HOST-SYMBIONT COEVOLUTION IN DIGITAL AND MICROBIAL SYSTEMS

By

Luis Zaman

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science – Doctor of Philosophy
Ecology, Evolutionary Biology, and Behavior – Dual Degree

2014

ABSTRACT

HOST-SYMBIONT COEVOLUTION IN DIGITAL AND MICROBIAL SYSTEMS

By

Luis Zaman

Darwin's image of the entangled bank captures foremost the pervasiveness of life as it clothes the earth, but it also captures how intimately species interact and often depend on one another. This interaction is particularly pronounced for obligate parasites, who's livelihoods depend on interactions with their hosts and who's hosts often pay severely. In my thesis, I first demonstrate how antagonistic coevolution in Avida leads to a diverse set of interacting host and parasite phenotypes: a digital entangled bank. Second, I show how further evolution is embedded within this community context by studying the coevolution of complexity driven by parasites' *population genetic memory* – where the diversifying community of parasites "remembers" previously evolved hosts.

Continuing to study the intersection of coevolution and community ecology, I investigate the structure of communities produced by the coevolutionary process in Avida. I show that a nested structure of interactions is common in our experiments, which is the same structure often found in natural host-parasite and plant-pollinator communities as well as many phage-bacteria interaction networks. In addition, I show that "growing" networks are nested by virtue of the process of incrementally adding nodes and edges. Thus, coevolution is expected to produce significantly nested communities when compared to random networks. However, the coevolved digital host-parasite networks are significantly more nested than expected from this neutral growth process.

The interactions between hosts and their intimately interacting partners are not just parasitic, instead they span a broad range and include many mutualistic interactions. In the last section of my thesis, I study evolution and coevolution along the parasitism-mutualism continuum using a temperate λ phage system that provides its host with access to an otherwise unavailable metabolic pathway. Instead of evolving more mutualistic phage as I predicted, both the phage and bacteria

evolved cheating strategies.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

The only thing harder than coming up with a title for this thesis is defining the discipline I belong to. With absolute certainty, I started off as a computer scientist. But somewhere along the way, I ended up on a path that would lead me to a pipet in one hand and the other waving a flask full of *E. coli* over a Bunsen burner flame. Computer science is a discipline of abstraction; the core of an algorithm is simply a well-defined process, and many details never matter *in theory*. I find computer science attractive because, starting from a completely abstract foundation, engineers can build things like telephones and unmanned aircraft. The first time evolution captured my attention was while working with robots. Unfortunately, that sounds less dramatic at Michigan State University than it would at many other places. I learned that evolution, just like the other abstract processes we call algorithms, could be used as a tool to solve engineering problems [48].

The field of evolutionary computation, where ideas from evolution are borrowed to help find solutions to a wide range of problems, started to gain popularity in the 1970s due to John Holland's theoretical work and the increase in available computational power [52, 73]. Fogel and Koza took ideas from genetic algorithms and applied them to computer programs, letting evolutionary exploration and optimization find new and hopefully better algorithms or automata rather than just parameters [47, 86]. Expanding these techniques to the artificial intelligence domain lead to a relatively large group of researchers interested in evolving the controllers and morphology of

robots – their "bodies and brains" [103, 109]. Evolved controllers are most often artificial neural networks, which are another biologically inspired abstraction of the networks and signals in nervous systems [45, 119]. The field of evolutionary robotics has continued to adopt biological abstractions such as genetic regulatory networks by HyperNEAT and the evolution of development in Josh Bongard's work [17, 152, 153].

SIMULATION OF GENETIC SYSTEMS. VI   151

II. STRUCTURE OF THE EPISTASIS PROGRAMME

The programme has six main sections:

(1) Extract, without replacement, a pair of parents at random from the given set of parents.
(2) Form a set of progeny from these parents.
(3) Determine the phenotypes of the progeny.
(4) Select potential parents from the set of progeny.
(5) Repeat (1)–(4) until all parents have produced the specified number of progeny.
(6) Print out any required information.
(7) Repeat (1)–(6) using the selected progeny as parents.

Figure 1.1: Outline of the first published genetic algorithm used to study epistasis in 1960 (from [55]).

While robots initially peaked my interests in evolution, I quickly turned to another avenue of research: using computer science as a way of studying evolution. Mathematical and computational biology have a long history, and evolution has enjoyed a particularly prolific partnership with computer science [102]. In fact, the first published genetic algorithm was used to simulate evolution on epistatic landscapes in 1960, not to solve optimization problems (see Figure 1.1) [55]. Many models in ecology and evolution require nothing more than pencil and paper, like Volterra's famous 1926 model investigating Adriatic fisheries [173]. However, more complicated models often require the use of extensive computational power such as the adaptive Dynamic Global Vegetation Model, which attempts to capture the dynamics of plant productivity on a global scale (aDGVM) [49, 71].

Computational contributions are not limited to methods of numerical integration. They include individual and agent based systems that can be used to investigate the role of variation at different scales (e.g., individual, spatial, genetic, etc.) [35, 101]. More recently, "digital organisms" have provided an alternative instance of Darwinian evolution to study *in silico* [124, 126].

It was the work done with Avida, a digital life platform, that convinced me computer scientists could do more than develop tools to study evolutionary biology. Instead of using natural populations or cultures of microbes to investigate the processes of evolution, Avida allows researchers to study self-replicating computer programs competing for resources in a virtual world [3]. Avida was originally inspired by the Core War video game, where hand written programs battled for virtual space [38]. Steen Rasmussen added mutations to a system based on Core War and studied interesting but short lived dynamics in his Coreworld system [131]. Tom Ray's Tierra fixed some of the shortcomings in Coreworld and was able to see interesting dynamics resulting from true Darwinian evolution such as the evolution of cheaters and then cheater-resistant organisms (he called these parasites in his original work, but they impose no direct cost to their victims) [134]. Avida added the ability to configure arbitrarily complex environments and sophisticated data tracking tools, which enabled its use as a research platform. Studies using Avida as a model system for experimental evolution have been well received, and topics such as the evolution of sexual recombination, epistasis, evolvability, complexity, and adaptive radiations have been published in top evolutionary biology journals [25, 96, 98, 113, 178].

The thought of studying host-parasite coevolution in Avida was immediately appealing to me. My interests were very clearly in biology, though my direction was less than predictable. I spent a year and a half exploring different directions without a clear desired outcome. Fortunately the host-parasite coevolution literature is extensive, and the fairly new field of eco-evolutionary dynamics grabbed my attention [144]. There is no better example of how ecology and evolution interact than host-parasite or predator-prey coevolution [79]. The host population is a major part of the parasites' ecology, and evolution in the parasite population has immediate consequences on the hosts' ecology [15]. Some of the first published examples of "Rapid Evolution", where

evolution changes ecological dynamics on an ecological time scale, were of model predator-prey systems [41, 43, 111, 183]. Since then, more and more studies have documented evolution occurring in ecologically relevant timespans [5, 24]. Hairston *et al.* used a novel statistical approach to partition the effects of evolution from those of ecology in a theoretical predator-prey model, the Grant's longterm data set on Darwin's finches, and the life history evolution of freshwater copepods [64]. In every case, evolution substantially affected ecology.

Diversity is often a major focus in ecological research, yet evolution is responsible for its creation and occasionally its destruction. Host-parasite coevolution is one mechanism often cited as creating diversity, and there are many empirical and theoretical studies confirming this hypothesis [54, 66, 106]. Once I realized that diverse communities of interacting species were potential outcomes of host-parasite coevolution, I began trying to understand how coevolution continues embedded within a community context. There has been a recent push to bridge community ecology and evolutionary biology, insisting that neither one can be understood without the other [80]. Historically, coevolution was thought of as the pair-wise coadaptation between two lineages, but Daniel H. Janzen wrote in 1979 that coevolution could also be "diffuse", where adaptation is driven by "an array of populations that generate a selective pressure as a group" [34, 78]. Many examples of diffuse coevolution come from the plant biology literature, where most focus on evolutionary responses to a community of herbivores [65, 155]. Several studies have found evidence that the presence of an additional species alters patterns of focal trait evolution, which supports the importance of diffuse coevolution [13, 77].

Around the same time I became interested in the role of community ecology in coevolution, I started collaborating with Miguel Fortuna, who was then a postdoc at Princeton working with Simon Levin. Miguel's work was primarily on network analysis of mutualistic and antagonistic communities, but he became interested in Avida because one could study how coevolution creates and shapes the networks of interacting species in perfect detail. Network analysis became a buzz word in the early 2000s, with small-world and scale-free networks showing up everywhere anyone looked for them (e.g., airplane networks, the social web, metabolic networks, etc.) [18].

However, Joshua Weitz pointed out that many of these studies are interesting to networkologists, but perhaps not to biologists, since general patterns do not necessarily suggest general processes [176]. Fortunately, the history of food webs in ecology predates this network analysis boom, and there has been real biological insight gained from networkology [10, 26]. One of the most intriguing results, though it potentially falls victim to Wietz's criticisms, is the pervasiveness of nested communities in host-parasite and host-mutualist communities [11, 50]. A recent meta-analysis of bacteria-phage assemblages found that 27 out of 38 analyzed communities obtained from the lab, the field, and even dairy plants, were significantly nested [46]. The hypotheses about why nestedness is such a pervasive pattern span the range from molecular details about interactions to the stability of overall communities and remains an open problem for research in ecological networks [12, 91].

Much of the biological literature surveyed here has hinted at how parasites, despite being harmful to their hosts, may promote properties that ecologists and evolutionary biologists consider beneficial: diversity, stable communities, and rapid evolution [75]. Parasites may also evolve lower virulence and thus a more benevolent interaction with their hosts. The old "conventional wisdom" was that parasites ought to evolve complete avirulence since their livelihood depends on interactions with hosts; however, a more integrated understanding of ecological and evolutionary dynamics has replaced old conventional wisdom with one that predicts the evolution of intermediate virulence [22, 95]. Parasites may even provide their hosts with benefits, and in some environments be considered mutualists [42]. For example, some fish infected with acanthocephalan parasites can thrive in higher levels of heavy metals than their "healthy" peers [156]. Examples of these conditional benefits are most abundant in microbial systems, where plasmids and phages are the offenders [137, 158]. Some of the most prevalent examples include parasitic plasmids that carry antibiotic resistance accessory genes and pathogenicity factors, which convert otherwise benevolent bacteria into disease causing agents [42, 100]. While these may not seem like conditional mutualists, they are from the bacterium's perspective. But why do these parasites evolve to be conditional mutualists? Does evolution or coevolution reinforce conditional mutualisms, or are they transient? Since microbial examples made up many relevant instances of conditional mutualism, Justin Meyer

and I started working together on developing a microbial system that could answer some of these questions.

This path took me from a computer scientist studying robots to an evolutionary ecologist with strong computational skills studying microbial evolution. However, I think many of the results I've obtained along the way are relevant to computational endeavors. Niching and fitness sharing methods attempt to overcome evolutionary computation's propensity to converge on local optima [143]. Multi-objective optimization is a relatively new approach that explicitly maintains a population along multi-dimensional Pareto frontiers to maintain diverse solutions [36]. More recently, the role of negative frequency-dependent selection has become appreciated in evolutionary computation, with its own catchphrase of "novelty search" [93]. These computational methods maintaining diversity necessarily ignore many ecological details. Adding ecologically mechanistic processes to evolutionary computation may lead to more complex and more diverse solutions [58]. Indeed, parasites were first used to solve engineering problems with evolutionary computation by Hillis in 1990 [72]. Hillis used groups of test-cases as the *parasites* that would exploit the weaknesses in solutions to engineering problems. Thus, parasites would increase in fitness as they found more and more cases where the candidate solutions would fail. Solutions would then continue to improve as parasites continued to exploit new weaknesses [72]. However, understanding more about the diversity created by coevolution, the structure of host-parasite communities, and the way coevolution and communities interact at multiple scales will certainly lead to new computational approaches.

My thesis includes several chapters highlighting work that moves along this path. First, we demonstrate that coevolution with parasites rapidly produces a diverse community of interacting hosts and parasites in Chapter 2. This chapter also describes an interesting interaction between the presence of mutations and parasites, suggesting that the ecological host-parasite interaction is altered by the presence of novel genetic variation. After showing that coevolution produces a complex community, Chapter 3 demonstrates how outcomes of coevolution can be dependent on the community context in which it occurs. Specifically, we show how antagonistic coevolution leads to

a trend of increasing complexity because the community of coexisting parasites collectively imposes selection for novel traits by maintaining a "population genetic memory" of previously evolved host resistance. Thus, complexity arises due to simple initial conditions, and continued selection for novel traits imposed by the parasite community. Chapter 4 provides an overview of how digital evolution can be used to study ecological networks, while Chapter 5 details our first set of analyses with these coevolved networks. Chapter 6 details lab work I have done investigating evolution along the parasitism-mutualism continuum using a temperate phage system. Finally, Chapter 7 provides a final synthesis of my work along this path, and outlines my future work on host-parasite coevolution in digital and microbial model systems.

Chapter 2

Coevolution of Diversity in a Digital Host-Parasite System

Authors: Luis Zaman, Suhas Devangam, and Charles Ofria [184].

2.1 Introduction

Theodosius Dobzhansky's now famous words, "Nothing makes sense in biology except in the light of evolution" are especially true for the problem of biodiversity [39]. Evolution is the process that shaped all of life, extant and extinct. However, biodiversity is typically thought of in an ecological framework, interested in how a static set of species with static interactions can be stably maintained. Evolution is instead reserved for grand scales, defined by Slobodkin as happening on the order of half-a-million years [151]. Contrary to this view, many studies have revealed substantial evolution occurring over very short time scales; this concept of rapid evolution is reviewed in [24, 56, 64, 144]. Sometimes evolution occurs on such short time scales that it has significant ecological effects on communities [111, 183]. This new view of eco-evolutionary dynamics means the problem of biodiversity must include the feedbacks between ecology and evolution to understand how diversity is maintained in communities [161].

Parasitic species are paramount to biodiversity, where nearly half of all known species classified

8

as parasites [129, 130]. They are not only incredibly diverse and successful, but parasites have been shown to increase host diversity theoretically and experimentally [14, 21, 40, 41, 147]. Similarly, the escape from natural parasites that would otherwise limit growth is a leading hypothesis for why some introduced species become invasive [167]. For these and many other reasons, parasites can be used counterintuitively as indicators of ecosystem *health*, where communities with a high diversity of hosts often have many parasites [75]. Another counterintuitive effect of host-parasite coevolution is the increase in protection against emerging diseases it provides. Host diversity is a key factor in mitigating novel pathogens and their potential for rapid evolution, but a major source of genetic diversity results from coevolution between hosts and parasites. Thus, in order to prevent emerging disease disasters, we must learn how to protect and foster the coevolution of hosts and their native parasites [8]. Antagonistic coevolution also offers a perfect scenario for studying the effects of rapid evolution, since ecological feedbacks are inherent in their coevolutionary dynamics [15].

We are generally interested in how rapid coevolution creates and maintains diversity in host-parasite communities, and specifically interested in how novel variation affects the resulting diversity. Even when studying rapid evolution, several generations can take years to observe and performing detailed experiments in a natural setting is infeasible. Thus, many examples of rapid evolution come from experimental evolution experiments using microbes [144]. However, even these microbial systems have drawbacks when studying rapid evolution, such as difficulties in assessing the entire population of interesting traits, and the inability to control for random events like mutations. In order to investigate the role of novel variation in host diversity, we turn to artificial communities.

To study the coevolution of host-parasite communities *in silico*, we used the Avida research platform [120, 121]. We implemented parasitic organisms and a mechanism for them to infect hosts based on genetically encoded phenotypes. We compared independent populations of digital organisms in the presence and absence of these parasites, as well in the presence and absence of novel variation created by random mutations. We found that hosts coevolving with parasites were more diverse than hosts evolving alone. This result held both in the presence and absence of novel variation. However, new variation increased diversity in host-parasite communities more than it did

for hosts in the absence of parasites, suggesting the importance of novel variation in maintaining stably diverse communities.

2.2 Materials and Methods

2.2.1 Avida

For all experiments in this chapter, we used Avida 2.13.0 r4173. Avida is a digital life research platform that maintains a population of self-replicating computer programs ("digital organisms"), which compete for resources. Digital organisms exist in a well-mixed environment, interacting randomly with any other organism in their world. Genomes consist of a circular list of instructions from a Turing complete programming language, executed on virtual hardware. Each instruction directs the organism to perform a simple operation such as arithmetic, flow control, or environmental interaction.

During replication, an organism loops through its genome copying each instruction sequentially until reaching its end. It then executes a `divide` instruction, separating off its offspring and placing it into a random cell in the world, replacing any previous occupant. Instead of always producing a perfect replica of the parent genome, the copying process is noisy and introduces errors. These mutations and can be insertions, deletions, or substitutions.

Organisms replicate by using their virtual CPU to execute an appropriate series of instructions. In these experiments, there is a single type of resource that must be metabolized for successful replication. To metabolize a portion of the available resource, organisms perform logical tasks on environmental inputs. The default Avida environment contains nine logic tasks. In these experiments, all nine default tasks are available, but they metabolize the same resource assuming there is a sufficient quantity available (See section 2.2.3).

Organisms in Avida possess one of several virtual hardware types, varying in instruction set and architecture. In the hardware type we used for this study, organisms have: four stacks to store and manipulate numerical values, a set of genome memory spaces in which organisms execute and copy

instructions, and a set of heads that point to positions in each memory space. Organisms identify a specific stack, memory space, or head with a label consisting of no-operation instructions ("nops").

Since Avida has heritable variation, and environmentally driven selection, evolutionary dynamics are a natural byproduct of the system [30]. For this reason, Avida has been used successfully to understand ecological and evolutionary dynamics, as well as to perform more applied research in distributed systems and software engineering. For a detailed introduction to Avida, see [120, 121].
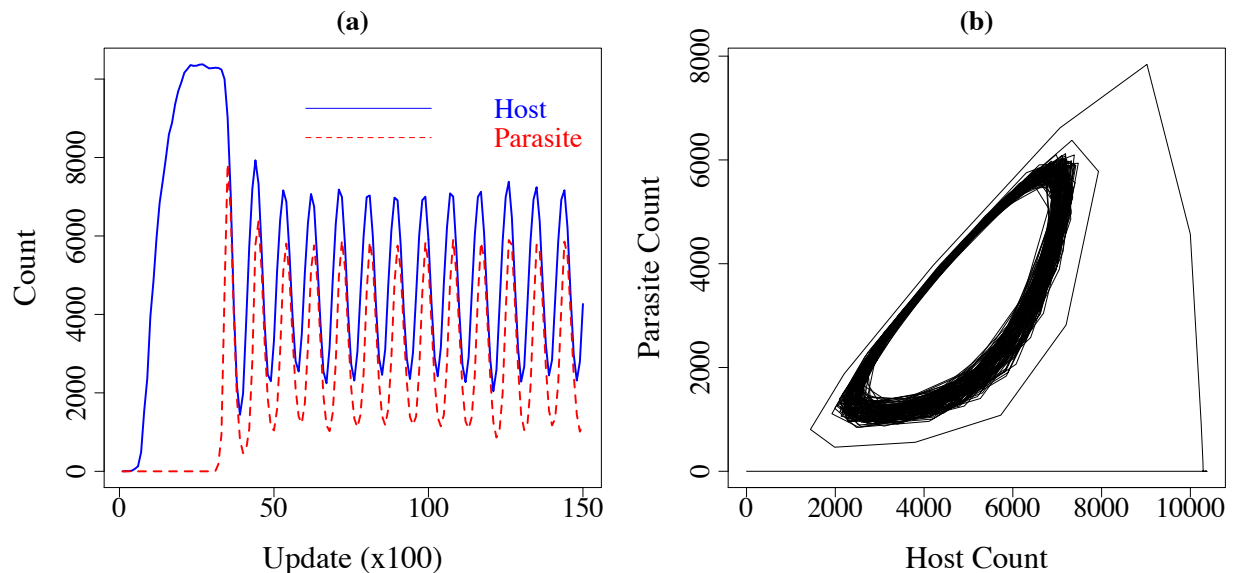
Figure 2.1: Host-Parasite ecological dynamics when parasites consume all of their host's resources (virulence of one). (a) shows host and parasite frequencies through time, it shows only the first 15,000 updates for clarity. (b) is a phase plane including data from all 200,000 updates. Both plots demonstrate classic Lotka-Volterra dynamics with phase shifted oscillations and a limit cycle in phase space.

2.2.2 Parasites

Parasitic digital organisms are self-replicators that operate inside hosts, relying on them to provide energy in the form of CPU cycles. Note that these parasites are distinct from those in the Tierra system, which operated independently of, and did not directly harm, their hosts [87, 134]. Instead of executing a `divide` instruction to finish replicating, a parasite must inject its offspring into a host. When the `inject` instruction is executed, the parasite offspring attempts to infect the organism

in a randomly chosen location. If successful, the new parasite is treated like a thread in the host organism, consuming CPU cycles and thus reducing it's host's fitness. Infection is successful if any of the logical tasks performed by the parasite match any of the tasks the host is performing. Infection will fail if there is no overlap in tasks, if the chosen location is empty, or if the organism is already infected. The probability of a parasite steeling a CPU cycle from its host is configurable, and we will refer to it as "virulence". When virulence is set to one, parasites steal all CPU cycles from their hosts, killing them and using them for energy like predators. Indeed, when observing the ecological dynamics of parasites with maximal virulence, we find classic Lotka-Volterra dynamics (Figure 2.1) [85]. When virulence is set to 0.5, parasites and hosts split CPU cycles evenly and there is a stable equilibrium where hosts and parasites coexist.

Hosts and parasites in Avida are similar to *E. coli* and lambda phages. These phages must attach to receptors on the surface of bacteria in order to infect, and the bacteria must have receptors in order to consume resources from the environment. However, consuming resources leaves them susceptible to phages. The bacteria evolve resistance by changing their surface receptors, but lambda phages can counter resistance by evolving their tail fibers to attach to these new proteins [90]. Similarly, in Avida, hosts must perform logic tasks to consume resources and thus replicate, but this action leaves them susceptible to infection. Resistance can evolve by changing the logic task(s) used to consume resources, but the parasites can counter adapt by evolving the ability to perform the new task. Figure 2.2 depicts the mechanics of infection in Avida.

2.2.3 Configuration

All experimental runs were done in a well-mixed environment, where host and parasite offspring were randomly placed in the world. Each run started with a 320-instruction-long host organism capable of performing only the NOT task and self-reproduction. In runs with parasites, after 3,000 updates 400 cells in the world were exposed to 80-instruction-long ancestral parasites capable of performing only NOT and self-reproduction. The parasites in these experiments had a virulence of 0.80 unless otherwise noted. In order to become resistant, hosts must lose their ancestral NOT
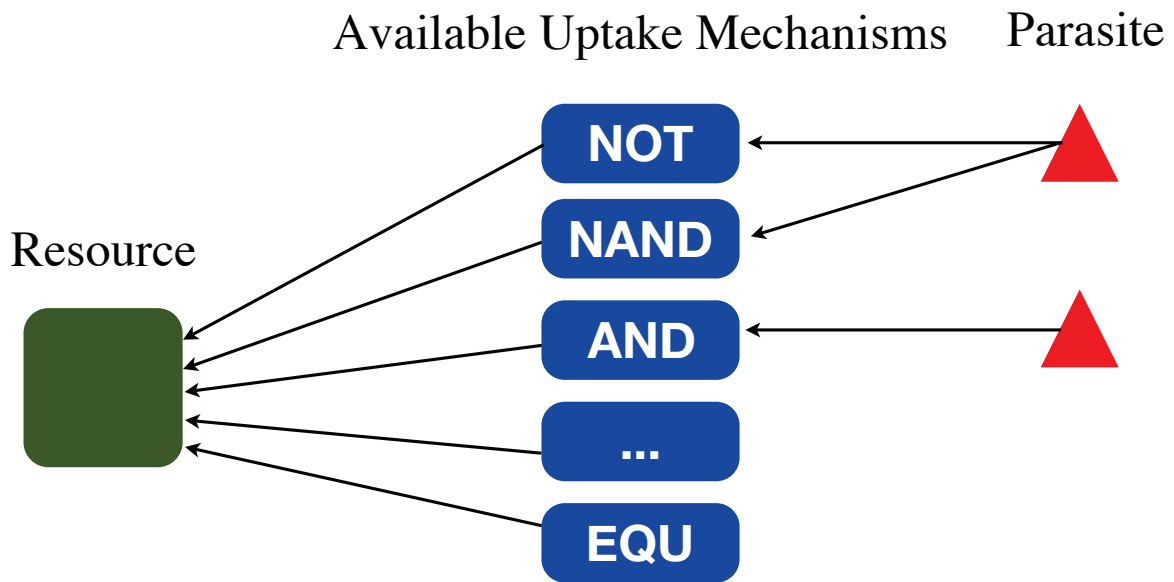
Figure 2.2: A diagram of the traits governing host and parasite interactions. The single resource type in the environment that must be consumed for successful host replication is depicted by the green square. Hosts can use any of the nine default logic tasks, indicated in blue, to consume part of this resource if it is available. Parasites, depicted as red triangles, target the mechanism hosts use to consume resources - the logic tasks.
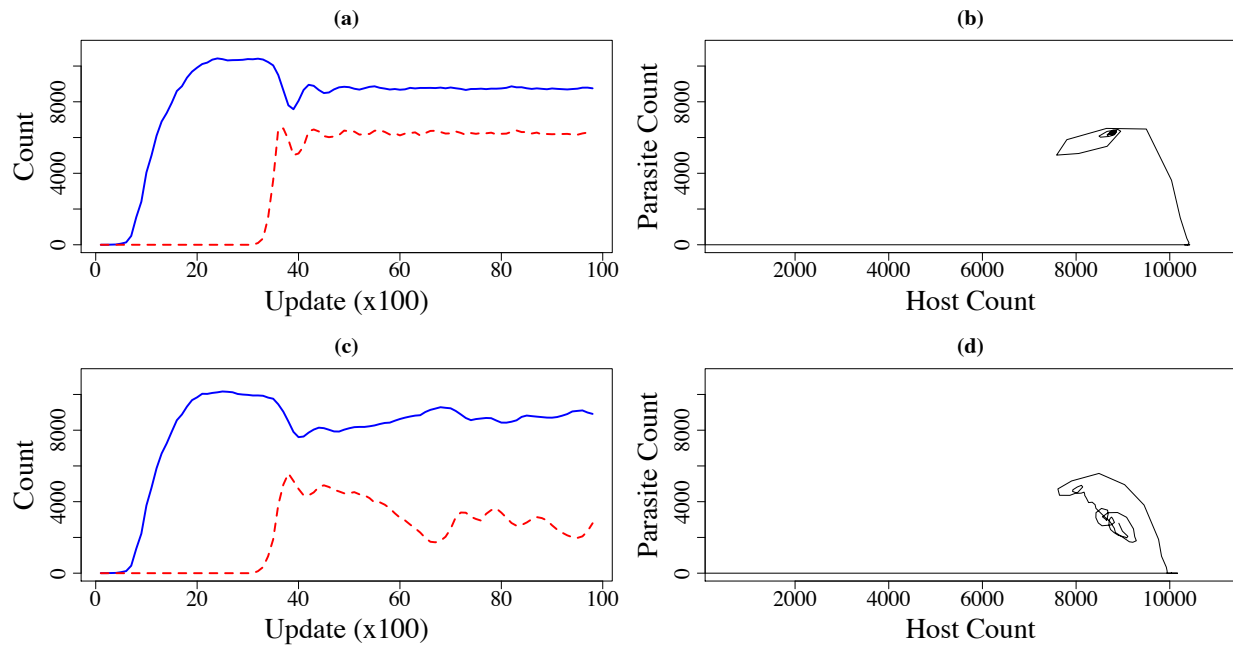
Figure 2.3: Eco-evolutionary dynamics of host and parasite (blue solid and red dashed respectively) interactions. (a) and (b) are respectively a typical frequency plot and phase plane of host-parasite interactions in the absence of evolution - the ecological dynamics. (c) and (d) are the otherwise identical, except that mutations are allowed, and thus evolution can occur. Both subfigures are of approximately 100 generations, representing an ecological time scale. These figures demonstrate the disruption of typical ecological dynamics (compare (a) to (c) and (b) to (d)) by evolution on ecologically relevant time scales.

function (so that the parasites cannot infect them) while also evolving a novel task (so that they can continue to collect the resources required for replication). Similarly, in order for parasites to infect hosts that evolve resistance, they must also evolve the novel task. These ancestral hosts and parasites were capable of performing only the most basic task in the environment and self-replicating, the rest of their genomes were padded with no-operation instructions. Each run was allowed to execute for 200,000 updates, where one update is the amount of CPU time needed for each organism in the population to execute an average of 30 genomic instructions.

We disallowed multiple infection by setting the maximum number of threads an organism can have to two. Organisms were not given access to instructions manipulating or creating their own threads, preventing host organisms from becoming resistant by simply creating additional threads. We also disallowed vertical transmission, the direct inheritance of parental parasites, by clearing infections on successful host division. Only allowing horizontal transmision prohibited the association between parasite reproduction and host reproduction, since we consider cell devision the birth of two daughter-cells.

Novel variation could take the form of point, insertion, and deletion mutations. These mutations were applied per site, and are split 10:1 point mutations to insertions/deletions. The individual rates were set so that, on average, hosts had a single mutation every four offspring, and parasites had a single mutation every two offspring.

There was a single resource necessary for successful host replication. This resource was kept at a low level to prevent the world from filling: there were 14,400 potential locations for organisms and approximately 10,000 were filled in the absence of parasites. For a host to have successfully consumed this resource, there must have been a sufficient quantity in the environment. For these experiments we required two units be available, and one unit was consumed. If a host did not successfully consume resources before executing the `divide` instruction, replication failed and the organism began execution again without producing any offspring. Eventually, organisms died of old age if they did not successfully divide (since asexual reproduction produced two daughter cells, age was reset on successful division) before reaching the maximum age of $30 \times$ genome length.

15

Cell death also occurred as a result of offspring being randomly placed in the world. In this case, the occupant was overwritten by the newly divided cell.

2.2.4 Measuring Diversity

We measured diversity as the Shannon diversity index (H) of binary phenotypes. That is, we looked only at whether each of the nine tasks was performed or not without accounting for expression level and considered each unique binary string a different phenotype. Thus, the maximum number of phenotypes possible in an environment containing nine tasks is 2^9. To calculate diversity, we used equation 2.1

$$H = - \sum_{i=1}^{S} p_i \ln p_i \tag{2.1}$$

where S was the total number of phenotypes, and p_i was the proportion of phenotype i in the population. This metric is optimized when both species richness and evenness are maximized.

2.2.5 Measuring the Effect of Parasites on Host Diversity

To compare the overall effect of parasites on host diversity, we ran 50 replicate populations where we introduced parasites and 50 replicates where we did not. We then measured the Shannon diversity index of the final set of hosts as described in Section 2.2.4. The difference in host diversity between these two treatments was the effect coevolution with parasites had in these communities.

We also measured how parasites influenced diversity in *ecological* communities, where no new variation was being introduced, by running 50 replicate populations that contained parasites, and 50 replicates that did not. All runs (co)evolved for 100,000 updates, then we disallowed mutations and the runs continued for an additional 100,000 updates to settle into an *equilibrium*. We then compared the Shannon diversity index of hosts from the resulting communities in these two treatments to quantify the parasites' *ecological* contribution to diversity.

Note that both equilibrium and ecological are misnomers, though in this case they are the most

proximate terms. Parasites are frequently in a state of non-equilibrium, even in ecological contexts, and populations are technically evolving if there are changes in mean phenotype over time – which can happen in a community context even in the absence of new variation.

2.2.6   Measuring the Effect of Novel Variation on Host Diversity

To control for novel variation in host-parasite communities, we harnessed the repeatability of independent evolutionary runs in Avida. Essentially, we asked what would have happened if we went back in time and allowed the communities to continue mutating by having ensured identical coevolutionary histories. We determined the effect of novel variation by measuring the difference in host diversity between paired runs where novel variation was continued versus when it was stopped.

Unfortunately, there are confounding effects when measuring the contribution of novel variation this way, since new variation is obviously a source of diversity. To correct for this effect, we compared the increase in diversity observed in communities with parasites to the level in communities without parasites (also pairing runs of only hosts as described above). If diversity increases further with parasites than in hosts evolving alone, we know that the effect of novel variation in communities with parasites is not due to the trivial additional diversity new variation brings about. Additionally, the new variation in communities without parasites can still have some influence on diversity above the trivial effect, but these analyses are conservative with respect to the actual contribution of novel variation.

2.3   Results

The signature of rapid evolution is the disruption of typical ecological dynamics. Figure 2.3(a) and 2.3(b) show the frequency and phase plane plots for a typical host-parasite community without evolution, describing the pure ecological dynamics for this system over approximately 100 generations. Figure 2.3(c) and 2.3(d) depict the dynamics in the presence of novel variation and thus evolution. If evolution occurs on different time scales than ecology, we would not expect differences over such
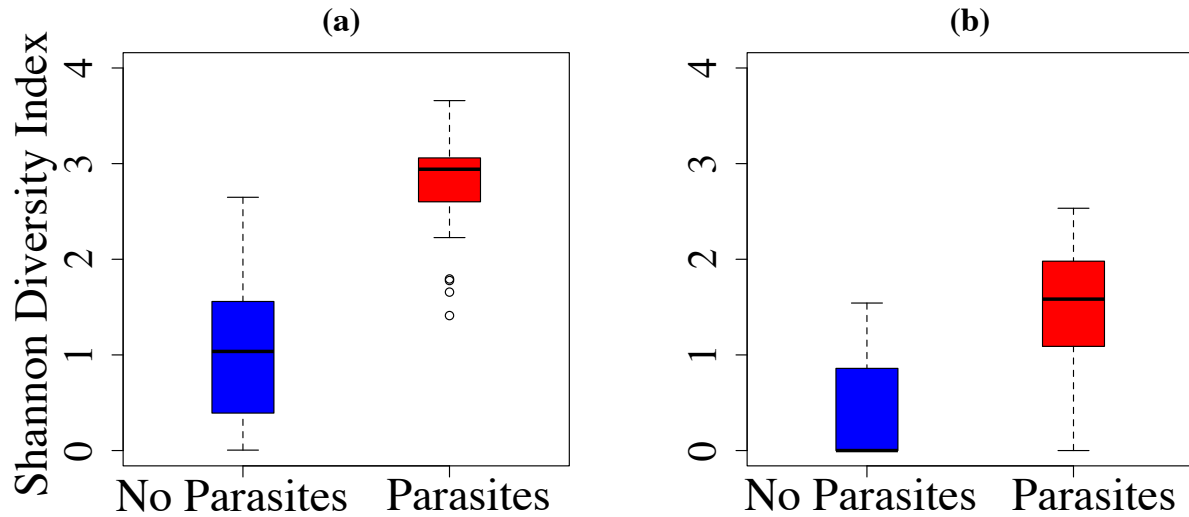
Figure 2.4: Host diversity in runs that evolved without parasites compared against host diversity in runs that coevolved with parasites. (a) depicts host diversity when all 200,000 updates had mutations, and (b) depicts host diversity when mutations were stopped at 100,000 updates. Thus, (b) shows the *ecological* effects parasites have on host diversity.

few generations. However, comparing 2.3(a) to 2.3(c) and 2.3(b) to 2.3(d) shows a pronounced difference in community dynamics, suggesting that coevolution occurs on ecological time scales in this system.

2.3.1 Effect of Parasites

When considering any replicate runs with parasites, we removed communities that lost parasites in either the presence or absence of mutations from analysis. In other words, to be considered in the analysis, parasites had to persist in both the 50 original runs, as well as the 50 replays where mutations were continued. Only one of the paired-runs withheld from analysis came from a community that maintained parasites in the absence of novel variation, but lost them when the runs were replayed with mutations. A single community lost parasites after novel variation was removed, but they were also lost when mutations were continued. Thus, the loss of parasites in this case was not due to instability after stoping new variation. Seven other communities lost parasites prior to losing novel variation. These nine runs, as well as their paired replayed runs were removed

from analysis (a total of 18/100 runs were withheld). Coevolved communities were able to maintain parasites robustly in the absence of novel variation.

Figure 2.4(a) depicts the Shannon diversity distributions of hosts in communities with and without parasites. Communities with parasites had significantly more diversity (Mann-Whitney U = 1996, $p \ll 0.001$). The presence of parasites resulted in an increase of host Shannon diversity by 1.784 with a 95% confidence interval of [1.506, 2.063].

To measure the *ecological* effects parasites had on host diversity, we removed the possibility for novel mutations and measured the resulting communities' host diversity. Figure 2.4(b) depicts the Shannon diversity distributions of communities evolved with and without parasites after 100,000 updates in the absense of mutations. Again, communities with parasites had significantly higher diversity than those without parasites (Mann-Whitney U = 1816, $p \ll 0.001$). In these *ecological* communities, parasites increased the Shannon diversity by 1.15 with a 95% confidence interval of [0.933, 1.434].

2.3.2 Effect of Novel Variation

The pairwise subtraction of host diversity between runs where novel variation was stopped at 100,000 updates from the runs where mutations continued produced the distribution of increases in Shannon diversity that would have occurred had the *ecological* community not lost their sources of novel variation. We paired these runs since the communities at 100,000 updates were identical, and thus the only difference was whether or not mutations continued. Figure 2.5 depicts the distribution of increases in host diversity due to novel variation in communities with and without parasites.

The statistical difference between increases in diversity with and without parasites is a measure of how novel variation affected host diversity in the presence of parasites above and beyond the trivial effects of new mutations. There was a significant difference between these two distributions, where communities with parasites had a 0.652 increase in host diversity with a 95% confidence interval of [0.321, 0.973] (Mann-Whitney U = 1508, $p = 0.00012$).
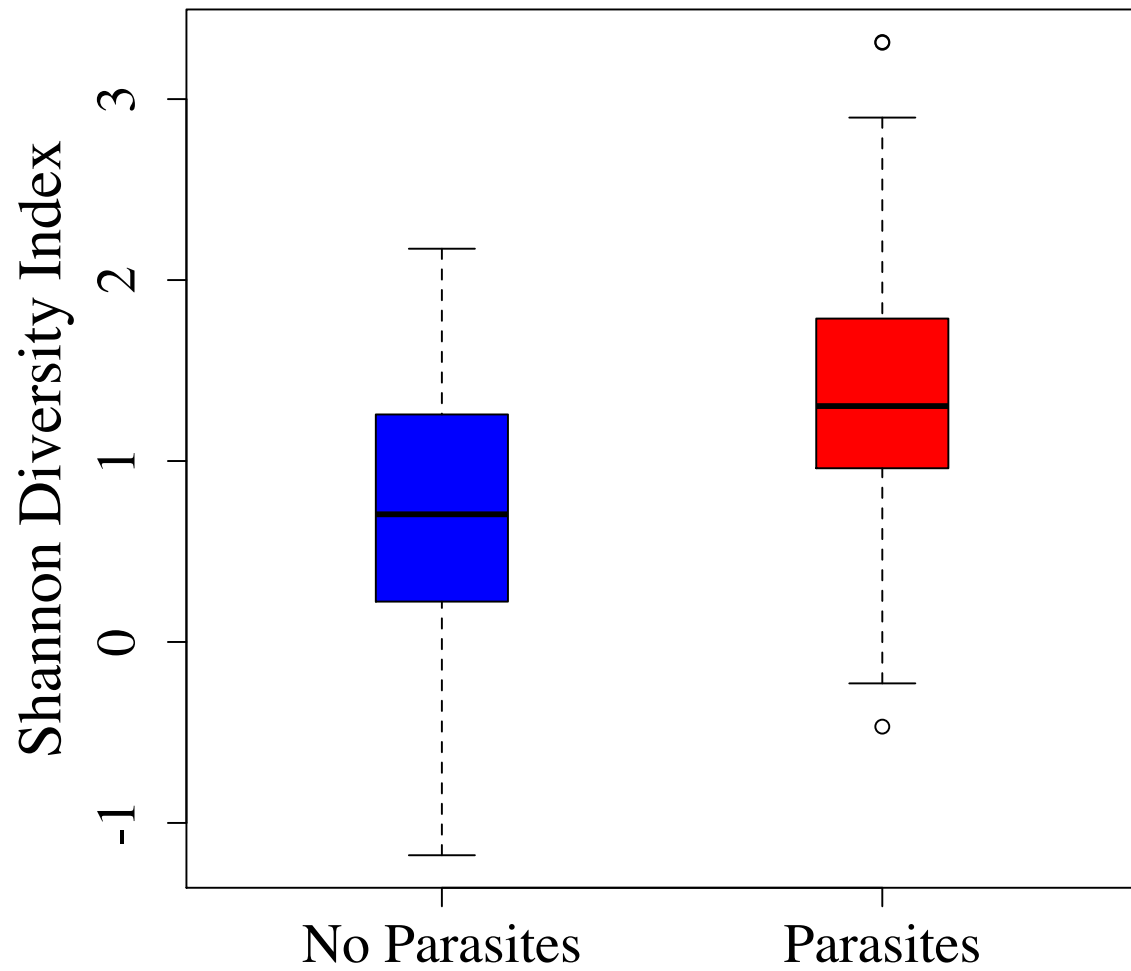
Figure 2.5: Increase in diversity when runs that lost novel variation were replayed with continued mutations in communities with and without parasites. Runs with parasites had significantly larger increases in host Shannon diversity in the presence of mutations than runs without parasites.

(a) Host Tasks

(b) Parasite Tasks

(c) Host/Parasite Counts
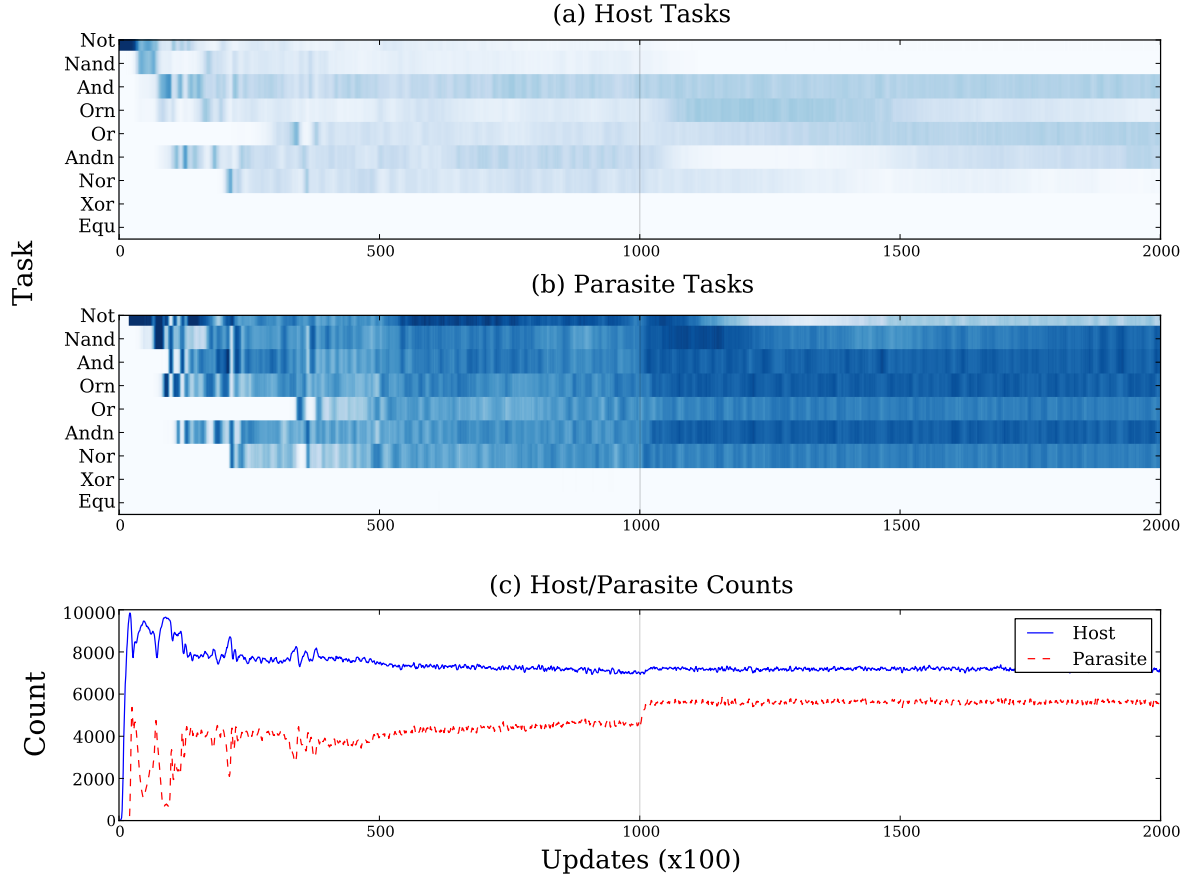
Figure 2.6: Frequencies of phenotypic traits in hosts and parasites for a sample coevolutionary community where mutations stoped at 100,000 updates (grey line). (a) depicts the relative frequencies of traits hosts used to consume resources in the community through time, while (b) depicts the relative trait frequencies parasites used to infect hosts. Parasites tracking of host phenotypes is apparent by the similarity between the phenotype heat maps. (c) depicts the number of hosts and parasites through time.

2.4 Discussion

Evidence of evolution happening rapidly enough to co-occur with and influence ecological dynamics is now widely documented [24]. Host-parasite coevolution is intimately connected to ecological dynamics, and thus a likely candidate for these eco-evolutionary feedbacks. Despite the large amount of evidence showing ecological and evolutionary dynamics interacting, little is known about how important this feedback is for maintaining communities. Novel variation is only one aspect of rapid evolution, but it is an important one nearly impossible to test in laboratory or natural settings.

We have presented an *in silico* instance of coevolution and demonstrated ecological and evolutionary dynamics interacting on similar time scales. If ecological processes were happening much faster than evolution, we would not expect the rapid disruption of typical community behavior when we introduce novel genetic variation. However, as Figure 2.3 depicts, novel variation did indeed disrupt the ecological dynamics. To understand the effect that this eco-evolutionary feedback had on host diversity in a community context, we first quantified the effects of parasites.

Regardless of whether there was a source of novel variation, parasite presence significantly increased host diversity in this system, consistent with empirical and theoretical results [14,21,40,41]. There are multiple, non-exclusive mechanisms by which parasites can increase host diversity (see discussion in [21]). Parasites may target the most frequent host phenotype, and the hosts would then evolve resistance against these parasites. However, as the new resistant host increases in frequency, parasites would experience selection to target it. Thus, parasite-imposed negative frequency-dependent selection can maintain diversity by keeping the frequency of any one particular host genotype (or phenotype) at bay. Alternatively, hosts may experience trade-offs between resistance and competitive ability allowing coexistence of sensitive and resistant types. When parasites are common, resistance may be worth the cost it carries; but at low parasite densities, the cost of resistance may become a burden. We plan to disentangle these mechanisms maintaining host diversity in the future.

The stability of these coevolved host-parasite communities is surprising. Of all 50 replicate communities that experience the loss of new mutations, only one completely excludes parasites

after losing novel variation (a handful eliminated parasites prior to the loss of new variation). In other words, 41 out of 50 host-parasite communities were able to persist both in the presence and absence of continuing novel variation.

Figure 2.6 depicts a single run where all mutations stopped at 100,000 updates. Moving from left to right on the figure represents going forward in time, and the heat maps show the relative frequencies of each task performed by hosts (a) and by parasites (b). In the first 100,000 updates, there was rapid change in host and parasite phenotypes, but as time goes on the variation saturates. This saturation is also evident in 2.6(c), which depicts host and parasite frequencies. Interestingly, after mutations stopped, host and parasite frequencies appeared to reach an *equilibrium*, but phenotypes still changed through time. Additionally, it is clear that neither hosts nor parasites collapsed into just one or two phenotypes, rather community diversity persisted.

Having shown that parasites increased host diversity both in the presence and absence of novel variation, we aimed to distinguish the effects that mutations had on diversity in host-parasite communities. Novel variation trivially increased diversity, since it often produced new phenotypes. However, host-parasite communities could have been affected by this novel variation differently than hosts alone. By asking what would have happened had we not stopped novel variation at 100,000 updates in coevolved host-parasite communities, we effectively *replayed the tape* [60], but this time allowing for continued mutations. This ability to replay the tape enabled us to measure the actual increase in host diversity had novel variation continued. It is important to note that the measured increase is the actual increase in diversity rather than an expected increase if mutations continued, since we guaranteed identical coevolutionary history. We also measured the effect of novel variation in communities without parasites to estimate the trivial effects of mutations on diversity. Figure 2.5 depicts the distribution of actual increases in diversity had mutations continued for the full 200,000 updates. We measured the statistical difference between these two distributions, and conservatively called this value the non-trivial effect novel variation had on host-parasite communities (see Section 2.3.2). Since the difference was significant, novel variation had more than a trivial effect on host-parasite community dynamics.
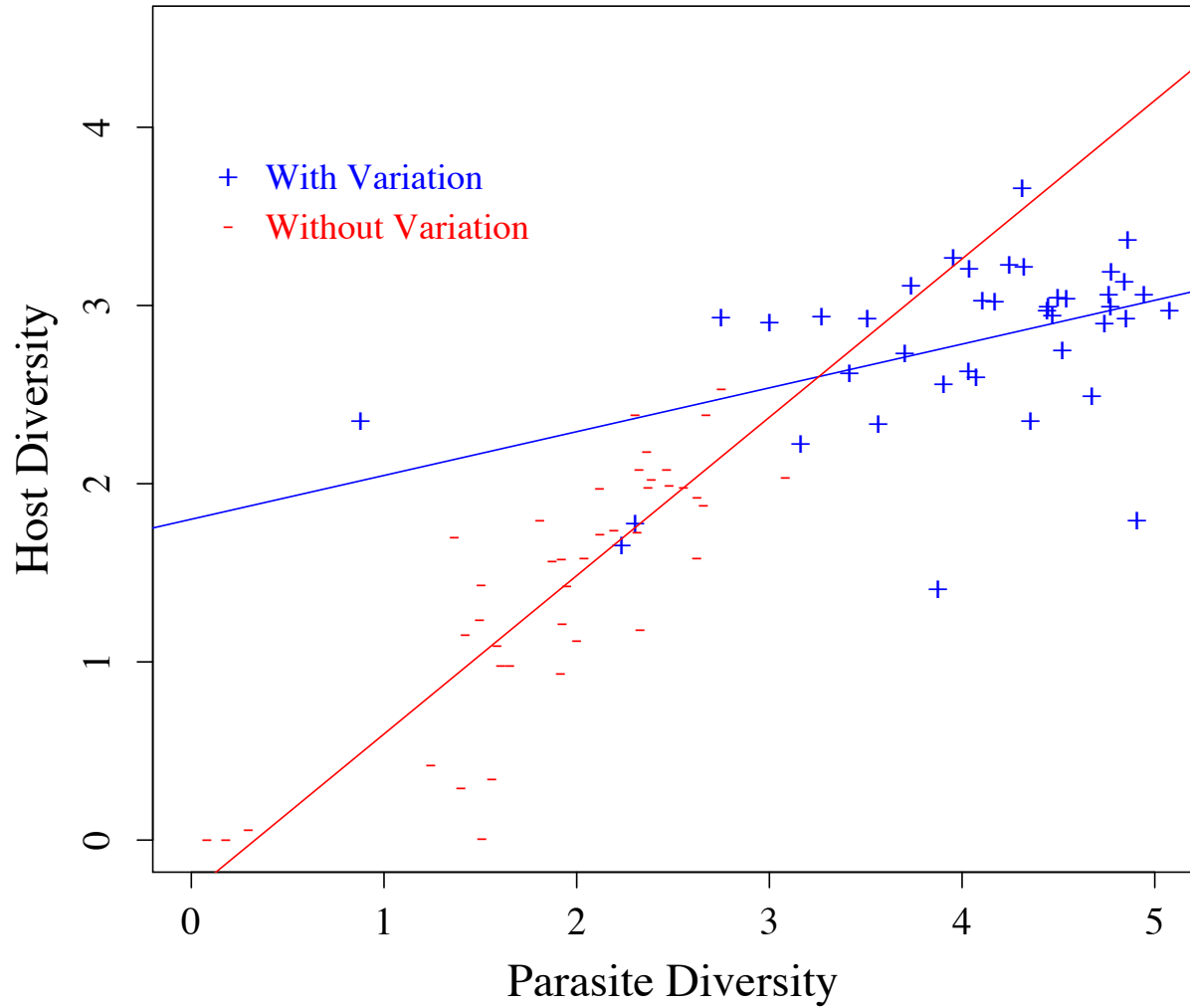
Figure 2.7: Relationship between host diversity and parasite diversity across 41 communities from each treatment. Points with red "-" signs are from runs where mutations stopped at 100,000 updates, and points with blue "+" signs are from replayed runs with continued mutations. There is a strong relationship between host and parasite diversity in runs without new variation for 100,000 updates. On the other hand, there is a weak relationship in paired runs with continued novel variation, suggesting other factors contributing to diversity in the presence of mutations.

Looking at how host diversity varies with parasite diversity between runs that experienced continued mutations and those without continued novel variation further suggests the interaction of new variation on community dynamics. Figure 2.7 is a plot of host diversity versus parasite diversity, and the linear regression for the two treatments of novel variation. The relationship between host and parasite diversity in runs without novel variation for 100,000 updates was strong (Adjusted $R^2 = 0.70$). On the other hand, when novel variation was continued throughout the run, the relationship was much weaker (Adjusted $R^2 = 0.18$). The amount of variation unexplained by the relationship of host and parasite diversity in runs that continued to experience novel variation suggests that additional community or evolutionary dynamics are influencing host and parasite diversity. The large amount of explanatory power the relationship has in the absence of novel variation adds support to this view. Understanding the mechanisms acting on this variation to produce non-trivial increases in host diversity will shed light on the important eco-evolutionary interactions shaping community dynamics [144].

Chapter 3

Coevolution Drives the Emergence of Complex Traits and Promotes Evolvability

Authors: Luis Zaman, Justin R. Meyer, Suhas Devangam, David M. Bryson, Richard E. Lenski, Charles Ofria

3.1   Introduction

Life emerged on Earth some four billion years ago and has evolved increasingly complex traits, including intricate biochemical pathways, elaborate developmental networks, and powerful neural architectures [146, 157]. However, the processes responsible for promoting this complexity remain poorly understood [33, 34, 62, 76, 104, 145, 146, 150, 157]. Is adaptation by natural selection largely responsible for this complexity and, if so, what is the nature of that selection? Or is this apparent trend an artifact that reflects the initial conditions and lower bounds to complexity? Given the limitations of historical data for answering these questions, experimental evolution offers an alternative approach to explore these issues and test specific hypotheses. However, the emergence of complexity in nature is a slow process, one not readily replicated in the laboratory [34, 68]; and without an objective way to measure the complexity of organismal traits [1, 2], rhetorical arguments may obscure and delay empirical research on this fundamental problem.

26

Fortunately, computational approaches have advanced beyond traditional numerical simulations, and it is now possible to test evolutionary hypotheses by running experiments with computer programs that self-replicate, mutate, compete, and evolve [126]. In one study, Lenski et al. [98] used the Avida [120] system to examine the role of selection for intermediate steps along many evolutionary paths to a particularly complex trait, the EQUALS (EQU) logic function. Because Avida is computational, the authors could readily observe changes over thousands of generations; moreover, the complexity of traits could be objectively quantified as the number of building blocks (in this case, NAND instructions) required for their execution. By allowing initially identical populations to evolve in different environments, Lenski et al. demonstrated that the most complex traits emerged only when simpler functions were also selectively favored, which promoted the accumulation of the necessary building blocks [98].

Here we use this system to ask whether coevolution–specifically, parasite-host interactions–can drive complexity to higher levels than would otherwise be achieved. Several authors, including Dawkins and Krebs [33] and Vermeij [171], have proposed that coevolutionary arms races" lead to increased complexity as adaptations and counter-adaptations favor more and more extreme traits [62]. Indeed, we show that host-parasite coevolution produced substantially more complex host traits than did evolution in the absence of parasites. Moreover, we show that this complexity arose in the evolving computer programs, in part, by an unexpected process: selection for increased evolvability, which was achieved by genetic mechanisms reminiscent of so-called contingency loci" that are found in many pathogenic bacteria [117].

In Avida, both host and parasite organisms are self-replicating programs that must expend CPU cycles to execute instructions in their genomes [184]. The genetic instruction set includes basic arithmetic and input/output operations as well as operations that allow storage and manipulation of binary numbers in temporary memory via a set of stacks. Coordinated execution of appropriate sets of instructions allows organisms to obtain resources (in the case of hosts) or infect hosts (in the case of parasites) and copy their genomes instruction-by-instruction to produce offspring. The copying process occasionally introduces mutations including point mutations, insertions, and deletions that

may affect the progenys phenotype. As in nature, most mutations are deleterious or neutral, but occasional beneficial mutations improve an organisms ability to acquire resources, infect hosts or resist parasites, or reproduce. These benefits may enable genotypes to increase in frequency as they displace less fit conspecifics because of their faster acquisition and more efficient use of CPU cycles. Thus, populations of digital organisms, like their counterparts in nature, typically evolve to better fit their environments [126].

Figure 3.1 shows a schematic overview of the relationships between hosts, functions, resources, and parasites in our experiments. Hosts obtain the resources necessary for their reproduction by performing one or more logic functions, but those functions also make the host vulnerable to infection by a parasite that can perform the same function. Thus, an infection can occur only if a particular host and parasite share at least one function, although the specific genetic encoding that a host and parasite employ to perform that function rarely, if ever, correspond at the sequence level. After a successful infection, the parasite acquires 80% of the infected hosts CPU cycles, which the parasite uses to execute and copy its own genome, while imposing a severe cost on the host. As a consequence, coevolution occurs when hosts and parasites acquire and lose functions.

The experimental configuration allowed for nine different logic functions, which require varying numbers of `NAND` instructions to be executed with the proper inputs used for each; `NAND` is the only logic function available in the genetic instruction set. The minimum number of `NAND`s required for each function's performance is known and provides a simple, objective measure of the complexity of that function [98]. The most complex function, `EQU`, requires five `NAND`s, and the shortest program that can perform `EQU` requires nearly 20 precisely interacting instructions, although there are many longer programs that also encode `EQU` [98]. In the absence of parasites, a previous study found that 23 of 50 populations evolved the ability to perform `EQU` when the other eight functions were rewarded with additional CPU cycles that increased with their complexity (i.e., minimum required `NAND`s), thus allowing essential building blocks to accumulate in the evolving genomes [98]. Here, we test whether host-parasite coevolution can drive increased complexity without explicitly rewarding building blocks. To that end, we ran similar experiments except with coevolving parasites

28
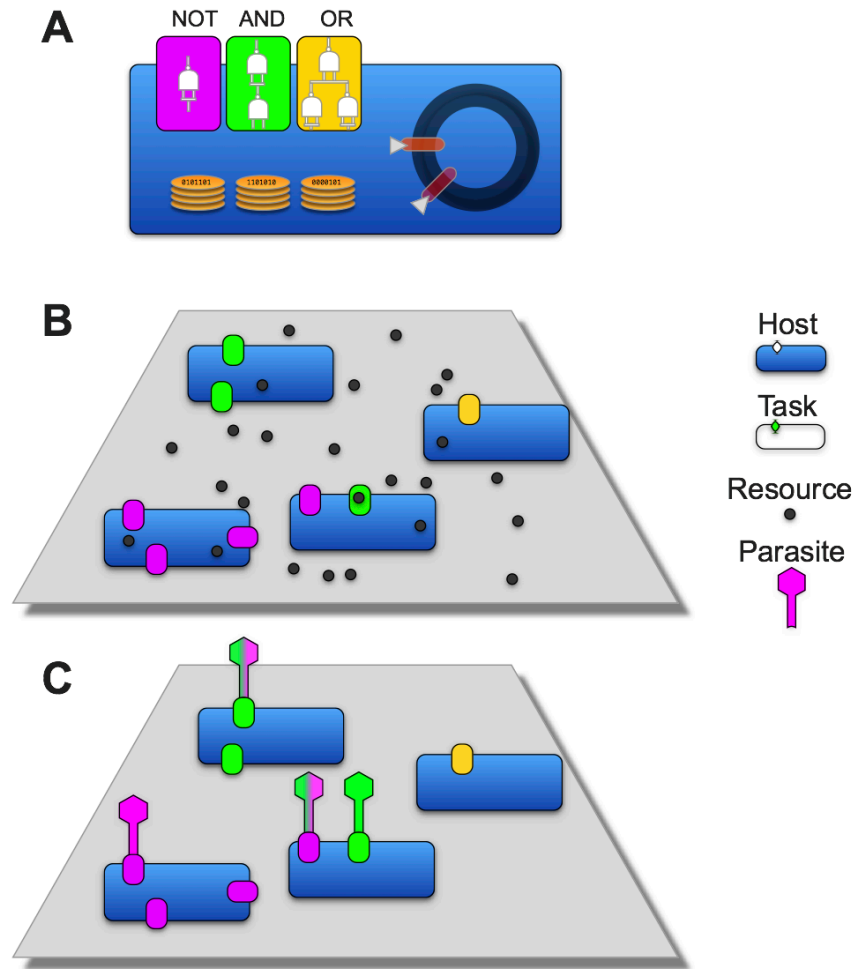
Figure 3.1: Hosts, parasites, functions and resources in Avida. (A) A host organism with stacks used to store binary values, a circular genome with pointers used to execute its code, and three functions NOT, AND, and OR shown in different colors. Functions vary in complexity as measured by the number of NAND gates (shown as 1, 2, and 3 logic gates within the respective colored function circuits) required to perform them. (B), These functions enable organisms to take up resources from their environment. (C), Parasites target the resource-uptake mechanisms of the hosts in this system by performing the corresponding function. Note that some parasites can perform multiple functions (shown by multiple colors) and thus infect hosts via multiple uptake systems. When a parasite infects a host, it acquires a portion of the hosts CPU cycles. Executing a single operation costs an organism a single CPU cycle. Time in these experiments is measured in "updates", which corresponds to a per capita average of 30 executed CPU cycles.

in one-half of the replicates and without the progressive reward structure used in the previous work.

3.2 Results and Discussion

3.2.1 Parasites Drive Greater Host Complexity

Figure 3.2 shows that coevolution with parasites drove host populations to evolve more complex functions in order to obtain the resources necessary for their replication, without any greater reward for performing the more difficult functions. Host complexity increased in both the presence (red) and absence (blue) of parasites, but it did so much faster and reached much higher levels in the coevolution treatment ($p \ll 0.001$, Mann-Whitney $U = 2304$). The effect of parasites on the rise of complexity is exemplified by `EQU`, the most complex function; the ability to perform `EQU` evolved in 17/50 host populations that coevolved with parasites, but in none that evolved without parasites ($p \ll 0.001$, Fishers exact test). In a third treatment, parasites were removed at the mid-point of the runs, and the cured host populations (green) evolved substantially reduced complexity relative to the coevolution treatment ($p \ll 0.001$, Mann-Whitney $U = 543.5$), although the cured hosts retained greater complexity than those that never saw the parasites ($p = 0.002$, Mann-Whitney $U = 1703$).

The increased complexity relative to the ancestor observed in the absence of parasites ($p \ll 0.001$, Wilcoxon signed-rank $W = 1275$) accords with a simple model that couples a random walk in complexity with a selective constraint that limits functional degradation; Gould dubbed this model the "drunkards walk", alluding to how a patron leaving a pub eventually stumbles to the curb because the pub itself limits backward movement [104]. In our experiments, all populations started from the same ancestral program that could perform only the simplest function, `NOT`, and hence they were the least complex programs able to obtain resources and reproduce. Any less complex genotypes generated by mutation could not reproduce and were thus eliminated. More complex organisms also arose by mutation; although they obtained no additional resources for performing more complex functions (and, in fact, might replicate more slowly), they nonetheless could reproduce and thereby persist. Over time, this asymmetrical constraint allowed complexity to increase, albeit slowly and to
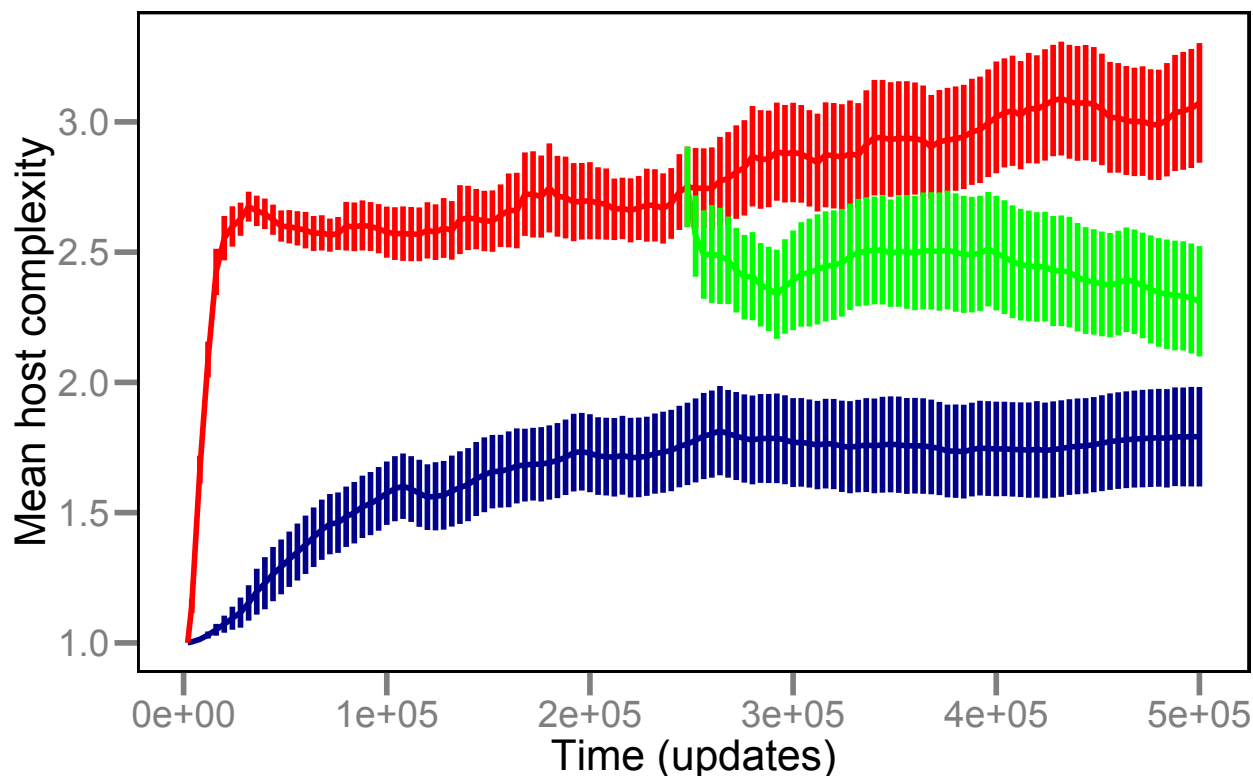
Figure 3.2: Parasites promote the evolution of host complexity. Complexity was measured as the minimum number of NAND instructions that must be executed by a host to perform its most complex logic function, averaged over all individuals in a population. The blue trajectory shows the grand mean complexity across 50 replicate populations (i.e., runs) that evolved in the absence of parasites. The red trajectory shows the corresponding values for 50 host populations that coevolved with parasites. In 12 runs, the parasites went extinct, in all but one case after 22,000 updates and after the hosts had evolved either the XOR (complexity 4) or EQU (complexity 5) function. The green trajectory shows mean values for the same 50 parasite populations, except here they were cured" by experimentally eliminating the parasites after 250,000 updates. All populations started with a single host genotype that performs only the NOT function. Updates are arbitrary Avida time units (see Methods). Error bars are +/- 2 Standard Errors of the Mean (SEM).

31

a limited extent. This explanation of complexity evolving as a "drunkards walk" does not imply that evolution as a whole operates as a random walk; instead, it only implies that complexity might follow such a pattern.

The coevolutionary process clearly produced greater functional complexity in the hosts. In broad outline, this effect occurs because parasites constantly select for new host phenotypes and thereby cause host populations to explore adaptive landscapes more broadly than hosts that are evolving alone [180]. However, it is not obvious why the effect was so large and continued for so long. Understanding the initial increase in complexity is seemingly straightforward–hosts must evolve some function other than NOT to avoid infection yet still reproduce, and all except one of the other functions have higher complexity than NOT. But this explanation alone cannot explain even the initial step, because the first new function to arise by mutation was, in the vast majority of cases, the one other function, NAND, that also requires executing only a single NAND instruction. In fact, the average complexity of the first new function was only 1.10 (1.011.19 95% confidence interval), and the maximum was only 2 in any case. What then might account for the large and sustained rise in complexity? One plausible explanation is an escalatory arms race that gives rise to progressively more extreme and complex adaptations [33, 177, 180]. For example, coevolution between cheetahs and gazelles may have favored ever-increasing speed, which was achieved by evolving more complex musculoskeletal systems. In many systems, however, coevolution does not occur along a single axis, but instead involves many traits [57] and can lead to fluctuating frequency-dependent selection instead of an arms race [181]. For example, such frequency-dependent fluctuations appear to dominate the interactions between *Daphnia magna* and its parasite *Pasteuria ramosa*, as determined by reviving eggs and spores from various sediment depths representing different historical states of the interaction [37].

Escalating arms races and negative frequency-dependent cycling, in general, are the two main outcomes of host-parasite coevolution. Escalation could lead to an increase in complexity if, for example, more complex tasks provided hosts with resistance to any less complex parasites. However, there is no such task "dominance" in Avida. Instead, a particular parasite can infect

a particular host provided they share at least one function. Given that requirement, there is no inherent reason that escalation must occur [163, 164]–for example, the host and parasite populations could cycle repeatedly between two states–and so we can reject the arms-race hypothesis as a sufficient explanation for the emergence of more complex traits in hosts that coevolved with parasites. Nonetheless, it is important to note that frequency-dependence and escalation are not mutually exclusive processes.

3.2.2 Parasites Retain "Memory" of Previous Hosts

How could negative frequency-dependence drive a sustained increase in host complexity rather than producing simple cycles? One possible explanation is that parasites maintain a "memory" of previously encountered host states. If so, then hosts can escape infection only by evolving in a previously unexplored direction–in the Avida system, by evolving an entirely new and therefore usually more complex function to acquire resources, rather than recycling one that was previously discarded after it was targeted by the parasite. The simplest way to achieve such memory is if a parasite population evolves generalist phenotypes that can infect multiple hosts, including types no longer common in the community. Indeed, the coexistence of multiple host types maintained by negative frequency-dependent selection would favor parasites with broad host-ranges. To examine whether this population-genetic memory existed, we quantified the average number of functions that parasites could perform. Consistent with the memory hypothesis, parasites evolved to become generalists that often performed four or five functions and thereby could infect several different host types (Figure 3.3). By contrast, we expect the hosts to evolve primarily as specialists because an individual needs to perform only one function to obtain resources, and performing multiple functions makes it vulnerable to a broader range of parasites. Indeed, most hosts performed only a single function (Figure 3.3), although that function became much more complex over time (Figure 3.2).

To verify that the parasites population-genetic memory drove the evolution of host complexity, we performed another set of coevolution experiments using a "challenge" design. This design is
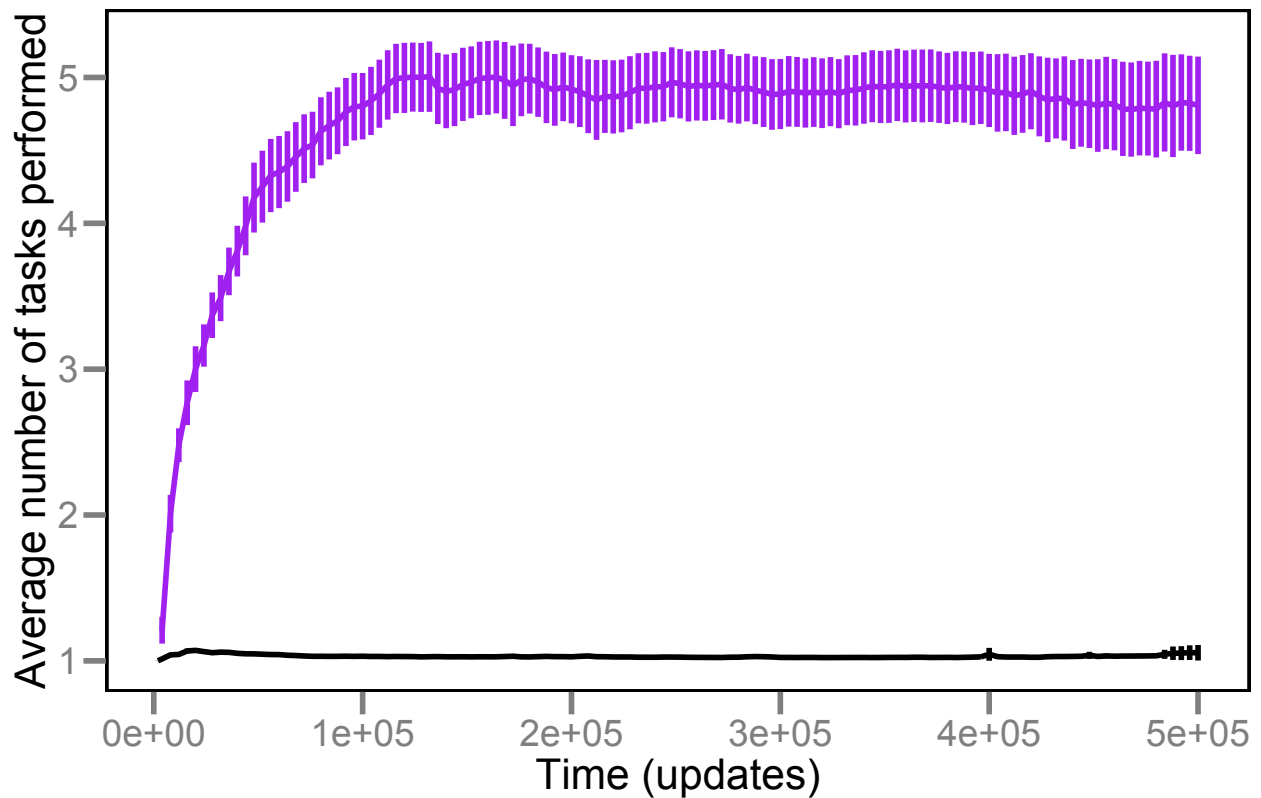
Figure 3.3: Parasites evolve generalist strategies while hosts remain specialists. The purple trajectory shows the average number of different functions performed by individual parasites across 50 replicates of the coevolution treatment. In 12 cases, the parasite population eventually went extinct, and so the number of replicates declines to 38 over time. The black trajectory shows the corresponding average for individual hosts; host populations were excluded from the average after the corresponding parasite populations had gone extinct. Error bars are +/- 2 SEM.

analogous to a microbiological approach in which bacteria are challenged with phage, a single resistant mutant is isolated, the phage are then challenged with the resistant host, a single host-range mutant is isolated that can overcome the resistance, and the cycle is repeated [105]. Using this design, diversity is lost because only individual mutants are retained at each step, and the advantage to the parasite of retaining a broad host-range (i.e., memory of prior hosts) is reduced or eliminated. Therefore, if the parasites population-genetic memory drove the evolution of host complexity in the original coevolution treatment (Figure 3.2), then we expect hosts to achieve reduced complexity under the challenge regime. Indeed, the resulting host complexity was much lower in the challenge treatment than with coevolution ($p \ll 0.001$, Mann-Whitney $U = 2373$); in fact, the challenge treatment was indistinguishable from the populations that had evolved without parasites ($p = 0.43$, Mann-Whitney $U = 1298$).

We can form an intuitive understanding of the parasites population-genetic memory and its effects on the evolution of complexity using the imagery of an adaptive landscape. Consider the case where increasing complexity is disadvantageous because performing more complex functions requires more resources than performing simpler tasks. In the absence of parasites, hosts will evolve the simplest viable functions (Figure 3.4A). However, when this host is targeted by parasites, the landscape is deformed, creating a new peak at a slightly more complex task (Figure 3.4B). As coevolution continues, additional hosts and parasites will evolve and a diverse set may be maintained through negative frequency-dependent selection. This community further depresses the landscape, thus moving the peak toward even higher levels of complexity (Figure 3.4 C-D). To evaluate whether our experiments supported this intuitive model, we measured the proportion of parasites unable to infect hosts performing each one of the nine logic functions on its own. That proportion represents a critical fitness component of the host because it reflects the hosts ability to resist infections by the parasites present in its environment. Figure 3.4 E-H shows the empirical relationship between average host fitness (i.e., resistance) and the complexity of the task performed over evolutionary time. In support of our population-genetic memory hypothesis, the fitness peak shifted strikingly toward higher levels of complexity as coevolution progressed. Thus, the diversity of parasites–with

their individually and collectively broad host-ranges–sustained a memory of previously evolved

host phenotypes and generated an adaptive landscape for the host that favored increasingly complex

tasks.
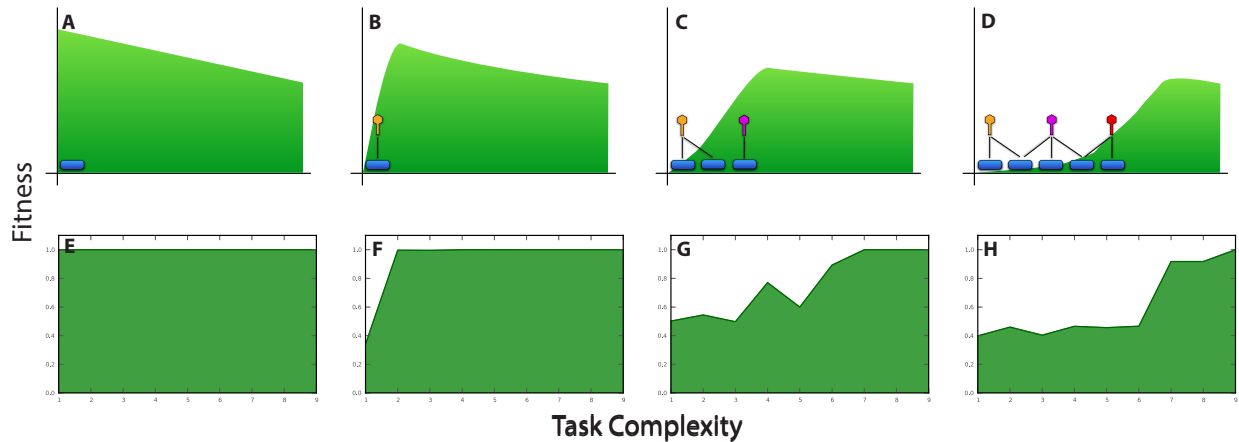
Figure 3.4: Effects of parasites on the hosts adaptive landscape. (A) Assuming that unnecessary complexity is costly in the absence of any direct benefit, the fitness peak corresponds to the simplest host phenotype. (B) Once parasites are introduced, the landscape is deformed and selection favors a more complex host phenotype. (C-D) As coevolution continues, the parasites maintain a population-genetic memory of host phenotypes, which pushes the fitness peak toward higher and higher levels of complexity. (E-H) In the coevolution runs, we quantified the effect of parasites on the host adaptive landscape as the proportion of parasites that were unable to infect hosts performing each of the nine logic functions.

3.2.3 Effects of Breaking the Coevolutionary Feedback

To test whether the fitness landscape shaped by a coevolved population of parasites was sufficient to

drive the evolution of complexity observed in our original coevolution treatment, we performed a

new treatment in which the parasite population began with genotypes frozen" at the frequency they

occurred within each original replicate at 250,000 updates (the halfway point, when the majority

of host complexity and parasite diversity had evolved), but further evolution of the parasite was

precluded. To maintain constant frequencies of the parasite genotypes, each newly reproduced

parasite was assigned a random genotype from the 250,000-update set. After 500,000 updates

in this complex-but-static environment of frozen parasite frequencies, hosts evolved significantly

higher complexity than in the treatment without parasites ($p = 0.003$, Mann-Whitney $U = 1684$). However, the hosts confronted with the complex-but-static parasite populations did not reach as high a level of complexity as when the parasites coevolved ($p \ll 0.001$, Mann-Whitney $U = 1946$, Figure 3.5). This disparity may indicate an effect of fluctuating environments, such that dynamic parasite environments favor increased host complexity more than complex-but-static parasite environments. To test this hypothesis, we then allowed hosts to evolve in environments where we "replayed" the changing parasite genotype frequencies over time from the coevolution treatment, but where these parasite genotypes did not respond to the host evolution that was occurring within any particular replicate. Again, the host populations that evolved in this replay treatment achieved significantly greater complexity than those that evolved without the parasites ($p = 0.034$, Mann-Whitney $U = 1529$), but the hosts in the replay treatment still did not reach as high levels of complexity as the coevolved hosts ($p \ll 0.001$, Mann-Whitney $U = 1908$, Figure 3.5).

Thus, coevolved parasites–whether constant (frozen) or varying over time (replayed)– favored the evolution of hosts with more complex functions than hosts that evolved without parasites at all. Nonetheless, the hosts under these treatments failed to evolve the highest level of complexity, which they achieved with coevolving parasites. Coevolution involves reciprocal changes in which the host population influences how the parasite population responds, both ecologically and evolutionarily, and vice versa. Although the parasite population was diverse in both the frozen and replayed treatments, and while it varied in time in the latter treatment, the evolution of the parasite population was decoupled from the evolutionary changes that occurred in the host population. Taken together, these experiments thus indicate that the special push-and-pull of coevolution played a major role in the evolution of host complexity. They also imply a more dynamic view of population-genetic memory, one in which negative frequency-dependence constantly tunes the parasite population in response to host evolution. Without coevolutionary reciprocity, the interactions between host and parasite populations are dissonant and population-genetic memory is ineffective.
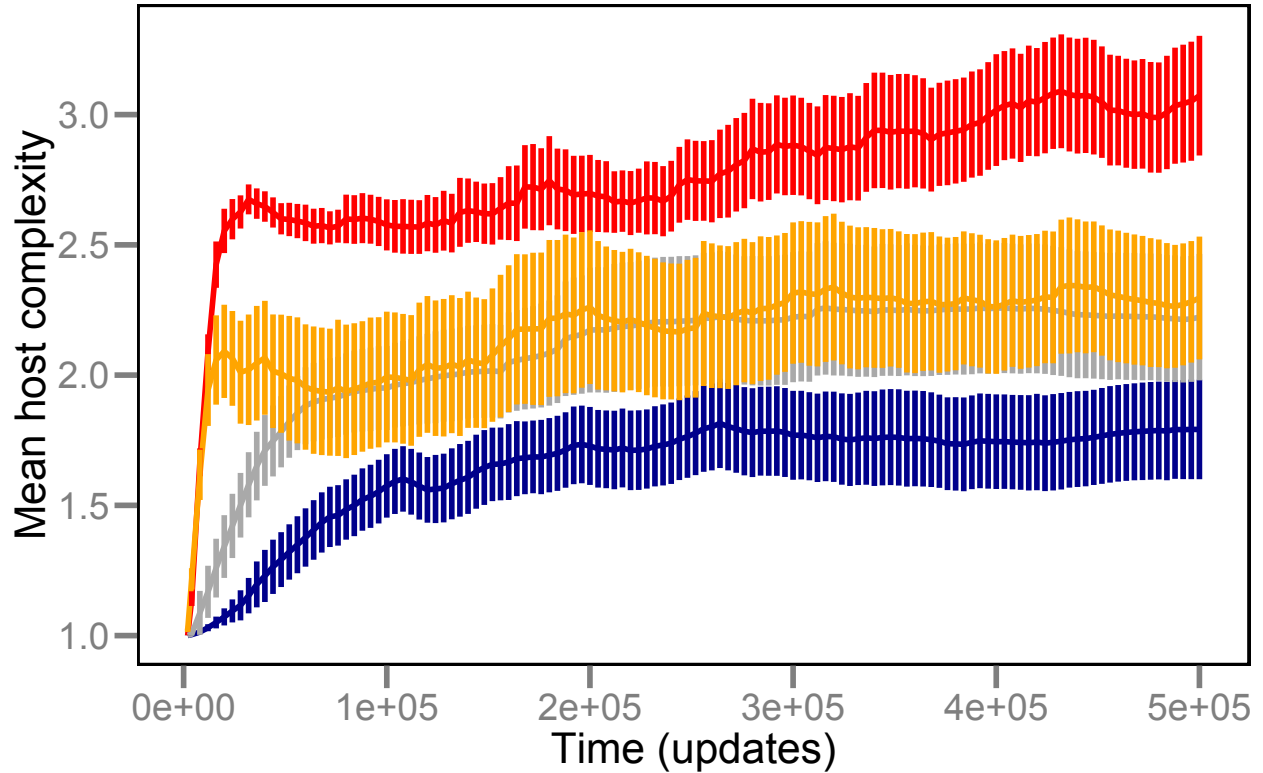
Figure 3.5: Frozen and replayed parasite genotypes fail to recapitulate the level of complexity seen in the coevolution treatment. Host complexity was measured as in Figure 3.2, and the coevolved (red) and evolved-without-parasites (blue) treatments are shown as before. The grey trajectory indicates the mean level of host complexity that evolved when parasite genotype frequencies were frozen at the values observed after 250,000 updates of coevolution. The orange trajectory shows the level of complexity that hosts evolved in the replay treatment, where they faced changing, but not coevolving, parasite populations. In this treatment, the parasite genotype frequencies were set to the levels observed during coevolution runs at 1,000-update intervals. The parasites went extinct before 250,000 updates in one of the coevolution replicates, and so the frozen treatment started with 49 replicates. In three of the 49 replicates of the frozen treatment, the hosts overcame the parasites and drove them extinct. In the replay treatment, a total of 30 host populations drove the replayed parasites extinct (including the 12 that went extinct during coevolution). Error bars are +/- 2 SEMs.

3.2.4 Effects of Coevolving Parasites on Host Phylogeny

Coevolution with parasites also had profound effects on the phylogenetic structure of host populations and on the phenotypic evolvability of host genomes. With respect to phylogenies, the frequency-dependent nature of host-parasite interactions promotes not only greater diversity at any given moment but also deeper branches that reflect the preservation of diversity through time. In Avida, we can track genealogies precisely and thus construct exact phylogenetic trees, avoiding uncertainty about historical states and branch lengths. Figure 3.6 shows representative trees for host populations that evolved in the presence and absence of parasites, and they differ strikingly in their coalescence profiles. To formalize this difference, we calculated the time since the most recent common ancestor (MRCA) for all 50 host populations in the coevolution and evolution-without-parasites treatments (Figure 3.7). The MRCA in coevolved host populations usually arose soon after the experiment began (median 6% of the total elapsed time), whereas the MRCA in the absence of parasites typically dated to well after the midpoint (median 74%), and this difference is highly significant ($p \ll 0.001$, Mann-Whitney $U = 1975$). Thus, coevolution not only affects the outcome of adaptation, but also fundamentally changes how those outcomes are reached. Coevolution was similarly found to increase the rate of adaptation when embedded in multispecies networks of mutualists [63]. Although the systems and form of interactions are different, their similar results suggest the important role reciprocity plays in evolving systems.

3.2.5 Effects of Coevolving Parasites on Host Evolvability

Previous research using Avida showed that different treatments could drive populations into qualitatively different regions of the fitness landscape; specifically, populations that experienced higher mutation rates evolved onto lower but flatter regions of genotypic space than populations that evolved at lower mutation rates, a phenomenon dubbed "survival of the flattest" [179]. Here we examine whether coevolution with parasites produced host genomes that were more evolvable with respect to escaping infections. To that end, we mapped phenotypic changes onto every possible one-step point mutation for the most common host genotype from all evolved and coevolved populations
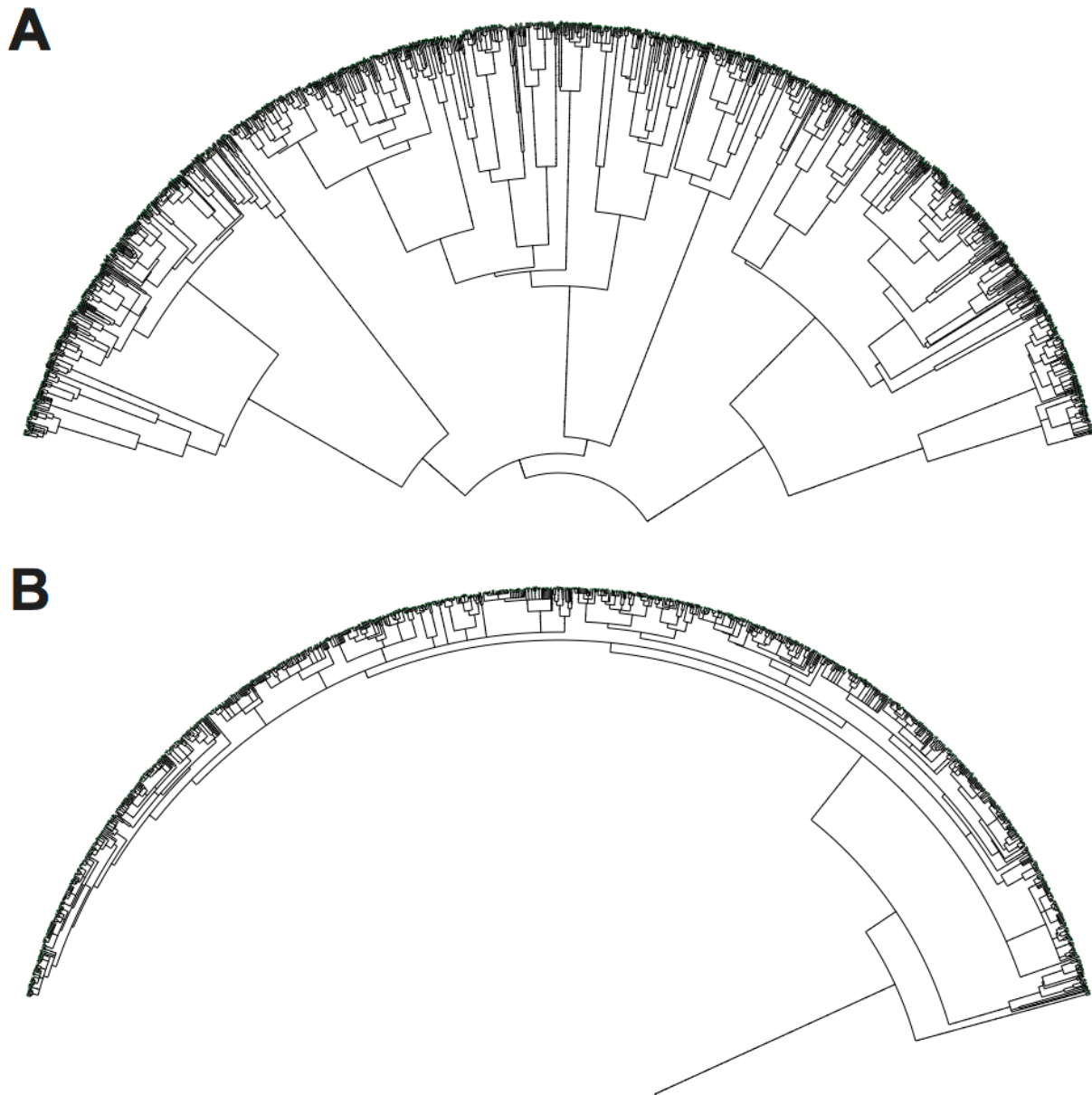
Figure 3.6: Effect of coevolving parasites on host phylogenies. Representative phylogenies for hosts that evolved in the (A) presence and (B) absence of parasites. The branch leading to the original ancestor is too short to be seen in (A). The phylogenies show all of the host genotypes present at the end of the run, and the phylogenies are known exactly in this system.
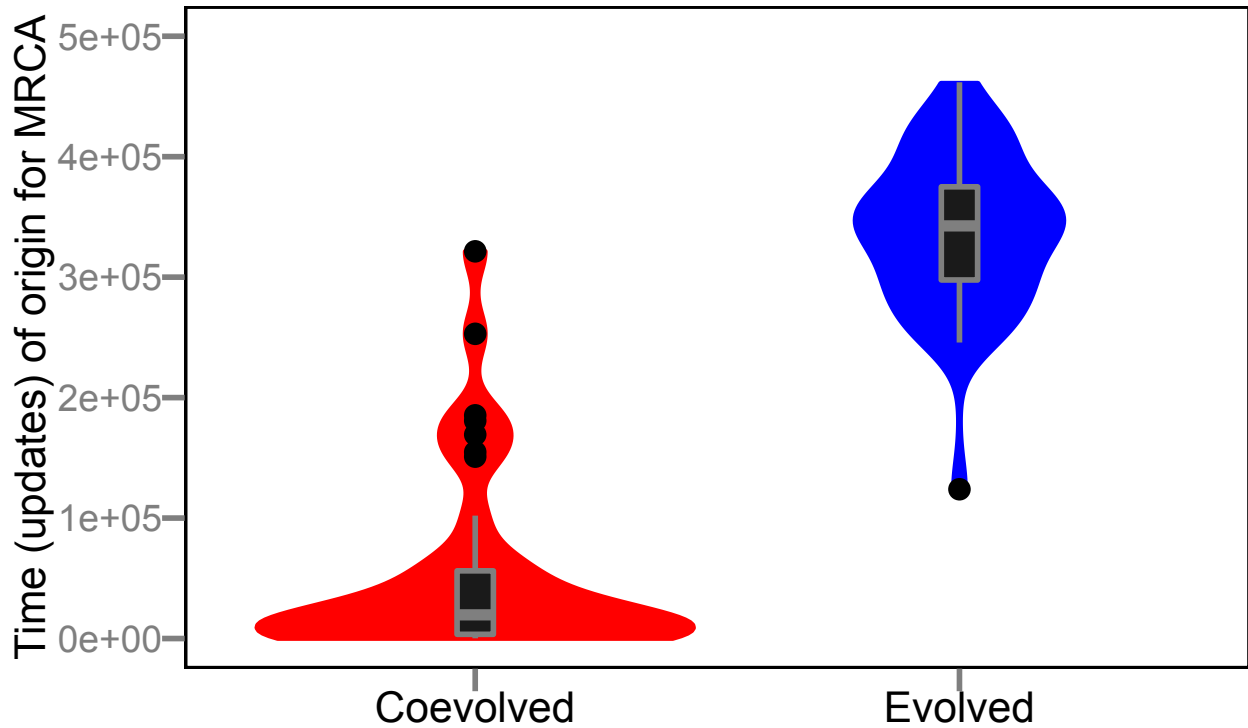
Figure 3.7: Effect of coevolution on coalescence times in host phylogenies. The data are shown as box plots and smoothed frequency distributions for the times of origin of the most recent common ancestors (MRCA) in 38 host populations that coevolved with parasites (excluding the 12 runs where the parasites went extinct) and 50 populations that evolved without parasites. The MRCAs arose significantly earlier in the coevolution treatment. The tail of the distribution for the coevolution treatment is more pronounced if we include the host populations where the parasites went extinct, but the difference remains highly significant ($p \ll 0.001$, Mann-Whitney $U = 2053$). Box hinges depict first and third quartiles and whiskers extend 1.5 x Inter Quartile Range (IQR) out from their corresponding hinge.

at the end of the experiment. Several types of phenotypic changes are possible including the gain of a function, the loss of a function, or switching which function is performed without changing the total number of functions performed. Mutations in the last category are of particular interest because, in the presence of parasites, the ability to switch functions without requiring intermediate steps (adding a new function before losing the old one) could be adaptive. That is, more evolvable hosts would be able to change phenotypes faster and could thereby escape coevolving parasites more readily. While selection does not directly favor hosts with more evolvable genotypes, they are more likely to produce surviving lineages when coevolving with parasites; thus, second-order selection could drive the evolution of evolvability. In strong support of this hypothesis, function-switching mutations were > 10-fold more common in hosts that evolved with parasites than in hosts that evolved without parasites ($p \ll 0.001$, Mann-Whitney $U = 2338$, Figure 3.8). To evaluate whether this effect might somehow merely reflect the more complex tasks typically performed by coevolved hosts, we analyzed pairs of genotypes from the coevolved and evolved host populations that perform identical sets of tasks. The coevolved hosts were still significantly more evolvable than their paired evolved host ($p \ll 0.001$, Wilcoxon signed-rank $W = 112616.5$, Figure 3.9), although the frequency of task-switching mutations tended to be lower in both treatments after this pairing procedure. Thus, coevolution drove host populations to occupy more evolvable regions of the adaptive landscape.

Taken together, our experiments show that parasites pushed hosts to levels of functional complexity that were well beyond what they achieved by random walks (Figure 3.2). This complexity resulted from population-level processes [81, 92, 168], in which frequency-dependent interactions sustained generalist parasites (Figure 3.3) that were supported by phenotypically and phylogenetically diverse hosts (Figures 3.6 and 3.7). If population-level effects were eliminated, as in the challenge experiments, then host complexity remained low. Moreover, if the coevolutionary feedback between hosts and parasites was broken by freezing or replaying parasite genotypes, then hosts did not evolve such complex tasks as when parasite populations could respond to the changing host population (Figures 3.4 and 3.5). Although the form of interactions between the hosts, their resources, and parasites in our study system (Figure 3.1) strongly constrained host evolution (e.g., hosts performing

42
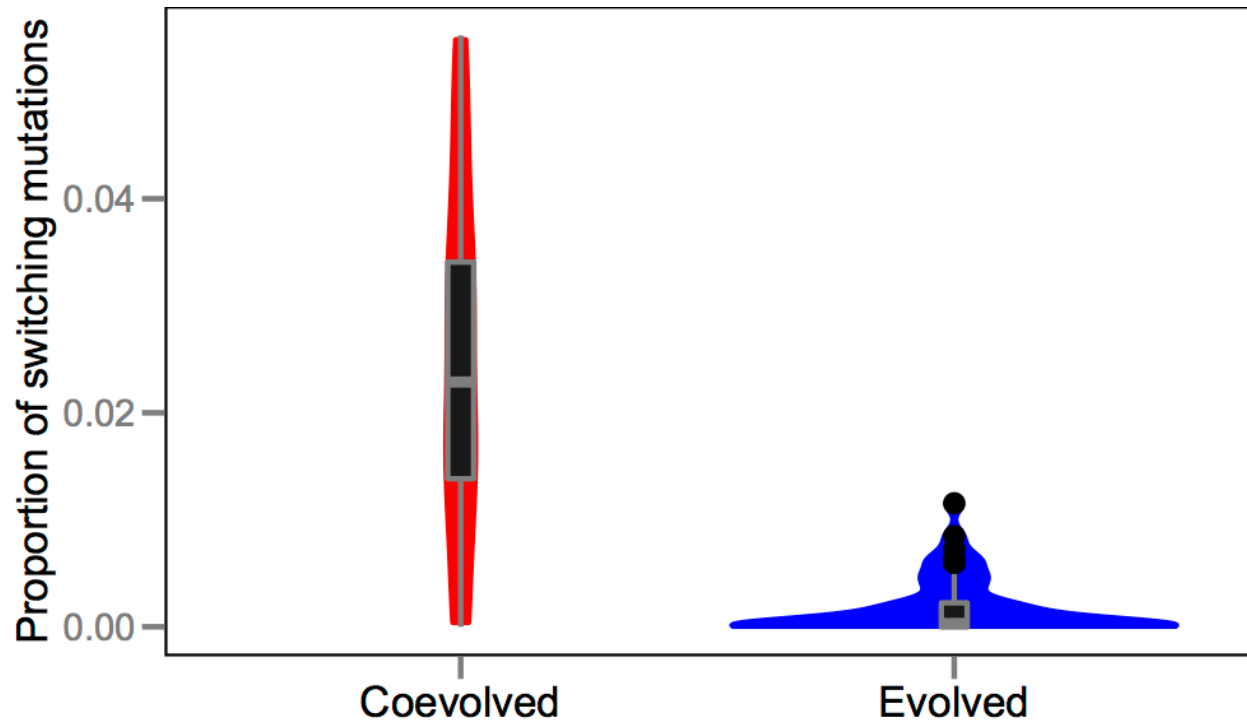
Figure 3.8: Proportion of point mutations in host genomes that switch functions without changing the number of functions performed. The data are shown as box plots and smoothed frequency distributions. Proportions were obtained by testing all possible one-step point mutations in the genetic background of the most abundant host genotype at the end of all 50 runs with and without parasites. Box hinges depict first and third quartiles and whiskers extend 1.5 x IQR out from their corresponding hinge.
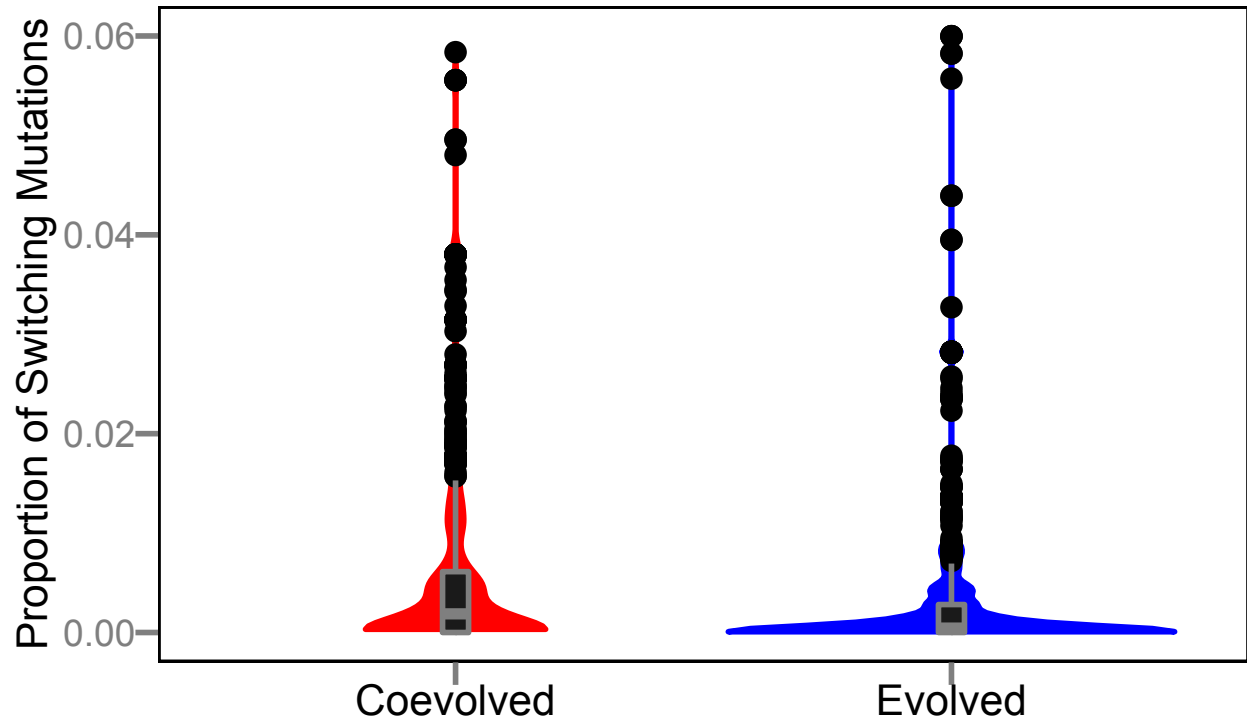
Figure 3.9: Proportion of point mutations that switch functions without changing the number of tasks performed in paired genotypes, where each pair includes hosts from the coevolution and evolution treatments that perform identical sets of tasks. Proportions were obtained by testing all possible one-step point mutations in each genetic background. Box hinges depict first and third quartiles and whiskers extend 1.5 x IQR out from their corresponding hinge.

multiple functions were more broadly susceptible to parasites and rarely observed), hosts nevertheless overcame these limitations by becoming more evolvable (Figure 3.8). In particular, host genomes evolved such that a much larger proportion of mutations caused a switch from one resource-acquisition function to another, thereby allowing hosts to escape, in a single step, parasites that targeted the first function. These results–from an unusual but highly tractable system–add to growing evidence from experiments and theory that coevolutionary processes promote biological diversity, new functions, and evolvability [37, 57, 81, 92, 106, 110, 112, 117, 163, 164, 166, 168, 171, 177, 181].

3.3   Materials and Methods

3.3.1   Evolution Experiments

All experiments were performed using the Avida 2.13.0 software, which is available without cost (http://avida.devosoft.org/). Configuration files with the parameter settings used will be deposited in the Dryad database upon publication. Host and parasite populations lived in a well-mixed chemostat-like environment, with a single type of resource entering at a constant rate. Hosts obtained resources required for replication by performing any of nine distinct one- and two-input logic functions, provided there were resources available in the environment. A parasite could infect a host if they performed at least one function in common, and an infecting parasite then acquired 80% of its hosts energy (CPU cycles) [51]. The ancestral hosts and parasites could perform only NOT, one of the two simplest functions. We initially monitored evolution under two main treatments, each with 50-fold replication: host organisms evolved alone in one treatment, and they coevolved with parasites in the other. Each replicate started with a different numerical seed, and the resulting sequence of pseudo-random numbers influenced mutations, parasite-host encounters, and other probabilistic events. The parasites went extinct in 12 coevolution runs; except where otherwise noted, we included those runs in our analyses. In a third treatment, the parasites were experimentally removed halfway through each run, with the first half being identical to a run in the coevolution treatment (i.e., using the same initial seed).

All runs lasted for 500,000 updates; an update is an absolute time unit in Avida equal to the execution, on average, of 30 instructions per individual host organism. Generation times for the ancestral host and parasite genotypes were 63 and 23 updates, respectively, although generation times changed as genomes evolved. Each host population began with one individual; the carrying capacity was 14,400 in the absence of parasites. In the coevolution treatment, 400 parasites were introduced after 2,000 updates; only a single parasite could infect an individual host. Mutation rates were 0.25 and 0.5 per genome replication for the ancestral host and parasite, respectively, of which 90% were point mutations and 5% each were insertions or deletions of single instructions. Per-site mutation rates were constant, so total genomic rates varied with changes in genome length. Mutations occurred at random with respect to genome position.

3.3.2 Challenge Experiment

To eliminate all population-level interactions in both species, we screened individual hosts and parasites for defenses and counter-defenses, rather than using evolving populations. Starting from the same ancestral host, we generated thousands of individuals using the same mutation regime as in the evolution experiments, and we randomly chose a single host mutant that was resistant to the ancestral parasite. We then repeated this process for parasites, again using the same mutation regime as in the evolution experiments, and we isolated a host-range mutant able to infect that resistant host mutant. We continued the pairwise challenges using the derived host and parasite genotypes for 50 rounds. A challenge experiment was stopped if we failed to isolate a relevant mutant after screening 500,000 individuals. In the comparisons with the evolution and coevolution treatments, we used 56 challenge experiments (out of 100 started) that achieved the full 50 rounds of reciprocal defenses and counter-defenses. However, the truncated runs appeared to be indistinguishable from those that went the full duration.

3.3.3 Freeze and Replay Experiments

In these experiments, we allowed host populations to evolve with either "frozen" or "replayed" parasites. During the original coevolution experiments, we saved each replicates entire set of host and parasite genotypes every 1,000 updates. We modified the Avida source code such that this record of genotypes can be loaded into an on-going run at any point by adding an option to override the normal replication process with one that samples from a genotype list. When organisms reproduce, instead of inheriting their parents genome, the offspring is assigned a random genotype from the list. This procedure can be implemented for hosts, parasites, or both; however, in the freeze and replay experiments presented here, we manipulated only the parasite populations using this new procedure. In both treatments, we injected 1,500 parasites into the host population after 2,000 updates; this number was increased relative to the coevolution treatment to ensure that the frozen and replayed parasite populations, which were sometimes poorly adapted to the ancestral host, did not go extinct.

In the freeze treatment, each host population confronted a parasite population that was complex and diverse, but constant in its genotypic frequencies over an entire run (except for the fluctuations associated with births and deaths of the parasites). The composition of each parasite population was based on the list of parasites taken at the mid-point (i.e., 250,000 updates) of one of the coevolution treatment runs. Thus, the genetic composition of the parasite population was frozen throughout the run, although the total number of parasites could rise or fall in accord with the dynamics of infections. Under the replay treatment, the frequencies of parasite genotypes changed over time, but those changes were based on parasite evolution that had taken place in an earlier coevolution run, rather than on the dynamics that were occurring in the replay itself. That is, the list of parasite genotypes from which new parasites were drawn was changed every 1,000 updates to reflect what had happened in the earlier run. As a consequence, the host could evolve in response to the changing frequencies of the various parasite genotypes, but not vice versa–the coevolutionary feedback was broken, although parasite diversity and the temporal changes in that diversity were preserved.

3.3.4 Phylogenetic Analysis

In Avida, the genealogy of organisms is known perfectly and, when coupled with the asexual lineages studied here, allows construction of the exact phylogenetic history for a population. We used the python ete2.1 module to represent (Figure 3.6) all of the genotypes present in two host populations along with their ancestries through the various coalescences, the most recent common ancestor for the entire population, and the founding genotype.

3.3.5 Evolvability Analysis

We tested every possible one-step point mutation in the genetic background of the most abundant host genotype at the end of all 50 evolution and coevolution runs. Each mutant was placed into one of the following categories based on the phenotypic changes relative to its parent: (i) the mutant cannot perform any functions or is otherwise nonviable; or (ii) the mutant is viable and (a) there is no difference in the number or identity of functions performed; (b) the mutant performs more functions; (c) the mutant performs fewer functions; or (d) the identity of functions performed has changed, but the number has not. The last category, which we call "switching", was the focus of our analysis.

We also modified this analysis to take into account possible effects of differences in the number and complexity of tasks performed by pairing host genotypes isolated from the evolved and coevolved populations that performed identical sets of tasks. Genotypes were pooled across the replicate runs based on what tasks they could perform. All of the coevolved populations were compared with all of the evolved populations to identify paired host phenotypes that performed identical sets of tasks. For each pair of phenotypes thus identified, a genotype from the evolved and coevolved populations that performed the appropriate set of tasks was chosen at random, and all possible one-step mutations were then generated for both genotypes.

Chapter 4

Evolving Digital Ecological Networks

Authors: Luis Zaman[+], Miguel A. Fortuna[+], Aaron P. Wagner, Charles Ofria [51]

[+] These authors contributed equally

4.1 Overview

Evolving digital ecological networks are webs of interacting, self-replicating, and evolving computer programs (i.e., digital organisms) that experience the same major ecological interactions as biological organisms (e.g., competition, predation, parasitism, and mutualism). Despite being computational, these programs evolve quickly in an open-ended way, and starting from only one or two ancestral organisms, the formation of ecological networks can be observed in real-time by tracking interactions between the constantly evolving organism phenotypes. These phenotypes may be defined by combinations of logical computations (hereafter tasks) that digital organisms perform and by expressed behaviors that have evolved. The types and outcomes of interactions between phenotypes are determined by task overlap for logic-defined phenotypes and by responses to encounters in the case of behavioral phenotypes. Biologists use these evolving networks to study active and fundamental topics within evolutionary ecology (e.g., the extent to which the architecture of multispecies networks shape coevolutionary outcomes, and the processes involved).

4.2 Introduction

In nature, species do not evolve in isolation but in large networks of interacting species (see Figure 4.1). One of the main goals in evolutionary ecology is to disentangle the evolutionary mechanisms that shape and are shaped by patterns of interaction between species [56, 144, 165]. A particularly important question concerns how coevolution, the reciprocal evolutionary change in local populations of interacting species driven by natural selection [162], is shaped by the architecture of food webs, plantanimal mutualistic networks, and host-parasite communities. The concept of diffuse coevolution, where adaptation is in response to a suite of biotic interactions [78], was the first step towards a framework unifying relevant theories in community ecology and coevolution. Understanding how individual interactions within networks influence coevolution, and conversely how coevolution influences the overall structure of networks, requires an appreciation for how pairwise interactions change due to their broader community contexts as well as how this community context shapes selective pressures [53, 160]. Accordingly, research is now focusing on how reciprocal selection influences and is embedded within the structure of multispecies interactive webs, not only on particular species in isolation [165].

Coevolution in a community context can be addressed theoretically via mathematical modeling and simulation [57, 182], by looking at ancient footprints of evolutionary history via ecological patterns that persist and are observable today [59, 136], and by performing laboratory experiments with microorganisms [16]. In spite of the long time scales involved and the substantial effort that is necessary to isolate and quantify samples, the latter approach of testing biological evolution in the lab has been successful over the last two decades [99]. However, studying the evolution of interspecific interactions, which involves dealing with more complex webs of multiple interacting species, has proven to be a much more difficult challenge. A meta-analysis of bacteria-phage interaction networks, carried out by Weitz and his team [46], found a striking statistical structure to the patterns of infection and resistance across a wide variety of environments and methods from which the hosts and phage were obtained. However, the ecological mechanisms and evolutionary processes responsible have yet to be unraveled.
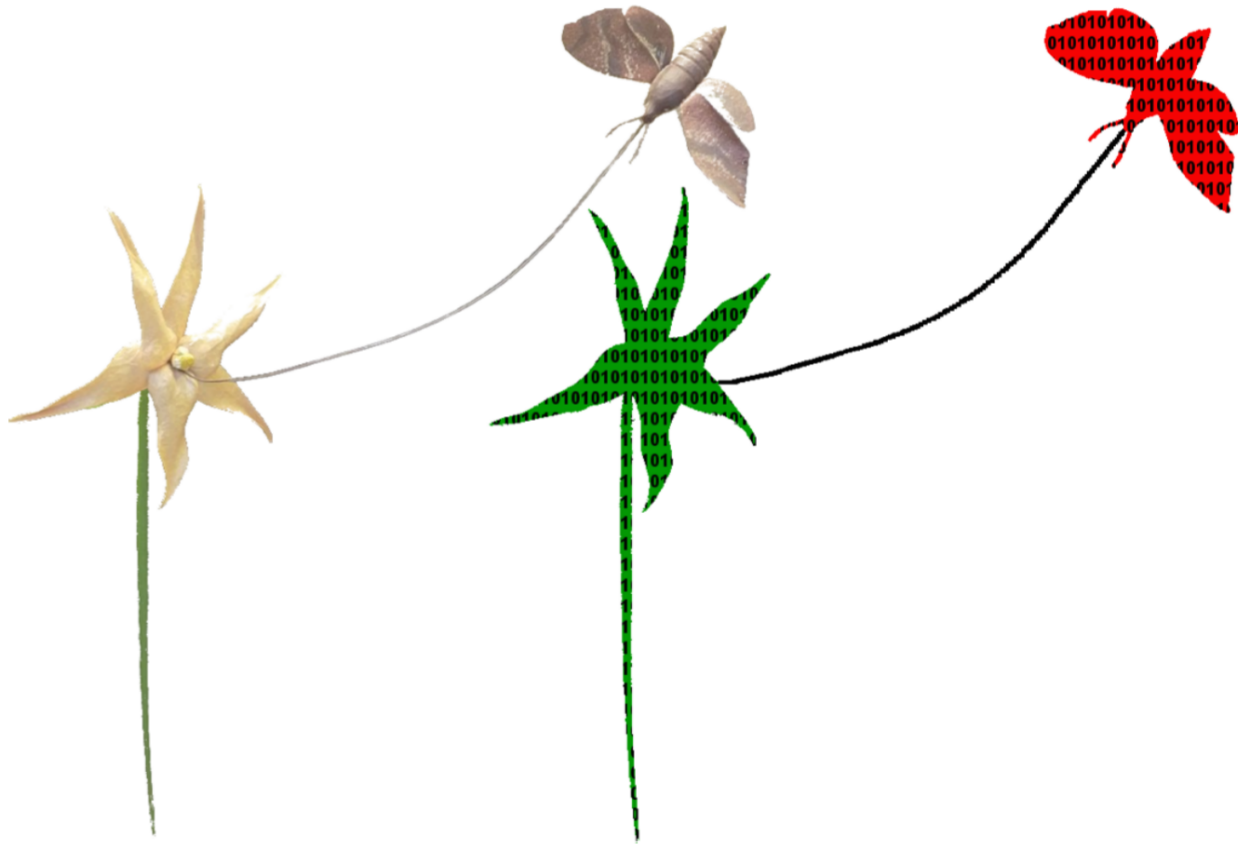
Figure 4.1: When Darwin received an orchid (*Angraecum sesquipedale*) from Madagascar whose nectary was one and a half feet long, he surmised that there must be a pollinator moth with a proboscis long enough to reach the nectar at the end of the spur [89]. In its attempt to get the nectar, the moth would have pollen rubbed onto its head, and the next orchid visited would then be pollinated. In 1903, such a moth was discovered: *Xanthopan morgani*. This was a remarkable example of an evolutionary prediction. However, because species coevolve within large networks of multispecies ecological interactions, this example of pairwise coevolution is more the exception than the rule.

Digital ecological networks enable the direct, comprehensive, and real time observation of evolving ecological interactions between antagonistic and/or mutualistic digital organisms that are difficult to study in nature. Research using self-replicating computer programs can help us understand how coevolution shapes the emergence and diversification of coevolving species interaction networks and, in turn, how changes in the overall structure of the web (e.g., through extinction of taxa or the introduction of invasive species) affect the evolution of a given species. Studying the evolution of species interaction networks in these artificial evolving systems also contributes to the development of the field, while overcoming limitations evolutionary biologists may face. For example, laboratory studies have shown that historical contingency can enable or impede the outcome of the interactions between bacteria and phage, depending on the order in which mutations occur: the phage often, but not always, evolves the ability to infect a novel host [110]. Therefore, in order to obtain statistical power for predicting such outcomes of the coevolutionary process, experiments require a high level of replication. This stochastic nature of the evolutionary process was exemplified by Stephen Jay Gould's inquiry ("What would happen if the tape of the history of life were rewound and replayed?") [61]. Because of their ease in scalability and replication, evolving digital ecological networks open the door to experiments that incorporate this approach of replaying the tape of life. Such experiments allow researchers to quantify the role of historical contingency and repeatability in network evolution, enabling predictions about the architecture and dynamics of large networks of interacting species.

The inclusion of ecological interactions in digital systems enables new research avenues: investigations using self-replicating computer programs complement laboratory efforts by broadening the breadth of viable experiments focused on the emergence and diversification of coevolving interactions in complex communities. This cross-disciplinary research program provides fertile grounds for new collaborations between computer scientists and evolutionary biologists.

4.3 History

4.3.1 Coreworld

The field of digital life was inspired by the rampant computer viruses of the 1980s. These viruses were self-replicating computer programs that spread from one computer to another, but they did not evolve. Steen Rasmussen was the first to include the possibility of mutation in self-replicating computer programs by extending the once-popular Core War game, where programs competed in a digital battle ground for the computer's resources [132]. Although Rasmussen observed some interesting evolution, mutations in this early genetic programing language produced many unstable organisms, thus prohibiting scientific experiments. Just one year later, Thomas S. Ray developed an alternative system, Tierra, and performed the first successful experiments with evolving populations of self-replicating computer programs [133].

4.3.2 Tierra

Thomas S. Ray created a genetic language similar to earlier digital systems, but added several key features that made it more suitable for evolution in his artificial life system, Tierra. Primarily, he prevented instructions from writing beyond the privately allocated memory space, thus limiting the potential for organisms writing over others [20]. The only selective pressure in Tierra was for rapid self-replication. Over the course of evolution, this pressure lead to shorter and shorter genomes, reducing the time spent copying instructions during replication. Some individuals even started executing the replication code in other organisms, allowing those cheaters, which were originally referred to as parasites in Ray's work, to further shrink their genetic programs. This form of cheating was the first evolved ecological interaction between organisms in artificial life software. Ray's cheaters pre-dated the formal study of evolving ecological interactions using Tierra-like digital evolution platforms by 20 years.

4.3.3 Avida

In 1993, Christoph Adami, Charles Ofria, and C. Titus Brown created the artificial life platform Avida [121] at the California Institute of Technology. They added the ability for digital organisms to obtain bonus CPU cycles for performing computational tasks, like adding two numbers together. In Avida, researchers can define the available tasks and set the consequences for organisms upon successful calculation [121]. When organisms are rewarded with additional CPU cycles, their replication rate increases. Since Avida was designed specifically as a scientific tool, it allows users to collect a comprehensive suite of data about evolving populations. Due to its flexibility and data tracking abilities, Avida has become the most widely used digital system for studying evolution. The Devolab (http://devolab.msu.edu/) at the BEACON Center currently continues development of Avida.

4.4 Implementation

4.4.1 Digital Organsims

Digital organisms in Avida are self-replicating computer programs with a genome composed of assembly-like instructions. The genetic programing language in Avida contains instructions for manipulating values in registers and stacks as well as for control flow and mathematical operations. Each digital organism contains virtual hardware on which its genome is executed. To reproduce, digital organisms must copy their genome instruction by instruction (see Figure 4.2) into a new region of memory through a potentially noisy channel that may lead to errors (i.e., mutations). While most mutations are detrimental, mutants will occasionally have higher fitness than their parents, thereby providing the basis for natural selection with all of the necessary components for Darwinian evolution. Digital organisms can acquire random binary numbers from the environment and are able to manipulate them using their genetic instructions, including the logic instruction NAND. With only this instruction, digital organisms can compute any other task by stringing together

various operations because NAND is a universal logic function [97]. If the output of processing random numbers from the environment corresponds to the result of a particular logic task, then that task is incorporated into the set of tasks the organism performs, which, in turn, defines part of its phenotype.

Figure 4.2: The circular genome of a digital organism, on the left, consists of a set of instructions (represented here as letters). Some of these instructions are involved in the copy process and others in completing computational tasks. The experimenter determines the probability of mutations. Copy mutations occur when an instruction is copied incorrectly, and is instead replaced by a random instruction in the forming offspring's genome (as can be seen in the offspring, on the right). Other types of mutations, such as insertions and deletions are also implemented. All three of the parent's hardware pointers are represented: the instruction pointer (indicated by an i), the write-head pointer (indicated by a w), and the flow pointer (indicated by an f). Arcs inside the circular genome represent the execution flow, showing most of the CPU cycles being used during the copying process. After genome replication is complete, the parent organism divides off its offspring, which must now fend for itself within the Avida world.

4.4.2 Digital Interactions

Interactions between digital organisms occur through phenotypic matching, which, in the case of task-based phenotypes, results from the performance of overlapping logic functions (see Figure 4.3). Different mechanisms for mapping phenotypic matching to interactions can be implemented,

depending on the antagonistic or mutualistic nature of the interaction.

4.4.2.1 Host-parasite interactions

In host-parasite interactions, the parasite organisms benefits at the expense of the host organisms. Parasites in Avida are implemented just like other self-replicating digital organisms, but they live inside hosts and execute parasitic threads using CPU cycles stolen from their hosts [184]. Because parasites impose a cost (lost CPU cycles) on hosts, there is selection for resistance, and when resistance starts to spread in a population, there is selective pressure for parasites to infect those new resistant hosts. Infection occurs when both the parasite and host perform at least one overlapping task. Thus a host is resistant to a particular parasite if they do not share any tasks (see Figure 4.3). This mechanism of infection mimics the inverse-gene-for-gene model [44], in which infection only occurs if a host susceptibility gene (the presence of a logic task) is matched by a parasite virulence gene (a parasite performing the same task). Additional infection mechanisms, such as the matching allele and gene-for-gene models [4], can also be implemented.

In traditional infection genetic models, host resistance and pathogen infectivity have associated costs. These costs are an important part of theory about why defense genes do not always fix rapidly within populations [16]. Costs are also present in digital hostparasite interactions: performing more or more complex tasks implies larger genomes and hence slower reproduction. Tasks may also allow organisms access to resources present in the abiotic environment, and the environment can be carefully manipulated to control the relative costs or benefits of resistance.

By keeping track of task-based phenotypes as well as tracking information about successful infections in the community, researchers are able to perfectly reconstruct the interaction networks of digital coevolving hosts and parasites (see Figure 4.4). The structure of these networks is a result of the interplay between ecological processes, mainly host abundance, and coevolutionary dynamics, which lead to changes in host specificity [128].
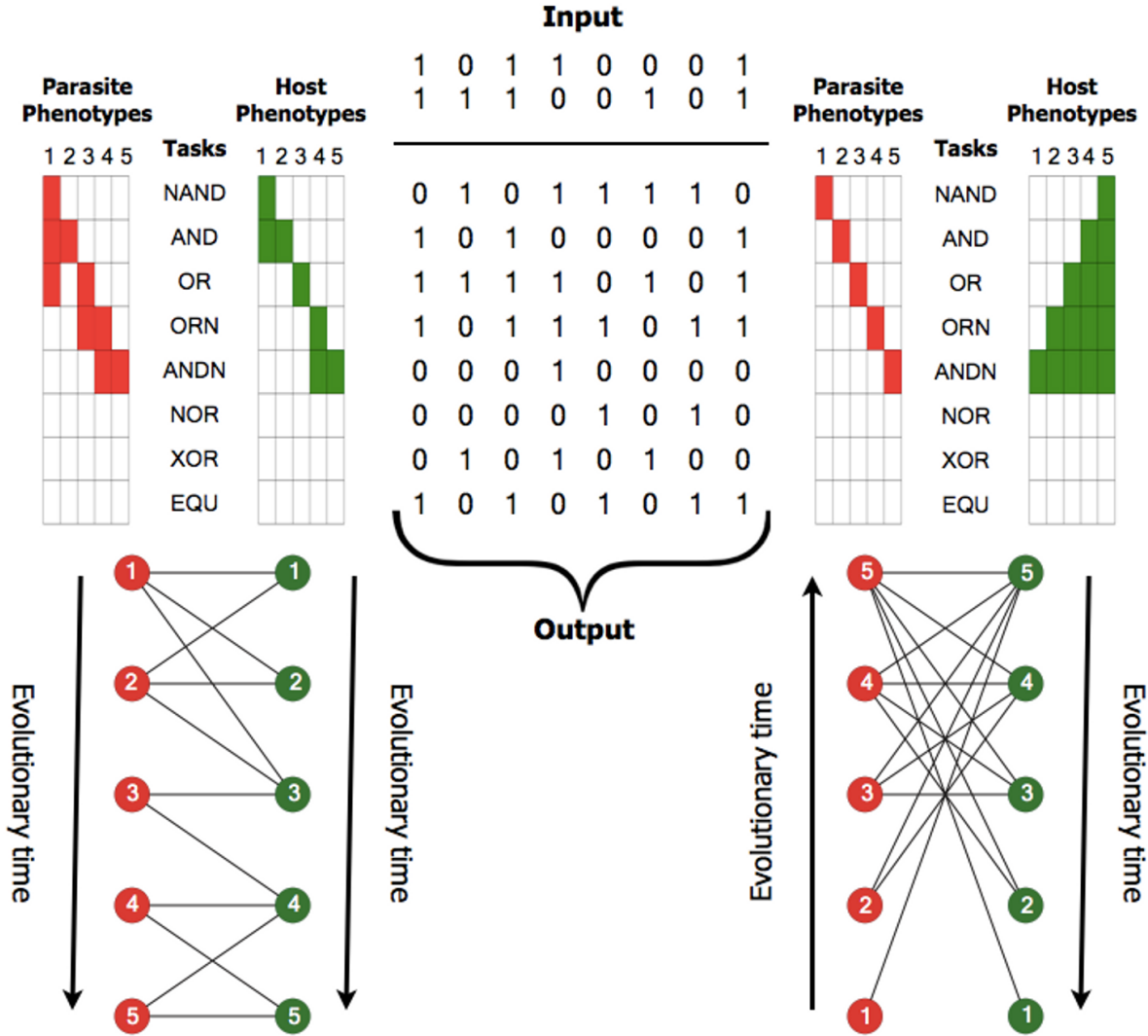
Figure 4.3: Digital organisms process binary numbers taken from the environment using the instructions that constitute their genomes. When the output of processing those numbers equals the result of applying a logic function, the digital organism is said to have performed that task. The combination of tasks performed by a digital organism partially defines its phenotype. The center of the figure depicts the output of applying eight logical operators (tasks) on the two input numbers above. On the left and right, five hypothetical host (green) and parasite (red) phenotypes are represented as columns (on the top) and as circles (below). On the top, each column depicts a phenotype and each row represents a task. Tasks performed by each phenotype are filled. In the lower part, the interaction networks between hosts and parasites are illustrated, which result from phenotypic matching: a parasite infects a host (indicated by a line) if it performs at least one task that is also performed by the host. Inset numbers indicate the identity of phenotypes represented on the top. Arrows represent the temporal direction of the coevolutionary process: from the earliest phenotype to the most recent one. The order of tasks (from top to bottom) indicates the time needed for a digital organism to perform that task over the course of the evolutionary trajectory. Depending on the pattern of tasks performed by the digital organisms, a modular (left) or nested (right) interaction network can emerge.
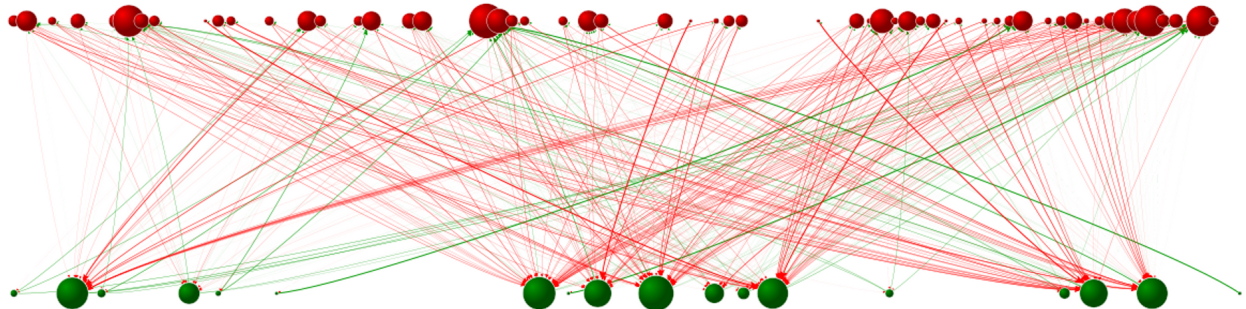
Figure 4.4: Starting from a host phenotype (green node) and a parasite phenotype (red node), a complex network of interactions (arrows) between hosts and parasites emerges out of the coevolutionary process. Nodes representing new host and parasite phenotypes appear and disappear over evolutionary time. The abundance of individuals expressing each phenotype changes continuously (indicated by node size) altering interaction patterns, and thus influencing subsequent coevolutionary dynamics. Interactions between a host phenotype and a parasite phenotype are depicted as arrows pointing in opposite directions: the thickness of red arrows indicates the fraction of infections that a particular parasite is responsible for inflicting on the indicated host phenotype, while the thickness of the green arrows indicates the fraction of all of the hosts a particular parasite phenotype infects that is accounted for by the indicated host phenotype. Often asymmetry between the thicknesses of arrow-pairs leads to red arrows dominating the picture. At these times, most parasite phenotypes are infecting only a small fraction of hosts expressing a given phenotype. Instead, the majority of those hosts are being infected by parasites with other phenotypes.

4.4.2.2 Mutualistic interactions

Interactions in which both species obtain mutual benefit, such as those between flowering plants and pollinators, and birds and fleshy fruits, can be implemented in evolving digital experiments by following the same task matching approach used for hostparasite interactions, but using free-living organisms instead of parasitic threads. For example, one way to set up a plant-pollinator type of interaction is to use an environment containing two mutually exclusive resources: one designated for "plant" organisms and one for "pollinator" organisms. Similar to parasites attempting infection, if tasks overlap between a pollinator and a plant it visits, pollination is successful and both organisms obtain extra CPU cycles. Thus, these digital organisms obtain mutual benefit when they perform at least one common task, and more common tasks lead to larger mutual benefits. While this is one specific way to enable mutualistic interactions, many others are possible in Avida. Interactions that begin as parasitic may even evolve to be mutualistic under the right conditions. In most cases, coevolution will result in concurrent interactions between multiple phenotypes. Thus, observed networks of mutualistic interactions can inform our understanding about the outcomes and processes of coevolution in complex communities [63].

4.4.2.3 Predatorprey interactions.

While host-parasite and mutualistic interactions are determined by task-based phenotypes, predator-prey interactions are determined by behavior. Predators are digital organisms that have evolved from ancestral prey phenotypes to locate, attack, and consume organisms. When a predator executes an attack instruction (acquired through mutation), it kills a neighboring organism. When predators kill prey, they gain resources required for reproduction (e.g., CPU cycles) proportional to the level accumulated by the consumed prey. Selection favors behavioral strategies in prey that enable them to avoid being eaten. At the same time, selection favors predators with behavioral strategies that improve their food finding and prey attacking abilities. The resulting diversity in the continuously evolving behavioral phenotypes creates dynamic predatorprey interaction networks in which selective forces are constantly changing as a consequence of the emergence of new, and

loss of old, behaviors. Because predators and prey move around in and use information about their environment, these experiments are typically carried out using spatially structured populations. On the other hand, hostparasite and mutualistic coevolution are often done in well-mixed environments, though the choice of the environment is at the discretion of the experimenter.

4.5   Research Directions

Understanding how biodiversity is organized in natural ecosystems requires going beyond the study of pairs of interacting species. Using digital organisms, one can find generalities about the evolutionary and ecological processes shaping the web of interactions among species, as well as the coevolutionary processes embedded within these networks. By tracing the evolution of digital communities and their ecological networks, researchers obtain perfect fossil records of how the number and patterns of links among interacting phenotypes evolved.

The stabilitydiversity debate [108] is a long-standing debate about whether more diverse ecological networks are also more stable. Until recently, this debate has focused on one component of biodiversity: species diversity. However, newer research has begun dealing with another component of biodiversity: diversity in species interactions. Mathematical models show that a mixture of antagonistic and mutualistic interactions can stabilize population dynamics and that the loss of one interaction type may critically destabilize ecosystems [116]. Studies with digital organisms can shed light on this debate from an empirical perspective because the types of interactions included can be manipulated and the stability of the resulting evolving digital ecological network can be measured.

Equally addressable using evolving digital ecological networks are many of the open questions concerning the coevolution of ecological interactions in multispecies communities. For example, do coevolutionary dynamics change as communities become richer? Is there any limit to their richness? Is the evolution of interactions between multispecies networks historically contingent Why do some ecological scenarios lead to predictable network structures and others do not [159]? Do genetic

constraints play a large role in the evolution of ecological networks? These are only a few of many open questions concerning the coevolution of ecological interactions in multispecies communities.

These and many related questions require researchers to look across the evolutionary history of ecological network formation. For natural systems, those data are very difficult to collect. With digital organisms, watching both the coevolutionary process and ecological network formation is possible in real time. Data on the abundance of interacting phenotypes are recorded without error; hence, the evolutionary implications of ecological processes can be explored in-depth.

The study of self-replicating and evolving computer programs offers a tantalizing glimpse into the evolution of interactions among organisms that do not share any ancestry with the biochemical life of Earth. This comes with potential caveats in translating predictions of evolving digital networks to biological ones because mechanistic details differ substantially between interacting digital organisms and interacting biological organisms. Nevertheless, these digital networks contain the necessary components for ongoing coevolutionary dynamics in large webs of interacting organisms. In spite of the differences between biological and digital evolution, the study of evolving digital ecological networks can lead to a more predictive understanding of natural dynamics. Because the general operational processes (e.g., Darwinian evolution, mutualism, parasitism, etc.) do not differ, studies utilizing digital networks can uncover rules operating on and within ecological networks. Together with microbial experiments, they create opportunities for furthering the understanding of the interplay between ecological and evolutionary processes among interacting species.

Chapter 5

Coevolution of Nested Communities

Authors: Luis Zaman, Miguel A. Fortuna, Charles Ofria

5.1   Introduction

Coevolution rarely occurs between isolated species; it is often embedded within a complex web of biotic interactions [11, 165]. With the incorporation of tools developed by the growing field of network science, ecologists have been able to quantify the structures of ecological communities [114]. By doing so, they have identified some seemingly general patterns in the structure of food webs and, more recently, bipartite networks [10, 127]. Host-parasite and host-mutualist communities are prime examples of bipartite networks, where there are two *types* of nodes (organisms), and edges (interactions) only occur between different types. In other words, pollinators can only pollinate plants, not other pollinators.

Nestedness is a frequently observed pattern in host-parasite webs, where parasite *species* (or phenotypes, more generally) with a narrow host-range interact with a subset of the hosts infected by parasites with broader host-ranges. A meta-analysis of bacteria-phage communities found that they were often nested (27/38), and several of these networks represented coevolved communities starting from a single host and parasite genotype [46]. This structure has been shown to reduce the effects of competition, as well as maintain more diversity than in unstructured communities [12].

Thus, nestedness appears to be more than the consequence of probabilistic interactions between species.

Why are host-parasite networks so often nested? Several mechanisms have been proposed as explanations for nestedness, such as physical constraints on the mechanism of interaction (e.g., running speed in predator-prey interactions), community-level benefits such as ecological stability, an ongoing process of coevolution which only appears nested when viewed statically, and finally that the nested structure is a byproduct of a neutral process operating on species' abundances [10, 46, 88]. While none of these mechanisms excludes the others, the prevalence of nested communities suggests the mechanisms are also quite general. Coevolution is a universal process that is responsible for both "growing" and constantly pruning the complex ecological networks we observe. Could nestedness be an outcome of the coevolutionary process in general, rather than specific genetic or physical constraints?

Although community level effects are important for the maintenance of ecological networks, they are unlikely to explain the origin of nestedness in isolated coevolving communities like several of the bacteria-phage networks in [46]. One of the few studies of dynamic bipartite network formation monitored plant-pollinator interactions over a relatively short period of time (two seasons) [123]. Although they found that abundance and phenophase were sufficient to explain the level of nestedness observed, this mechanism cannot be separated from the evolutionary origin of the plant-pollinator interactions in the first place. Additionally, abundance may be the result of nested interaction networks rather than their cause. In order to tease apart the ecological and evolutionary mechanisms shaping nested interaction networks, we need a tractable system that allows us to gather high-resolution interaction data over many generations while still being open-ended so that nestedness is not a "built-in" feature.

The Avida digital evolution framework provides us with these features as well as many others [121]. Host and parasite organisms are self-replicating computer programs living in a simulated environment [184]. They can interact with their environment, other organisms, and their own internal state through executing simple assembly-like instructions. While we have essentially

perfect knowledge of digital organism biology, their complexity and ability to interact with their biotic and abiotic environments make Avida an astonishingly rich open ended system for addressing evolutionary questions [51]. With Avida, we can watch while populations initiated with a single host and parasite genotype coevolve for thousands of generations into complex communities of interacting phenotypes, all in a matter of hours. We can save data about every host and parasite phenotype, and their interactions over time. It is this ability to collect such detailed data that allows us to answer questions regarding the evolutionary formation of ecological networks, and what is shaping their structure. [51].

Here we investigate the structure of diverse host-parasite communities that coevolve in Avida. We find that they are significantly more nested than expected by chance. To test if the process of a network growing from a single interaction into a full complex web is responsible for some or all of the nestedness we observed, we developed new process-based null models. These models generated significantly nested networks, but did not account for the level that coevolved. Next, we turn our attention to abundance based drivers of nestedness. We argue that this is a potentially flawed explanation of networks' nestedness because abundance could itself be a product of the community and its structure. We show that abundance indeed produces extremely nested communities, as the literature suggests. However, when information about which phenotypes can and cannot interact is included, abundance no longer accounts for the level of nestedness that coevolved.

5.2 Material and Methods

5.2.1 Avida

Avida runs were configured identically to those in Chapter 3, except the runs were shorter (50,000 updates) and data were collected much more frequently.

Every 50 updates, the location and phenotype of every host and parasite organism was recorded. Co-occurring host and parasite phenotypes thus represent an active infection. By enumerating the host-parasite co-occurrences, we compiled a quantitative interaction network of unique host and

parasite phenotypes. Host and parasite abundances were similarly tabulated. This quantitative interaction network can be represented by a $M \times N$ matrix by considering each row a unique host phenotype from the set of M host phenotypes, and each column a unique parasite phenotype from the set of N parasite phenotypes. The value at (m_i, n_j) thus corresponds to the number of host organisms with phenotype i that are infected by parasites with phenotype j. This quantitative matrix can also be transformed into a binary *incidence* matrix, where all non-zero values are replaced with a value of 1.

We processed the time series of quantitative interaction networks to generate a new series of network *events*. We compared each pair of consecutive timepoints to calculate the number of hosts, parasites, and edges that were added or removed in each 50-update window.

5.2.2 Nestedness Calculations for Incidence Networks

Calculations of nestedness are performed using the Nestedness measure based on Overlap and Decreasing Fill (NODF) metric [6, 169]. Each row and column of the incidence matrix is sorted by the total number of unique interactions. Thus, the matrix is rearranged to create the highest density of interactions in the top-left corner (and, therefore, the maximum amount of nestedness as calculated by NODF), while preserving the identity of edges. NODF compares all-pairs of rows and columns, and scores the pair based on their degree of overlap. A perfect score is given to a pair when the more specialized species is a subset of the more generalized species. Thus, intermediate scores are calculated based on the proportion of overlap between rows or columns. The scores for rows and columns are normalized by the size of the matrix and combined when the final NODF value is calculated.

5.2.2.1 Incidence Null Models

To test if an interaction matrix had a significant value of NODF, we compared the empirical value with a null distribution. This test requires a method of generating null interaction matrices from our empirical ones. The standard model generates null networks by randomizing the identity of edges

while keeping the total number of interactions the same. We refer to this as the *fill* null model, since it maintains the empirically observed matrix fill. Unless stated otherwise, we used 1000 randomly generated networks to calculate null distributions.

When we begin considering how interaction networks grow over coevolutionary time, the typical fill null models no longer capture all the relevant information; every time point is considered entirely independent from the others. In reality, there is a great deal of interdependence in the sequence of coevolved interaction networks, which is the product of a process that adds and removes nodes as well as edges.

To account for this interdependence, we developed two null models using process-based Monte Carlo simulations of network growth. The first recapitulates the exact sequence of network events (e.g., adding host nodes, removing edges) that occurred in the coevolving communities. However, the identity of nodes and edges are not preserved, thus producing a unique interaction network every time one is grown. This method provides us with a time series of NODF values from networks that share an evolutionary history, similar to those observed in coevolving networks. We refer to this method as the *event* null model.

The second growth model aims to capture the same evolutionary history, but without detailed information about the exact sequence of network events. We realize that computational systems like Avida provide an incredible amount of information, but most coevolutionary systems will not be so amenable to our event null model. Instead, we can grow networks assuming that they grow with monotonically increasing numbers of hosts (M), parasites (N), and edges (E) over some number of timesteps (t_m). Given a quantitative interaction matrix and t_m, this method adds hosts, parasites, and edges every timestep based on a random draw from a Poisson distribution parameterized by the expected number of events per timestep. The simulation is continued until all nodes and edges have been added. We refer to this method as the *monotonic* null model.

Figure 5.1 shows that the level of nestedness produced from the monotonic null model is fairly robust to the value of t_m used.
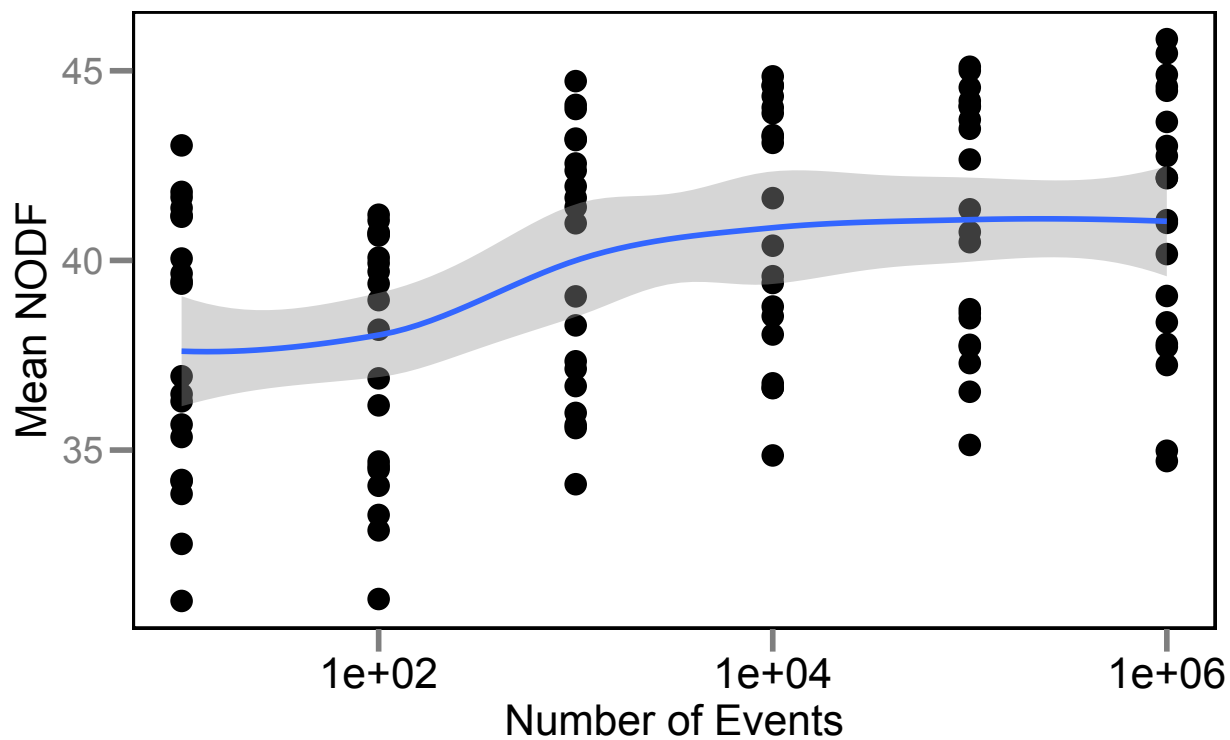
Figure 5.1: The effect of simulation length on *NODF* values. While the effect is significant, it is relatively small. Nevertheless, we use a value of 10,000 for all our analyses since it is past the inflection point. The blue line is a local regression function (loess) and the shaded region depicts confidence in the regression.

5.2.3 Nestedness Calculations for Quantitative Networks

We used the WNODF metric to calculate nestedness values in quantitative networks [7]. This metric is similar to the NODF value introduced above, but is *weighted* by the proportion of interactions that follow the expected quantitative decrease in the number of interactions as species become more specialized. Thus, nestedness is maximized when the most generalist species also have the most frequent associations with their partners.

5.2.3.1 Quantitative Null Models

Similar to NODF, statistical tests of WNODF require a null distribution, and thus a method of generating null networks. We developed a simple Monte Carlo method to generate quantitative networks. Our method requires a vector of host densities (d^m), a vector of parasite densities (d^n), the total number of interactions (q), and a matrix (S) that describes the probability of a successful infection for all pairs of $M \times N$ host and parasite phenotypes, where $s_{i,j}$ is the probability that host phenotype n_i and parasite phenotype m_j interact. In the simplest case, setting S to the all-ones matrix would generate null networks based purely on abundances because all host and parasite phenotypes are equally able to interact. The inclusion of an explicit S matrix thus allows us to generate networks that take into account information about infection mechanisms and genetics when constructing our null networks. While we use this S matrix to represent infection mechanisms here, it is a generic probability matrix that can be used to capture spatial, temporal, or other ecological information that affects the likelihood of infection.

A quantitative null network is thus generated by repeatedly sampling a host and parasite phenotype based on their densities and connecting them with a probability of $s_{i,j}$. If the interaction is added, the host and parasite densities are updated to reflect their removal from the available pool. This process is repeated until all q edges have been added. However, with a sufficiently constrained S matrix, it is possible to run out of potential interacting pairs before adding all q edges. To prevent deadlock, we impose a maximum number of repetitions ($100 \times q$), after which the random matrix is returned.

5.3 Results and Discussion

5.3.1 Digital Host-Parasite Coevolution Produces Nested Interactions

We tested the final interaction networks (update 50,000) for nestedness by comparing the empirical *NODF* value to its corresponding distribution of *NODF* values obtained using the fill method to generate null interaction networks. Consistent with natural bipartite networks, we found a substantial level of nestedness in each coevolved replicate ($p < 0.001$ by Monte Carlo Simulation, Figure 5.2).
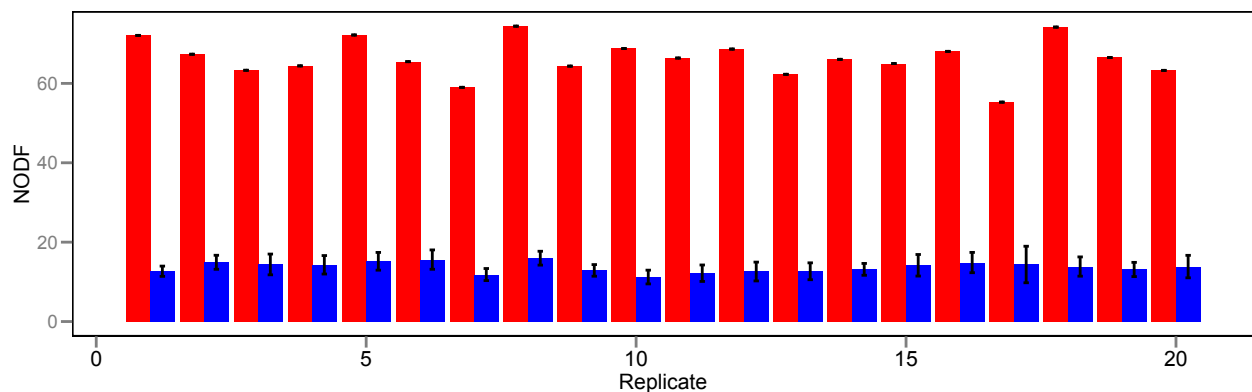
Figure 5.2: NODF values observed from coevolved (red) networks and the null networks (blue) using the fill method. Error bars depict 2x standard deviation.

One potential explanation for the level of nestedness we observed is that the phenotypic mechanism that determines if a particular host can infect a particular parasite is itself nested. To test this, we used the same fill method to generate a null *NODF* distribution for the network composed of all possible interacting host and parasite phenotypes (511×511). We found that the Avida network is actually slightly less nested than expected by chance ($p < 0.001$ by Monte Carlo Simulation). Thus, the evolution of nestedness is not due to a random set of phenotypes interacting through an inherently nested mechanism.

5.3.2 Growing Networks Produces Nested Interactions

Although the complete interaction network is not nested, coevolution still produces nested networks. Perhaps the process of growing a network from a few interactions into a full web of host and parasite phenotypes creates significantly nested structures. We tested two different ways of growing networks, the event and monotonic methods (Section 5.2.2.1).

Both the event and monotonic method produced significantly nested networks (Figure 5.3 and 5.4. This demonstrates that coevolution, in general, produces nested communities even in the absence of ecological and genetic details. One reason these growth processes produce significant levels of nestedness is that older nodes have had more chances at being randomly assigned edges. Whether or not older species also have more interacting partners in nature is an empirical question, which can be answered by combining phylogenetic information with interaction data.
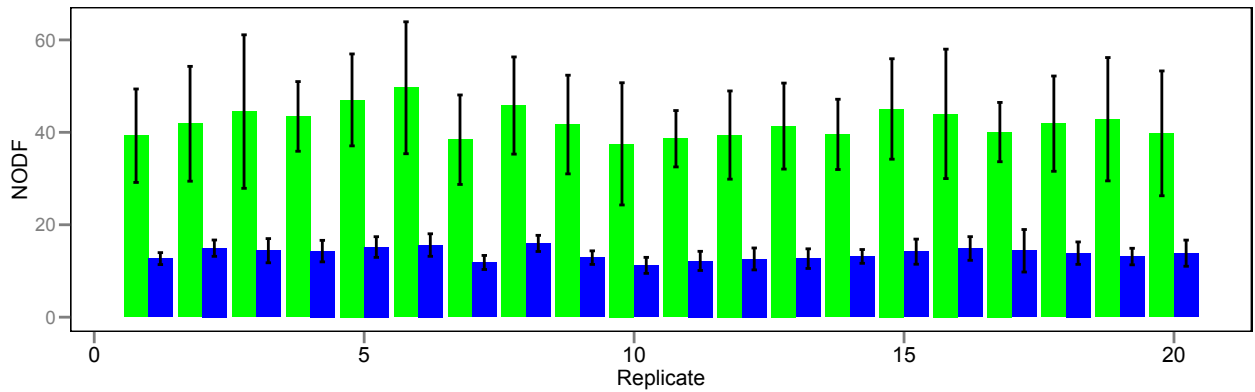
Figure 5.3: NODF values from the final monotonically grown network (green) compared to the fill method (blue). Monotonically grown networks are significantly more nested than the randomly shuffled networks traditionally used to generate null distributions. Error bars depict 2x standard deviation.

Interestingly, the monotonic growth method lead to higher values of $NODF$ than the event based method Figure 5.4. Fortunately, this is also the method that is more generally useful. It is rare that an empirical system provides enough detail to generate a sequence of network events, but the monotonic method simply requires an empirical network and the length of time to run. In addition, this method is fairly robust to changes in simulation length (t_m), which effectively changes the

number of nodes and edges added to the network in each timestep (Figure 5.1).

These methods for generating growing networks allow us to analyze the trajectory of nestedness over time. They also provide a new null model that predicts significant levels of nestedness from network growth processes, like the one present during coevolution. Using just the traditional fill based null model would have led to a significantly over represented level of nestedness in our system.

5.3.3   Coevolved Networks are Still Nested

Even after using our most conservative method to generate a null distribution, the coevolved networks are still significantly nested (Figure 5.4, $p \ll 0.001$, Two Sample $t = 17.7$). Additionally, it appears that coevolution is continuing to increase nestedness, while the null methods are leading to decreased nestedness. Thus, simply considering a dynamic network of host and parasite phenotypes and their interactions is not sufficient to explain the structure of coevolved networks. Coevolution, of course, is more than just a random process of adding organisms and interactions. There are complex frequency-dependent interactions (Chapter 3) which are responsible for the evolution and maintenance of numerous host and parasite phenotypes (Chapter 2). There are also likely trade-offs and constraints that contribute to the phenotypes likely to evolve [98]. While these go beyond the scope of a null model, *per se*, our dynamic models can incorporate more information about the ecology of host-parasite interactions.

5.3.4   Abundance as a Driver of Nestedness

Another mechanism potentially explaining the pervasive nested structure is that species abundances combined with random interactions often lead to nested communities. The commonly observed species richness distributions (exponential, log-normal, broken stick, etc. [107]) have a long tail of rare species, and would lead to a densely connected core with rare species (specialists) much more likely to interact with abundant ones (generalists). Indeed, this has been theoretically demonstrated using Monte Carlo simulation and analytical derivations [9, 88, 170]. Non-manipulative observations
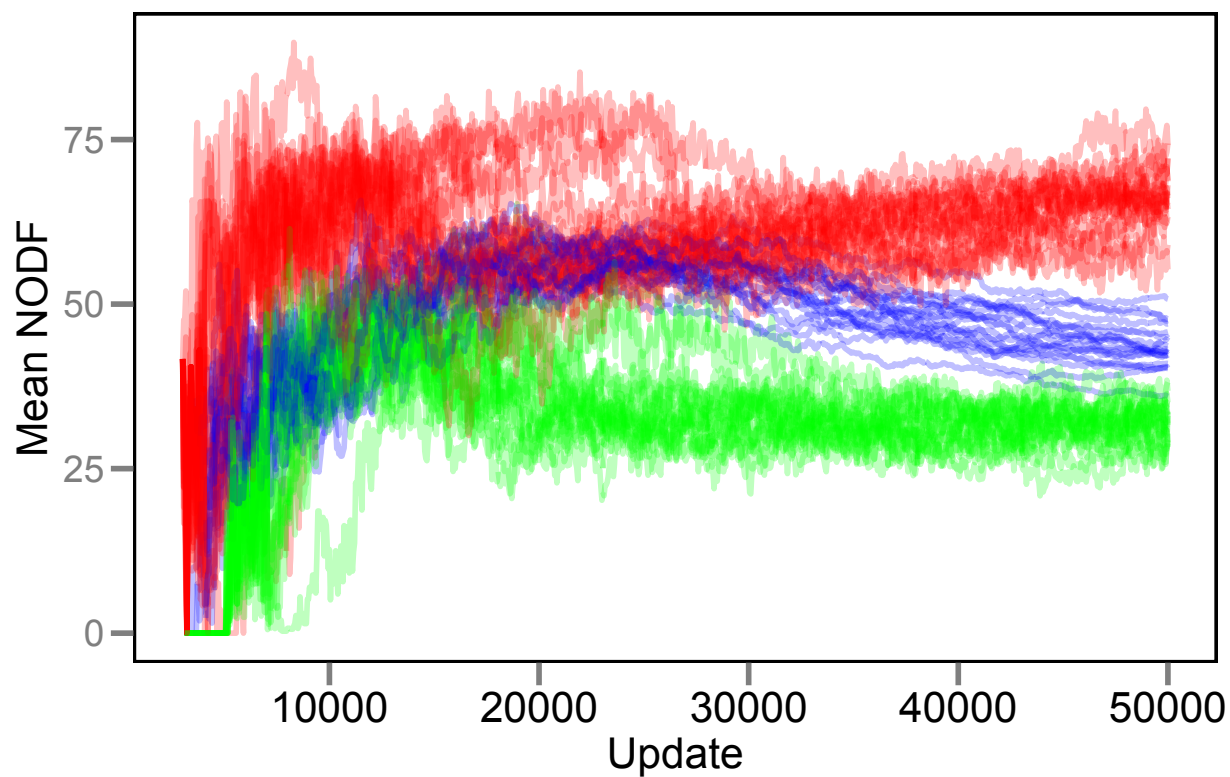
Figure 5.4: Empirically coevolved communities (red) are still more nested than grown networks using either the monotonic (blue) or event (green) method. Despite the significant level of nestedness we observed by growing networks rather than shuffling edges, the empirical communities were still significantly nested. Empirical and event values are calculated every 50 updates, but the monotonic values are saved every 10 timesteps and are trimmed to the shortest length time series obtained.

have also corroborated this mechanism [23, 32].

However, assuming that the species richness distributions are the cause of nested networks is a potential logical flaw. It seems likely that these distributions could be the product of nested networks rather than their cause. With Avida, we have previously shown that parasites are indeed responsible for the maintenance of host diversity (Chapter 2). Although these experiments are carried out in idealized communities, which are not embedded within larger ecological networks, they suggest that abundance is driven by the community and its structure rather than the other way around.

In addition, abundance produces strongly nested networks when links are proportionally assigned to hosts and parasites, but this proportional mechanism ignores any genetic and phenotypic information about which organisms can interact and which cannot (i.e., forbidden links) [82, 122]. Obtaining data about these forbidden links is technically challenging and often not directly measurable [172]. However, we know what phenotypes can possibly interact in Avida.

We compared the quantitative nestedness metric, WNODF, measured for our empirical network with null networks generated from the host-parasite abundance data. We either generated null networks purely based on abundances, or used information about which phenotypes cannot interact by providing our null model with appropriate S matrices. Similar to previous studies, we found that networks constructed while only considering abundance data generate highly nested networks. In fact, these networks are significantly more nested than our empirical ones ($p \ll 0.001$, Two Sample $t = -7.9$). However, once we take into account the phenotypic mechanism that determines successful infection attempts in addition to the abundance data, our empirical networks are significantly more nested than expected ($p \ll 0.001$, Two Sample $t = 9.5$). This result suggests that abundance may have a role in determining the degree of nestedness in bipartite networks, but its importance is overrated due to the lack of direct information about forbidden interactions.
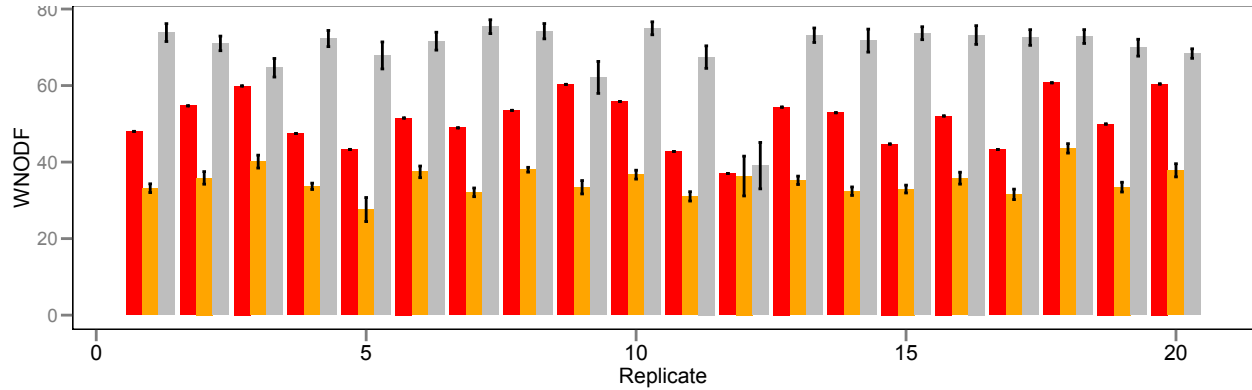
Figure 5.5: Networks constructed using abundance data with completely neutral interactions (gray) or with forbidden link information (orange). Empirical networks (red) are significantly less nested than networks with neutral interactions, but are significantly more nested than networks that take into account forbidden links. Error bars depict 2x standard deviation.

5.4 Conclusion

Together, our results suggest that the general and universal process of coevolution is likely responsible for significantly nested host-parasite networks. Exactly which aspects of coevolution are responsible are still elusive; However, we have ruled out several mechanisms that consider nestedness an incidental rather than primary outcome in our coevolving networks.

The methods presented here represent novel ways of accounting for dynamics in coevolving interaction networks. While we apply these methods to Avida, a somewhat unnatural system [94], we hope that ecologists and evolutionary biologists will find our methods broadly useful.

Chapter 6

Evolution Along the Parasitism-Mutualism Continuum

Authors: Luis Zaman, Justin R. Meyer, Charles Ofria, and Richard E. Lenski.

6.1 Introduction

Symbiotic interactions do not always fall neatly into discrete parasitic or mutualistic categories. Interactions exist on a continuum with virulent parasitism and obligate mutualism at opposite ends. In addition, their position along this continuum need not be fixed. The outcomes of many symbiotic interactions are entangled in their environmental context. For example, plant-associated rhizobial mutualists may become parasitic in nitrogen-rich soil, and costly plasmids carrying accessory genes for antibiotic resistance are rapidly lost in the absence of selection yet abound in nature [69,100,118].

In addition, symbiotic interactions are evolutionarily labile. There are many examples of mutualism breakdown, where altering environments or transmission modes can have dramatic evolutionary consequences on symbiotic interactions [125, 139]. In one outstanding example of a mutualism arising from parasitism, the long term tracking of *Wolbachia* populations identified a rapid shift, where fecundity of infected *Drosophila simulans* went from having a 15%-20% fecundity reduction to a 10% fecundity advantage [175]. The relative paucity of parasitism-to-

mutualism examples is not due to their rarity in nature. Indeed, phylogenetic analyses have identified several independent mutualistic clades that evolved from parasitic origins [138]. A more recent and comprehensive analysis found that the majority of proteobacterial mutualists are more likely to have evolved from parasitic ancestors rather than free-living ones [140].

Although observing these transitions is challenging, microbial evolution provides an opportunity to empirically study the evolutionary flexibility of the parasitism-mutualism continuum. Sachs and Wilcox manipulated a jellyfish mutualist's transmission mode from primarily vertical to horizontal and observed a resulting shift to parasitism [141]. J.J Bull *et al.* manipulated the level of partner fidelity in a non-lethal filamentous phage infecting *E. coli* and observed a reduction in the cost of infection when vertical transmission was allowed [22]. The importance of spatial structure for the evolution of more prudent exploitation and even cooperation has been experimentally demonstrated as well [67, 84]. These results reinforce the view of symbiotic interactions as dynamic, environment-dependent, and the targets of ongoing selection. Still, few studies have directly manipulated the environment in a context-dependent interaction to test if evolution reinforces the interaction's new position on the parasitism-mutualism continuum (e.g., [19]).

Here we investigate evolution and coevolution along the parasitism-mutualism continuum with bacteriophage λ and its *E. coli* host. Several factors make phages, and specifically λ, a useful system for this investigation, including rapid generation times, large potential for evolution, and well-characterized biology. Briefly, we used an engineered phage containing the lacZα subunit along with a bacterial host that had a deficient allele of this subunit (REL606 lacZ$^-$). In the absence of phage, the ancestral host is unable to consume lactose in the growth media. Association with phage may complement the host's deficient machinery, enabling lactose metabolism. However, phage are typically considered deadly parasites; how could this association become mutualistic? Crucially, λ can employ either a lytic or lysogenic lifecycle (Figure 6.1A) and, evolution is able to tune the switch between these alternate strategies [135]. Lysogens, bacterial hosts infected with a dormant phage genome (prophage), thus have access to the lacZα subunit. Indeed, lysogenic phage in nature play several mutualistic roles. Prophage can function as a toxin to sensitive (i.e., non-lysogenic) bacteria

76

when they stochastically enter the lytic cycle, which enables lysogens to invade non-lysogenic populations [137]. From the bacterial perspective, phage encoding virulence factors that facilitate colonization of a human host are mutualists [20]. Recent work has also demonstrated that phage in the mammalian gut can play a mutualistic role with their bacterial partners by providing mechanisms for the rapid spread of antibiotic resistance genes after treatment [115].

In this system, λ prophage are maintained in their dormant state by the Repressor protein (the product of gene *cI*). When Repressor is bound to three operators ($O_R\{1,2,3\}$), most of the prophage's genes are silenced. Interestingly, the same mechanism that maintains prophage integration also prevents coinfection since infecting phage DNA is bound by Repressor before it can be circularized (Figure 6.1B). Lysogeny is a complex and curious trait of many bacteriophage, and is itself environmentally dependent [85, 90]. Although much is known about the genetic mechanisms of lysogeny, the conditions that favor its evolution and maintenance are still mysterious [154].

We performed evolution experiments where the evolving phage population was diluted and transferred every 48 hours into a fresh culture of ancestral hosts. We also performed coevolution experiments where both bacteria and phage were diluted into fresh media every transfer. Contrary to our hypotheses, we found no signs of mutualism in the evolution experiments. In the coevolution experiments, we did see mutualism arise, but it was short lived. Further investigation suggests that novel cheaters evolved in both cases. In the evolution experiment, phage evolved the ability to infect the otherwise resistant lysogens. In the coevolution experiment, hosts stole the lacZα subunit from the phage rendering the association unnecessary.

6.2   Methods

6.2.1   Media

Unless otherwise stated, all cultures were grown in 50ml Erlenmeyer flasks containing 10ml of media. Flasks were incubated at 30°C and were shaken at 120rpm.

The abbreviation LB refers to Luria-Bertani media in this manuscript. Our primary experimental
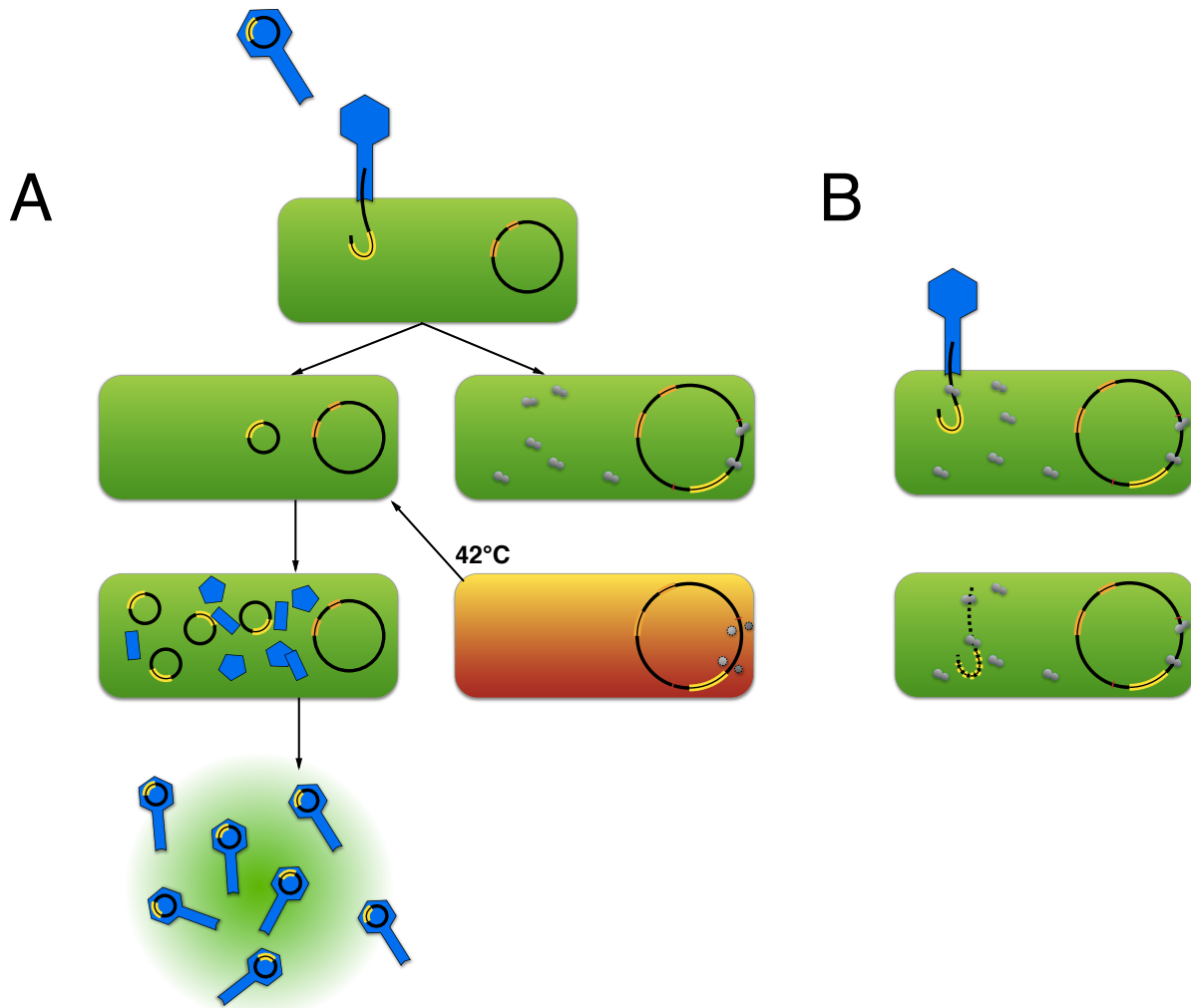
Figure 6.1: Lambda phage life history. **(A)** When a λ phage particle binds to an *E. coli* cell and ejects its DNA, it can enter into either the lytic (left) or lysogenic (right) cycle. In the lytic cycle, the phage chromosome is circularized, the host's machinery is hijacked to replicate the phage DNA and produce new virions. In the lysogenic cycle, the phage's DNA is integrated into the host chromosome and remains dormant through the binding of several Repressor proteins. However, our phage ancestor has a temperature sensitive cI gene (the gene that is responsible for Repressor) that degrades at high temperatures. When Repressor degrades, it induces the dormant prophage into the lytic cycle. **(B)** Lysogens are resistant to coinfection because infecting phage DNA is bound by extra Repressor proteins and fails to circularize or integrate into the host chromosome.

media (mM9) was M9 [142] supplemented with 4% v/v LB broth without added NaCl and 1g/L MgSO$_4$. The small amount of LB was added so bacteria could start growing in the media with lactose since λ infection only succeeds when *E. coli* is metabolically active. The high concentration

of $MgSO_4$ was added to improve phage growth [110]. Sugar was added at a concentration of 1g/L of either lactose ($mM9_L$) or fructose ($mM9_F$). When we revived strains, we inoculated them in M9LB, a rich media made by mixing M9 salts with LB that lacks NaCl.

LB agar plates were used extensively and were often supplemented with additional ingredients. Ampicillin supplemented LB plates had a concentration of 100ug/ml (LB+Amp). We used LB plates supplemented with X-Gal and IPTG to quantify the proportion of lactose metabolizing colonies (LB+X-Gal) [149], and minimal lactose plates (ML) were used as a positive screen. Tetrazolium maltose plates were used to quantify the proportion of λ resistant colonies since the most frequently observed resistance mutations eliminate *E. coli*'s ability to consume maltose [105, 110]. Phage were suspended in a layer of soft agar (LB with half the concentration of agar, SA) atop LB plates.

6.2.2 Bacteria and Phage Strains

Bacterial cultures were cryopreserved in 1ml aliquots supplemented with 20% glycerol by volume and frozen at -80°C. Cultures were revived by inoculating M9LB media with a scraping of the frozen culture and growing overnight at 30°C. Phage stocks were isolated from revived lysogens with the protocol described bellow.

All experiments were performed using a modified strain of REL606 (*E. coli* B). We replaced the wild type lacZα with a deficient copy into the REL606 ancestor, denoted REL606 lacZ$^-$ here, through "gene gorging" [70]. The deficient lacZα allele was moved from the plasmid pSwtRlacZwhiteRz, which contains 5 alanine substitutions that disrupt the dimer- and trimerization of β-gal [83, 148] We confirmed REL606 lacZ$^-$ was lactose deficient by monitoring for growth in liquid media, solid media, dilute rich media supplemented with large amounts of lactose, and X-Gal indicator plates. In addition, spontaneous recovery of lactose metabolism was never observed, likely because recovery requires several reversions.

We were able to induce the prophage into the lytic cycle by heat shock because our phage has the *cI857* allele, and thus produces a Repressor protein that is unstable at high temperatures. Induction is therefore performed by first growing a culture to exponential phase at 30°C, heat shocking the

culture in a water bath set to 42°C for 30 minutes, and then allowing the induced prophage to complete lysis at 37°C for 45 minutes. After the induction cycle is complete, a 1ml aliquot is treated with 10ul of chloroform and spun down at $16,873 \times g$ for one minute. The supernatant is reserved as a phage stock. Any reference to phage stock was prepared using this outlined method.

During lysogeny, most phage genes are silenced [90]. Thus, while lysogens have access to the lacZα subunit, it may not be expressed at substantial levels. To isolate a phage that conferred active expression of lacZα during lysogeny, we first made a phage stock from *E. coli* lysogen SYP045 [148], which contains a thermally inducible lambda prophage (*cI857*) with an *R::lacZα* fusion. Because lacZα in this phage is fused with the R endolysin gene, it is only highly active during the final stages of lysis [174]. We thus performed additional recombination and selection steps to generate phages that provided more active lacZα expression.

To generate recombinant phage, we transfected REL606 lacZ^{-} with the high-copy number pCR 2.1 plasmid containing an active lacZα subunit and lac promoter. We screened potential transformants on LB+Amp plates and isolated a viable colony. We then inoculated a culture of transformants in M9LB and added 100ul of the phage stock induced from SYP045. By co-culturing transformants and lysogenic phage overnight at 30°C, we encourage the recombination of the phage lacZα sequence with the region in the high-copy pCR plasmid. After overnight growth, a new phage stock was made from the transfected lysogens following the heat-shock protocol above.

We treated 1ml of phage stock with 10ul of DNase for 24h to destroy any remnant plasmids that may have contaminated the phage stock. We then used this treated phage stock to infect replicate populations of REL606 lacZ^{-} in mM9$_{L}$. This additional step of growth in mM9$_{L}$ selects for phage able to successfully lysogenize their hosts as well as grow on the large amounts of lactose in the media. After the populations were visually more turbid than cultures without phage (approx. 48-72h), we streaked colonies on ML and LB+X-Gal plates to ensure we were isolating only lysogens capable of growing on lactose. We re-streaked all isolated lysogens to ensure we were not picking up any free phage particles. Finally, we haphazardly chose one of the lysogen clones, LZL107, to prepare phage stocks that would serve as the ancestor for all of our experiments.

6.2.3 Evolution and Coevolution Experiments

In the evolution experiment, we held the host constant and only allowed the phage to evolve. We maintained 10 evolving phage populations, 5 of which were in $mM9_F$, and the other 5 in $mM9_L$. Every 48 hours, we would induce all evolving populations by transferring 1ml of culture into tubes with 4ml of fresh M9LB and heat shocking it following the protocol above. Then, 100ul of the phage stock was used to infect a fresh corresponding flask of REL606 lacZ$^-$ with an initial host density of approximately 5×10^7 colony forming units (cfu).

We also maintained 10 populations for the coevolution experiment, with 5 in $mM9_F$ and 5 in $mM9_L$, but instead of transferring only the phage population, we transferred 100ul of the mixed population into fresh media (i.e., we performed a 100-fold dilution). Every 5 transfers we froze a 1ml aliquot from each population of phage stocks from the evolution experiment and mixed populations from the coevolution experiment.

6.2.4 Plating Assay for Phage Density

Phage density was estimated by plating 2ul of a 1/10 phage stock dilution series onto a lawn of REL606 lacZ$^-$ hosts suspended in SA. The spot with the most clearly defined plaques was counted, and the density of plaque forming units (pfu) per ml was calculated.

6.2.5 Cost/Benefit Assay

We developed an assay to measure the operational cost and benefit of hosts associating with phage over time in the evolution experiment. The cost of association is apparent when we consider obligately lytic phages, where every infection results in host sterility and death. On the other hand, the benefit is easy to imagine with an obligately lysogenic phage: the prophage provides resistance to coinfection (even of lytic conspecifics) as well as access to an otherwise unavailable resource. However, a major complication of measuring the cost and benefit of associating with lysogenic phage is the dynamic nature of their interaction. A typical growth cycle includes a period

of exponential growth followed by rapid killing by the phage, and a later growth phase of lysogens.

Calculating the area under a growth curve (AUC) provides a single value that summarizes the periods of growth and death in a population. We can compare the values of AUC for populations with phage (AUC^+) and without phage (AUC^-), which thus provides information about the cost and benefit of the association. To tease apart part of the cost from the benefits, we can compare AUCs in the fructose environment to AUCs in the lactose environment. Since lactose is only consumable by lysogens, values of AUC^+ that are greater than AUC^- indicate a beneficial interaction. In the fructose environment, there are no metabolic benefits of phage association, so we can get a clearer picture its costs. We calculate the proportional reduction in AUC as $(AUC^- - AUC^+)/AUC^-$, thus normalizing the difference between growth curves by the value of AUC^-, which we expect to be the larger value, at least at first. If this proportion reduction value is positive, it represents a cost to the association. However, if this value is negative, it means the association with the phage produces a net increase in growth.

To measure growth curves, we used a Molecular Devices SpectraMax M3 configured to read absorbance at 420nm every 3 minutes for a total of 48 hours. We used 96-well microtiter plates with samples arranged such that each well with phage was adjacent to a paired well without phage which was used in calculating the integral ratio. This pairing effectively eliminates any systematic column or row effect.

Bacteria were inoculated at a density of approximately 5×10^7 per ml and phage were diluted to approximately 1×10^4 per ml.

6.2.6 Plating Assay for Lactose Metabolism and Phage Resistance

Lactose metabolism was scored by plating cultures on LB+X-Gal plates, allowing them to grow overnight at 30°C, and then leaving them at 4°C for another 24 hours. Colonies that were consuming lactose turn bright blue, while lactose deficient cells grow into white colonies.

We used TM plates to score phage resistance. This is possible because mutations that eliminate lamB, the porin that our phage infects through, also eliminate maltose metabolism. Although this is

only one way our *E. coli* strain can evolve resistance, it is repeatedly observed as the first step of coevolution with λ phage in previous experiments [105, 110].

6.3 Results and Discussion

6.3.1 Evolution Experiment

To empirically test our hypotheses, we performed a phage evolution experiment with two resource treatments at opposite ends of the environmental spectrum, where the major carbon source was either available to the ancestral host (fructose), or was only available to lysogens (lactose). Given our hypothesis that evolution reinforces conditional mutualisms, we would predict that costs of association would decrease when the phage was evolving in the lactose treatment, and increase when evolving in the fructose treatment. On the other hand, benefits should only increase in the lactose environment. Contrary to our expectations, we found essentially no change in cost or benefit in either treatment (Figure 6.2).

The lack of change in the fructose environments could simply be due to our ancestral phage starting off with nearly optimized lysogeny rates. However, the lack of change in the lactose treatment is more puzzling. It is unlikely that phage populations lacked variation for rates of induction given previous results showing that λ can evolutionarily tune its switch between lysis and lysogeny on timescales shorter than our experiment [135]. Another explanation is that our hypothesis does not take into account the full evolutionary potential of phage λ.

6.3.2 An Evolved Cheater

These complexities can be difficult to track down, but coincidentally we noticed that some of the evolved phage from later in the experiment were able to infect lysogens. Recalling Figure 6.1B, wild type λ confers lysogens with resistance to coinfection. However, a strain of λ able to productively infect lysogens was first described in 1954 ([74]). This *vir* phenotype requires 3 mutations that
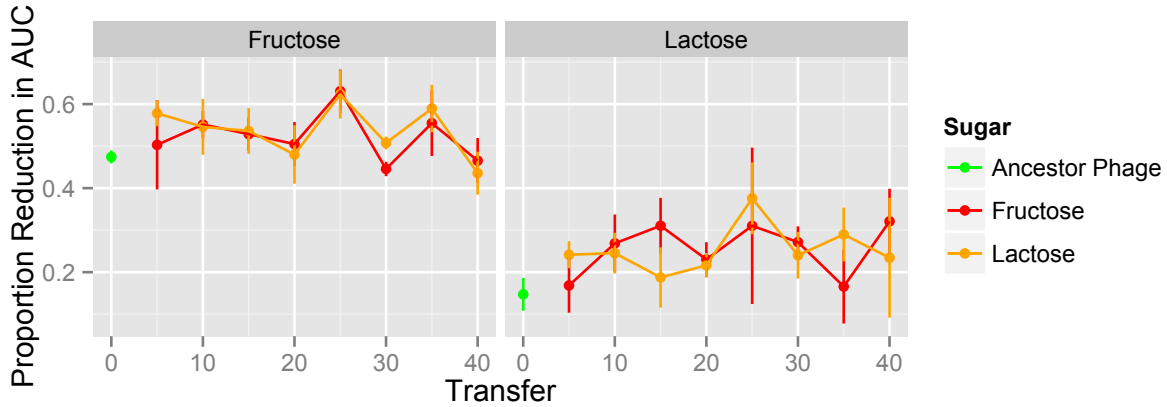
Figure 6.2: Cost and benefit of phage association assayed by growth curves in the evolution experiment. The evolved sugar is indicated by line color, and the assay sugar is indicated by the two panels. In the fructose assay environment, we are measuring the cost of phage association in the absence of any metabolic benefit. In the lactose assay environment, we are measuring the benefit of phage association. The green point indicates the ancestral phage values. Despite our hypotheses, we saw essentially no change from the ancestor. Lines represent the mean of the 5 replicates, and error bars depict $2\times$ standard error of the mean.

disrupt the binding of Repressor protein to operators $O_R\{1,2,3\}$. In 4/5 of the fructose replicates and all of the lactose populations, this λ *vir* phenotype was first observed between the 20th and 30th transfer, which is consistent with the difficulty of evolving multiple mutations.

Once this phenotype evolves, we no longer expect the evolution of more mutualistic interactions since they would fall victim to λ *vir*. Although the evolution of this complex *vir* phenotype is interesting in its own right, in the context our phage evolution experiment, it may technically be the *de novo* evolution of cheaters. However, λ *vir* evolves in the fructose treatment as well. Understanding the evolutionary dynamics that give rise to this phenotype is an interesting investigation, but it is one that must be saved for future work.

6.3.3 Coevolution Experiment

In the coevolution experiment, the association between bacteria and phage has the potential to be long lived. Where as, in the evolution experiment, the association between a prophage and its host

is severed every transfer. As long as a prophage is dormant, its association is maintained in the coevolution experiment. This provides the opportunity for a greater accumulation of benefit, as well as coadaptation between the phage and the host. However, this potential for coadaptation makes analyzing the cost and benefit of association combinatorially more difficult. While overall cost and benefit may be difficult to measure, correlated responses such as phage titer and the speed at which populations evolve resistance are readily observed. Results from the coevolution experiment indicate that, at least at first, the host-phage association was likely beneficial. The phage titers were higher in the lactose treatment, and despite this higher parasite load, resistance was slower to evolve (Figure 6.3). Additionally, the proportion of lactose consuming colonies rapidly increased in the lactose environment, but was never detectable in the fructose treatment (Figure 6.3). This suggests that lysis was the primary mode of interaction in the fructose environment, or that the phage quickly lost the lacZα gene.
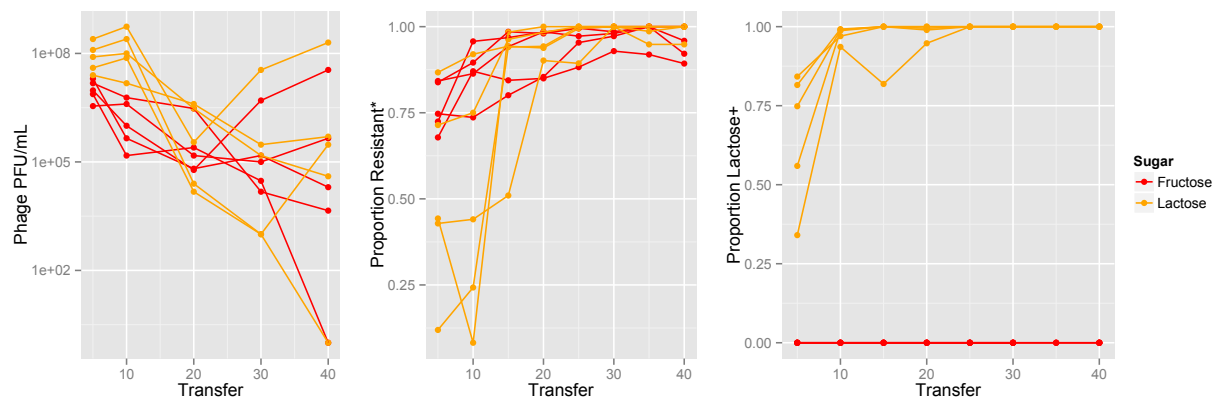
Figure 6.3: Phage titers and the proportions of resistant and lactose consuming cells in the coevolution experiment. Phage titers are higher in the lactose environment, consistent with a mutualistic interaction. In addition, resistance evolved more slowly and lactose consumption nearly fixes in the lactose treatment. Note that phage PFU is on a log scale.

6.3.4 A Coevolved Cheater

Although lactose consumption nearly fixed in the lactose treatment, the phage titer was erratic, and 4/5 replicates dropped by several orders of magnitude by the end of the experiment. Thus, the

association between the bacteria and phage became unstable. Confirming this, we isolated several clones from the end of the lactose coevolution experiment and found that, although they could consume lactose, they were no longer harboring inducible phage. This finding suggests that the prophage either was inactivated or that its lacZα region was recombined with the host gene, thus fixing our ancestor's deficient copy. In either case, the hosts effectively defected on the symbiotic relationship with their phage and became cheaters. Further analysis is required to uncover when this transition occurred and how it affected coevolutionary dynamics.

6.4   Conclusion

In some respects, the evolution of cheaters makes this story more interesting than it would have been if everything evolved as predicted. Cheaters, in the traditional social sense, are players that get the benefits of an interaction without paying the full (or any) cost. In the case of the λ *vir* phage, it is defecting on the association between lysogenic phage and their hosts rather than directly defecting on the host. We never observed any *vir* phenotypes in the coevolution experiment, perhaps because this extreme level of virulence is only beneficial when the association between host and parasite is fleeting. Although there is a constant inflow of sensitive hosts, it seems unlikely that the *vir* phenotype would have evolved in response to an advantage on these naive ancestral bacteria. For one, it requires several mutations that are exceedingly unlikely to occur together by chance. If, instead, these were individually beneficial mutations for phage growing on wild-type hosts, we would expect the *vir* phenotype to be commonplace. Thus, some aspect of the ecology created by lysogenic phage being transfered daily into fresh bacterial hosts creates a selective advantage for λ *vir*. Understanding what exactly these aspects are will be the focus of our future work with this system.

We uncovered two interesting cases of the evolution of *de novo* cheating. In the evolution experiments, the phage evolved a *vir* phenotype, removing the benefit from mutualistic association. In the coevolution experiments, the hosts "duped" their phage partners by stealing the lacZα subunit,

rendering the symbiotic interaction unnecessary. Although we did not predict these results, they are retrospectively unsurprising. The evolutionary maintenance of cooperation is an intensely studied subject, and our results reinforce the necessity of understanding its intricacies. That simple environments can give rise to complex ecological interactions over shot evolutionary timescales has been observed in many experimental evolution studies, and our results add to the growing evidence that community complexity is a general evolutionary outcome.

Chapter 7

Conclusion

Ecology's importance in evolution has been appreciated since Darwin's discovery of natural selection [31]. However, evolution affecting ecology is only recently becoming appreciated [24, 144, 183]. Host-parasite coevolution is a perfect example of how ecology and evolution are intimately entangled. However, coevolution is typically more complicated than the pair-wise adaptations often envisioned. Instead, interactions occur in complex communities that vary in time and space [80, 163].

In this thesis, I followed the arrows from ecology to evolution and back again. In Chapter 2 I showed how host-parasite coevolution drives diversification in a computational model system. Because a complex network of interacting host and parasite phenotypes arose, the ecological context of adaptation evolved into community ecology. In Chapter 3 I showed that this community context had substantial effects on further evolution in the hosts and, in this case, led to a trend of increasing complexity. These two chapters together show how evolution can influence ecology, and how new ecological conditions can influence further evolution. These results also lend insight to evolutionary computation, where incorporating more ecologically mechanistic and open-ended coevolution in evolutionary computation may produce far more complex solutions to problems than those typically evolved.

The community that formed played a central role and, interestingly, was not just a random assemblage of interacting host and parasite phenotypes. Instead, it exhibited a nested structure,

which has been observed in many natural communities [12, 46]. Despite its prevalence, understanding why nestedness arises has been challenging (Chapter 4) [10]. In Chapter 5 I investigated why nestedness occurs in this particular computational system. By developing novel null models, I demonstrated that coevolution as a process is responsible for the nested structure. Although the story is incomplete, I was able to make significant advancements by using explicitly dynamic models and the perfect information about phenotype interactions Avida provides.

In the first few chapters, I showed that parasites can be beneficial in an ecological and community context by favoring diversity, complexity, and structured communities. In Chapter 6, I aimed to study how evolution can move a deadly parasite along the parasitism-mutualism continuum, thus turning harmful interactions into beneficial ones. Instead of using Avida, the computational model system used in the previous chapters, I used a bacteria-phage model system. This chapter marked my introduction to the world of wet biology. Phage were an ideal system for this study since many relevant microbial traits are encoded by accessory genes, and temperate phage have well characterized genetic switches that determine if they destroy their host immediately or lie dormant. While similar experiments could have been carried out with Avida, the details of bacteria-phage interactions were particularly pertinent in this case. Indeed, these details ended up being important to the story that emerged. Instead of seeing evolution along the continuum as I predicted, the details of the bacteria and phage biology led evolution sideways. Chapter 6 ended up being a story about cheaters rather than mutualists, although the two strategies are inherently related.

I have used simple computational simulations, relatively simple models, complicated digital organisms, and microbial model systems in this thesis. Instead of choosing one to use exclusively, I have chosen a field to explore using the tools best suited to answer my questions. To build a cohesive body of work for this thesis, I necessarily left out several ongoing and completed projects using yet another tool. With Brian Connelly, I built an agent based simulation to investigate the role spatial structure plays in ecology and evolution [27–29]. My major computational contribution to this project was a dynamic programing method for building large random planar graphs with a specified expected neighborhood size in a reasonable amount of time. These graphs allowed us to

vary how far in space interactions could occur while maintaining two-dimensional geometry and the computational speed of explicit neighborhood lists. While much of the work I contributed was computational, the motivation was in answering biological questions.

All of the experiments presented in this thesis were done in simple well-mixed environments, but nature is not so convenient. Just as investigating temporal dynamics in coevolutionary processes is necessary to understand the outcomes (e.g., Chapters 3 and 4), investigating how structured environments alter processes and patterns will lead to a broader and hopefully more predictive understanding of coevolution. Every chapter of this thesis provides interesting followup questions using structured populations. In structured populations, is the diversity driven by coevolution with parasites split up into relatively homogenous patches with variation at the metapopulation level? Is there substantial local adaptation and does the level of maladaptation vary with the spatial properties of the patch (e.g., connectance and betweenness of patches)? Do we still see the coevolution of complexity when the parasite communities are variable in space, or do we see more fluctuations like we did in the absence of community effects? Is the network structure different within vs. between populations? Perhaps we will see nestedness within populations and a more modular structure at the metapopulation level. Does spatial heterogeneity promote or hinder the evolution of mutualism? Can we virulent and mutualistic phage coexist when the benefits of mutualism are patchy?

As I move on, I will study some of these questions using computational, mathematical, and microbial methods. Having just "gotten my hands wet" as Richard Lenski put it, I am spending a postdoc learning more about experimental coevolution with phage as part of Ben Kerr and Eric Klavins' Lab at the University of Washington. In addition to addressing some of these followup questions, I will be learning about new synthetic biology tools. These tools are enabling experiments that would have been previously impossible in natural systems. I hope that by combining the digital and computational approaches I have used in the past with these new biotechnology breakthroughs, I will continue to shed light on the coevolutionary process and its outcomes.

APPENDICES

Appendix A

Glossary of Cross-Disciplinary Terms

CPU Cycles - Central processing unit cycles are the fundamental unit of time for computers. Every cycle, the CPU executes a single instruction and continues to do so until the computer is powered off.

Memory Space - A simple type of *context*, which includes the computer instructions to be executed by the CPU as well as the values of local variables.

Thread - In its simplest form, a thread is a semi-independent series of instructions that can be executed along side of other threads. They are only semi-independent because threads could be executing the same set of instructions (the same program), or could be interacting through a shared memory space or message passing. The most basic way of *scheduling* threads is through a round-robin process where CPU cycles are distributed one at a time to each thread in turn.

Update - In Avida, time is measured in updates, a population-size dependent unit. An update represents enough CPU cycles for every individual in the population to execute 30 instructions.

Task - Digital organisms perform computational tasks by manipulating random 32-bit numbers. These tasks enable organisms to interact with resources and other organisms in their environment.

Coevolution - The reciprocal adaptation of one population to another.

Diffuse Coevolution - Coevolutionary adaptations to a suite of populations rather than just pair-wise interactions. Although some contention exists about diffuse coevolution rendering traditional pair-wise coevolution obsolete, the former is really just viewing the later in a more connectionist

context.

Complexity - This is perhaps the most difficult term to succinctly define, in part because there is no universally accepted definition. An intuitive metric I like and tend to use is the number of interacting parts. Although this does not set a threshold for when something goes from being simple to complex, it gives a quantitative scale that allows for comparisons.

Drunkard's Walk - In general, the drunkard's walk refers to a biased random walk. The story goes that the drunkard leaving the pub will eventually stumble his way into the gutter because every time he stumbles backwards, his fall is caught by the pub's wall. Stephen Jay Gould conjured the illusion of the drunkard's walk to argue that apparent trends of increasing complexity is due simply to a random walk where complexity is bounded below by the simplest living organism. Over billions of years and countless diversification events, average complexity will have had to increase.

Arms Race - Another common image used in biology, especially when talking about antagonistic coevolution. As one side builds up its armament, the other side must also in order to stay defended. This creates a positive feedback loop, or a "snowball" effect where both sides are racing to build more and more arms.

Appendix B

Introduction to Parasites in Avida

This introduction assumes some basic knowledge about Avida. In particular, familiarity with the basic organism and hardware will be very useful to have.

(Very) Brief Overview of the TransSMT Hardware

With that said, parasites currently do not work in the default hardware but rather one that supports better threading capabilities – the TransSMT hardware. The differences aren't huge, but they deserve their own documentation. Instead, I will just highlight major differences between the hardware types important for parasites, namely memory spaces and threads.

Memory Spaces   Memory spaces are regions of memory reserved for genetic instructions such as an individual's genome. To access these memory spaces, organisms execute the `Set-Memory` instruction followed by one or more `Nop` instructions specifying which space to use. In this hardware, the genome copy produced during self-replication must also be in a separate memory space, as well as any thread processes an individual spawns.

Threads   Threads in this hardware are distinct code sequences that are executed either in parallel, where all threads execute an instruction per CPU cycle awarded to an individual, or round-robin, where a single instruction from a single thread is executed per awarded CPU cycle and each thread

executes in turn. The number of threads an organism is allowed to have, as well as how they are scheduled is controlled by the following config options in the `avida.cfg` file.

- MAX_CPU_THREADS 1 - Maximum number of Threads a CPU can spawn
- THREAD_SLICING_METHOD 0 - 0 = One thread executed per time slice. - 1 = All threads executed each time slice.

Parasites in the TransSMT Hardware

Parasites in Avida are almost identical to hosts, self-replicating by copying their genome instruction-by-instruction into a new memory space. However, instead of dividing this new genome off into the world, parasites attempt to infect a **random** organism in its host's neighborhood (globaly if `BIRTH_METHOD=4` or `WORLD_GEOMETRY=7`, and honoring the `WORLD_GEOMETRY` if `BIRTH_METHOD` is set to any other value) with their offspring parasite genome, becoming a new thread on the host organism. Parasites attempt infection by calling the `Inject` instruction, which is also `Nop`-modified to identify the memory space the parasitic thread should occupy.

In order for infection to succeed, the host must be able to accept a new thread in the memory space the parasite is attempting to occupy. This means the host must have fewer than `MAX_CPU_THREADS` and that the host has not used the memory space specified by the `Inject` instruction. More than one parasite per host is not currently supported, thus we typically set `MAX_CPU_THREADS=2`. We can eliminate the effect of host-parasite coevolution via memory space allocation and specification (as well as any unforeseen side-effects such as parasites overwriting host offspring when specifying a particular memory space) by giving parasites memory spaces entirely separate from their host's (`PARASITE_MEM_SPACES=1`).

Parasites as well as hosts can perform logic tasks, and we can use their task-based phenotypes to implement additional mechanisms determining if infection will succeed or not. The config option `INFECTION_MECHANISM` already has several mechanisms implemented. The implemented options have the following behavior:

- 0 - Infection will succeed independent of task-based phenotypes

- 1 - Infection will succeed if the parasite and host have at least one overlapping task (Inverse Gene-for-Gene)

- 2 - Infection will succeed if the parasite does at least one task the host does not perform

- 3 - Infection will succeed if the parasite and host do the same tasks (Matching Alleles)

- 4 - Infection will succeed if the parasite performs all the tasks the host does as well as at least one additional task (Gene-for-Gene)

- 5 - Infection will probabilistically succeed based on the proportion of tasks that match between the host and parasite raised to a configurable exponent QMA_EXPONENT. (Quantitative Matching Allele)

To have more control over how many CPU cycles a parasite steals from it's host, PARASITE_VIRULENCE determines the probability that a CPU cycle will be given to the parasite. Thus, when this option is set to 1, the parasite is completely virulent and overtakes all of its host's CPU cycles. It is also possible to let the parasites evolve their own virulence by setting VIRULENCE_SOURCE=1, and choosing values for both VIRULENCE_MUT_RATE, which controls the probability of mutating a parasites virulence when a new parasite is born, and VIRULENCE_SD, which is the standard deviation of a normal distribution used to determine how much virulence changes when it mutates.

Events Typically Used With Parasites

InjectParasite is typically called near the beginning of a run to infect a range of cells. It takes a parasite organism file, the memory space label, and the range of cells which should be infected.

PrintParasiteTasksData and PrintHostTasksData print the tasks performed by parasites and hosts respectively. Similarly, PrintHostPhenotypeData and PrintParasitePhenotypeData split up the phenotype data, such as the Shannon Diversity and Richness of unique host and parasite phenotypes.
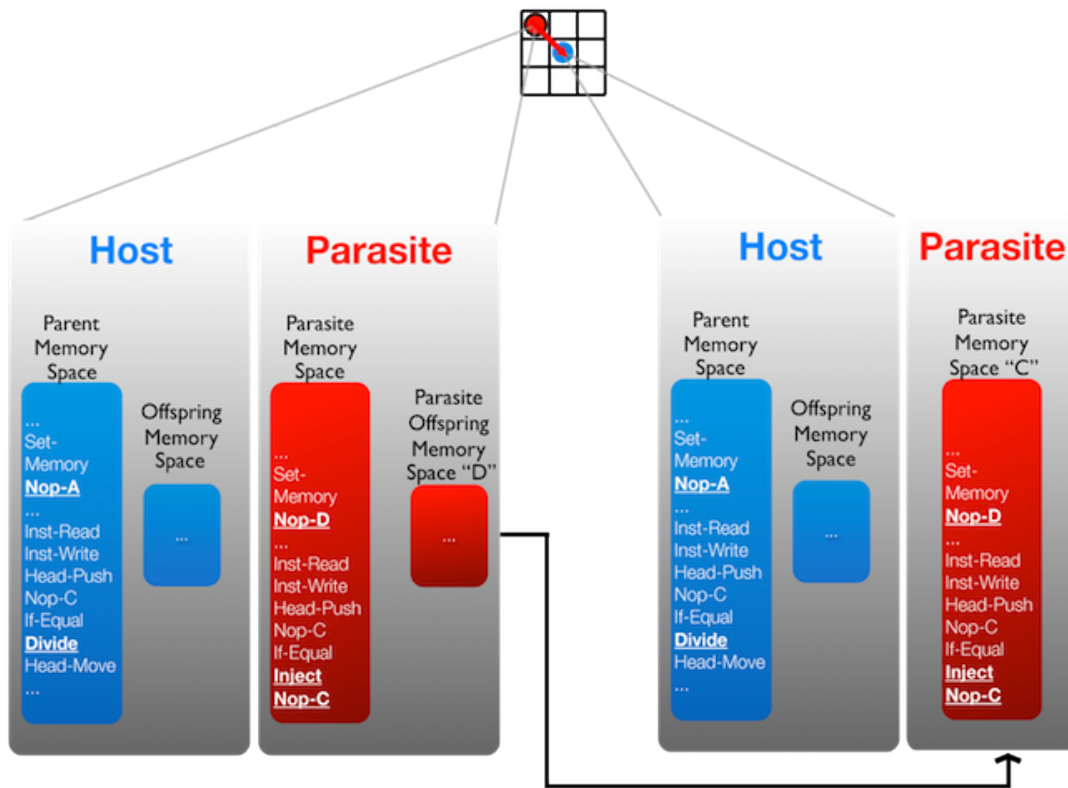
Figure B.1: Depiction of a Host-Parasite Interaction. Here, the original memory space allocation mechanism is depicted. The infected organism has a parasite thread that attempts to infect a host's "C" memory space, as indicated by the underlined sequence of instructions. Upon successful infection, the offspring parasite is copied into the newly infected hosts's memory. See Figure 3.1 for a depiction of the task based infection mechanism.

Typical Config Settings

- BIRTH_METHOD 4 - Population is well-mixed

- INJECT_METHOD 1 - Parasite thread is reset on successful infection

- MAX_CPU_THREADS 2 - Only allow one parasite per host

- INFECTION_MECHANISM 1 - Parasites infect hosts when they have at least one overlapping task

- PARASITE_VIRULENCE 0.8 - Parasites steal 80$ of their host's CPU cycles

- VIRULENCE_SOURCE 0 - Parasites use virulence value from config, instead of evolving it

- PARASITE_MEM_SPACES 1 - Parasites get their own memory spaces

- PARASITE_NO_COPY_MUT 1 - Parasites don't use copy mutation rates, so they can have independent mutation rates

- REQUIRE_SINGLE_REACTION 1 - Require hosts to perform at least one successful reaction to reproduce

A set of complete config files and ancestral organisms can be found at https://github.com/zamanlh/AvidaConfigs.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] C. Adami. What is complexity? *BioEssays*, 24:1085–1094, 2002.

[2] C. Adami, C. Ofria, and T. C. Collier. Evolution of biological complexity. *Proceedings of the National Academy of Sciences*, 97:4463–4468, 2000.

[3] C. Adami and C. O. Wilke. Experiments in digital evolution (editors' introduction to the special issue). *Artificial Life*, 10:117–122, 2004.

[4] A. Agrawal and C. M. Lively. Infection genetics: gene-for-gene versus matching-alleles models and all points in between. *Evolutionary Ecology Research*, 4:79–90, 2002.

[5] A. A. Agrawal, M. T. J. Johnson, A. P. Hastings, and J. L. Maron. A field experiment demonstrating plant life-history evolution and its eco-evolutionary feedback to seed predator populations. *The American Naturalist*, 181:S1–S11, 2013.

[6] M. Almeida-Neto, P. Guimaraes, P. R. Guimarães, R. D. Loyola, and W. Ulrich. A consistent metric for nestedness analysis in ecological systems: reconciling concept and measurement. *Oikos*, 117:1227–1239, 2008.

[7] M. Almeida-Neto and W. Ulrich. A straightforward computational approach for measuring nestedness using quantitative matrices. *Environmental Modelling & Software*, 26:173–178, 2011.

[8] S. Altizer, D. Harvell, and E. Friedle. Rapid evolutionary dynamics and disease threats to biodiversity. *Trends in Ecology & Evolution*, 18:589–596, 2003.

[9] A. I. Araujo, A. M. de Almeida, M. Z. Cardoso, and G. Corso. Abundance and nestedness in interaction networks. *Ecological Complexity*, 7:494–499, 2010.

[10] J. Bascompte. Disentangling the web of life. *Science*, 325:416–419, 2009.

[11] J. Bascompte, P. Jordano, C. Melián, and J. M. Olesen. The nested assembly of plant–animal mutualistic networks. *Proceedings of the National Academy of Sciences of the United States of America*, 100:9383–9387, 2003.

[12] U. Bastolla, M. A. Fortuna, A. Pascual-García, A. Ferrera, B. Luque, and J. Bascompte. The architecture of mutualistic networks minimizes competition and increases biodiversity. *Nature*, 458:1018–1020, 2009.

[13] C. W. Benkman, T. L. Parchman, and E. T. Mezquida. Patterns of coevolution in the adaptive radiation of crossbills. *Annals of the New York Academy of Sciences*, 1206:1–16, 2010.

[14] C. Bérénos, K. M. Wegner, and P. Schmid-Hempel. Antagonistic coevolution with parasites maintains host genetic diversity: an experimental test. *Proceedings of the Royal Society B: Biological Sciences*, 278:218–224, January 2011.

[15] A. Best, A. White, E. Kisdi, J. Antonovics, M. A. Brockhurst, and M. Boots. The Evolution of Host-Parasite Range. *The American naturalist*, 176:63–71, 2010.

[16] B. Bohannan and R. Lenski. Linking genetic change to community evolution: insights from studies of bacteria and bacteriophage. *Ecology Letters*, 3:362–377, 2000.

[17] J. Bongard. Morphological change in machines accelerates the evolution of robust behavior. *Proceedings of the National Academy of Sciences*, 108:1234–1239, 2011.

[18] S. P. Borgatti, A. Mehra, D. J. Brass, and G. Labianca. Network analysis in the social sciences. *science*, 323:892–895, 2009.

[19] J. E. Bouma and R. E. Lenski. Evolution of a bacteria/plasmid association. *Nature*, 335:351–352, 09 1988.

[20] E. F. Boyd and H. Brüssow. Common themes among bacteriophage-encoded virulence factors and diversity among the bacteriophages involved. *TRENDS in Microbiology*, 10:521–529, 2002.

[21] M. A. Brockhurst, P. B. Rainey, and A. Buckling. The effect of spatial heterogeneity and parasites on the evolution of host diversity. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 271:107–111, 2004.

[22] J. J. Bull, I. J. Molineux, and W. R. Rice. Selection of benevolence in a host-parasite system. *Evolution*, pages 875–882, 1991.

[23] E. Canard, N. Mouquet, D. Mouillot, M. Stanko, D. Miklisova, and D. Gravel. Empirical evaluation of neutral interactions in host-parasite networks. *The American Naturalist*, 183:468–479, 2014.

[24] S. P. Carroll, A. P. Hendry, D. N. Reznick, and C. W. Fox. Evolution on ecological time-scales. *Functional Ecology*, 21:387–393, 2007.

[25] S. S. Chow, C. O. Wilke, C. Ofria, R. E. Lenski, and C. Adami. Adaptive radiation from resource competition in digital organisms. *Science*, 305:84–86, 2004.

[26] J. E. Cohen. *Food webs and niche space*. Princeton Univ Pr, 1978.

[27] B. D. Connelly, L. Zaman, and P. K. McKinley. The seeds platform for evolutionary and ecological simulations. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion*, pages 133–140. ACM, 2012.

[28] B. D. Connelly, L. Zaman, P. K. McKinley, and C. Ofria. Modeling the evolutionary dynamics of plasmids in spatial populations. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 227–234. ACM, 2011.

[29] B. D. Connelly, L. Zaman, C. Ofria, and P. K. McKinley. Social structure and the maintenance of biodiversity. In *Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, pages 461–468.

[30] T. F. Cooper and C. Ofria. Evolution of stable ecosystems in populations of digital organisms. *Artificial life eight*, pages 227–232, 2003.

[31] C. Darwin. *On the origin of species by means of natural selection*. John Murray, London, 1859.

[32] W. Dáttilo, F. M. D. Marquitti, P. R. Guimarães Jr, and T. J. Izzo. The structure of ant-plant ecological networks: is abundance enough? *Ecology*, 2013.

[33] R. Dawkins. Human chauvinism. *Evolution*, 51:1015–1020, 1997.

[34] R. Dawkins and J. R. Krebs. Arms races between and within species. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 205:489–511, 1979.

[35] D. L. DeAngelis and L. J. Gross, editors. *Individual-based models and approaches in ecology: populations, communities and ecosystems*. Chapman & Hall, 1992.

[36] K. Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

[37] E. Decaestecker, S. Gaba, J. A. Raeymaekers, R. Stoks, L. Van Kerckhoven, D. Ebert, and L. De Meester. Host–parasite 'red queen'dynamics archived in pond sediment. *Nature*, 450:870–873, 2007.

[38] A. K. Dewdney. In the game called core war hostile programs engage in a battle of bits. *Scientific American*, 250:15–19, 1984.

[39] T. Dobzhansky. Biology, molecular and organismic. *American Zoologist*, 4:443–452, 1964.

[40] M. Doebeli and U. Dieckmann. Evolutionary branching and sympatric speciation caused by different types of ecological interactions. *American Naturalist*, 156:77–101, 2000.

[41] M. A. Duffy and L. Sivars-Becker. Rapid evolution and ecological host–parasite dynamics. *Ecology Letters*, 10:44–53, 2007.

[42] S. Fellous and L. Salvaudon. How can your parasites become your allies? *Trends in parasitology*, 25:62–66, 2009.

[43] F. Fenner. The florey lecture, 1983: biological control, as exemplified by smallpox eradication and myxomatosis. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 218:259–285, 1983.

[44] A. Fenton, J. Antonovics, and M. A. Brockhurst. Inverse-gene-for-gene infection genetics and coevolutionary dynamics. *The American Naturalist*, 174:E230–E242, 2009.

[45] D. Floreano, P. Dürr, and C. Mattiussi. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1:47–62, 2008.

[46] C. O. Flores, J. R. Meyer, S. Valverde, L. Farr, and J. S. Weitz. Statistical structure of host–phage interactions. *Proceedings of the National Academy of Sciences*, 108:E288–97, 2011.

[47] D. B. Fogel. An introduction to simulated evolutionary optimization. *Neural Networks, IEEE Transactions on*, 5:3–14, 1994.

[48] D. B. Fogel. What is evolutionary computation? *Spectrum, IEEE*, 37:26–28, 2000.

[49] J. A. Foley, S. Levis, M. H. Costa, W. Cramer, and D. Pollard. Incorporating dynamic vegetation cover within global climate models. *Ecological Applications*, 10:1620–1632, 2000.

[50] M. A. Fortuna, D. B. Stouffer, J. M. Olesen, P. Jordano, D. Mouillot, B. R. Krasnov, R. Poulin, and J. Bascompte. Nestedness versus modularity in ecological networks: two sides of the same coin? *Journal of Animal Ecology*, 79:811–817, 2010.

[51] M. A. Fortuna, L. Zaman, A. P. Wagner, and C. Ofria. Evolving digital ecological networks. *PLoS Computational Biology*, 9:e1002928, 2013.

[52] J. A. Foster. Evolutionary computation. *Nature Reviews Genetics*, 2:428–436, 2001.

[53] L. R. Fox. Diffuse coevolution within complex communities. *Ecology: a publication of the Ecological Society of America (USA)*, 1988.

[54] S. A. Frank. Evolution of host-parasite diversity. *Evolution*, pages 1721–1732, 1993.

[55] A. S. Fraser. Simulation of genetic systems by automatic digital computers VI. Epistasis. *Australian Journal of Biological Sciences*, 13:150–162, 1960.

[56] G. F. Fussmann, M. Loreau, and P. A. Abrams. Eco-evolutionary dynamics of communities and ecosystems. *Functional Ecology*, 21:465–477, 2007.

[57] R. T. Gilman, S. L. Nuismer, and D.-C. Jhwueng. Coevolution in multidimensional trait space favours escape from parasites and pathogens. *Nature*, 483:328–330, 2012.

[58] S. Goings and C. Ofria. Ecological approaches to diversity maintenance in evolutionary algorithms. In *Artificial Life, 2009. ALife'09. IEEE Symposium on*, pages 124–130. IEEE, 2009.

[59] J. M. Gómez, M. Verdú, and F. Perfectti. Ecological interactions are evolutionarily conserved across the entire tree of life. *Nature*, 465:918–921, 2010.

[60] S. J. Gould. *Wonderful life: The Burgess Shale and the nature of history*. WW Norton & Company, 1990.

[61] S. J. Gould. *Wonderful life: The Burgess Shale and the nature of history*. W W Norton & Co, 1990.

[62] S. J. Gould. *Full house*. Harmony Books, 1996.

[63] P. R. Guimaraes Jr, P. Jordano, and J. N. Thompson. Evolution and coevolution in mutualistic networks. *Ecology letters*, 14:877–885, 2011.

[64] N. G. Hairston Jr, S. P. Ellner, M. A. Geber, T. Yoshida, and J. A. Fox. Rapid evolution and the convergence of ecological and evolutionary time. *Ecology Letters*, 8:1114–1127, 2005.

[65] J. R. Haloin and S. Y. Strauss. Interplay between ecological communities and evolution. *Annals of the New York Academy of Sciences*, 1133:87–125, 2008.

[66] W. D. Hamilton. Pathogens as causes of genetic diversity in their host populations. *Population biology of infectious diseases*, pages 269–296, 1982.

[67] W. Harcombe. Novel cooperation experimentally evolved between species. *Evolution*, 64:2166–2172, 2010.

[68] R. M. Hazen, P. L. Griffin, J. M. Carothers, and J. W. Szostak. Functional information and the emergence of biocomplexity. *Proceedings of the National Academy of Sciences*, 104:8574, 2007.

[69] K. D. Heath and P. Tiffin. Context dependence in the coevolution of plant and rhizobial mutualists. *Proceedings of the Royal Society B: Biological Sciences*, 274:1905–1912, 2007.

[70] C. D. Herring, J. D. Glasner, and F. R. Blattner. Gene replacement without selection: regulated suppression of amber mutations in *Escherichia coli*. *Gene*, 311:153–163, 2003.

[71] S. I. Higgins and S. Scheiter. Atmospheric co2 forces abrupt vegetation shifts locally, but not globally. *Nature*, 488:209–212, 2012.

[72] W. D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena*, 42:228–234, 1990.

[73] J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan press, 1975.

[74] N. Hopkins and M. Ptashne. Genetics of virulence. *Cold Spring Harbor Monograph Archive*, 2:571–574, 1971.

[75] P. J. Hudson, A. P. Dobson, and K. D. Lafferty. Is a healthy ecosystem one that is rich in parasites? *Trends in Ecology & Evolution*, 21:381–385, 2006.

[76] J. Huxley. *Evolution in action*. Harper New York, 1953.

[77] B. Inouye and J. R. Stinchcombe. Relationships between ecological interaction modifications and diffuse coevolution: similarities, differences, and causal links. *Oikos*, 95:353–360, 2001.

[78] D. H. Janzen. When is it coevolution. *Evolution*, 34:611–612, 1980.

[79] M. T. J. Johnson and A. A. Agrawal. The ecological play of predator–prey dynamics in an evolutionary theatre. *Trends in Ecology & Evolution*, 18:549–551, 2003.

[80] M. T. J. Johnson and J. R. Stinchcombe. An emerging synthesis between community ecology and evolutionary biology. *Trends in ecology & evolution*, 22:250–257, 2007.

[81] M. T. J. Johnson and J. R. Stinchcombe. An emerging synthesis between community ecology and evolutionary biology. *Trends in ecology & evolution*, 22:250–257, 2007.

[82] P. Jordano, J. Bascompte, and J. M. Olesen. Invariant properties in coevolutionary networks of plant–animal interactions. *Ecology letters*, 6:69–81, 2003.

[83] D. H. Juers, B. W. Matthews, and R. E. Huber. Lacz β-galactosidase: Structure and function of an enzyme of historical and molecular biological importance. *Protein Science*, 21:1792–1807, 2012.

[84] B. Kerr, C. Neuhauser, B. J. M. Bohannan, and A. M. Dean. Local migration promotes competitive restraint in a host–pathogen'tragedy of the commons'. *Nature*, 442:75–78, 2006.

[85] B. Kerr, J. D. West, and B. J. M. Bohannan. *Bacteriophages: Models for exploring basic principles of ecology*, chapter 2, pages 31–63. University Press, Cambridge, U.K., 2008.

[86] J. Koza and R. Poli. Genetic programming. *Search Methodologies*, pages 127–164, 2005.

[87] A. R. Kraaijeveld. Cost of resistance to parasites in digital organisms. *Journal of Evolutionary Biology*, 20:845–853, 2007.

[88] A. Krishna, P. R. Guimaraes Jr, P. Jordano, and J. Bascompte. A neutral-niche theory of nestedness in mutualistic networks. *Oikos*, 117:1609–1618, 2008.

[89] G. Kritsky. Darwin's madagascan hawk moth prediction. *American Entomologist*, 37:206–210, 1991.

[90] E. Kutter and A. Sulakvelidze. *Bacteriophages: biology and applications*. CRC, 2005.

[91] S. J. Labrie, J. E. Samson, and S. Moineau. Bacteriophage resistance mechanisms. *Nature Reviews Microbiology*, 8:317–327, 2010.

[92] D. Lawrence, F. Fiegna, V. Behrends, J. G. Bundy, A. B. Phillimore, T. Bell, and T. G. Barraclough. Species interactions alter evolutionary responses to a novel environment. *PLoS biology*, 10:e1001330, 2012.

[93] J. Lehman and K. O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. *Artificial Life*, 11:329—336, 2008.

[94] R. E. Lenski. Twice as natural. *Nature*, 414:255–255, 2001.

[95] R. E. Lenski and R. M. May. The evolution of virulence in parasites and pathogens: reconciliation between two competing hypotheses. *Journal of Theoretical Biology*, 169:253–265, 1994.

[96] R. E. Lenski, C. Ofria, T. C. Collier, and C. Adami. Genome complexity, robustness and genetic interactions in digital organisms. *Nature*, 400:661–664, 1999.

[97] R. E. Lenski, C. Ofria, T. C. Collier, and C. Adami. Genome complexity, robustness and genetic interactions in digital organisms. *Nature*, 400:661–664, 1999.

[98] R. E. Lenski, C. Ofria, R. T. Pennock, and C. Adami. The evolutionary origin of complex. *Nature*, pages 139–144, 2003.

[99] R. E. Lenski and M. Travisano. Dynamics of adaptation and diversification: a 10,000-generation experiment with bacterial populations. *Proceedings of the National Academy of Sciences*, 91:6808–6814, 1994.

[100] B. R. Levin. The accessory genetic elements of bacteria: existence conditions and (co) evolution. *Current opinion in genetics & development*, 3:849–854, 1993.

[101] S. A. Levin. The problem of pattern and scale in ecology: the robert h. macarthur award lecture. *Ecology*, 73:1943–1967, 1992.

[102] S. A. Levin, B. Grenfell, A. Hastings, and A. S. Perelson. Mathematical and computational challenges in population biology and ecosystems science. *Science*, 275:334–343, 1997.

[103] H. Lipson. Evolutionary robotics and open-ended design automation. *Biomimetics*, 17:129–155, 2005.

[104] M. Lynch. The frailty of adaptive hypotheses for the origins of organismal complexity. *Proceedings of the National Academy of Sciences*, 104:S8597–S8604, 2007.

[105] H. M., J. A., and B.-B. C. Lamb mutations in e. coli k12: growth of lambda host range mutants and effect of nonsense suppressors. *Molecular Genetics and Genomics*, 145:207–213.

[106] M. F. Marston, F. J. Pierciey Jr, A. Shepard, G. Gearin, J. Qi, C. Yandava, S. C. Schuster, M. R. Henn, and J. Martiny. Rapid diversification of coevolving marine synechococcus and a virus. *Proceedings of the National Academy of Sciences*, pages 4544–4549.

[107] R. M. May et al. How many species are there on earth?. *Science(Washington)*, 241:1441–1449, 1988.

[108] K. S. McCann. The diversity–stability debate. *Nature*, 405:228–233, 2000.

[109] P. McKinley, B. H. C. Cheng, C. Ofria, D. Knoester, B. Beckmann, and H. Goldsby. Harnessing digital evolution. *Computer*, 41:54–63, 2008.

[110] J. R. Meyer, D. T. Dobias, J. S. Weitz, J. E. Barrick, R. T. Quick, and R. E. Lenski. Repeatability and contingency in the evolution of a key innovation in phage lambda. *Science*, 335:428–432, 2012.

[111] J. R. Meyer, S. P. Ellner, N. G. Hairston Jr, L. E. Jones, and T. Yoshida. Prey evolution on the time scale of predator–prey dynamics revealed by allele-specific quantitative PCR. *Proceedings of the National Academy of Sciences*, 103:10690–10695, 2006.

[112] J. R. Meyer and R. Kassen. The effects of competition and predation on diversification in a model adaptive radiation. *Nature*, 446:432–435, 2007.

[113] D. Misevic, C. Ofria, and R. E. Lenski. Sexual reproduction reshapes the genetic architecture of digital organisms. *Proceedings of the Royal Society B: Biological Sciences*, 273:457–464, 2006.

[114] G. G. Mittelbach. *Community ecology*. Sinauer Associates, 2012.

[115] S. R. Modi, H. H. Lee, C. S. Spina, and J. J. Collins. Antibiotic treatment expands the resistance reservoir and ecological network of the phage metagenome. *Nature*, 499:219–222, 2013.

[116] A. Mougi and M. Kondoh. Diversity of interaction types and ecological community stability. *Science*, 337:349–351, 2012.

[117] E. R. Moxon, P. B. Rainey, M. A. Nowak, and R. E. Lenski. Adaptive evolution of highly mutable loci in pathogenic bacteria. *Current Biology*, 4:24–33, 1994.

[118] C. Neuhauser and J. E. Fargione. A mutualism–parasitism continuum model and its application to plant–mycorrhizae interactions. *Ecological modelling*, 177:337–352, 2004.

[119] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology*. MIT Press, 2000.

[120] C. Ofria, D. M. Bryson, and C. O. Wilke. *Artificial Life Models in Software*, chapter 1, pages 3–32. Springer, 2nd edition, July 2009.

[121] C. Ofria and C. O. Wilke. Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10:191–229, 2004.

[122] J. M. Olesen, J. Bascompte, Y. L. Dupont, H. Elberling, C. Rasmussen, and P. Jordano. Missing and forbidden links in mutualistic networks. *Proceedings of the Royal Society B: Biological Sciences*, 278:725–732, 2011.

[123] J. M. Olesen, J. Bascompte, H. Elberling, and P. Jordano. Temporal dynamics in a pollination network. *Ecology*, 89:1573–1582, 2008.

[124] B. O'Neill. Digital evolution. *PLoS Biology*, 1:e18, 2003.

[125] T. M. Palmer, M. L. Stanton, T. P. Young, J. R. Goheen, R. M. Pringle, and R. Karban. Breakdown of an ant-plant mutualism follows the loss of large herbivores from an african savanna. *Science*, 319:192–195, 2008.

[126] R. T. Pennock. Models, simulations, instantiations, and evidence: the case of digital evolution. *Journal of Experimental & Theoretical Artificial Intelligence*, 19:29–42, 2007.

[127] R. Poulin. Are there general laws in parasite ecology? *Parasitology*, 134:763–776, 2007.

[128] R. Poulin. Network analysis shining light on parasite ecology and diversity. *Trends in parasitology*, 26:492–498, 2010.

[129] R. Poulin and S. Morand. The diversity of parasites. *Quarterly Review of Biology*, 75:277–293, 2000.

[130] P. W. Price. *Evolutionary biology of parasites*. Princeton University Press, 1980.

[131] S. Rasmussen, C. Knudsen, R. Feldberg, and M. Hindsholm. The coreworld: Emergence and evolution of cooperative structures in a computational chemistry. *Physica D: Nonlinear Phenomena*, 42:111–134, 1990.

[132] S. Rasmussen, C. Knudsen, R. Feldberg, and M. Hindsholm. The coreworld: Emergence and evolution of cooperative structures in a computational chemistry. *Physica D: Nonlinear Phenomena*, 42:111–134, 1990.

[133] T. Ray. An approach to the synthesis of life. In *Proceedings of Artificial Life*, volume 2, pages 371–408, 1992.

[134] T. S. Ray. An approach to the synthesis of life. *Artificial life II*, 10:371–408, 1992.

[135] D. Refardt and P. B. Rainey. Tuning a genetic switch: experimental evolution and natural variation of prophage induction. *Evolution*, 64:1086–1097, 2010.

[136] E. L. Rezende, J. E. Lavabre, P. R. Guimarães, P. Jordano, and J. Bascompte. Non-random coextinctions in phylogenetically structured mutualistic networks. *Nature*, 448:925–928, 2007.

[137] M. J. Roossinck. The good viruses: viral mutualistic symbioses. *Nature Reviews Microbiology*, 9:99–108, 2011.

[138] J. L. Sachs, U. G. Mueller, T. P. Wilcox, and J. J. Bull. The evolution of cooperation. *The Quarterly Review of Biology*, 79:135–160, 2004.

[139] J. L. Sachs and E. L. Simms. Pathways to mutualism breakdown. *Trends in ecology & evolution*, 21:585–592, 2006.

[140] J. L. Sachs, R. G. Skophammer, N. Bansal, and J. E. Stajich. Evolutionary origins and diversification of proteobacterial mutualists. *Proceedings of the Royal Society B: Biological Sciences*, 281:2013–2146, 2014.

[141] J. L. Sachs and T. P. Wilcox. A shift to parasitism in the jellyfish symbiont symbiodinium microadriaticum. *Proceedings of the Royal Society B: Biological Sciences*, 273:425–429, 2006.

[142] J. Sambrook, E. F. Fritsch, T. Maniatis, et al. *Molecular cloning*, volume 3. Cold spring harbor laboratory press New York, 3 edition, 2001.

[143] B. Sareni and L. Krahenbuhl. Fitness sharing and niching methods revisited. *Evolutionary Computation, IEEE Transactions on*, 2:97–106, 1998.

[144] T. W. Schoener. The newest synthesis: understanding the interplay of evolutionary and ecological dynamics. *Science*, 331:426, 2011.

[145] T. Shanahan. Evolutionary progress from darwin to dawkins. *Endeavour*, 23:171–174, 1999.

[146] T. Shanahan. *The evolution of Darwinism: Selection, adaptation and progress in evolutionary biology*. Cambridge University Press, 2004.

[147] J. Shao and T. S. Ray. Maintenance of species diversity by predation in the tierra system. In *12th International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, pages 533–540, 2010.

[148] Y. Shao and N. Wang. Bacteriophage adsorption rate and optimal lysis time. *Genetics*, 180:471–482, 2008.

[149] H. A. Shuman and T. J. Silhavy. The art and design of genetic screens: Escherichia coli. *Nature Reviews Genetics*, 4:419–431, 2003.

[150] G. G. Simpson. *The meaning of evolution: a study of the history of life and of its significance for man*. Yale Univ Press, 1967.

[151] L. B. Slobodkin. *Growth and regulation of animal populations*. Holt, Rinehart and Winston New York, 1961.

[152] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15:185–212, 2009.

[153] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evol. Comput.*, 10:99–127, June 2002.

[154] F. M. Stewart and B. R. Levin. The population biology of bacterial viruses: why be temperate. *Theoretical population biology*, 26:93–117, 1984.

[155] S. Y. Strauss, H. Sahli, and J. K. Conner. Toward a more trait-centered approach to diffuse (co) evolution. *New Phytologist*, 165:81–90, 2005.

[156] B. Sures and R. Siddall. Pomphorhynchus laevis: the intestinal acanthocephalan as a lead sink for its fish host, chub (leuciscus cephalus). *Experimental Parasitology*, 93:66–72, 1999.

[157] J. W. Szostak, D. P. Bartel, and P. L. Luisi. Synthesizing life. *Nature*, 409:387–390, 2001.

[158] D. R. Taylor, A. M. Jarosz, D. W. Fulbright, and R. E. Lenski. The acquisition of hypovirulence in host-pathogen systems with three trophic levels. *The American Naturalist*, 151:343–355, 1998.

[159] E. Thébault and C. Fontaine. Stability of ecological communities and the architecture of mutualistic and trophic networks. *Science*, 329:853–856, 2010.

[160] J. N. Thompson. *The coevolutionary process*. University of Chicago Press, 1994.

[161] J. N. Thompson. Rapid evolution as an ecological process. *Trends in Ecology & Evolution*, 13:329–332, 1998.

[162] J. N. Thompson. The evolution of species interactions. *Science*, 284:2116–2118, 1999.

[163] J. N. Thompson. Coevolution: the geographic mosaic of coevolutionary arms races. *Current Biology*, 15:R992–R994, 2005.

[164] J. N. Thompson. *The geographic mosaic of coevolution*. University of Chicago Press, 2005.

[165] J. N. Thompson. The coevolving web of life (american society of naturalists presidential address). *The American Naturalist*, pages 125–140, 2009.

[166] J. N. Thompson. *Relentless evolution*. University of Chicago Press, 2013.

[167] M. E. Torchin, K. D. Lafferty, A. P. Dobson, V. J. McKenzie, and A. M. Kuris. Introduced species and their missing parasites. *Nature*, 421:628–630, 2003.

[168] M. M. Turcotte, M. S. Corrin, and M. T. Johnson. Adaptive evolution in ecological communities. *PLoS biology*, 10:e1001332, 2012.

[169] W. Ulrich, M. Almeida-Neto, and N. J. Gotelli. A consumer's guide to nestedness analysis. *Oikos*, 118:3–17, 2009.

[170] D. P. Vázquez, N. Blüthgen, L. Cagnolo, and N. P. Chacoff. Uniting pattern and process in plant–animal mutualistic networks: a review. *Annals of Botany*, 103:1445–1457, 2009.

[171] G. J. Vermeij. *Evolution and escalation: an ecological history of life*. Princeton University Press, 1993.

[172] J. Vizentin-Bugoni, P. K. Maruyama, and M. Sazima. Processes entangling interactions in communities: forbidden links are more important than abundance in a hummingbird–plant network. *Proceedings of the Royal Society B: Biological Sciences*, 281:20132397, 2014.

[173] V. Volterra. *Variazioni e fluttuazioni del numero d'individui in specie animali conviventi*. Memoria / R. comitato talassografico italiano, no. 131. 1927.

[174] N. Wang, J. Deaton, and R. Young. Sizing the holin lesion with an endolysin-β-galactosidase fusion. *Journal of bacteriology*, 185:779–787, 2003.

[175] A. R. Weeks, M. Turelli, W. R. Harcombe, K. T. Reynolds, and A. A. Hoffmann. From parasite to mutualist: rapid evolution of wolbachia in natural populations of drosophila. *PLoS Biology*, 5:e114, 2007.

[176] J. S. Weitz, P. N. Benfey, and N. S. Wingreen. Evolution, interactions, and biological networks. *PLoS Biology*, 5:e11, 2007.

[177] J. S. Weitz, H. Hartman, and S. A. Levin. Coevolutionary arms races between bacteria and bacteriophage. *Proceedings of the National Academy of Sciences of the United States of America*, 102:9535–9540, 2005.

[178] C. O. Wilke and C. Adami. The biology of digital organisms. *Trends in Ecology & Evolution*, 17:528–532, 2002.

[179] C. O. Wilke, J. L. Wang, C. Ofria, R. E. Lenski, and C. Adami. Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412:331–333, 2001.

[180] H. T. Williams. Coevolving parasites improve host evolutionary search on structured fitness landscapes. In *Artificial Life*, volume 13, pages 129–136, 2012.

[181] M. E. Woolhouse, J. P. Webster, E. Domingo, B. Charlesworth, and B. R. Levin. Biological and biomedical implications of the co-evolution of pathogens and their hosts. *Nature genetics*, 32:569–577, 2002.

[182] J. B. Yoder and S. L. Nuismer. When does coevolution promote diversification? *The American Naturalist*, 176:802–817, 2010.

[183] T. Yoshida, L. E. Jones, S. P. Ellner, G. F. Fussmann, and N. G. Hairston. Rapid evolution drives ecological dynamics in a predator-prey system. *Nature*, 424:303–306, 2003.

[184] L. Zaman, S. Devangam, and C. Ofria. Rapid host-parasite coevolution drives the production and maintenance of diversity in digital organisms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 219–226. ACM, 2011.