average wait times for single-species dominance, which constitute qualitatively distinct behavior.

Second, we are in the midst of developing additional tools for analysis of the simulations. We are developing relaxation algorithms that find the approximate curvature of the interspecies interface as a function of time. As interface curvature is the geometric factor that influences the spatial balance of asymmetric competitors, we hypothesize that in the quasi-stable situations, the stability transition is ‘nucleated’ by fluctuations in interface curvature. We will test this hypothesis with kymograph analysis of the temporal evolution of interface curvature. In general, we will test different potential ‘reaction coordinates’ that characterize the transitory nature of the dynamics – for instance, one potential reaction coordinate might be a measure of distance or area between the current interface and the line of maximum interface curvature. Such metrics may prove useful in characterizing the kinetic landscape that leads to the wait-time distribution discussed above. We are also developing analysis tools to measure effective system dynamics. Using both whole-population and spatial time-series data, we will characterize how autocorrelation times (effective dynamics) depend on kinetic parameters.

Third, building on these kinetic metrics, we will map out a portion of the parametric phase space of the system – in particular, characterizing wait-time distributions and stability kinetics as a function of geometric parameters—primarily pillar spacing—and competition parameters. With sufficient computational power, we will map out features (e.g. scaling exponents) of the wait-time distribution to determine if certain regions of the kinetic phase space obey qualitatively different dynamics.

Fourth, while the size of the parameter space is daunting, multispecies competitive ecosystems (3+ species) existing within structured environments (i.e. containing steric objects) are of significant interest as a step toward modeling more realistic natural contexts.

As a student, this project has been educational – it gave me experience in algorithm design, code implementation, debugging, and optimization, specifically within the context the powerful MATLAB language; it provided opportunities for narrative construction around my project – both in written and presented forms; it allowed me to observe and engage with the trajectory of a multi-year research project, and I developed skills for scientific inquiry and organizational techniques for long-term project management. I look forward to seeing this project progress and the insights it will uncover.

Coding Methods

In addition to the fundamental parameters that control geometries and stochastic processes in the simulations, the simulation script offers control over many aspects of simulation setup, data structures, visualization and saving of simulation data and meta-data. In this section we discuss some of the more salient control parameters; the full code designed and used in this work can be found in Appendix A. The simulation gives the option to use a random or preset seed for the random number generator. If a random (time-based) seed is chosen, simulations with the same kinetic and geometric parameters sample distinct stochastic trajectories through phase space with the same underlying statistical ensemble. While our initial simulation run did not perform large-scale sampling of this type, this is how future simulations will sample (for instance) residence-time distributions. Conversely, preset seeds that follow the same stochastic trajectory are useful for unit testing, diagnostics, and code improvement. All simulations are initialized with a flat interspecies boundary exactly between the pillars – this ensures that there are no biases from initial conditions. Choosing an appropriate time step is critical, as time steps that are too short waste computational cycles and times steps that are too long will not accurately simulate the underlying stochastic dynamics. The killing and filling probabilities, associated with the first steps of the decision tree, monotonically increase with the size of the time step. If the time step is too large these probabilities approach 1 and killing and filling become semi-deterministic, which eliminates stochasticity from the simulation and inaccurately represents the dynamics. To ensure an appropriate choice of time step we calculated the maximum possible rate of a pixel-state change k_{max}, which depended both on the fastest of the kinetic parameters $(k_{0A}, k_{0B}, k_{AB}, k_{BA})$ and on the number of pixels in the local interaction neighborhood. We then chose the simulation time step to be a fraction of this inverse rate; for the simulations shown here we used $\Delta t = (4k_{max})^{-1}$.

The temporal resolution of recorded species matrix data, images, and other meta-data during the simulation was set by the user-specified parameter $t\_capture$, measured in natural time units of k_{0X}^{-1}. This resolution was chosen to balance output data size with the requirement that successive images had substantially non-zero autocorrelations (i.e. the flow of time displayed continuity from one image to the next). Each simulation also saves all relevant simulation parameters and data into a standardized and labeled output data structure, called $compdata$. Each such output data structure can be used as the input to reinitialize / repeat that simulation and/or to re-construct the visualizations of a particular simulation.

The main body of the simulation can be split into the three conceptual parts. First, the code determines the minimum rectangular area of pixels where any possible state changes in pixel identity can occur. This significantly improves computational efficiency because the vast majority of the time, dynamics are spatially localized to a curved, approximately 1D interface between the pillars. Pixels more than a neighborhood-radius away cannot change state and hence are excluded from the current calculation to save on computational time. This rectangular region is dynamically updated at each time-step. Second, within the entire simulation area, a group of 'active' pixels are determined by a series of logical evaluations that determine exactly which pixels could *possibly* change state, and the decision tree is only applied to those pixels, again to save on computational time. Active pixels that can change state must either be open pixels with at least one occupied pixel in their neighborhood, or occupied pixels with at least one competitor pixel in their neighborhood. This combination of dynamic update rules ensures that at each time step only those pixels that have a chance of changing identity are subjected to the relatively costly process of the decision tree. Finally, for each of the active pixels, the appropriate rates of killing (if occupied) or filling (if open) are calculated given the composition of the local neighborhood, the change probabilities are calculated, stochastic selection against a uniform random distribution occurs, and the stochastic decision tree is followed as described earlier.

Appendix A: Simulation Code (Matlab2019B)

```
%Isa Richter & Tristan Ursell
%May 2021
%Stochastic competition in 2D environments with steric structure
% Matlab R2019B
%
% Parameters:
%
% - each pixel = ~ 1 cell = ~ 1 micron
% - set k0A = k0B, set k0X = 1, which gives dt meaning
% - 'biology' of contact-mediated killing suggests 'minimal' values for r_hood = 2 (4 or
8-conn)
% - fix R, vary L -- this will sample kappa_max
% - t_total & W -- pick high enough value, but no point in full sampling
%
%
%%%%%%%%%%%%%%%%%%%%%%%%
%simulation parameters
%%%%%%%%%%%%%%%%%%%%%%%%
%
% DIMENSIONS
% L_box = box height (pixels)
% W_box = box width (pixels)
% D = center-to-center pillar distance (pixels)
% R = pillar radius (pixels)
% x_pillar = x position of pillar (pixels, width)
%
% RATES & TIME
% t_total = total simulation time (k_{0X})
% k0A = rate of empty space colonization of A (=1 for now, sets time scale)
% k0B = rate of empty space colonization of B (=1 for now, sets time scale)
% kAB = rate of A killing B (k_{0X})
% kBA = rate of B killing A (k_{0X})
%
% BOOLEANS
% save_image_q = save (8-bit) images every t_capture?
% disp_fig_q = display live figure in real time?
% save_data_q = save 'compdata' to mat-file?
% save_fig_q = save output figure?
% rand_q = 0, uses fixed rng seed, = 1 uses varying, clocked-based seed
% test_q = unit testing for fast convolution (not important generally)
%
% HARD WIRED (change in code)
% dt = proportional to a fraction of the fastest time scale (max([k0A,k0B,kAB,kBA]))
% t_capture = rate of capture in natural units
%
%
%stochastic_competition_v4(L_box,W_box,D,R,x_pillar,r_hood,t_total,k0A,k0B,kAB,kBA,
save_image_q,disp_fig_q,save_data_q,save_fig_q,test_q)
%
function compdata =
stochastic_competition_v4(L_box,W_box,D,R,x_pillar,r_hood,t_total,k0A,k0B,kAB,kBA,s
ave_image_q,disp_fig_q,save_data_q,save_fig_q,rand_q)
%------------------------------------------------------------------------
-
%randomize seed
seed0 = 128248780;
seed1 = round(prod(clock));
if rand_q ==0
   rng(seed0)
else
   rng(seed1)
end

%Initial layout conditions and set up
%random initial conditions
%here M(i,j) = +1 --> species 1
%here M(i,j) = -1 --> species 2

%%%%%%%% NEIGHBORHOOD / STREL for Pixel of Interest (POI) %%%%%%%%%
if r_hood==0
   conn_nb = [ 1 1 1; 1 0 1; 1 1 1]; %8 connected neighborhood w/o poi
else
   strel1 = strel('disk',r_hood,0);
   conn_nb = strel1.Neighborhood; % circular neighborhood
   conn_nb(r_hood+1,r_hood+1) = 0; %set center to zero
end

%get conn_nb radius
r_conn = (size(conn_nb,1)-1)/2;
```

```
%%%%%% RANDOM IC / PILLAR CENTERED ON EDGE %%%%%
%M = 2*(rand(L,L)>(1-f))-1; %rand conditions

%%%%%% 50-50 CENTERED IC / PILLAR WITH EXTRA SPACE
%x-position of pillars (hardwired for now)
if x_pillar==0
   x_pillar = round(W_box/2);
end

M = zeros(L_box,W_box);
M(:,1:x_pillar) = 1;
M(:,x_pillar+1:end) = -1;

%pillar centers
origin_x1 = x_pillar;
origin_y1 = round((L_box - D)/2);
origin_x2 = x_pillar;
origin_y2 = round(L_box - (L_box - D)/2);

%create pillars
strel_R = strel('disk',R,0);
temp1 = zeros(size(M));
temp1(origin_y1,origin_x1)=1;
temp1(origin_y2,origin_x2)=1;
mask1 = imdilate(temp1,strel_R);

%apply pillars to M and set to NaNs
M(mask1==1) = NaN;
M(1:origin_y1,x_pillar - r_conn:x_pillar + r_conn) = NaN;
M(origin_y2:end,x_pillar - r_conn:x_pillar + r_conn) = NaN;
mask_nan = isnan(M);

%matrix whose values are the number of local (conn_nb) non-nans
strel_mat = conv2(~isnan(M), conn_nb, 'same');

%get size of non-nan area
non_nan_area = sum(~isnan(M(:)));

%%%%%%%%%% TIMING %%%%%%%%%%%%%%%%%%%%
%set dt to be smaller than any inverse rate constant
kmax = max([k0A,k0B,kAB,kBA]);
```

```
dt = 0.25/(kmax*sum(conn_nb(:)));

%t_capture = round(0.25/dt); %very fine scale
%t_capture = round(2/dt); %fine scale
t_capture = round(5/dt); %medium-fine scale
%t_capture = round(10/dt); %medium scale
t_vec = 0:(dt*t_capture):t_total;

%population record
popRecord = zeros(round(t_total/dt*1/t_capture),2);
%-------------------------------------------------------------
%create image matrix
if save_image_q
    %image_out = zeros([size(M),length(t_vec)],'uint8');
    image_out = zeros([size(M),length(t_vec)],'single');
end

%generate output file base name
basename1 = ['stochastic_competition_R-' num2str(R) '_dt-' num2str(dt) '_r-hood-'
num2str(r_hood) ...
    '_k0A-' num2str(k0A) '_k0B-' num2str(k0B) '_kAB-' num2str(kAB) '_kBA-'
num2str(kBA) '_ID-' num2str(seed1)];

%save parameters to output
compdata.name = basename1;
compdata.L = L_box;
compdata.W = W_box;
compdata.D = D;
compdata.R = R;
compdata.r_hood = r_hood;
compdata.r_conn = r_conn;
compdata.Reff = sqrt(sum(strel_R.Neighborhood(:))/pi);
compdata.Rmin = compdata.Reff*sqrt(((D-1)/(2*compdata.Reff))^2-1); %
1/kappa_max
compdata.x_pillar = x_pillar;
compdata.IC = M;
compdata.t_total = t_total;
compdata.dt = dt;
compdata.t_capture = t_capture;
compdata.conn_neighborhood = conn_nb;
compdata.k0A = k0A;
compdata.k0B = k0B;
compdata.kAB = kAB;
compdata.kBA = kBA;
compdata.save_image_q = save_image_q;
compdata.save_data_q = save_data_q;
compdata.save_fig_q = save_fig_q;
compdata.rand_seed = seed0;
compdata.unique_ID = seed1;

%determine initial species amounts at perfect pillar center-line
temp_bound = sum(isnan(M),2);
bound_IC_length = sum(temp_bound==0);
compdata.A0 = sum(M(:)==1) - bound_IC_length/2;
compdata.B0 = sum(M(:)==-1) + bound_IC_length/2;

%Initial parameters/conditions for simulation
disp('v1.2')
if save_image_q
   w1 = whos('image_out');
   disp(['output image size (GB): ' num2str(w1.bytes/1e9)])
end
pause(1)
close all
clc
drawnow

%this is boundary calculation stuff
Madd = zeros(size(M));

%-------------------------------------------------------------------------
-
%Simulation time!
meanA = zeros(size(M));
meanB = zeros(size(M));
mean0 = zeros(size(M));
tic
cond1_mat = isnan(M); %is nan
v = 0;
for p = 1:round(t_total/dt)
   %get species matrices
   temp_Mp1 = M== 1;
```

```
    temp_Mm1 = M==-1;

    %counts species numbers in active neighborhood
    %this creates a continually updated, much smaller matrix for convolution (hence
faster)
    %on p=1, will automatically go to 'catch' -- this needs to happen
    try
        %get sub-matrix of M
        min_r_act = min(r_act);
        max_r_act = max(r_act);
        min_c_act = min(c_act);
        max_c_act = max(c_act);

        Mtemp = M(min_r_act - r_conn-1:max_r_act + r_conn+1,min_c_act - r_conn-
1:max_c_act + r_conn+1);

        tempA_nb = conv2(Mtemp == 1, conn_nb, 'valid');
        tempB_nb = conv2(Mtemp ==-1, conn_nb, 'valid');

        indsY = min_r_act-1:max_r_act+1;
        indsX = min_c_act-1:max_c_act+1;

        A_nb(indsY,indsX) = tempA_nb;
        B_nb(indsY,indsX) = tempB_nb;

        fast_convq = 1;
    catch
        A_nb = conv2(temp_Mp1, conn_nb, 'same');
        B_nb = conv2(temp_Mm1, conn_nb, 'same');

        fast_convq = 0;
    end

    %remove pixels with trivial conditions
    cond2_mat = and(temp_Mp1,A_nb==strel_mat); %A surrounded by A's
    cond3_mat = and(temp_Mm1,B_nb==strel_mat); %B surrounded by B's
    cond_all = (cond1_mat + cond2_mat + cond3_mat)==0;

    %get active block and vectorize active positions
    [r_act,c_act] = find(cond_all);
    active_vec = sub2ind(size(M),r_act,c_act);
```

```
Mpoi = M(active_vec);
Apoi = A_nb(active_vec);
Bpoi = B_nb(active_vec);

%run through active pixels
Mcopy =  M;
for q=1:length(active_vec)
   % Is it an open spot?
   if Mpoi(q) == 0
      Ao_potential = Apoi(q)*k0A;
      k_sum_colonize = Ao_potential + Bpoi(q)*k0B;

      %Does our poi get colonized?
      if rand < (1 - exp(-dt*k_sum_colonize))
         if rand < (Ao_potential/k_sum_colonize)
            Mcopy(active_vec(q)) = 1; %set to species A
         else
            Mcopy(active_vec(q)) = -1; %set to species B
         end
      end
   end

   % Is poi an A with any non A's around (B's)?
   if and(Mpoi(q) == 1, Bpoi(q) > 0)
      %Does the A poi get killed by a B?
      if rand < (1-exp(-dt*Bpoi(q)*kBA))
         Mcopy(active_vec(q))=0;
      end
   end

   % Is poi a B with any non B's around (A's)?
   if and(Mpoi(q) == -1, Apoi(q) > 0)
      %Does the B poi get killed by an A?
      if rand < (1-exp(-dt*Apoi(q)*kAB))
         Mcopy(active_vec(q))=0;
      end
   end
end

%update to new matrix
M = Mcopy;
```

```
%save average of M over t_capture time steps
%(for contour finding)
Madd = Madd + M;

%output figure
if mod(p,t_capture)==1
    toc
    if fast_convq==0
        disp('Error (edge?) disabling fast sub-matrix convolution.')
    end
    v = v + 1;

    %split current data by species
    temp_Mp1  = M==  1;
    temp_Mm1  = M== -1;
    temp_0    = M==  0;

    %update means
    meanA = meanA + temp_Mp1;
    meanB = meanB + temp_Mm1;
    mean0 = mean0 + temp_0;

    %record population levels
    temp_p1  = sum(temp_Mp1(:));
    temp_m1  = sum(temp_Mm1(:));
    popRecord(v,1) = temp_p1;
    popRecord(v,2) = temp_m1;

    if or(disp_fig_q,save_image_q)
        %%{
        temp = 255*mat2gray(Madd.^2);
        temp(mask_nan) = -1;
        Madd = zeros(size(M));
        %}

        %{
        temp=M;
        temp(temp_Mm1) = 0;
        temp(temp_0) = 1;
        temp(temp_Mp1) = 2;
```

```
            temp(isnan(M)) = 3;
            %}

            if disp_fig_q
                imagesc(temp*3/255,[-1 3])
                xlabel('X')
                ylabel('Y')
                box on
                axis equal tight
                title([num2str(p) ' of ' num2str(t_total/dt) ', [k_{0A}, k_{0B}, k_{AB}, k_{BA}] =
['...
                    num2str(k0A) ', ' num2str(k0B) ', ' num2str(kAB) ', ' num2str(kBA) '],
r_{conn} =' num2str(r_conn)])
                drawnow
            else
                disp([num2str(p) ' of ' num2str(t_total/dt)])
            end
        end

        if save_image_q
            %image_out(:,:,v) = uint8(temp);
            image_out(:,:,v) = single(temp);
        end

        %check condition for terminal (case of all one type)
        if or(temp_p1 == non_nan_area,temp_m1 == non_nan_area)
            break
        end
        tic
    end
end
disp('Sim done.')

%collect output population data, terminal time
compdata.t_vec = t_vec(1:v);
compdata.popA = popRecord(1:v,1);
compdata.popB = popRecord(1:v,2);
compdata.meanA = meanA/v;
compdata.meanB = meanB/v;
compdata.mean0 = mean0/v;
```

```
%estimate autocorrelation
[acf,lags,~] = autocorr(compdata.popA,round(v/2));
ind1 = find((acf-exp(-1)).^2==min((acf-exp(-1)).^2),1,'first');
t_corr = compdata.dt*compdata.t_capture*lags(ind1);
compdata.t_corr = t_corr;

%create and save RGB mean output image
%T(:,:,1) = isnan(M)*0.4 + 0.9*compdata.meanA;
%T(:,:,2) = isnan(M)*0.5 + 0.9*compdata.meanB;
%T(:,:,3) = isnan(M)*0.9 + 0.9*compdata.meanA;
T(:,:,1) = isnan(M)*0.5 + 0.9*compdata.meanA;
T(:,:,2) = isnan(M)*0.5 + 0.9*compdata.meanB;
T(:,:,3) = isnan(M)*0.5 + 0.9*mat2gray(compdata.mean0);
compdata.mean_im_rgb = T;

%*************************************
%to re-create figs, execute from here
%*************************************

%mean-adjusted population vectors;
popA = compdata.popA - compdata.A0;
popB = compdata.popB - compdata.B0;

%figure output
f1 = figure('Position',[250 300 1640 550]);
subplot(1,3,1)
hold on
plot(compdata.t_vec,popA,'linewidth',2,'color',[1 0.2 0])
plot(compdata.t_vec,popB,'linewidth',2,'color',[0 0.2 1])
xlabel('Time (units of k_{0X}^{-1})')
ylabel('Population Levels (r = A, b = B)')
box on
title(['[k_{0A}, k_{0B}, k_{AB}, k_{BA}] = ['...
   num2str(compdata.k0A) ', ' num2str(compdata.k0B) ', ' num2str(compdata.kAB) ', '
num2str(compdata.kBA) ']'])

subplot(1,3,2)
hist_vec = min([popA; popB]):4:max([popA; popB]);
histA = hist(popA,hist_vec);
histB = hist(popB,hist_vec);
hold on
```

```
b1 = bar(hist_vec,histA,1,'r','edgecolor','none');
b2 = bar(hist_vec,histB,1,'b','edgecolor','none');
xlabel('Population Levels (r = A, b = B)')
ylabel('frequency (a.u.)')
box on
title(['autocorrelation time (k_{0X}) \sim ' num2str(t_corr)])
b1.FaceAlpha = 0.6;
b2.FaceAlpha = 0.6;

subplot(1,3,3)
hold on
imagesc(compdata.mean_im_rgb)

Reff = compdata.Reff;
Rmin = compdata.Rmin;
D = compdata.D;
alpha_c=atan2(sqrt(-4*Reff^2/D^2+1),2*Reff/D);
x0 = (1/2)*(2*cos(alpha_c)^2*Reff+2*sin(alpha_c)^2*Reff-
cos(alpha_c)*D)/sin(alpha_c);
y0 = (origin_y1 + origin_y2)/2;

theta0 = pi/2:0.01:(3*pi/2);
plot(x0 + x_pillar + Rmin*cos(theta0),y0 + Rmin*sin(theta0),'--
','linewidth',2,'color',[0.25 0.25 0.25])
axis equal tight
xlabel('X')
ylabel('Y')
box on
title(['r_{conn} = ' num2str(compdata.r_conn)])

axes('pos',[.675 .83 .07 .07])
imshow(compdata.conn_neighborhood)

%fig-file output
if save_fig_q
   savefig(f1,[basename1 '.fig'])
end

%mat-file output
if save_data_q
   save([basename1 '.mat'],'compdata')
```

```
end

%image output
if save_image_q
    save([basename1 '.mat'],'image_out','-append')
    %{
    out_name = [basename1 '.tif'];
    for i =1:v
        image_save(uint8(image_out(:,:,i)),out_name)
    end
    %}
End
```

Bibliography

1. Faust, K. *et al.* Microbial Co-occurrence Relationships in the Human Microbiome. *PLOS Comput. Biol.* **8**, e1002606 (2012).
2. Guo, S. *et al.* Protists as main indicators and determinants of plant performance. *Microbiome* **9**, 64 (2021).
3. McNally, L. *et al.* Killing by Type VI secretion drives genetic phase separation and correlates with increased cooperation. *Nat. Commun.* **8**, 14371 (2017).
4. Coyte, K. Z., Schluter, J. & Foster, K. R. The ecology of the microbiome: Networks, competition, and stability. *Science* **350**, 663–666 (2015).
5. Obadia, B. *et al.* Probabilistic Invasion Underlies Natural Gut Microbiome Stability. *Curr. Biol.* **27**, 1999-2006.e8 (2017).
6. Kerr, B., Riley, M. A., Feldman, M. W. & Bohannan, B. J. M. Local dispersal promotes biodiversity in a real-life game of rock–paper–scissors. *Nature* **418**, 171–174 (2002).
7. Hart, S. P., Turcotte, M. M. & Levine, J. M. Effects of rapid evolution on species coexistence. *Proc. Natl. Acad. Sci.* **116**, 2112–2117 (2019).
8. Lowery, N. V. & Ursell, T. Structured environments fundamentally alter dynamics and stability of ecological communities. *Proc. Natl. Acad. Sci.* **116**, 379–388 (2019).
9. Daniel, R. The metagenomics of soil. *Nat. Rev. Microbiol.* **3**, 470–478 (2005).
10. Stocker, R. & Seymour, J. R. Ecology and Physics of Bacterial Chemotaxis in the Ocean. *Microbiol. Mol. Biol. Rev.* **76**, 792–812 (2012).
11. Ley, R. E., Peterson, D. A. & Gordon, J. I. Ecological and Evolutionary Forces Shaping Microbial Diversity in the Human Intestine. *Cell* **124**, 837–848 (2006).

12. Henkel, J. S., Baldwin, M. R. & Barbieri, J. T. Toxins from bacteria. in *Molecular, Clinical and Environmental Toxicology: Volume 2: Clinical Toxicology* (ed. Luch, A.) 1–29 (Birkhäuser, 2010). doi:10.1007/978-3-7643-8338-1_1.

13. Mavridou, D. A. I., Gonzalez, D., Kim, W., West, S. A. & Foster, K. R. Bacteria Use Collective Behavior to Generate Diverse Combat Strategies. *Curr. Biol.* **28**, 345-355.e4 (2018).

14. Hardin, G. The Competitive Exclusion Principle. *Science* **131**, 1292–1297 (1960).

15. Melkonian, C. *et al.* Finding Functional Differences Between Species in a Microbial Community: Case Studies in Wine Fermentation and Kefir Culture. *Front. Microbiol.* **10**, (2019).

16. Kalziqi, A. *et al.* Immotile Active Matter: Activity from Death and Reproduction. *Phys. Rev. Lett.* **120**, 018101 (2018).

17. Steinbach, G., Crisan, C., Ng, S. L., Hammer, B. K. & Yunker, P. J. Accumulation of dead cells from contact killing facilitates coexistence in bacterial biofilms. *J. R. Soc. Interface* **17**, 20200486 (2020).

18. Miura, H., Kondo, Y., Matsuda, M. & Aoki, K. Cell-to-Cell Heterogeneity in p38-Mediated Cross-Inhibition of JNK Causes Stochastic Cell Death. *Cell Rep.* **24**, 2658–2668 (2018).

19. Wiles, T. J. *et al.* Host Gut Motility Promotes Competitive Exclusion within a Model Intestinal Microbiota. *PLOS Biol.* **14**, e1002517 (2016).

20. Ursell, T. Structured environments foster competitor coexistence by manipulating interspecies interfaces. *PLOS Comput. Biol.* **17**, e1007762 (2021).

21. Rivera, J. *et al.* Bacillus anthracis produces membrane-derived vesicles containing biologically active toxins. *Proc. Natl. Acad. Sci.* **107**, 19002–19007 (2010).

22. Silverman, J. M., Brunet, Y. R., Cascales, E. & Mougous, J. D. Structure and Regulation of the Type VI Secretion System. *Annu. Rev. Microbiol.* **66**, 453–472 (2012).

23. Brunet, Y. R., Espinosa, L., Harchouni, S., Mignot, T. & Cascales, E. Imaging Type VI Secretion-Mediated Bacterial Killing. *Cell Rep.* **3**, 36–41 (2013).