THE EVOLUTION OF DIGITAL COMMUNITIES UNDER LIMITED RESOURCES

By

Bess Linden Walker

A DISSERTATION

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

DOCTOR OF PHILOSOPHY

Computer Science
Ecology, Evolutionary Biology and Behavior

2012

ABSTRACT

THE EVOLUTION OF DIGITAL COMMUNITIES UNDER LIMITED RESOURCES

By

Bess Linden Walker

Schluter (1996) describes adaptive radiation as "the diversification of a lineage into species that exploit a variety of different resource types and that differ in the morphological or physiological traits used to exploit those resources". My research focuses on adaptive radiation in the context of limited resources, where frequency-dependence is an important driver of selection (Futuyma & Moreno, 1988; Dieckmann & Doebeli, 1999; Friesen *et al.*, 2004). Adaptive radiation yields a community composed of distinct organism types adapted to specific niches.

I study simple communities of digital organisms, the result of adaptive radiation in environments with limited resources. I ask (and address) the questions: How does diversity, driven by resource limitation, affect the frequency with which complex traits arise? What other aspects of the evolutionary pressures in this limited resource environment might account for the increase in frequency with which complex traits arise? Can we predict community stability when it encounters another community, and is our prediction different for communities resulting from adaptive radiation versus those that are artificially assembled?

Community diversity is higher in environments with limited resources than in those with unlimited resources. The evolution of an example complex feature (in this case, Boolean EQU) is also more common in limited-resource environments, and shows a strong correlation with diversity over a range of resource inflow rates. I show that populations evolving in intermediate inflow rates explore areas of the fitness landscape in which EQU is common, and that those in unlimited resource environments do not. Another feature of the limited-resource environments is the reduced cost of trading off the execution of building block tasks for higher-complexity tasks. I find strong causal evidence that this reduced cost is a factor in the more common evolution of EQU in limited-resource environments.

When two communities meet in competition, the fraction of each community's descendants making up the final post-competition community is strongly consistent across replicates. I find that three community-level factors, ecotypic diversity, community composition, and resource use efficiency can be used to predict this fractional community success, explaining up to 35% of the variation.

In summary, I demonstrate the value of digital communities as a tractable experimental system for studying general community properties. They sit at the bridge between ecology and evolutionary biology and evolutionary computation, and offer comprehensible ways to translate ideas across these fields.

For Jay Genglebach, who unknowingly spurred half a year of productivity simply by assuming I was a productive person.

ACKNOWLEDGMENTS

This dissertation has been brought to you by Swedish Fish®, which fueled the last couple weeks of writing.

I am deeply grateful to the many people who have aided and supported me through my graduate studies. My mother Ruth E. W. Walker has been a curious ear, supportive shoulder, and helpful editor throughout this process. Thanks for years of love, tolerance, and support are owed to Kaben Nanlohy, whose unique history with Avida has made him an excellent sounding board for many pieces of this research.

I would not have gotten very far without the input of my academic committee. My advisor, Dr. Charles Ofria, has been the best advisor anyone could wish for: intellectually engaging, warm-hearted, and patient. His contribution to this work cannot be overstated. I would also like to thank the rest of my committee for their contributions to my research: Dr. Richard Lenski, Dr. Philip McKinley, and Dr. William Punch. Their diverse perspectives have added depth to this interdisciplinary effort.

My time at Michigan State has been spent in the company of my fellow Devolab students, in whom I have been very lucky. Every one of them has been friendly and supportive, and they have formed the core of my friendships here. I would especially like to thank Sherri Goings, who is one of my collaborators for the research presented in Chapters 3 and 4. Her skill for developing incisive tests is one I envy and have benefitted from.

I owe a great deal to Dr. Tim Cooper, whose preliminary work on the outcome of community competition forms the solid foundation of my research in Chapter 5. His summary report of that work has been invaluable, both in guiding my research toward promising paths and in steering it away from already-explored blank alleys.

This material is based in part upon work supported by a National Science Foundation Graduate Research Fellowship, by Michigan State University under a University Distinguished Fellowship, and by the National Science Foundation under Grant No. CCF-0643952 and Cooperative Agreement No. DBI-0939454.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

In this introductory chapter, I provide a brief overview of my study system and the two central topics of this thesis, communities and complex features: the formal details of these concepts are more fully addressed in each chapter as they arise. In Chapter 2, I provide a detailed description of my study system, the Avida digital evolution platform, and define some experimental environments and terminology used throughout this thesis. In Chapter 3, I examine the relationships between resource limitation, community diversity, and the evolvability of complex features. In Chapter 4, I explore other negative frequency-dependent effects of resource limitation on the evolvability of complex features. I show that the reduced cost of trading the ability to perform a common task for a rare one increases the frequency of the evolution of a complex feature. In Chapter 5, I investigate community-level predictors of success in competition with another community. Finally, in Chapter 6, I describe extensions that will carry this work forward.

1.1 Study System

I use the digital evolution software Avida (Ofria *et al.*, 2009), which allows precise manipulation of resource supply and a complete record of the course of evolution. The Avida system consists of a collection of digital organisms, each with a simple circular genome composed of instructions from an assembly-like Turing-complete instruction set.

By executing the instructions in its genome, each organism is capable of self-reproduction; during this process, copy mutations may be introduced into the offspring's genome. Because the genetic instructions are drawn from a Turing-complete language, the organisms are also theoretically capable of any Turing-computable task[1]. The organisms have access to integers that they can manipulate; the researcher can choose to reward certain manipulations, known as *tasks*, with additional CPU cycles. These additional CPU cycles allow the organism to execute its genome more quickly and thus increase fitness. By tying the reward for each task to a separate depletable resource, we can achieve adaptive radiation driven by negative frequency-dependent selection.

Chapter 2 contains a more detailed explanation of this study system.

1.2 Communities

As in nature, negative frequency-dependent selection in Avida drives the evolution of ecologically stable communities that consist of multiple genotypes exploiting different niches (Cooper & Ofria, 2002). These communities are of interest in this thesis for three reasons. First, by using communities, results in Avida are more comparable to many real-world phenomena; the single-niche populations we often study are more tractable than communities, but correspond closely only to monocultures and laboratory experiments. Second, the evolution of a community under negative frequency-dependent selection would seem to be inherently different from the evolution of a single-niche population under no such diversifying selection; I study this difference in the context of the evolution of complex features (see Section 1.3 below). Third, the interaction between community members is of interest; I

[1] That is, they are theoretically capable of any computation that any other traditional computer can perform.

examine this specifically in the context of competition between communities by using both evolved and artificially assembled communities.

In Chapter 5, I demonstrate that the outcome of such competitions is quite consistent. What, then, makes one community more likely to out-compete another community when the two come into contact? In nature, such community-community clashes are rare, though likely to be important; when we study a community's success in competition, it is usually in reference to one or perhaps a few invasive species. Using a digital evolution system, we can expand questions about resistance to invasion (many vs. one) to questions about success in competition with another fully-evolved community (many vs. many). For example, diverse communities tend to be invasion-resistant (Case, 1990; Kennedy *et al.*, 2002; Tilman, 2004; Zavaleta & Hulvey, 2004; Maron & Marler, 2007); is diversity also a predictor of how well a community will survive when two communities meet? Or is success in community competition driven by some other factor, such as specialization or resource utilization?

Chapters 3 and 4 examine the evolution of complex features in communities; Chapter 5 investigates community success in competition for both evolved and artificially assembled communities.

1.3 Complex Features

Digital organisms are little more than computer programs, and therefore have no complex organs in the classical sense (e.g., the lensed eye). Avidian organisms do, however, have concrete behaviors. Among these is the ability to perform certain mathematical tasks. We call the more difficult of these tasks "complex" in the same way that certain organs such as the eye are complex. In this the case of these digital organisms, doing the mathematical tasks requires coordination of several subtasks and many locations in the genome, and removing any piece of the whole computational structure means the organism is now completely unable to do the task. I use the broad term "complex feature" to encompass both complex organs and complex mathematical tasks.

There is strong evidence that evolution by natural selection can and does produce complex features in the natural world; for an excellent review, see Gregory (2008). Certainly it does so in Avida (Lenski *et al.*, 2003). My work investigates a different question: given that complex features can and do evolve, are there simple ecological selective pressures that cause them to evolve more often, or more quickly? Conversely, are there pressures that decrease the likelihood of their evolution? What are these pressures? How and why do they affect the evolution of complex features?

I have focused on a set of specific questions about the evolution of complex features in multiple-niche communities. Why might the evolution of complexity be different in a community than in a single-niche population? First, the negative frequency-dependent nature of a limited-resource environment encourages adaptive radiation and maintains diversity (Tilman, 1982; Schluter, 1996; Chesson, 2000; Hall & Colegrave, 2007; Abrams *et al.*, 2008), resulting in broader exploration of the fitness space (Friedrich *et al.*, 2009). Even naïvely, we may expect that exploring more possibilities will result in a complex feature appearing more frequently. Second, mutations causing the loss of access to depleted resources have a reduced impact, thus facilitating the crossing of fitness valleys (Weissman *et al.*, 2009). Third, the evolution of any feature often requires trade-offs (Roff and Fairbairn 2007, Lenski et al. 2003). When an adaptive trait corresponds to an untapped resource, its high availability mitigates the loss of traits tied to depleted resources, making the trade-off worth-while. I explore these hypotheses in Chapters 3 and 4.

1.4 Links to Evolutionary Computation

The field of evolutionary computation focuses on evolving solutions to complex problems. Although the core idea comes from evolutionary biology, until recently the computational field has tended to be limited in its adoption of new biological ideas, and vice versa. Preserving diversity on the path to complex solutions is a major research area in evolutionary computation (Friedrich *et al.*, 2009). The research I describe below brings these results, designed for encouraging the evolution of solutions to complex problems, back into the context of evolutionary theory, and reveals their relevance to questions in evolutionary biology about the evolution of complex features.

Chapter 2

The Avida Digital Evolution Platform

In the Avida digital evolution platform (Ofria *et al.*, 2009), self-replicating computer programs must compete for limited space in a virtual world. Avida is a true instantiation of open-ended evolution, often producing new and unexpected survival strategies. Digital organisms in Avida are subject to errors (mutations) as they copy their genetic programs, as well as selective pressures for survival and rapid replication. As in nature, almost all mutations are detrimental or neutral, but occasional beneficial mutations do occur (Lenski *et al.*, 1999). These rare beneficial mutations typically allow digital organisms to either metabolize additional resources, boosting the energy available for offspring production, or else improve efficiency, reducing energy requirements.

Avida has been under development for almost two decades and has been used to study, among a host of other topics, complexity (Lenski *et al.*, 1999; Adami *et al.*, 2000; Lenski *et al.*, 2003; Ofria *et al.*, 2008), robustness (Lenski *et al.*, 1999, 2006; Elena *et al.*, 2007), fitness landscapes (Wilke *et al.*, 2001), adaptive radiation (Cooper & Ofria, 2002; Chow *et al.*, 2004), and sexual reproduction (Misevic *et al.*, 2006). The software is open-source and available without cost at http://avida.devosoft.org.

Some sections of this chapter are adapted from Walker & Ofria (2012).

2.1 Avidians and the Avida World

We refer to digital organisms in the Avida system as "Avidians". An Avidian consists of two basic components: a genetic program and the virtual hardware that executes that program. The genetic program is a series of computer instructions drawn from an alphabet of simple assembly-like Turing-complete instruction set (see Appendix A). Through mutations, these instructions can be substituted with other instructions, deleted, or be inserted into the genetic program; they form the genome of the organism. The simulated hardware, or virtual CPU, supplies the mechanisms to execute the genomic instructions. Four heads (the instruction pointer, flow head, read head, and write head) allow looping, conditional logic, and self-replication. Three registers and two depth-10 stacks provide storage for any data the organism is manipulating; their use is controlled by the genetic instructions.

The world of Avida consists of either a toroidal[1] or bounded grid of *cells*, set to 60 × 60 in this work; each cell is either empty or holds one Avidian. In the work presented in this thesis, organisms remain in the same grid cell for their entire lives. Each cell also contains three random integer inputs that the resident Avidian may use as input for the calculations it performs; these values are randomly chosen per cell, and are independent values from one cell to another.

Time in Avida is measured in *updates*; each update corresponds to an average number of 30 instruction executions per organism in the population. Organisms running quickly will execute more than 30 instructions per update, while slow organisms will execute fewer.

Populations in Avida may begin with any ancestor organisms chosen by the researcher. In this thesis, unless otherwise stated, an initially empty world is seeded with 3600 clones of

[1] i.e., when an organism moves off of one side of the grid, it appears on the other side.

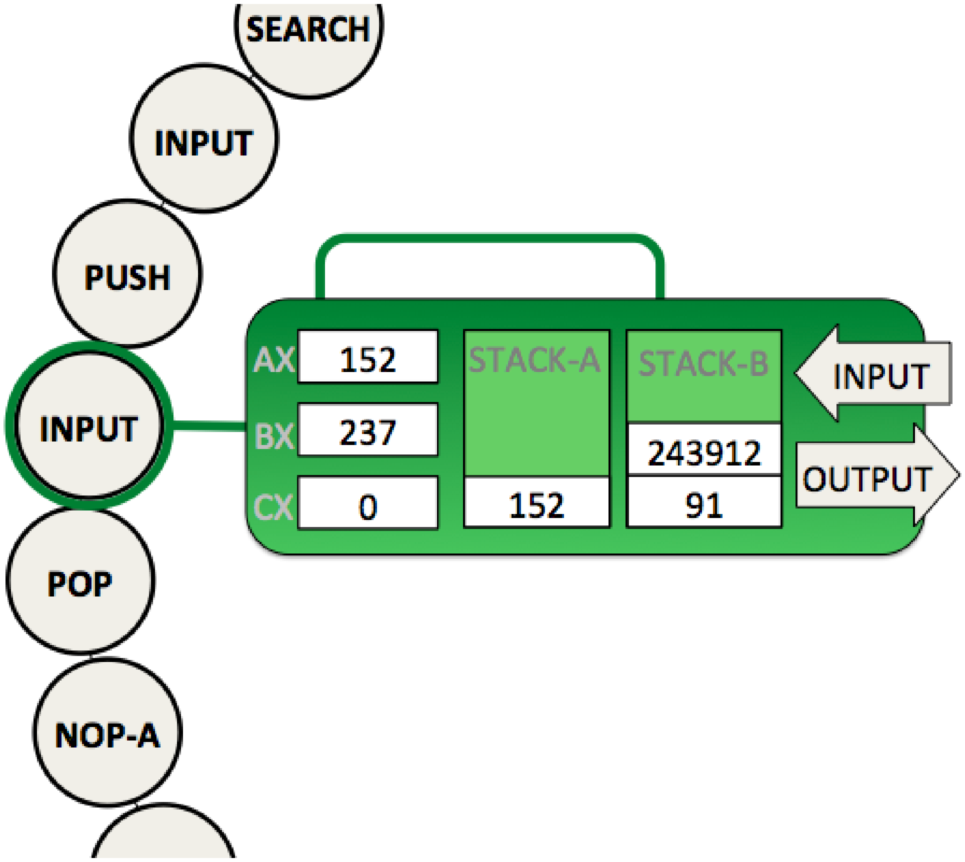

Figure 2.1: The Avidian genetic program and virtual hardware. Circles on the left represent instructions in a portion of the genome. The green box represents the virtual hardware that executes the genetic program, with three registers, two stacks, and input/output capabilities. The instruction pointer is currently at the `input` instruction; the flow, read, and write heads are not shown. (From Beacon 101: Experimental Evolution with Digital Organisms. Used with permission.) For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

an asexual ancestor organism capable only of self-replication.

2.2 Self-Replication in Avida

By executing its genome, each (viable) Avidian is capable of self-reproduction; during this process, copy mutations may occur in the offspring's genome. Both insertion and deletion mutations may also occur in the offspring's genome when it is born.

Organisms reproduce into any of the nine cells surrounding and including the Avidian itself, preferring empty cells if any are available. Because the Avidian world grid can accommodate only a fixed number of organisms (e.g., 3600), and newly born organisms overwrite current organisms, the basic evolutionary pressure in Avida is the competition for space, or equivalently, the need to reproduce before being overwritten. The predominant selective pressure is therefore the pressure to reproduce more quickly, either by reducing the number of instruction steps necessary for reproduction or by completing tasks which increase the rate (relative to other organisms) at which the organism executes instructions.

In the experiments described in this thesis, the world grid begins filled with Avidians. The only way for an Avidian to be removed from the grid is for it to be overwritten by another Avidian or to die. Organisms in Avida die if they live for “too many” (in this thesis, 30 × their genome length) instruction cycles without replicating.

2.3 Boolean Logic Tasks

As noted above (Section 2.1), each organism has access to three 32-bit integers that it can manipulate; the researcher can choose to reward certain manipulations with additional CPU cycles. These additional CPU cycles allow the organism to execute its genome more quickly and thus increase fitness. In this thesis, Avidians are rewarded for performing Boolean bitwise logic functions (see Table 2.1).

When an Avidian outputs a 32-bit integer, the system checks to see if that integer is the solution to any of nine Boolean bitwise logic tasks, using any of the integers available in that cell. If so, the organism is rewarded by an increase in metabolic rate[2], allowing it to use

[2] Metabolic rate was called “merit” in previous Avida works.

more CPU cycles per update. In essence, its metabolism increases with respect to its peers.

In my work, the magnitude of these rewards is based on the difficulty of the task. There are many ways to evaluate difficulty; the convention in Avida is to rank tasks by the amount of code required to implement them. Since the only logical operator in the Avidian instruction set is `nand`, we approximate the implementation difficulty of each logical task by counting the minimal number of logical NAND operations necessary to compute it (Table 2.1).

Task name	Boolean operation	Task value (V)
NOT	$\neg A; \neg B$	1
NAND	$\neg(A \wedge B)$	1
AND	$A \wedge B$	2
ORN	$(A \vee \neg B); (\neg A \vee B)$	2
OR	$A \wedge B$	3
ANDN	$(A \wedge \neg B); (\neg A \wedge B)$	3
NOR	$\neg A \wedge \neg B$	4
XOR	$(A \wedge \neg B) \vee (\neg A \wedge B)$	4
EQU	$(A \wedge B) \vee (\neg A \wedge \neg B)$	5

Table 2.1: NAND-count-based task reward values. The symbol "¬" denotes negation, while semicolons separate symmetrical functions. See Appendix B for a fuller treatment.

Rewards may be applied in an "additive" or a "power" structure. To increase the differences in rewards, we use 2^V rather than V to modify metabolic rate. In the additive reward structure, performance of a task with value V is rewarded by *adding* 2^V to the organism's metabolic rate. In the power reward structure, the organism's metabolic rate is *multiplied* by 2^V. Details of the particular reward structures used in this thesis are given in the chapters in which they are discussed.

2.4 Limited Resources

The Avida default is to always give the same reward for the same task. However, it is also capable of linking depletable resources to tasks; resource amounts can then modulate task rewards. These resources are implemented in a chemostat-like manner: a constant amount of each resource flows into the world, is depleted as it is used by organisms, and flows out at a constant rate. Each task has a separate linked resource, and all are non-spatial; all organisms access the same resource pools.

I refer to any such environment as a *limited-resource environment*, although, of course, the degree to which resources are limited depends on inflow rate (R_{INFLOW}), outflow rate ($R_{OUTFLOW}$), and whether Avidians are performing tasks to deplete the resource pools. The default Avida inflow and outflow rates for each resource pool are 100.0 units of resource per update and 1% of the available resource per update, respectively; inflow and outflow happen smoothly throughout the course of the update.

A task with value V (see Table 2.1) using resource amount A receives a reward of $2^{V \times A}$. Avidians use 0.25% of the relevant resource pool (with a cap of one unit of resource); more resource in the pool therefore means a greater reward for performing the task. In effect, one can think of an environment without limited resources as one that has infinite inflow, allowing the organisms to always use 1.0 units of resource when performing tasks: I refer to this as an *unlimited-resource environment*.

This system of limited resources creates negative frequency-dependent selection based on the amount of each resource available, and allows the evolution of ecologically stable communities consisting of multiple coexisting ecotypes (Cooper & Ofria, 2002).

2.5 Phenotypes and Ecotypes in Avida

A *phenotype* in Avida, as in natural biology, is a collection of traits describing how an organism's genotype is expressed during the organism's lifetime. In most cases, we simplify this measurement to how many times the organism can perform each of the Boolean logic tasks over the course of its lifetime. This set of values gives information about both the organism's ability to perform complex functions and the drain it imposes on the world's resource levels.

When two or more genotypes have similar phenotypes, we refer to them as being in the same *ecotype*. In environments where each distinct task is rewarded only once (Chapters 3 and 4), we define an ecotype as the group of genotypes which share the same phenotype. In Chapter 5, where distinct tasks can be rewarded more than once, ecotypes are determined by the hcluster package (Eads, 2008) by agglomerative clustering of log-transformed task vectors. I use the centroid linkage method with Euclidean distance. The hierarchical clustering is collapsed into flat clusters with a threshold of 1.0. These settings come from unpublished work by Christopher Strelioff, another Avida researcher. The results with these settings appear reasonable; that is, in test communities, genotypes with very similar phenotypes are clustered together, and genotypes with obviously different phenotypes fall into different clusters.

At several points in this thesis I also refer to the *binary task signature* of an organism. This categorization is simply whether or not the organism can perform each task, a true/false variant of the phenotype – in fact, exactly equivalent to the phenotype in environments which reward each distinct task only once. In my experiments with environments which reward each distinct task more than once (see Chapter 5), it is possible but not common for two or more ecotypes to have the same binary task signature if they differ strongly in the number of times they do at least one task.

2.6 The Ecological Period

At the end of any period of time during which mutations can occur (an *evolutionary period*), the Avida population consists of many genotypes. Some are incapable of reproducing themselves ("inviable"); others are newly born mutants that have a much lower fitness than their competitors, but haven't had time to be overwritten yet. Still others are near-duplicates phenotypically, coexisting simply because they are both more fit than the background. A community with so many genotypes to consider is difficult to analyze; while some subset of these genotypes may be all that is needed to form a community that is ecologically stable, it is difficult to pick that subset out.

Therefore, for many experiments I include an *ecological period* when the evolutionary period ends. Typically evolutionary and ecological periods each last 100,000 updates, for a total of 200,000 updates. In this period, all mutations (substitution, insertion, and deletion) are turned off. Competition in this period is purely ecological, allowing the community to settle out into a core set of ecotypes; all of the inviable and low-fitness genotypes are weeded out early, and fit genotypes compete in an environment where small fitness differences dramatically reduce the probability for phenotypically-identical genotypes to coexist.

In some cases genotypes may be so phenotypically similar that an ecological period of 100,000 updates is not sufficient to drive one to extinction, which is why we characterize the phenotypes of the final population either in terms of clustered ecotypes or genotypes grouped by binary task signature. However, the smaller set of genotypes remaining after an ecological period makes this process far more tractable and the results intuitively verifiable.

Chapter 3

Evolutionary Potential is Maximized at Intermediate Diversity

When Darwin first posed his theory of evolution by natural selection, critics denounced its ability to produce traits of significant complexity (Mivart, 1871). Darwin responded by noting that traits of lesser complexity can be used as building blocks and that most complex problems can be solved in many ways, only one of which needs to be discovered by evolution (Darwin, 1872). Evidence for these ideas has accumulated in the historic record, where the evolutionary origins of a range of complex traits are now well understood (Sage, 2004; Gregory, 2008; Flowers *et al.*, 2010), as well as in theoretical proofs (Melendez-Hevia *et al.*, 1996; Nilsson & Pelger, 1994) and in computational experiments (Lenski *et al.*, 2003; Graham *et al.*, 2007). However, the dynamics that drove the evolution of these complex traits are not always clear.

Lenski *et al.* (2003) have investigated the evolutionary origin of complex features in Avida, using Boolean EQU as the specific complex task under study. This is the most complex of the one- and two-input Boolean operations to calculate, requiring at least five logical NAND operations. An Avidian organism requires at least 19 coordinated instructions to perform EQU, including the execution of at least five `nand` instructions. The ancestor starts out with none of these instructions in its genome; Lenski *et al.* found that in the 23 of 50 populations that evolved EQU in their experiments, EQU evolved in anywhere from 51 to 721 mutational steps.

In practice, the evolution of EQU is dependent on rewarding building blocks: the one- and two-input Boolean tasks of lower complexity. When Lenski *et al.* evolved populations in environments where EQU was the only rewarded task, none of the populations evolved EQU. However, they also found that the evolution of EQU does not depend on any particular building block or pair of building blocks. In fact, EQU can evolve in many different ways; all 23 of Lenski *et al.*'s EQU-evolving populations evolved building blocks in different orders and organized them differently in their genomes.

Computational experiments like Lenski *et al.* (2003) have focused on single-niche populations with minimal ecological forces at play. The theory can be logically extended to multiple-niche communities that more closely mimic complex natural environments. In my initial investigations, I designed a limited-resource environment with $R_{INFLOW} = 100$ in parallel to Lenski *et al.*'s unlimited-resource environment (see Section 3.2, below, for details). I tested whether limited resources affect the rate of complex trait evolution. In the unlimited-resource treatment, EQU evolved in only 67 of 200 populations by the end of 100,000 updates[1], whereas 167 of 200 evolved EQU in this same period in the limited-resource treatment, a highly significant difference ($p < 10^{-23}$, Fisher's exact test, two-tailed).

This result is strong evidence that limited resources promote the evolution of novel complex features, but it does not elucidate the specific mechanisms that are responsible for this effect. In this chapter, I examine diversity as a possible mechanism.

Some sections of this chapter have been adapted from Walker & Ofria (2012).

[1] For the treatments in this chapter and Chapter 4, the median number of generations in 100,000 updates was 27,545.85. Th'e fifth percentile was 14,322.845 generations, and the 95th was 31,601.475 generations.

3.1 Diversity as a Potential Factor in the Evolution of Complex Features

Diversity in a population is often cited as a major facilitator for the evolution of new complex features (Friedrich *et al.*, 2009). The intuition behind this dynamic is that if a population is exploring multiple regions of a fitness landscape, more opportunities exist to find new functionality. Since limited-resource environments possess multiple niches (Gavrilets & Vose, 2005; Hall & Colegrave, 2007), facilitating diverse populations and increasing exploration of the fitness landscape (Sareni & Krähenbühl, 1998; Friedrich *et al.*, 2009), I postulate that this diversity is a driving factor of the more frequent evolution of EQU in a limited-resource environment in Avida. In this chapter, I explore the effect of multiple limited resources on phenotypic Shannon diversity and, in turn, on the evolvability of the EQU complex task.

Of the many types and measures of diversity, we choose to examine phenotypic Shannon diversity. We choose phenotypic over genotypic diversity because we are interested in how limited resources promote task acquisition; since tasks define an organism's phenotype, and only phenotype affects the environment, phenotypic diversity is the natural choice. (Note that, since the environments in this chapter reward each distinct task only once, grouping genotypes by phenotype, ecotype, or binary task signature is equivalent: of these possible terms, I choose to refer to *phenotypic* diversity since it is most likely to be intuitively understood.)

We choose to measure phenotypic diversity as the Shannon entropy of these groupings because Shannon entropy effectively balances the two main interesting qualities in diversity: the range of possible results and the evenness in their distribution. To be precise, we calculate phenotypic Shannon diversity as:

$$H = -\sum_{i} P(x_i)\, log(P(x_i)) \tag{3.1}$$

where x_i is each genotype group, and $P(x_i))$ is the number of organisms in that group divided by the number of organisms total in the population.

We show that Shannon diversity peaks at intermediate levels of resource availability to the population, and we map the evolvability of the EQU task on this availability-diversity gradient. While the evolvability of EQU is highest at intermediate availabilities, it does not peak at the same resource inflow level as phenotypic Shannon diversity, and it is more robust than diversity in its response to inflow level. These results indicate that while phenotypic Shannon diversity may play into the evolution of complex features, the selective pressures caused by diversity cannot be the only pressures behind such evolution.

3.2 Experimental Environments

Resource inflow and availability is a major factor affecting ecosystem diversity (Tilman, 1982; Chesson, 2000; Hall & Colegrave, 2007; Abrams *et al.*, 2008; Cardinale *et al.*, 2009). I explore the effect of the availability of multiple limited resources on phenotypic Shannon diversity, and use this availability-mediated diversity gradient to examine the relationship between Shannon diversity and the evolution of complex features.

3.2.1 The unlimited-resource environment

In their investigation of the evolutionary origin of complex features, Lenski *et al.* rewarded digital organisms once for each distinct Boolean task performed. The value of each task corresponded to its complexity as approximated by the minimum number of Boolean NAND operations necessary for its performance (see Table 3.1), using a power reward system as discussed in Section 2.3.

Function name	Reward
NOT	$\times 2^1$
NAND	$\times 2^1$
AND	$\times 2^2$
ORN	$\times 2^2$
OR	$\times 2^3$
ANDN	$\times 2^3$
NOR	$\times 2^4$
XOR	$\times 2^4$
EQU	$\times 2^5$

Table 3.1: Task rewards in Lenski *et al.* (2003). An organism that performs a task has its current execution rate multiplied by the amount of the task's reward; it can be rewarded for each task only once per lifetime.

3.2.2 The limited-resource environment

Chow *et al.* (2004) have previously performed an investigation into the relationship between resource inflow and diversity in Avida. They measured diversity as species richness. Since the digital organisms are asexual, Chow *et al.* used a clustering algorithm based on phylogenetic distance to determine which genotypes belonged to the same "species". Species richness in this setup was the result of negative frequency-dependent selection due to nine depletable resource pools, each linked to a task. R_{INFLOW} units of resource flow into each resource pool at a constant rate over each update, and 1% of each pool flows out, modeling a chemostat.

$$Inflow : R_{TASK} = R_{TASK} + R_{INFLOW} \tag{3.2}$$

$$Outflow : R_{TASK} = 0.01 \times R_{TASK} \tag{3.3}$$

Chow *et al.* used the same set of Boolean computational tasks as Lenski *et al.*, but linked each task to a separate resource pool. The amount of resource in a resource pool (R_{TASK}) determines the benefit to be gained by performing the associated task; they did not incorporate the difficulty (via NAND-count) of tasks into their environmental design. An individual organism depletes A_{TASK} units of resource from the task-linked pool when performing a Boolean task, reducing the amount available to the rest of the population and resulting in negative frequency-dependent selection (Cooper & Ofria, 2002). Rewarding an organism for the performance of a task again consists of multiplying its current execution count by the amount of the reward.

$$A_{TASK} = 0.0025 \times R_{TASK} \tag{3.4}$$

$$Depletion : R_{TASK} = R_{TASK} - A_{TASK} \tag{3.5}$$

$$Reward : \times 2^{A_{TASK}} \tag{3.6}$$

In the environments of Chow *et al.*, high inflow rates result in populations that converge on a single genotype specialized on replication efficiency: these rarely perform more than one or two of the simpler Boolean tasks. This is because the authors do not incorporate the difficulty of the task into the task's reward; at high resource abundance, there is little to no pressure to seek new resources, and thus no reason to do difficult tasks. On the other hand, the environments of Lenski *et al.* reward tasks based on difficulty. However, without the negative frequency-dependence of Chow *et al.*'s environments, populations converge to a single generalist genotype that performs all tasks (Figure 3.1).

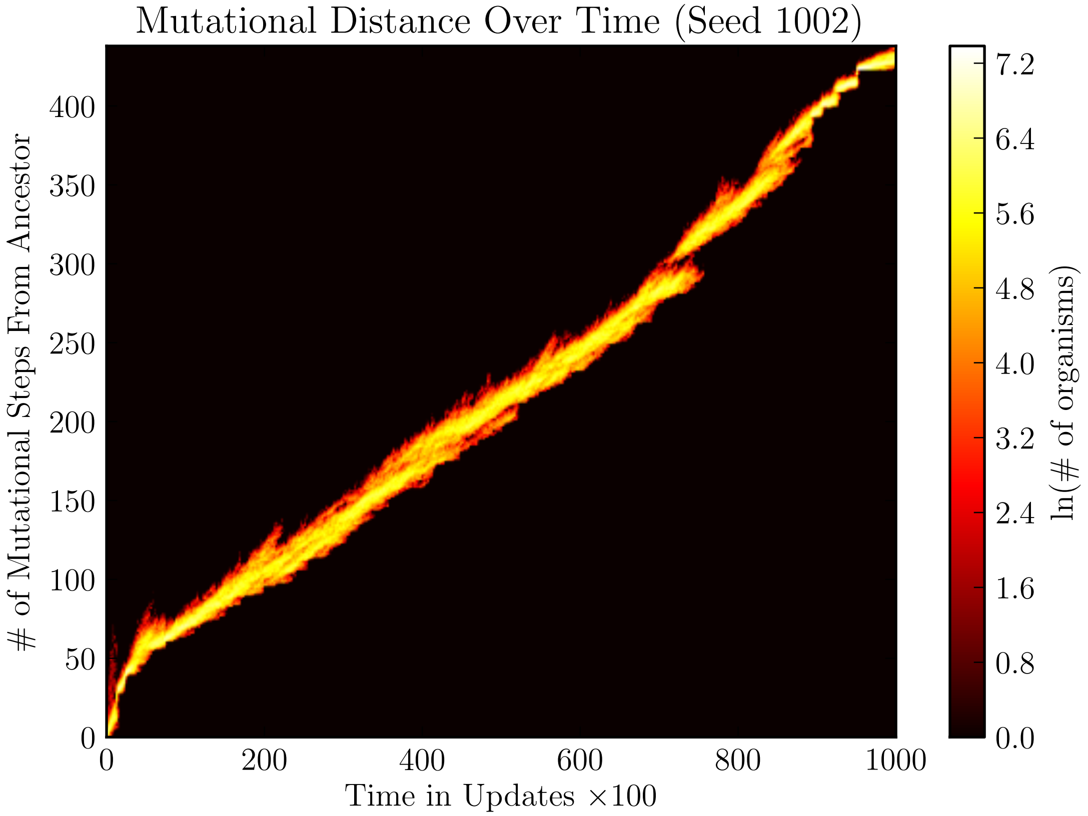

Figure 3.1: A representative pattern of mutational distance over time in a population evolving in an unlimited-resource environment. The x-axis spans 100,000 updates, while the y-axis counts the mutational steps from each currently living genotype to the original ancestor. Color values give the natural log of the number of organisms at a particular mutational distance. The population as a whole never experiences adaptive radiation, and so each member genotype remains similarly distant from the common ancestor.

In studying the effect of resource supply on both phenotypic Shannon diversity *and* the evolution of complex features, it is useful to create environments in which both the difficulty of the task and its rarity in the population (via the availability of its associated resource) contribute to the reward that organism receives for performing that task. Moreover, since we use Lenski *et al.*'s unlimited-resource environment as a control, our environment should differ from theirs as little as possible. To this end, we have devised a limited-resource environment starting with Lenski *et al.*'s difficultly-dependent reward scheme, but where linked resource pools mediate the amount of the reward as in Chow *et al.*; Table 3.2 describes this hybrid reward scheme; Figure 3.2 shows a typical example of how genetic distances from the ancestor diverge over time in the hybrid reward system.

Task	# NAND	Depletion	Reward
NOT	1	A_{NOT}	$\times 2^{1 \times A_{NOT}}$
NAND	1	A_{NAND}	$\times 2^{1 \times A_{NAND}}$
AND	2	A_{AND}	$\times 2^{2 \times A_{AND}}$
ORN	2	A_{ORN}	$\times 2^{2 \times A_{ORN}}$
OR	3	A_{OR}	$\times 2^{3 \times A_{OR}}$
ANDN	3	A_{ANDN}	$\times 2^{3 \times A_{ANDN}}$
NOR	4	A_{NOR}	$\times 2^{4 \times A_{NOR}}$
XOR	4	A_{XOR}	$\times 2^{4 \times A_{XOR}}$
EQU	5	A_{EQU}	$\times 2^{5 \times A_{EQU}}$

Table 3.2: Hybrid task rewards, based both on task complexity and resource availability (A_{TASK} denotes the number of resource units an organism uses from the $TASK$'s pool). An organism that performs a task has its current execution rate multiplied by the amount of the task's reward; it can be rewarded for each task only once per lifetime.

3.2.3 Specific configuration details

Our experiments use a development version of Avida 2.12.3 with the default instruction set (inst-heads.cfg). The executable was complied from publicly-available source code (at avida.devosoft.org); the specific revision is tagged "walker-complexfeatures".

We set the per-site copy mutation rate to 0.0025, while we left the per-divide rates of insertion and deletion mutations at the Avida default value, 0.05 each (meaning that there was a 5% chance of a single insertion and a 5% chance of a single deletion at each organism's

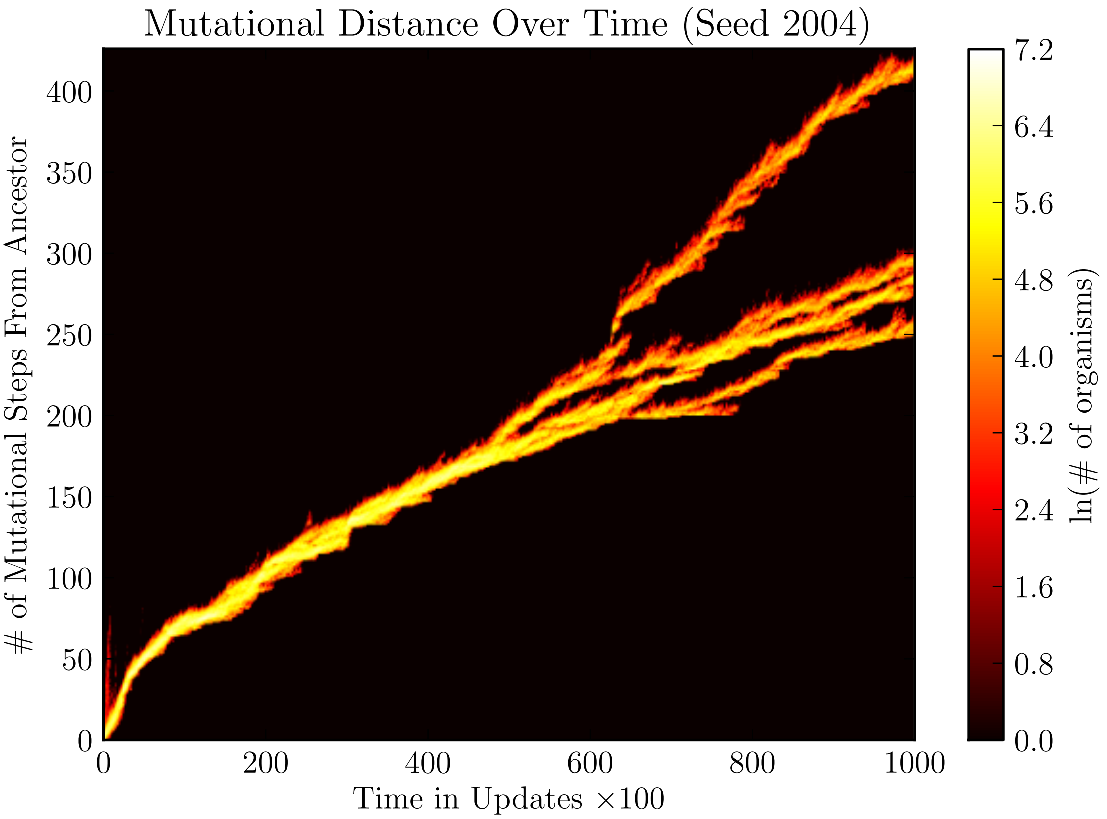

Figure 3.2: A representative pattern of mutational distance over time in a population evolving in a limited-resource environment with the hybrid reward system (Table 3.2) and $R_{INFLOW} = 100$. The x-axis spans 100,000 updates, while the y-axis counts the mutational steps from each currently living genotype to the original ancestor. Color values give the natural log of the number of organisms at a particular mutational distance. The population experiences negative frequency-dependent selection due to the hybrid reward system, and divergent lineages emerge, each with a different mutational distance to the common ancestor.

birth). The ancestor organism is a modification of the default ancestor that ships with Avida, default-heads.org, to reduce its genotype from length 100 to length 50 by removing 50 lines of “blank tape” no-op instructions; we cut it down from the current default length because the ancestor organism in Lenski *et al.* (2003) is also 50 lines long. (Chow *et al.* used a 100-length ancestor.)

3.3 Diversity Peaks at Intermediate Productivity

Of the resource inflow rates we examined, the intermediate R_{INFLOW} of 10 (Figure 3.3) had the highest diversity; observing the highly unimodal trend of these data, we conclude that diversity in this system peaks somewhere between an R_{INFLOW} of 3 and 30. At lower inflow rates, Shannon diversity drops off sharply; with too-low resource levels, each pool supports too few organisms to make any substantial impact on diversity. The number of phenotypes may remain high, but the Shannon entropy of the population as a whole is low. At higher inflow rates, diversity drops more gradually as resources become so plentiful they might as well be unlimited. Indeed, at inflow levels of 1000 and above, negative frequency-dependent pressures are effectively removed. This result is consistent with the results in other studies of the effects of resource supply on diversity (e.g. Kassen *et al.*, 2000; Chow *et al.*, 2004; Hall & Colegrave, 2007)

Median diversity	∞ Inflow	100-unit Inflow	Significance
At first appearance of EQU	.034	.962	$p = 4.334 \times 10^{-56}$
At fixation of EQU	.314	.948	$p = 5.515 \times 10^{-29}$
At end of run	.314	.877	$p = 2.732 \times 10^{-53}$

Table 3.3: Detailed phenotypic diversity measures during the evolution of EQU. Shannon diversity of viable phenotypes in 200 runs each of the infinite- and 100-unit-inflow environments is measured at three points: the first appearance of EQU in the population, its fixation on the line of descent, and the end of the run (100,000 updates). P-values compare the diversity of the two treatments using a two-tailed Mann-Whitney U test with a sequential Bonferroni correction.

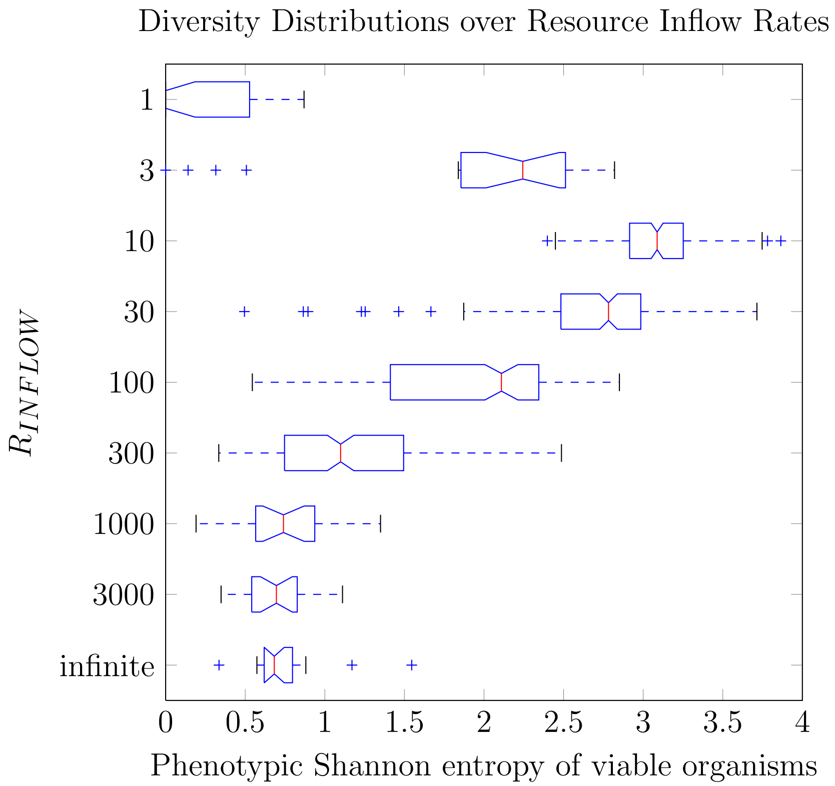

Figure 3.3: Diversity distributions across inflow rates, measured as the phenotypic Shannon entropy of all viable organisms in the population. Data for inflows 1, 3, 1000, 3000, and infinite are drawn from 20 populations; inflows 10, 30, 100, and 300 from 200 populations.

3.4 The Evolution of EQU Peaks at (Higher) Intermediate Productivity

We are now equipped to examine the evolvability of complex features on this resource inflow gradient, and to observe how it relates to the corresponding Shannon diversity gradient. In this case, we measure the evolvability of complex features by the proportion of populations that have evolved EQU by the end of 100,000 updates. At 20 populations per treatment (Figure 3.4), it is clear that the evolvability of EQU is highest at intermediate productivities. Indeed, between the intermediate inflow levels of 10 and 300 units per resource per update, the evolvability of EQU seems robust to increasing resource supply and decreasing phenotypic

Shannon diversity.

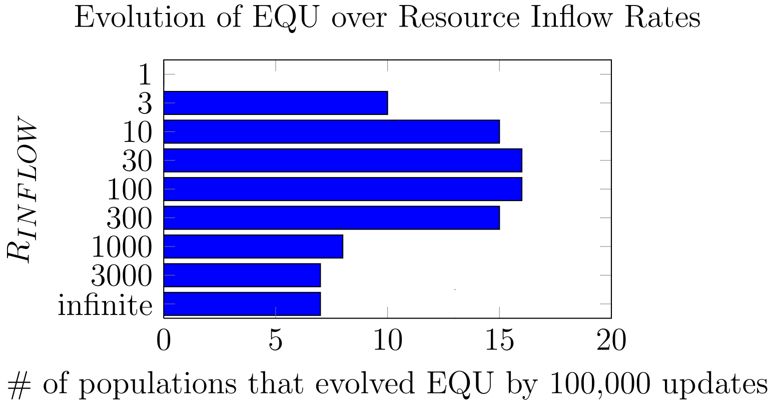

Figure 3.4: Number of populations for which some genotype in the final population can perform EQU across a range of resource inflow rates. Data for all inflows is drawn from 20 populations.

To determine whether only complex tasks are sensitive to resource supply levels, we also examined the evolvability of the other 8 tasks rewarded in this environment (Figure 3.5). As a general trend, these tasks indicate that more complex tasks are more sensitive to the resource supply level.

We focused on the inflow rates in which populations were most successful in evolving EQU (10, 30, 100, and 300), and performed 10 times as many experimental runs at each to gain a higher resolution (Table 3.4). At this resolution, we saw that the evolvability of EQU is not truly unaffected by the variation of resource supply and Shannon diversity in this inflow range. The number of populations evolving EQU by the end of 100,000 updates is significantly higher at the 100 unit inflow rate than at the 10, 30, or 300 unit inflow rates. While these data do not indicate the precise R_{INFLOW} at which the evolutionary potential of EQU peaks, it is clearly a different – and greater – R_{INFLOW} than that at which phenotypic diversity reaches its peak.

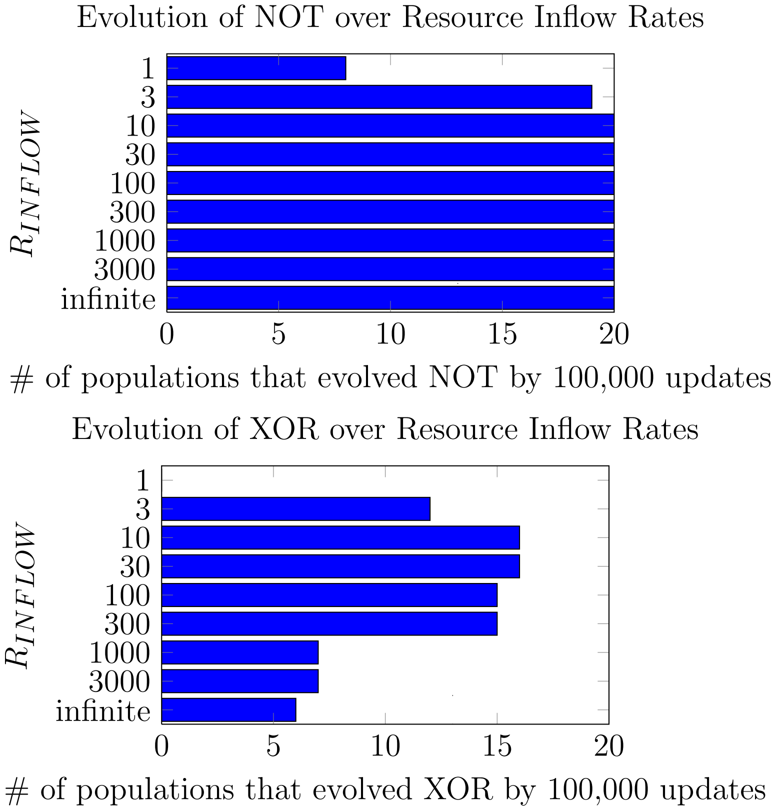

Figure 3.5: Number of populations which evolved tasks requiring fewer NAND operations than the EQU task. The first seven tasks (only NOT is shown) showed results across resource inflows that were qualitatively similar to the NOT task shown here, with low numbers of populations evolving the task at $R_{INFLOW} = 1$ and very high numbers for all other inflow rates. The XOR task shows results qualitatively similar to the EQU task, with XOR being more likely to evolve at intermediate values of R_{INFLOW}. Data for all inflows and tasks is drawn from 20 populations.

R_{INFLOW}	10	30	100	300
$\#pops/200$	141	152	171	152
p-value	< 0.00001	< 0.045	N/A	< 0.045

Table 3.4: Number of populations out of 200 that evolved EQU at intermediate inflow rates. We performed a chi-squared test to determine if the evolvability of EQU for at least one inflow rate differed significantly from the rest (p<.005, $\chi^2 = 13.156$, 3 degrees of freedom). With this confirmed, we calculated the significance of each ratio's difference from 171/200 (the result at $R_{INFLOW} = 100$) with Fisher's exact test, two-tailed, and corrected with the sequential Bonferroni correction; the n=2 correction was applied to both the $R_{INFLOW} = 30$ and $R_{INFLOW} = 300$ data, since they can be ordered arbitrarily.

3.5 Populations at Intermediate Diversity Evolve EQU More Often Even When It Is Not Rewarded

The Shannon diversity of phenotypes in the 100-inflow limited-resource environment is substantially higher (Figure 3.3, Table 3.3; also see Figure 4.1), and both diversity and the evolution of EQU peak within a range of intermediate resource rates. However, these data demonstrate only correlation, and it is clear that the evolvability of EQU does not peak in the same environmental conditions as phenotypic diversity.

To tease out the effects of diversity itself, rather than some co-correlate, on the evolvability of EQU, we observed that if diversity is responsible for the more frequent evolution of EQU via expanded exploration of the fitness landscape, we expect mutations to lead to EQU more frequently, even when EQU is a) unrewarded and b) prevented from fixing.

3.5.1 Preventing the fixation of EQU

To test this hypothesis, we created treatments in which EQU was unrewarded for three important inflow rates: infinite inflow (unlimited resources), 10-unit inflow (the inflow region of highest diversity), and 100-unit inflow (the region of highest evolvability for EQU). We prevent the appearance of EQU-granting mutations in the population by testing each new organism before it is added to the population. We evaluate it in a test environment; if it can perform EQU, we record that an EQU-granting mutation happened, and replace it with a new mutation drawn from the original mutation distribution. We continue this test-and-replace (or “resampling”) process until the new organism cannot perform EQU; it is then added to the population.

We choose to resample such mutations rather than simply sterilizing the new organism because such sterilization increases the lethal load of mutations; all EQU mutations become lethal. By resampling, we distribute the imbalance across all mutational effects: the resampled mutation may be beneficial, neutral, detrimental, or lethal at the same rate as any non-resampled mutation. Further explanation of resampling and its effects on mutational load can be found in Covert III (2011).

Testing the new organism in a test environment is not flawless. Such a test will catch organisms that perform EQU deterministically, but not necessarily those which perform it stochastically. Stochastically performing EQU is a rare behavior in our experiments, but we do see it. To remove stochastically-performing EQU mutations from the population, we change the reward for performing EQU to always multiply the performing organism's metabolic rate by 0. If any organism performs EQU, it is effectively sterilized. Because such mutations are not caught in the test environment, they are not recorded as EQU-granting mutations. This sterilization induces a selective pressure for organisms to avoid portions of the mutational landscape in which it is simple to evolve stochastic EQU performance. However, since there is nothing that differentially rewards stochasticity across the environments, we believe that this does not affect the overall trend.

3.5.2 Testing the EQU-granting mutation encounter rate

EQU-granting mutations were replaced in the unlimited-resource populations at a median of 3.6 per 1000 updates, significantly less frequently than 28.6 times in 100-unit inflow populations ($p < 10^{-8}$, Mann-Whitney U test, two-tailed). The 10-unit inflow populations encountered EQU-granting mutations only 2.9 times per 1000 updates, again significantly less often than the 100-unit inflow populations ($p < 10^{-13}$, Mann-Whitney U test, two-tailed). Not only do populations in the unlimited-resource and 10-unit inflow environments fail to fix EQU as often as those in the 100-unit inflow environment, mutations in these environments do not produce EQU as often in the first place.

3.6 Low Individual Distinct Task Counts Limit the Evolution of EQU

We hypothesize that this greater resource availability and lesser phenotypic diversity represent environments where more-abundant resources allow the desperate scramble for survival to relax slightly, allowing organisms to accumulate a collection of building blocks necessary for complex tasks.

Lenski *et al.* showed that the evolution of EQU relies on simpler tasks as building blocks, implying that fewer tasks performed by each individual might constrain the evolution of EQU. We therefore examined task performance in each of the three environments. Although the populations in the 10-inflow environment performed the same *range* of task types as populations in other environments, each individual performed significantly fewer distinct tasks (Table 3.5).

R_{INFLOW}	# Tasks per individual		
	5th %ile	50th %ile	95th %ile
10	1	2	3
100	4	6	8
unlimited	5	6	7

Table 3.5: Number of distinct resources used per individual. The 5th, 50th, and 95th percentiles for each environment are displayed. We compared the distributions from the uncapped 100-inflow and unlimited environments to the uncapped 10-inflow environment using a two-tailed Mann-Whitney U test; for both, $p \ll 10^{-12}$.

To test the importance of the number of distinct tasks each individual performs, we constructured variations for each of the three environments in which organisms were rewarded for a maximum of three distinct tasks, as summarized in Table 3.6. With this restriction, the number of populations acquiring EQU in the limited-resource environment was not significantly different than in the severely-limited environment (131/200 vs. 122/200, p=0.4068, Mann-Whitney U test). We conclude that the average number of tasks performed by an individual limits the evolvability of EQU, even when the population is highly diverse.

R_{INFLOW}	# Tasks per individual			# Populations	Significance
	5 %ile	50 %ile	95 %ile	evolving EQU	
10	1	2	3	122/200	$p = 0.5350$ (vs. 129/200)
100	3	3	3	131/200	$p = 4.711 \times 10^{-4}$ (vs. 167/200)
unlimited	3	3	3	17/200	$p = 7.578 \times 10^{-9}$ (vs. 67/200)

Table 3.6: Distinct tasks and the evolution of EQU with a three-task cap. The 5th, 50th, and 95th percentiles for each environment are displayed. The number of populations that evolved EQU in each treatment is also given; p-values compare the capped treatment to the uncapped treatments within each environment using two-tailed Fisher's Exact tests with a sequential Bonferroni correction.

3.7 Conclusion

I conclude that although wider exploration of the fitness landscape is certainly responsible for more frequent evolution of EQU, diversity cannot be the only driver of this phenomenon. While the evolvability of complex features is indeed *high* at peak Shannon diversity, it seems that complex features require more productive environments than those which produce the highest diversity to evolve most often.

Building block tasks, however, appear to be essential to the evolution of EQU. In addition to encouraging adaptive radiation and maintaining diversity, the negative frequency-dependent selection in the limited resource environments makes task rewards dependent on the population's current task performance. In Chapter 4, I investigate how this environment-dependent flux in task rewards mitigates the cost of trading one task for another, and how the decreased penalty for switching out tasks plays into the evolution of EQU.

Chapter 4

Reduced Cost of Trade-Offs in Limited Resource Environments Promotes the Evolution of Complex Features

In Chapter 3, I discussed the close relationship between diversity and the potential for evolving complex features in a limited-resource environment. However, increased diversity is only one effect of the negative frequency-dependent selection active in that environment: tying task rewards to resource pool levels not only enables the development of stable multi-niche communities, but also switches the emphasis of task value from task complexity to task rarity among the members of the community.

Mutations that cause the loss of access to depleted resources have a reduced impact in such an environment, which facilitates the crossing of fitness valleys (Weissman *et al.*, 2009). In Section 4.1, I examine whether such mutations – that is, those in which at least one task is lost, but no tasks are gained – have an effect on the evolution of the EQU complex feature.

Extending this idea, I examine task trade-off mutations, in which the mutation causes simultaneous task loss and task gain. In the unlimited-resource environment, such a trade-off is deleterious if the new task's reward is lower than the rewards any lost tasks. Moreover, because the contribution of the lost tasks is always substantial, such a trade-off is never ideal: in the unlimited-resource environment, final populations usually consist of a single dominant genotype which can perform all tasks (Lenski *et al.*, 2003). However, limited resources produce a frequency-dependent reward structure where losing access to multiple depleted resources for a single untouched resource is almost always advantageous; the lost tasks were not contributing significantly to the organism's fitness. I hypothesize that this effect promotes novelty – in practice, it promotes the evolution of complex tasks.

Following the results of Chapter 3, I examine only one limited-resource environment in this chapter, the 100-inflow variant that proved most friendly to the evolution of EQU.

4.1 Task Loss Without Gain

Deleterious mutations can help populations cross fitness valleys and are thus important for adaptive evolution (Covert III, 2011). We hypothesize that the reduced harm of task loss when the associated resources are depleted may contribute to the more frequent evolution of EQU in the limited-resource environment. Mutations causing task loss with no concurrent gain immediately preceded the introduction of EQU in 42 of the 167 lineages that produced this task in the limited-resource environment, but in only three of 67 such lineages in the unlimited-resource environment. In fact, two of the task losses in the limited-resource lineages were actually beneficial: the loss of the task was accompanied by a substantial increase in efficiency.

4.1.1 Testing for task loss mutations: technical restrictions

In order to determine when a mutation causes a "loss without gain" (LwG) mutation, we ran each genotype of the lineage in a test CPU and recorded which tasks it did, then compared those tasks to the tasks of its predecessor. The test CPU method has limits: namely, it cannot guarantee detection of tasks that are performed stochastically. This limitation results in false positives: mutations tagged as LwG that are not true losses, merely stochastically performed tasks that did not show up in the test CPU.

We have discounted those LwG mutations where we are convinced they are false: namely, those that both have a beneficial effect, which we do not expect from such a mutation, and do *not* decrease the gestation time of the genotype, which might otherwise account for the beneficial effect. The fitness data and gestation time data used in the screen are taken directly from the actual population record, and are therefore a faithful measure of what happened during the actual evolution of the lineage. Technical limits prevent us from outputting the tasks performed during evolution: for stochastic tasks, different organisms of the same genotype may perform different tasks. We track lineages by genotype rather than by organism, so we have no way of knowing which tasks the actual ancestors on the lineage performed.

4.1.2 Deleterious "loss without gain" mutations do not detectably affect the evolution of EQU

To isolate the importance of this particular type of deleterious mutation, we evolved 200 additional populations in the unlimited-resource and limited-resource treatments, sterilizing any mutant that experienced task loss without simultaneously gaining at least one other task; see Covert III (2011) for a detailed description of this process in the general case of deleterious mutations.

In the unlimited-resource environment, 55/200 populations evolved to perform EQU, and in the limited-resource environment 160/200 populations did so. While both results are less than the original treatments, we were not able to detect a significant difference in either (Fisher's Exact test, two-tailed, 67/200 vs. 55/200 $p = 0.2322$, 167/200 vs. 160/200 $p = 0.4375$). While deleterious task loss without gain mutations do occur on the path to evolving EQU, we were unable to find evidence to establish them as an important factor in the success of populations in the limited-resource environment.

4.2 Task Trade-Offs: Loss Accompanied by Gain

A common assumption is that trait evolution is restricted or biased by fitness trade-offs. (Roff & Fairbairn, 2007) Trade-offs have been demonstrated in the Avida system, where the evolution of new complex tasks is often associated with the loss of simpler tasks (Lenski *et al.*, 2003). I show that limited resources *facilitate* evolutionary task trade-offs, reducing the cost of sacrificing a common trait associated with a depleted resource in exchange for a less common alternative, that is, decreasing the associated fitness trade-off. This dynamic, in turn, promotes the evolution of rare tasks, which in practice are typically more complex.

For example, assume that many organisms are performing the NAND task. Resource levels for a performed task tend to settle roughly between 30 and 100 units of resource available, so also assume roughly 100 units of the NAND-linked resource exist in the environment. Any particular organism can use .25% of the resource available (capped at 1.0 unit of resource), so the performance of NAND would be rewarded by a metabolic rate multiplication of only

$2^{1\times(100\times.0025)}$, or 1.189. On the other end of the scale, the first organism to perform EQU would have access to an untapped resource pool with well over 400 units of resource, thus hitting the hard consumption cap of 1.0 unit. This untapped pool results in a metabolic rate increase of $2^{5\times(1.0)}$-fold; that is, by 32-fold – the same amount as in the unlimited-resource environment.

As mentioned earlier, in the unlimited-resource environment, a task loss must be compensated for with new tasks of equal or greater combined value. Moreover, losing even a single task always loses at least a ×2 multiplier to metabolic rate. The limited-resource environment relaxes this restriction. The opportunity cost for losing a widely-used resource is less – as calculated above, the organism may lose as low a metabolic rate multiplier as ×1.189. As a result of this, the community's average multiplier is lower, so the relative reward for accessing an unexploited resource is greater. This effect allows many mutations that involve task trade-offs to be beneficial in the limited-resource environment even though they would have been detrimental in the unlimited-resource environment.

4.2.1 Trade-offs at the evolution of EQU

We identified the 167 mutations that first conferred EQU in the populations from the limited-resource environment, each of which was beneficial when it occurred. However, had these mutations happened in the unlimited-resource environment, 50 (29.94%) of them would have been either neutral (9) or deleterious (41) (Figure 4.1) and thus would be unlikely to fix the ability to perform EQU. It seems likely that discouraging the fixation of EQU in this manner is responsible for at least some of the unlimited-resource environment's poor performance in evolving EQU.

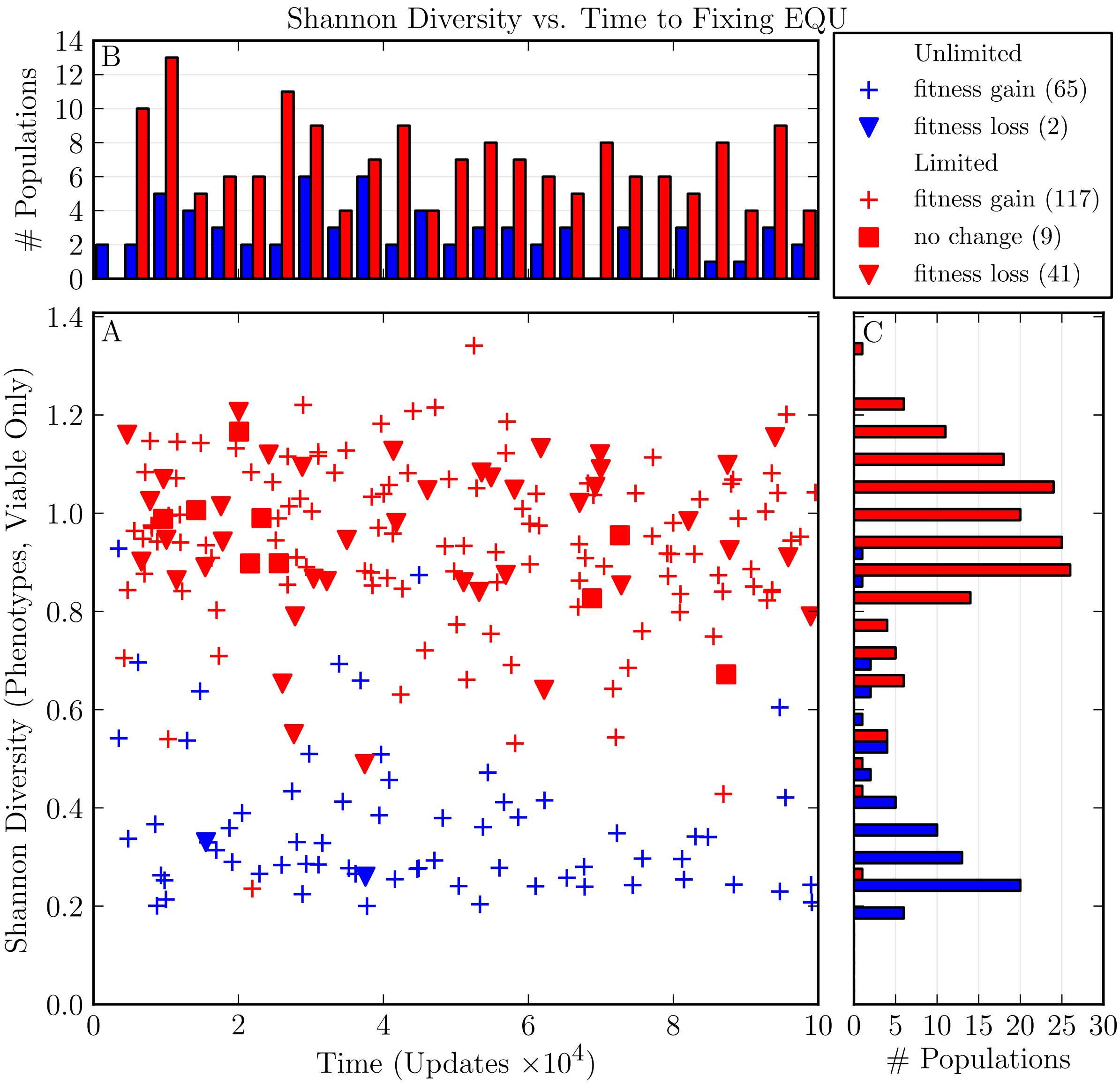

Figure 4.1: Shannon diversity vs. fixation time of EQU. (A) Diversity of each population that evolved EQU versus time at which EQU fixed on the line of descent. Blue markers represent populations evolved in the unlimited-resource environment, while red represent those evolved in the (100-inflow) limited-resource environment. Symbols denote the fitness effect of the EQU-granting mutation as evaluated in the unlimited-resource environment. Beneficial mutations are marked with +, neutral with ■, and detrimental with ▼; the legend indicates how many mutations in each treatment fall into these categories. Histograms show the distribution of (B) the fixation of EQU over time and (C) the phenotypic diversity when EQU fixes. Appendix C explains how two of the unlimited-resource populations were able to fix detrimental EQU-granting mutations.

4.2.2 Eliminating the possibility of a neutral or detrimental trade-off for EQU

To test the hypothesis that some EQU-granting mutations do not fix in the unlimited-resource environment because of task trade-off costs, we designed a variant reward system. We raised the reward for performing EQU from a $\times 2^5$ metabolic rate multiplier to $\times 2^{21}$. The total multiplier from all other tasks remains $\times 2^{20}$. Therefore, with this reward system, EQU-granting mutations are beneficial even if they cause the loss of all other tasks.

Out of 200 populations using this reward structure in an unlimited-resource environment, 100 could perform EQU at the end of 100,000 updates, a significant increase from 67 populations with the original reward structure ($p = 0.006883$, Fisher's Exact test, two-tailed, sequential Bonferroni correction), although still significantly fewer than the 169 from the limited-resource environment with the original reward structure ($p = 1.507 \times 10^{-11}$, Fisher's Exact test, two-tailed, sequential Bonferroni correction). We conclude that removing the possibility of a detrimental effect when sacrificing other tasks for EQU increases the evolvability of EQU.

4.2.3 Eliminating the possibility of a neutral or detrimental trade-off when trading "up"

However, this variant reward structure removed the detrimental effect of trade-offs only for EQU. Lenski *et al.* (2003) showed that EQU requires the lower-difficulty tasks as building blocks, and data from Section 3.6 also indicated that the presence of other tasks increases the probability of EQU evolving. We therefore extended our hypothesis: the limited-resource environment is better suited to evolving EQU not just because it provides an incentive to trade other tasks for EQU, but in fact because it provides an incentive for any trade-off of a common task for a rare task.

We tested this extended hypothesis by designing another variant reward structure, called EQU-81, in which *any* trade-off for a more complex task is always beneficial (Table 4.1). The values associated with the nine tasks are designed such that each increase in task complexity (still approximated by the necessary number of NAND operations) confers a value V greater than the sum of all previous values combined, that is, the $\times 2^V$ multiplier is twice as large as the product of all multipliers for less complex tasks. Using this reward structure, 136/200 populations in an unlimited-resource environment evolved EQU, significantly more than the 67/200 populations with the original reward structure ($p = 8.478 \times 10^{-11}$, Fisher's Exact test, two-tailed, sequential Bonferroni correction) and the 100/200 populations in the EQU-21 reward structure ($p = 2.877 \times 10^{-03}$, Fisher's Exact test, two-tailed, sequential Bonferroni correction). However, this number of populations is *still* significantly fewer than the 167/200 in the limited-resource environment with the original reward structure ($p = 3.015 \times 10^{-3}$, Fisher's Exact test, two-tailed).

Function name	Reward
NOT	$\times 2^1$
NAND	$\times 2^1$
AND	$\times 2^3$
ORN	$\times 2^3$
OR	$\times 2^9$
ANDN	$\times 2^9$
NOR	$\times 2^{27}$
XOR	$\times 2^{27}$
EQU	$\times 2^{81}$

Table 4.1: The EQU-81 reward structure. Any trade-off for a more complex task is guaranteed to be beneficial. An organism that performs a task has its current execution rate multiplied by the amount of the task's reward; it can be rewarded for each task only once per lifetime.

We conclude that providing an incentive to trade off simple tasks for more complex ones is an important aspect of the limited-resource environment, responsible for some of its greater propensity to evolve EQU. However, the trade-off cost mitigation cannot fully account for the greater frequency of EQU evolution in limited-resource environments.

4.3 Conclusion

We conclude that both diversity and ease of trade-offs for rare tasks play into the greater evolution of complexity in limited-resource environments. A limited-resource environment – as long as it is not so severely limited as to restrict the areas of the fitness landscape that may be explored – acts as a negative frequency-dependent pressure to maintain both of these traits, significantly increasing the probability of complex tasks evolving.

Chapter 5

Predicting the Outcome of Competitions Between Communities

Thus far, I have focused on the evolution communities from the perspective of the origin of new traits, with some examination of community diversity. In this chapter I will further examine how these evolved communities stack up against each other, with a particular focus on determining if certain community-level traits can predict the outcome of competitions.

Interaction among individuals is of vastly different character in a system of many niches than in a single niche, especially when some limiting factors (such as space or resources) are shared across well-defined niches, requiring that individuals compete both within niches and across them. Thus, the communities formed by these interactions are true ecologies, exhibiting properties on the community level as well as the individual level.

For the multiple niche environment I again turn to a limited resource setup. In this chapter, I use the default limited-resource environment that ships with Avida. In contrast to the limited-resource environment used in Chapters 3 and 4, the reward system in this environment is additive, as explained in Section 2.3. Since metabolic rate does not increase exponentially in such a system, Avidians may to receive a reward for up to 25 performances of a particular task. The phenotype is therefore not equal to the binary task signature; for these communities, I group genotypes into ecotypes with the clustering algorithm described in Chapter 2.

5.1 Measures of Community-Level Properties

5.1.1 Measuring community success

I measure the success of a community A in an A vs. B competition as the fraction of individuals in the final merged community (at the end of the competition) descended from member genotypes of the original community A, the *fractional community success.* In a competition between community A and community B, if community A ends up with the larger fraction of descendant organisms in the final community, community A has “dominated” community B; if community B has *no* descendant organisms in the final post-competition community, community A has “excluded” community B.

5.1.2 Measuring community diversity

Just as community-level diversity proved to be valuable in the evolution of complex traits, it is also a likely factor in community success in competition: after all, in many vs. one competitions (i.e. invasion), more diverse communities have repeatedly proven more resistant (Case, 1990; Kennedy *et al.*, 2002; Tilman, 2004; Zavaleta & Hulvey, 2004; Maron & Marler, 2007).

5.1.3 Measuring community composition

Community composition describes the degree to which a community is composed of specialist or generalist ecotypes. I use *average ecotypic generality* to measure community composition;

it averages how generalist the community's member ecotypes are. I calculate this value as the weighted average of the task diversity of the community's member ecotypes. Each ecotype's task diversity is, in turn, the weighted average of the task diversity its member genotypes, which is the Shannon entropy of their phenotypes (i.e. task performance count vectors; see Section 2.5).

Low values indicate specialist genotypes, while high values indicate generalists. For example, an organism that does NAND 90 times and performs other tasks has 0 entropy; on the other hand, an organism that does all nine tasks but only 10 times each has an entropy of 2.1972 (in nats).

5.1.4 Measuring community breadth

Community breadth describes how well the community as a whole fills its available niches. *Task coverage* is the Shannon entropy of the community-level task performance vector, that is, the sum of the phenotypes of all member organisms. Lower numbers indicate that the community as a whole covers a small portion of the available niches, while higher numbers indicate communities that cover most niches evenly. I would expect low community breadth to be a poor strategy in competition, since it means this community is underexploiting some of the available niches. High task coverage, on the other hand, tells us nothing about the community's composition; it might be the result of a complete community of specialists (with low average ecotypic generality) or a community consisting of a single complete generalist (with high average ecotypic generality).

5.1.5 Measuring community resource use efficiency

I measure community resource use efficiency via *differential total-resource use.* When comparing communities A and B, I calculate this measure as the difference between the total amounts of resource remaining in each community when resources have been drawn down. In this case, a higher total amount of resource remaining in isolation indicates a community that is not efficiently using its available resources. The metric thus ranges from negative values when community A is more efficient, and so has a lower total resource level, to positive values when community B is more efficient.

Task coverage, the measure of community breadth, may also be thought of as a measure of resource use efficiency: high task coverage means the community is using all resources. However, It is not a good measure for directly *comparing* the resource use efficiency of two communities: it measures only the degree to which a community uses all resources equally, not the level to which it is capable of drawing down those resources.

In this chapter, I consider only differential total-resource use as a measure of resource use (in)efficiency. These values can sometimes be very large if only one of the competitors is capable of doing a particular task.

5.2 Generation of Test Communities

In order to study how community structure is tied to predictability of community success, I have performed community competition experiments with four types of communities: one evolved, and three constructed.

1. *Evolved* communities are seeded with a single ancestor organism, capable of self-replication but not of performing any Boolean tasks. I ran 100 communities for 100,000

updates in the additive limited-resources environment, then for a further 100,000 updates with mutations turned off so that the community structure could settle out. These 100 evolved communities also provide the genotype pool from which the other communities are assembled.

2. *True random* communities are assembled by randomly assigning the genotypes of the original 100 evolved communities to 100 new communities (without replacement). The only constraint is that each assembled community must have at least 3 genotypes. To satisfy this constraint without introducing any other form of bias, I simply re-ran the assembly algorithm until this property held true. (In practice, this took at most 100 tries.)

3. *Ecologically complete* communities are generated one-by-one from the genotypes of the original evolved communities such that each new community can perform all tasks. When assembling each of the 100 ecologically complete communities, the nine tasks are ordered randomly. For each task, a genotype that can perform that task is randomly selected (without replacement) from the pool of potential genotypes. This process continues for each unincluded task until the assembled community contains at least one genotype that performs each task. Once the core of each community has been assembled, any genotypes remaining in the pool are randomly assigned across the assembled communities. In practice, this method was always able to assemble the full 100 communities because the source pool had substantial redundancy.

4. *Specialist-preferred ecologically complete* communities use the previous assembly technique, but specialist genotypes are preferred during community construction. In practice, this meant that communities that were assembled first tended to be dispropor-

tionately composed of specialists.

5.3 Community Competition

To compete two communities, I create a Avida world with 7200 cells and place both communities into it (original communities contain 3600 organisms each). All the organisms of each community are marked with an inherited tag, allowing me to distinguish their descendants at the end of the competition.

I start the experiment such that the two populations are kept separate, with separate resource pools. The environments are the same as those in which the communities evolved. The communities replicate in isolation for 10,000 updates, allowing them to draw down resources to their equilibrium levels. At the end of those 10,000 updates, the two communities are joined, and the two sets of resource pools are combined into one global set of pools. The organisms are now free to replicate into any cell in the combined world, and the communities interact through this replication and through drawing on the combined resource pools.

5.3.1 Self-competition

In order to test this community competition method, I competed each of the 100 base communities against a copy of itself. Each copy of the community was provided a unique tag for tracking purposes. As community A and community B had identical composition, I expected to see community A dominate 50% of the time. I observed community A dominate in 52 competitions, and community B in 48 competitions. At a p-value of > 0.8875 (Fisher's Exact Test, two-tailed), this ratio does not appear substantially different than the expected 50/50 split. This result implies that the competition method is likely to be mechanically sound, not promoting either community simply for its identity as the A (or B) community.

5.3.2 Competition results are consistent

How well does the result of a single competition between two communities predict the results of replicate competitions between those two communities? That is, how well does the dominance result of a single competition predict the dominant community in replicate competitions?

I created 50 independent competitor pairs from the 100 evolved base communities. I replicated these 50 competitions 10 times each. I observed how many pairs had inconclusive dominance – that is, not all replicates dominated by the same original community. If a single competition were always a perfect predictor of all competitions between a pair, I expected to see 0/50 instances of inconclusive dominance. For those competitor pairs with inconclusive dominance, I examined to what degree the competitions were inconclusive. I found that of 50 the competitor pairs, only one had exhibited inconclusive dominance, indicating that the dominance result of a single competition between a pair is highly predictive of the dominance outcomes of all competitions between that pair.

Competitions between assembled communities showed similar consistency. In fifty independent competitions between the randomly assembled communities (again with 10 replicates per competition), only 3/50 displayed inconclusive dominance. In fifty competitions between the ecologically complete assembled communities, 1/50 were inconsistent; 2/50 competitions between independent pairs of specialist-weighted ecologically complete communities were inconsistent.

I examined the inconclusive competitions more closely. For the replicates of the competitor pair of evolved communities with inconclusive dominance, all ten final communities consisted of A's descendants and B's descendants in nearly equal proportions: the replicate with the greatest imbalance had 55.14% of the final population descended from B, and the one with the least imbalance had 50.85% descended from A.

Five of 50 competitions between independent pairs of randomly assembled communities were inconclusive. In two of them I observed the same roughly 50% dominance as described above. In the remaining three, success seems to have been dependent on some tipping point: either one competitor community dominates by some percentage (in these cases, community A), or the other community (B) completely excludes it. If indeed dominance in these competitions is dependent on some stochastic event, it seems possible that extending the competition period would result in the exclusionary community eventually winning every competition. If the stochastic event is historically contingent or capable of affecting only the early organization of the combined community, however, these competitions are unpredictable and simply have two stable states.

Seven of 50 competitions between independent pairs of ecologically complete assembled communities displayed inconclusive dominance. Only one was the easily explicable type, with every dominance result near 50%. Three displayed the "tipping point" pattern described above, appearing to be dependent on some stochastic event to determine the winner. Of the remaining cases, one displayed consistent results of roughly 75% dominance by community B in 9/10 replicates; in the odd one out, community A dominated by 50.16%. It seems possible that this could be caused by an unlikely loss of a key member of community B. In the final two cases, the winning community dominated by at least 65%, but both communities were winners over the ten replicate populations. Without detailed examination of the data from these runs, it is difficult to say what could cause such an effect: perhaps these two competitions are truly unpredictable.

Nine of 50 competitions between independent pairs of specialist-preferred ecologically complete assembled communities were inconclusive. Two displayed roughly 50% dominance in each competition; the remaining seven appear to be of the type in which the outcome is determined stochastically. Three of these stochastically-determined competitions are truly unpredictable: in each replicate, the dominating community completely excluded the other community. There is thus no chance that extending these three runs would have led to a conclusive result.

For competitions between evolved communities I observed no qualitatively inconclusive results. It is therefore reasonable to do only one competition per competitor pair when competing evolved communities, and let that result approximate the results of all competitions for that pair. This is especially true when using a non-binary measure of dominance such as fractional community success, described above, because it reveals these qualitatively consistent results.

5.3.3 Evolved communities have a nearly-linear dominance hierarchy

In order to make predictions about dominance, factors must exist that determine which community will win in a competition. As long as these factors are not cyclic in nature, there must be a dominance hierarchy. From 20 additional evolved base communities I constructed all possible non-self competitions and then counted the number of intransitive dominance triads, e.g., community A dominates community B, and community B dominates community C, yet community C dominates community A. Each competition consisted of only one replicate; as observed above, for competitions of evolved communities, one replicate provides a reliable approximation of the outcome.

I observed 6 intransitive triads out of an expected 285 (if dominances are assigned randomly) and a maximum possible 330 (Kendall & Smith, 1940). This result shows a strongly linear dominance hierarchy ($p \ll 0.001$, Appleby K-test (Appleby, 1983) with $\chi^2 = 165.75$, $df = 26$), which indicates that a predictor of community dominance exists.

I did not test for dominance hierarchies among the assembled communities; there are no such guarantees that they can be predicted.

5.4 Correlations with Community Success

In this section, I examine how the candidate predictors correlate with fractional community success in competitions between evolved communities. If I have chosen reasonable predictors, I expect to see a strong correlation with at least one of them, since the presence of a nearly linear dominance hierarchy indicates that a predictor should exist. I examine the same set of candidate predictors for the assembled communities, asking: if we know which community measures correlate with success for evolved communities, what does this tell us about their correlations with success for assembled communities?

To test for correlations between our hypothesized predictors and fractional community success, I used the non-parametric Spearman's rank correlation ρ (Savicky, 2009; R Core Team, 2012). Because this is a non-parametric test relying on rank, and in some cases the data contains tied values, the p-values obtained and given below are not guaranteed to be exact. However, the ρ values are high enough that I am confident of significance at the $p = 0.05$ level. All p-values for correlations within the same set of competitions have been corrected via the sequential Bonferroni method with $n = 4$.

5.4.1 Community diversity

In competitions between evolved communities, I found significant correlation between a community's fractional success and the difference between its ecotypic diversity and that of its competitor ($p < 0.0006$, Spearman's rank correlation test, $\rho = 0.5129$, $S = 10143.86$). Figure 5.1 graphs this relationship for each of the 50 competitions between evolved communities.

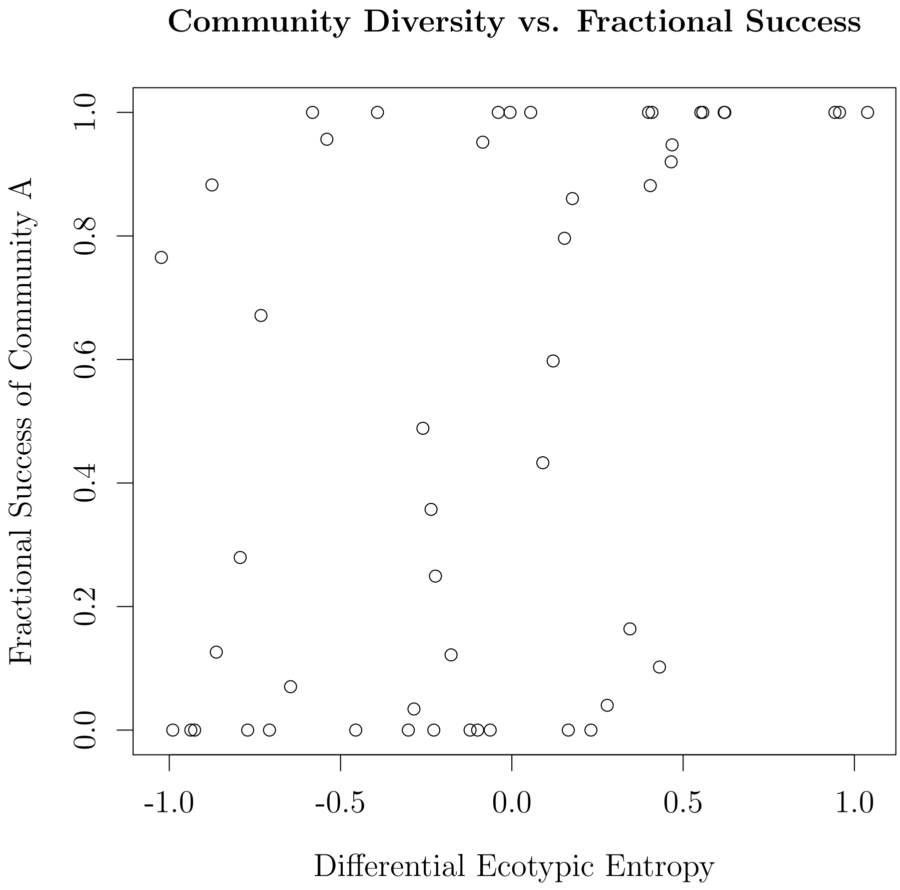

Figure 5.1: Community diversity vs. fractional community success for evolved communities in competition. Spearman's $\rho = 0.5129$.

In the case of competitions between assembled communities, the data could not determine whether there was significant correlation for any of the three types. For randomly assembled communities, $p > 0.08$, $\rho = 0.2452$, and S=15719.48. For ecologically complete communities,

$p > 0.9$, $\rho = 0.001468$, and $S = 20794.43$; for specialist-preferred assembled communities, $p > 0.3$, $\rho = -0.1232$, and $S = 18248.76$.

Two of the three assembled communities are explicitly constructed to be ecologically complete, and a stable ecologically complete community is likely to have higher diversity than a stable but incomplete community. I therefore I suspected that it might be competitions involving non-complete evolved communities driving the correlation between success and diversity for evolved communities. I created a subset of the fifty competitions for which both competitor communities were ecologically complete ($n = 40$) and examined the correlation between success and community diversity on just this subset. It remained significant ($p = 0.000274$, $\rho = .553$, $S = 4768.342$, uncorrected). I conclude that in the competitions between evolved communities, diversity is truly correlated with community success; the relationship is not being driven by outlier non-ecologically-complete competitors.

5.4.2 Average ecotypic generality

Fractional community success in competition for evolved communities was also significantly correlated with the average ecotypic generality of the community ($p < 0.0006$, Spearman's rank correlation test, $\rho = -0.5207$, $S = 31669.38$); with a negative ρ, this indicates a positive correlation between fractional success and average ecotypic specialization (Figure 5.2).

I also found a significant relationship in the competitions between the randomly assembled communities ($p < 0.003$, $\rho = 0.4471$, $S = 11513.54$), but with a *positive* ρ, indicating that in these competitions those communities with *generalist* member ecotypes were more likely to dominate (Figure 5.3). Random communities with generalists are more likely to be ecologically complete, so this result is not inconsistent.

However, all ecologically complete assembled communities are guaranteed to have been

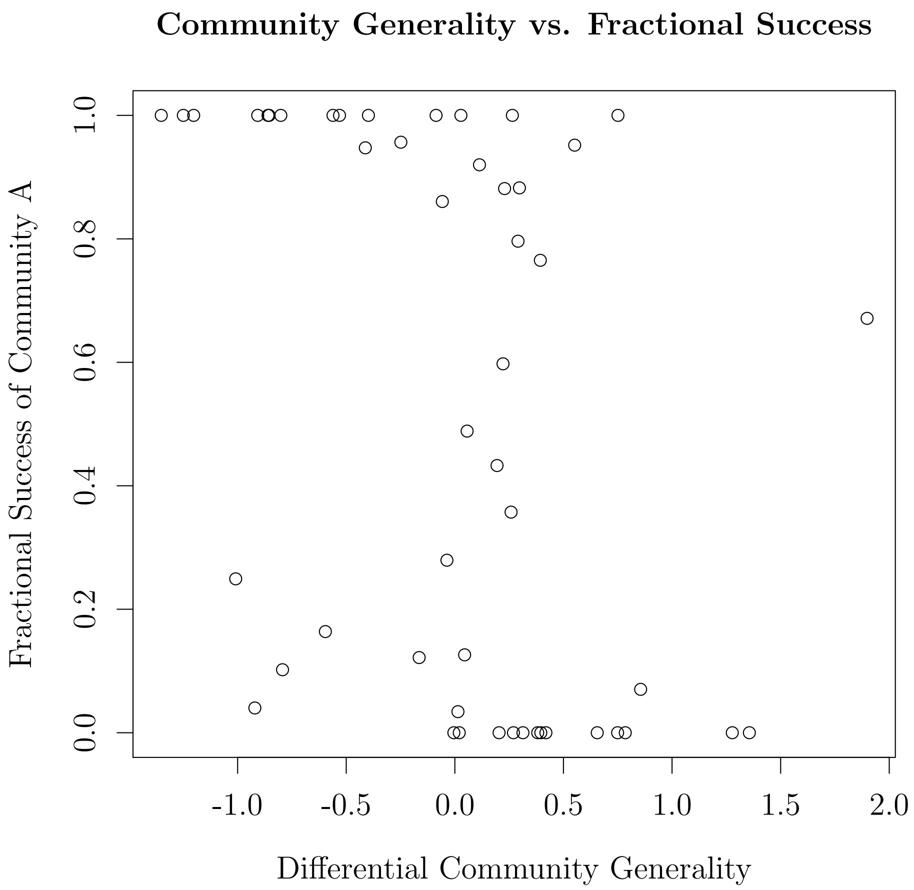

Figure 5.2: Community generality vs. fractional community success for evolved communities in competition. Spearman's $\rho = -0.5207$.

so before their 50,000-update ecological settling-out period, but ecotypic generality is also positively correlated with success for these competitions ($p < 0.035$, $\rho = 0.4178$, $S = 31669.38$; see Figure 5.4). I conclude that there may be an effect of community origin (i.e. evolved as a community vs. assembled from a pool of genotypes) on how generality plays into success in community competition.

In competitions between the specialist-weighted assembled communities, there was insufficient evidence to reject the null hypothesis that average ecotypic generality was not correlated with community success ($p > 0.1$ after correction, $\rho = 0.2917$, $S = 13911.04$).

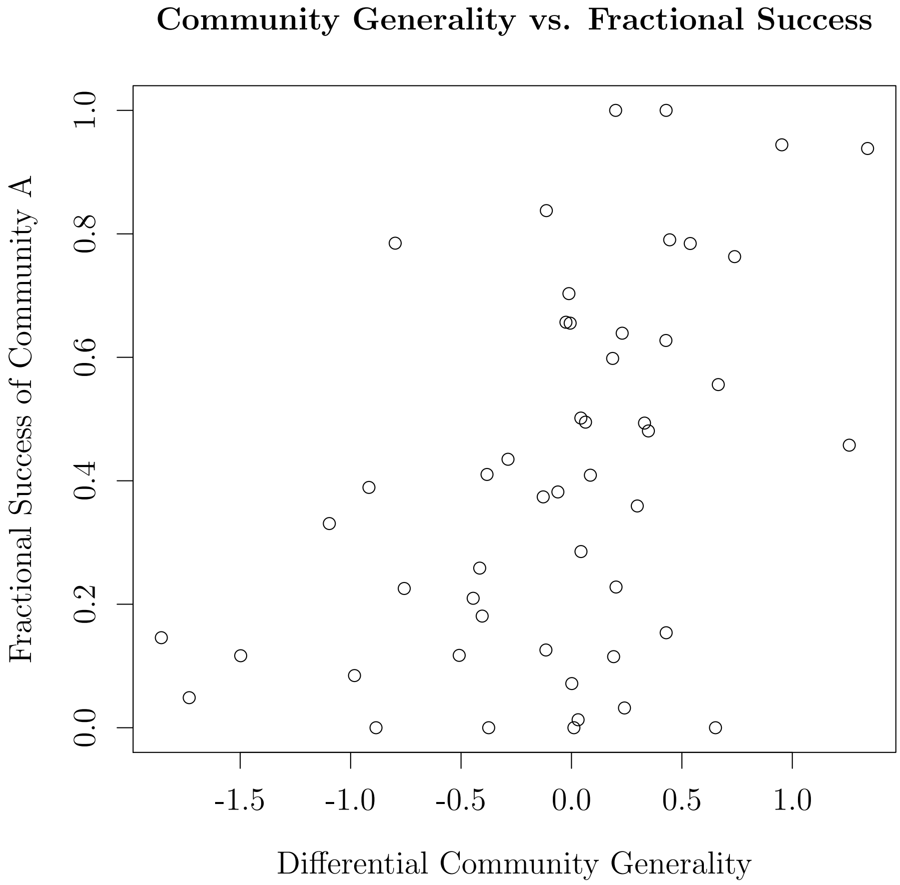

Figure 5.3: Community generality vs. fractional community success for randomly assembled communities in competition. Spearman's $\rho = 0.4471$.

5.4.3 Community breadth

In competitions between randomly assembled communities, community breadth measured as task coverage was significantly correlated with community success ($p < 0.003$, Spearman's rank correlation test, $\rho = 0.4558$, $S = 11332.49$; see Figure 5.5). I also found correlation between success and task coverage for ecologically complete assembled communities ($p < 0.035$, $\rho = 0.3419$, $S = 13705.12$; see Figure 5.6) and for specialist-preferred ecologically complete assembled communities ($p < 0.05$, $\rho = 0.3422$, $S = 13698.7$; see Figure 5.7).

This correlation is intuitive for randomly assembled communities; since there is no im-

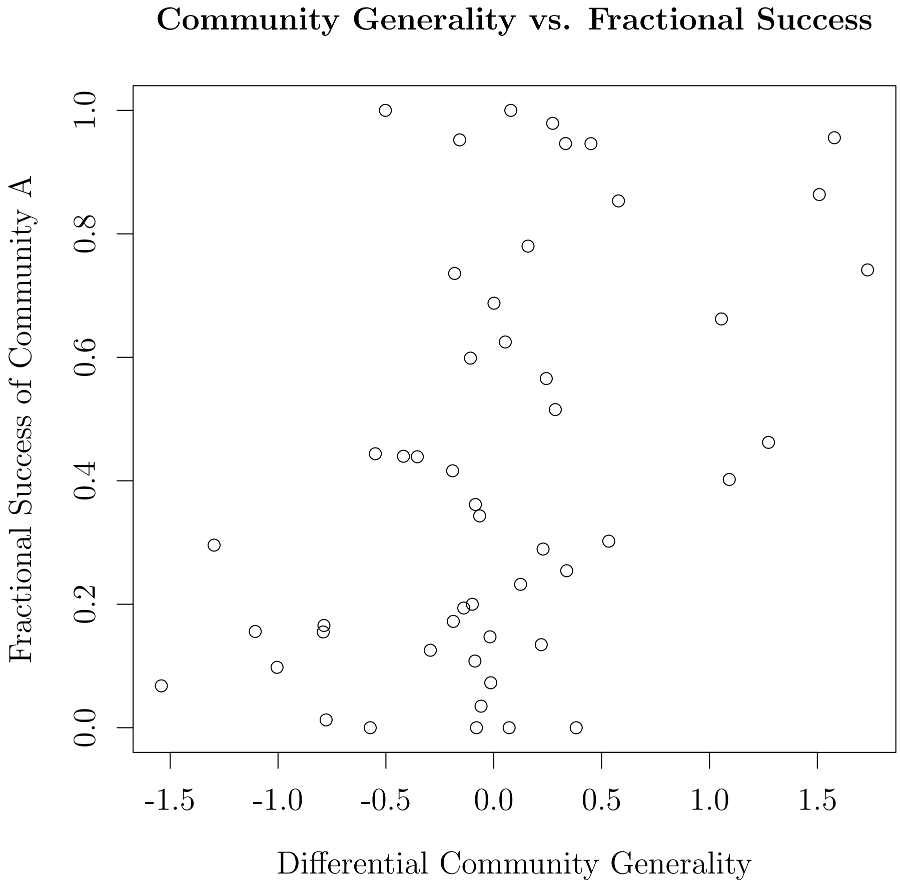

Figure 5.4: Community generality vs. fractional community success for ecologically complete assembled communities in competition. Spearman's $\rho = 0.4178$.

plicit or explicit requirement for them to be ecologically complete, much less cover all tasks equally, the lucky communities that ended up with good task coverage are likely to be more successful. In the case of both types of ecologically complete assembled communities, there is still no requirement that the assembled community cover all tasks equally well – as long as at least one member can perform each task at least one time, the community is considered complete. This correlation thus implies that a balanced occupation of niches is valuable in competition even when both communities nominally occupy all available niches.

There was not sufficient evidence for correlation between success and task coverage in

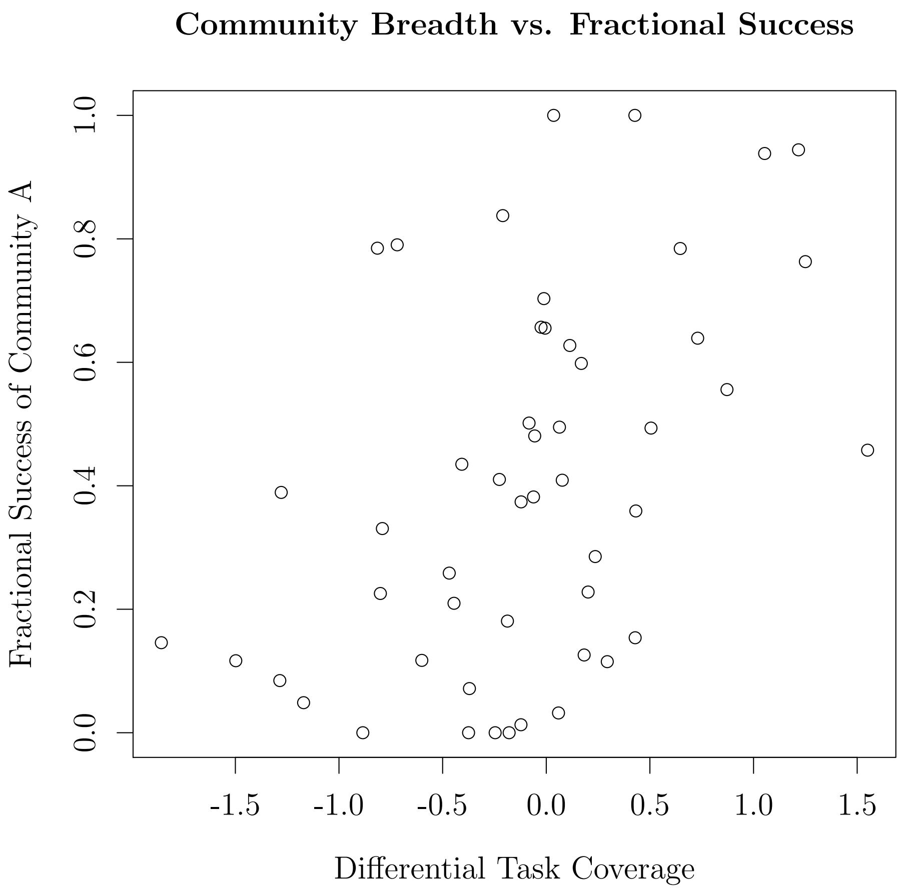

Figure 5.5: Community breadth vs. fractional community success for randomly assembled communities in competition. Spearman's $\rho = 0.4558$.

competitions between evolved communities ($p > 0.07$, $\rho = -0.2538$, $S = 26110.99$); evolution in a limited-resource environment provides an implicit negative frequency-dependent pressure to occupy all open niches as equally as possible. Each community therefore covers all the tasks its members are capable of fairly evenly, and the difference between communities that are not ecologically complete and those that are, while part of this metric, is not enough to make task coverage a significant correlate in this case.

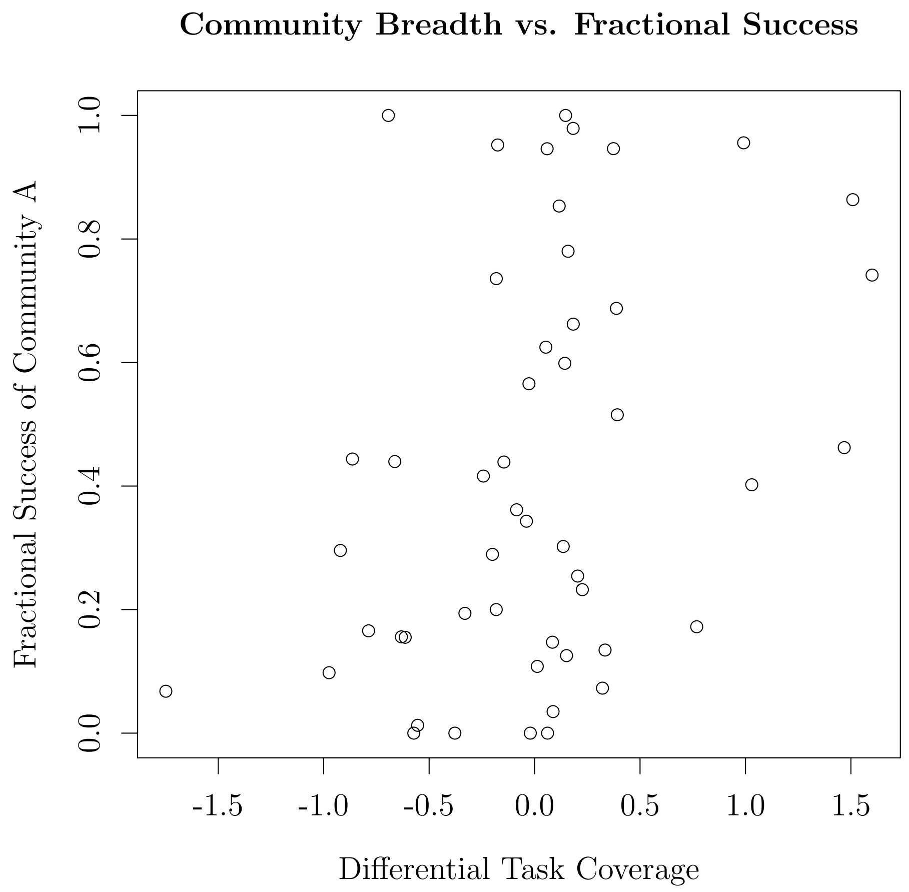

Figure 5.6: Community breadth vs. fractional community success for ecologically complete assembled communities in competition. Spearman's $\rho = 0.3419$.

5.4.4 Resource use efficiency

Differential total unused resource showed a significant correlation with fractional community success in competitions between the evolved communities ($p < .003$, $\rho = -0.4525$, $S = 30250.19$), although the correlation appears to be driven by outliers (Figure 5.8).

I therefore also evaluated the correlation between total unused resource and community success for the subset of competitions in which both evolved communities were ecologically complete, so that no untapped resources affected the difference in total unused resource. For this subset, ρ is lower, and the p-value, 0.0198, is higher. Nevertheless, p is still significant,

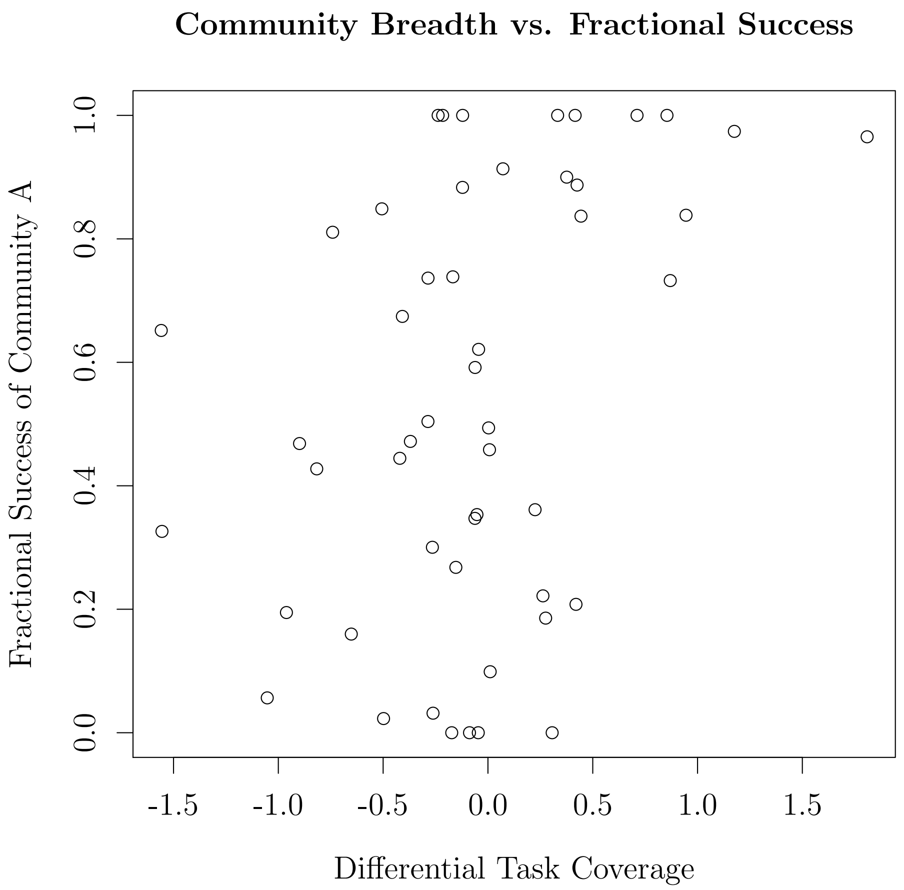

Figure 5.7: Community breadth vs. fractional community success for specialist-preferred ecologically complete assembled communities in competition. Spearman's $\rho = 0.3422$.

and would remain so if this test were incorporated into the sequential Bonferroni correction, indicating that the extreme outlier unused resource differences are not driving the entirety of this correlation.

The correlation was also significant for the randomly assembled communities ($p < 7.3 \times 10^{-6}$, $\rho = -0.6293$, $S = 33930.46$), between the ecologically complete assembled communities ($p < 0.001042$, $\rho = -0.5003$, $S = 31242.75$), and between the specialist-preferred assembled communities ($p < 0.02$, $\rho = -0.3949$, $S = 29048.04$): resource use efficiency is correlated with community success in competition no matter what the origin of the commu-

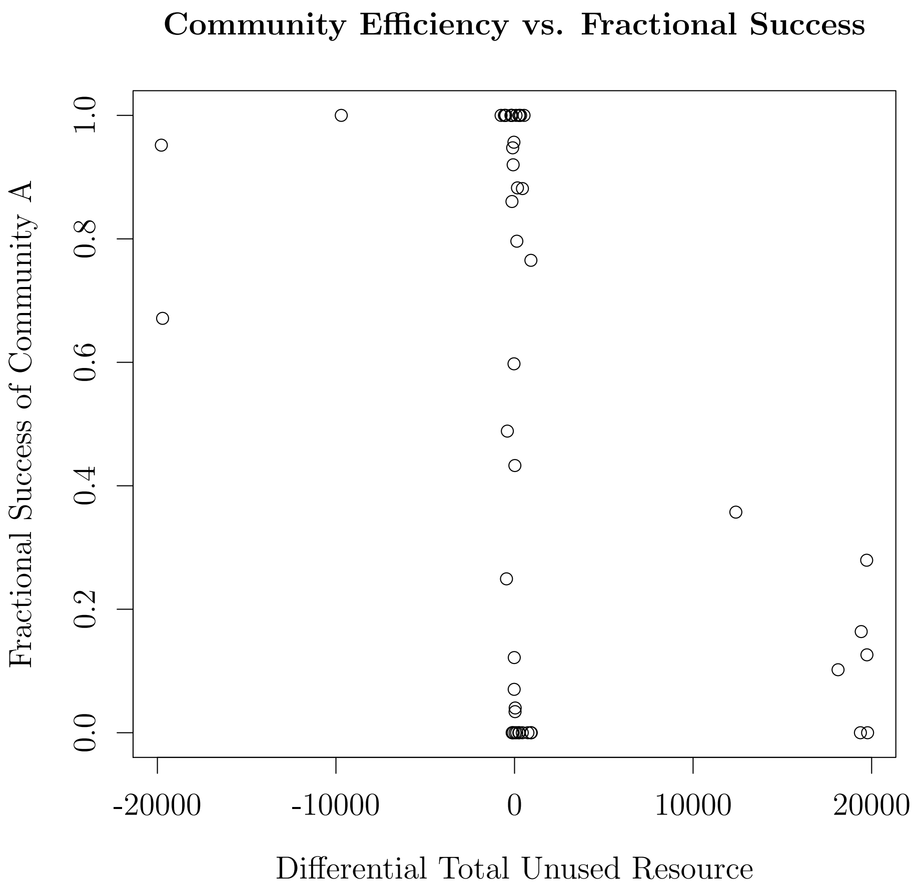

Figure 5.8: Community resource use efficiency vs. fractional community success for evolved communities in competition. Spearman's $\rho = -0.4525$.

nity (Figures 5.9, 5.10, and 5.11).

As the measure of differential resource use efficiency actually has negative values when the dominant community draws resource levels lower than its competitor, i.e. is more efficient, it is not surprising to see a negative ρ for these correlations: it indicates a positive correlation between fractional community success and the ability to draw resources down to a lower level than the competitor community.

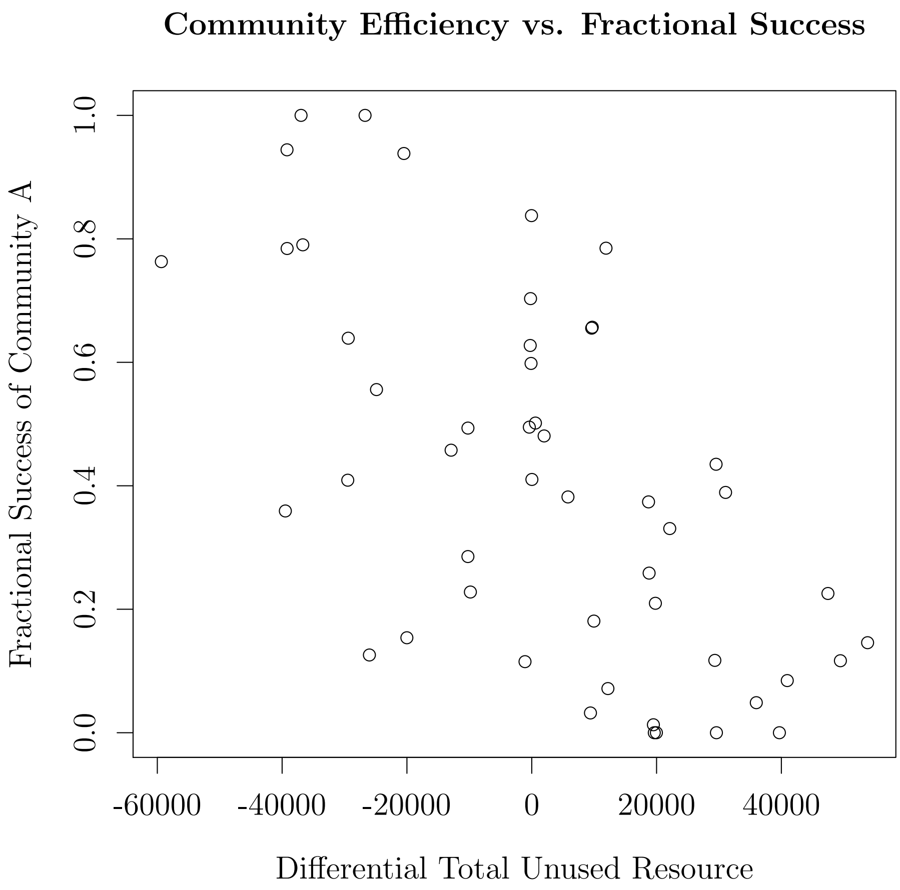

Figure 5.9: Community resource use efficiency vs. fractional community success for randomly assembled communities in competition. Spearman's $\rho = -0.6293$.

5.5 Linear Models

Of the predictors I have considered, it seems that diversity, generality, and resource use efficiency may play into the success of evolved communities in competition. I therefore constructed all one- and two-factor linear models combining these predictors (including two-factor models with interaction). I additionally constructed one containing all predictors and one with all predictors all interactions. I calculated the AIC and BIC values for each model (Bolker & Team, 2012; R Core Team, 2012); Table 5.1 shows the five models with AIC differences less than three, as well as the weights assigned to each model. I conclude

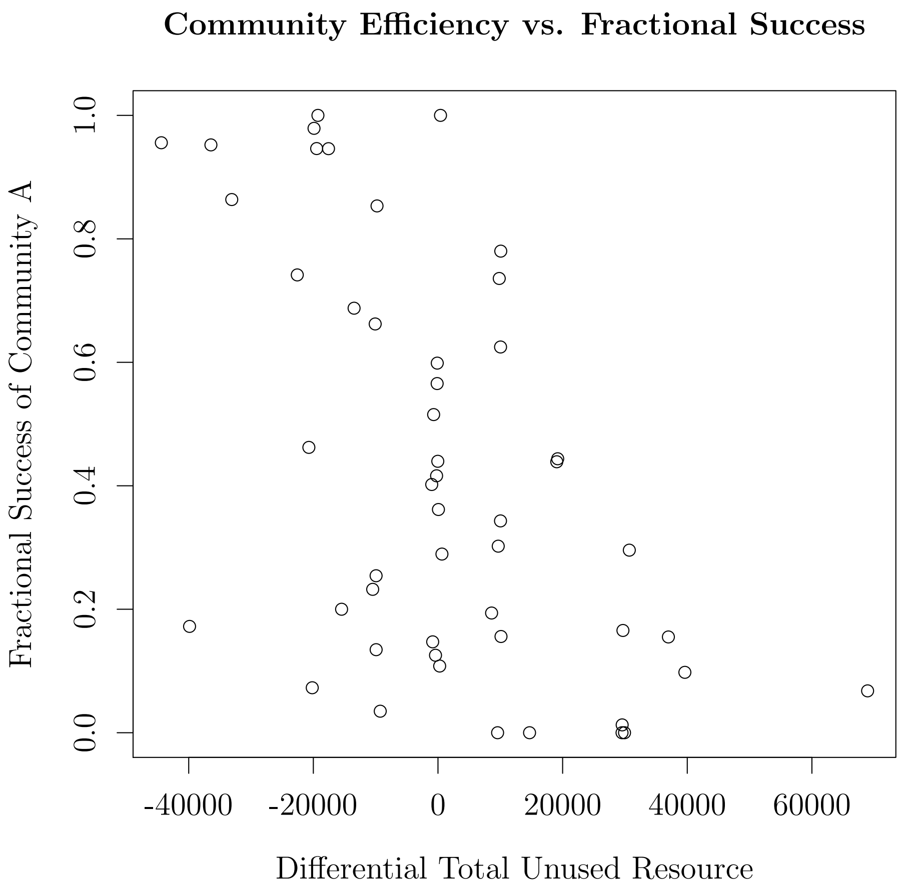

Figure 5.10: Community resource use efficiency vs. fractional community success for ecologically complete assembled communities in competition. Spearman's $\rho = -0.5003$.

that differential resource efficiency and community generality together can predict fractional community success (that is, not just predict which community will win, but by how much). The adjusted R^2 value for this model is 0.3425, indicating that the model predicts 34.25% of the variation in fractional community success. Community diversity is seems largely unhelpful for prediction unless community generality is not known.

Generality, task coverage, and efficiency each showed correlation with fractional success for the competitions between randomly assembled communities and between ecologically complete assembled communities. I therefore constructed the same families of linear models

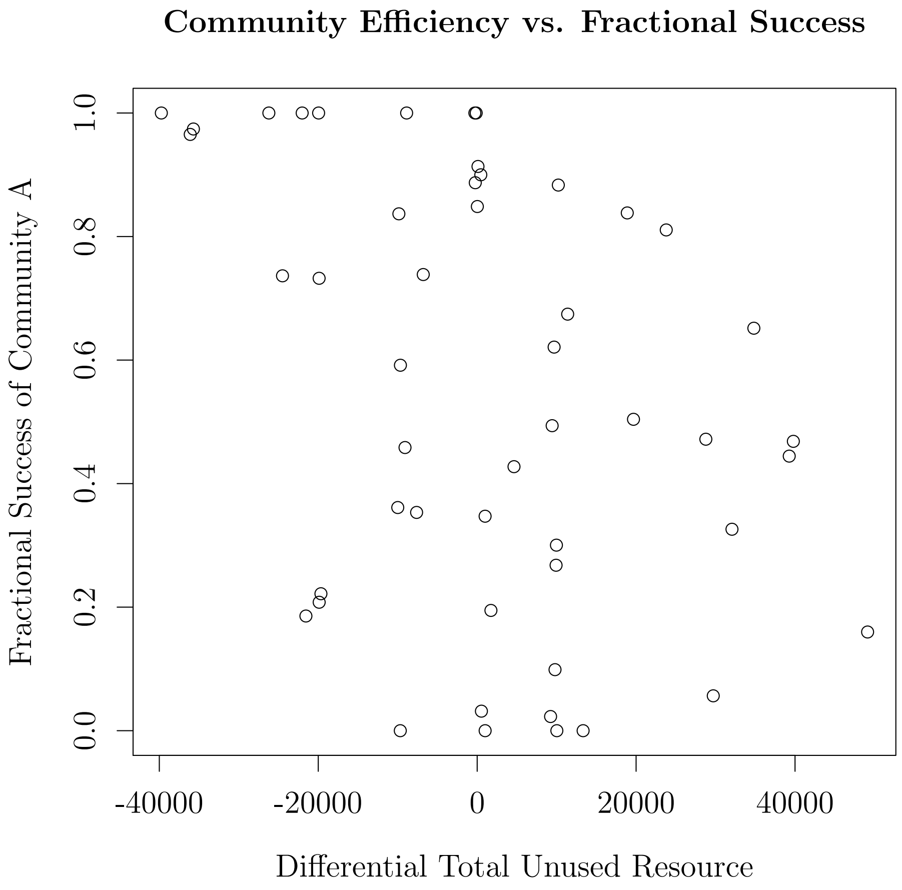

Figure 5.11: Community resource use efficiency vs. fractional community success for specialist-preferred ecologically complete assembled communities in competition. Spearman's $\rho = -0.3949$.

Model	AIC weight	BIC weight
$fcs = 0.558 + -0.303g + -0.0000260e$	0.355	0.536
$fcs = 0.566 + \mathit{0.1591d} + \mathit{\text{-}0.212g} + -0.0000235e$	0.289	0.167
$fcs = 0.556 + -0.306g + -0.0000257e + \mathit{\text{-}0.00000151ge}$	0.133	0.0770
$fcs = 0.573 + -0.384d + -0.0000242e + \mathit{\text{-}0.0000187de}$	0.127	0.0735
$fcs = 0.572 + 0.345d + -0.0000189e$	0.0972	0.157

Table 5.1: Linear models predicting the fractional community success (fcs) of evolved communities based on differential community diversity (d), generality (g), and resource use (in)efficiency (e). The five models with AIC differences below three are displayed; AIC and BIC weights (considering only these five models) are displayed. Factors that are insignificant at $p = .05$ inside each model are italicized.

for each: a null model, three one-factor models, three two-factor models without interaction and three with interaction, and two full models, with and without all two- and three-way interactions. For each community type, I examined the AIC and BIC values of all the models; I present those models with AIC difference of less than three. In the case that the preferred model as judged by BIC was not one of these, I present that as well.

Table 5.2 presents four models of fractional community success in competition for randomly evolved communities. Resource use efficiency is the only factor in the favored model, and the only significant factor in the models with the next three highest AIC weights. I conclude that resource use efficiency is the most important predictor of fractional community success for these communities. The model including only the efficiency factor accounts for 38.66% of the variance (adjusted R^2) in fractional community success.

Model	AIC weight	BIC weight
$fcs = 0.422 + -0.00000710e$	0.472	0.735
$fcs = 0.419 + \textit{-0.0528c} + -0.00000822e$	0.216	0.130
$fcs = 0.422 + \textit{-0.00252g} + -0.00000715e$	0.174	0.104
$fcs = 0.393 + \textit{0.0518g} + -0.00000665e + \textit{-0.00000228ge}$	0.139	0.0319

Table 5.2: Linear models predicting the fractional community success (fcs) of randomly assembled communities based on differential community generality (g), task coverage (c), and resource use (in)efficiency (e). These are the four models with AIC difference less than three; AIC and BIC weights (when considering only these models) are displayed. Factors that are not significant at $p = .05$ inside each model are italicized.

For ecologically complete assembled communities in competition, only two models of fractional community success had AIC weight less than three: these were the full model and the full model with all interactions, presented in Table 5.3. However, the highest weight model in the BIC table was the one-factor model including resource use efficiency, $fcs = 0.425 + -0.000008203e$; this is also the only significant factor in the full model with interactions. This one-factor model accounts for 27.5% of the variation in fractional community success for ecologically complete communities, while the full model incorporating all three factors explains 34.36%. I conclude that community generality, breadth, and efficiency are all important in predicting success for the ecologically complete assembled communities.

Model	AIC weight	BIC weight
$fcs = 0.421 + 0.318g + -0.374c + -0.00000940e$	0.794	0.994
$fcs = 0.401 + \mathit{0.212g} + \mathit{\text{-}0.298c} + -0.0000115e$ $+\mathit{0.135gc} + \mathit{\text{-}0.00000809ge} + \mathit{0.0000109ce} + \mathit{0.00000183gce}$	0.206	0.00463

Table 5.3: Linear models predicting the fractional community success (fcs) of ecologically complete assembled communities based on differential community generality (g), task coverage (c), and resource use (in)efficiency (e). These are the two models with AIC difference less than three; AIC and BIC weights (when considering only these models) are displayed. Factors that are not significant at $p = .05$ inside each model are italicized.

Only community breadth (i.e., task coverage) and community resource use efficiency were significantly correlated with fractional community success for competitions between specialist-preferred ecologically complete communities. I therefore constructed only a null linear model, the two one-factor models, and full two-factor models with and without interaction. All models except the null model had AIC distance less than three; I present them in Table 5.4. For such communities, both task coverage (breadth) and resource use efficiency appear to be important predictors. However, the two single-factor linear models each predict only about 12% of the variance in success on their own, and the models incorporating both factors do no better, leading me to conclude that for these communities breadth and resource use efficiency are roughly equivalent. Note that of all the community types, competitions between the specialist-preferred communities were least consistent (above), so it is not surprising that the results of these competitions cannot be predicted as well as those between other community types.

Model	AIC weight	BIC weight
$fcs = 0.553 + 0.208c$	0.508	0.381
$fcs = 0.449 + -0.00000613e$	0.349	0.262
$fcs = 0.553 + \textit{0.136c} + \textit{-0.00000315e}$	0.118	0.231
$fcs = 0.532 + \textit{0.134c} + \textit{-0.00000341e} + \textit{-0.00000212ce}$	0.025	0.127

Table 5.4: Linear models predicting the fractional community success (fcs) of specialist-preferred ecologically complete assembled communities based on differential community task coverage (c) and resource use (in)efficiency (e). These are the two models with AIC difference less than three; AIC and BIC weights (when considering only these models) are displayed. Factors that are not significant at $p = .05$ inside each model are italicized.

5.6 Conclusion

The fractional success of an evolved community in competition with another evolved community is both strongly consistent and predictable by community-level measures: in this case, the differential community generality and resource use efficiency between the two can predict up to 34.25% of the variation in fractional success. In fact, differential resource use efficiency is a valuable predictor in competitions between any two communities of the same type, both evolved and assembled.

Community generality is a predictor for evolved communities and two types of assembled communities, but biases the competition in different ways between the evolved and assembled communities. In linear models predicting the success of evolved communities in competition, the coefficient of generality is negative. However, in most linear models predicting the success of *assembled* communities in competition, the coefficient of generality is positive; when it is negative, the generality term is insignificant in the model. I observed the same trend with the sign of Speaman's ρ when calculating correlation between community success and generality: this provides strong evidence that differential community generality biases competitions between evolved communities differently than it does those between assembled communities, and suggests that there is something fundamentally different in their structures.

In nature, we often do not have access to detailed data about every member of a community, but we can estimate community-level measures such as the ones laid forth in this chapter. The predictors I have examined here are in common use for characterizing the competition of two populations or the success of an invasive species. I have shown here that they are also informative measures for competitions between entire communities and can predict what fraction of the resulting post-competition community will be descended from each competitor. In this chapter I have shown only correlation relationships; while community-level properties are difficult to manipulate cleanly even in computational systems like Avida, I have shown that experiments which might be able to tease out a causal relationship between any one of these properties and fractional community success are worth facing that difficulty.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

I have examined the diversity, task reward structure, and competition of digital communities, the result of adaptive radiation due to negative frequency-dependent selection in limited-resource environments. They are more diverse than those evolving in environments where the only resource is space to replicate, and evolve complex features more readily. Tying the rewards of complex and building-block tasks to resource levels additionally reduces the cost of trading simple tasks for complex ones, and I have shown that this has a strong causal relationship with the evolution of complex features.

Additionally, I have shown that the results of competitions between communities, especially evolved communities, are consistent and predictable. These results establish a strong basis for future work in this area that will further explore the relationships between community-level and ecotype-level characterizations of communities and how that relates to their ability to survive in competition. Moreover, investigations at the ecotype level will allow us to understand how two communities combine when the outcome of competition is not a simple exclusion of one community by another.

Systems of limited resources have shown significant promise in evolutionary computation (Goings & Ofria, 2009; Goings, 2010; Goldsby, 2011), with applications in multi-objective optimization and in problem decomposition. A building-block task or “hint” sub-solution can be easily rewarded simply by incorporating a resource associated with it. In systems where solutions control their own reproduction, limited resources allow the simple construction of complex landscapes, with an implicit fitness function that changes smoothly to reflect the current solution set. Some of the extensions to this work, discussed below, focus on investigating how flexible limited resource systems can be in terms of building-block tasks and even dynamic addition or removal of rewards.

The role of diversity in encouraging the evolution of optimal solutions to complex problems has long been acknowledged in the field of evolutionary computation (Sareni & Krähenbühl, 1998; Friedrich *et al.*, 2009), and methods of diversity preservation such as fitness sharing (Goldberg & Richardson, 1987) employ negative frequency- or density-dependent mechanisms. On the other hand, I have found no literature addressing how diversity or negative frequency-dependent selection drive the evolution of biological complex features. This comparative lack suggests that other evolutionary computation strategies for solving complex problems may be worth further investigation as pressures driving the evolution of complex traits in nature, returning inspiration to the field of evolutionary biology from which they arose. While evolutionary computation has often borrowed ideas from evolutionary biology, its unique focus on using evolution to solve complex real-world problems has led to unique and valuable insights into the evolution of complex features.

6.2 Future Work

6.2.1 Building blocks: features of lesser complexity

In Chapter 3, I found some evidence that the sensitivity of the evolution of a Boolean task to resource inflow levels increases as the complexity of the task increases. In Chapter 4, I showed that a reward system that eliminates detrimental trade-offs of all tasks for more complex tasks results in more evolution of EQU than simply eliminating the detrimental trade-off for EQU itself. In the study of the evolution of complex tasks, it seems naïve to neglect the explicit study of building-block tasks. At what level of intermediate complexity do these tasks begin to respond to the same pressures that drive the evolution of those we have labeled "complex"? Do entirely different pressures drive the evolution of low-complexity tasks? If so, how do those pressures affect the evolution of high-complexity tasks?

In Avida and in those evolutionary computation systems in which they can be identified, we provide the building block tasks, often at several intermediate levels of complexity. How does the final evolution of a complex feature, or the speed of the evolution of a complex problem, react to different collections of building blocks: for example, only the lowest-complexity tasks, or only tasks just below the target in complexity? In evolutionary computation, we often seek the evolution of multiple complex features which may have (or seem to have) disjoint building blocks. How does the presence of lower complexity tasks unrelated to the target task affect the evolution of the target task? Can building blocks be included and excluded dynamically? In Avida we could implement this by explicitly setting a task's value to 0 on the fly, or by stopping inflow to a linked resource pool.

6.2.2 Competitions between evolved and assembled communities

In Chapter 5, I have only examined competitions between communities constructed with the same implicit or explicit criteria. Are competition results still predictable, either by community-level or ecotype-level descriptors, when the competitor communities differ in construction criteria? This test may be of particular interest in competitions between evolved and assembled communities. Will evolved communities be inherently better because the individuals evolved together? They certainly behave differently than assembled communities: community generality seems to play a different role in each. Within-type, evolved communities composed of specialists do well, whereas in assembled communities, those made up of generalists have an advantage. What is the role of community generality / specialization when the two types compete? Second, the interactions between evolved and assembled communities have application to several fields: not only to ecology, but also to computer security where artificial immune systems face viruses and other malware.

6.2.3 Ecotype-level community descriptors

In Chapter 5, I established that whole-community properties can be used to construct linear models that account for around 35% of the variation in community success in competition. Considering the high consistency of the results of these competitions, especially for evolved communities, it seems likely that more of the variation is explainable. The next obvious set of predictor candidates are those that deal with individual ecotypes in the community: are some simply better than others, no matter the circumstance? If so, can we detect them without testing them in competition? Perhaps they share a common set of tasks. Perhaps they might be particularly conservative with task execution, not wasting instruction cycles performing more tasks than they will be rewarded for.

Or are certain types of community organization more successful than others? Even in this simple nine-resource environment — or ten-resource, since space is the fundamental resource in Avida — it seems unlikely there is one ideal Platonic community, but are there certain community compositions that perform better than others? Does evolution tend to reliably produce such communities? Can we reliably construct them? Of particular interest and application in the natural world, especially the management of invasive species, can we reliably strengthen or weaken communities with artificial additions or removals?

APPENDICES

Appendix A

Instruction Set

A.1 No-ops

`nop-A` The no-operation instructions do nothing when executed. However, they can modify the behavior of other instructions immediately preceeding them, or act as labels in the genome. When modifying other instructions, nop-A stands for register AX or the instruction pointer, nop-B stands for register BX or the read head, and nop-C stands for register CX or the write head. Note that only one no-op can modify each instruction.

`nop-B`

`nop-C`

A.2 Flow control

`if-n-equ` Compares register ?BX? with its complement; if they are not equal, the next instruction (after any modifying no-op instruction) is executed. If they are equal, the next instruction is skipped.

`if-less` Compares register ?BX? with its complement; if ?BX? is the lesser value, the next instruction is executed. If not, the next instruction is skipped.

`if-label` Reads the no-op sequence following it, interpreting it as a label, and tests if the complement label was the most recently copied sequence of instructions. If so, it executes the instruction following the label; if not, it skips that instruction. (This is commonly used by organisms in determining whether they are done producing an offspring copy.)

`mov-head` Jumps the ?IP? to the memory location of the flow head.

`jmp-head` Reads the CX register and jumps the ?IP? forward in the organisms genome by that amount.

`get-head` Copies the memory location of the ?IP? into the CX register.

`set-flow` Moves the flow head to the memory location in the ?CX? register.

A.3 Single argument math

`shift-r` Shifts the bits in the ?BX? register right by one.

`shift-l` Shifts the bits in the ?BX? register left by one, shifting in a 0 and truncating any bits beyond the 32-bit capacity of the register.

`inc` Increments the contents of the ?BX? register by one.

`dec` Decrements the contents of the ?BX? register by one.

`push` Places the contents of the ?BX? register at the top of the active stack, leaving the ?BX? register unchanged.

`pop` Removes the top element from the active stack and places it into the ?BX? register.

`swap-stk` Toggles which stack is active. Other instructions act only on the active stack.

`swap` Swaps the contents of the ?BX? register and its complement register.

A.4 Double argument math

`add` Adds the contents of the BX and CX registers, placing the result in the ?BX? register.

`sub` Subtracts the contents of the CX register from the contents of the BX register, placing the result in the ?BX? register.

`nand`

A.5 Biological operations

`h-copy` Reads the organisms memory at the position of the read head and copies it to the position of the write head. If copy mutations are turned on, a random (chosen from the full set of instructions with equal possibility) instruction may be placed at the write head instead.

`h-alloc` Allocates additional memory for the organism, up to the maximum it is allowed to use for its offspring.

`h-divide` Divides the offspring from the parent. The parent keeps its memory up to the location of the read head; the offsprings memory is initialized to everything between the read head and the write head.

A.6 I/O and Sensory operations

`IO` Outputs the contents of the ?BX? register, checking for any tasks that have been performed, then places a new input into the same ?BX?.

`h-search` Reads the no-op sequence following it, interpreting it as a label, and finds the location of a complement label in the organisms memory. The flow head is moved to the location of the complement label, the BX register is set to the distance between the instruction pointer and the complements location, and the CX register is set to the size of the label. If there is not a no-op sequence, the flow head is moved to the instruction following h-search, and the BX and CX registers are set to 0.

A.7 Definitions

complement The complement of nop-A is nop-B, of nop-B is nop-C, and of nop-C is nop-A.

?BX? If not followed by no-op, this instruction acts on register BX by default. If followed by a no-op, `nop-A` indicates register AX, `nop-B` indicates register BX, and `nop-C` indicates register CX.

?IP? If not followed by a no-op, this instruction acts on the Instruction Pointer by default. If followed by a no-op, `nop-A` directs the instruction to act on the instruction pointer, `nop-B` indicates the read head, and `nop-C` indicates the write head.

Appendix B

Boolean Logic

All Boolean logic functions can be written in terms of the Boolean NAND function. This Appendix presents the Boolean logic functions used in the chapters above, their logical formulae, and their English meaning. Truth tables and circuit diagrams are also included: the circuit diagrams show both the function's representation using a full range of logic gates and implemented with only NAND logic gates.

A NAND B is shorthand for $\neg(A \wedge B)$, a logical statement which reads "It is not true that both A and B are true." That is, in plain English, if A NAND B is true, either A is not true, or B is not true, or both are not true. Table B.1 gives the truth table for NAND when A and B are each one bit; that is, either 0 (false) or 1 (true). Figure B.1 shows a representation of a NAND gate as it appears in a circuit diagram.

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

Table B.1: Truth table of binary NAND. This function is false only when both A and B are true.

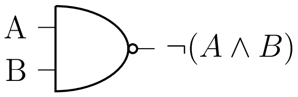

Figure B.1: The NAND logic gate in a circuit diagram.

NOT A is written as $\neg A$ in logical notation, and reads "A is not true." Table B.2 gives the simple truth table for NOT.

A	NOT A
0	1
1	0

Table B.2: Truth table of unary NOT. This function is false when the input A is true.

Implementing NOT in terms of NAND is simple once we observe that $A \wedge A$ is equivalent to A, and therefore $\neg A$ is equivalent to $\neg(A \wedge A)$. This expression simply represents a NAND operation where the two inputs are the same.

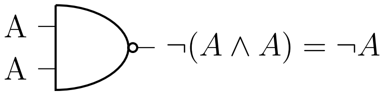

Figure B.2: The NOT function implemented with NAND in a circuit diagram. Feeding the input A into both inputs of a NAND gate produces $\neg A$.

A AND B is written as $A \wedge B$ in logical notation, and reads "Both A and B are true." Table B.3 gives the truth table for AND, and Figure B.3 shows the symbol for an AND logic gate and its implementation using two NAND gates.

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Table B.3: Truth table of binary AND. This function is true only when both A and B are true.

To implement AND in terms of NAND, observe that we know how to implement both NAND and NOT in terms of NAND. We simply compute NAND and use another NAND to negate the result, producing AND using two NAND operations.

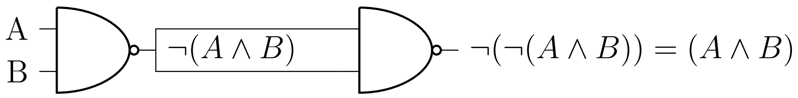

Figure B.3: The AND function implemented with NAND gates in a circuit diagram. AND is implemented with two NAND gates.

A ORN *B* is shorthand for the logical statement $A \vee \neg B$. “ORN” is short for “or not”, and this statement reads “Either A is true or B is false.” Note that by switching the names of A and B, ORN can also mean “Either B is true or A is false.” Table B.4 shows the truth table for ORN: notice that only one of the “A is true” and “B is false” statements must hold for the whole statement to be true.

A	B	A ORN B
0	0	1
0	1	0
1	0	1
1	1	1

Table B.4: Truth table of binary ORN. This function is false only when A is false and B is true.

Even though ORN is phrased in terms of “or” rather than “and”, it can be implemented with NAND operations. First, apply De Morgan’s law: $A \vee \neg B$ is equivalent to $\neg(\neg A \wedge B)$. Observe that this is simply the NAND function, but with one input negated. It takes one NAND to implement negation, so the total number of NAND operations is two. The circuit diagram of this implementation is given in Figure B.4.

A OR *B*, in logical notation, is $A \vee B$. It reads “Either A is true, or B is true, or both of them are true.” Table B.5 shows the truth table for OR.

In order to express OR in terms of NAND, we apply De Morgan’s law: $A \vee B$ is equivalent to $\neg(\neg A \wedge \neg B)$, that is, the NAND function with both inputs negated. This three-NAND implementation is presented in Figure B.5.

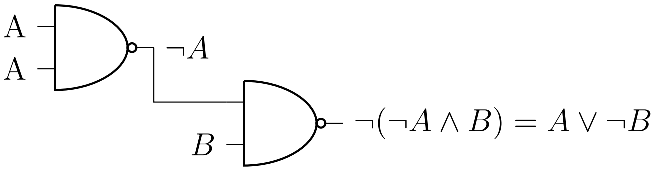

Figure B.4: The ORN function implemented with NAND gates in a circuit diagram. We apply De Morgan's law to see that the computed result, $\neg(\neg A \wedge B)$, is equivalent to $A \vee \neg B$, the ORN function.

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Table B.5: Truth table of binary OR. This function is false only when both A and B are false.

A ANDN B is shorthand for the logical expression $A \wedge \neg B$, which reads "A is true and B is false." The truth table for ANDN is shown in Table B.6.

A	B	A ANDN B
0	0	0
0	1	0
1	0	1
1	1	0

Table B.6: Truth table of binary ANDN. This function is true only when A is true and B is false.

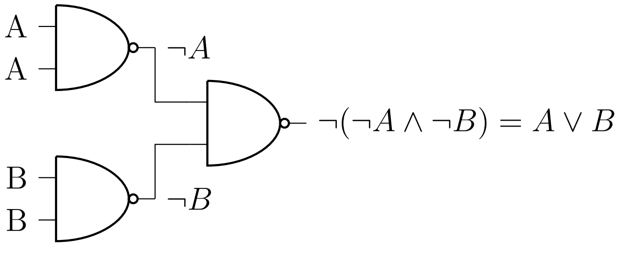

Figure B.5: The OR function implemented with NAND gates in a circuit diagram. We apply De Morgan's law to see that the computed result, $\neg(\neg A \wedge \neg B)$, is equivalent to $A \vee B$, the OR function.

ANDN is simply the AND function with one of its inputs negated. We implement it in terms of NAND by using one NAND for the input negation and two to implement AND, for a total of three NAND operations. The circuit diagram laying out this implementation is presented in Figure B.6.

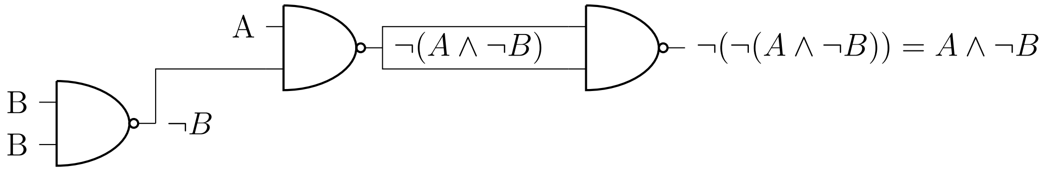

Figure B.6: The ANDN function implemented with NAND gates in a circuit diagram. Implementing ANDN requires three NAND gates.

A NOR B is written $\neg(A \vee B)$ in logical notation, and reads "Neither A nor B is true," that is, "Both A and B are false." The truth table for NOR is shown in Table B.7.

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

Table B.7: Truth table of binary NOR. This function is true only when both A and B are false.

To implement NOR in terms of NAND, we use the already existing implementations for OR and NOT. Three NAND operations are used to implement OR, and a final NAND is used to negate the output, producing NOR by using four NAND operations.

A XOR B is written $A \oplus B$ in logical notation, which reads "Either A is true or B is true, but *not* both," or equivalently, "A and B are different." Table B.8 gives the truth table for XOR.

XOR can more more awkwardly phrased "Either A is true and B is false, or A is false and B is true." It is impossible for both statements to be true at the same time. If we

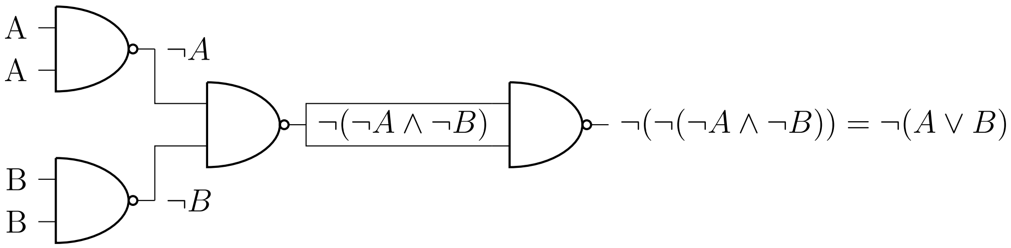

Figure B.7: The NOR function implemented with NAND gates in a circuit diagram. Implementing NOR requires four NAND gates; we apply De Morgan's law to observe that the output, which simplifies to $\neg A \wedge \neg B$, is equivalent to $\neg(A \vee B)$, the NOR function.

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Table B.8: Truth table of binary XOR. This function is true only when A and B are different.

expand this statement with De Morgan's law, a naïve implementation requires five NAND operations. However, XOR requires only four NAND operations, and this is easy to show with a little symbolic manipulation:

$$A \oplus B$$

$$(A \wedge \neg B) \vee (\neg A \wedge B)$$

$$(A \wedge \neg A) \vee (A \wedge \neg B) \vee (\neg A \wedge B) \vee (B \wedge \neg B)$$

$$(A \wedge (\neg A \vee \neg B)) \vee ((\neg A \vee \neg B) \wedge B)$$

$$(A \wedge \neg(A \wedge B)) \vee (\neg(A \wedge B) \wedge B)$$

$$\neg(\neg(A \wedge \neg(A \wedge B)) \wedge \neg(\neg(A \wedge B) \wedge B))$$

$$\neg(\neg(A \wedge X) \wedge \neg(X \wedge B))$$

Notice that $\neg(A \wedge B)$, shortened to X, is repeated, but we will only have to calculate it once. That uses one NAND operation. The equation in X is a simple NAND of two smaller NAND operations: in total, four NAND operations.

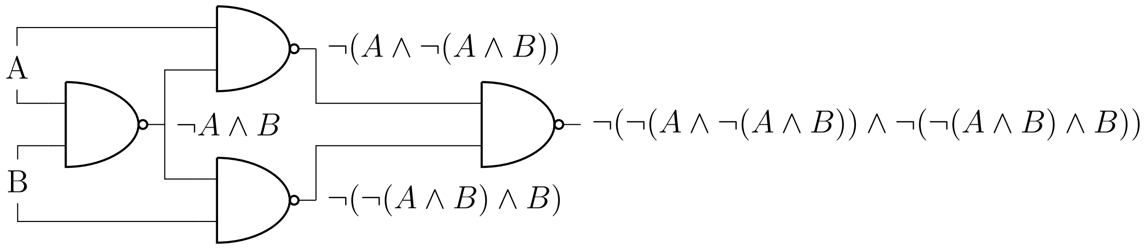
A
B
$\neg A \wedge B$
$\neg(A \wedge \neg(A \wedge B))$
$\neg(\neg(A \wedge B) \wedge B)$
$\neg(\neg(A \wedge \neg(A \wedge B)) \wedge \neg(\neg(A \wedge B) \wedge B))$

Figure B.8: The XOR function implemented with NAND gates in a circuit diagram. Implementing XOR requires four NAND gates; we simplify the result by following the derivation backward to obtain $(A \wedge \neg B) \vee (\neg A \wedge B)$, the XOR function.

A EQU B is read quite simply as "A is equal to B," or "A and B are the same." The simple truth table is presented in Table B.9. This concept of equality or matching, while quite simple for us, requires at least five NAND operations in a computer.

A	B	A EQU B
0	0	1
0	1	0
1	0	0
1	1	1

Table B.9: Truth table of binary EQU, sometimes known as XNOR. This function is true only when A and B are the same.

However, implementing EQU in terms of NAND is simple if we consider the functions already implemented: EQU is simply the negation of XOR, requiring one extra NAND operation for a total of five NANDs. This implementation appears in Figure B.9. Equivalently, EQU can be expressed as "Both A and B are true, or both A and B are false." Because we discount the possibility that A and B can be both true and false at the same time, it is redundant to add "but not both." Since EQU is now expressed in terms of both "and" and "or", we employ De Morgan's law to express it only in terms of "and": $(A \wedge B) \vee (\neg A \wedge \neg B)$ is equivalent to $\neg(\neg(A \wedge B) \wedge \neg(\neg A \wedge \neg B))$. This implementation also requires five NAND operations.

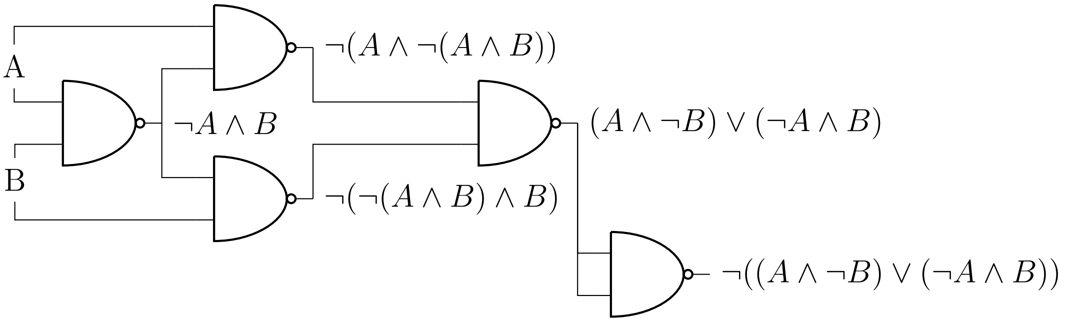
A
B
$\neg(A \wedge \neg(A \wedge B))$
$\neg A \wedge B$
$\neg(\neg(A \wedge B) \wedge B)$
$(A \wedge \neg B) \vee (\neg A \wedge B)$
$\neg((A \wedge \neg B) \vee (\neg A \wedge B))$

Figure B.9: The EQU or XNOR function implemented with NAND gates in a circuit diagram. Implementing EQU requires five NAND gates; the result simplifies to $(\neg A \wedge \neg B) \vee (A \wedge B)$, the EQU function.

Appendix C

How two deleterious EQU-granting mutations fixed on the line of descent

As shown in Figure 4.1, two of the EQU-achieving populations evolved in the unlimited-resource environment of Chapters 3 and 4 fixed a *detrimental* EQU-granting mutation on the line of descent. (These are runs 1089 and 1191.) While unusual, this is not out of the question. I examined these mutations in detail, and determined why these detrimental mutations were able to fix.

First, both mutations were nearly neutral. The fitness ratio of the detrimental mutation fixed in population 1089 was 0.997674, and that of population 1191 was 0.999416. In both cases, the net task trade-off effect on the merit multiplier was 0 (Table C.1, Table C.2). Presumably the mutation also increased gestation time slightly. Second, the next mutation on each lineage is super-compensatory. Population 1191's compensatory mutation doubled its fitness; population 1089,s resulted in a more-than-15-fold increase in fitness!

Pop. 1089	NOT	NAND	AND	ORN	OR	ANDN	NOR	XOR	EQU
pre-EQU	✓	✓	✓	✓	✓	✓	✓		
EQU		✓	✓	✓	✓	✓			✓
post-EQU		✓	✓	✓	✓	✓	✓		✓

Table C.1: Task performance snapshot of Population 1089 before, at, and after the detrimental fixation of EQU. The EQU-granting mutation trades a 2^1 task (NOT) and a 2^4 task (NOR) for the 2^5 EQU task; the task trade-off net gain is 0. The next mutation on the lineage regains the 2^4 NOR task.

Pop. 1191	NOT	NAND	AND	ORN	OR	ANDN	NOR	XOR	EQU
pre-EQU			✓	✓	✓		✓		
EQU					✓	✓			✓
post-EQU					✓		✓		✓

Table C.2: Task performance snapshot of Population 1191 before, at, and after the detrimental fixation of EQU. The EQU-granting mutation trades both 2^2 tasks (AND, ORN) and a 2^4 task (NOR) for the 2^5 EQU task and a 2^3 task (ANDN); the task trade-off net gain is 0. The next mutation on the lineage trades the 2^3 ANDN task for the 2^4 NOR task.

BIBLIOGRAPHY

BIBLIOGRAPHY

Peter A. Abrams, Claus Rueffler and Gary Kim. 2008. Determinants of the strength of disruptive and/or divergent selection arising from resource competition. *Evolution*, **62**, 1571–1586.

Christoph Adami, Charles Ofria and Travis C. Collier. 2000. Evolution of biological complexity. *Proceedings of the National Academy of Sciences of the United States of America*, **97**, 4463.

Michael C. Appleby. May 1983. The probability of linearity in hierarchies. *Animal Behaviour*, **31**, 600–608.

Ben Bolker and R Development Core Team. *bbmle: Tools for general maximum likelihood estimation*, 2012. URL `http://CRAN.R-project.org/package=bbmle`. R package version 1.0.5.2.

Bradley J. Cardinale, Helmut Hillebrand, W. S. Harpole, Kevin Gross and Robert Ptacnik. 2009. Separating the influence of resource 'availability' from resource 'imbalance' on productivity–diversity relationships. *Ecology Letters*, **12**, 475–487.

Ted J. Case. December 1990. Invasion resistance arises in strongly interacting species-rich model competition communities. *Proceedings of the National Academy of Sciences*, **87**, 9610–9614.

Paul Chesson. 2000. Mechanisms of maintenance of species diversity. *Annual Review of Ecology and Systematics*, **31**, 343–366.

Stephanie S. Chow, Wilke O. Claus, Charles Ofria, Richard E. Lenski and Christoph Adami. July 2004. Adaptive radiation from resource competition in digital organisms. *Science*, **305**, 84–86.

Tim F. Cooper and Charles Ofria. Evolution of stable ecosystems in populations of digital organisms. In *Proceedings of VIII International Conference on Artificial Life*, pages 227–232, 2002.

Arthur W. Covert III. *On the beneficial effects of deleterious mutations*. PhD thesis, gradworks.umi.com, 2011.

Charles Darwin. *The origin of species by means of natural selection, or, The preservation of favored races in the struggle for life.* John Murray, London, 6th edition, 1872.

Ulf Dieckmann and Michael Doebeli. July 1999. On the origin of species by sympatric speciation. *Nature*, **400**, 354–357.

Damian Eads, 2008. URL http://scipy-cluster.googlecode.com/. hcluster: Hierarchical Clustering for SciPy.

Santiago F. Elena, Claus O. Wilke, Charles Ofria and Richard E. Lenski. March 2007. Effects of population size and mutation rate on the evolution of mutational robustness. *Evolution*, **61**, 666–674.

Timothy J. Flowers, Hanaa K. Galal and Lindell Bromham. 2010. Evolution of halophytes: multiple origins of salt tolerance in land plants. *Functional Plant Biology.*

Tobias Friedrich, Pietro S. Oliveto, Dirk Sudholt and Carsten Witt. 2009. Analysis of diversity-preserving mechanisms for global exploration. *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, **17**, 455–476.

Maren L Friesen, Gerda Saxer, Michael Travisano and Michael Doebeli. 2004. Experimental evidence for sympatric ecological diversification due to frequency-dependent competition in Escherichia coli. *Evolution.*

Douglas J. Futuyma and Gabriel Moreno. 1988. The evolution of ecological specialization. *Annual Review of Ecology and Systematics*, **19**, 207–233.

Sergey Gavrilets and Aaron Vose. December 2005. Dynamic patterns of adaptive radiation. *Proceedings of the National Academy of Sciences of the United States of America*, **102**, 18040–18045.

Sherri Goings. *Natural niching: Applying ecological principles to evolutionary computation.* PhD thesis, Michigan State University, September 2010.

Sherri Goings and Charles Ofria. 2009. Ecological approaches to diversity maintenance in evolutionary algorithms. *Artificial Life, 2009. ALife '09. IEEE Symposium on*, pages 124–130.

David E. Goldberg and Jon Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms and Their Application*, pages 41–49. L. Erlbaum Associates Inc., October 1987.

Heather J. Goldsby. *The Evolution of Division of Labor in Digital Organisms*. PhD thesis, Michigan State University, Michigan State University, February 2011.

Lee Graham, Franz Oppacher and Steffen Christensen. September 2007. Irreducible complexity in a genetic algorithm. *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, pages 3692–3697.

T. Ryan Gregory. 2008. The evolution of complex organs. *Evolution: Education and Outreach*, **1**, 358–389.

Alex R. Hall and N Colegrave. January 2007. How does resource supply affect evolutionary diversification? *Proceedings of the Royal Society B: Biological Sciences*, **274**, 73–78.

Rees Kassen, Angus Buckling, Graham Bell and Paul B. Rainey. August 2000. Diversity peaks at intermediate productivity in a laboratory microcosm. *Nature*, **406**, 508–512.

M. G. Kendall and B. Babington Smith. 1940. On the method of paired comparisons. *Biometrika*, **31**, 324.

Theodore A. Kennedy, Shahid Naeem, Katherine M. Howe, Johannes M. H. Knops, David Tilman and Peter Reich. June 2002. Biodiversity as a barrier to ecological invasion. *Nature*, **417**, 636–638.

Richard E. Lenski, Jeffrey E. Barrick and Charles Ofria. 2006. Balancing robustness and evolvability. *PLoS Biol*, **4**, e428.

Richard E. Lenski, Charles Ofria, Travis C. Collier and Christoph Adami. 1999. Genome complexity, robustness and genetic interactions in digital organisms. *Nature*, **400**, 661–664.

Richard E. Lenski, Charles Ofria, Robert T. Pennock and Christoph Adami. May 2003. The evolutionary origin of complex features. *Nature*, **423**, 139–144.

John Maron and Marilyn Marler. 2007. Native plant diversity resists invasion at both low and high resource levels. *Ecology*, **88**, 2651–2661.

Enrique Melendez-Hevia, Thomas G. Waddell and Marta Cascante. 1996. The puzzle of the Krebs citric acid cycle: Assembling the pieces of chemically feasible reactions, and opportunism in the design of metabolic pathways during evolution. *Journal of Molecular Evolution*, pages 293–303.

Dusan Misevic, Charles Ofria and Richard E. Lenski. February 2006. Sexual reproduction reshapes the genetic architecture of digital organisms. *Proceedings. Biological sciences / The Royal Society*, **273**, 457–464.

St. George Mivart. *On the Genesis of Species.* Book on Demand, January 1871.

Dan-Eric Nilsson and Susanne Pelger. April 1994. A pessimistic estimate of the time required for an eye to evolve. *Proceedings of the Royal Society B: Biological Sciences*, **256**, 53–58.

Charles Ofria, David M. Bryson and Claus O. Wilke. Avida: A Software Platform for Research in Computational Evolutionary Biology. In Maciej Komosinski and Andrew Adamatzky, editors, *Artificial Life Models in Software*, pages 3–35. Springer London, London, 2009.

Charles Ofria, Wei Huang and Eric Torng. 2008. On the gradual evolution of complexity and the sudden emergence of complex features. *Artificial Life.*

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2012. URL `http://www.R-project.org/`. ISBN 3-900051-07-0.

D. A. Roff and D. J. Fairbairn. 2007. The evolution of trade?offs: where are we? *Journal of Evolutionary Biology*, **20**, 433–447.

Rowan F. Sage. 2004. The evolution of C4 photosynthesis. *New Phytologist.*

Bruno Sareni and Laurent Krähenbühl. September 1998. Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, **2**, 97–106.

Petr Savicky. *pspearman: Spearman's rank correlation test*, 2009. URL `http://CRAN.R-project.org/package=pspearman`. R package version 0.2-5.

Dolph Schluter. November 1996. Ecological causes of adaptive radiation. *The American Naturalist*, **148**, S40–S64.

David Tilman. *Resource Competition and Community Structure.* Princeton University Press, Princeton, NJ, 1982.

David Tilman. July 2004. Niche tradeoffs, neutrality, and community structure: a stochastic theory of resource competition, invasion, and community assembly. *Proceedings of the National Academy of Sciences of the United States of America*, **101**, 10854–10861.

Bess L. Walker and Charles Ofria. Evolutionary Potential is Maximized at Intermediate Diversity Levels. In *International Conference on the Simulation and Synthesis of Living Systems*, pages 116–120. MIT Press, July 2012.

Daniel B. Weissman, Michael M. Desai, Daniel S. Fisher and Marcus W. Feldman. 2009. The rate at which asexual populations cross fitness valleys. *Theoretical Population Biology*, **75**, 286–300.

Claus O. Wilke, Jia Lan Wang, Charles Ofria, Richard E. Lenski and Christoph Adami. 2001. Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, **412**, 331–333.

Erika S. Zavaleta and Kristin B. Hulvey. November 2004. Realistic species losses disproportionately reduce grassland resistance to biological invaders. *Science*, **306**, 1175–1177.