ON THE EVOLUTION OF MUTATION BIAS IN DIGITAL ORGANISMS

By

Matthew Rupp

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

2011

ABSTRACT

ON THE EVOLUTION OF MUTATION BIAS IN DIGITAL ORGANISMS

By

Matthew Rupp

Mutation is one of the primary drivers of genetic change. In this work I study mutation biases, which are sets of different genetic-state inflow probabilities. Mutation biases have the potential to change the composition of genomes over time, leading to divergent short- and long-term evolutionary outcomes. I use digital organisms, self-replicating computer programs, to explore whether or not mutation biases are capable of altering the long-term adaptive behavior of populations; whether mutation biases can be competitive traits; and whether mutation biases can evolve.

I find that mutation biases can alter the long-term adaptive behavior of mutation bias-obligate populations in terms of both mean fitness and complex trait evolution. I also find that mutation biases can compete against one another under a variety of conditions, meaning mutation bias can selectable over relatively-short periods of time. The competitive success of a mutation bias does not always depend upon the presence of beneficial mutations, implicating an increase in the probability of neutral mutations as a sufficient mechanism for bias selection. Finally, I demonstrate that by giving organisms a mutable mutation bias allele, populations preferentially evolve to possess specific biases over others.

Overall, this work shows that mutation bias can act as a selectable trait, influencing the evolution of populations with regard to both their internal-genetic and external environments.

I dedicate this work to my friends, family, and the individuals of Michigan State University, past and present, who continue to inspire me.

ACKNOWLEDGMENTS

I would like to thank my two co-advisors: Dr. Eric Torng and Dr. Charles Ofria who brought me into the Digital Evolution (DEVO) Lab as an undergraduate and contributed greatly to my academic upbringing. Throughout the years, they have not only provided me with sound advice but also served as role models.

I thank the two other members of my committee, Dr. Thomas M. Schmidt and Dr. William F. Punch. Dr. Schmidt introduced me to the world of microbiology and was a major reason I straddle the lines of computer computer science and biology. He provided insightful feedback for the experiments in this work and helpful suggestions on how to communicate my efforts to a wider audience. Dr. Punch provided needed perspective on the computer science applications of this work as well as guidance on some of the experiment configurations.

The advice and conversations of past and present members of the DEVO Lab contributed greatly to my understanding of this material. The now graduated members of DEVO Lab: Dr. Arthur Covert, Dr. Sherri Goings, Dr. Jeff Clune, Dr. Ben Beckman, Dr. Laura Grabowski, Dr. Dusan Misevic, Dr. Gabriel Yedid, Dr. Wei Huang, Dr. Dehua Huang, and Mr. Jason Stredwick provided me with great conversation and feedback on my work over the years. I also am grateful to all those present members of the lab (and affiliates) who continue to provide me with insight especially: Drs. Heather Goldsby and David Knoester; Rosangela Canino-Koning, David Bryson, Mairin Chesney, Brian Connelly, Anu Pakanati, Mike Wiser, Bess Walker, and Luis Zaman. Lastly, I would like to especially thank Brian Baer for all of the work he's done to keep the lab's research infrastructure in shape over the years.

My parents Kathy and Kirk Rupp and grandparents Carol and Richard Whelton have supported me greatly through out the years. I give them my heartfelt thanks.

Jennie Gies, Heather Rickets, Lisa Asmus, Emily Barry, and F. Amelia Howes have my

lasting gratitude. Their supportive friendship kept me motivated, grounded, and entertained.

Finally, I especially thank Jason Moon. His love and support were invaluable during this endeavor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

An Introduction to Mutation Biases

Mutation is one of the primary drivers of genetic change. Along with recombination and gene transfer, it creates genetic variation that selection can act upon. As the variation in genomes affects how species adapt to their environment, an understanding of the effects of mutation is crucial in understanding how organisms evolve.

1.1 What is a mutation bias?

I define a mutation bias to be a distribution of inflow probabilities for each possible genetic state (a nucleotide, amino acid, or Avidian instruction) per mutation event. In this work, I focus on two mutation events, insertions and substitutions. During each of these events, a specific site in the genome (a new site for an insertion or an existing site for a substitution) adopts a particular state with a fixed, unconditional probability. The set of probabilities for all possible states is a mutation bias.

This is a simple model for mutation rates with a few important features that I want to highlight. First, for substitution events, I assume that the probability for each new state is independent of the original state. Second, I do not include any mechanism to repair mutations. Third, in my substitution model, it is possible for the resulting state of a substituted site to be the same as before the mutation event. Finally, I have deliberately separated mutation rate from mutation bias. The rate that mutations occur is independent of the probability distribution for the new state. In nature, mutation bias, mutation rate, and how individual nucleotides mutate to one another are inherently tied together. However, this simple model allows me to isolate and clearly illustrate the effects of mutation bias.

1.2 Is mutation bias an adaptation for evolvability?

Because mutation can produce selectable variation, mutation bias may be a selectable trait that confers evolvability, the ability to leverage or mitigate the effects of genetic change. This dissertation examines whether mutation bias meets the criteria to be an adaptation for evolvability.

In order for mutation bias to serve as an adaptation for evolvability, it must be heritable, variable, and selectable. In this dissertation, I make mutation bias a fully heritable trait in my organisms. This is partially supported in the natural world. For example, mutation bias (along with rate) is partially heritable by virtue of the DNA replication and repair machinery encoded in any organism's genome. Of course, mutation bias is affected by other elements such as the inherent mutability of the genetic substrate and the particular environment the organism exists in. In this work, I make mutation bias variable by creating a limited number of different mutation biases an organism can have.

This leaves selectability. Determining the selectability of mutation bias is the primary focus of this dissertation. It is not entirely clear that mutation bias can be selectable. Certainly, a particular bias might create beneficial mutants that will allow the bias to hitchhike to fixation in a population; however, the benefit conferred by a mutation bias (e.g. the set of genetic changes) might be so uncoupled from its consequence (the discovery of a novel, adaptive trait that selection can act directly upon) as to have the fate of the mutation bias left to chance. This observation implies that mutation bias is something that can only influence long-term evolution and is not something that competition at the individual level would be able to distinguish.

Selectionist	Neutralist
Amino acid composition is driven by GC content [1, 37, 43].	GC content correlates positively with "ambivalent" amino acids [15].
Protein-coding regions are positively correlated with GC content [27].	Third codon sites are most correlated with genomic GC [27].
GC content might help with DNA thermostability in hostile environments [3].	Only structural RNA GC content is correlated with temperature [17].
Strong mutation bias can lead to the selection of non-optimal novelty [45].	Mutation is a weak pressure in mutation-selection balance [37, 45].
GC:AT content differences might be due to a competition for resources [33].	GC:AT content differences are the result of neutral processes [44].

Table 1.1. **Selectionist and neutralist hypotheses for GC:AT biases.** Broadly, there are two different ideas about the importance of composition biases: selectionist and neutralist. Selectionists argue that composition biases of genomes serve an adaptive purpose. Neutralists believe that composition differences are the consequence of selectively-neutral processes.

1.3 Previous work on mutation bias

Previous work on mutation bias falls into three primary categories: the observation of compositional biases in organisms, the causes of compositional biases in organisms, and the consequences of compositional biases in organisms. The majority of work focuses on the observation and causes of compositional biases in bacterial chromosomes. The work on consequences of compositional bias covers a broader range of species. Table 1.1 highlights some potential hypotheses of variable GC:AT biases.

Genetic differences of GC:AT content in bacteria were observed as early as the late 1950s. While looking for correlations between DNA and RNA in bacteria, Belozersky and Spirinfound found there to be a variety of GC:AT ratios from the 19 bacterial species they examined, ranging from ratios of 0.45 ± 0.02 to 2.73 ± 0.02 [2]. Other ratios, purine to pyrimidine (AG:CT) and GT:AC showed no major deviation from 1.0 for all examined species, owing to Watson-Crick pairings.

To explain this difference in GC:AT content, two researchers independently theorized that the composition bias is the result of a varying mutation bias operating under equilibrium (selectively-neutral) conditions [13, 39]. There is evidence to support this hypothesis as GC:AT biases are especially noticeable in relatively neutral parts of the genome (e.g. spacers and third-position sites in codons relative to chromosomal regions producing structural RNA [27]). Consequently, mutation bias is potentially important for neutral evolution [40].

There are many causes of mutation bias in nature, due to both inherent DNA instability and the capability of replication and repair enzymes. One of the most cited reasons for a GC:AT bias is the inherent instability of cytosine, which creates an asymmetric pressure away from GC to AT pairs. In its methylated form, cytosine mutates via deamination to thiamine, creating a transition mutation. When it is not methylated, it deaminates into uracil, a correctable state provided appropriate repair enzymes are available. As such, DNA has a natural propensity to mutate toward a higher AT content.

However, repair mechanisms are not always available nor is there a unidirectional preference for higher AT content, as is the case with the *mutT E. coli* strain. *mutT* mutants were created through X-ray induced mutation and discovered to have a three-fold higher mutation rate than the wildtype [41]. The higher mutation rate produced an increase in the number of mutations from AT→GC, increasing the genomic concentration of GC [6]. The *mutT* enzyme, knocked out in *mutT* strains, provides protection from mutation by degrading a potential mutagen, 8-oxodGTP, an oxidized form of the nucleotide guanine [24]. In the absence of the mutT enzyme, as caused for example by an insertion element-induced knockout in the original *mutT* strain [4], 8-oxodoGTP can pair with either cytosine or adenine, potentially leading to a transversion during subsequent replication if not detected and repaired by additional enzymes [10].

There are other potential causes of mutation bias. Because mutation is more likely in single-stranded than double stranded DNA (at least for cytosine and adenine) [12, 23, 19], the location of a nucleotide relative to the origin and terminus of replication [26, 25], the strand the nucleotide occupies [26, 22], and the frequency it is transcribed [11] are all factors in determining both the rate and direction of mutation.

I now consider prior work that studies the consequences of GC:AT biases. Several papers have examined how GC:AT content affects protein encoding. Even before the discovery of codon mapping, researchers observed a correlation between genome GC content and protein amino acid content [38]. Using the often degenerate third position of a codon as a metric for directional mutation pressure between GC and AT nucleotide content, many authors have found that a strong positive correlation between genomes with high third position GC content and the amino acids glycine, alanine, arginine, and sometimes proline (GAR[P]) and negative correlation with phenylalanine, tyrosine, isoleucine, lysine, and sometimes methionine (FY[M]INK) [38, 18, 15, 35, 1]. Providing more direct evidence for a causal role for mutation, research has found a strong substitution asymmetry favoring GARP to FYMINK in GC-rich and GC-poor homologs between two divergent species [43]. The extent to which changes in amino acids are driven by mutation bias under neutral evolution [35, 43, 15] or selection for protein function [1] is not completely known.

There have been suggestions that GC content is directly selected and not just the byproduct of mutation bias. One possible reason for GC content being directly selected is that GC rich regions of chromosomes tend to be more thermally stable than AT rich regions. This is because a GC pair has three hydrogen bonds where as AT only has two [42]. This hypothesis is supported by the observation that “warm-blooded” vertebrates tend to have more GC-rich regions (isochores) than “cold-blooded” vertebrates [3]. There is also a link between optimal growth temperatures for eubacteria and archaea with the GC content of their structural RNAs (but not the GC content of their genomes) [17]. Alternately, there is a hypothesis for direct selection of higher AT content genomes to reduce the energy needed to replicate [33].

Finally, there has been little direct evidence from previous work that mutation bias can cause change in evolutionary outcomes. There is no work from natural systems because of the difficulty in manipulating mutation bias in natural systems. The only previous work I am aware of are two relatively simple computational model studies. In the first study, with an extremely simple model, the authors show that extraordinarily biased mutation rates can lead to non-optimal selective outcomes in small populations [45]. Several years later, one of the authors further explored this topic using an NK model [37][1]. He showed that a model with an alphabet of four "nucleotide" states and a translation step to map codons into "amino acids" could lead to an evolved bias in amino acid composition. However, it is important to note that the only meaning for nucleotides and amino acids is captured by the codon encoding and associated random fitness values. In particular, the result seems to follow directly from the codon encoding used in the study.

1.4 Can mutation biases confer evolvability?

In order for mutation bias to have an evolutionary affect, it must be shown that applying a mutation bias can change the outcome of evolution in obligate populations. For it to be evolvable, it must both be something that selection can distinguish and be itself heritable and mutable.

As discussed above, there is evidence to suggest that mutation bias in nature fits these criteria. Mutation bias has been implicated in altering protein composition. It has also been shown to be modulated (along with mutation rate) by the presence of different replication and repair enzymes and cytosine methylation. Consequently, mutation bias in nature can be derived in part from the product of genes which are themselves evolving. What is missing from the studies discussed above is direct evidence that mutation bias, in a non-trivial system, can meet these three criteria.

In this body of work, I directly test whether or not mutation bias can be a selectable trait for evolvability. To do so, I show that applying a mutation bias can lead to different rates

[1] In an NK model, fitness is determined from the combined contribution of N sites with each site having K interactions with other sites [20].

and scope of environmental adaptation. I also show that the effects of different mutation bias can be distinguished by selection. Finally, I show that mutation bias, given a mutable, heritable representation, can be consistently selected by individuals in a population.

CHAPTER 2

Avida: A Digital Life Model

Studying evolution under experimental conditions is a challenge. The timescale over which evolution occurs is typically orders of magnitude more than the length of a graduate program. Consider an extreme example: suppose one wants to examine how a herd of elephants evolves some novel adaptation to an environmental change. Elephants are not capable of reproduction until they reach about thirteen years of age, and their gestation time is 22 months. Considering it generally takes many generations for a phenotype to fix in a population, using elephants to study novel trait evolution is clearly infeasible. To test any evolvability claim would require finding (or producing) a reliable control herd that lives without the environmental change doubling these problems, albeit in parallel; even then the statistical power from such an experiment would be quite low.

It is for these reason researchers choose to study evolution in laboratory model systems with much shorter gestation times: fruit flies, microorganisms, and the like. However these systems are also fraught with their own idiosyncratic problems. Organic systems require careful maintenance of their support equipment, routine checks for contamination, and multiple physical resources to test experimental hypotheses. Added all up, a single set of evolution experiments could easily create many years worth of work in up-keep alone, to say nothing of the time and resources it takes to collect and analyze data. The dividends of such experiments may well be worth the costs; but broad examinations of evolutionary theories over multiple dimensions become cumbersome. Digital life helps mitigate these problems by providing a more tractable and less resource-intensive platform to explore a wide variety of evolutionary hypotheses.

Digital life is largely defined by the self-replicating computer program. Such programs contain machine code capable of instructing the hardware they are instantiated upon to replicate. They may also be evolved to solve problems by introducing mutations and providing fitness benefits for demonstrating good solutions. A number of different software packages enable such programs to "live" inside a computer. Perhaps the most famous early digital life package is Tierra [32]. Organisms in Tierra are machine code programs that share a common memory space "soup". Tierra itself acts like an operating system, allocating CPU time for each organism to replicate and "reaping" old organisms to keep space available for offspring. Digital organisms in the simplest Tierra environment evolved to optimize their replication algorithm and to exploit other organisms to reproduce faster.

For my experiments, I chose to use the digital evolution platform Avida [29]. Avida follows in the footsteps of Tierra. Unlike Tierra's "soup", Avida enforces a strict separation of each organism's memory space. This helps keep asexual lineages identifiable and fosters an easier understanding of the causes of adaptive change. This chapter will be devoted to a description of the terms used throughout this work, Avida, and the methods used to implement the experiments in the following chapters.

2.1 Avida

Avida is a software package that allows researchers to explore evolution through experimentation using digital organisms. Digital organisms in Avida experience competition for limited resources, inheritance, and genetic mutation; therefore they are capable of evolving adaptations to their environment. Avida allows the researcher to tightly control environmental conditions and easily observe resulting population responses, allowing for hypotheses that are difficult to study in natural systems to be readily explored.

An Avida experiment can be thought of as a digital analogue of a microbial experiment. An Avidian is a digital organism. It is comprised of two parts: a sequence of virtual CPU instructions (its genome) and a virtual CPU (its biological equipment). Everything outside of an organism is its environment. The organism's environment provides an essential metabolic resource called SIPs (Single Instruction Processing units [21]), which give the organisms the ability to process their genome. How many SIPs an organism receives depends upon how it interacts with the environment to produce products or behaviors established by the researcher. The composite of these products and behaviors defines an organism's phenotype.

The genome is the heritable substrate in Avida. It is comprised of 26 different virtual CPU instructions. Like a bacterial chromosome, it is circular, so that the first instruction in the genome follows the last. These instructions are capable of changing the internal state of the virtual CPU and are Turing-complete. Roughly, these instructions are split into four different classes: no-operation/behavior modification, genome-processing control, reproduction, and numerical operation. Appendix A provides a detailed description of the function of each of these twenty-six instructions.

The concepts of gene and locus in Avida are less clear than in microorganisms. To avoid ambiguity, I will use the word site when discussing a particular location in an Avidian genome. The length of a genome is simply be the number of instructions it contains.

2.1.1 Avidians: genomes and life-cycles

Figure 2.1 depicts an Avidian genome and how it is processed in one generation. The virtual CPU (not shown) that processes the genome contains three different registers, two stacks (currently capped at ten elements), four specialty pointers that identify different locations in the genome, and a special purpose register that traces the most recent sequence of `nop` instructions read during replication.

Execution of an organism begins with the virtual CPU being reset so that all four special-purpose heads (the instruction pointer, the read, write, and flow heads) point to the first site in the genome, labeled with a star in Figure 2.1. All registers are set to the value zero and the stacks are empty. When the organism receives a SIP, a quantum of metabolic energy in Avida, it processes the instruction at the site pointed to by the instruction pointer. Depending on the identity of the instruction, mathematical operations may be carried out, the environment may provide input or receive output, or the state of the virtual CPU's stacks, registers, or heads may be altered. Unless the instruction explicitly alters the state of the instruction pointer, the instruction pointer moves to the next site after processing. When the instruction pointer reaches the end of the organism's genome, it is moved to the start. This makes an Avidian genome circular.

The organism depicted in Figure 2.1 is the ancestor used to seed many of the experiments in this work. It consists of two functional components: an initialization module at the start of the genome that prepares the organism for replication; and a copy-loop at the end of the genome that copies the individual instructions in the genome to the offspring's memory. A long "backbone" of the virtual CPU instruction `nop-C` occupies the space in the genome between these two sections. The `nop-C` backbone provides blank space for adaptations to evolve.

The `h-alloc` instruction at the start of the organism's initialization module causes the virtual CPU to append `nop-A` instructions to an organism's genome, tripling the genome's length. The first part of the genome also uncouples the read and write-head, the former still pointing to the first instruction in the genome and the latter moved to the first site to receive an instruction copied from the parent genome. The action is accomplished by the `h-search` instruction using `nop` templating, wherein a specific pattern of `nop`s, complements of the `nop`s located after the `h-search` instruction, are located within the genome. The `h-search` instruction places the flow-head (the final of the four genome pointers) after that pattern. The `mov-head` instruction then places the write-head at the location of the flow-head. This prepares the genome for copying, since the read-head is at the start of the genome and the write-head is now at the start of the newly allocated space for the offspring.

In the default organism in Figure 2.1, all copying from the parent genome to what will become the offspring occurs in the copy-loop at the end of the parent genome. The copy-loop contains the h-copy instruction, which copies the instruction at the site pointed to by the read-head to the site pointed to by the write-head. As nop instructions are copied, their identities are noted and stored in a special purpose label register. This register is used by the instruction if-label to change the behavior of the instruction pointer to conditionally process pieces of code. When a non-nop instruction is copied, the label register is cleared.

The copy-loop of the genome depicted in Figure 2.1 terminates when the instructions at the end of the genome are copied because the h-divide instruction gets processed when the complement of the nops located after the if-label instruction in the copy-loop gets copied to the offspring's genome. Although h-divide can be processed at any time, it will not be successful unless specific conditions, set in the experiment's configuration, are met. These conditions include the number of instructions processed and the number of instructions copied. Processing h-divide before these conditions are met consumes SIPs but performs no work.

Once h-divide is successfully processed, the offspring's memory is cleaved from the parent's genome by using the position of the read and write-heads to define the beginning and end of the offspring genome. Any excess instructions allocated by h-alloc are removed. Avida then provides the offspring genome its own piece of virtual hardware and places the organism into the population. Since the offspring has not processed its genome, it initially receives the same number of SIPs as its parent.

2.1.2 The Avida population, environment, and organism fitness

The world that the Avidians occupy may be thought of as a virtual Petri dish. Typically this world contains a matrix of cells that only a single Avidian may inhabit at a time. These cells are linked together to form a torus – a borderless two-dimensional shape. This world can be further subdivided into isolated subunits called demes. Chapter 5 will make use of demes to select mutation biases.

A population is the collection of organisms occupying the world at a particular time.

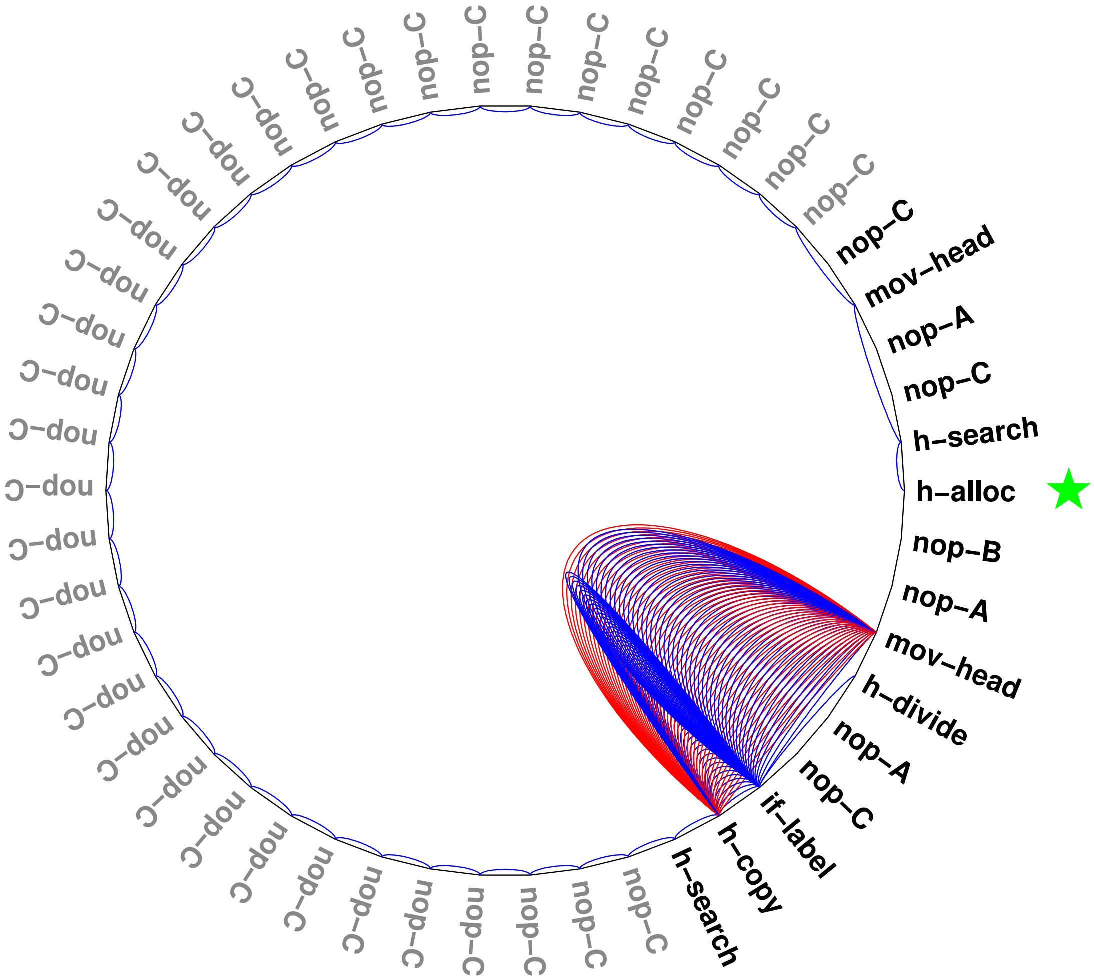

Figure 2.1. **The genome of a default Avidian.** Execution of instructions begins at the pentagram and proceeds counter-clockwise. Forward movement of the instruction pointer is shown in blue; backward movement is shown in red. Sites comprising the `nop-C` backbone are in gray. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

As organisms reproduce, their offspring are placed in the virtual Petri dish, many times displacing and killing an existing organism. The specific placement of an organism upon its birth depends on settings established by the researcher. Typically, offspring are either placed randomly in the world to produce unstructured populations or in one of the cells immediately adjacent to the parent (without concern for the properties of the organism the offspring may be replacing) producing structured populations.

Organisms that reproduce faster will occupy an ever increasing amount of the digital Petri dish, and eventually drive all other clades extinct unless frequency dependent selection is possible, such as when there are multiple limited resources. For this work, it is the competition for space that drives selection.

The duration of an Avida experiment as well as the timing of researcher-specified events is often measured in a unit called an update. Updates may be thought as the time it takes a population to exhaust a fixed supply of SIPs. Typically, when an organism receives a SIP, it is able to process the next instruction in its genome on its virtual CPU. The number of SIPs available in any update is dependent on the population size. For experiments in this work, the total number of SIPs available per update is 30 times the number of organisms in the population. How the total number of SIPs in a given update is allocated to individual organisms creates relative processing speed differences within the population.

An organism-specific value called merit determines how many SIPs the organism is given in per update. Going back to the bacterial experiment analogy, merit may be thought of as the “metabolic” rate of the organism. The higher its value compared to its peers, the “faster” an organism processes its genome. Merit is initially set to be proportional to the minimum of either the number of instructions the organism processes in order to divide or the number of instructions it copied into the genome of its last offspring. In practice, this prevents organisms from being penalized for having longer genomes. Additional merit may be gained by environmental adaptation.

Merit is thus defined as:

$$M = M_0 \prod^{T} B_t \tag{2.1}$$

where M_0 is proportional to number of SIPs processed prior to replication or offspring length, T is the set of observed adaptations that are rewarded with an associated multiplicative increase of magnitude B_t.

The environment in Avida is everything "outside" of an organism. In its simplest form, the environment supplies organisms with SIPs and requested pseudo-random bit strings. This simple environment also examines the bit strings that an organism produces when it processes an instruction that outputs a bit-string. Since the environment keeps track of both the inputs an organism receives and the output it sends, it is possible to treat the organism as a black box that maps inputs into outputs. By adjusting merit based on observed mappings, organisms receive a different number of SIPs, effectively changing the relative speed at which they operate.

Researcher-defined mappings are called tasks. Typically tasks are either simple mathematic or Boolean logic operations, but they can also be complex behaviors or interactions. The set of tasks and their associated merit adjustments define the reward structure of the environment. The most common reward structure used in an Avida experiment is the `Logic-9`, shown in Table 3.1. In the `Logic-9`, performing any of nine Boolean logic tasks provide merit rewards. The size of the reward roughly is based on the complexity of the individual tasks, as defined by the number of `NAND` operations required to perform each operation. Merit rewards are applied multiplicatively to the merit of an organism. There is a maximum of one reward per task per organism and no limit on the number of organisms that can be rewarded for performing the task. Consequently, organisms which perform a large number of complex tasks are selectively advantageous in the `Logic-9` environment.

The absolute fitness (ω) of an organism is the ratio of its merit to its replication efficiency:

$$\omega = \frac{M}{\tau} = \frac{M_0 \prod^{T} B_t}{\tau} \tag{2.2}$$

where M is the merit of the organism as defined in Equation 2.1, and τ is the number of SIPs needed for replication. Unlike a genetic algorithm, which typically use *explicit* fitness functions to determine which genomes get propagated to the next generation fitness in Avida is *implicit*, with the environment providing metabolic rewards to organisms for performing tasks. In Avida, high absolute fitness does not guarantee survival; Avida is fitness-agnostic when it comes to replacing organisms with newly created offspring. Consequently, organisms with high absolute fitness may be replaced by less fit organisms.

2.1.3 Instructions, mutations, and mutation biases

Mutations occur during replication in the form of substitutions, insertions, and deletions. The rates of these mutations are established by the researcher at the start of the experiment and may be modified mid-experiment.

Unlike the four nucleobases in DNA or the standard 20 amino acids in proteins, Avida typically uses a 26-letter genetic alphabet with the symbols `a-z` mapped to each instruction. Further different from natural systems, there is no translation step between genome and function in Avida; instruction identity is atomic, although the behavior of an instruction may be modified by other instructions (particularly no-operation or `nop`s) anywhere in the genome. For example, the presence of a `nop-A` immediately following an `add` instruction will change which virtual register receives the summation. Appendix A defines the behavior of each instruction as well as how its behavior is modified by the presence of `nop` instructions at various places in the genome.

A collection of virtual CPU instructions and any metabolic or mutagenic properties associated with each instruction defines an instruction set. For the purposes of the experiments in this work, all mutation biases are represented by predefined instruction sets. I use the term "mutation bias" to refer to a single instruction set with a distinct set of mutagenic properties for each instruction. A mutation bias skews the probability of substitution to or insertion of each instruction from parent to offspring.

Although mutations may occur to an offspring's genome either during replication or after dividing from the parent, the experiments in this work apply substitutions during instruction copying and insertion and deletions (indels) occurring during division.

When a substitution or insertion occurs, a new instruction is drawn at random from the instruction set. By default, every instruction has the same chance being selected. However, the mutation probability for each individual instruction may be altered by changing its redundancy in the instruction set. An instruction's redundancy is simply its abundance in the instruction set. The higher an instruction's relative redundancy (abundance), the more likely it is to be drawn from a random sample of the collection.

Another difference between Avida and natural systems is that there is the possibility of a substitution occurring and not altering the offspring's genome. Since mutation outcomes are based on relative abundance of instructions in the instruction set, there is a possibility that a mutation occurs but produces no change in the genome. This decouples the researcher-specified mutation rate and the actual mutation rate. Consequently, the mutability of a particular site depends on the instruction at that site and its redundancy. For a particular site with instruction i, its rate of mutation is $(1-p_i)\mu$, where p_i is the relative probability of picking the instruction and μ is the substitution rate. The difference between the specified and actual mutation rate will become important when I discuss the reason for the selection of a mutation bias in Chapters 4 and 5.

The manner in which instruction sets with different redundancies (mutation biases) are inherited and mutate differ between experiments. In Chapter 3, a single mutation bias will be applied to populations as a whole to find the differences in evolutionary behavior of obligate populations under different mutation biases. In Chapter 4, multiple mutation biases are present in the population and are inherited perfectly from parent to offspring. Both inheritance of multiple mutation biases and the mutation between biases is allowed in the experiments in Chapter 5 by specifying an additional mutation parameter, μ_b, which is the probability that the bias will be altered during transmission from parent to offspring.

2.2 An Avida experiment

Each Avida experiment is an instance of evolution [31]. During the course of an experiment, a researcher can capture a variety of information. Some examples include the average fitness of the population, the kinds of adaptations present, and the evolutionary history of each organism in the population. Figure 2.2 contains plotted data from one of the experiments in Chapter 3. This experiment will serve as a case-study of a typical Avida experiment.

Each Avida experiment is an independent instance of digital organism evolution. In order to test the effects of different treatments, experiments are often repeated multiple times with identical configurations. A single integer value, called a seed, differentiates replicates by initializing a random number generator to produce different values when stochasticity is required. For example, the placement of an organism in the population or which sites in the genome get mutated depend on the values produced by the random number generator. Since all of the stochasticity in each experiment depends only upon a single random number seed, an experiment may be repeated multiple times with the same outcome.

As all organisms in the initial population are of the same fitness, Avida gives each organism the same probability of receiving a SIP. As the clones process their code, they will begin to copy their genome. It takes the default ancestor 189 SIPs to create a copy of itself (τ in Equation 2.2). As described above, this process involves four steps: (1) allocation of offspring memory, (2) configuration of the virtual CPU to begin copying, (3) copying each instruction from parent to offspring, and (4) termination of the copy process when the instruction `h-divide` is processed. During the copying of an instruction, it may be mutated prior to being written to the offspring's genome. During division, additional instructions may be inserted or deleted from the offspring genome. These substitutions and indels provide heritable variation for selection to act upon. Following replication and division, the parent and offspring CPUs are reset to their initial state with merit updated based on the tasks the parent has completed.

At the end of the first few updates of an Avida experiment, some organisms will have gained more SIPs than others due to chance. This set of organisms will have a greater chance of propagating to the next generation. As the experiment proceeds, mutations accumulate in the genome and may influence either the efficiency of replication or the merit (τ and M in Equation 2.2). If these organisms are not overwritten, they will sweep the population by replacing all organisms with their relatively more fit descendants.

Avida experiments proceed until a researcher-specified condition is met. Typically this is a limit on the update time or average generation count of the population. In Chapter 4, experiments are set to terminate when one of two subpopulations drives the other to extinction. By the end of an Avida experiment, the handwritten ancestor will have adapted to the environment. These adaptations may not be optimal or exploit the environment completely; the maximum population fitness at the end of each experiment will vary for the same experimental setup.

When examining the historic path of a population, the primary line of descent (PLoD) is often used. This line of descent begins with the default ancestor and contains each ancestral genotype produced throughout time until the final dominant genotype in the population at the end of the experiment is reached. Each step along this line of descent is at a different genetic depth: the number of distinct genotypes beginning with the handwritten ancestor at depth 0. The PLoD is stored during the course of the experiment and may be analyzed afterward. Figure 2.2-A shows the relationship of the PLoD to the rest of the Avida population over the course of an experiment.

Each genotype in the PLoD may be examined at the end of an experiment for historic behavior; they may also be re-examined in a virtual test environment. Many times the behavior of a genotype may be more complicated than its history suggests. For example, phenotypic stochasticity, the random ability for a phenotypes to emerge, is possible. Phenotypic stochasticity can alter an organism's fitness because of genome-environment interactions altering gestation time or task demonstration. To test for phenotypic stochasticity, I loaded genomes

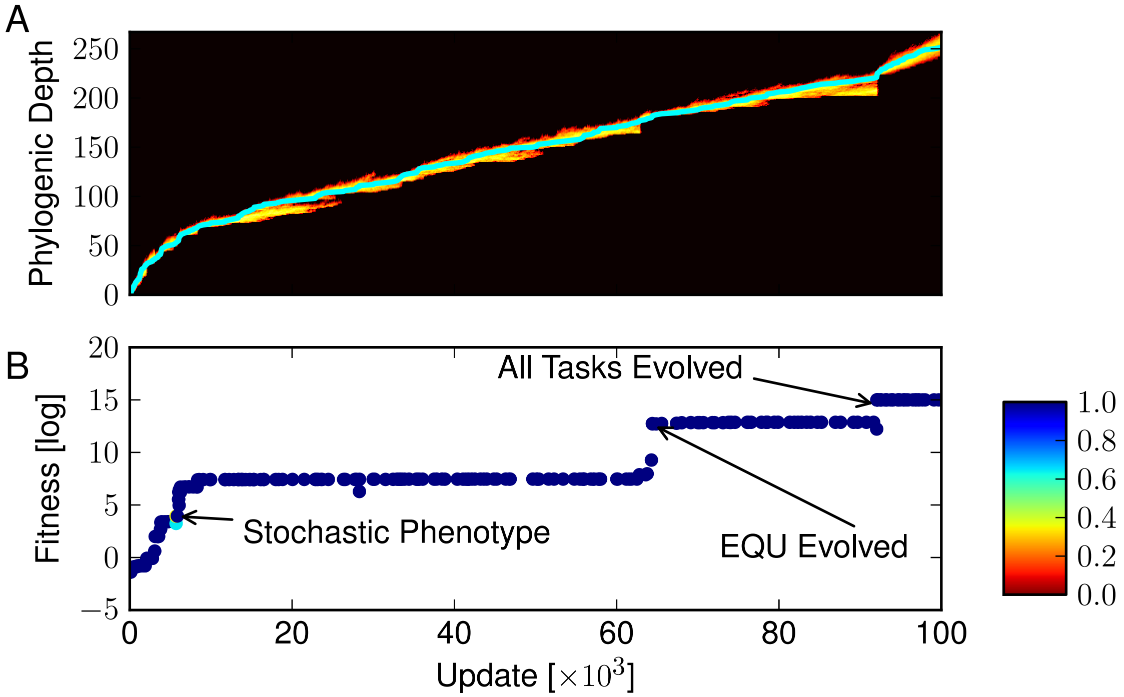

Figure 2.2. **Evolutionary trajectory of an Avida experiment.** A. A flame graph showing the abundance of organisms at different phylogenetic depths over time. Abundances increase from black through red to yellow. The trajectory of the PLoD is plotted as a cyan color line. B. The fitness trajectory of the PLoD over time. The color of the markers indicate the stochasticity of phenotypes as measured in the test environment.

on the PLoD into a virtual test environment and fully-processed them 1000 times. Phenotypes are marked unique if they have a different set tasks, a different number of times tasks performed, or if gestation time is different among all tests. I classify genotypes as being stochastic if they posses more than one phenotype after analysis; otherwise I declare them to be static. Figure 2.2-B shows the fitness behavior of our case-study experiment with the color of colored markers indicating phenotypically stochastic genotypes and annotations showing which tasks are being performed.

2.3 Experiment-specific features

Additional features that will be used in the experiments in this work will be introduced when they are used. These include the ability to switch mutation biases for the entire population in Chapter 3, the ability to have multiple mutation biases in a population in Chapter 4, and the ability for individual organisms to possess and mutate between different mutation biases using both multiple demes and single population setups.

CHAPTER 3

The Adaptive Consequences of Mutation Bias

Adaptive evolution requires mutation, inheritance, and selection. Altering the outcome of any of these three processes has the potential to change the evolutionary trajectory of a population in a non-trivial manner. Consequently, biasing the ability of a population to sample its genetic landscape has the potential to alter evolutionary adaptation. In order for a mutation bias to evolve as an adaptation, a bias must (1) alter evolutionary adaptation in a beneficial manner, (2) be visible to selection, and (3) be subject to heritable variation. In this chapter, I explore the first of these three requirements.

To begin, I created a pair of potentially advantageous and disadvantageous mutation biases using information from evolving populations. I then examined the ability of these mutation biases to alter fitnesses, evolve complex tasks, and transform genetic compositions of obligate populations. Taken as a whole, these experiments demonstrate that there exist mutation biases that are able to beneficially alter the evolutionary trajectory of populations of digital organisms.

3.1 Creating a favorable mutation bias

Given that most changes in a genome are deleterious, deriving a mutation bias that will allow populations to adapt to their environment better than chance requires information about the environment and its effects on the population's genetic composition. There is no a priori expectation that a random mutation bias will serve any adaptive effect. As such, I used information about successful mutations to identify a beneficial mutation bias.

In order to find a potentially beneficial bias, I began by employing the methodology used to create the Dayhoff PAM matrix (an amino acid substitution matrix) using the virtual instructions in Avida. I then removed some of the restrictions of the PAM-matrix methodology and derived a single vector of mutation inflow rates probabilities. From this vector, I created the two mutation biases used in this chapter: the positive (`POS`) and negative (`NEG`) biases.

3.1.1 Dayhoff PAM-like matrix

The inspiration for using a mutation bias to improve evolution came from earlier work to create a Dayhoff PAM-like matrix for Avida [7]. The PAM-matrix is hand-created amino acid substitution matrix for protein residues a particular PAM distance apart. PAM distance is defined by the number of "accepted" amino acid substitutions per 100 residues between two proteins[1]. Consequently a PAM-N matrix defines the likelihood of some amino acid i being substituted by amino acid j in proteins a PAM distance N apart. Because the history any two residues are not rooted in the creation of the PAM matrix, the PAM-N matrix is bidirectionally tabulated. That is, the count of substitutions from $i \rightarrow j = j \rightarrow i$. Traditionally, the PAM-250 is used in maximum likelihood estimations; though to some extent newer bioinformatic applications substitute the PAM matrix with another type of transition matrix, the BLOSUM-62 [16].

One of the nice summary statistics from the Dayhoff PAM matrix is its asymptotic "steady state" at large PAM distances. By applying the PAM matrix many times (or taking its first eigenvector), the distribution of final states of transition becomes uncoupled from initial conditions. This vector is the theoretical distribution of amino acid states between

[1] "Accepted" is not a well-defined term. Dayhoff and Schwartz define it to be when a mutation becomes the "new predominant form" for a species.

well-separated residues. I exploited this property to create the mutation biases I will use in later experiments.

Using Avidian genomes instead of protein sequences, I repeated Dayhoff's method to create a transition matrix, albeit with slight modifications. To begin, I evolved a set of 20 replicates populations under the `Logic-9` reward structure. I list those few non-default configuration settings in Appendix B.2, though it doesn't vary much from the default Avida setup. To get an equivalent to a PAM-1 matrix, I collected whole populations and their evolutionary history from each replicate every 100 updates. At the end of the experiment, I combined these populations and their evolutionary history into a single genetic tree, which is a sample of the entire evolutionary history of the population. Depending on how I tabulated the substitutions, I was able to get different types of transition matrices.

Unlike Dayhoff's amino acid substitutions, the evolutionary history of each genotype in Avida is perfectly encoded. Consequently, two assumptions could be relaxed in creating an Avidian PAM-like matrix: bidirectionally (so $i \to j \neq j \to i$) and the assumption of a PAM-distance (since the distance between any two genotypes need not be inferred). These simplifying assumptions made the tabulation of transition matrices easier and provided a clearer means of interpreting the resulting matrix. Instead of the resulting matrix representing distances 1 PAM apart, the individual transition probabilities the possibility of a non-lethal mutation from instruction i to instruction j within a single generation.

From the historical information contained in the population samples, I created two different matrices M^{bi} and M^{uni}, where I used bidirectional and unidirectional tabulations, respectively. For either matrix $M_{i,j}$ is the probability of an instruction i mutating to instruction j during replication. I created the bidirectional M^{bi} by equating transitions between instructions i and j. The bidirectional matrix is not a symmetrical matrix ($M_{i,j} \neq M_{j,i}$) because each row i is normalized by the total number of $i \to j$ transitions observed. I created a second matrix M^{uni} by keeping the tabulations unidirectional.

Both methods have some merit: bidirectional assumes some degree of neutrality for substitutions (as assumed in the PAM matrix [7]) and unidirectional captures the exact evolutionary trajectory of the population. I chose to use the bidirectional matrix as a basis for the remainder of my work. The chief reason for this was incidental: the "steady state" value of M^{bi} deviated from a uniform distribution more so than M^{uni}. As I will show later, this decision did allow me to produce a `Logic-9` friendly mutation bias, albeit using a more simplified version of the bias than the original M^{bi} would produce.

Figure 3.1 shows M^{bi} and its near "steady-state" condition. As mentioned before, one of the advantages of the "steady state" is that the original identity of an instruction no longer plays a determining factor in the outcome of a substitution. That is, any instruction i has an equal probability of mutating to a particular instruction j. Put another way, an infinitly-sized population of genotypes under a random walk can expect to have a composition $\hat{b}$ equal to one row of the matrix shown in 3.1-B after a sufficient amount of time has elapsed.

Although in the short term, mutations between states in the genome may be dependent on the initial identity of each site, the long-term consequences of a mutation bias should drive populations to a similar genetic composition outcome, barring strong selection. Consequently, a major assumption of my work is that a mutation bias can be represented by a single vector $\hat{s}$ where each element is the relative abundance of one genetic state as compared to another. Selection, of course, disrupts this assumption because the context of one genetic state over another has the potential to be adaptively meaningful; but the overall direction of genetic change is adequately captured by $\hat{s}$.

3.1.2 Further simplifications of biases

Although the mutation bias generated from M^{bi} above is well-specified, some of the parameters used to generate it were chosen more arbitrarily. For instance, the sampling of populations every 100-updates might give a good approximation for the complete evolutionary history of the population but was chosen more for observational completeness of the transition matrix rather than capturing evolutionarily meaningful artifacts. Further, I had

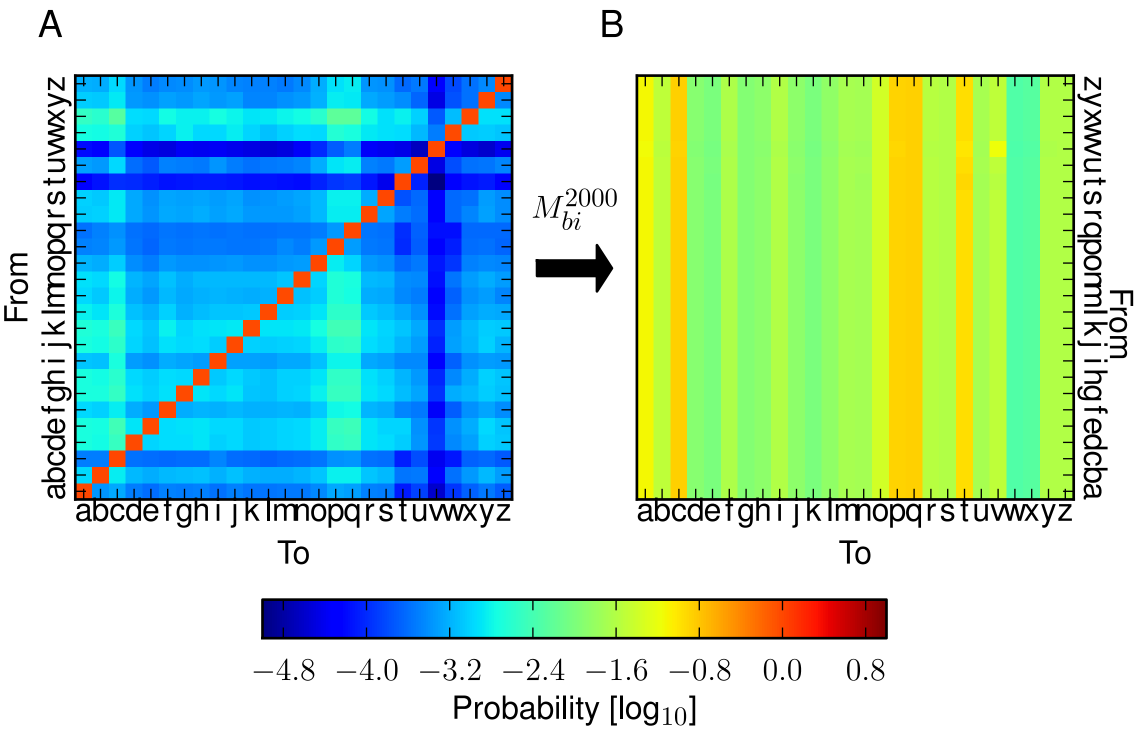

Figure 3.1. **Transition matrices.** Letters along the axes correspond to virtual-CPU instructions in Appendix A. A. A Markov transition matrix generated from point and insertion mutations accumulated during the course of evolution in the Logic-9 reward structure. B. The near-saturation state of this matrix.

some question as to whether each sample replicate should be equally weighted when tabulating the values for any resulting transition matrix. After all, some replicates will have populations more adapted to the environment than others due to chance alone. Finally, there was the distribution of the bias: how meaningful are the many different relative probabilities?

In order to simplify $\hat{s}$ and remove variation in $\hat{s}$ due to arbitrary decisions to the concerns above, I decided to use only three different instruction probabilities: low, medium, and high. These three probabilities were loosely fit to the original eigenvector $(\hat{b})$ of M^{bi} as follows: medium is $3\frac{1}{3}$ more likely than low; and high is 5 times more likely than low.

Once the relative ratios were established, the question became how to apply them across the 26 instructions of the default Avida instruction set. As shown in Figure 3.2 there are clearly some instructions that are more represented in $\hat{s}$ than others, especially those dealing with logical task IO {`nand, io`} NOP modification {`nop-A, nop-C`} , or replication {`h-copy`} . Since other work has shown there to be an antagonistic relationship between task acquisition (and therefore higher fitness) and replication efficiency [34], I decided to place {`h-copy`} in the low category.

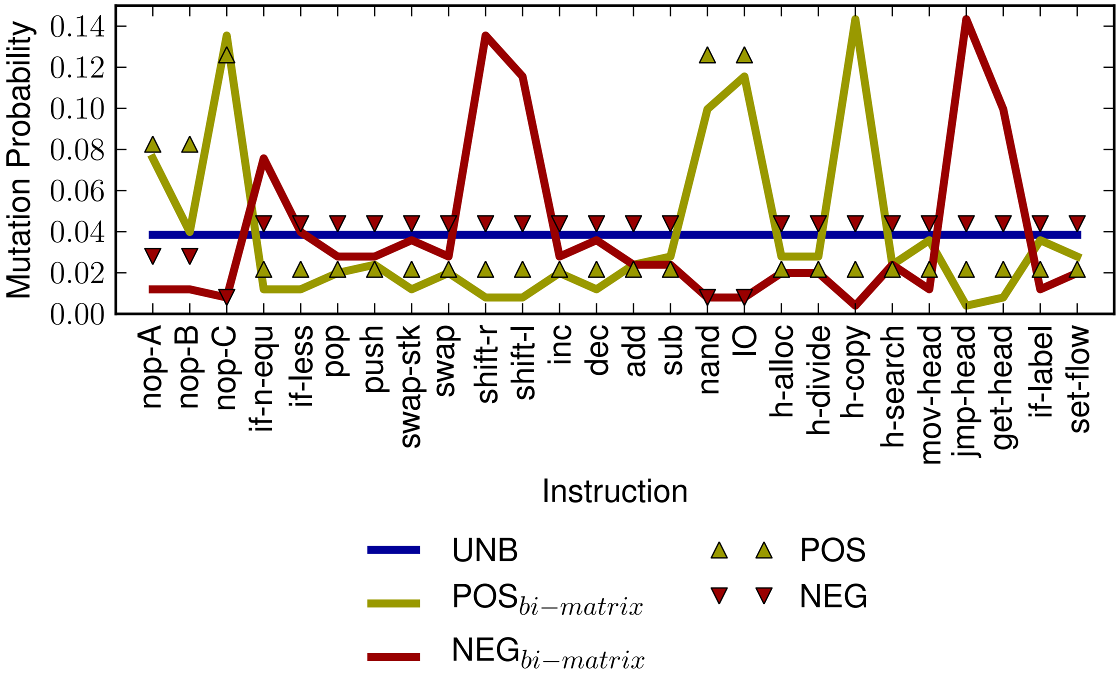

Figure 3.2. **A comparison of the saturation values of the transition matrix and corresponding simplified values used in the `INST-3` mutation bias spectrum.**

3.1.3 The `INST-3` spectrum

For the remainder of this work, I will refer to the potentially advantageous mutation bias as `POS`; `NEG` will be an "anti-bias" made by swapping instructions with high and low probability in `POS`. `UNB` will be a bias where each instruction is equally probable. Figure 3.2 compares the original Dayhoff-like versions of these biases to their simplified derivatives.

In order to simplify terminology later on as I add more mutation biases, I refer to the set of mutation biases identified as UNB, POS, and NEG in Figure 3.2 as the INST-3 mutation bias spectrum. Although transitions between the different biases are prohibited in the work presented in this chapter (unless explicitly stated), it will be helpful to think of the mutation biases as being states in an evolvable system. This will become more important for the evolution of mutation biases in Chapter 5.

3.2 Mutation biases can change the evolutionary trajectory of obligate populations.

All experiments in this chapter will be in one of the two environment reward structures listed in Table 3.1. The Logic-9 reward structure is well studied in Avida. It rewards fitness based upon the complexity of binary logic tasks on received bit-strings. The most complicated task in the Logic-9 is EQU. Previous work shows that EQU requires simpler rewarded tasks to evolve [21]. EQU also tends to evolve largely through expatiation: EQU co-opts sites in the genome from other tasks [28].

As a test of the ability of a mutation bias to influence the effects of adaption, I repeat the experiments of Lenski et al. [21] to determine whether or not a presumed beneficial mutation bias can modulate the rate and number of replications acquiring the complex task EQU, its effect on fitness, and its ability to evolve EQU in an environment that does not reward for simpler tasks.

My experiments in this chapter examine three different questions: (1) can mutation bias alter the outcome of evolution if it is applied at the start of an experiment (Section 3.2.1)?; (2) can a change in mutation bias alter the outcome of evolution in already adapted populations (Section 3.2.2)?; and (3) can mutation bias alone evolve the complex task EQU (Sections 3.2.3)? I will also examine how bias changes the genetic composition of populations (Section 3.3).

	Task								
	NOT	NAND	AND	ORN	OR	ANDN	NOR	XOR	EQU
Logic-9	2^1	2^1	2^2	2^2	2^3	2^3	2^4	2^4	2^5
EQU-only	0	0	0	0	0	0	0	0	2^{25}

Table 3.1. **Environment reward structures used in this chapter.** Rewards are applied multiplicatively when an organism divides and is contingent on successful demonstration of the displayed Boolean logic operations on bit-strings provided by the environment.

3.2.1 The effects of mutation bias on task adaptation

A natural place to begin exploring the effects of mutation bias on evolution is to examine the influence of biases on a set of replicate treatments, all beginning with a common set of initial conditions. I used differences in the speed of adaptation (task acquisition or fitness gain per update), frequency of complex task evolution, and final average population fitness as measures of environmental adaptation.

The first set of experiments to test the effect of mutation bias on adaptation begins with three treatments: POS, UNB, and NEG-only variants. Each treatment had 300 replicates with evolution beginning with a fully-seeded population of the default length-50 ancestor genome (see Figure 2.1) under the Logic-9 reward structure. Each replicate ran for 10^5 updates.

At the end of each experiment, I identified the PLoD (primary line of descent) for each replicate, and evaluated its genotypes for fitness, task performance, and stochasticity. I tabulated a population as performing a task if any genotype along the PLoD deterministically demonstrated the task. I used the behavior of genotypes along the PLoD as a proxy for the historical behavior of the population since reproduction is asexual and metabolic rewards for tasks are inexhaustible. This combination of factors makes it unlikely for more than one deeply-rooted clade to persist for long periods of time. Consequently, the PLoD should reflect the predominant historical state of the population over time. I used the number of steps from the common ancestor to the first genotype performing each task as a measure of task acquisition speed. I also collected the average final fitness of each population and the

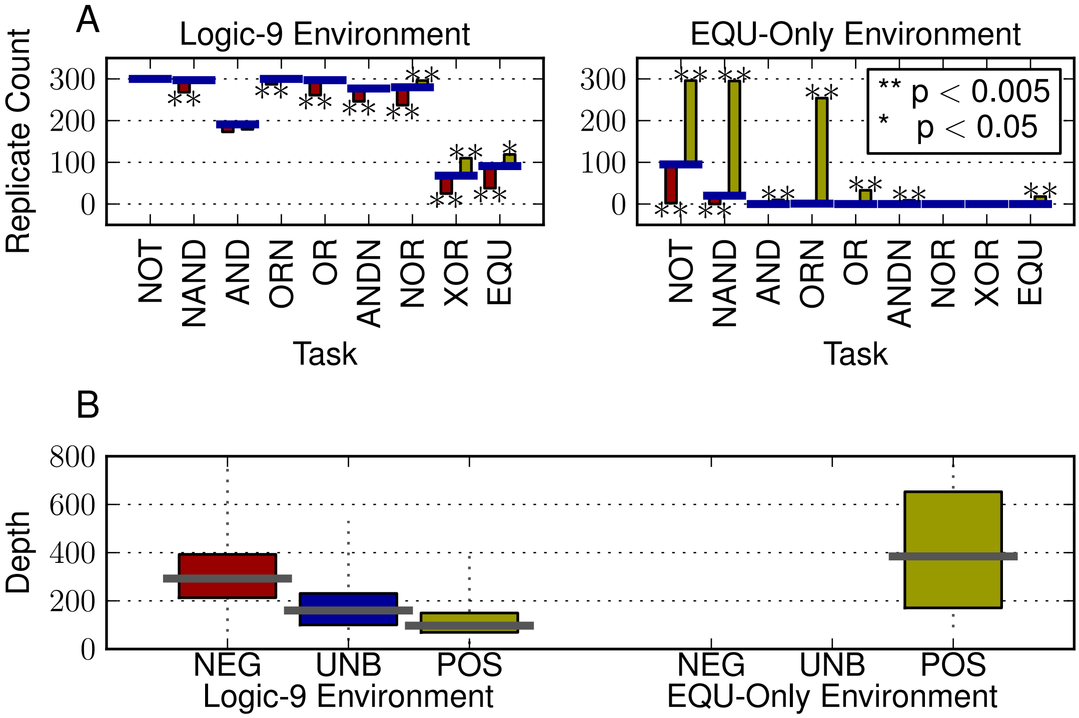

Figure 3.3. **The evolution of tasks under different mutation biases and environment reward structures.** A. The number of replicates acquiring each task under mutation biases: yellow, `POS`; blue, `UNB`; red, `NEG`. B. The distribution of the number of genetic steps from the ancestor genotype to the first acquisition of `EQU` along the PLoD. Colored boxes show the the inner quartiles; the median is shown by a gray line. The upper and lower range of the distribution is shown by dotted lines.

genetic composition of the population over time.

Since I based the `POS` upon successful mutations in the `Logic-9` reward structure, I hypothesized that the `POS` treatment will adapt better to the environment (both in terms of adaptive speed and overall ability), and end with final populations that have a higher average final fitness. Likewise, the `NEG` treatment should be less adapted to the environment than both `UNB` and, especially, `POS`.

Figures 3.3 and 3.4 show the results of these three treatments. The results are as expected: the `POS` replicates showed significantly higher final fitnesses than the `UNB` replicates (Figure 3.4, median=14.93 [log_2] versus 14.35 [log_2], $p = 0.002$, Mann-Whitney U-Test). In terms of tasks, the POS replicates demonstrated complex tasks in significantly more replicates than UNB replicates. Specifically, the tasks NOR, XOR, and EQU had significantly more replicates demonstrating them along the PLoD ($p < 0.05$, 1-tailed Fisher's Exact Test, Bonferroni corrected). The speed of task acquisition also significantly improved for all tasks ($p < 0.05$, Mann-Whitney U-Test Bonferroni corrected) except xor, which failed to fall within the Bonferroni corrected confidence interval ($p = 0.010$, uncorrected Mann-Whitney U-Test). A good example of this change in acquisition speed is for the most complex tasks in the Logic-9 reward structure, EQU. EQU acquisition depth of populations under POS decreased relative to UNB populations, with a median EQU acquisition depth of 160 genomes in UNB populations as compared to 90 genomes in POS replicates ($p < 0.001$, uncorrected Mann-Whitney U-Test).

Also as expected, the replicates under the NEG bias did not fair as well as the UNB or POS replicates. Final average population fitness was significantly lower than UNB (median=12.94 [log_2], $p < 0.001$, Mann-Whitney U-Test). Likewise, fewer replicates obtained the three most complicated tasks ($p \ll 0.001$, Bonferroni corrected) under the Logic-9 reward structure.

Overall, performance on all adaptive metrics was lowest in the NEG replicate populations and highest in the POS populations. These results demonstrate mutation bias is sufficient to change the evolutionary trajectory of populations, both helping and hindering adaptation. However, the extent to which the two biases explored above can influence adaptation has not been fully demonstrated. The experiments above all began with the same, hand-written ancestor; whether adapted populations can utilize a mutation bias to adapt more so than mutationally unbiased peers has not been shown. Further, how well the Logic-9 reward structure is captured by the POS bias is also not known.

3.2.2 Mutation bias can alter evolutionary trajectory of adapted populations

In order to test whether or not mutation bias can influence already adapted populations, I repeated the experiments above but added five more treatments. All populations began

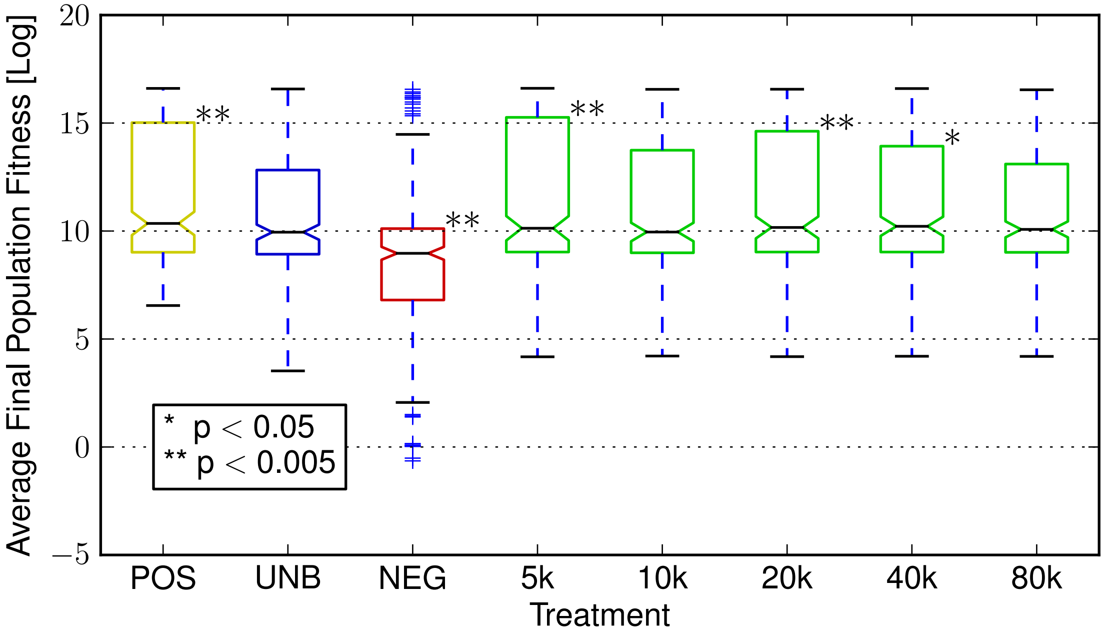

Figure 3.4. **The distribution of final population average fitnesses under different treatments in the Logic-9 environment reward structure.** Asterix markers show distributions significantly different from the UNB treatment using a 2-tailed Mann-Whitney U-Test. POS, NEG, and UNB are treatments where populations are forced to use one mutation bias; the remaining treatments switch biases from UNB to POS en masse at the times indicated.

as before with the same default hand-written ancestor and configuration. Each population began under UNB conditions. Each treatment switched from all organisms being unbiased to using the POS mutational bias at either 5, 10, 20, 40, and 80 $\times 10^3$ updates. Also as before, each treatment was repeated 300 times.

The fitness comparisons of these treatments are shown in Figure 3.4. Populations that switched to POS at 5, 20, and 40 $\times 10^3$ had average final population fitnesses higher than UNB populations ($p < 0.05$, Mann-Whitney U-Test, Bonferroni corrected). Likewise, those same treatments showed significant improvement in EQU acquisition, displayed in Figure 3.5. Specifically, those treatments showed individual improvements in the number of replicates that evolved EQU after the switch from UNB to POS ($p < 0.05$, 1-tailed Fisher's Exact Test, uncorrected) relative to replicates continuing in UNB to the end of the experiment.

The two anomalous populations at 10 and 80×10^3 updates did not see improved fitness or EQU acquisition. The latter can be explained by a lack of time to evolve EQU[2]. I believe the difference at 10×10^3 is an anomaly or potentially the result of a non-uniform probability of getting EQU with some transition around 10×10^3 as points immediately before and after it show improved ability to acquire EQU.

Although the results from this set of experiments are not as strong as previous comparisons between treatments that were either always under UNB or POS, there is evidence that the effect still exists. In the next chapter I will explore these ideas more when I compete mutation biases against each other using genotypes evolved under different biases. I do, however, find these are results enough to support the claim that environmentally-adapted genotypes can adapt to the Logic-9 reward structure better under the POS bias than under unbiased conditions.

3.2.3 Mutation bias can allow for the evolution of complex tasks de novo

The POS bias was generated using information from successful substitutions in the Logic-9 reward structure; consequently, it contains some information about it. Replaying these mutations in the form of a bias as shown above demonstrates an ability to move populations to better adapt to the Logic-9 reward structure. One plausible reason for this behavior is that POS mutational bias creates a pressure to emulate the genetic-composition of simple tasks. If this is the case, then these simple tasks should appear in greater propensity under POS than under UNB even in the absence of selective pressure. In other words, chance alone could allow for simple tasks to occur. Fixation would not occur without selection (except perhaps through unlikely drift). However, if this mutational pressure is strong enough,

[2] Using a sliding window of 20×10^3 updates and examining UNB-only replicates, there is diminishing ability of replicates to obtain EQU over the course of the runs. Data not shown.

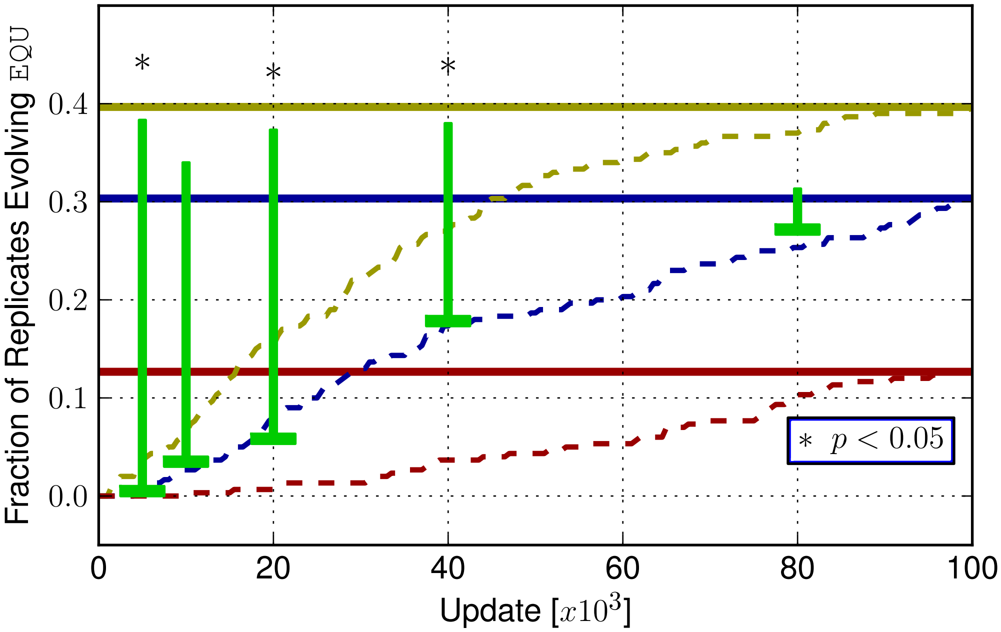

Figure 3.5. **The acquisition frequency of EQU.** Colored horizontal lines show the fraction of replicates that evolved EQU in populations that experienced only one mutation bias (yellow=POS, blue=UNB, red=NEG). Dashed colored lines show the fraction of replicates that have evolved EQU over time. Green horizontal bars show the fraction of replicates that have demonstrated EQU at each of the five updates during which a switch from POS to UNB occurred. Vertical green bars show the additional fraction of treatment replicates that acquired EQU after the swap from UNB to the POS. Treatments that swapped from UNB to POS where significantly more replicates acquired EQU after the swap than the UNB-only treatment are indicated with an asterisk.

there is a possibility for complex tasks to evolve “spontaneously” through a manipulation of mutation probabilities alone. Adding strong selection for complex tasks then allows for the possibility of fixation and maintenance in the population.

In order to test the possibility that mutation bias alone can allow for the spontaneous generation of complex tasks, I repeated the first set of experiments in this chapter in the EQU-only reward structure. As the name suggests, this reward structure only provides merit increases for the task EQU. In Lenski et al. [21], this reward structure was used as a control to test whether or not simpler tasks were required to act as building blocks for the evolution of the task EQU. The authors of the study found that simpler rewarded tasks allow for, and are required for, the evolution of EQU. Here, I repeat their experiment with minor modifications (as before, see Appendix B.1), to test whether or not a mutation bias is capable of evolving EQU in a reward structure absent these necessary building blocks.

Under each mutation bias in INST-3, I evolved 300 replicates for 10^5 updates under the EQU-only reward structure, beginning with fully seeded populations of the default length-50 handwritten ancestor. As before, I examined the PLoD for task acquisition. Although the simpler 8 tasks of the Logic-9 were not rewarded, chance does allow for spontaneous generation of these tasks (see Figure 3.3).

EQU evolved in 18 out of the 300 replicates under the POS bias. EQU was not demonstrated by any replicate under either the UNB (repeating the results of Lenski et al.) or NEG biases. Replicates under the POS bias also showed spontaneous evolution of significantly more simple tasks (see Figure 3.3) than the other two biases, despite the fact that they were not rewarded. These results indicate that the POS bias is capable of emulating some of the selective behavior of the Logic-9 reward structure, allowing for otherwise unlikely tasks to evolve.

One interesting observation is that, despite the fact that these populations were only rewarded for EQU, simpler building blocks may have contributed to the evolution of EQU. Specifically 13 of the 18 replicates showed evidence of phenotypic stochasticity for EQU immediately prior to EQU being correctly implemented. Stochasticity, as discussed in Chapter 2, is measured by repeatedly testing genomes with different sets of pseudo-random numbers as inputs. An example of an input-dependent version of EQU performed by one of the POS replicates is this implementation:

$$EQU \approx 1 + I_1 + I_2 \tag{3.1}$$

where, I_1 and I_2 are two inputs given to the organism via an IO instruction execution. It so happens that this has an approximately 10^{-3} probability of correctly calculating EQU because of the length of the bit-strings randomly generated by the environment. Specifically, each organism receives one of three different bit-strings whenever it executes an IO. Each bit-string received is 32-bits in length, with the upper 8-bits being deterministic to assure that all bit-wise combinations are present for any pair of the three inputs. The lower 24-bits of each bit-string are randomly generated.

Genotypes that implemented the simple function in Equation 3.1 exploited the fact that there is a $(\frac{3}{4})^{24}$ ($\approx 10^{-3}$) chance that at least one of the first 24-bits in each pair of inputs contains a one. For each of the first 24-bits, adding 1 will do one of three things: (1) in the case 0+0+1, 1 will be the result; (2) 1+0+1 = 0+1+1 = 0 carry 1; or (3) 1+1+1 = 1 carry 1. In the last two cases, the resulting value is identical to that of EQU. Only in the first case would the function return an incorrect value. Consequently, as long as there is not a position in the first 24 bits of the two input bit-strings that contains a zero, then the genotype will be successful at demonstrating EQU. (The upper 8-bits are set by Avida; only the highest bit contains zero in both bit-strings, not affecting the resulting calculation.) Although the probability is low, such combinations do occur, and selection is strong enough under the EQU-only reward structure to cause the implementation to persist.

Overall, 42 of 300 replicates under the POS bias demonstrated EQU statically or stochastically (24 only stochastically) along the PLoD compared to two replicates under UNB that demonstrated EQU only stochastically. Although phenotypic stochasticity appears to be important for the evolution of EQU in the EQU-only owing to its presence in the majority (13 of 18) of replicates immediately before they achieved EQU statically, the exact role that phenotypic stochasticity plays in evolution has not been completely explored and may prove a fruitful avenue of investigation, especially with regard to difficult to evolve tasks or when the relative cost of developing a task is high.

3.3 Other effects of mutation bias

One of the key assumptions about my implementation of mutation bias is that the bias reflects a type of "compositional optimum" for evolved genomes. I do not claim that the `POS` is the *most* optimum, but I believe it likely better than `UNB`.

In Dorn et al. [9], the authors present a hypothesis that a "monomer abundance distribution biosignature" (MADB) exists to distinguish biotic and abiotic conditions. In addition to using amino acid and carboxylic acid concentrations from both biotic and abiotic samples, the authors used Avida to detect the presence of a MADB biosignature. In this case, they examined the effect of selection on two different implementations of a handwritten ancestor evolving under `UNB` conditions over seven different mutation rates. The expected distribution of instructions under abiotic conditions would be, on average, equal for each instruction. The authors find, however, that irrespective of ancestor and (except at lethal levels) mutation rates, a common biosignature emerges for all treatments. The authors concede that there is some variation between the distributions caused by the ancestors because of historical contingency dealing with replication method (the two ancestors had extremely different means of replicating themselves), but for the most part instructions that were over or under-represented by one ancestor were also favored or disfavored by the other.

As a consequence of their work, Dorn et al. provide a convincing argument that evolution with unbiased mutations (as the authors used) will result in a common compositional outcome regardless of ancestor or mutation rate. Additional work by Dorn [8] also provides evidence that the mutation bias does not contribute much to the final compositional signature. This leaves the environment and its selective pressures as the drivers of genome composition.

The overall consequence of selection being the deciding factor in composition is that it implies some kind of environment-specific compositional optimality. The `POS` bias seeks to emulate this composition. In order to see how genetic compositions of `POS` and `UNB` populations replicates change over time, I measure the average Kullback-Leibler (KL)-divergence of the replicate populations from sections 3.2.1 and 3.2.2.

KL-divergence is defined as:

$$D_{KL}(P\|Q) = \sum_i P(i) log \frac{P(i)}{Q(i)} \tag{3.2}$$

where P is the distribution of instructions at a particular update for a single replicate's entire population and Q is the expected distribution of instructions of the POS or UNB. For cases where $P(i) = 0$, I assigned $P(i) log \frac{P(i)}{Q(i)}$ to be 0.

All values here are reported in bits, which means the KL-divergence of the two distributions can be thought of as the average number of extra bits needed to convey a message composed from states using a distribution of P when optimally encoded for the distribution Q. For our purposes, this value can be thought of as the average number of bits of information genomes under selection differ from the mutation bias. Figure 3.6 shows the KL-divergence from populations in sections 3.2.1 and 3.2.2.

In general, the average composition of all populations are closer to POS than UNB over time. All treatments had an average KL-distance of less than 1.5 bits under POS. In contrast, all treatments were at least 1.9 and as high as 2.2 bits different from NEG. Interestingly, switching from UNB to POS can visibly change the composition of the population. After these treatments became POS biased, their average composition began to deviate from their UNB-only counterpart, asymptotically approaching the average composition of POS relative to UNB. Despite this observation, there appears to be a gradual accumulation of artifacts in all populations, as the average composition distance of all replicates has a slight upward trend away from POS over time.

3.4 Summary

In this chapter, I created a mutation bias (POS) and demonstrated it can (1) improve adaptation under the Logic-9 reward structure; (2) allow for already adapted populations to better

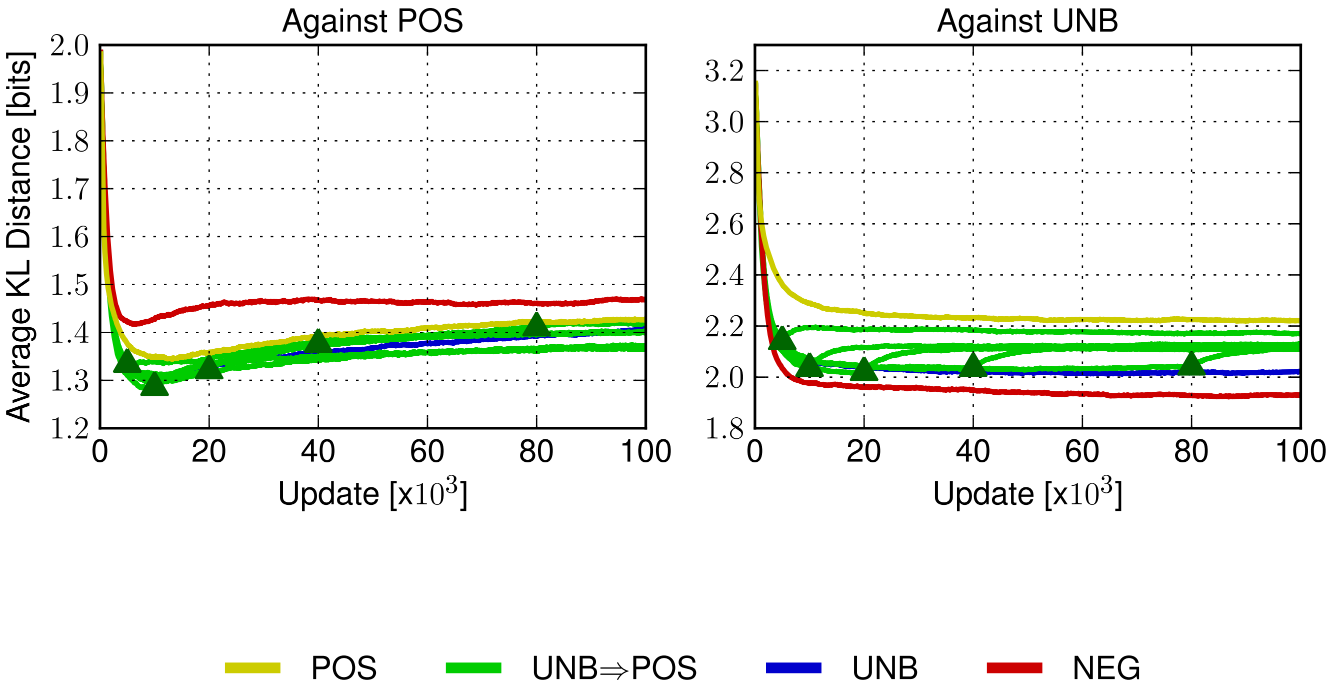

Figure 3.6. **Population-wide genetic composition changes over time relative to POS and UNB mutation biases.** The colored lines show different obligate treatments: blue=UNB only; yellow=POS only; red=NEG only; green=swap from UNB to POS. Green triangle markers show the point at which the mutation bias switched for each bias-swap treatment.

adapt to their environment than under `UNB`; (3) emulate some of the genome-composition pressure under the `Logic-9` reward structure to spontaneously produce both simple and complex tasks. I also showed that an “anti-bias” created by inverting the redundancies of `POS`, called `NEG`, can retard adaptation in the `Logic-9` reward structure. Further, I found that mutation bias can affect population-level genetic composition but that selection does drive divergence from expected compositions.

Overall, the results from this study meet the first of the three components needed for the evolution of a beneficial mutation bias: the potential to improve adaptation in a given environment. In the next chapter, I will study if selection can distinguish between two mutation biases. I will also explore how genotypes evolved under one mutation bias are affected by their history with regard to the selective advantage of a mutation bias.

CHAPTER 4

Selecting a Mutation Bias

I demonstrated the ability of mutation bias to affect the outcome of evolution but can evolution select for an optimal mutation bias? Many processes that confer long-term evolvability are at odds with short-term selective pressures experienced by individuals. Alternatively, the effects of mutation bias might not even be visible to selection, affecting only the long-term adaptive behavior of the population.

Mutation bias is not alone in having a potential disparity between short and long-term selective behavior. In terms of the disparate behavior between short and long-term selective differences, perhaps one of the most examined evolvability-conferring processes is the evolution of mutation rates. There is a tradeoff between short and long-term outcomes (reviewed by [36]). In the short-term there is an increase in the number of deleterious mutations being accumulated; over time with a sufficiently smooth fitness landscape ([5]) or a high number of accessible beneficial mutations (for example [14]), high-mutation mutants can outcompete their wildtype peers. The key to whether or not a high-mutation strain goes to fixation depends on its short-term advantage. If novel beneficial adaptations are a number of mutational steps away, a higher mutation rate may not be selectively favorable. In particular, if the selective advantage is not high enough at the individual rather than per capita level, high mutation populations with small initial abundance may be lost to drift.

Mutation biases may have a similar difficulty. Even though there is a tendency for populations to adapt better as a whole under the `POS` than the `UNB` bias, the short-term effects of the `POS` bias might not be beneficial. The short-term consequence of mutation biases described in the previous chapter are not known. In the preceding experiments, I applied mutation biases on a population as a whole. As a result, the selective effect of one bias relative to another in the short-term cannot be gauged.

Fortunately, there is reason to suspect that the POS bias from the previous chapter is at least slightly beneficial relative to the UNB bias in the short-term. This is based on two observations. The first is from the the KL-distances in Section 3.3 with averages shown in Figure 3.6. Comparing the the KL-distance using POS as a basis (Q in Equation 3.2) against UNB, the POS bias is closer to the composition of all experimental averages over all sampled times than the unbiased distribution. As such, the POS bias is consistently closer to the actual composition of organisms in the Logic-9 reward structure than UNB. Next is the observation made in Chapter 2 that researcher-specified and actual mutation rates vary because of the way mutation is implemented in Avida: sites might mutate but not change identity. If a mutation bias closely matches the genetic composition of a population, as in the case with POS relative to UNB, than the probability that an instruction will mutate to itself is maximized. Consequently, overall mutational load decreases as more synonymous mutations occur.

Despite these observations, the advantage of a beneficial bias such as POS might not be visible to selection. Small decreases in mutational load might not be large enough (especially in small populations with high levels of drift) to have any noticeable selective benefit in the short-term. Further, the long-term benefits of a mutation bias might be so decoupled from the cause – a better mutation bias relative to peers– as to be a trait that is difficult to hitchhike upon, leaving its fate to chance.

In this chapter, I will explore whether the mutation biases from the previous chapter are visible to selection by competing mutation biases against one another.

4.1 Mutation bias may be visible to selection

In order to compete different mutation biases in head to head competitions, I needed to modify Avida to handle multiple instruction sets within a single experiment and monitor which instruction sets are in use by organisms in the population.

I began mutation bias competitions with the population being fully-seeded using either the default length-50 `nop-C` ancestor (used to begin experiments in the preceding chapter) or the final dominant organism from 30 of the 900 experiments in Section 3.2.1, with 10 genotypes from each mutation bias.

I measured the competitive advantage of each mutation bias against the `UNB` bias by competing two subpopulations against each other. I seeded the test subpopulation to be a fraction, π_t, of the population. It used one of the three mutation biases used in Chapter 3: the `POS`, `UNB`, or `NEG`. This subpopulation competed against the remaining $1 - \pi_t$ of the population that I assigned to be mutationally unbiased (`UNB`). Mutation biases did not change over the course of the experiments and were inherited from parent to offspring. The experiments proceeded until the offspring of one of the subpopulations drove the other to extinction.

I repeated each competition a number of times and tabulated the frequency competitions drove `UNB` to extinction, denoted Ω_t. By looking at how Ω_t differed between treatments relative to their initial abundance, I was able to determine how selectable mutation biases were relative to `UNB` and how different parameters affect outcome. Treatments that drove `UNB` to extinction better than chance “won” competitions and were selectively advantageous. Those treatments that were driven to extinction by a `UNB` subpopulation worse than chance “lost”. Treatments where outcome was indistinguishable from chance were “tied”, where neither bias had an advantage over the other.

Figure 4.1 displays the full range of π_t and Ω_t are along the x and y-axes, respectively. The neutral expectation, where $\pi_t = \Omega_t$, is displayed with caricatured error bars. Outcomes with Ω_t above the neutral expectation show better-than-chance outcome *relative to its competitor*, indicating its mutation bias is selectively-advantageous compared to its an unbiased competitor. Outcomes below the neutral expectation indicate relatively negative selection. Every set of treatments includes a control where UNB is competed against itself. Any significant variation from the neutral expectation in these controls would indicate something is amiss with the implementation of the experiment. I found no significant deviation in the control experiments outside of chance.

For the remainder of this chapter, I will use the experimental design above to examine whether or not mutation bias is opaque to selection relative to another bias using both the default ancestor and organisms evolved under the three mutation biases from the previous chapter. I will also examine what the effects of population size are on outcome (Section 4.1.2); the effects of population structure on outcome (Section 4.1.3); and if there are any differences in the success of mutation biases between evolved and unevolved organisms (Section 4.2). Finally, I will begin to explore why mutation biases are selected (Section 4.3), a topic that will be continued in Chapter 5.

4.1.1 Baseline competition results

In this section I describe baseline competition results. This baseline will be used to contrast how changes in population size and structure affect the selective advantage of mutation biases. For the baseline parameters, I used an unstructured (mass action) population size with a maximum size of of 3600 organisms. Later I will increase the population size to 10,000 and introduce structure in the population by restricting offspring placement. Other experimental parameters were identical to those of experiments performed in Section 3.2.1.

In order to see if the two non-uniform biases in the previous chapter (POS and NEG) are subject to selection, I competed them against mutationally-unbiased (UNB) organisms. Each competition began with a fully-seeded population of the default nop-C ancestor. I assigned a fraction of the population, π_t, to be either POS, NEG, or UNB (the latter serving as a control).

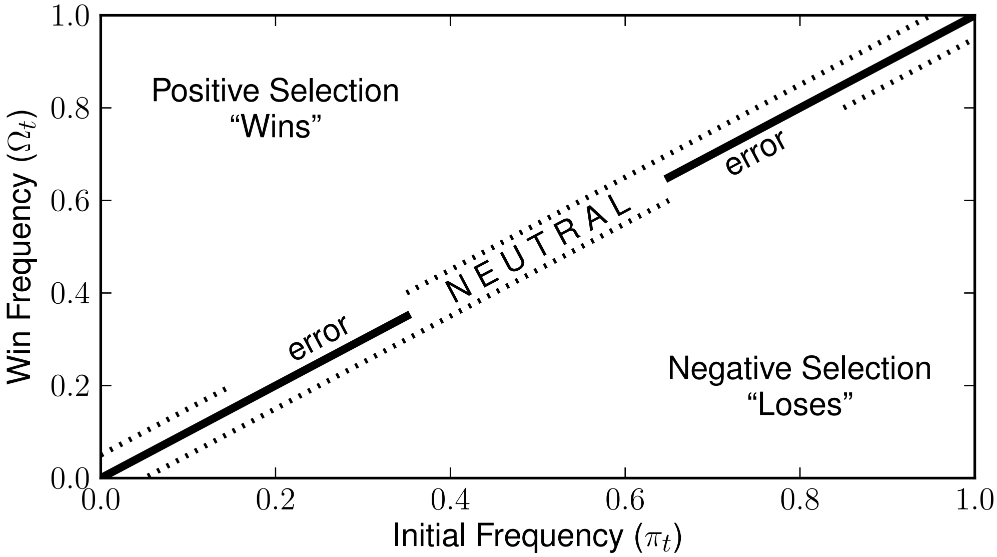

Figure 4.1. **Baseline competition results.** In the absence of selective pressure, competition outcome will be based solely upon the initial frequency that a particular bias has in the population. Results that fall above the diagonal line show evidence of positive selection (“wins”) relative to a peer subpopulation; outcomes below the line show evidence for a selective disadvantage (“losses”) relative to a peer subpopulation.

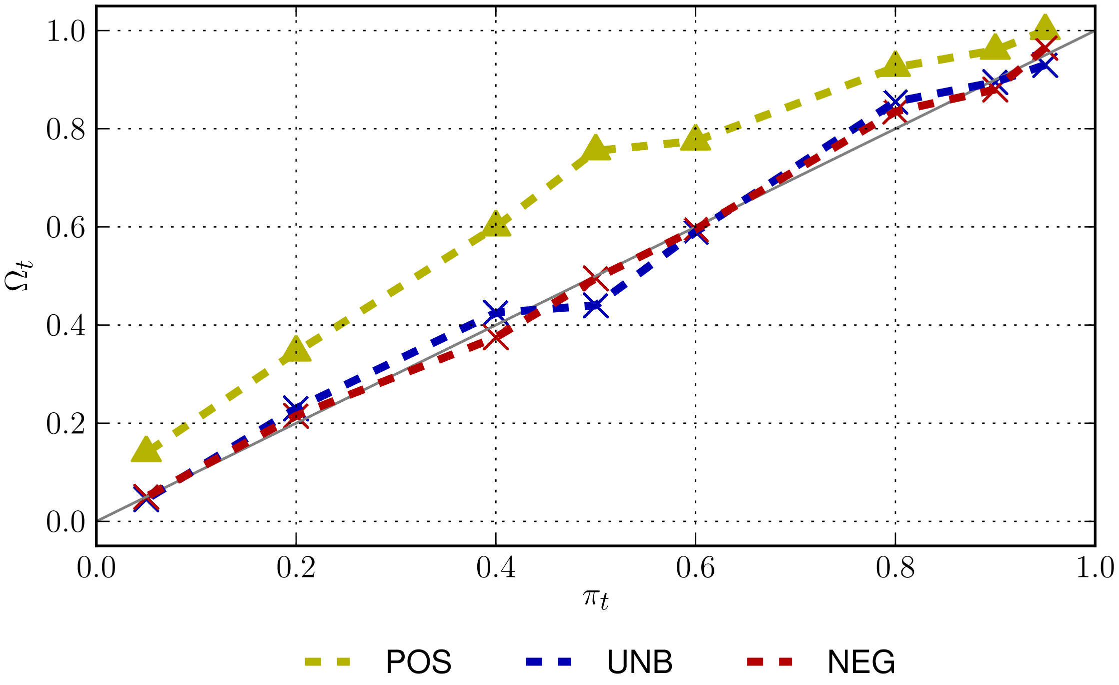

Figure 4.2. **Competition against UNB in an unstructured population.** Initial competitor abundances (π_t) are shown against competitive success (Ω_t). Yellow, blue, and red lines indicate competitions between `UNB` and `POS`, `UNB`, and `NEG`, respectively. Treatments significantly different than chance ($p < 0.05$, 2-tailed Binomial Test) are shown by triangle markers; competitions not different than chance are shown by crosses. Dashed lines connecting treatments are shown to contrast outcome against the neutral expectation.

I assigned the remaining $(1 - \pi_t)$ of the population to be mutationally unbiased (`UNB`). For this set of experiments, I tested π_t at $0.05,\ 0.10,\ 0.20,\ 0.40,\ 0.50,\ 0.60,\ 0.80,$ and 0.95.

The location of an organism in its toroidal world determined whether it was assigned to have either the test bias or be part of the `UNB` competitor. Both subpopulations were grouped together. I repeated each competition 200 times, with experiments ending when the offspring of only one of the the two competitors remain in the population.

In general, `POS` won competitions under all initial abundances (π_t) and `NEG` tied with `UNB` (Figure 4.2). Of the assayed values of π_t, all `POS` had significantly more populations outcompeting their `UNB` peers than chance ($p < 0.05$, 2-tailed Binomial Test, Bonferroni corrected). All NEG and UNB versus UNB competitions had outcomes not significantly different than chance for all values of π_t. These results show that the experimental design is valid (as UNB versus UNB tied) and, more importantly, selection cannot distinguish between NEG and UNB.

The observation that POS won against UNB with initial population sizes as small as 180 of 3600 is encouraging, as anytime a new mutation bias-producing phenotype arises, it will be extraordinarily rare and subject to the effects of drift in finite populations. In the following chapter I explore the ability of mutation bias to arise at low frequency and come to dominate the population.

The fact that NEG is indistinguishable from UNB despite their divergent long-term behavior is surprising. There are at least a couple of reasons why this might be so: UNB and NEG might have similar inflow rates and magnitudes of beneficial mutations under these experimental conditions; it could also be that the rates of beneficial mutations differ, but are both so low that selection cannot distinguish between them at the population size being evaluated.

4.1.2 The effect of population size on the relative selectability of mutation bias

An increase in the population size relative to the baseline competitions should decrease the effects of genetic drift and increase the absolute number of genotypes evaluated, leading to the discovery of more beneficial mutations. Further, increased population sizes also allow for a finer distinction between genotypes with close fitness values. As mentioned above, if NEG and UNB have different rates or distributions of beneficial mutations, but both are really low, the difference would be indistinguishable at small population sizes. So, if competition outcomes differ between differently-sized populations, it would provide evidence that inflow rates or distributions of beneficial mutations differ between NEG and UNB populations.

The dashed-lines in Figure 4.3 shows POS and NEG competitions against UNB under unstructured (mass action) conditions. The darker lines come from treatments with a population size of 10,000; the lighter lines are simply the outcome of the experiments in the previous section. Significant differences in outcome between 10,000 and 3600 population sizes are shown by triangle markers ($p < 0.05$, 1-tailed Fisher's Exact Test); crosses show differences where population size had no significant difference on competition outcome.

Increasing the population size did not allow selection to distinguish NEG from UNB. For all values of π_t, NEG and UNB tied, making the results no different from the smaller population baseline. These results again suggest that NEG and UNB biases are similar in competitive ability.

The larger population sizes did allow POS to win more often for intermediate values of π_t (=0.20, 0.40, 0.60) relative to the baseline. For ranges of π_t closer to either extreme, POS organisms did not win significantly more often. The lack of improvement at small values of π_t suggests that the effects of drift were negligible on the experiments from the preceding section. If drift were a significant factor in outcome, higher initial abundances in the larger population size should allow for significant increase in success (Ω_t).

The lack of significant differences for high values of π_t under POS could be due to the effects of saturation, especially with the large values of π_t where Ω_t exceeds 0.90 for both population sizes when $\pi_t \geq 0.80$. These treatments had little room for improvement, so significant differences were difficult to obtain. For example, for the POS treatment with $\pi_t = 0.80$ at a population size of 3600, Ω_t was 0.925 (185/200 treatments). In order for there to have been a significant improvement, Ω_t would have to have reached 0.970 (194/200 treatments), so almost complete success of POS over UNB. Mid-range values of π_t, which competed POS against UNB on a more even footing (since neither POS or UNB had an advantage due to initial abundance), showed significant difference as one would expect.

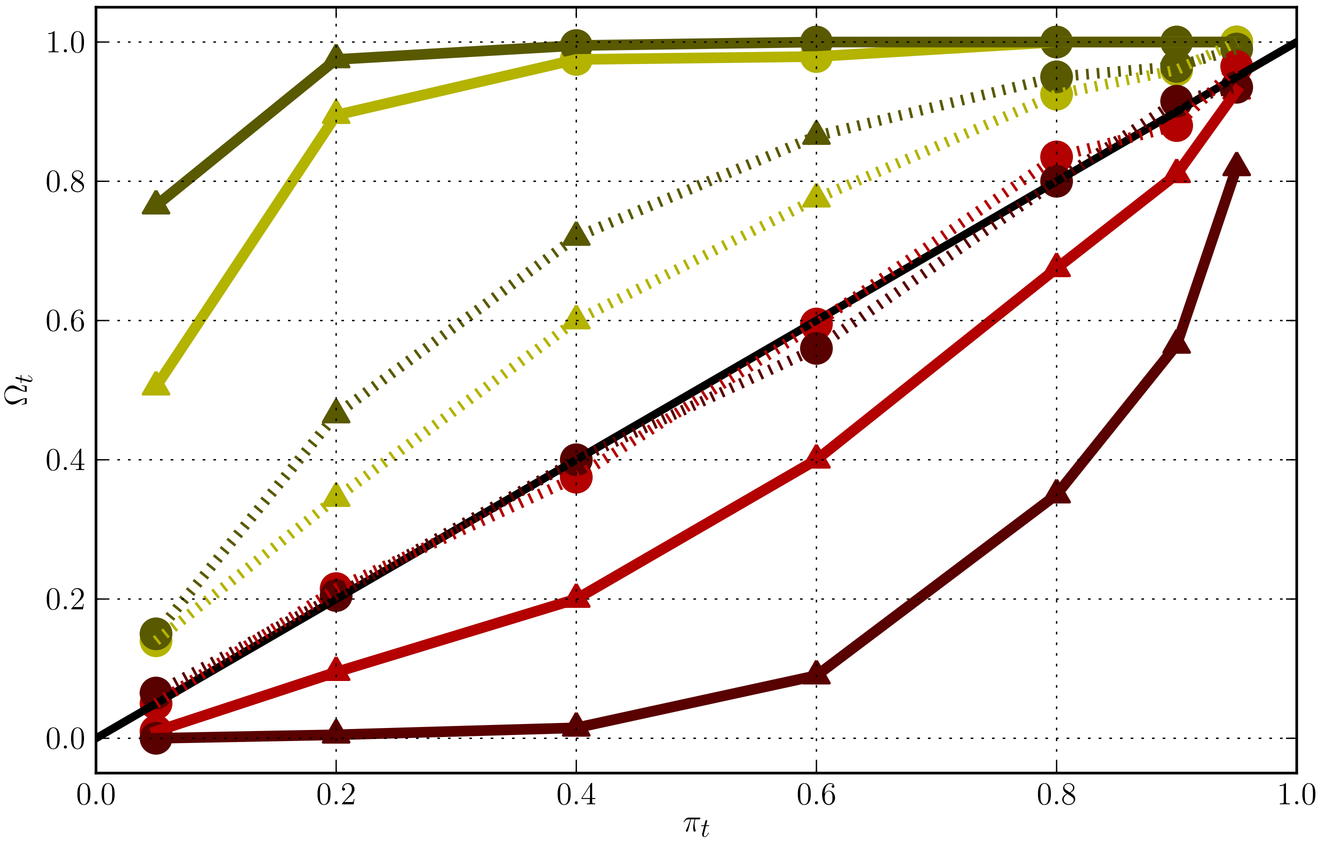

Figure 4.3. **Mutation bias competitions with different population sizes.** Initial competition seed frequency (π_t) is plotted against the fraction of competitors that drove `UNB` competitors to extinction (Ω_t). `POS` competitions are in yellow; `NEG` are in red. Lighter colored lines competed in a population of size 3600 whereas darker colored lines competed in a population of size $10,000$. Unstructured (mass action) and structured (local replacement) populations are distinguished by dashed and solid lines, respectively. Competition results that significantly differ from chance ($p < 0.05$, 2-tailed Fisher's Exact Test) between population sizes for the same replication method are shown by a triangle; insignificant results are shown by a circle.

4.1.3 Population structure is an important factor in the selection of mutation biases.

In this section, I vary the placement of offspring to provide structure to populations. In the baseline competitions above, offspring were placed randomly in the population creating a structureless world. In the experiments below, I require offsprings to be placed in the local neighborhood of cells around their parent. This change creates a structured population as colonies can only expand at their edges. If bias requires mutations to interact, then structured populations will give competing populations time to accumulate them.

If a rare or highly beneficial mutation arises in an offspring in a population without structure, it will spread at an exponential rate. Consequently, it is difficult for multiple deeply-divergent clades to co-exist for relatively long periods of time in unstructured populations when sparse or uncommon highly-beneficial mutations predominate and sweep to fixation at exponential rates. This lack of structure makes the genotypes in the population relatively homogenous as the recent common ancestor of the population isn't very distant from extant genotypes.

Under local replacement offspring are placed in the cells adjacent to their parents, creating a structured population. This type of population structure reduces the rate at which a beneficial mutation sweeps the population as a sweeping clade can grow only at the boundary between it and its peers. This change in replication method should allow for more mutations to accumulate in each competing clade, creating the possibility for mutations to interact more so than under random offspring placement. Consequently, if local placement improves outcome, then the increase of success for `POS` over `UNB` or `UNB` over `NEG` might be the result of an increase in epistasis between mutations. If there is no change, then it is possible that there is simply a set of independent beneficial mutations that cause `POS` to win over `UNB` and `NEG` to be selectively identical to `POS`.

Figure 4.4 compares the outcomes of competition in 3600-sized populations under both structured (dashed lines) and unstructured (solid lines) populations. I repeated each competition treatment 200 times and used the default `nop-c` ancestor under the same experimental conditions as in the previous section. With the exception of the treatment where $\pi_t = 0.95$ (due to saturation, both structured and unstructured treatments had $\Omega_t = 1.0$), structured population significantly improved the competition outcome in favor of `POS` relative to `UNB` ($p < 0.05$, 1-tailed Fisher's Exact Test, Bonferroni corrected).

`NEG` treatments, on the other hand, performed significantly worse in structured rather than unstructured populations relative to `UNB`. With the exception of the case where $\pi_t = 0.95$, all other treatments show significantly lower Ω_t ($p < 0.05$, 1-tailed Fisher's Exact Test, uncorrected) relative to their population-wide placement counterparts. `NEG` now performed significantly worse than chance for the first time over all but the smallest initial abundance assessed. Like `POS`, I believe it is likely that `UNB` also had a relative advantage to its `NEG` competitor because of better increased ability to accumulate multiple mutations to create a selectively-beneficial effect in a structured population.

These results suggest that population structure plays an important role in how selectable a mutation bias is by virtue of allowing mutations to accumulate. Well-mixed populations experience faster selective sweeps when a beneficial mutation arises; structured populations slow the progress of the sweep. By allowing mutations to accumulate in a structured population, the effects of mutation biases are more pronounced and, therefore, more distinguishable.

Relatively speaking, the effects of roughly tripling the population size from 3600 to 10,000 organisms has less of an influence on Ω_t than changing the structure of the population. Figure 4.3 shows the difference in competitive outcome for all four replication method × population size treatments for `POS` and `NEG` versus `UNB` competitions. Neighborhood replication improves competitive outcome for `POS` by making it almost a certainty that `POS` will outcompete `UNB` for initial abundances greater than 0.40. Switching to a larger population size improves outcome significantly for treatments with lower values of π_t ($p \ll 0.05$, 1-tailed Fisher's Exact Test, Bonferroni corrected). At the lowest initial abundance examined ($\pi_t = 0.05$),

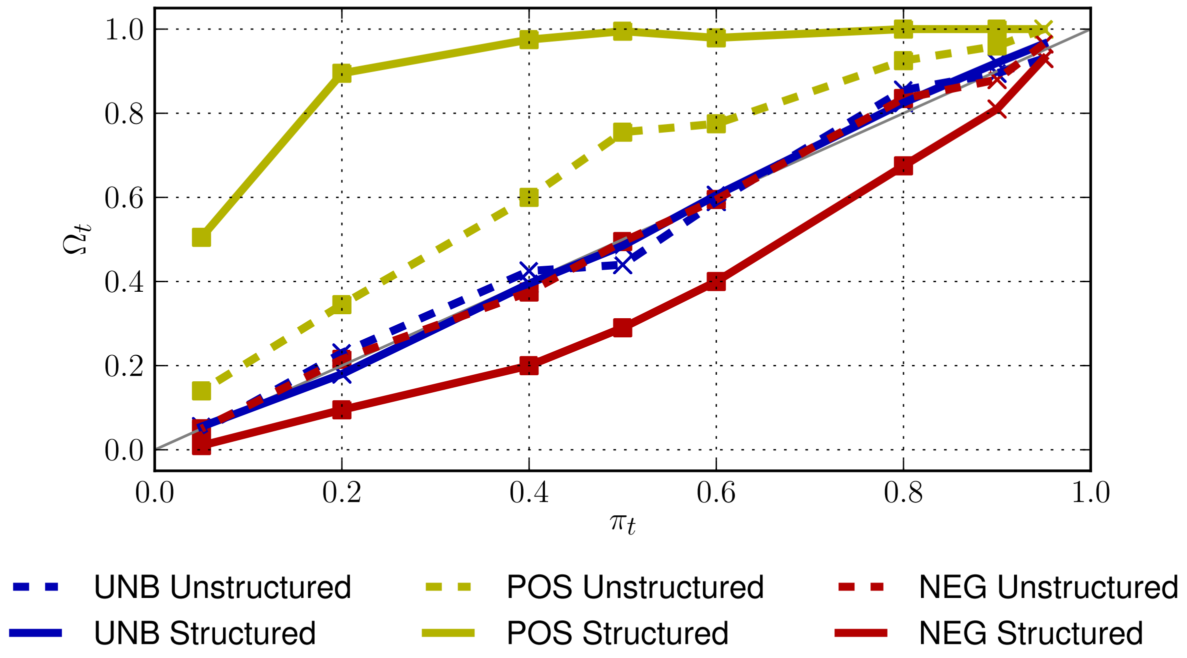

Figure 4.4. **Competitions with different population structure.** Each line above shows the outcome a series of competition experiments over initial abundance (π_t) of mutation biases with yellow, blue, and red coloring showing competition between `POS`, `UNB`, and `NEG` biases against `UNB`, respectively. Experiments with outcomes significantly different between structured (local replacement) and unstructured (mass action) populations ($p < 0.05$, 2-tailed Fisher's Exact Test) are shown by square markers; insignificant results are shown by crosses.

there is improvement to be gained by increasing the population size when using a structured populations but not when using unstructured populations. This difference suggests that structured populations are capable of amplifying the effects of a particular mutation bias even at low initial abundances, an effect not shown by simply increasing the population size alone.

4.2 Does adaptation influence the selection of a mutation bias?

The length-50 nop-C-backbone ancestor genotype used to seed all of the previous experiments allows for selection to distinguish between mutation biases, especially in structured populations that use a local offspring placement policy. However, the default ancestor's nop-c backbone has a lot like a "blank tape" for mutations to accumulate upon without significantly hurting fitness of the organism. Evolved genotypes, in contrast, may have their fitnesses more affected by mutation than this unevolved genotype. Consequently, mutation bias may not be as subject to selection with evolved genotypes as additional restraints such as increased epistasis are present in evolved organisms relative to the default ancestor.

It is important that mutation biases be distinguishable in well-adapted genotypes. Natural systems are inherently well-adapted to their environment, so a mutation bias must be selectable in combination with evolved genotypes for it to be evolutionarily meaningful in nature.

In this section, I will compete 30 of the final dominant genotypes from experiments in Section 3.2.1 with 10 from each of the mutation biases used in that experiment. The genotypes selected all were from runs that obtained EQU along the PLoD. I examined whether (1) mutation biases were still visible to selection if evolved ancestors were used to seed competitions; (2) if there were any changes in outcome relative to competitions with the ancestor genotype; and (3) if the mutational history of the evolved genotypes influenced competition outcome.

4.2.1 Mutation bias is still selectable with an evolved genotype.

Figure 4.5 shows competitive outcome of a single evolved genotype taken from the end of one of the replicates with an unbiased mutational background to the default ancestor baseline

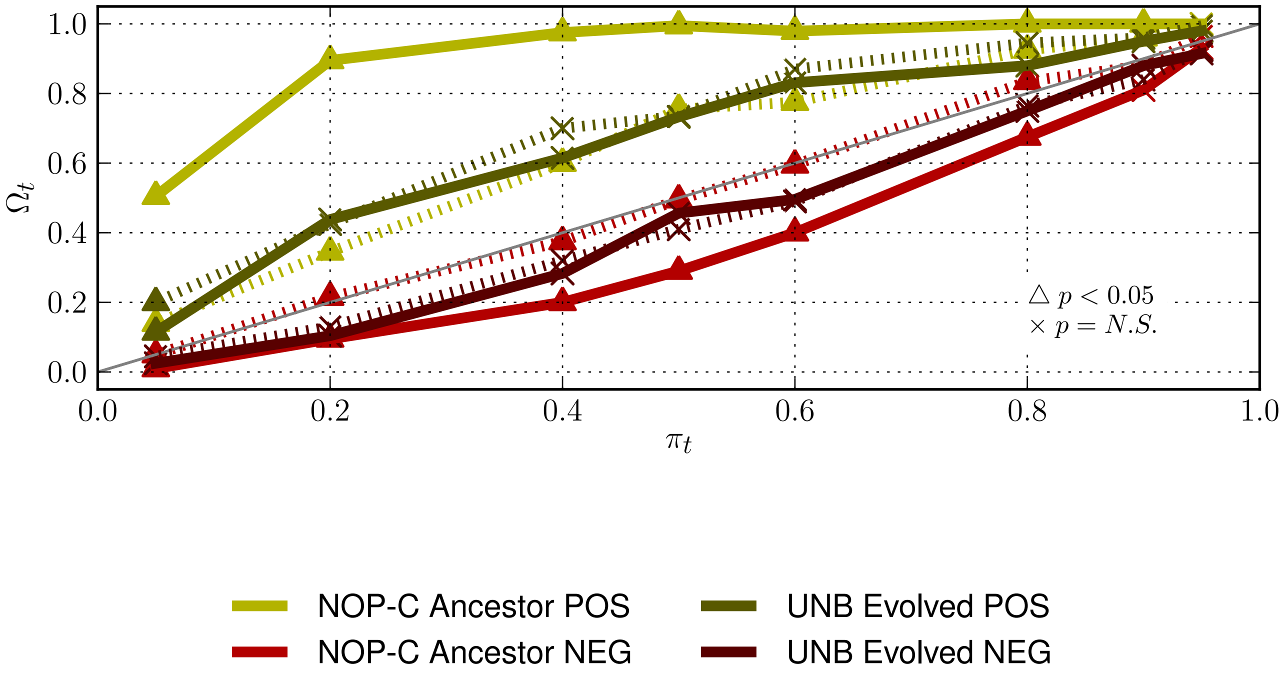

Figure 4.5. **Unevolved and evolved genotype bias competition results.** Colored lines show the results of competition of `POS`, `NEG` (yellow and red, respectively) against `UNB`. Darker colored lines show competitions using a genotype evolved under `UNB`. Lighter colored lines show results from competitions using the default ancestor. Solid lines show competitions that were structured; dashed lines show competitions which were unstructured. Triangle markers show competitions where population structure significantly altered outcome ($p < 0.05$, 2-tailed Fisher's Exact Test); square markers show competitions where population structure did not change outcome.

competitions from Section 3.2.1. As with previous competitions, I competed POS, NEG, and UNB subpopulations of this genotype against UNB over a range of initial abundances in both structured (local replacement) and unstructured (mass action) populations.

In general, the POS significantly outperformed UNB in both structured and unstructured populations, with the exception of the largest initial abundance of POS ($\pi_t = 0.95$) in structured population. The unstructured treatments performed significantly better than chance for all values of π_t examined ($p < 0.05$, 2-tailed Binomial Test, Bonferroni corrected). However, outcomes from the structured "neighborhood" population are less clear. Only the smallest initial abundances ($\pi_t < 0.60$) were still significant ($p < 0.05$) after correction for multiple comparisons.

NEG competitions against UNB were even more difficult to compare under both structured and unstructured populations. Unstructured treatments were not significantly different from chance at $\pi_t = 0.95$ and 0.80 but either marginally or significantly different from a neutral outcome at other values of π_t. It is a similar case for NEG competitions in structured populations where outcomes were not different than chance for $\pi_t = 0.90, 0.80, 0.50,$ and 0.05 but either marginally or significantly different from chance for other values of π_t. Clearly, the strength of selection against NEG-biased genotypes is at best marginal against UNB at small initial abundances with this particular evolved genotype.

Comparing the evolved and unevolved genotypes, there was little difference in outcomes in unstructured populations for both POS and NEG competitions against UNB. There were a couple treatments that show marginal differences between evolved and unevolved genotypes in the POS competitions (at $\pi_t = 0.60$ and 0.40) and the NEG competitions (at $\pi_t = 0.60$ and 0.20) but any significance disappears after a Bonferroni correction for multiple comparisons.

Perhaps the most striking difference between competitions between evolved and unevolved genotypes takes place in structured populations. Earlier, I showed that population structure created a large difference in competitive outcome using the unevolved ancestor, likely due to an increase in mutation interactions. With this evolved genotype, POS does not win significantly more often in structured or unstructured populations except at the smallest initial subpopulation size ($\pi_t = 0.05$). Similarly, NEG performed no differently in structured versus unstructured populations, tying with UNB. This would seem to indicate that a structured population (and resulting mutation accumulation) is not sufficient to allow these two biases to significantly alter competitive outcome relative to unstructured populations, alluding to possible differences in the local landscape between evolved and unevolved genotypes.

4.2.2 Mutation bias is selectable using additional evolved genotypes.

To get a more general understanding of the difference between evolved and unevolved genotype competitions, I competed each of the 30 final dominant genotypes using the same configuration as previous experiments. However, instead of assessing outcome over a range of π_t, I evenly split all experiments between the test bias and UNB. As before, experiments ran until one bias drove the other to extinction. I performed all experiments in populations with a capacity of 3600 organisms beginning fully seeded with the test genotype. I tested both structured (local replacement) and unstructured (mass action) populations. I replicated each competition 100 times.

Figure 4.6 shows the outcomes of these competitions relative to a neutral expectation and the unevolved nop-c ancestor competitions. In general, the outcomes were similar to the ancestor: POS outcompeted UNB significantly more often than chance in both structured and unstructured populations. Of the 30 genotypes tested, 27 outcompeted UNB using POS more often than chance ($p < 0.05$, 1-tailed Binomial Test, Bonferroni corrected) in both structured and unstructured populations. However, the non-competitive genotypes were not the same between the structured and unstructured competitions as only one evolved genotype was not competitive under both conditions.

The pattern for NEG competition was not as clear. Of the 30 genotypes tested, 17 of them

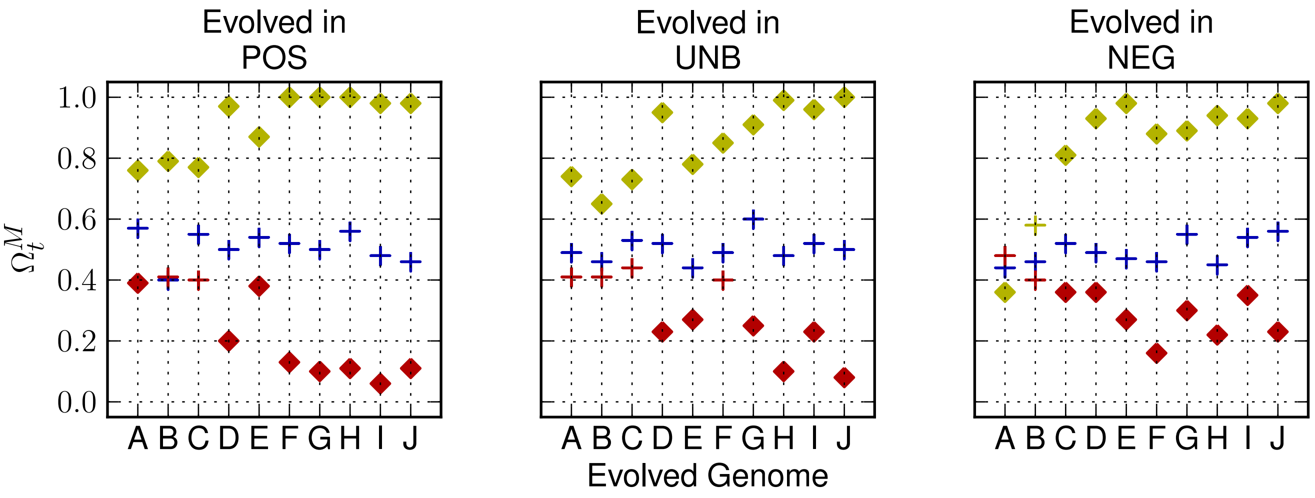

Figure 4.6. **Competition results using evolved genotypes as compared to neutral outcome.** 30 final dominant genotypes were taken from experiments in Section 3.2.1, with 10 from each mutation bias background. Yellow, blue, and red markers show the outcome of 100 competitions against `UNB` for `POS`, `UNB`, and `NEG`, respectively. Each treatment began with the population being evenly divided between the test bias and `UNB`. Significant wins or losses are shown by diamond markers ($p < 0.05$, 2-tailed Binomial Test); ties are shown by a plus marker.

significantly differed from chance outcomes in unstructured populations ($p < 0.05$, 2-tailed Binomial Test, Bonferroni corrected). Of those that were not significant under a multiple comparison correction, 8 were never significant and the other 5 were significant prior to the correction, with 3 almost meeting the cutoff criteria. Outcome in structured populations was similar but slightly improved: only 8 of the 30 genotypes did not outcompete `UNB` significantly better than chance, with 2 being close to the cutoff criteria after correction.

One of the key differences between structured and unstructured populations when competing `nop-c` genotypes using different biases was the improved outcome of structured populations relative to unstructured ones. This difference is not present for all of the evolved genotypes and is shown in Figure 4.7. Indeed, with the `POS` competitions, only 1 of the 30 evolved genotypes showed significant differences between structured and unstructured populations, after correcting for multiple comparisons ($p < 0.05$, 2-tailed Fisher's Exact Test, Bonferroni corrected). Without this correction, a total of 5 of the 30 genotypes showed outcomes statistically different between structured and unstructured populations. The strength of the difference in outcomes between evolved and the unevolved genotype is quite telling. The `nop-c` ancestor competitions improved significantly from 75% wins (151/200) to almost 100% (199/200) when structured populations were used. The single evolved genotype significantly improved from 87% (87/100) to 99% (99/100). These results clearly indicate that for `POS` competitions against `UNB`, population structure does not play as great of a role in determining selective outcome using evolved genotypes as it does with the unevolved genotype.

Comparing `NEG` competitions against `UNB` in both structured and unstructured populations results in a similar difference. None of the 30 genotype competitions are significantly different between structured and unstructured populations after Bonferroni correction. Compared to the `nop-C` unevolved ancestor, which showed a 67% (108 to 80/200) reduction in the ability of `NEG` to compete against `UNB`, the worst loss for evolved genotypes was only 9% (11 to 9/100).

Overall, the difference in outcomes between evolved and unevolved genotypes with respect to population structure may explain why a particular bias outcompetes another. If mutation accumulation is important, then one would expect a difference between structured and unstructured populations. Evolved genotypes simply do not have the "blank tape" to accumulate mutations that the `nop-c` ancestor possesses. Epistasis between sites and adaptations might simply disrupt the process of mutation accumulation necessary for a bias to get a boost from slower sweep times.

4.3 Why are mutation biases selected?

It may be taken for granted that the potential benefit of a mutation bias is the product of an increased probability or larger selective effect for beneficial mutations. However, modulation of beneficial mutations are not the only route for a mutation bias to be selectively advantageous.

For a mutation bias to be positively selected, it must (relative to a competitor) lower the lethal and deleterious loads by increasing either the probability that a mutation is neutral or beneficial or by decreasing the negative effect of deleterious mutations. This reduction allows for the possibility that an increase in the probability of neutral mutations is sufficient to allow for selection of mutation bias. In this section, I examine whether or not mutation bias can be selected in environments where positive mutations are disallowed. If mutation bias can still be selected, then beneficial mutations are not the only route by which a mutation bias can come to dominate a population.

Avida allows the researcher to "turn-off" beneficial mutations by examining the behavior of a mutant and sterilizing any offspring that shows improvement over its parent. This has a practical effect of increasing the lethal load experienced by a population, as offspring would be unable to replicate if they have beneficial mutations. In order for such populations to thrive, the number of beneficial mutations accessible to the population must be driven down

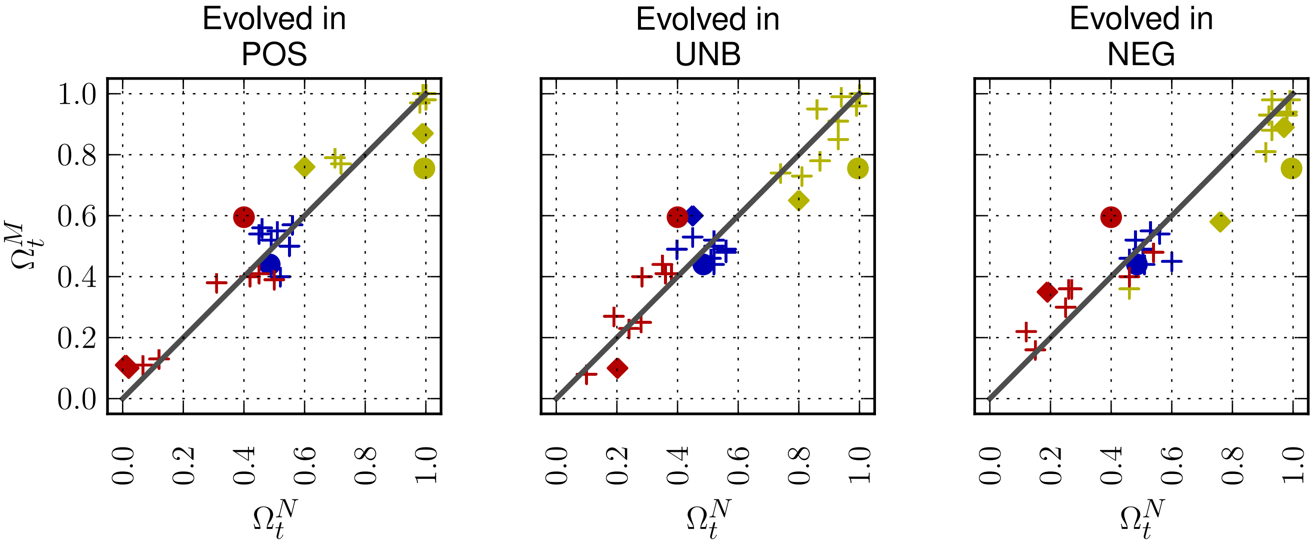

Figure 4.7. **Competition results of evolved genotypes under different population structures.** Each plot shows the outcomes of competitions for a genotypes evolved in a particular mutation bias. The color of the markers in each plot identify the bias being competed against UNB: yellow, POS; blue, UNB; red, NEG. Genotypes where population structure mattered are shown with a diamond marker ($p < 0.05$, 2-tailed Fisher's Exact Test); unevolved genotypes are shown with a circle and are significant for differences in outcome between population structures for POS and NEG competitions.

relative to neutral and deleterious mutations.

Evolved genotypes should have relatively few beneficial mutations compared to the unevolved ancestor. They also are more representative of the behavior of extant, well-adapted natural systems. As these genotypes are the product of hundreds of generations of evolution, they are well-adapted to the environment. Consequently, converting beneficial mutations accessible to these evolved genotypes into effectively lethal changes did not hurt the population too much. This allowed me to see if mutation biases can be selected based exclusively upon their effect on a change in neutral and deleterious mutational loads.

The use of evolved genotypes also allowed for another avenue of questioning. Genotypes that evolved under a particular mutation bias might be adapted to that bias in addition to their environment. If this dynamic is occurring, then genotypes competing with their native bias should perform better than genotypes competing with a non-native bias.

4.3.1 Mutation bias can be selected in the absence of beneficial mutation.

As before, I competed the 30 evolved genotypes from Section 3.2.1 in one of the three mutation biases against `UNB` at an initial frequency of $\pi_t = 0.50$. I performed each competition with beneficial mutations on and off, with 250 replicates for each treatment. Figure 4.3.1 shows the results of these competitions with respect to a neutral outcome and compares the difference between treatments where beneficials are or are not allowed for the same genotype.

In the absence of beneficial mutations, only one genotype was not able to compete against `UNB` under a `POS` bias in a manner significantly different than chance ($p < 0.05$, 2-tailed Binomial Test, Bonferroni corrected); only seven genotypes, all evolved in the `NEG` bias, competed as well as chance when competing in the `NEG` versus `UNB` competition. This result compares quite favorably with a repeat of competitions with beneficial mutations where one genotype of the 30 evolved genotypes did not perform differently from chance in `POS` competitions and seven genotypes performed about as well as chance competing in `NEG`. These results

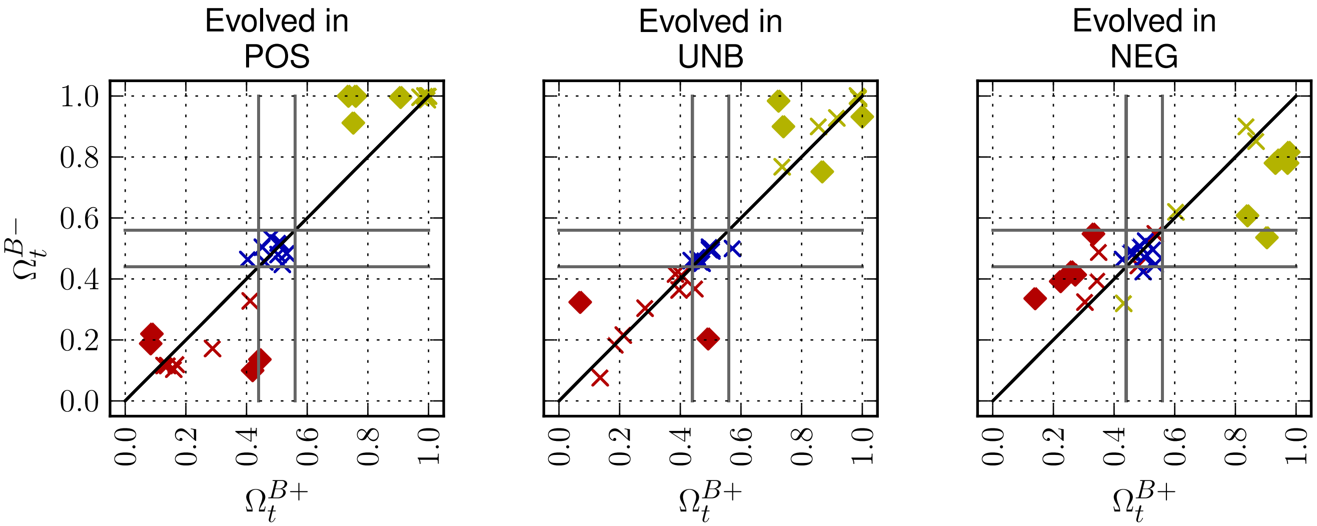

Figure 4.8. **Mutation bias competitions of evolved ancestors in the presence and absence of beneficial mutations.** Each plot distinguishes competitions for genotypes evolved in a particular mutation bias background. The color of the markers indicate the bias competed against `UNB`: yellow, `POS`; blue, `UNB`; red, `NEG`. Horizontal axes give the frequency of successful competitions against `UNB` in environments with beneficial mutations present; vertical axes show results from environments where beneficial mutants are sterilized. Points falling within the vertical and horizontal bars about the middle of the plot are not significantly different than a chance outcome for competitions with beneficial and no beneficial mutations, respectively ($p < 0.05$, 2-tailed Binomial Test, N=100). Diamond markers indicate where there is a significant difference between outcomes when beneficial mutations are or are not present for a single genotype ($p < 0.05$, 2-tailed Fisher's Exact Test, uncorrected.).

strongly indicate that there is not much difference in the number of genotypes that perform differently than chance when beneficial mutations are disallowed in evolved genotype competitions. Therefore selection can distinguish mutation biases without beneficial mutations.

There is some suggestion that the lack of beneficial mutations does not disrupt selection from distinguishing between biases when comparing individual genotypes between treatments where beneficial mutations are either allowed or sterilized. In `POS` competitions, 14 of the 30 genotypes showed significant differences in outcomes between competitions with and without beneficial mutations ($p < 0.05$, 2-tailed Fisher's Exact Test, Bonferroni corrected). Of those 14, 6 showed significantly *improved* outcomes when beneficial mutations are sterilized ($p < 0.05$, 1-tailed Fisher's Exact Test, Bonferroni corrected). `NEG` competitions fared similarly. 11 of the 30 genotypes showed differences between competitions with and without beneficial mutations, and 8 of those 11 showed significantly improved outcomes when beneficial mutations are sterilized ($p < 0.05$, 1-tailed Fisher's Exact Test, Bonferroni corrected), especially in genotypes evolved in the `NEG` bias, which account for 5 of those 8.

The results above not only indicate that selection is able to distinguish between mutation bias in competitions of evolved genotypes in the majority of cases, but also that competition outcomes can be improved if beneficial mutations are not present. There are two possible reasons this might be the case: there could be an increase in the availability of neutral mutations or the effects and abundance of deleterious mutations could be lower. In any case, this shows that it is not necessary for mutation biases to hitchhike on beneficial mutations to allow selection to choose between them.

4.3.2 Mutation bias selection can be historically-contingent.

Although one mutation bias might be selected over another due to a change in the prevalence and distribution of fitness effect for deleterious mutations, a change in the probability of neutral mutations might be sufficient, especially given the implementation of mutations in Avida. As mentioned previously, the declared and true mutation rate in Avida are distinct.

During a mutation event, it is possible for a site to be substituted with the instruction presently at that site. Consequently, high frequency instructions in the genome that have a high probability of being mutated to would cause the rate of synonymous substitutions to increase, lowering the effective mutation rate.

One way to observe this behavior is to see whether or not genotypes evolved in and competed with the same mutation bias outperform peers that were evolved under another bias in an environment absent of beneficial mutations. Figure 4.9 shows the distribution of outcomes for competitions, separating out genotypes that evolved under the same conditions as the competition bias from those that did not. When beneficial mutations are not present, there is a significant difference in the distribution of outcomes between native and non-native competitors for both `POS` ($p < 0.001$, 1-tailed Mann-Whitney U-Test) or `NEG` competitions ($p < 0.001$). Competitions with beneficial mutations available did not show any significant difference in outcomes of `POS` competitions ($p = 0.052$) and `NEG` competitions ($p = 0.223$) between native and non-native genotypes.

These results suggest that genotypes that evolved in a particular mutation bias compete better in that bias when beneficial mutations are not available. Competition results are almost indistinguishable when beneficial mutations are available. Additionally, these results suggest that a neutrality (in the form of a lower effective mutation rate) may play an important role in mutation bias selection when beneficial mutations are not present, as genotypes that are closer to the bias they compete within may perform better.

One way to examine whether or not neutrality via synonymous substitution is a factor in determining outcome is to compare the KL distance of evolved genotypes against the bias they compete in. If the case that genomes are closer in identity to the bias they compete in influences outcome, there should be a correlation between KL distance and outcome. Figure 4.10 compares outcomes to KL distance (defined in Equation 3.2).

Only `NEG` competitions in an environment absent of beneficial mutations show a correlation that would support the hypothesis that genotypes closer in identity to their competition

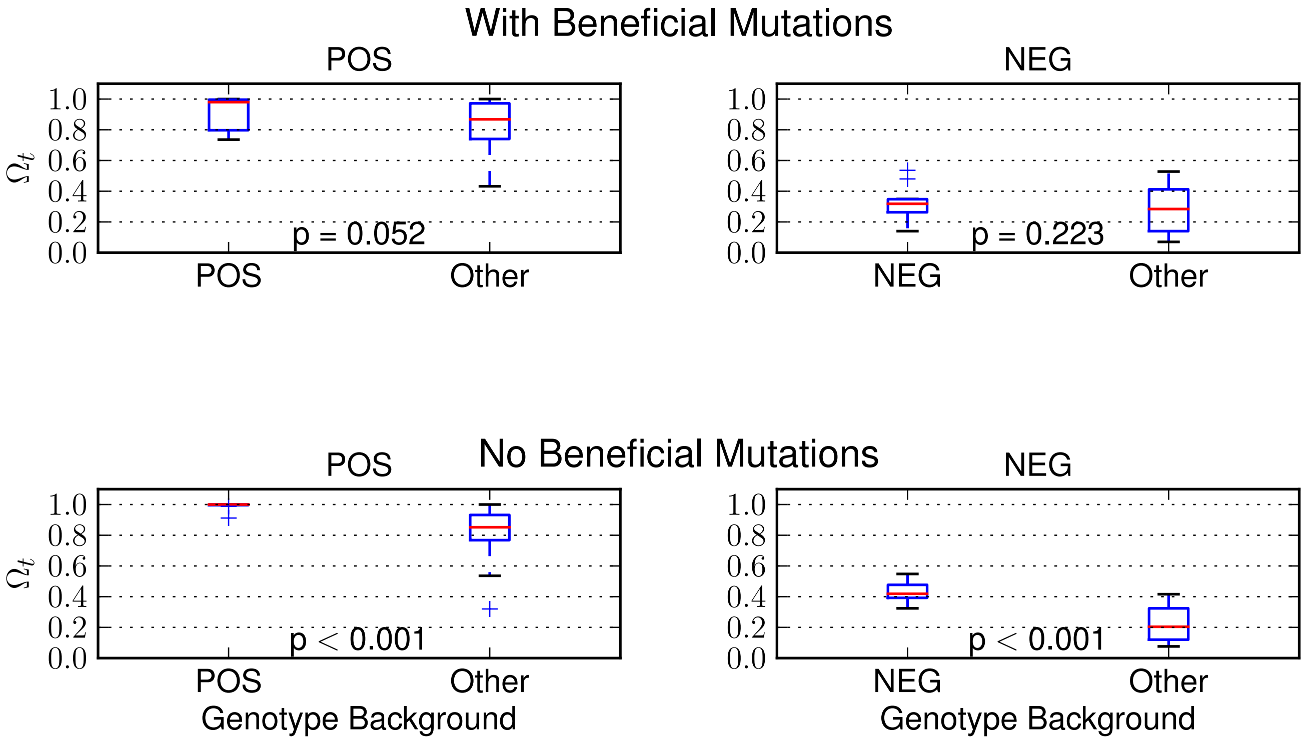

Figure 4.9. **Comparison of competition winners against mutation bias background of evolved seed genotypes.** p-values shown are from 1-tailed Mann-Whitney U-Tests of same-as-competition background genotypes competition wins compared to wins of evolved genotypes from other mutation bias backgrounds.

bias outperform those that do not. In this case, there is a significant negative correlation ($r = -0.789, p < 0.001$) between KL distance and competition outcome. A similar, though weaker trend is present when beneficial mutations are present in POS competitions ($r = -0.511, p = -0.004$), but the presence of beneficial mutations does not test neutrality alone. There are no significant correlations between outcomes and KL-distance for experiments where beneficial mutations are not present and POS is being competed against UNB ($r = -0.199$, $p = 0.291$) or where beneficial mutations are present and NEG is being competed against UNB ($r = 0.163$, $p = 0.390$).

The results above show that genotypes competing in a NEG environment have a strong correlation with how close the genotypes are to the bias with which they are competing. For all other cases, there is either no correlation or the effects of beneficial mutations cannot be ruled out. One possible explanation for the strong signal with NEG competitions and correlation is how divergent NEG is from the selective pressure of the Logic-9 environment. Genotypes have a tendency to be closer in identity to POS than NEG (compare the range of KL-values in Figure 4.10). It may just be the case that the closeness of all genotypes to POS makes it impossible for selection to distinguish among small differences between genotype composition, meaning that all genotypes from all backgrounds are virtually identical with regard to the effects of synonymous mutation.

4.4 Summary

In this section, I examined whether or not mutation biases are visible to selection. I began by testing if selection can distinguish between mutation biases when competed against one another using the default nop-C ancestor and a variety of initial configurations. Of the two non-uniform biases introduced in the previous chapter, POS and NEG, POS is almost always selectively distinguishable from UNB. The structure of the population, whether there is local or global placement of offspring, plays an important role in altering outcome. Local replacement

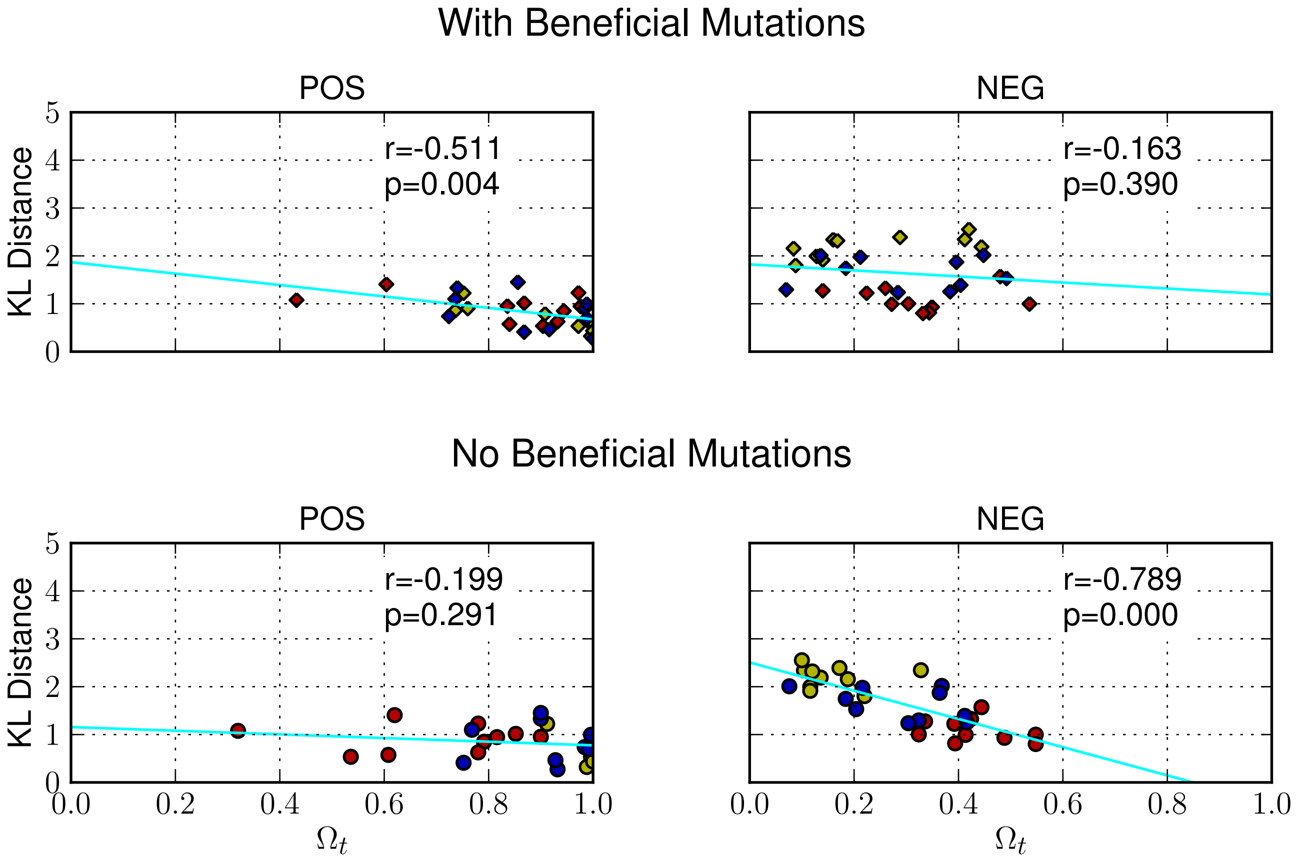

Figure 4.10. **Correlation of KL-Distance of evolved seed genotypes to competition outcomes.** Each plot shows competition results of `POS` or `NEG` against `UNB`. Beneficial mutations were either allowed or disallowed for each set of competitions. The color of the markers indicate which background the genotypes evolved under (yellow, `POS`; blue, `UNB`; red, `NEG`). KL-distance is based on the composition of the genotypes relative to the bias in which they competed, measured in bits. Correlation coefficients (r) and significance measures (p) are shown with each figure along with a least-squares fit line to the points.

allows for a stronger distinction of both POS and NEG relative to UNB. This difference is likely due to an increase in the interaction of mutations made possible by slowing selective sweeps. Population size also plays a role in the strength of distinguishing between biases; however, it is not as strong as population structure for the values tested.

Because the unevolved ancestor had a lot of room for mutations to accumulate, it is possible that the distinction of mutation biases is only an artifact of its lack of adaptation. I examined whether or not genotypes evolved under the three different mutation biases (10 from each) could also have selection distinguish mutation biases. I found that mutation biases can be distinguished by selection using evolved genotypes, but that the difference that population structure makes is not consistent, suggesting that mutation interaction is not as important with evolved genotypes as the unevolved ancestor.

In order to examine whether a decrease in deleterious mutations absent of beneficial mutations is sufficient for one mutation bias to be selected over another, I repeated the competition experiments using the 30 evolved genotypes. I found that there is virtually no difference in the number of genotypes that compete differently from random chance in experiments with and without beneficial mutations. Further, I found that there were some genotypes that competed better in competitions without beneficial mutations than with them.

This strange outcome led me to examine whether or not there is a correlation in outcome of competition and the mutation bias in which the genotype evolved. If such a correlation existed, it would explain why some genotypes performed better in competitions without beneficial mutations and indicate that there is a historically-contingent effect of mutation bias. I found that genotypes compete better when they use their native bias than when they do not. I also found that, at least for the NEG bias, there is a correlation between genotype composition and competition bias when beneficial mutations are not present. This observation implicates an increase in synonymous mutation as the cause of successful outcomes in NEG competitions. No such significant correlation was observed with POS competitions without beneficial mutations. One reason this may be the case is that genotypes tend to be more similar to `POS` than to `NEG` making small differences in composition have no effect on outcome.

Although selection can distinguish between mutation biases, especially when the biases are seeded at a relatively high abundance in the population, it is not certain that they can evolve. In the next chapter, I will examine whether or not a mutation bias can mutate into a population and be selected.

CHAPTER 5

Evolving Mutation Bias

In this chapter, I explore whether mutation biases can evolve. Previous chapters have shown that mutation bias can alter the outcome of competition and be a selectable trait. This chapter examines whether populations can evolve a mutation bias if isolated subpopulations and individuals are given the ability to mutate between biases.

This chapter is split into four parts: a discussion of difficulties and experimental modifications needed to investigate the evolution of mutation biases (Section 5.1); an overview of evolving mutation bias in segregated subpopulations called "demes" (Section 5.2); a discussion of experiments to evolve mutation biases without demes (Section 5.3); and finally a presentation of biased genetic landscapes, which helps to answer why one bias is selected over another (Section 5.4).

5.1 How to represent a mutation bias

I say that a mutation bias evolved if populations can maintain that bias at a frequency significantly greater than chance. In order for mutation bias to evolve it must be heritable, selectable, and subject to variation through mutation. In the previous chapters I have provided the first two requirements to organisms with regard to mutation bias but not the third.

For mutation biases to vary, there must be some genetic construct that maps its state to a set of mutation probabilities. This exercise is an instance of trying to solve the "representation problem", which is at the heart of evolutionary problem solving. Ideally, a good representation for a mutation bias (or any phenotypic trait) would allow for smooth phenotypic transitions, allow genomes to sample genotype space over the course of evolution, and allow selection to find the optimal phenotype. Before I can proceed with testing whether or not a mutation bias can evolve, I must first come up with some way to represent the mutation bias that adequately meets the preceding criteria and would also yield a clear signal that genotypes in a particular environment preferentially evolve one mutation bias over another.

The simplest way to get a clear indication that selection prefers individuals to have one bias over another is to reduce the number of possible states that the representation of a mutation bias can have. This is not an easy task. Consider a representation that allows each of the 26-instructions to take one of two values: low and high. For the sake of argument, let an instruction labeled as low have an inflow 10-fold higher than low. This representation allows for just over 76 million possibilities. For any given experiment, a population would be able to examine only the smallest fraction of this space. This is no different from any other evolutionary-problem solving effort: optimality is not guaranteed because a small fraction of the overall genotype space is ever examined by the population. However, the exercise here is not to find *the* optimal bias but to see whether or not individuals can evolve a bias with particular properties that are selectively advantageous.

It turns out that obtaining a signal in this example is difficult. In one attempt (data not shown), I found that the final mutation biases from populations evolved with just such a two state-setup for each instruction were entropically indistinguishable from random noise. In other words, there was no signal at all resulting from evolving a mutation bias in this manner. There are at least a few reasons why this setup didn't work. First, there was no possible selective advantage using the setup for the bias. I find this unlikely. Second, selection couldn't find a beneficial bias because of the averaging effect of the 26 different parameters. If instructions are split evenly between high and low probabilities, then the *relative* difference between probabilities is decreased. This averaging effect might make it difficult for selection to single out a set of instructions to be highly represented. Finally, the rate of change for the mutation bias might have been too high.

To simplify matters, I chose to limit the set of possible biases to three or six different options. If the majority of individuals across populations used of one or a set of these biases significantly more frequently than the others, then I said the populations have evolved their mutation bias.

A "mutational spectrum" describes which mutation biases are available to an organism. In this chapter, mutational spectra come in two varieties: a three-bias spectrum and a six-bias spectrum. The three bias spectrum is comprised of the three biases used in the preceding chapters: `POS`, `UNB`, and `NEG`. The six-bias spectrum is derived from these three biases. I created the three additional biases by expanding the relative abundance of each instruction in `POS` or `NEG` relative `UNB` by a particular factor. For example, 2×`POS` has the relative difference between each instruction in `POS` increased by a factor of 2 relative to `UNB`; $\frac{1}{2}$×`POS` has half the difference in abundance between `POS` and `UNB`; and $\frac{1}{2}$×`NEG` has half the difference between `NEG` and `UNB`. The six-bias spectrum transitions from `NEG` through 2×`POS` smoothly and is shown in Figure 5.1.

Changes to mutation biases may be applied either in a structured or unstructured manner. When structured, mutation bias can only change to the two nearest biases in their spectrum. Biases at the extremes of either spectrum are able to mutate directly to each other. Unstructured changes allow any mutation bias to mutate to any other in one step.

5.2 Mutation biases can evolve using demes.

My chief concern before starting this project was whether or not rare mutation biases would be lost by drift before their effects could be adequately evaluated by selection. In the competitions in Chapter 4, the rarest initial abundance of a mutation bias was 5%: 180 out of 3600 individuals. At this size, there was little effect of drift on competition outcome (there was no major difference when increasing population size) and only a slight advantage

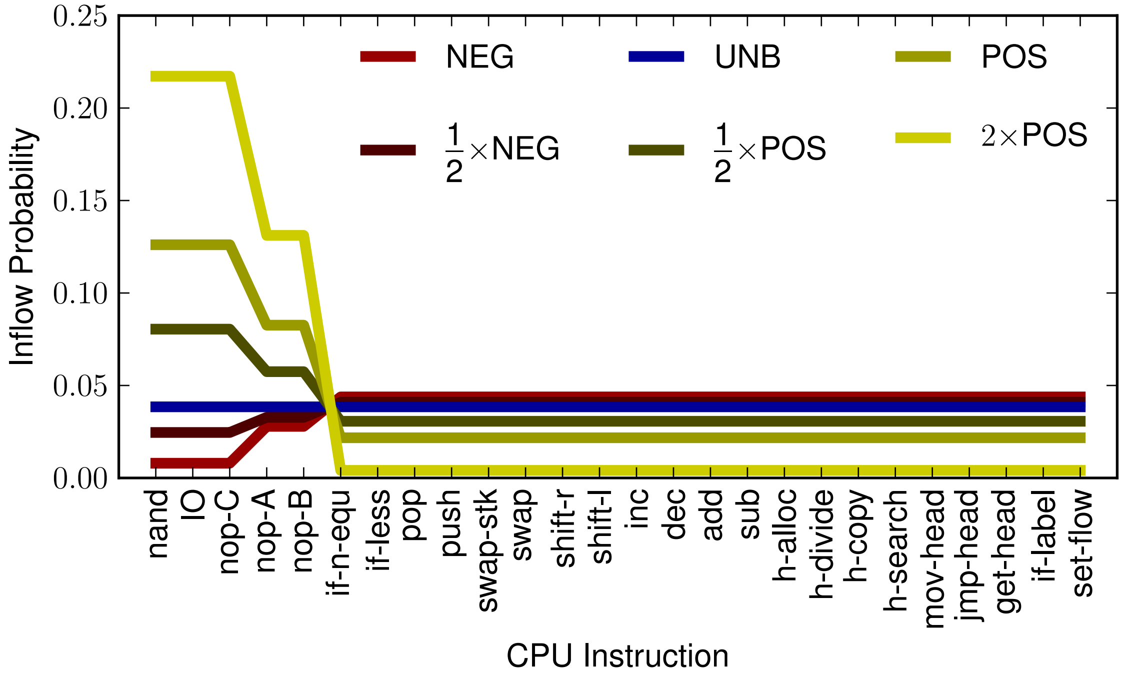

Figure 5.1. **The six-bias mutation spectrum.** All biases are derived by modifying the relative distance between instructions in `POS` or `NEG` relative to `UNB` by a particular factor, identified in the legend.

for `POS` winning more often than chance against `UNB`. It is not clear from these results that rare mutation biases would have much of a chance to thrive, especially since mutation bias does not confer an immediate fitness advantage.

To determine if selection can identify a mutation bias as being advantageous relative to another when allowed to change, I explored the evolution of mutation biases using demes rather than individuals as the carrier of mutation bias.

5.2.1 What are demes?

Demes are segregated subpopulations that co-exist but do not interact unless action is taken by the experimenter. For the experiments in this chapter, there was no migration of individuals between demes.

One of the uses of demes is to examine group selection. Demes can be made to compete against one another based on how well the individuals in each deme perform a particular group-level behavior. For this set of experiments which demes survive and replicate were based on the average fitness of each deme at the end of a round of competition. At that time, demes underwent tournament selection where each deme was seeded using the deme with the highest average fitness compared to three other demes chosen at random from the population. The only information transmitted between competition rounds were the demes' mutation bias. Each round began anew with all demes being fully-seeded with the default ancestor.

One of the initial difficulties with setting up deme experiments was deciding on a set of deme parameters. The number of demes, the size of each deme, and the length of competition were all additional considerations not present in an Avida experiment without demes. In order to quickly assay the outcome over a set of deme parameters, I collected average final population fitnesses over a range of population sizes. By examining variation in final average population fitness among experiments of different population sizes and different mutation biases, I found which experimental conditions were likely to allow for mutation biases to evolve in demes competitions.

Figure 5.2 summarizes the results of a set of calibration experiments I used to determine the size of each deme and the length of each round of competition. Populations of 200, 400, 800, and 1600 evolved for a total of 16,000 updates (the entire time range is not shown) under the same conditions demes would experience during each round of competition. Each population evolved using one of the three mutation biases in the three-bias spectrum. By examining when the average population fitness values differed *between bias treatments* for a particular population size, I could find when deme competitions would be likely to show competitive differences.

As a result of these calibration experiments, I selected a deme size of 400 with a total of 50 demes in each experiment. I held tournament selection between demes every 2000

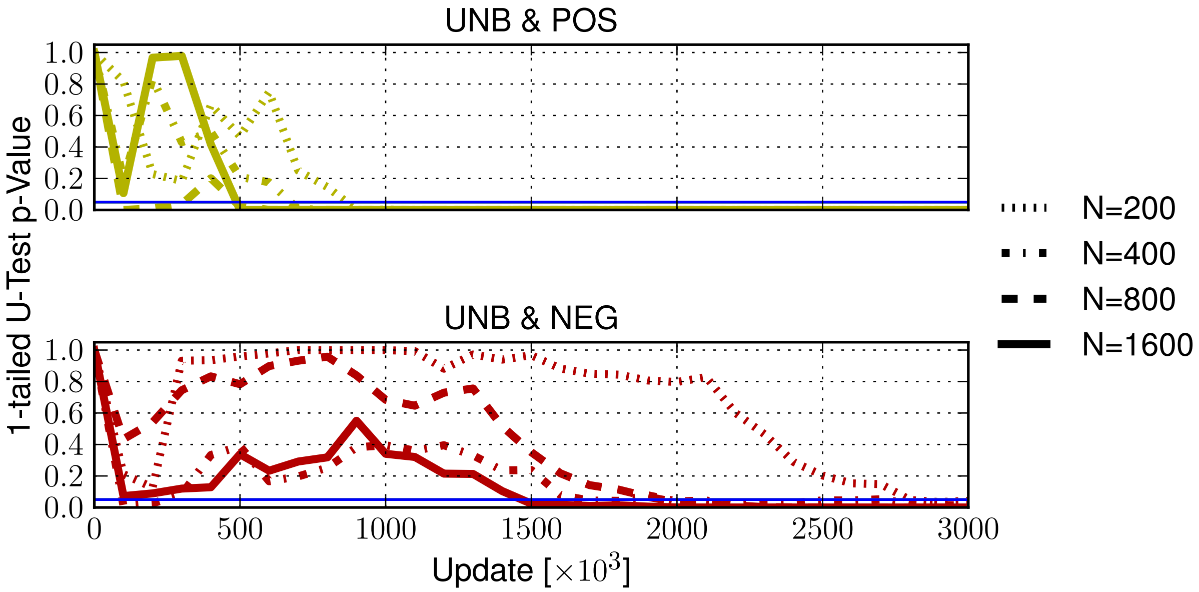

Figure 5.2. **Calibration experiments to find a good pair of deme size and competition length.** Populations of four different sizes evolved under each bias in the three-bias mutational spectrum. How significantly different the average population fitnesses are between each bias-treatment using a 1-tailed U-Test is shown along the vertical axis for each population size (N). Comparisons between POS and NEG treatments are not shown.

updates, a time identified by the calibration experiments to be where there was a significant difference between populations under the three biases. (Although assessed, Figure 5.2 does not show the significance of fitnesses between POS and NEG over time.)

5.2.2 Mutation bias can evolve with demes.

Mutation bias was able to evolve in demes under a variety of initial conditions, mutation methods, and with different mutational spectrums.

Figure 5.3 shows the average frequency demes possess each bias in thee three-bias spectrum. Experiments began with either all demes being assigned to be UNB or were given a random mutation bias. Each of the two setups was repeated 30 times and with each bias

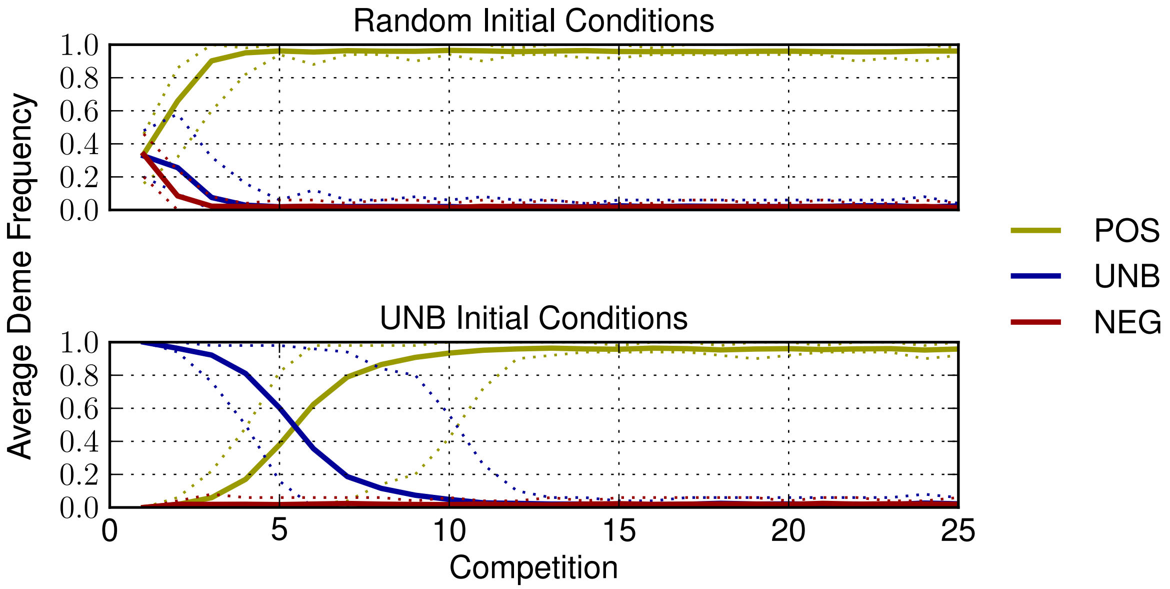

Figure 5.3. **Deme competitions beginning with different initial conditions.** The figures above show the average frequency of each bias in the three-bias mutational spectrum in 30 deme competition experiments over time. Treatment minimum and maximum are shown with dotted lines about the mean. Each experiment used 50 demes of 400 organisms with tournament selection every 2000 updates.

having an inflow rate of one for each round of competition to prevent competitive exclusion. When experiments began with demes being randomly assigned one of the three biases, `POS` demes consistently came to dominate experiments within five rounds of competition, with `UNB` and `NEG` populations quickly being driven to their minimum allowed frequency. When experiments began with all demes being `UNB`, it took a little more time for selection to choose `POS`, but by around 10 competitions into the experiment `UNB`, demes were driven out of the population. `NEG` never made a noticeable advancement its frequency.

Using the six-bias spectrum, deme competition revealed that 2×`POS` dominated experiments but `POS` was not excluded from the populations. Figure 5.4 shows how the distribution of demes with particular mutation bias changed over the course of competition using structured and unstructured mutations. Each of these two treatments had bias change at a rate of 0.05 per deme, per competition and no guarantee that all biases would be present for each round of competition.

Deme competition experiments that used structured mutation biases forced demes to transition from UNB to 2×POS via $\frac{1}{2}$×POS and POS. Approximately 5 rounds of tournament selection into the experiment, UNB had, on average, the same frequency as $\frac{1}{2}$×POS. A few rounds later, $\frac{1}{2}$×POS had reached its all time maximum average abundance in experiments; from then on, it lost ground to POS and, eventually 2×POS. Interestingly, 2×POS is not shown by these competition experiments to competitively exclude POS. Since POS is a bias created by successful mutations, I expected 2×POS to only exaggerate the effect. However, by the end of the experiments, POS had an average frequency of about 0.80 in experiments; 2×POS occupied an average a little less than 0.20. This would seem to indicate that, although POS is an overall better bias in these competitions, there is a small but reasonable chance that 2×POS will outperform POS.

Allowing demes to mutate between biases in the six-bias spectrum directly, a couple of distinct periods are noticeable. While the average presence of UNB in experiments was driven down, $\frac{1}{2}$×POS, POS, and 2×POS are all increasing in abundance at about the same rate. Approximately 5 rounds of competition into the experiments, the rate of $\frac{1}{2}$×POS increase slowed relative to the other two POS-based biases. It began to decline as POS and 2×POS increased their average abundance. Just after 5 rounds of competition, POS increased at a rate faster than 2×POS eventually having an average of 0.70 frequency. 2×POS reached an average frequency of 0.40 and declined to about 0.20 as POS continued to increase its presence in experiments.

The transition point at 5 rounds of competition is interesting since it shows that mutation biases of different effect can all increase simultaneously by competing against less successful biases; it was only when they reached a particularly high presence in the populations that they began to frequently compete against each other in tournament selection. These results

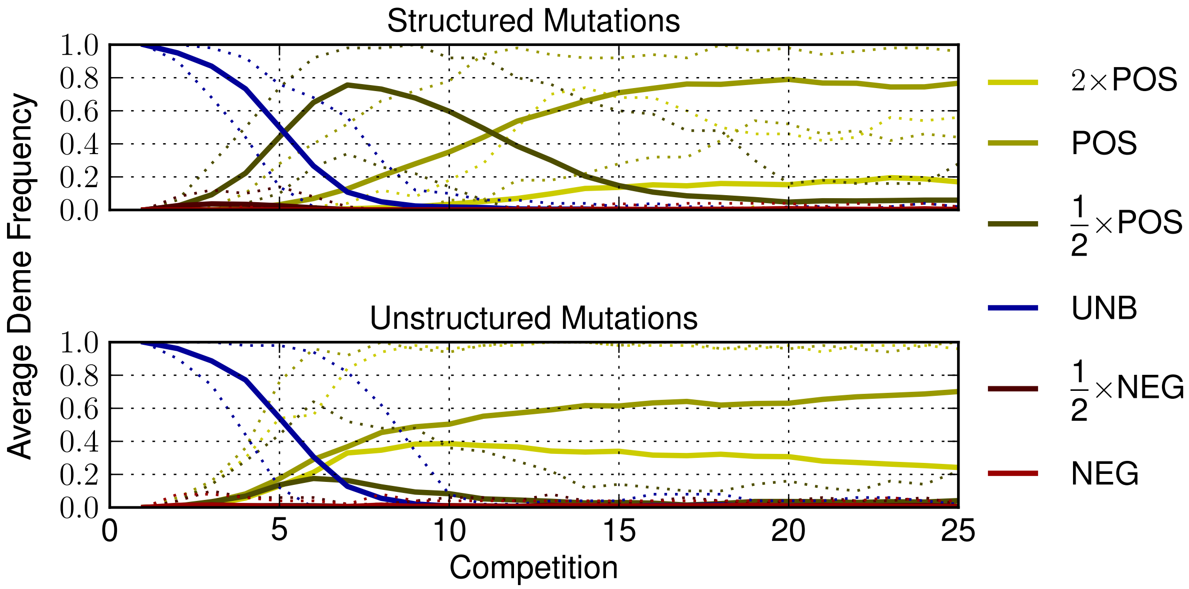

Figure 5.4. **Deme competitions having different mutation methods.** The solid lines above show the average frequency of mutation biases in 50 demes from 30 replicates. Dotted lines show the minimum and maximum of instruction set frequencies from the 30 treatments. Two treatments are shown above differing only in the way mutation bias can change after each round of competition. Structured mutations limit mutations to immediate neighbors in the six-bias spectrum. Unstructured mutations allow any bias to mutate to any other in only one step.

suggest that a variety of relatively-beneficial mutation biases can co-exist, provided there is a competitor competitively-worse than them.

5.3 Mutation biases can evolve without demes.

The success of the POS mutation bias in deme competitions shows the potential for a mutation bias to evolve; however, the outcome was expected due to the initial parameter assays. To truly test whether or not populations can evolve one of the mutation biases, I allowed individuals in a single population to evolve their mutation bias.

Figure 5.5 shows the results of allowing 300 replicate populations to evolve a mutation bias over 10^5 updates using the six-bias mutational spectrum. All populations began, fully seeded with the default `nop-C` ancestor and had the `UNB` bias. Mutation biases changed at a rate of 10^{-3} per generation. Biases were freely allowed to mutate over the entire six-bias spectrum. Other configuration settings were identical to those used in the majority of this work and listed in Appendix B.1.

Very quickly, populations had a strong preference for evolving to use 2×`POS`. By 3100 updates into the experiments, `UNB` lost its dominant state among all replicates, being replaced by 2×`POS`. By that time, there was also an increase at a smaller rate for both $\frac{1}{2}$×`POS` and `POS`. By the end of the experiment, 2×`POS` was the dominant mutation bias in 83% of populations. `POS` was the next most abundant dominant bias, dominating in less than 10% of populations.

The major difference in outcome between this experiment and the deme experiments above is the identity of the victor. In the deme experiments, `POS` strongly dominated experiment outcome. Here, 2×`POS` is clearly the dominant form in the vast majority of replicates.

To see if there was any difference in evolutionary outcome using 2×`POS` relative to `POS` and `UNB` I repeated the experiment in Section 3.2.1 using 2×`POS` as the only bias the replicate populations were able to use. The distribution of final average fitnesses are shown in Figure 5.6. I found no significant difference ($p = 0.075$, 2-tailed Mann-Whitney U-Test) in the final fitness between 2×`POS` and `UNB`; and a significant difference between `POS` and 2×`POS`.

Combined with the previous experiment, these results indicate that evolution can evolve a mutation bias; however, the evolve mutation bias might not be optimal for long-term success.

5.4 Why are mutation biases selected?

In the previous chapter, I showed that mutation biases can be selected even without the presence of beneficial mutations. I speculated that an overall reduction in the deleterious

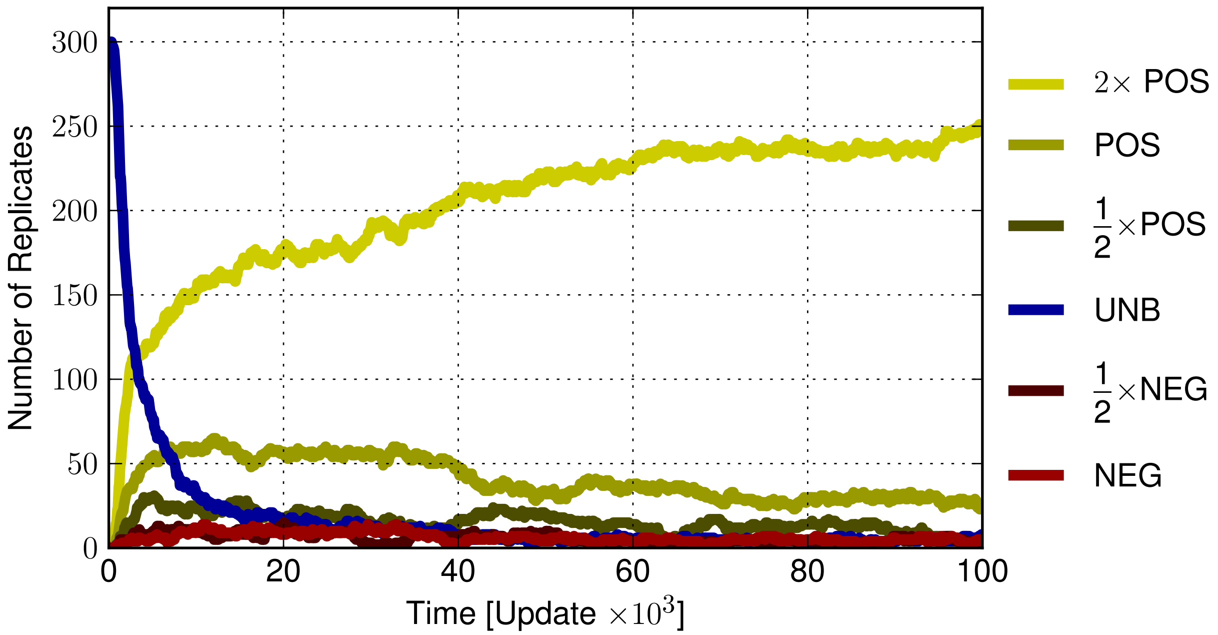

Figure 5.5. **The evolution of a mutation bias in a population without demes.** The number of replicates refers to the number of the 300 replicates where one mutation bias is expressed more than by half of the population. All populations began with organisms all having the UNB bias.

mutational load, which encompasses both an increase in neutral and beneficial mutations might be the cause of one bias outcompeting another. In this section, I expand upon my previous work by examining the local fitness landscapes of random viable genotypes over the course of a typical Avida experiment in this work.

From each of 100 replicate populations under conditions identical to the UNB treatment in Section 3.2.1, I chose a random viable genotype every 5000 updates. For each genotype, I then collected fitness information about its 1-substitution away neighbors. To get the 1-substitution neighbors, I mutated every site in the sample genome to each of the 25 instructions. I assigned a fitness to each mutant using the average fitness from 10 trials to account for phenotypic stochasticity. By comparing the relative fitness of each mutant to its original, I was able to find the probability that a substitution would be either beneficial,

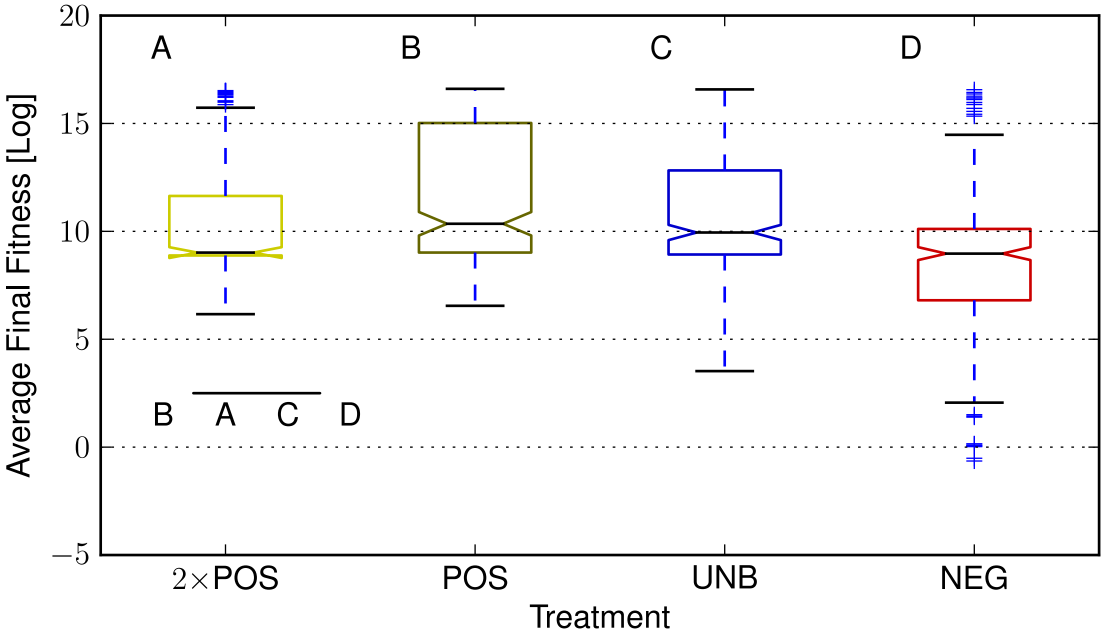

Figure 5.6. **Average fitness distribution in single bias experiments.** The 2×POS and UNB treatments had statistically identical final population average fitnesses ($p = 0.075$, 2-tailed Mann-Whitney U-Test)

neutral, deleterious, or lethal. To account for finite population size effects, I assigned genotypes with a relative fitness of (1.0 ± 3600^{-1}) to be neutral. To test for differences in the landscape due to mutation bias, I created fitness vectors where the relative abundance of each mutant was derived from its probability of arising under POS, UNB, and NEG biases.

Figure 5.7 shows how the average distribution of beneficial, neutral, deleterious, and lethal mutations vary in the local landscape of the random viable genotypes from 100 replicate populations under three mutation biases when synonymous mutations are allowed. In general, POS shows relatively higher probability of achieving a neutral mutation relative to the two other biases, and a correspondingly lower average probability of deleterious and lethal mutations.

In order to see if there is any quantitative difference between POS and UNB fitnesses, I

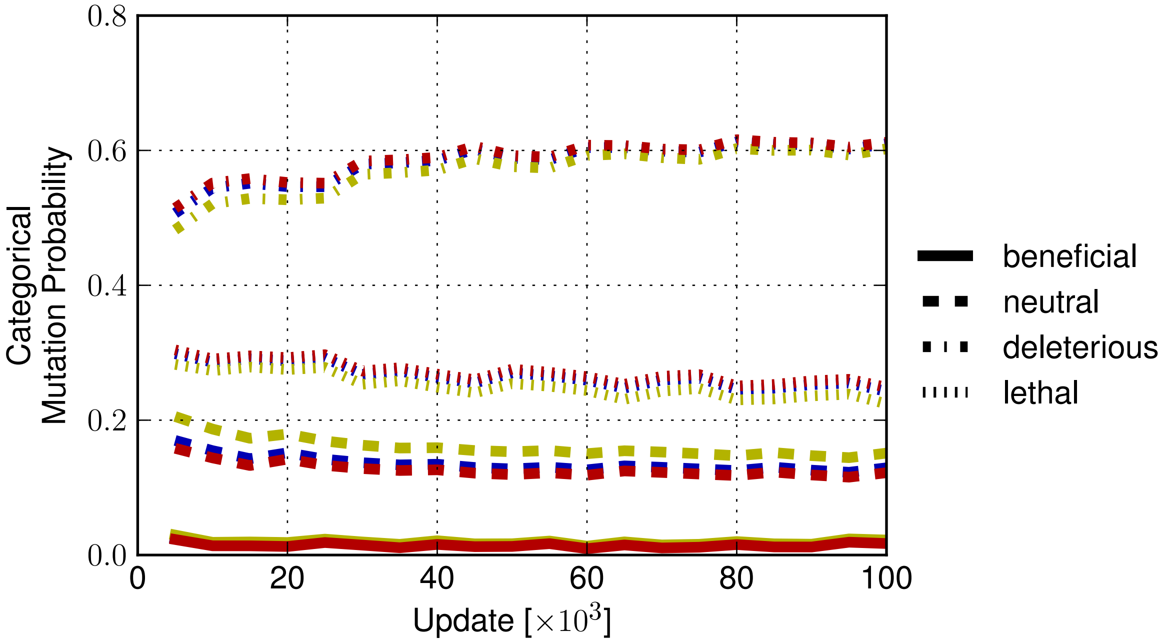

Figure 5.7. **Average distribution of mutation effects around random viable genotypes.** Each line in the graph above shows the average probability of a type of mutation around random viable genotypes from 100 replicates sampled every 5000 updates. The type of mutation is identified in the legend. Colored lines show the differences in distribution of mutation effect under `POS` (yellow), `UNB` (blue) and `NEG` (red).

compared the distribution of fitness vectors under each bias, both allowing and excluding the possibility of synonymous mutations for each sample landscape. Not accounting for multiple comparison corrections, only a couple samples had significantly different fitness vectors between `POS` and `UNB` ($p < 0.05$, 1-tailed U-Test) when synonymous substitutions were not included in the fitness vectors. Such results were not indicative of anything short of identical fitnesses of local neighbors regardless which of the two biases are used. Allowing vectors with synonymous substitution, there were many more instances where `POS`-biased fitness vectors have a significantly higher range of fitness values than `UNB`-biased counterparts; however, significance vanishes after correcting for multiple comparisons. For each of the 20 sample time points, at least one and as many as eight of the 100 replicates have `POS`-

fitness vectors with distributions shifted significantly to the right of `UNB`-fitnesses ($p < 0.05$, uncorrected 1-tailed U-Test). However, in both cases there was no significant indication of major differences between the distribution of 1-mutant fitnesses when correcting for multiple comparisons sans a single comparison made with synonymous mutations allowed at a one time point.

There is only a small and very weak difference in the local fitness landscapes of random viable genotypes between `POS` and `UNB` biases. How can evolution pick between biases with such a small local fitness?

One possibility for the small differences in local landscapes is that they could be the result of the sampling procedure. Since most mutations are at least deleterious (as shown in Figure 5.7), any random sample in the population is likely to not be the most fit. At any given time, all things being equal, any organism has only a N^{-1} (N being population size) chance of its offspring outcompeting all other organisms. This value decreases when fitness differences are taken into consideration. So, the samples taken from this set of experiments could just be from those common genotypes with lineages that preferentially become extinct over time.

Indeed, only organisms along the PLoD up to the most recent common ancestor at the end of an experiment were truly successful. Perhaps a rare event, such as a favorable local fitness landscape, a positive epistatic pairing, or a more fit competitor being randomly replaced in the population allows for success. In those cases, small, but constant trait-based biases might be amplified to the point where selection can distinguish them. Since mutation bias consistently decreases deleterious loads, its long-term influence might be great.

Perhaps a more interesting experiment would be to compare the frequency of significant differences of fitness distributions along the PLoD under different mutation biases relative to those of random samples. Even if there are slightly more differences that survive the scrutiny of strict statistical testing, there would be evidence to strengthen the argument that small, chance increases in beneficial outcome are the driving force for the evolution of mutation biases as well as singling out those organisms whose offspring will have a disproportionate chance to dominate.

Another possible reason why biases might be selected despite their statistically-identical local fitness landscapes is that the aggregate behavior of a bias might be a driving force for selectability over time. In other words, biases might not need to have a large local effect to be selectable; a small but consistent advantage might be sufficient for a bias to be preferentially selected.

5.5 Summary

In this chapter, I examined the ability for a mutation bias to evolve. With consideration of the large possibility state for mutation biases, I chose to use small three and six-bias mutational spectrums. I allowed both demes and individuals to evolve to use either the three or six biases. At the end of deme experiments, regardless of whether mutation was structured or unstructured, demes evolved to use `POS` preferentially over the other biases. When individuals were allowed to evolve a mutation bias, 2×`POS` rose to dominance in the in six-bias treatments, despite the fact it does not significantly improve adaptation relative to `UNB` in the long-term.

Looking for why one bias might be chosen over another, I sampled random viable genotypes periodically over the course of 100 replicate experiments. I then found the local fitness landscape around these genotypes and classified mutants as being either beneficial, neutral, deleterious or lethal relative to the sample. I also assigned fitnesses to each mutant. Although there appeared to be small qualitative differences between the landscapes biased by `POS` and `UNB` I found little statistical difference between the distribution of relative fitnesses under different biases for these landscaped samples. This lack of difference led me to consider the possibility that rare increases in the supply of non-deleterious mutations or small but consistent benefits might be the cause for a mutation bias to be selected.

CHAPTER 6

Discussion and Further Avenues to Explore

In the preceding chapters, I have shown that mutation bias can significantly change the outcome of evolution; can be visible to selection; and can evolve in a limited trait-space. One element, why a particular bias is chosen by evolution over another, I have only speculated about. I now discuss four possible ways to extend this work.

6.1 The role of synonymous mutations

One possible reason for the selection of a mutation bias such as `POS` may be the small increase in robustness caused by synonymous mutations during substitution events. The `POS` bias increases the probability of synonymous mutations relative to the `UNB` biased based on my findings in Chapter 3 that the `POS` bias more closely matches the composition of genomes than the `UNB` bias does. My findings in Chapter 5 provide some evidence that synonymous mutations may be a reason why the `POS` bias allows organisms to produce better offspring than the `UNB` bias. Synonymous mutation also may explain my Chapter 4 findings where I show that the `POS` bias can outcompete the `UNB` bias even when beneficial mutations are not allowed.

In future work, I plan to determine more precisely the role of synonymous mutations in the selection of mutation bias. I plan to do this by making synonymous mutations a researcher-controlled option that can be turned off. If a researcher does not allow synonymous mutations, the system will force a site to change to a new instruction when the site is selected for a substitution mutation. This implies the researcher-specified mutation rate will be identical to the effective mutation rate, something that is not quite true in the current implementation of Avida.

I plan to use this feature as follows. First, I will conduct more competition experiments from Chapter 4 where I disallow synonymous mutations and where I disallow both synonymous and beneficial mutations. With the first experiment, I can test whether beneficial mutations and other non-synonymous neutral mutations are sufficient to allow one bias such as POS to outcompete another bias such as UNB. With the second experiment, I can test whether non-synonymous neutral mutations are sufficient to allow one mutation bias such as POS to outcompete another mutation bias such as UNB.

6.2 The role of the environment

I performed all of the experiments in this work using a common set of experimental conditions and some variant of the Logic-9 reward structure. The Logic-9 reward structure is well-studied, making it an excellent test case; however, it is not known if these results are generalizable to other reward structures or environments.

In future work, I plan to repeat the preceding experiments using the path finding environment from Avida, where an organism is rewarded for its ability to follow a trail of resources. The path finding reward structure is fundamentally different than the Logic-9 reward structure in three ways. First, the critical instructions needed for performing a path finding behavior, which include sensing markers, turning, and moving, are significantly different than the instructions used to perform a logic operation such as NAND or IO. Second, because the rewarded behaviors are significantly different, the evolution of good solutions may be quite different in the path finding reward structure than the Logic-9 reward structure. One piece of evidence that supports this difference is that only 6% of path finding replicates typically evolve to high fitness where as 35% of `Logic-9` replicates typically evolve high fitness populations.

6.3 The role of mutation bias in warping fitness landscapes

I've only examined the effect of mutation bias on the neighbors closest to viable genotypes selected at random. I found that there is no significant difference in the distribution of fitnesses of these mutants when weighted by biases, but an overall small qualitative difference in the distribution of beneficial, neutral, and deleterious mutations. Even with these small changes mutation bias may warp the fitness landscape to allow different adaptations to emerge.

One way to test if biases produce different adaptations is to examine organisms evolved. Looking at which instructions are used to implement adaptations might provide information about the structure of the adaptation. If different biases produce differently structured adaptations, then there is evidence that warping the fitness landscape using a bias leads to different regions of the fitness landscape. Such an understanding is important because it provides an additional means of varying selective search by pitting selection and mutation against one another as drivers of evolution.

6.4 Allowing the evolution of arbitrary mutation biases

In this work, I created a small number of "complete" mutation biases that I let organisms use. Some reasons for this were practical: I wanted to have clear signals when selection chose a mutation bias, and I wanted to have a well defined set of states to analyze. However, these biases only explore a fraction of possible biases and were created from knowledge about evolution in the `Logic-9` reward structure. Allowing mutation biases to evolve with fewer constraints would strengthen this work. Further, additional parameters such as mutation rate could be varied to see if they affect the evolution of mutation bias.

One attempt to evolve mutation bias with additional degrees of freedom is shown in Figure 6.1. Replicates in these experiments were allowed to use any of 27 mutation biases: one of the biases was `UNB`; the other 26 increased the probability of mutating to a single instruction by a factor of ten.

I repeated each experiment 300 times over four different scalings of the default substitution rate to see if there were any changes in bias evolution. In general, `h-copy` was greatly preferred by populations, with the uniform bias's dominance decaying quickly, losing its top standing across all replicates by about 20×10^3 updates in all but the lowest mutation rate treatment. Under the mutation rate used in the majority of this work, a high `h-copy` bias came to dominance in 234 of the 300 replicates (78%). At both ends of the mutation rate assay, the preference for high `h-copy` weakened. At a substitution rate of 10^{-3} of normal, `h-copy` was in the majority of only 39 replicates. At a ten-fold higher substitution rate than usual, `h-copy` was only dominant in only 36 of 300 populations. `IO` had the most replicates evolving it at the end of that treatment with 46 replicates predominantly using it.

It would make sense that the mutation bias would be influenced by mutation rate. It is known that mutation rates higher than usual allow populations to better adapted to the `Logic-9` reward structure [5]. I expect populations might be able to leverage a task-producing instruction such as `IO` to a greater degree at these high mutation rates for either purposes of robustness due to a greater need for synonymous mutation or task-production.

Finally, having biases evolve to a greater degree on their own might provide additional information about environments, leading to the possibility of greater self-adaption of genomes to the environment.

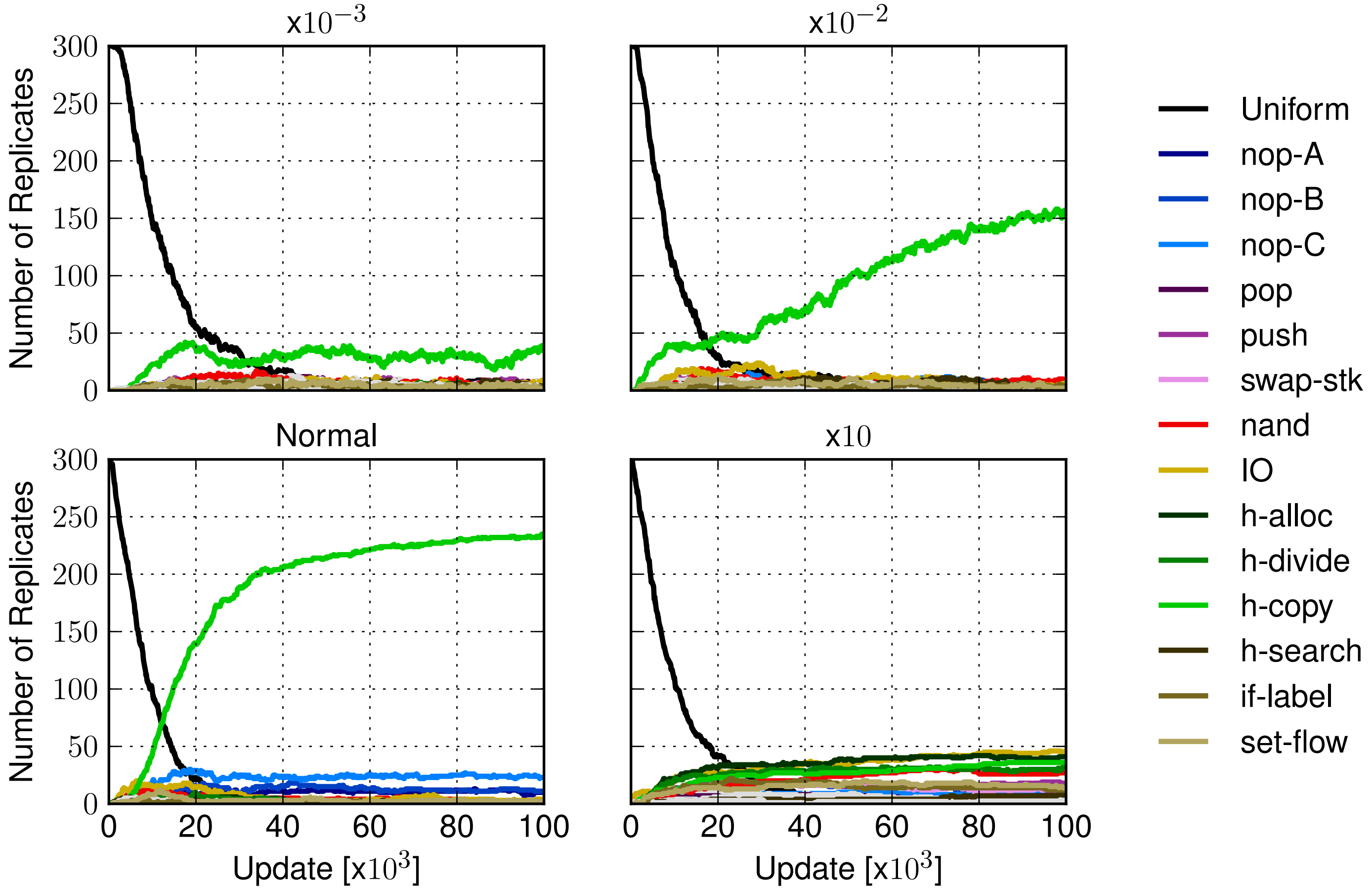

Figure 6.1. **The evolution of mutation biases over different mutation rates.** Each plot shows the number of replicates (out of 300) where a single mutation bias dominates in the population. 27 biases were allowed: `UNB` and a bias where a single instruction had an inflow rate 10 higher than the others. The legend shows the biases that a relatively high number of replicates preferred. The title for each plot shows the scaling of the rate of substitution from the default in Appendix B.1. Insertion and deletions were not scaled.

APPENDICES

APPENDIX A

Avida Virtual CPU Instructions

The virtual CPU instructions listed below were used in all Avida experiments in this work. Instruction descriptions were adapted from the documentation from Avida v2.6 [30]. Each instruction in an Avidian genome is represented by an alphabetic symbol.

By default, many of the instructions use the `BX` register and its compliment the `CX` register for inputs. This behavior may be modified by a `nop`, causing the instruction to examine a different register pair (listed below). The notation `?BX?` will be used to denote the conditional target pair of registers for relevant instructions below.

Instructions which move heads may have their behavior modified by `nop`s as follows: `nop-A` refers to the instruction pointer, `nop-B` to the read head, and `nop-C` to the write head. The conditional target heads will be labeled as `?IP?` below.

Table A.1. **Description of default Avida virtual CPU instructions.**

Symbol	Name	Description
a	`nop-A`	The instructions `nop-A` (a), `nop-B` (b), and `nop-C` (c) are no-operation instructions and will not do anything when executed. They can, however, modify the behavior of the preceding instruction or act a label that denotes a location in the genome. `nop`s are often paired with their compliment when specifying the behavior of the preceding instruction. `nop-A` is complimented with `nop-B`; `nop-B` with `nop-C`; and `nop-C` with `nop-A`.
b	`nop-B`	
c	`nop-C`	
d	`if-n-equ`	`if-n-equ` and `if-less`, along with `if-label` execute the next instruction (possibly following a modifying `nop`) if the contents of register `?BX?` is not equal to or less than `?CX?`, respectively. Any `nop` following the instruction will specify the register compliment pair to be used in the comparison.
e	`if-less`	
f	`pop`	`push` and `pop` will places from the value `?BX?` register into the active stack or moves the value from the top of the active stack to the `?BX?` register.
g	`push`	

Table A.1 cont'd

Symbol	Name	Description
h	`swap-stk`	Toggles the active stack.
i	`swap`	Swaps the contents of register `?BX?` with its compliment.
j	`shift-r`	These instructions modify the value of register `?BX?`. `shift-r` and `shift-l` perform a bitwise shift of the contents of register `?BX?` either right or left, respectively, padding with zeros. `inc` and `dec` raise or lower the value of register `?BX?` by 1. `add` sums register `?BX?` and `?CX?`; `sub` subtracts `?CX?` from `?BX?`. `NAND` performs a bitwise not-and on the contents of register `?BX?` and `?CX?`.
k	`shift-l`	
l	`inc`	
m	`dec`	
n	`add`	
o	`sub`	
p	`nand`	
q	`IO`	This instruction simultaneously transmits the contents of register BX and receives a value from the environment to place into BX.
r	`h-alloc`	This instruction allocates additional space to store the offspring genome at the end of the parent's genome. The new sites are initialized to `nop-A`. The size of the new allocation can be set in the configuration file.
s	`h-divide`	This instruction divides its offspring and (by default) resets the parent. The memory from the start of the organism to the read head constitutes the parent; the memory from the read head to the write head constitutes the offspring; excess sites are lost.

Table A.1 cont'd

Symbol	Name	Description
t	`h-copy`	This instruction copies the instruction located at the read head to the site located at the right head.
u	`h-search`	This instruction will place the flow head at the first following occurrence of a set of `nops` matching the compliment of the `nops` following it. If no `nops` follow `h-search`, then the flow head is placed immediately after it. Register `BX` and `CX` will receive the distance from the current position of the instruction pointer to the compliment and the size of the label, respectively. If no following `nop`s are present, the values of register BX and CX are set to 0.

v
w
x
	`mov-head`
	`jmp-head`
	`get-head`
		These instructions move `?IP?`. `mov-head` jumps `?IP?` to the location of the flow head. `jmp-head` moves `?IP?` the number of positions forward in the organism based on the value of register `CX`. `get-head` will copy the position of `?IP?` into the `CX` register.

y	`if-label`	`if-label` will read the `nop` label following it and tests whether or not the compliment of that label has been recently copied. If so, the following instruction gets executed; otherwise it gets skipped.
z	`set-flow`	This instruction moves the flow head to the position denoted in the `?CX?` register.

APPENDIX B

Experiment Settings

B.1 Common Settings

Many of the experiments in this work were performed using a common set of experimental settings. Below is a truncated list of these settings.

Table B.1. **Common experimental settings used in this work.**

Setting	Value	Description
`WORLD_GEOMETRY`	Torus	The shape of the Avida world. A Torus has no boundaries. This setting does not make any difference in the behavior of the Avida population when in Mass Action (global) placement; however, it does affect the behavior of Neighborhood (local) offspring placement.

Table B.1 cont'd

Setting	**Value**	**Description**
HARDWARE_TYPE	Original CPUs	All experiments in this work uses the original heads-based CPU. This virtual CPU contains three numerical registers, a special purpose register to store the last set of nops read, two stacks, and four pointers (instruction pointer and the read, write, and flow heads).
BIRTH_METHOD	Mass Action or Random in neighborhood	Depending on the experiment, this may be one of two values. In Mass Action, offspring are placed randomly in the population. In neighborhood, offspring replace an organism in the cells surrounding the parent.
PREFER_EMPTY	Yes	Preferentially select empty cells when using offspring placement.
ALLOW_PARENT	Yes	Allow the parent to be replaced by the offspring.

DEATH_METHOD
AGE_LIMIT
AGE_DEVIATION
	Exec = N*LIM+d
	N=20
	d=0
		Organisms will die when the number of instruction executed equals the length of its genomes * AGE_LIMIT + AGE_DEVIATION.

Table B.1 cont'd

Setting	**Value**	**Description**
DIVIDE_METHOD	Reset Parent	The parent's registers and heads will be reset after the division of the offspring.
CHILD_SIZE_RANGE	2.0	This setting sets the maximum difference in length between the parent and offspring. A value of 2.0 requires the offspring to be at least half as long as the parent and at most twice as long. This value also determines the number of instructions added to the end of the genome following the first execution of the h-alloc instruction.
MIN_COPIED_LINES	0.9	This sets the minimum fraction of the number of sites in the parent genome that must be copied before division is allowed. h-divide instructions executed before this criteria is met are ignored.
MIN_EXE_LINES	0.05	This sets the minimum fraction of instructions that must be executed by the organism before divide. h-divide instructions executed before this criteria is met are ignored.
REQUIRE_ALLOCATE	Yes	This setting requires that h-alloc be executed before a divide is allowed.

Table B.1 cont'd

Setting	Value	Description
FAIL_IMPLICIT	Yes	This setting prohibits offspring that are not identical to their parent in the absence of explicit mutations, such as the ones listed below.
POINT_MUT_PROB	0.0	There are no "cosmic-ray" mutations applied per update.
COPY_MUT_PROB	0.0025	This is the probability of a substitution (including synonymous) occurring during the copying of an instruction.
INS_MUT_PROB	0.0	These settings are for insertion (INS), deletion (DEL), or point (DIV) mutations applied per-site on divide.
DEL_MUT_PROB	0.0	
DIV_MUT_PROB	0.0	
DIVIDE_MUT_PROB DIVIDE_INS_PROB DIVIDE_DEL_PROB	0.0 0.05 0.05 0.0	These settings are for mutations applied on divide over the entire genome for point (MUT), insertion (INS) and deletion (DEL) events.

Table B.1 cont'd

Setting	Value	Description
`AVE_TIME_SLICE`	30	These settings affect how merit is mapped into SIPs. The first setting determines how many SIPs are to be made available to the population, in this case $30 \times N$, where N is the population size. The proportional slicing method distributes merit randomly based on the relative merit of each individual (e.g. it is not deterministic). The base merit method sets the initial merit (before task bonuses are applied) to be proportional to the minimum of the number of lines executed or copied by the parent. The default bonus is the initial merit before any tasks are applied.
`SLICING_METHOD`	Probabilistic	
`BASE_MERIT_METHOD`	$\propto min\{exec, copied\}$	
`DEFAULT_BONUS`	1.0	

B.2 Calibration Settings

The initial runs used to generate the steady state matrix differed from the common settings. Below are the differences between the common settings and the calibration settings.

Table B.2. **Deviant calibration experiment settings.**

Setting	**Value**	**Description**
DEATH_METHOD	Never die	In the calibration runs, organisms never died of old age.
MIN_COPIED_LINES	0.50	Fewer sites needed to be copied in the calibration runs before divide is allowed.
COPY_MUT_PROB	0.0075	The substitution rate was three times higher than the common settings.
FAIL_IMPLICIT	Off	Organisms are allowed to produce different organisms in the absence of mutation.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Banerjee, T., Gupta, S. K., and Ghosh, T. C. Role of mutational bias and natural selection on genome-wide nucleotide bias in prokaryotic organisms. *Biosystems 81*, 1 (Jul 2005), 11–8.

[2] Belozersky, A., and Spirin, S. A correlation between the compositions of deoxiribonucleic and ribonucleic acids. *Nature 183*, 4628 (1958), 111–112.

[3] Bernardi, G. Isochores and the evolutionary genomics of vertebrates. *Gene 241*, 1 (1 2000), 3–17.

[4] Bhatnagar, S. K., Bullions, L. C., Lew, G., and Bessman, M. J. Characterization of the defect in the Escherichia coli mutT1 mutator gene. *The Journal of Bacteriology 172*, 5 (5 1990), 2802–2803.

[5] Clune, J., Misevic, D., Ofria, C., Lenski, R. E., Elena, S. F., and Sanjuán, R. Natural selection fails to optimize mutation rates for long-term adaptation on rugged fitness landscapes. *PLoS Computational Biology 4*, 9 (09 2008), e1000187–.

[6] Cox, E. C., and Yanofsky, C. Altered base ratios in DNA of an Escherichia coli mutator strain. *Proceedings Of The National Academy Of Sciences Of The United States Of America 58*, 5 (1967), 1895.

[7] Dayhoff, M., Schwartz, R., and Orcutt, B. C. A model of evolutionary change in proteins, supplement 3. In *Atlas of Protein Sequence and Structure*, M. Dayhoff, Ed. National Biomedical Research Foundation, Washington, DC, 1978, pp. 345–352.

[8] Dorn, E. *Universal Biosignatures for the Detection of Life.* Phd, California Institute of Technology, May 2005.

[9] Dorn, E., Nealson, K., and Adami, C. Monomer abundance distribution patterns as a universal biosignature: Examples from terrestrial and digital life. *Journal of Molecular Evolution* (2011-01-21), 1–13.

[10] Fowler, R., and Schaaper, R. The role of the mutT gene of Escherichia coli in maintaining replication fidelity. *FEMS Microbiology Reviews 21*, 1 (AUG 1997), 43–54.

[11] Francino, M. P., and Ochman, H. Strand asymmetries in DNA evolution. *Trends in Genetics 13*, 6 (Jun 1997), 240–245.

[12] Frederico, L. A., Kunkel, T. A., and Shaw, B. R. cytosine deamination in mismatched base-pairs. *Biochemistry 32*, 26 (Jul 1993), 6523–6530.

[13] Freese, E. On evolution of base composition of DNA. *Journal of Theoretical Biology 3*, 1 (1962), 82–101.

[14] Gentile, C. F., Yu, S.-C., Serrano, S. A., Gerrish, P. J., and Sniegowski, P. D. Competition between high- and higher-mutating strains of Escherichia coli. *Biology Letters* (2011).

[15] Gu, X., Hewett-Emmett, D., and Li, W. H. Directional mutational pressure affects the amino acid composition and hydrophobicity of proteins in bacteria. *Genetica 102*, 3 (1998), 383–391.

[16] Henikoff, S., and Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences 89*, 22 (11 1992), 10915–10919.

[17] Hurst, L. D., and Merchant, A. R. High guanine-cytosine content is not an adaptation to high temperature: a comparative analysis amongst prokaryotes. *Proceedings Of The National Academy of Sciences of the United States of America 268*, 1466 (Mar 2001), 493–497.

[18] Jukes, T. H., and Bhushan, V. Silent nucleotide substitutions and G+C content of some mitochondrial and bacterial genes. *Journal of Molecular Evolution 24*, 1 (1986-12-01), 39–44.

[19] Karran, P., and Lindahl, T. Hypoxanthine in deoxyribonucleic acid: generation by heat-induced hydrolysis of adenine residues and release in free form by a deoxyribonucleic acid glycosylase from calf thymus. *Biochemistry 19*, 26 (12 1980), 6005–6011.

[20] Kauffman, S. A. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology 22*, 3 (3 1969), 437–467.

[21] Lenski, R., Ofria, C., Pennock, R., and Adami, C. The evolutionary origin of complex features. *Nature 423* (2003), 139–144.

[22] Lind, P. A., and Andersson, D. I. Whole-genome mutational biases in bacteria. *Proceedings Of The National Academy of Sciences of the United States of America 105*, 46 (Nov 2008), 17878–17883.

[23] Lindahl, T., and Nyberg, B. Heat-induced deamination of cytosine residues in deoxyribonucleic acid. *Biochemistry 13*, 16 (1974), 3405–3410.

[24] Maki, H., and Sekiguchi, M. Mutt protein specifically hydrolyses a potent mutagenic substrate for dna synthesis. *Nature 355*, 6357 (January 1992), 273–275.

[25] McLean, M. J., Wolfe, K. H., and Devine, K. M. Base composition skews, replication orientation, and gene orientation in 12 prokaryote genomes. *Journal of Molecular Evolution 47*, 6 (Dec 1998), 691–696.

[26] Mrazek, J., and Karlin, S. Strand compositional asymmetry in bacterial and large viral genomes. *Proceedings Of The National Academy of Sciences of the United States of America 95*, 7 (Mar 1998), 3720–3725.

[27] MUTO, A., AND OSAWA, S. The guanine and cytosine content of genomic DNA and bacterial evolution. *Proceedings Of The National Academy of Sciences of the United States of America 84*, 1 (JAN 1987), 166–169.

[28] OFRIA, C., HUANG, W., AND TORNG, E. On the gradual evolution of complexity and the sudden emergence of complex features. *Artifiical Life 14*, 3 (2008), 255–263. Artificial Life.

[29] OFRIA, C., AND WILKE, C. Avida: A software platform for research in computational evolutionary biology. *Journal of Artificial Life 10* (2004), 191–229.

[30] OFRIA, ET AL. The avida digital life platform. Version 2.6, http://avida.devosoft.org.

[31] PENNOCK, R. Models, simulations, instantiations and evidence: The case of digital evolution. *Journal of Experimental and Theoretical Artificial Intelligence 19*, 1 (2007), 29–42.

[32] RAY, T. S. Evolution, complexity, entropy and artificial reality. *Physica D: Nonlinear Phenomena 75*, 1-3 (1994), 239 – 263.

[33] ROCHA, E. P. C., AND DANCHIN, A. Base composition bias might result from competition for metabolic resources. *Trends in Genetics 18*, 6 (Jun 2002), 291–294.

[34] RUPP, M., TORNG, E., AND OFRIA, C. The evolution of novel body types under differing selective pressurs in digital organisms. In *Proceedings of the Tenth International Conference on Artificial Life* (Bloomington, IN, 2006), pp. 268–274.

[35] SINGER, G. A. C., AND HICKEY, D. A. Nucleotide bias causes a genomewide bias in the amino acid composition of proteins. *Molecular Biology and Evolution 17*, 11 (Nov 2000), 1581–1588.

[36] SNIEGOWSKI, P. D., GERRISH, P. J., JOHNSON, T., AND SHAVER, A. The evolution of mutation rates: separating causes from consequences. *BioEssays 22*, 12 (Dec 2000), 1057–1066.

[37] STOLTZFUS, A. Mutation-biased adaptation in a protein NK model. *Molecular Biology and Evolution 23*, 10 (2006), 1852–1862.

[38] SUEOKA, N. Correlation between base composition of deoxyribonucleic acid and amino acid composition of protein. *Proceedings of the National Academy of Sciences 47*, 8 (08 1961), 1141–1149.

[39] SUEOKA, N. On the genetic basis of variation and heterogeneity of DNA base composition. *Proceedings Of The National Academy of Sciences of the United States of America 48*, 4 (1962), 582–592.

[40] SUEOKA, N. Directional mutation pressure and neutral molecular evolution. *Proceedings Of The National Academy of Sciences of the United States of America 85* (1988), 2653–2657.

[41] Treffers, H. P., Spinelli, V., and Belser, N. O. A factor (or mutator gene) influencing mutation rates in Escherichia coli. *Proceedings Of The National Academy of Sciences of the United States of America 40*, 11 (11 1954), 1064–1071.

[42] Wada, A., and Suyama, A. Local stability of DNA and RNA secondary structure and its relation to biological functions. *Progress in Biophysics and Molecular Biology 47*, 2 (1986), 113–157.

[43] Wang, H. C., Singer, G. A. C., and Hickey, D. A. Mutational bias affects protein evolution in flowering plants. *Molecular Biology and Evolution 21*, 1 (2004), 90–96.

[44] Wernegreen, J. J., and Funk, D. J. Mutation exposed: A neutral explanation for extreme base composition of an endosymbiont genome. *Journal of Molecular Evolution 59*, 6 (Dec 2004), 849–858.

[45] Yampolsky, L., and Stoltzfus, A. Bias in the introduction of variation as an orienting factor in evolution. *Evolution & Development 3*, 2 (2001), 73–83.