This is to certify that the
dissertation entitled

Analyzing Biological Complexity with Digital Organisms

presented by

Wei Huang

has been accepted towards fulfillment
of the requirements for the

Ph.D degree in Computer Science and Engineering

Major Professor's Signature

12-6-05

Date

ANALYZING BIOLOGICAL COMPLEXITY WITH DIGITAL ORGANISMS

By

Wei Huang

A DISSERTATION

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

DOCTOR OF PHILOSOPHY

Computer Science and Engineering

2005

ABSTRACT

ANALYZING BIOLOGICAL COMPLEXITY WITH DIGITAL ORGANISMS

By

Wei Huang

We define an organism's biological complexity to be the amount of information contained in that organism's genome about its environment. Not only is this an intuitive definition of complexity, but it also allows us to treat the subject in a rigorously mathematical fashion. As such, we have designed a method to measure biological complexity based on the principle of mutation-selection balance from population genetics. Our method approximates the total information in a genome as the sum of the information at each position. The information content of a given position is calculated by testing all of the possible mutations for that position and calculating the expected frequencies of potential genomes at the equilibrium state.

We use our definition of biological complexity to analyze the evolution of complex traits in digital organisms. To many, the seemingly sudden appearance of new traits seems to contradict the gradual evolutionary processes of mutation, selection, and drift despite previous work that illustrates how complex organismal features can arise if simpler traits that can be used as building blocks are selected for. Our analysis shows that the underlying information associated with any trait evolves gradually and often results from a combination of reusing and extending information associated with simpler traits. Specifically, we demonstrate that the majority of the genomic information associated with a trait is primarily correlated with pre-existing traits, or is co-opted from traits that were lost in conjunction with the appearance of the new trait.

Next, we extend the concept of complexity to the community level where we quantitatively measure the distinct information stored anywhere in a whole community

about its environment. Community complexity is a new concept that is different from the traditionally studied community diversity. We developed a measure that provides a useful approximation of community complexity, which we plan to further refine in the future. Our current measure accurately reflects that community complexity increases due to information gain, even when diversity is unaffected. It also shows that the community complexity of a multi-niche environment is only slightly higher than it in a single-niche environment when the organismal traits are identical. In such a case, the individual organisms in the single-niche environment have higher complexity, but many more species exist in the multi-niche environment. We systematically test this concept across many environment types and demonstrate its robustness.

Finally, knowing how information is stored in different organisms also tells us about the relationships among them. When new information enters a population, it is transmitted over time from parent to child. When information is shared among organisms in the final population, those organisms are likely related. Inspired by this fact, we designed a character weighting technique to improve phylogeny reconstruction accuracy. In this method, sites are weighted based on the portion of the tree being reconstructed. We target new information that is likely to have arisen at the branching point we are trying to reconstruct as the basis to weight characters. This approach lays the groundwork for a new class of top-down phylogeny reconstruction algorithms.

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Charles Ofria, for his guidance and support throughout my graduate career at Michigan State University. I wish to thank him for letting me join his lab and for helping me make this thesis a reality. Charles has also been a good friend who helped me understand the American culture and overcome the language barrier.

All of my committee members have provided me with helpful and thoughtful advice. I want to thank both Dr. Eric Torng and Dr. Thomas Schmidt for being my co-advisors. Dr. Torng had many detailed and insightful suggestions for each of my projects and this dissertation. Dr. Schmidt helped guide the biological direction of my interdisciplinary research. They both helped me to meet my project milestones. I would also like to thank Dr. Erik Goodman for asking many interesting questions that helped to refine my ideas.

Dehua Hang and Jason Stredwick have both been wonderful labmates and friends. They helped me significantly with my personal life and studies. Dehua provided efficient instruction and invaluable advice at the beginning of my graduate work, and Jason contributed the remarkable programming skills which made Chapter 4 of this thesis possible.

Many other people had significant impact on my work. Dr. Richard Lenski, Dr. Elizabeth Ostrowski, Gabriel Yedid, Dr. Kristina Hillesland, and Dusan Misevic have been involved in many discussions of my projects, which helped fill in my biological background. Kaben Nanlohy and Brian Baer provided significant technique support. I also want to thank Sherri Goings, Jeffrey Clune, Matthew Rupp, Arthur Covert III, and David Bryson for their great friendship.

I would like to thank the MSU Computer Science and Engineering Department and the Quantitative Biology and Modeling Initiative for support and funding. Addition-

ally, this project was supported by National Science Foundation grants DEB-9981397 and EIA-0219229.

My heartfelt thanks go to my family members. My parents, Ling Wang and Yunguo Huang, gave me unconditional love and lifetime education. My parents-in-law, Jiedong Yuan and Deming Wu, provided all kinds of support, especially helping to take care of my baby during my thesis writing.

Last but not least, my deepest thanks go to my husband, Wei Wu, for all his love and support, and to our son Jason Yaheng Wu for having the sweetest smile a baby can have.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Overview

Biological complexity has long been a contentious topic. Most people accept that complexity generally increases through the process of evolution but few give an explicit definition and method of measurement. My interest is to understand the relationship between biological complexity and evolution. Is there a pervasive trend of complexity change through evolution? Is it always an uphill climb? How does new complexity arise? Is there a relationship between the complexity of an organism and its fitness in the environment? Can we measure the complexity of a whole community? Are there any applications to studying complexity? The answer to each of these questions depends on the nature of the complexity definition. Many definitions have been proposed in the past, but each of them has serious flaws, and few use a rigorously mathematical approach.

This chapter provides a review of the biological complexity measures that others have used, and explores the limitations of each. The discussion following details the concept of physical complexity previously developed by Adami and Cerf [2000] and refined by Adami, Ofria, and Collier [2000]. To facilitate this discussion, I will also review some concepts from Shannon Information Theory and briefly describe the Avida digital life platform, the system we will be using to examine the complexity measures discussed.

Our new approach to estimating complexity is described in chapter 2. It is based

on Adami et al.'s "physical complexity" [*Adami and Cerf*, 2000], which defines biological complexity as the genetic information that an organism has about its environment. We approximate the total information in a genome as the sum of the information at each position. The information content of a position is calculated by testing all the possible mutations for that position and calculating the expected frequencies of potential genomes at the equilibrium state. We discuss how this method reveals the way information is embedded in an organism during the evolutionary process, the advantages of our method, and the initial results from applying this approach.

In chapter 3, I examine how new complexity first arises in a population. Evolutionary theory explains the origin of complex organismal features through a combination of reusing and extending information from less-complex traits. While the appearance of a new trait may seem sudden, the underlying information associated with that trait must evolve gradually. We study this process within evolving digital populations. We show that when a new complex trait first appears in the population, its proper function requires the coordinated operation of many genomic positions. However, the total information stored in the genome only increases marginally. We demonstrate that the majority of the information associated with an emerging trait is primarily correlated with pre-existing traits or is co-opted from traits that were lost in conjunction with the appearance of the new trait.

Next, in chapter 4, we extend the concept of biological complexity to the community level. We define community complexity as the sum of all distinct information contained in the community about its environment; if multiple organisms all contain the same information, we only count that information once. We developed a measure that provides a useful approximation of community complexity, which we plan to further refine in the future. Community complexity is a new concept that is different from the traditionally studied community diversity. I reviewed two popular diversity measurements: The Shannon and Simpson diversity indices. We analyzed the com-

plexity changes during evolution from the information perspective and compared it with diversity measurements. We designed five environments that allow us to explore environmental impact on community complexity.

Next we examine applications of understanding organism complexity. Measuring the information content of different organisms allows us to better understand the relationship between them. When new information enters a population, it is transmitted to subsequent generations and could help indicate relationships among organisms in that population. Inspired by this fact, we designed a character weighting technique to improve phylogeny reconstruction accuracy, which I present in Chapter 5.

Phylogeny reconstruction seeks to find evolutionary relationships among taxonomic groups. We observe that sites where two distinct symbols are both highly represented are more likely to provide useful information for reconstructing deep bifurcations in the tree. Both symbols may originate from adaptive events at earlier stages of evolution. Giving high weight to these sites will decrease our uncertainty about the root branch. We demonstrate that the neighbor joining algorithm is significantly improved in reconstructing deep bifurcations if more weight is given to these sites. We further show the robustness of this technique with sustained reconstruction improvements as we vary a number of characteristics about the trees, including the number of leaves, the alphabet size used in the sequences, and the tree symmetry.

Finally in Chapter 6 I discuss further plans for refining and extending this work. For the technique of measuring the organismal complexity, I will consider epistasis and estimate the information in two or more sites together. For the community complexity, I will show the candidate methods that could help more accurately calculate the unique information in each organism. For the character-weighting technique, I will try to convert it to a fully functional algorithm to build a whole tree, not just deep bifurcations.

1.2 Background

1.2.1 Biological Complexity

KCS (Kolmogorov-Chaitin-Solomonoff) complexity is the most widely used complexity definition. It defines the complexity of a sequence as the shortest possible program that can generate that sequence [*Li and Vitanyi*, 1997]. This definition works in many intuitive cases, but has some serious problems: some apparently complex structures can be coded in short programs such as fractals and cellular automata [*Goertzel*, 1993], while a long sequence with no pattern, and no meaning (effectively random) would need a long program to generate it; one that just lists the entire sequence.

A related complexity definition is "logical depth". Bennett [1988] defines the logical depth of a sequence by the running time of the shortest program that computes it. Thus, while it overcomes some of the problems with KCS complexity because more complex structures will typically take a while to generate, there are still problems when it comes to random sequences with no actual meaning behind them. Additionally, it cannot answer questions such as, "What is the shortest program to generate a DNA sequence?" or "What is the running time for that program?" and thus is still not an easily quantifiable definition for a biologist to use in measuring complexity.

A count of the number of "parts" in an organism is perhaps the simplest definition of complexity, as suggested by Hinegardner and Engelberg [1983]. This, of course, depends on what we recognize as parts. Hinegardner and Engelberg suggest that at root, organisms are composed of molecules, but they do not take the differences in the complexity of those molecules into account. This definition may provide a useful approximation of complexity, but it neglects any complexity inherent in gene regulation or other connections between DNA, RNA, and proteins. Many different proteins may be synthesized from the same mRNA molecule due to manipulations on the mRNA and in the context of translation. In fact, over the last several hun-

dred million years, most evolution has occurred only in the form of gene regulation even though most biologists agree that there has been a huge increase in the overall complexity of organisms.

Adami and Cerf [2000] developed physical complexity as a method to calculate the complexity of symbolic strings. Ofria and Collier then worked with Adami [2000] to translate this concept more directly to the study the evolution of biological complexity. In their definition, the biological complexity of an organism is the information physically stored in the genome about the environment in which it lives. In chapter 2 I talk about their initial method to measure physical complexity (biological complexity) before detailing my new, refined approach.

1.2.2 Avida Digital Evolution Platform

It is a commonly held belief that the complexity of species always increases during evolution. However, evolution by natural selection is a unique process in the biological world—we only have one example of it—and all known life has a genetic basis consisting of strings of nucleotides and is all believed to share a single common ancestor. We don't know the evolutionary principles for other forms of life which scientists may discover later. The eminent biologist John Maynard Smith [1992] declared that the only way out of this quandary was to build a new form of life ourselves. "We badly need a comparative biology," he wrote. "So far, we have been able to study only one evolving system, and we cannot wait for interstellar flight to provide us with a second. If we want to discover generalizations about evolving systems, we will have to look at artificial ones."

Having a well controlled artificial system allows us to explore the importance of many historically contingent events in evolution. Random chance is believed to have played a large role in this evolutionary process, as Gould hypothesized with his thought experiment of "replaying life's tape" [*Gould*, 1989]. He states: "Any replay

of the tape would lead evolution down a pathway radically different from the road actually taken ... Each step proceeds for cause, but no finale can be specified at the start, and none would ever occur a second time in the same way, because any pathway proceeds through thousands of improbable stages. Alter any early event, ever so slightly and without apparent importance at the time, and evolution cascades into a radically different channel." This means that there might be many different ways a species could have adapted to its environment; we only see one end result. To answer whether complexity always increases down all of these pathways, we must have an experimental system in which to test it.

Avida [*Ofria and Wilke*, 2004] is a software platform used to perform experiments in evolutionary biology. The Avida system creates an artificial environment that maintains a population of self-replicating computer programs. These populations are subject to mutations and are in environments with limited space and resources (sources of energy) for which they must compete, therefore the organisms evolve by natural selection. A population in Avida adapts in a manner analogous to biological systems, both to maximize its replication rate and to beneficially interact with its environment. When an individual program attempts to replicate, it is subjected to random mutations that change instructions within its memory. Mutations are classified in a strictly Darwinian sense: any mutation that results in an increased ability to reproduce in a given environment is considered beneficial. Mutations causing the organism to fail to reproduce successfully are considered lethal. Neutral mutations cause no change in reproductive success.

Each organism (Figure 1.1) in an Avida population consists of a memory initialized to the genomic program, three 32-bit registers, two stacks, and input and output buffers for organisms to receive operands and return results to the environment. The genome of an organism is composed of a Turing-complete programming language; that is, they can perform any computable mathematical function—no explicit limitations

are imposed on what can be evolved. Indeed, we have witnessed a wide variety of unexpected and seemingly clever adaptations arise through evolution in Avida.

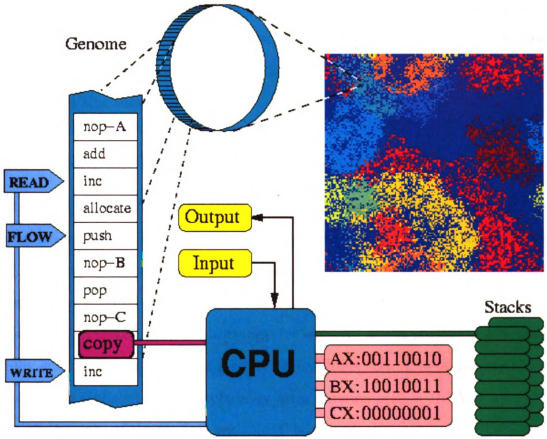

Figure 1.1: Structure of virtual CPU in Avida. Images in this thesis/dissertation are presented in color.

The phenotype of an organism corresponds to the set of computations an organism performs and related parameters such as how quickly it can perform each computation. Depending on the environment, an organism receives an energy bonus for performing specific computations. The fitness of the organism is then its total energy divided by its gestation time. Each update (the unit of time used in Avida) organisms receive a number of CPU cycles proportional to their energy. All organisms have their CPU cycles scheduled to execute their genomes in an order as close to parallel as possible.

Avida provides us with a system where the population dynamics can easily be explored and where we can trivially access the genome for any individual. Since the evolution in Avida is real, as opposed to a mere simulation, complex traits can arise on their own. Since the system also allows us to perform tests of genomes in isolation, it is a perfect choice for systematic studies of complexity, and we will revisit it throughout this thesis.

1.2.3 Shannon Information Theory

Information Theory [*Shannon*, 1948; *Cover and Thomas*, 1991] uses quantitative mathematics to formally define measures of disorder and uncertainty, which are then used, in turn, to define the information content of a message as the reduction of uncertainty attributed to the other message.

Information theory defines *uncertainty* (or *entropy*) as the number of bits needed to fully specify a situation, given a set of probabilities. Let X be a discrete random variable with alphabet $\mathcal{X}$ and probability mass function $p(x) = Pr(X = x)$, $x \in \mathcal{X}$. Thus, uncertainty of the random variable X is defined as:

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \tag{1.1}$$

Uncertainty is maximized when all probabilities are equal, that is, we have no idea about what the outcome will be. On the other hand, uncertainty is minimized when the probability of one symbol in alphabet $\mathcal{X}$ is 1 and the probabilities of each of the other symbols is 0.

Conditional entropy of one random variable given another is defined as the expected value of the entropies of the conditional probabilities, averaged over the con-

ditional random variable. Specifically:

$$H(X|Y) = \sum_{y \in \mathcal{Y}} H(X|Y=y) = \sum_{y \in \mathcal{Y}} p(y) \sum_{x \in \mathcal{X}} p(x|y) \log_2 p(x|y) \tag{1.2}$$

In Information Theory, the reduction in uncertainty of one random variable due to the knowledge of another random variable is called the mutual information between variables (also sometimes called mutual entropy).

$$I(X:Y) = H(X) - H(X|Y) \tag{1.3}$$

The mutual information $I(X:Y)$ is a measure of the dependence between two random variables. If variables X and Y are independent from each other, knowledge of Y won't decrease our uncertainty about variable X. In other words, uncertainty about X given Y ($H(X|Y)$) remains same as the uncertainty about X without knowing anything else, so the mutual information between X and Y is zero. Two variables can have a non-zero mutual information only if there is some correlation between them. The metric of mutual information is symmetric in X and Y, that is, $I(X:Y)$ is equal to $I(Y:X)$. Additionally, mutual information between a pair of variables is always non-negative, that is, knowledge of one variable can never, on average, increase our uncertainty about the other.

For a binary symmetric channel shown in Figure 1.2, the binary signal source inputs 0 or 1 with equal probability. The channel's output is equal to the input with the probability 3/4. On the other hand, with the probability of 1/4, a '0' is received when '1' is the input, and vice versa.

In this case, the entropy about the input X is:

$$H(X) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1 \tag{1.4}$$

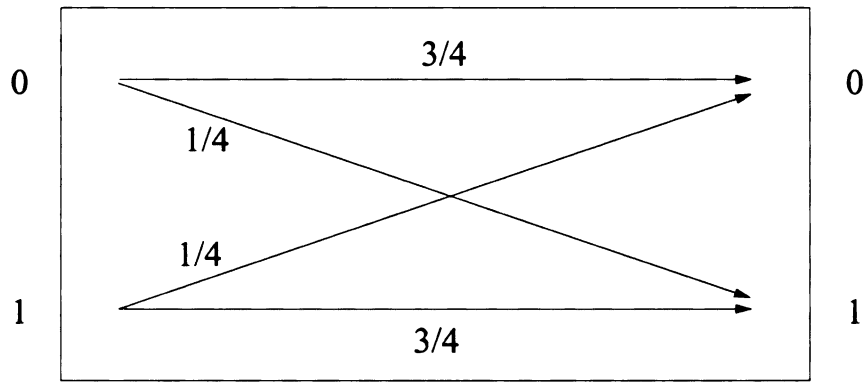

Figure 1.2: A symmetric binary channel.

Assume we received the output Y from the channel, the conditional entropy about input X is calculated as:

$$H(X|Y) = \frac{1}{2}(-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}) + \frac{1}{2}(-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}) = 0.8113 \quad (1.5)$$

The reduction of entropy due to the output Y is the mutual information between input X and output Y, which is

$$I(X:Y) = H(X) - H(X|Y) = 1 - 0.8113 = 0.1887 \quad (1.6)$$

Originally, information theory was used exclusively in telecommunications to maximize information transmission over a noisy channel. It is now used more frequently across many fields including biology [*Schneider*, 2000; *Adami et al.*, 2000]. In evolutionary biology, the replication from parent to offspring is thought of as an information transmission process. The information contained within a genome is about its environment and determines how the organism behaves in that environment; in particular, it determines whether or not the organism can replicate and how well it is able to survive. Mutations are responsible for the noise during the replication, and the quality of the resulting message will determine if a mutation is detrimental, neutral, or occasionally even beneficial.

Chapter 2

MEASURING COMPLEXITY WITH DIGITAL ORGANISMS

Adami, Ofria, and Collier [2000] put forth an elegant definition of biological complexity of an organism: the genetic information that an organism has about its environment. This builds upon Adami and Cerf's definition of the physical complexity of a symbolic string [2000] as well as Shannon Information Theory.

Adami, Ofria, and Collier developed a population-based method to measure biological complexity. They approximate the total information in a genome as the sum of the information at each locus. The information content of a locus is measured using information theoretic techniques on a population of organisms with the same phenotype. This population-based method for measuring the information content of a locus has inherent limitations such as requiring a full population at equilibrium, for genomes to be fixed-length, and for the environment to have only a single niche.

We have developed a significantly better method for measuring the biological complexity of an individual organism that overcomes many of these limitations. Our new method is still based on Shannon Information Theory, but we now harness the principle of mutation-selection balance from population genetics, which allow us extrapolate out to a full population from a single genome. Specifically, we determine the information content of each locus in the genome by testing all of the possible mutations at that position and measuring the expected frequencies of potential genes in the mutation-selection equilibrium state.

I will first introduce Adami et al.'s method in the next section and then further describe our new approach, its advantages, and initial results from applying this approach.

2.1 Population-Based Physical Complexity

Conceptually, the physical complexity of an organism is the amount of information that is stored in its genome about its environment. A genome stores information that is expressed into the functional capabilities of the organism in a given environment. Thus, the physical complexity of a genome or organism should mirror the functional capabilities or phenotype of the organism. Adami et al. use Shannon Information Theory in the following manner to relate phenotype to physical complexity.

If we know nothing about an organism, then we have maximal uncertainty about its genome (any genome is possible). On the other hand, if we know the organism's phenotype, we have less uncertainty about what the organism's genome is (only a small fraction of possible genomes corresponds to any specific phenotype). The difference between these uncertainties represents the information stored in the genome about its environment, and thus its physical complexity.

Unfortunately, it is difficult to exactly define the entropy or uncertainty of a genome given its phenotype. To approximate this uncertainty, Adami et al. proposed that most encodings of an identical phenotype would be similar to each other – typically only differing by a series of neutral mutations. Given this assumption, a large enough population where all individuals possess the same phenotype should contain the distribution of genomes necessary to calculate physical complexity. Unfortunately, it is difficult to get a "large enough" population, so Adami et al. showed that in most cases, it is sufficient to calculate the entropy of a population of genomes site by site. If there is no epistasis (non-linear interactions between genome positions), this will

give us the same result. This also has the beneficial side effect of identifying the sites within the genome that store the information.

To illustrate this technique, consider the case of approximating the physical complexity of a phenotype where the corresponding genome space is DNA sequences. We first need a population of genomes in an equilibrium state that all have this phenotype. We then calculate the per-site information of this population as follows: without any information about the genome, we can only assume that each of the four nucleotides has equal probability of occurring at any site i leading to a maximum site entropy of $H_{\max} = \log_2 4 = 2$. Let the frequencies for each nucleotide for site i within the actual population be $p_C(i)$, $p_G(i)$, $p_A(i)$, $p_T(i)$. The population entropy of this site is then

$$H_i = -\sum_{j}^{C,G,A,T} p_j(i) \log p_j(i) \tag{2.1}$$

The information content at site i would then be:

$$I(i) = H_{\max} - H_i = 2 - H_i \tag{2.2}$$

Finally, the physical complexity for this phenotype is approximated by applying this equation to each site and summing them together.

$$C = \sum_i I(i). \tag{2.3}$$

2.2 Limitations of Population-Based Complexity

In their previous study, Adami, Ofria, and Collier used the Avida platform to examine the evolution of physical complexity in digital organisms. Avida was setup in single-niche, mass-action mode. The single-niche aspect means that the organisms are all in direct competition against each other and the species with the highest fitness

phenotype will dominate. The fact that the population is mass-action means that there is no local structure to it so when a higher fitness species evolves it can to take over rapidly due to an exponential growth rate. Finally, the organisms were all forced to have the same genome length (100 sites) so that sequence alignment would not be necessary. During each experiment, they calculated the frequency of each instruction at each site by counting the number of organisms with that instruction in the site studied.

This population-based technique allowed for good estimates of physical complexity over time in many instances, but suffers from a number of limitations. First, the technique only produces accurate measurements if the population has reached an equilibrium state. For example, if a beneficial mutation causes a new species (and thus new phenotype) to take over a population, all otherwise neutral sites in the genome hitchhike to fixation, and it will take time before an equilibrium is reached representing most genotypes of the new phenotype. It is often the case that a new beneficial mutation will arise before equilibrium is reached preventing us from determining the true complexity of that phenotype. We can see these effects in Figure 2.1, the upper line of which displays the physical complexity over time using the population-based technique for a typical Avida experiment. Notice that each time complexity increases, it overshoots its mark and then gradually comes down again, typically to a higher resting level than it started.

The second problem with population-based physical complexity is the constraints that must be placed upon the organisms due to computational concerns. The population size must be small (typically 10,000 organisms or fewer) and the length of their genomes is fixed to prevent alignment problems. The finite population size limits the possible range of genotypes compounding the population diversity limitation noted earlier. If there are too many neutral sites, it would be unlikely for them all to be represented in such a limited population even at equilibrium. The fixed length genome

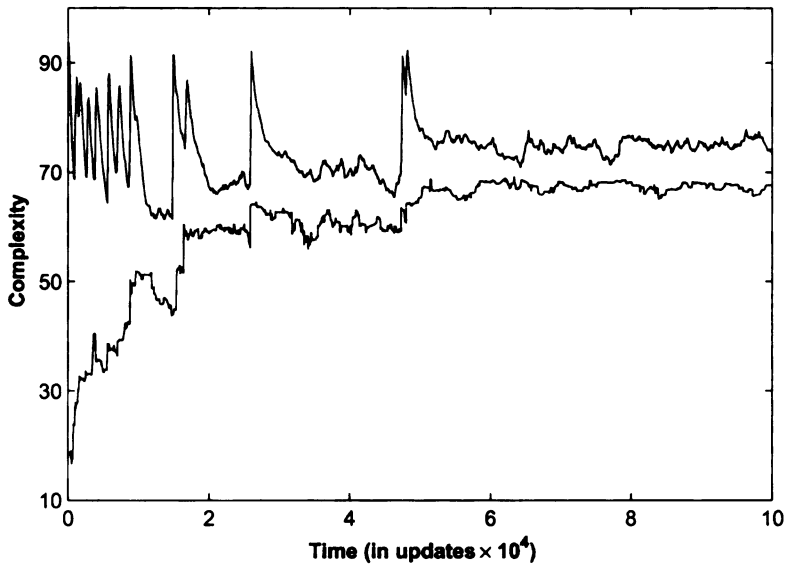

Figure 2.1: A comparison between the population-based method of calculating physical complexity (upper line) and our new method (lower line). The new method is applied to the lineage of the most abundant organism at the end of the experiment. The population size is 3600.

puts an inherent cap on complexity growth (there is only so much "blank tape" to write information into) and precludes many powerful forms of mutations such as gene duplications.

A final problem is ensuring that all organisms possess the same phenotype. This can only be achieved by using a single niche environment, and even with a single niche, a single phenotype only occurs at equilibrium. This limits the the range of interesting experiments that can be studied using population-based physical complexity, and excludes the possibility of studying ecosystem complexity. Ideally we want a process that will be able to calculate the complexity of a single organism given the rest of its environment as a constant.

2.3 Our Approach

Here, we demonstrate a new method for calculating physical complexity that has the ability to transcend the limitations listed above. We calculate the complexity of a single genome from its local fitness landscape by testing the fitness effects of all possible single-step mutants and using the principles of mutation-selection balance to calculate the expected frequency of each mutant. Since we calculate the complexity one organism at a time, we never have to worry about overshooting the correct complexity due to a biased sample, we do not impose any genome size limitations, and we can allow the environment to vary as long as we always test an organism using the state of the environment during its lifetime. In essence, we shift the complexity measure from the phenotypic level to the genotypic level.

This method does, however, suffer from limitations of its own. First, we only consider single-step mutants. In the future we plan to refine this method by examining multiple sites at once in an attempt to decipher epistatic interactions and improve our complexity measure. Second, a significant amount of extra processing power is required to generate all possible single point mutations from a genome and to test the fitness of each. In a computer this may be feasible, but in a natural system it is nearly impossible given our current technology. We must therefore limit ourselves to applying this physical complexity measurement technique to computational systems for the moment. This is not as severe a problem as it may seem since one of the main goals of quantifying complexity is to study its origin. In the natural world, evolution progresses too slowly to see significant changes to species in time spans shorter than centuries, so experimental macro-evolution already has this restriction placed on it. Furthermore, as we improve our techniques for calculating complexity in the digital world, we can use this to determine the quality of other complexity approximation algorithms that can more easily be applied to natural systems, in particular those that involve knockouts rather than all possible mutations.

2.4 The Technique

As our first step in calculating the physical complexity of Avida genomes, we developed a test environment that organisms can be inserted into. The test environment is initialized to the exact same conditions as the environment that the population is evolving in, but only one organism is tested at a time. That organism is processed until either it gives birth and we can measure its fitness, or else it dies of old age (indicating a zero fitness). Note that since the expected number of offspring per unit time will vary depending on local resources and competition, fitness can be highly dependent on the organism's environment. Once a test environment is constructed, any organism including its mutants can be placed in the test environment to determine its fitness in that exact state of the environment.

As in the previous method, we calculate the complexity of the whole genome by summing the complexities of the individual sites in that genome. To determine the complexity of site i, we start by mutating this site to all other possible states and then use the test environment to calculate the fitness of each. In the case of Avida, there are 26 instructions in the genetic alphabet, so we need to generate 25 new genomes to represent each possible mutation at site i. We then run each of the resulting 26 genomes through the test environment to determine their fitnesses. With these fitnesses and a mutation rate, we can predict the abundance of each instruction at this site were a population at equilibrium.

Intuitively, it is clear that if a genome has equal fitness no matter which instruction is at site i, then we would expect all possible instructions to appear with about equal frequency. Further, this would translate to a maximal entropy for that site, and thus a zero complexity. On the other hand, if only the original instruction has a non-zero fitness, then we expect that instruction to dominate in an equilibrium population (the others would persist at a small frequency due to detrimental mutations creating them.) In this case, the population would have a low entropy at this genomic position,

and it would contribute maximally to complexity. It is slightly more complicated to calculate the expected abundance at sites with mixed fitness levels; our techniques are discussed below.

As an example, we show a sample sub-sequence from a genome in Table 2.1 where a single site is mutated throughout. Its original state was 'm', but all others are tested as well and their fitnesses recorded.

Sequence	Fitness
...akapbkawbjbo**a**cpbnaqblafpq...	0
...akapbkawbjbo**b**cpbnaqblafpq...	6.46734
...akapbkawbjbo**c**cpbnaqblafpq...	0
...akapbkawbjbo**d**cpbnaqblafpq...	5.94
...	...
...akapbkawbjbo**m**cpbnaqblafpq...	6.46734
...	...
...akapbkawbjbo**z**cpbnaqblafpq...	3.23367

Table 2.1: Samples for genome sequences with all single-site mutations and the resulting fitness of each. The site being changed is marked in bold within each sequence above. The original symbol is "m".

We use the mutation-selection balance principle from population genetics to take the fitness values and determine the portion of the population that we expect each genotype to fill at equilibrium. Fisher, Haldane, and Wright, pioneers of population genetics, developed mathematical models quantifying the relative importance of selection and mutation in maintaining genetic variation. We simplify and specialize these equations to Avida, which has populations that are asexual, haploid, and have overlapping generations, and where we only consider the possibility of site i mutating, since we are not considering interactions between sites.

We need the following notation to define our mathematical model. Let p_j denote the percentage of the population occupied by genotype j at equilibrium, ω_j the fitness of genotype j, D the size of the instruction set (in our case $D = 26$), and μ the per-site mutation rate. Furthermore, we assume all mutations are equally probable. The

average fitness is defined by

$$\bar{\omega} = \sum_{k=1}^{D} p_k \omega_k \tag{2.4}$$

At equilibrium, the following equation must hold:

$$p_j = (p_j \omega_j / \bar{\omega})(1 - \mu) + \sum_{k=1}^{D} (p_k \omega_k / \bar{\omega}) \mu (1/D) \tag{2.5}$$

In this equation, $p_j \omega_j / \bar{\omega}$ is the relative replication rate of genotype j, and $1 - \mu$ is the probability that genotype j replicates without mutation at site i. These two factors are multiplied together to give us the rate of perfect replication within genotype j. For the second part of the equation, $\mu(1/D)$ is the probability that any genotype (including j) mutates into genotype j. We then multiply this by the relative replication rate for each genotype to determine the rate that each genotype mutates to genotype j. These two factors summed together represent the rate at which genotype j enters the population. Since all organisms leave the population with equal probability, at equilibrium, the rate at which genotype j enters the population must be the same as p_j, the percentage of the population occupied by genotype j at equilibrium. We use equation 2.4 to simplify equation 2.5 as:

$$p_j = (p_j \omega_j / \bar{\omega})(1 - \mu) + \mu(1/D) \tag{2.6}$$

To determine the final abundance of each of the 26 genotypes at equilibrium, we generate the 26 equations and solve them. This will always provide us with a unique solution if we bound all probabilities to be between zero and one. The solution will predict the abundance of each possible instruction at this site in an infinite population—exactly what we need. We can then calculate the physical complexity of this site and repeat this process for each other site in the genome, summing them up to determine the physical complexity of the genome as a whole.

There is only one modification that we need to make to this process to help us determine the correct physical complexity for genomes along a lineage that immediately precedes an adaptive event. It is possible that a single-step mutant will have a higher fitness and thus contain more information about the environment in its genome. We want to measure only the amount of information currently in the genome, so we treat any beneficial mutants as if they were neutral. That is, no single-step mutant is given a higher fitness than the original genome. In a population, we would expect this beneficial mutation to be selected for, but this information has not yet been incorporated into the test genome.

2.5 Experiments and Results

Our first experiments test how accurately our models predict the abundance of single-step mutants. We initiate Avida experiments with a population size of 3600 where only a single site is allowed to mutate. Given our instruction set of 26, there can only be 26 possible genotypes in the population. We then compare our predicted abundances to the observed abundances once an equilibrium is reached as shown in Figure 2.2. We have performed over 30 such comparisons, and all performed similarly well.

Our second set of experiments highlight the improved accuracy of our new method when a population is not at equilibrium compared with the previous population-based method for calculating complexity. In order to be able to calculate the population-based complexity, we perform Avida experiments with large (3600), single-niche populations with fixed-length genomes. Within each experiment, we first identify the lineage of the most abundant genotype at the end of the experiment. At each Avida update, we then compute both the population-based complexity (using all genotypes alive at that update) and our single-step mutant complexity of the current genotype

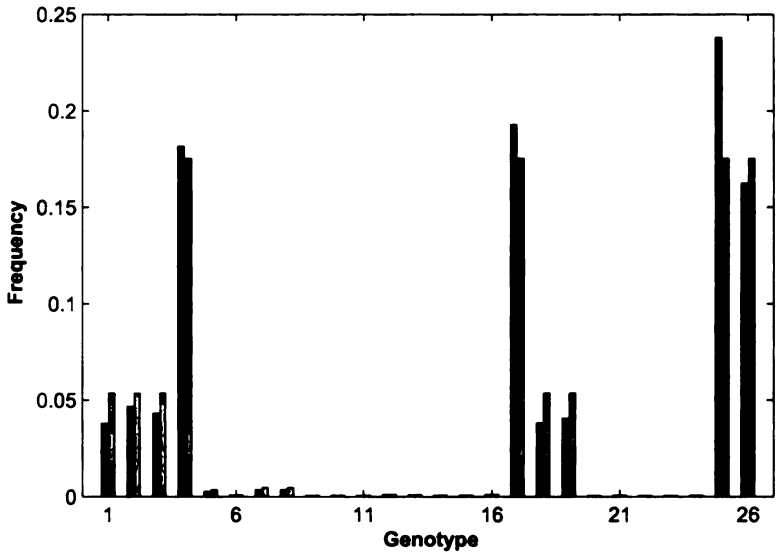

Figure 2.2: Mutation-selection balance at site 53 of a genome. Gray bars are mathematical predictions of instruction frequency, black bars are experimental results.

on the isolated lineage. Figure 2.1 contains a plot of both complexity measures at each Avida update for one Avida experiment.

As we saw in Figure 2.1, when the population is not at equilibrium, the population-based complexity is inaccurate for many of the updates as it suffers from hitch-hiking effects. On the other hand, the proposed growth in complexity over time (with minor fluctuations) is clearly visible in the single-step mutant complexity (lower line in Figure 2.1). The minor fluctuations in complexity are expected; there are occasional decreases due to the loss of information, or the evolution of a more compressed way to code for a phenotypic trait. Some fluctuations may also be caused by inaccurate approximations of the actual complexity due to the fact that we do not yet account for epistasis. At equilibrium, both methods provide a qualitatively similar result, though the population-based complexity measure is always higher than the single-step mutant complexity.

Our third set of experiments highlight the improved accuracy of our new method for calculating complexity when a population is at equilibrium compared with the pre-

vious population-based method. We perform 30 sets of Avida experiments (identical to those performed in our second set of experiments) with three different population sizes: 900, 3600, and 14,400. We focus on the comparison between the computed complexities once the population reaches a final equilibrium. We sampled one trial with each population size. Figure 2.3 shows the comparison of two methods for the population with size 900 and 14,400 and Figure 2.1 shows the comparison for the population size 3600. As the population size increases, we found the population-based complexity become closer to the single-step mutant complexity at equilibrium. The mean difference over 30 trials between complexity measures at equilibrium in the size 900 population experiments was 21.35 Complexity Units (CU), where 1 CU = 1 instruction containing maximal information. The mean difference between complexity measures at equilibrium in the size 3600 population experiments was 10.04 CU. Finally, the mean difference between complexity measures at equilibrium in the size 14,400 population experiments was a nearly negligible 1.61 CU. Clearly, as population size increases, these measures become dramatically closer.

These results can be easily explained. In our single-step mutant method, we simulate an infinite-sized population. The population-based method, on the other hand uses a finite-sized population, where some single-step mutants may not be available making entropy appear lower than it should; this, in turn, leads to an overestimate of complexity. As the population size increases, the accuracy of the population-based method should improve, and the results should approach those of the single-step mutant method. As our experiments show, this pattern holds, thus indicating our new method is more accurate than the population-based method even under optimal conditions for the population-based method.

Our final experiments demonstrate the increased flexibility of our new method for calculating physical complexity. In particular, we show that it can be applied to variable length genomes. We perform 50 Avida experiments with a population size

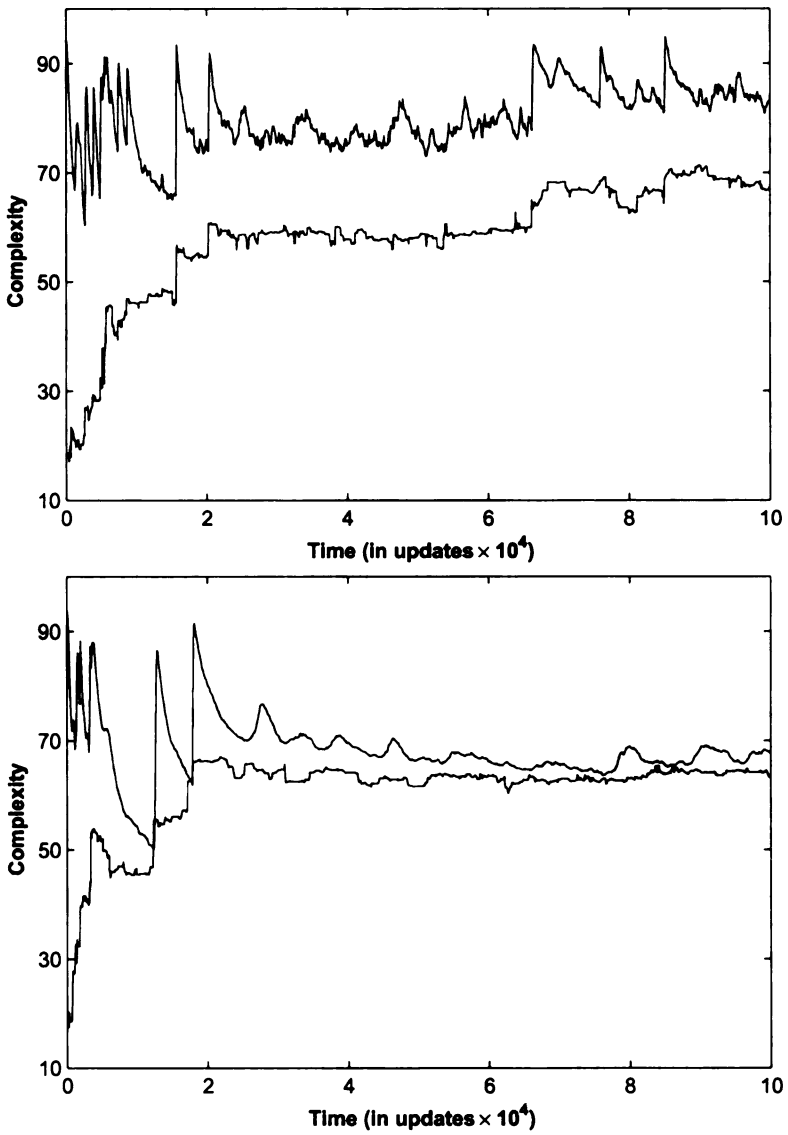

Figure 2.3: A comparison between the population-based method of calculating physical complexity (upper line) and our new method (lower line). The new method is applied to the lineage of the most abundant organism at the end of the experiment. The population size of two figures are 900 and 14,400.

of 10,000 organisms where we allow the size of the genome to change. Figure 2.4(a) is the result from a sample trial. It displays our new physical complexity measure of a lineage from this experiment. There are many clear jumps in complexity over time. Figure 2.4(b) shows how fitness changes for the same lineage and Figure 2.4(c) shows how the complexity jumps correlate with fitness increases. The correlation coefficient between the complexity and the log fitness is 0.979 for this case study; the correlation coefficient averaged over 50 trials is 0.920 with standard deviation 0.055. This suggests that we are accurately reflecting the true complexity as at each fitness jump, more information about the environment is encoded into the genome. We also observed downward spikes occasionally occur in both graphs. Clearly the mutations corresponding to the loss of information are detrimental mutations that briefly exist along the lineage and can be an important step on the way to significant fitness improvements [*Lenski et al.*, 2003].

We averaged physical complexity over 50 trials of population size 10,000. The complexity shows the trend of increasing in Figure 2.5. According to the Natural Maxwell's Demon proposed by Adami et al. [2000], complexity should increase because natural selection "monitors" the mutations; whenever the mutation contains information about the environment, it is allowed to enter the genome. Hence, the total information in the genome increases during the evolution, which also means the complexity increases.

2.6 Discussion

We designed a point-mutation method to measure physical complexity and compared this method with the old population-based method. The new method is not limited by the population size; the population does not have to be in equilibrium state to be measured for complexity. The method can be used to measure variable length

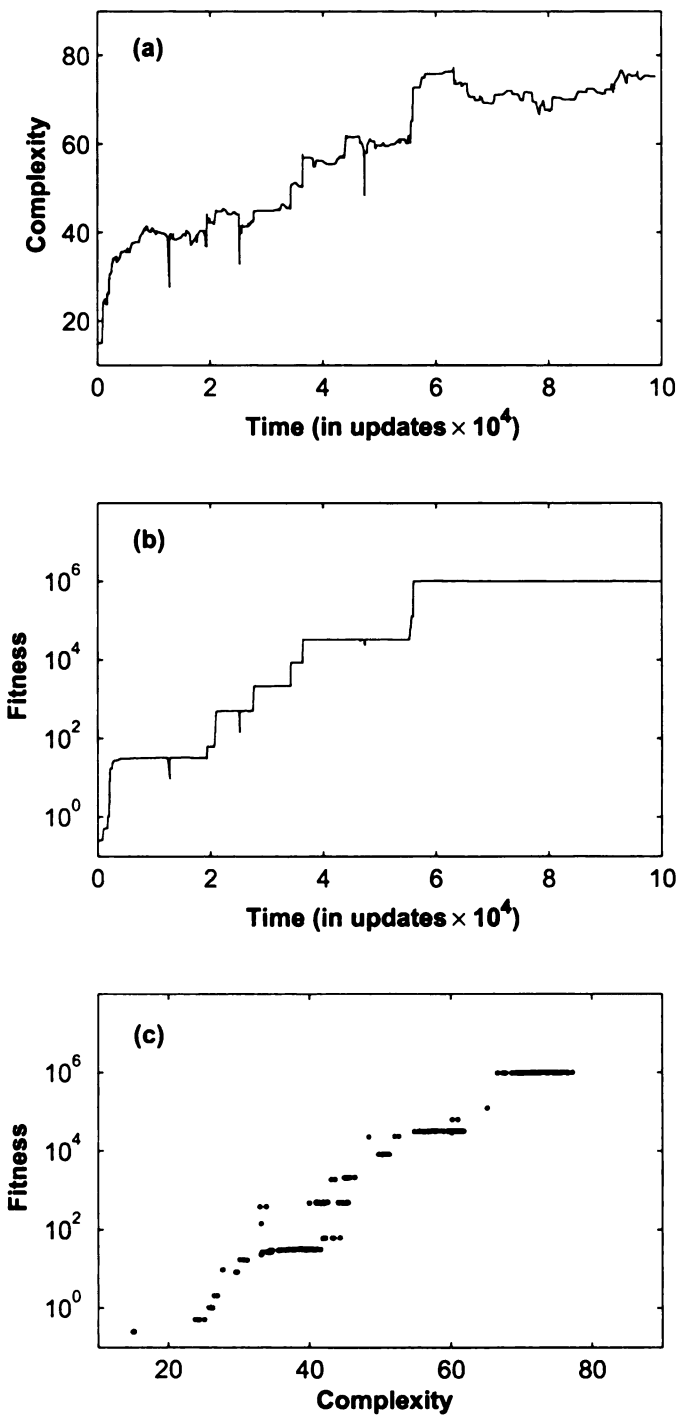

Figure 2.4: (a) Physical complexity over time for a lineage from an Avida experiment where genome length is allowed to change. (b) Fitness over time for the same lineage from the same experiment. (c) Fitness vs. complexity.

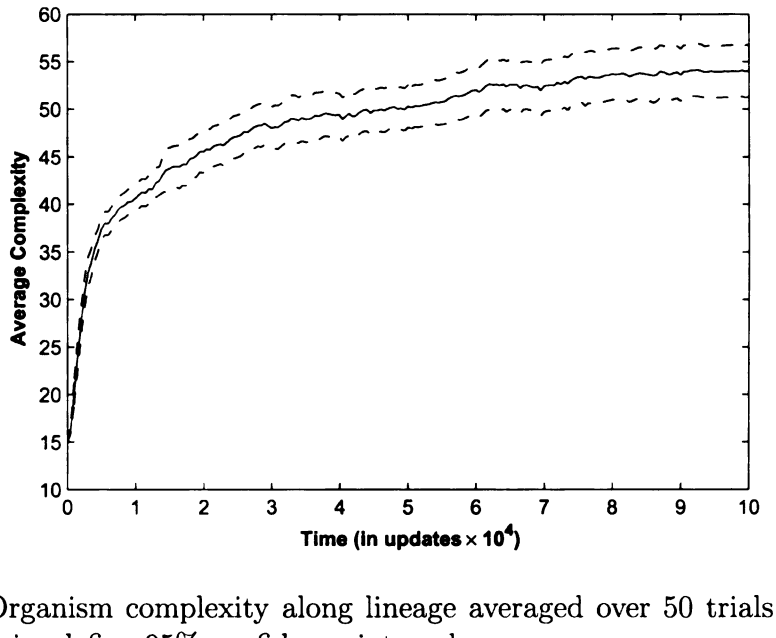

Figure 2.5: Organism complexity along lineage averaged over 50 trials. Upper and lower boundaries define 95% confidence interval.

genomes, so insertion and deletion mutations are allowed in the experiments. With our method, we are able to measure the organism's complexity under both single niche environment and multi-niche environment.

We performed experiments with digital organisms and analyzed the complexity changes of the organisms over evolution. The question whether complexity always has an increasing trend has been around for a long time. Since Darwinian evolution is a long-term procedure in nature that we only have one DNA-based example, it is nearly impossible to get a conclusive result from the natural world. In Avida, we are able to perform many experiments and observe macro-evolutionary dynamics so we can study how this process works. The data we have collected thus far have concurred that complexity does seem to always increase over time. These experiments, as well as the theory that led to them, assume a single-niche environment.

Initial experiments in environments with multiple, limited resources show that many species can easily co-exist in an Avida population and form primitive ecosystems [*Cooper and Ofria*, 2002]. While theory does not dictate that populations in

such naturally fluctuating environments always increase in complexity, although we have observed a much more rapid fitness increase in these populations. It would be impossible to show that any single species must always gain in complexity, but there is much more that may be said about the community as a whole. In chapter 4, I will talk about how we extend the technique to examine the complexity in the community.

Chapter 3

ON THE EVOLUTION OF COMPLEX FEATURES

Evolutionary theory explains the origin of complex organismal features through a combination of reusing and extending information from less-complex traits, and by only needing to exploit one of many possible pathways to a viable solution. While the appearance of a new trait may seem sudden, the underlying information associated with that trait must evolve gradually. To study this process we used digital organisms in Avida. Lenski et al. [2003] found that when a new complex trait first appears in a population of digital organisms, its proper function immediately requires the coordinated operation of many genomic positions. However, we show here that the total information stored in the genome only increases marginally. As the amount of information needed to perform a trait increases, the probability of its simultaneous introduction drops exponentially, so a truly complex trait appearing as a whole *de novo* is virtually impossible. We demonstrate that the majority of the information associated with a trait is primarily correlated with pre-existing traits or is co-opted from traits that were lost in conjunction with the appearance of the new trait. Thus, while total information in a genome only increases in small increments, traits that require much more information can still arise during the normal evolutionary process.

In this chapter, I first introduce previous research on complex traits, then describe our method of measuring genomic complexity and the experiment system we use. Finally I show analysis results and make conclusions.

3.1 Background

In 1871, the zoologist St George Mivart objected to Darwin's theory of evolution by natural selection on the grounds that, while it could optimize existing features of an organism, it could not explain the origin of entirely new traits [*Mivart*, 1871]. Specifically, Mivart felt that incipient forms on the way to produce useful structures served no purpose, and hence would not be supported by natural selection. Darwin had anticipated this difficulty, and in his first edition of *On the Origin of Species* [*Darwin*, 1859] noted that "it is so important to bear in mind the probability of conversion from one function to another" and furthermore, that "different kinds of modification would serve for the same general purpose." In other words, Darwin proposed that new traits arise due to functional shifts from previously existing traits, and that even though any specific modification may be unlikely to evolve, many different pathways may all lead toward the same goal; even if the outcome we witness seems incredibly unlikely, it was only one of many possibilities.

Substantial evidence has been collected indicating that complex traits can be produced through the evolutionary process, including such examples as the evolution of the eye [*Salvini-Plawen and Mayr*, 1977; *Goldsmith*, 1990; *Nilsson and Pelger*, 1994], the Krebs cycle [*Meléndez-Hevia et al.*, 1996], insecticide resistance [*Newcomb et al.*, 1997] and nutritive "milk" in the cockroach [*Williforda et al.*, 2004]. A detailed demonstration of the evolution of complex traits has even been previously performed in digital organisms [*Lenski et al.*, 2003].

The question remains as to how information flows into the genome. If a new trait must arise entirely, or even mostly, *de novo* then the probability of this trait to appear drops exponentially as the number of genomic positions it requires increases. Even considering that portions of the information required to express a new trait come from existing traits (that may be destroyed in the process) all of the unique information associated with the new trait must arise without the benefit of selection for the

incipient forms. Thus a new complex trait can only arise when the majority of its information has already made it into the genome through the presence of preexisting traits, typically of lesser complexity.

3.2 Measuring Genomic Complexity

We use information theory to frame our analysis of biological complexity by measuring the amount of information associated with each trait, and how much information is shared between traits. In general, an organism can be thought of as an information channel [*Ofria et al.*, 2003] that passes a message (its genome) to a recipient (its offspring's genome) in the presence of noise (mutations). The key difference here from traditional studies of information theory is a feedback loop: the message being passed will be used to build the next organism, which, in turn, will continue to pass the message on. Any flaws in the information transmitted may reduce the capacity of the subsequent channel. However, errors also have a small probability of being beneficial, and improving the quality of the offspring.

In a typical population, organisms will be subject to an approximately uniform mutation rate across their genomes. Positions that contain no information can mutate freely, while those that store information important to the organism's survival will typically be perfectly conserved. Here we use the technique from Chapter 2 to get the distribution of symbols at each position and estimate the amount of information stored. The sum of the information at each site determine the genomic information.

3.3 Experimental System

To study what happens to information in genomes during the evolutionary acquisition of a complex trait, we must use a system where we can isolate organisms associated with the adaptive event and then fully manipulate them to measure the information

content of their genomes. This requires a level of knowledge about the state of the system that is unprecedented in research with natural organisms, but can be easily obtained with digital organisms in Avida.

In all of the Avida experiments presented here, we provide unlimited resources to the organisms and space is the only limiting factor that the organisms must compete over. Additional CPU cycles allow an organism to execute their genome more rapidly, and therefore increase their replication rate. In the default Avida environment used here, nine resources are available, each associated with a different Boolean-logic operation. The most complex of these is called EQU, which we will focus this study around.

We can copy digital organisms into a separate, isolated environment to test them without influencing the course of evolution in the main population. In particular, in order to measure the amount of information contained at a genomic position, we perform every possible point mutation at that site and measure the relative fitness as compared to the unmutated wild-type. If all modifications at a position are lethal, that position clearly contains the maximum possible information. This translates to a value of 1 Complexity Unit (CU). If all changes to the site are neutral, the site clearly contains no information (or 0 CU). For intermediate ranges, we can use the fitness values of each mutant to calculate their expected relative abundance as described in Chapter 2.

We can also measure genomic information in an altered environment. For example, if we wish to measure the amount of unique information associated with the EQU trait, we calculate an organism's information content in the default environment, and then recalculate it in an environment that lacks the EQU resource. Losing the ability to perform EQU in the latter environment will not be disadvantageous, and hence mutations at sites associated with this trait will no longer be detrimental unless that site's information was shared with a secondary trait. The difference between

these two measures is the amount of information uniquely associated with the EQU-metabolizing trait.

3.4 Experiments and Results

We performed a set of 50 Avida trials using a default configuration. Specifically, we used population sizes of 3600 organisms, a per-site mutation rate of 0.0025, and a genome-level insertion and deletion rate of 0.05. We used an ancestor with a length 100 genome, which was only capable of self-replication but no other functional traits. The organisms evolved in an environment with unlimited resources associated with all nine basic bitwise logic operations. We then measured the complexity of organisms in four different environments.

3.4.1 Three sources of complexity

We tested the complexity of the organisms in four different environments to differentiate the sources of the complexity. *Environment I* is the default environment with unlimited resources for all 9 logic operations. Measuring the information content in this environment gives us the total complexity for an organism. *Environment II* is identical to Environment I, but without the resource associated with the complex trait EQU. The difference between an organism's complexity in environments I and II provides us with the amount of information uniquely associated with performance of the EQU trait. *Environment III* contains only the EQU resource, but none of the others and *Environment IV* contains no resources at all. The difference between complexity measures in environments III and IV results in the total amount of information used to perform EQU, even if this information is also associated with other traits.

Figure 3.1 shows these information measures for a typical Avida run. In this

example, the first organism in the population to possess the EQU trait appeared at update 24,891. While this organism did gain EQU (and, as such, possessed 4.9 CU of additional complexity over its parent), it simultaneously lost the ability to perform AND. When tested in an environment where EQU was not rewarded, the information content of the genome actually dropped by 9.9 CU that had been converted for use by EQU. In other words, the genome contained a total of 9.9 CU of complexity unique to EQU. This complexity combined with another 23.0 CU shared with other traits make a total of 32.9 CU required for EQU.

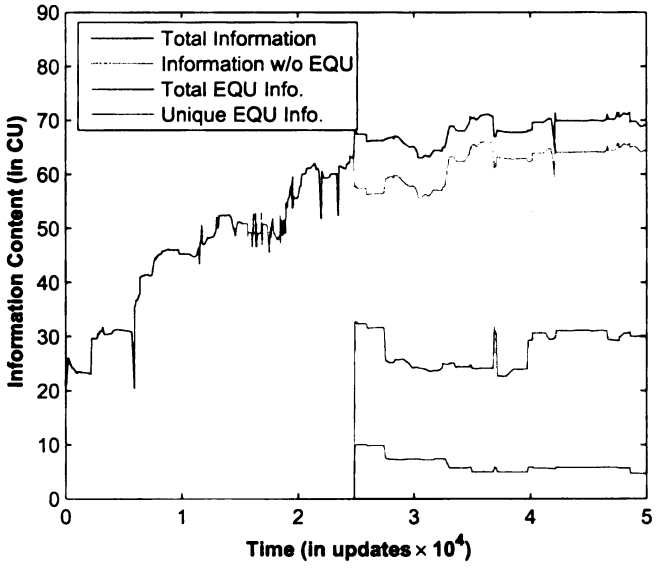

Figure 3.1: Information content over the course of a typical Avida experiment. The most abundant genotype was chosen from the final population, and the information content was measured for all organisms along its line of descent.

A total of 24 out of the 50 trials obtained the EQU trait. Figure 3.2 displays the average information content of genomes from these lineages, centered on update zero as the point where EQU is first acquired. From this figure, it is clear that the information associated with EQU comes from three different sources. The majority of the information (72.3%) is shared with other traits. The remainder is split between newly incorporated information (5.7%) and information used that was once part of now defunct traits (22.0%). The total information for task EQU is 26.8 CU averaged

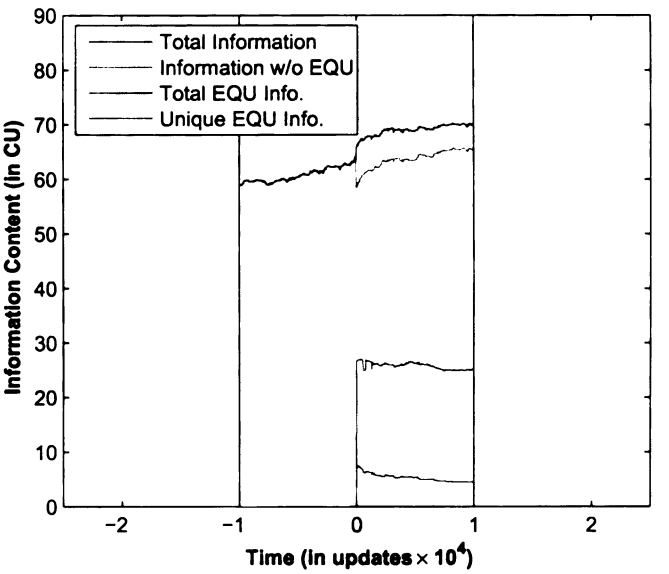

Figure 3.2: Average information during the acquisition of EQU. 24 trials were centered on their acquisition of EQU and had the information values averaged to create this graph. A window size is 10k updates around the time when EQU is first acquired.

over 24 trials. Thus, the mean information shared with other traits is 19.4 CU; the mean newly incorporated information is 1.5 CU and the mean information coming from now defunct traits is 5.9 CU.

The relative importance of these information sources varies widely from one trial to the next. To better understand the breakdown of where information comes from, we have created a histogram isolating the sources, as seen in Figure 3.3. Surprisingly, the change in organism complexity when EQU first arises was negative in three of our 24 cases (Figure 3.3a). This occurs when the traits lost during the acquisition of EQU actually had a greater combined complexity than the complexity of the EQU trait itself. In all three cases, the complexity was restored (through the reacquisition of lost traits) shortly after the rise of EQU. It is also unexpected that the complexity of the organisms can go up when EQU first appears in the environment where EQU is not rewarded (Figure 3.3c). This is due to other traits arising simultaneously with EQU, or to new material in EQU now being required for other traits to function properly. Figure 3.3b shows that the unique information associated with the EQU

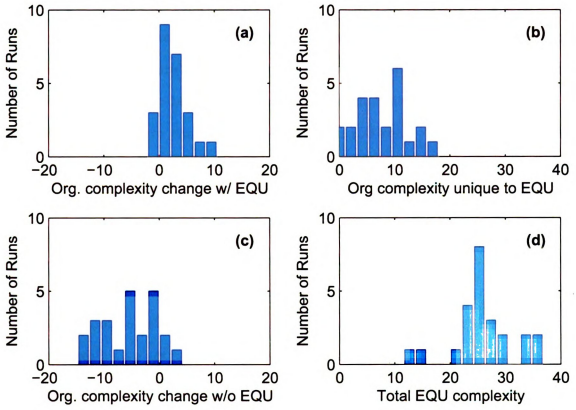

Figure 3.3: Histograms indicating the distribution of information in the 24 trials where organisms developed the EQU trait. (a) is the change in complexity for the whole organism when EQU first arises; (b) is the complexity unique to EQU, not shared by other traits; (c) is the change in complexity of all traits other than EQU at the time EQU arises; and (d) is the total complexity of the EQU trait, including information shared with other traits.

task. The range of this unique information is relatively large, from 0 to 16.7 CU. The total information associated with EQU differs among organisms, ranging from 12.8 to 35.3 CU (Figure 3.3d).

3.4.2 The distribution of complexity changes from a random beneficial mutation

To further examine the amount of information that can appear *de novo* in association with a beneficial mutation, we performed a set of experiments at three different time points in our 50 evolved populations (1k updates, 10k updates and 100k updates). At

each time point, we chose 2000 random organisms (with replacement) from each of the 50 populations to mutate. In each case we examined the complexity both before and after the mutation. Figure 3.4(a), (b), and (c) show the distribution of the complexity changes associated with beneficial mutations at each time point. While a total of 100,000 mutations were examined for each graph (50 runs $\times$ 2000 mutation tests per run), only a small fraction of them were beneficial. At 1000 updates, there were 18,675 beneficial mutations, at 10k updates, there were 5500 beneficial mutations and at 100k updates there were 2473 beneficial mutations. Over time populations became better adapted to the environment reducing the amount of information left to incorporate, which explains the decline in the number of beneficial mutations.

The distribution of complexity increases (i.e. where complexity change is greater than zero) exhibit a clear exponential distribution in both of the earlier time points, as shown in Figures 3.4(a) and (b). Every instruction which adds to complexity should have the same probability of occurring, meaning that more complex structures are exponentially less likely. This leads to the exponential distribution observed, as long as a sufficiently rich environment exists.

Figure 3.4(c) indicates a disproportionately high probability of obtaining a larger complexity increase at the later time point of evolution. This is largely due to the organisms becoming so well adapted to the environment that there is little room for improvement; the mean number of tasks evolved over 50 runs at 1000 updates is 0.1, at 10,000 updates it is 5.6, and at 100,000 updates it is 7.9 out of the possible 9 tasks. In other words, at 100,000 updates very few traits are still available to be acquired leading to this effect.

To account for the effects of simpler tasks on the continued evolution of complexity, we performed a set of control runs where the organisms evolve in an environment where the logic operations are not rewarded. We then analyzed the complexity changes from mutations (as above procedure) when these organisms and their mutants

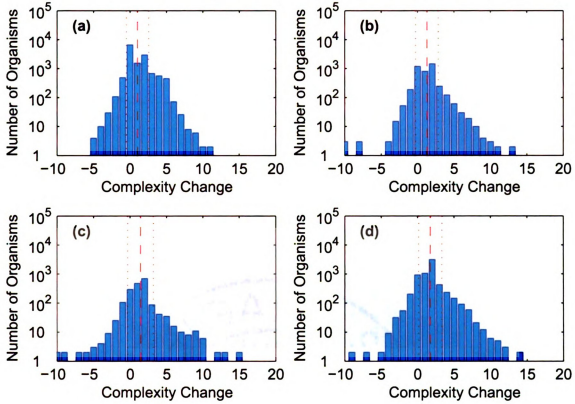

Figure 3.4: The distribution of complexity changes with beneficial mutations from 100,000 random sampled organisms in the populations (a) at 1000 updates, (b) at 10,000 updates, (c) at 100,000 updates under a 9 task environment, and (d) at 10,000 updates of control runs. Red dashed line represents the mean complexity changes and red dotted lines are one standard deviation from the mean.

were moved into the 9-resource environment. Figure 3.4(d) shows the distribution of these complexity effects at update 10,000. We again see a similar exponential distribution of positive complexity changes. This indicates that the absence of building blocks does not limit the amount of *new* complexity that comes into the genome; however, without building blocks it becomes nearly impossible to evolve more complex traits.

To improve our understanding of how complexity enters the genome, we need to focus on those mutations that provide a selective advantage and persist over evolutionary time scales. Clearly mutations that lead to beneficial traits will be selected for. Thus we expect mutations that have fixed in the population to provide more

complexity, on average, than random mutations (even random beneficial mutations). This is clearly shown in Figure 3.5 where we examine the distribution of complexity changes in lineages in the 9-resource environment.

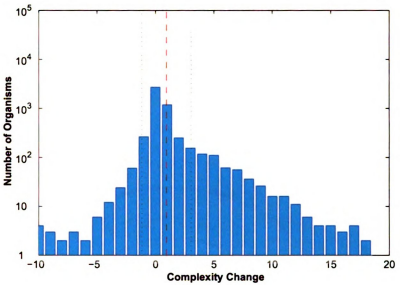

Figure 3.5: The distribution of complexity changes from a beneficial mutation along the lineages of 50 runs in the 9-task environment. Red dashed line represents the mean complexity changes and red dotted lines are one standard deviation from the mean.

3.5 Conclusions

We have demonstrated that the introduction of complexity into a genome has three aspects: complexity shared with other traits; complexity from once functional but now defunct traits; and complexity belonging to newly incorporated information.

We performed random mutation tests on populations at different time points. Based on the distribution of complexity changes due to beneficial mutations, we see that mutations leading to large complexity changes are exponentially rare. In particular, we see that the complex trait EQU requires on average 26.2 CU, but we rarely see more than half of this complexity (13.1 CU), appear in a single mutational step. If a complex trait such as EQU is to evolve, it must utilize preexisting complexity.

The experiments in this chapter were performed in relatively simple digital organisms, as compared to life in the natural world where there are far more than just 9 resources for the organisms to interact with. Note that every new trait that appears provides new building blocks to work with, which, in turn, increase the probability for for even more complex adaptation. Dramatically more complex traits can emerge in the natural world under such gradual increases in organismal complexity.

Chapter 4

BIOLOGICAL COMPLEXITY OF COMMUNITIES

Community complexity is an important topic in ecological biology. Traditionally biologists conflate the biological diversity of a community with that community's complexity. Biological diversity, in a broad sense, refers to the number of species in a community and evenness in abundance among those species [*Heywood*, 1995; *Magurran*, 2003]. Using these diversity methods to measure complexity, however, fails to associate the complexity of a community with the complexity of the individual organisms that live in it. For example, a community with a single complex individual may be more complex than a diverse community of very simple organisms.

In this chapter, I present a novel definition of community complexity — *the sum of the distinct information contained in the organisms of that community about the environment they live in* – that is a natural generalization of Adami et al.'s biological complexity of an individual. This definition of community complexity, like traditional diversity measures, increases as the number of species increases. However, rather than treating all species equally, each species contribution to the community complexity is based upon the unique information that species has about its environment. We present a specific method for approximately measuring community complexity that uses Shannon Information Theory to quantify the interactions between individuals and the environment.

The organization of this chapter is as follows: I will introduce the background in

4.1, define biological complexity of communities in section 4.2, describe the measuring technique in section 4.3, show experimental tests in 4.4, compare it with traditional measurements and discuss why our definition is valuable in section 4.5.

4.1 Biological Diversity

There are many methods to measure population diversity, among the most common are the Shannon index and the Simpson index. They both provide a single statistic that incorporate both the variety of the species and the relative abundance of them.

The Shannon index is based on Shannon Information Theory and measures the entropy of the population [*Pielou*, 1969]. The probability that a randomly selected individual will belong to the class S_i is p_i, where $\sum_i p_i = 1$. The diversity of the community is defined as:

$$H(p_1, p_2, ..., p_n) = -\sum_{i=1}^{n} p_i \log p_i \tag{4.1}$$

In Information Theory, this number represents per symbol entropy of a code composed of n different symbols. Put in the ecological context, H measures the uncertainty about the species of a random individual. In other words, if there are a large number of species in the community and distributions among species is close to even, there will be a high uncertainty about the species each individual belongs to. Although Shannon entropy is based on the assumption of infinite population size, when population size is big enough, it provides an accurate estimation. For example, Anand [1996] and Ray [1994] use this method to analyze the complexity of a plant community and a Tierra digital community.

The Simpson index (1949) [*Pielou*, 1969] was calculated as follows. If we draw two individuals at random and without replacement from an S species community,

the probability that both individuals belong to the same species is $\sum_i \frac{N_i(N_i-1)}{N(N-1)}$; each species has N_i individuals and the total size of community is N. If N_i is near N for some species, the diversity of the community would be low. So Simpson's index D is one minus this probability:

$$D = 1 - \sum_i \frac{N_i(N_i-1)}{N(N-1)} \tag{4.2}$$

4.1.1 Limitations of using diversity to measure complexity

Both the Shannon index and the Simpson index use two measures to calculate diversity. An increase of index may arise either due to higher number of species, greater evenness in the abundance of species, or both. To disentangle these components, Buzas et al. realize that the Shannon index can be separated into two parts: the natural log of the measure of evenness $E = e^H/S$ and the natural log of the number of species.

$$H = lnE + lnS \tag{4.3}$$

While these two traits are both highly relevant to diversity, they are not sufficient to fully represent the complexity of a community. For example, while a bacterial community and an animal community may have same number of species and same abundance distribution among the species, can we say that the two communities are equally complex? The factor we are failing to consider is the complexity of the individual organisms in the community. Our new definition of biological complexity for a community is based on the measurements of complexity for individual organisms that we introduced in chapter 2.

4.2 Biological Complexity of a Community

Adami et al. [2000] define biological complexity of an individual organism as the physical information saved in that organism's genomes about the environment it is living in. They point out that "genomic complexity is mirrored in functional complexity and vice versa. ... Genomic complexity can be defined in a consistent information-theoretic manner. [This complexity,] roughly speaking, reflects the number of base pairs in a sequence that are functional."

We extend the definition of biological complexity from the individual level to the community level. We define community complexity as the sum of the unique information contained in the organisms' genomes within a community about their environment. That is, if the same information is contained in two or more organisms' genomes, that information is included only once in the summation. If there are n organisms in the community, let us arbitrarily order them from 1 to n. We can use the notation from Information Theory to represent our definition of community complexity as $I(org_1, org_2, ..., org_n : E)$. Given this ordering, the community complexity is the sum of the unique information of organism i relative to its predecessors, organism 1 through organism $i-1$, which we denote as $I(org_i : E|org_1, \ldots, org_{i-1})$. For example, the new information in the third organism not contained in organisms 1 or 2 is $I(org_3 : E|org_1, org_2)$. Thus, the total distinct information in the community is:

$$\begin{aligned} I(community : E) &= I(org_1, org_2, ..., org_n : E) \\ &= I(org_1 : E) \\ &\quad + I(org_2 : E|org_1) \\ &\quad + I(org_3 : E|org_1, org_2) \\ &\quad + ... \\ &\quad + I(org_n : E|org_1, org_2, ..., org_{n-1}) \end{aligned} \tag{4.4}$$

There are two factors that are important for measuring community complexity: the number of species and the average unique information in each species. Our measure ignores the evenness of species; if information is present in a population, it is counted exactly once, no matter how many individuals possess that information. Organisms that belong to same species contain similar information about the environment, which they inherited from same common ancestor. Thus there tends to be minimal unique information in one organism compared to others in same species. The organisms belonging to different species usually have a lot of independently evolved information. Hence, when we compare the information content of those organisms, there will be much more unique information. Community level complexity increases due to that unique information explaining why communities with more species will maintain a higher complexity.

4.3 Measuring Method

As with the method for measuring biological complexity for individual organisms, we do not yet possess the technological power to measure the mutual information between organisms perfectly and therefore we cannot isolate the unique information that an individual possess about its environment. We must resort to using simplifying assumptions and approximations.

The first term in formula 4.5 is the complexity of an individual organism in Adami's definition. Adami et al. [2000] proposed a method to measure it in a single niche environment. Based on our new technique introduced in chapter 2, we evaluate the information content of each site through point mutations and the mutation-selection balance principle. Using this technique we are able to measure the complexity of individuals in both a single niche environment and multi-niche environment.

To simplify the calculations at this early stage of our research, we will use only

the organism that has highest correlation with our test organism (let us call it Y) to determine the unique information in Y given all previously measured organisms. We will call this comparison organism X,

$$I(Y|E, org_1, org_2, ..., Y-1) \approx I(Y|E, X) \tag{4.5}$$

The reasoning behind this approximation is that when parallel genomic evolution is rare (as this is the case in most evolving systems), correlated organisms will be close to each other in the true phylogeny and thus share the most information.

We calculate the correlations between two organisms by comparing the probability distribution of symbols site by site. Since alignment is not the key point in this thesis, we assume that all sequences have equal length and do not need to be aligned. We use capital letters for organisms and lower case letters for sites in the organisms. We generate all possible point mutations for each site. We then put each mutant in the environment, and test their fitness. Here, we make the assumption that for any site, all the mutations that make the mutant organisms have equal or higher fitness than the original organism have equal probability of occurring. We further assume that all deleterious and lethal mutations will make the organism soon be purged from the population, and therefor have a zero probability of occurring at that site. For example, if nucleotide T is the original symbol at site i and a point mutation to nucleotide C maintains the organism at the same fitness, but point mutations to nucleotides A or G make the organism have lower fitness, the probability of nucleotide C,T,A,G for the site i is:

0.5, 0.5, 0.0, 0.0

Similarly, we can get the distribution of nucleotides at site i of a second organism. If it is a neutral site, any nucleotide has equal probability:

0.25, 0.25, 0.25, 0.25

We draw the bar graph shown in Figure 4.1 for both distributions, the overlapping area is defined as the "overlap fraction" r. If two distributions are exactly same, the overlap fraction is 1. On the other hand, if two distributions have no overlap at all, the overlap fraction is 0. In the aforementioned case, the overlap fraction is *0.5*. The total overlap fraction summed over each site is used to represent the correlation level between two organisms. It is obvious that this sum is between 0 and the full genome length. We compare the new organism with each given organism and choose the one that is most correlated with the new organism. We use this chosen organism as the given condition in our calculation of the unique information in the new organism.

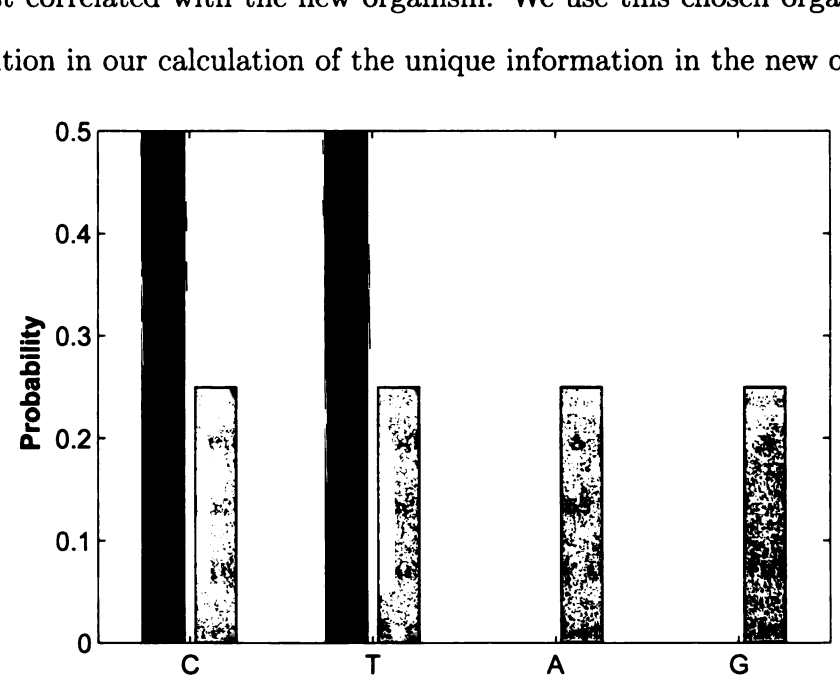

Figure 4.1: Nucleotide probability distribution of two comparison sites.

Following a similar approximation as the one used in Adami et al.'s paper [2000] and chapter 2, the unique information is calculated site by site. For each site i in organism Y, the new information in Y as compared to a previously measured organism X is defined as:

$$I(Y_i : E|X_i) = H(Y_i|X_i) - H(Y_i|E, X_i) \tag{4.6}$$

The first term $H(Y_i|X_i)$ is the conditional entropy of site i in organism Y given the possible symbol distribution of site i in organism X. In telecommunications,

conditional entropy is used as the entropy of a source message given the output message from the transmission channel, and a similar concept can be used in this case.

In evolutionary biology, an organism can be thought of as a channel that transmits its genome into its offspring (where mutations are noise in that channel). These channels can be concatenated over many generations to form a single channel that was responsible for transmitting information from an ancestor to its descendent. If there is no ancestor/descendent relation between two organisms, a channel can be thought of as a linking them through their most recent common ancestor.

The key difference between traditional information channels and those found in living organisms is that the information in a genome is a blueprint for building another organism. In other words, if the information is corrupted (i.e., a lethal mutation occurs), the message will not be transferred for additional generations. It is also possible for a message to be improved (beneficial mutations), changed without altering the information content (neutral mutations), or have a partial loss of information but still construct a mostly functional channel (deleterious mutations).

Without knowing the specific evolutionary history of each individual organism, it is impossible for us to determine which differences between organisms are due to a beneficial mutation (i.e., are the source of new information), and which ones are neutral or deleterious. This makes it difficult to calculate the amount of information that full knowledge about the genome of one organism gives us about another organism's genome. To overcome this difficulty, we designed a reasonable approach to estimate the entropy.

We utilize the overlap fraction r calculated above for each site to calculate the first term in equation 4.6. The entropy of site i in organism Y given the distribution

of corresponding site i in organism X is calculated as:

$$H(Y_i|X_i) = r * H(X_i) + (1-r) * H_{max} + H(r, 1-r) \tag{4.7}$$

The per-site entropy calculation depends on the degree of correlation between two sites. When two sites are overlapped with probability r, the entropy of the first site is decided by the entropy of the second site, that is, $H(X_i)$. When two sites are independent from each other with probability $1-r$, we have maximum uncertainty about the first site, which is 1. The last term $H(r, 1-r)$ represents the uncertainty we have about whether two sites are overlapped or not.

The second term in equation 4.6 is the conditional entropy of site i in organism Y given both the environment and organism X. In most situations, the entropy given two conditions is less than the entropy given one condition, $H(Y_i|E, X_i) \leq H(Y_i|E)$ because more information can never, on average, increase our uncertainty. In our special case organism X provides information about organism Y only in as much as X and Y both contain information about the same environment. Thus in equation 4.7 above, we needed X in $H(Y_i|X_i)$ to indirectly provide information about the environment and lower our uncertainty about Y. However, now that we are also given the environment, there is no additional information in X about Y, so we can make the simplification

$$H(Y_i|E, X_i) = H(Y_i|E) \tag{4.8}$$

The total new information in organism Y about the environment is calculated as the sum of new information at each site.

$$I(Y : E|X) = \sum_{i=1}^{l} I(Y_i : E|X_i) \tag{4.9}$$

where l is the genome length.

4.4 Experimental Setup

The absolute value of a complexity measurement is far less meaningful than its comparison to the same measurement performed on other communities, or when used to track the same community over time. My research interest is to analyze the impact of the environment on community complexity, particularly the changes in this measure during evolution. To perform these analyzes, we designed five environments with Avida.

In *Environment I* there are 9 tasks and no limitation on resources. It is a single niche environment, that is, only one species can persist in the environment over long time scales. *Environment II* also contains 9 tasks but there is a resource limitation for each task allowing the persistent coexistence of multiple species. In both cases, the organisms must implement a computational task to metabolize the associated resource. The amount of resources an organism can get depends on the resource availability in the environment. The resource levels in the environment change due to the resource inflow into the environment, the resource decay-rate, and the rate at which the other organisms are making use of the resource. In Avida, the more resources an organism receives, the more CPU cycles it can use, much of which go toward its replication.

To fully understand the relationship between the environment the population is evolving in and its community complexity, we examined evolution in three more environments. *Environment III* is a control environment where there is only one simple resource (for the task NOT). *Environment IV* starts out identical to Environment III, but we inject a new resource to the environment every 10,000 updates. Thus, the information available to be exploited in the environment continually increases. At update 80,000, Environment IV stops changing with the same nine resources as found in Environment II.

While Environment IV allows us to examine the introduction of new environmental

complexity on a community, *Environment V* focuses on its loss. It is identical to Environment II for the first 100,000 updates with the same 9 resources, but at that point we remove eight resources and keep only the one resource associated with the task "NOT". The environment thus becomes same as environment III. For the detailed setups for environments I, II, and III, see appendix A.

We performed 50 independent runs for each environmental setup to collect statistically powerful results. For each run, we fix the genome length at 100 and turn off all insertion and deletion mutations leaving only substitution mutations. Since testing all possible point mutations at each site of a genome is already very time-consuming, we simplify our analyses not allowing mutations that cause position shift and therefore skipping sequence alignment and focusing on calculating the correlation between each pair of organisms. The communities under all five environments include a single, identical organism at the beginning of evolution, which contains no information about specific environment.

4.5 Results and Discussion

We compare our measure of community complexity with traditional diversity measurements. We categorize a community into species based on the distinct phenotype of each organism. Here we define the phenotype of an individual as the set of tasks it has implemented (regardless of the specifics how quickly or how often the tasks are performed). If a species includes fewer than three organisms, the organisms belonging to that species are likely detrimental mutants and are therefore not considered in the calculation. The proportion of each species is the number of organisms belonging to that species divided by the total number of organisms.

To minimize our computational cost, we sample organisms from the original community to compose a smaller, representative community and measure the information

in it. We randomly sample one organism from each species to make sure the information content of the smaller community truly reflects the whole. To keep the sample size consistent, we pad the smaller community up to 50 organisms with randomly sampled organisms from the original community. This whole process is done without replacement so no individual is ever sampled more than once. The number of species in a community of our experiments rarely exceeds 50. There are 26 possible instructions at each site in an Avida genome instead of 4 nucleotides in DNA sequences, we use log base 26 to calculate information content in the community. As in previous chapters we describe 1 instruction of complexity as 1 Complexity Unit (CU).

4.5.1 Complexity of communities in a single niche: Environment I

Environment I can sustain only one abundant species since the organisms that obtain the most energy from any combination of resources will be the most successful; no species will never have its growth limited due to depletion of resources. All of the other species are either derived from the dominant species or temporary survivors from history on their way to extinction.

Since community complexity is defined as the sum of the *distinct* information in the community, if two organisms inherit the same information from a common ancestor, this information should be counted only once in the calculation of community complexity. Thus the more overlapping information among organisms, the greater the difference will be between the total information among individual species and the unique information in the whole community. As a case study, we sample a typical run evolved in Environment I and compare its total individual information for 50 organisms with the community level complexity. The results are shown in Figure 4.2.

The total information summed over individual organisms reached around 3000 CU. However, the community information is only around 700 CU at the end of the

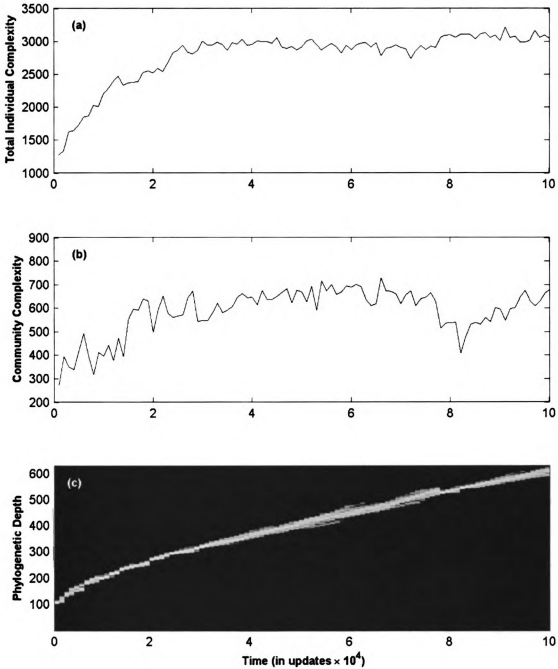

Figure 4.2: Measurements in a typical single-niche community over time. Metrics used are (a) total complexity of all 50 sampled individuals, (b) community complexity, and (c) phylogenetic depth of individuals.

evolution. The large difference is due to a lot of common information among the organisms in the population. We plot the phylogenetic depth of the population in Figure 4.2(c). Phylogenetic depth is the cumulative number of generations in which an organism's genotype differs from its parent. The result shows that the community at any time point is clustered together; there is no branching event to split the community into multiple persistent subpopulations.

We average the community complexity over 50 runs. The results (along with their 95% confidence intervals) are displayed in Figure 4.3(c). We also plot the average Shannon index and Simpson index as a comparison in Figures 4.3(a) and 4.3(b) respectively. Both diversity indices increase quickly at the beginning of the evolution and plateau around 10,000 updates. We cannot see what is changing in the community from them. However, our measurement shows the community continued to evolve information about its environment.

4.5.2 Complexity of communities with multiple niches: Environment II

Environmental heterogeneity is usually considered as the primary cause of genetic diversity. Even in a spatially homogeneous environment, Avida experiments with fixed-size populations show that the productivity of resources may influence the species richness. For example, it has been shown in Avida that the maximum species richness emerges at intermediate productivity [*Chow et al.*, 2004]. Our result is consistent with Chow et al.'s result, although we use a different method to classify species. We observed multiple species coexisting under Environment II. As a comparison to the above case from the single niche environment, we sampled a typical run from Environment II to analyze in detail.

We plotted the phylogenetic depth in Figure 4.4(c). Contrast to Figure 4.2(c), the range of phylogenetic depth from the run under the multi-niche system is notably

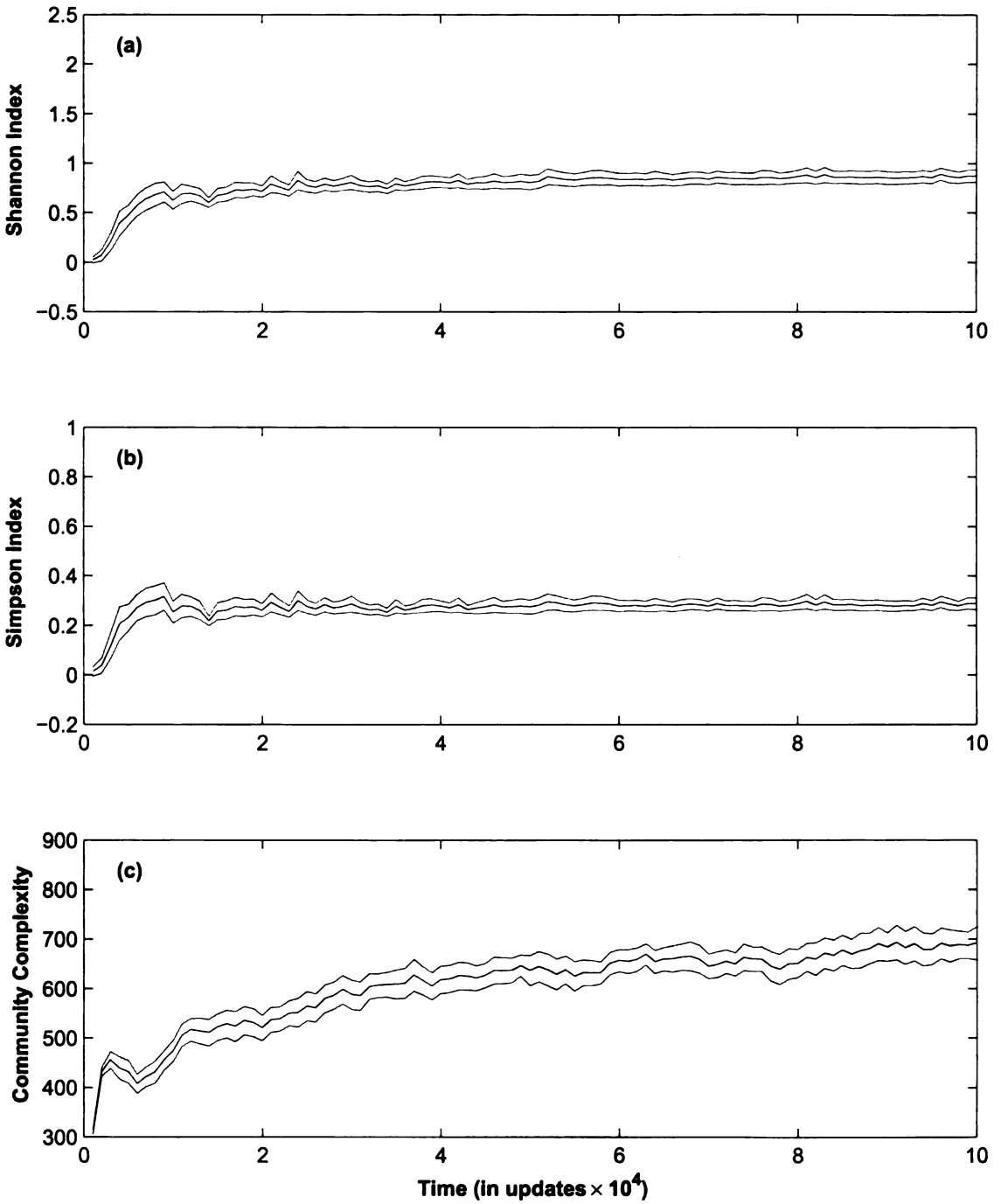

Figure 4.3: The results over 50 runs under Environment I. Upper and lower boundaries define 95% confidence interval. (a) average Shannon index, (b) average Simpson index, and (c) community complexity.

broader, which indicates that there are many species coexisting under the multi-niche environment. The dominant species in the single niche environment contained almost all the information about the environment. However, in a multi-niche environment different species are specialized on metabolizing different resources and thus the information is distributed throughout the community. The average information in each individual is comparatively smaller, as reflected in the total information in the sampled organisms, which is only around 2200 CU at the end of evolution. However, the community-level complexity is around 800 CU, which is higher than that in the run from the single-niche environment (around 700 CU). The main reason for this is that different species independently evolved similar information about the environment. Sometime a species other than specializes on metabolizing one resource, it compete same resource with other species. All independent information is counted in the community complexity. Another reason is that organisms from single niche environment contain more pleiotropic sites (where one gene influences more than one trait). Our current method counts the information contained in pleiotropic genes only once, which makes the information in the single niche environment a little lower.

We compared the average Shannon index and Simpson index with the average value of our complexity measure over 50 runs. Figure 4.5 demonstrates that both the Shannon index and the Simpson index are higher for the multi-niche environment than they were in the single-niche environment, which means a community is more diverse when evolved under the multi-niche environment. As with the result from the single-niche environment, the diversity indices do not change from the very early stage of evolution. Conversely, our complexity measurement steadily increases until the end of evolution which means that more information continually enters the community.

The new information in the community could come from two sources. One is when existing species evolve new information, and the other is when more species are formed in the community; each new species must carry some unique information

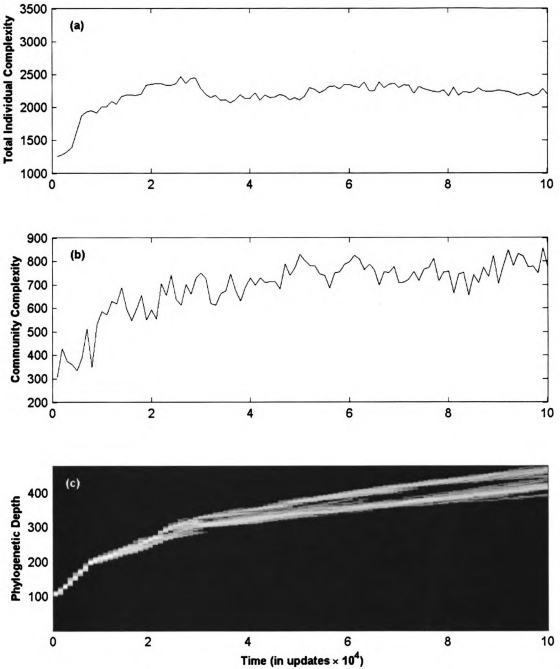

Figure 4.4: Measurements in a typical multi-niche community over time. Metrics used are (a) total complexity of all 50 sampled individuals, (b) community complexity, and (c) phylogenetic depth of individuals.

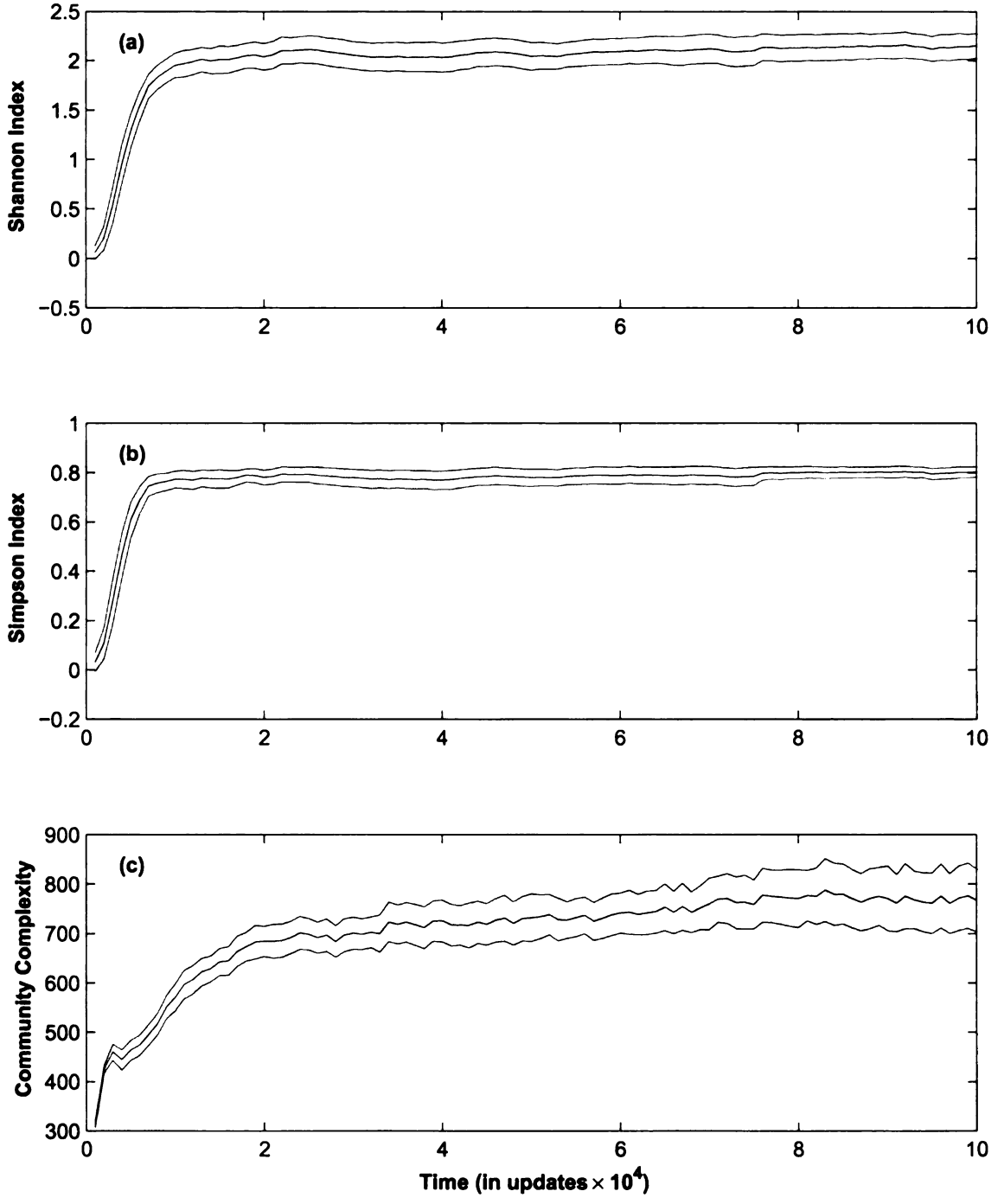

Figure 4.5: The results over 50 runs under Environment II. Upper and lower boundaries define 95% confidence interval. (a) average Shannon index, (b) average Simpson index, and (c) community complexity.

to stably coexist with the other species in the community. The experimental data is consistent with our reasoning as shown in Figure 4.6.

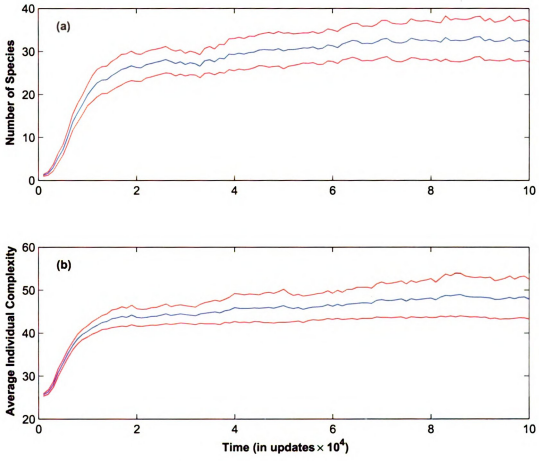

Figure 4.6: The results over 50 runs under Environment II. Upper and lower boundaries define 95% confidence interval. (a) average number of species and (b) average individual complexity.

4.5.3 Environmental impact on community complexity: Environments III, IV and V

To explicitly show the relationship between the environment and the community, we designed environments III and IV. Environment III contains only one resource for simplest computational task "NOT". Environment IV initially also contains only resource for task "NOT", but we add in a new resource every 10,000 updates until

all 9 basic resources are present at 80,000 updates.

The average result from Environment III (Figure 4.7) shows the Shannon index is around 0.5 nats and the unique information in the communities is around 620 CU at the end of evolution. Due to the simpler environment, these values are significantly lower than what we have seen previously.

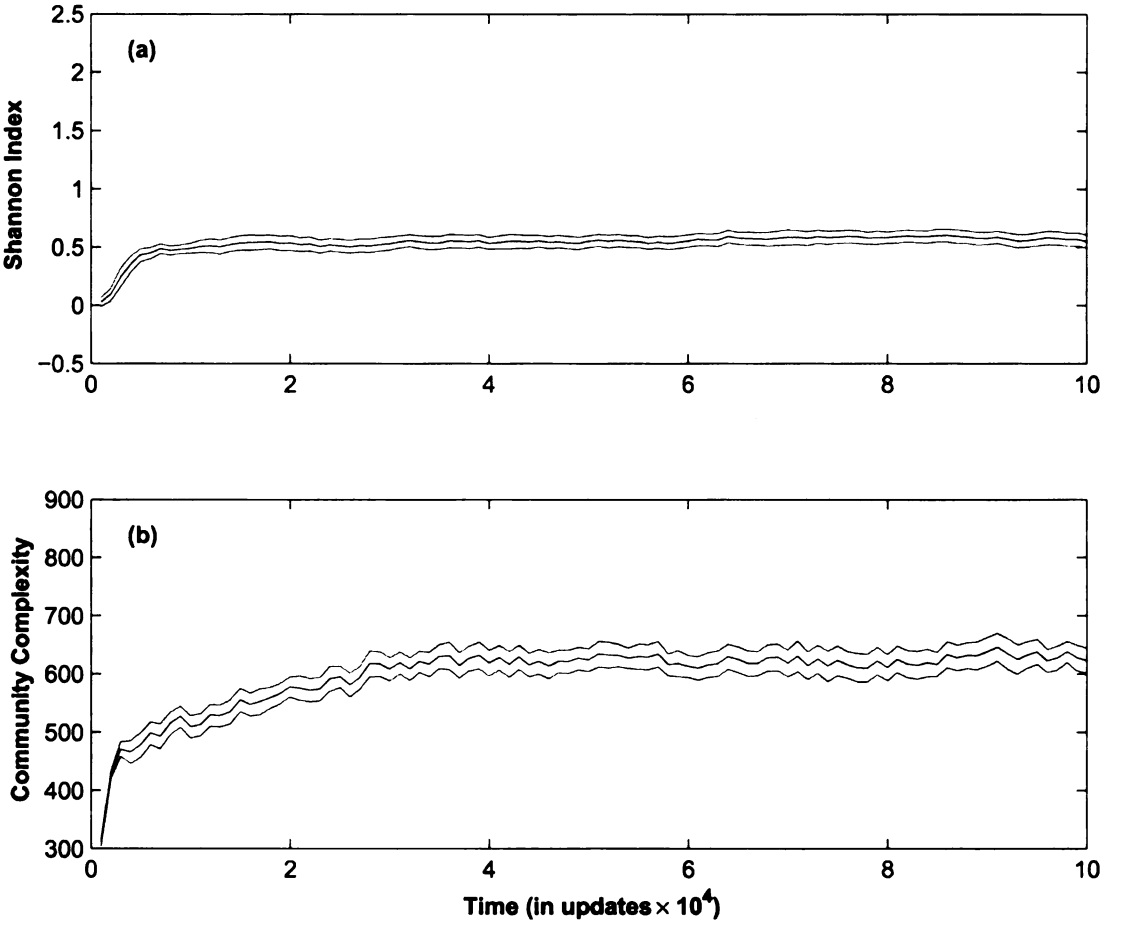

Figure 4.7: The results over 50 runs under Environment III. Upper and lower boundaries define 95% confidence interval. (a) average Shannon index and (b) community complexity.

From Environment IV (Figure 4.8) we observed that both the diversity index and the information content keep increasing as we add in new resources. At the end of the run, the Shannon index is around 2.2 nats and the information content is around 750 CU. As expected, all of these numbers are comparatively higher than those from

Environment III.

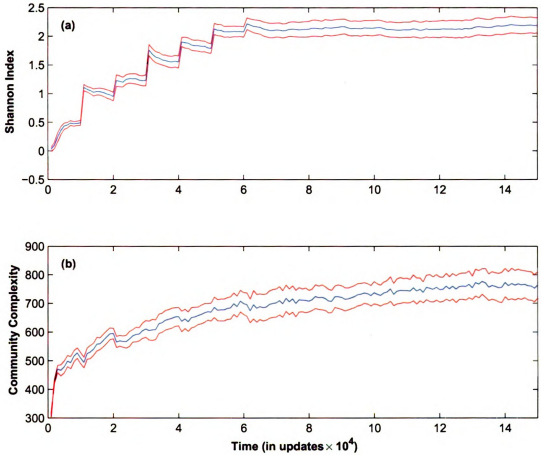

Figure 4.8: The results over 50 runs under Environment IV. Upper and lower boundaries define 95% confidence interval. (a) average Shannon index and (b) community complexity.

We compared the results from Environment IV with Environment II. The information in the community under Environment II reaches 700 CU of complexity around 20,000 updates while the information in the community under Environment IV do not reach the same complexity level until 60,000 updates. This shows that the information available in the environment limits the amount of information that can be contained in the community. We let the communities evolve another 70,000 updates after the resources in Environment IV are all present and we found that the information content in these communities converge to the same levels as Environment II

over time.

Environment V is identical to Environment II until update 100,000 when we remove all of the resources except for the one associated with the task "NOT" (making it identical to Environment III). We let the evolution continue for another 50,000 updates to let the communities adapt to their new environment. Figure 4.9 shows how the information content of the community drops after we remove most of the resources from the environment. After the environment changes at 100,000 updates, we observe that the information content in the communities under Environment V converges to the levels seen in Environment III. This result, together with the results

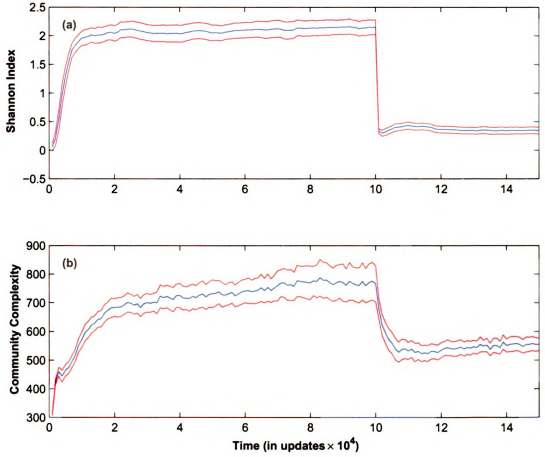

Figure 4.9: The results over 50 runs under Environment V. Upper and lower boundaries define 95% confidence interval. (a) average Shannon index and (b) community complexity.

from environments III and IV, show that the environment has a major impact on the evolution of community complexity.

4.6 Conclusions

We define community complexity as the sum of all distinct information in the community about the environment. Based on Shannon Information Theory, we developed a method to estimate new information found in each organism. There are two important factors which influence the measured result: the number of species and the average distinct information in each species.

We designed five environments in Avida and used our method to measure information in the community. We compare it with the Shannon index and the Simpson index for a single-niche and a multi-niche environment. The diversity indices increase quickly at the beginning of evolution and then stay almost constant over time. Compared to them, our measure of community complexity shows the information content continues to increase in most communities as long as there is environmental complexity left to be exploited. We compared the information content under different environments and found the community complexity in the single niche environment is less than it is in the multi-niche environment; the community complexity in a low information environment is less than it is in a high information environment. The community complexity changes as the available information changes in the environment. As expected, the environment plays an important role in how community complexity evolves.

In summary, our measurement of community complexity is more sophisticated and produces more intuitive results than previously used methods. We measure the degree of interaction between a community and its environment (including interactions between organisms in that environment). It is markedly different from traditional

diversity measurements and provides a valuable complement to them in determining community complexity.

Chapter 5

INFORMATION-BASED PHYLOGENY RECONSTRUCTION

In all of the research thus far, I have focused on the design of complexity measuring methods and have used them to analyze complexity changes in the course of evolution. Here I apply these concepts to an important biological problem. Measuring the information content of different organisms allows us to better understand the relationships among them. When new information enters a population, it is transmitted through the generations. If different genomes store distinct information at the same site, it can help us classify the organisms in that population into two groups. Inspired by this fact, we designed a character weighting technique to improve the accuracy of phylogeny reconstruction.

Character-weighting methods are often used to improve the accuracy of phylogeny reconstruction algorithms. Previous methods focus on removing highly variable sites (because they are likely to show false similarities between unrelated sequences) or sites that are too conserved (because they provide no information for reconstruction). However, little consideration has been given to the idea of using different weights based on the portion of the tree currently being reconstructed. Sites that provide information about the root of a tree may have little information about the leaves, and likewise sites that are informative about the specific relationships among neighboring

leaves, may appear random when examined across all available sequences.

We observe that sites where two distinct symbols are both highly represented are most likely to provide useful information for reconstructing deep bifurcations in the tree. We demonstrate that the neighbor joining algorithm is significantly more likely to correctly reconstruct deep bifurcations if more weight is given to these sites. We further show the robustness of this technique with sustained reconstruction improvements as we vary a number of characteristics about the trees, including their size, the number of distinct symbols used in the sequences, and the tree symmetry.

This chapter is organized as follows. First we introduce the background about phylogenetic reconstruction. Next, we describe our approach in more detail and introduce related works. Then we give our results and discussion. Finally we provide a conclusion.

5.1 Background

Designing algorithms to determine the evolutionary relationships between different species is an important research topic for computational biologists. Some researchers need to know these relationships to trace the transmission of viruses [*Dumpis et al.*, 2003] and identify emerging diseases; others use it to assist in drug design by predicting the conserved residues on protein surfaces [*Sen et al.*, 2004]. This is even an important question in agriculture, where accurate phylogenies can be used to aid in increasing the production of crops [*Soltis and Soltis*, 2003]. In general, knowing the correct evolutionary relationships is useful for classifying organisms into meaningful groupings, commonly called clades [*Aguinaldo et al.*, 1997; *Woese et al.*, 1990].

One of the most challenging aspects of phylogeny reconstruction is correctly identifying the deepest bifurcations within the phylogeny. For example, during the Cambrian Explosion, which occurred sometime between 570 and 530 million years ago,

all of the known phyla of animals appeared. However, existing techniques are unable to use the fossil record to determine almost anything about the process by which this occurred. Similarly, we currently are unable to resolve the evolutionary relationships between the phyla of Bacteria, and thus these phyla, which number well over 25, are treated as independent groups. The goal of this chapter is to build a better telescope into the past that will allow us to understand more about the fundamental innovations that make up the root systems of the phylogenetic trees that we study.

The need for accurate phylogenies has driven the creation of a wide variety of algorithms that will take in genetic data and attempt to reconstruct the original tree. We assume the genetic data is aligned nucleotide sequence data, though our techniques are generalized to other inputs as well. We refer to each position in the aligned sequence as a site or character, and we refer to the actual nucleotide value at a site as a nucleotide or character state.

What makes phylogeny reconstruction possible are "synapomorphies": inherited character states that are preserved in a set of species with a common ancestor that are not shared by more distant ancestors. Over time, synapomorphic signals are weakened as sites continue to mutate; meaning some descendants of a common ancestor will lose the shared character state. What makes phylogeny reconstruction difficult are "homoplasies": a character state shared by a set of species that is not the result of inheritance from their common ancestor. Homoplasies can simply be coincidental, or potentially the result of convergent evolution. Homoplasies are problematic because they are false synapomorphies and therefore provide false evidence for grouping together a collection of species. If there were no homoplasies, then we have what is referred to as the perfect phylogeny problem, which can be solved correctly in polynomial time [*Gusfield*, 1991].

In a typical phylogeny reconstruction problem, different sites or characters will incur different numbers of mutations resulting in synapomorphies, lost synapomorphies,

and homoplasies. Specific sites will provide useful information for reconstructing specific bifurcations but may provide no information or misleading information for other bifurcations. We propose a simple method for identifying characters that are likely to be useful for reconstructing deep bifurcations; specifically, characters with two well-represented character states. We also present a simple character weighting scheme to reinforce the information contained in these characters. We then validate our methods by showing that the Neighbor Joining algorithm is able to identify deep bifurcations more accurately when using our character weighting scheme instead of a uniform weighting scheme. While we have only applied our technique to Neighbor Joining, we believe it should enhance the recovery of deep bifurcations for all standard phylogeny reconstruction techniques.

5.2 Our Approach

The German entomologist Willi Hennig [1979] argued that only shared derived characters should be used as indicators of common descent. As noted earlier, a specific character may be a shared derived character for one set of organisms and thus provide evidence for bifurcation associated with them, but may not provide evidence for bifurcations in other parts of the tree.

For example, consider the symmetric tree structure shown in Figure 5.1. Suppose there is a mutation in the leftmost internal branch at character i and that no other mutations occur in character i. Then the organisms represented by leaf nodes T1, T2, T3, and T4 will share a common character state in character i, while the organisms represented by leaf nodes T5, T6, T7, and T8 will share a different common character state in character i. Thus, character i will be extremely informative for reconstructing this deepest internal bifurcation. On the other hand, character i will provide less information for reconstructing any of the other bifurcations in the phylogeny.

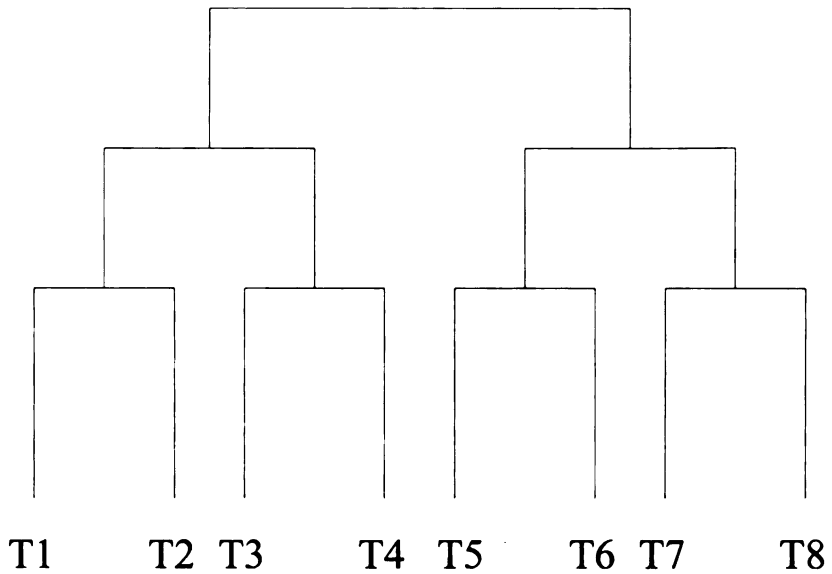

Figure 5.1: An example symmetric tree.

The problem with finding informative sites for deep bifurcations is that few synapomorphies that formed from early mutations will manage to persist until the present time. If a synapomorphy does persist, it is likely due to an adaptive mutation where any further mutations at this character would typically be detrimental to the organism. In many cases, some noise will be introduced as either some of the common descendants lose the shared character state through further mutations or other organisms gain the shared character state through further mutations creating homoplasies. However, even with noise, a combination of informative characters may still provide enough information to reconstruct a deep bifurcation. The problem is identifying sites where the information to noise signal is strong and reinforcing these sites so that they overcome the noise included in other sites. In the next two subsections, we present our methodology for identifying and reinforcing informative characters for reconstructing deep bifurcations.

5.2.1 Identifying Informative Characters for Deep Bifurcations

How can we decide which characters have useful information for reconstructing a specific bifurcation without knowing the true phylogeny? This turns out to be tractable for deep bifurcations, or perhaps more accurately "equal bifurcations". Suppose a bifurcation divides the species into two groups of size X and $N - X$. The best possible result is that the first group of X species will share one character state, while the second group of $N - X$ species will share a different character state. For deep or equal bifurcations, X could be $N/2$ while for shallow bifurcations, X will be much smaller.

Thus, the key for identifying useful characters for deep bifurcations is finding characters that have two well-represented character states. In more detail, after aligning sequences, we measure the abundance of each character state (either nucleotides or amino acids, depending on the type of data we are working with) at each site. We then sort the characters by the abundance of the *second* most abundant character state for that character. For example, if there are 64 species and at the first site 32 of them have nucleotide A, 30 of them have nucleotide G, 2 have nucleotide C and none have nucleotide T; this site would be given a rating of "30".

5.2.2 Reinforcing Informative Characters for Deep Bifurcations

Given our results from the previous discussion, we now have a way to rate characters with regard to their ability to inform reconstruction of deep or equal bifurcations; namely the cardinality of the second most abundant character state. We now need a method for weighting characters based on this ranking. To simplify our investigation, we restrict our attention to a two-parameter weighting scheme (t, w). The first

parameter t is a *threshold* where we increase the weight of all characters that have a rating at or above the threshold. The second parameter w is the weight given to the highly weighted sites.

Even for this two-parameter weighting scheme, the optimal choices for t and w are not clear. In this work, we consider all possible threshold values t but only consider $w = 2$; sites with ratings below the threshold are given weight 1. We will investigate more sophisticated weighting schemes in future work, for example, continuous weighting schemes where a higher rated site gets more weight than any lower rated site.

5.2.3 Methods

To determine if our approach of identifying and reinforcing sites with two abundant character states improves reconstruction of deep bifurcations, we designed a set of experiments to compare Neighbor Joining (NJ) [*Saitou and Nei*, 1987] where all sites are equally weighted with Neighbor Joining where sites are weighted using our $(t, 2)$ weighting scheme. Neighbor Joining is one of the most popular distance-based phylogeny reconstruction techniques because of its efficiency and relative effectiveness. As it is a distance-based approach, all sequence information is translated into a collection of distances between pairs of sequences. With a uniform weighting scheme, each site contributes equally to the distance between any pair of sequences. With our $(t, 2)$ weighting scheme, highly rated sites contribute twice as much as other sites to the distance between any pair of sequences. While we only test our method with a distance-based reconstruction method, we believe the results would also hold for character-based methods as well.

We test our method using a computer simulation approach similar to the seq-gen application [*Rambaut and Grassly*, 1997]. That is, we start with a random ancestor and a model tree topology. Based on this model tree, we use a Monte-Carlo simulation

procedure of molecular sequence evolution to generate the sequence for each internal node in the tree and for all of the leaves. In this simulation we assume that all symbols are equally likely to be mutated to and thus no additional information is available from some character substitutions being more likely than others. We also do not allow insertions or deletions so that sequence alignment is not an issue. One minor difference between our simulation and seq-gen is that our edge lengths represent the actual number of mutations incurred between the parent and child while the edge lengths in most applications such as seq-gen represent a probability that each site suffers a mutation. For each tree topology, we randomly generated 1000 groups of sequence data. We then apply the Neighbor Joining method to the resulting sequence data using a uniform character weighting scheme as well as our $(t, 2)$ character weighting scheme for all values of t. For each topology, we compute the percentage of cases that Neighbor Joining is able to successfully reconstruct the deepest bifurcation with both weighting schemes. Since we have 1000 replicates for each topology, we are able to produce statistically significant results.

Our default parameter settings are the following. The sequence length is 200, the character set is 4 (to represent nucleotides), the number of leaves or final sequences is 32, and the default tree topology is symmetric so the deepest bifurcation is an equal bifurcation. To ensure the information available to reconstruct this deepest bifurcation is limited, our default topology has short internal branch lengths of 5 while branches near the leaves have 80 mutations. A graphical depiction of this tree topology is given in Figure 5.2.

We study several variants of our default settings in order to understand how different factors affect our methodology. One of the factors we vary is the tree size; we also study trees with 16 leaves and 64 leaves. When we restrict to 16 leaves, we eliminate the first internal branch from Figure 5.2. When we expand to 64 leaves, we merge two of the 32 leaf trees by adding new branches with 5 mutations from the

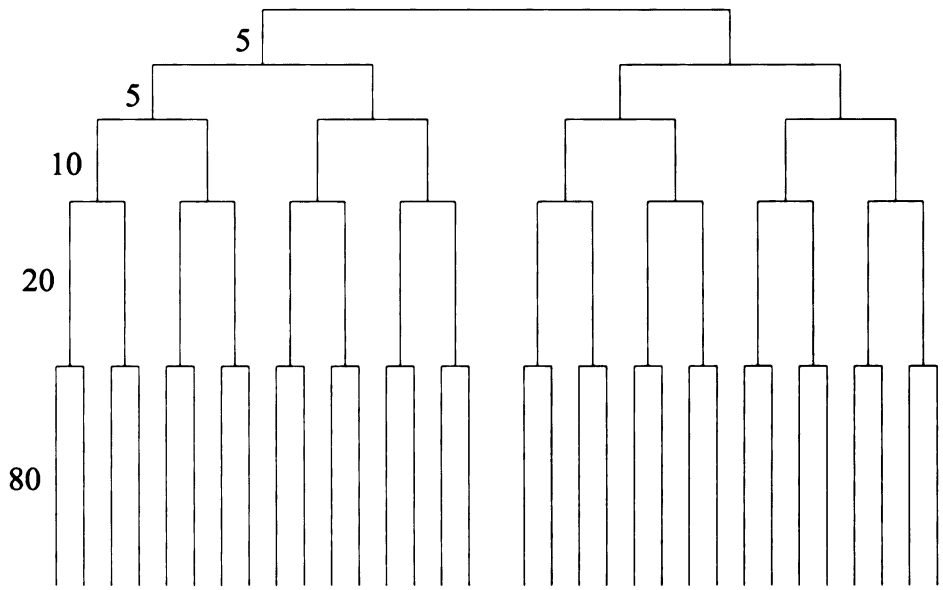

Figure 5.2: Our default 32-leaf symmetric tree topology.

new root to each of the old roots.

A second factor we vary is the number of symbols in the character set as we also consider a size 20 character set to mimic phylogeny reconstruction from amino acid sequences. When we use this expanded character set, we examine trees both identical to the default tree as well as ones where the terminal branches have 140 mutations instead of 80. This extension is necessary because the larger alphabet size reduces the probability of homoplasy. More mutations must occur to increase the difficulty of the reconstruction [*Semple and Steel*, 2002]. If a Neighbor Joining algorithm with uniform character weighting reconstructs a tree perfectly, it is impossible for us to show any improvement.

A third factor that we vary is the symmetry of the tree. Our default tree is symmetric, which means the deepest branch is an equal bifurcation. We also study asymmetric trees to determine if our method works when there is no bifurcation that partitions the sequences into two equal-sized groups. Assuming that the correct topology is a binary tree, we note that any binary tree will contain a partition (though not necessarily the deepest partition) that divides the sequences into two groups

where the smaller group is at least one third of all the sequences. We test our method using an asymmetric tree topology of 32 sequences where the smaller partition of the deepest bifurcation has 12 (roughly 1/3) of all the sequences. This asymmetric tree is graphically depicted in Figure 5.3.

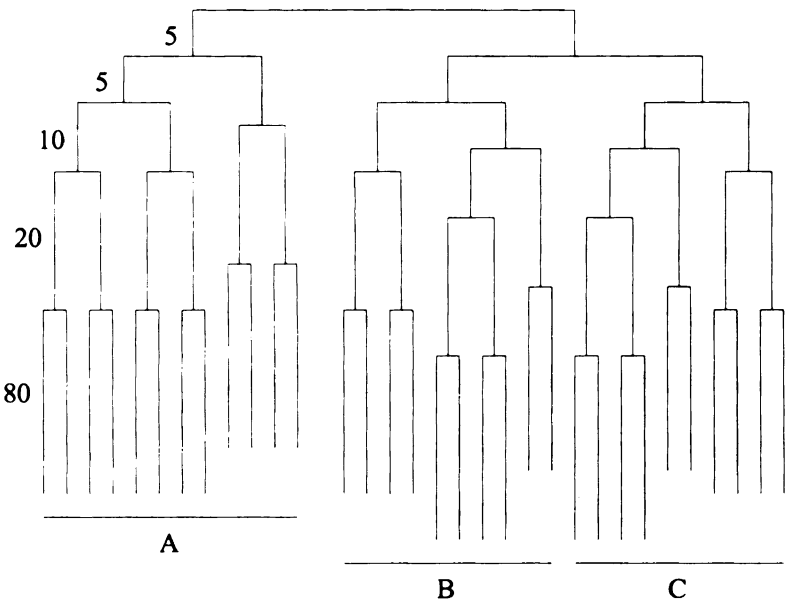

Figure 5.3: Our asymmetric tree topology with 32 leaves.

For each of the experiments described above, we measure the number of sites that have each rating and produce a bar graph to show their distribution. We then overlay each bar graph with a plot depicting the accuracy with which the most internal bifurcation can be reconstructed assuming we set the rating threshold t to the value on the X-axis. This accuracy indicates the frequency with which this bifurcation was reconstructed perfectly (out of the 1000 sets of sequences tested per point). We also include a 95% confidence interval around each accuracy plot.

We note that the performance of Neighbor Joining using a uniform weighting scheme can be observed for thresholds t where either no sites receive weight 2 or all sites receive weight 2. This second option is guaranteed to occur when $t = 0$, so we do not provide a separate plot of the performance of Neighbor Joining with a uniform weighting scheme in our figures.

5.3 Related Work

Many *character weighting* and *character-state weighting* techniques have been used in phylogeny reconstruction. For example, an extreme form of character weighting is to eliminate characters associated with introns and hyper-variable regions. A close analogy is the method of successive weighting [*Farris*, 1969; *Carpenter*, 1994], which refines an initial tree estimate by reducing the weight of characters that appear to be homoplastic on the tree so that the signal is made to stand out more strongly. For instance, mutations occur more frequently on the third position of a codon than on either of the first two, so nucleotides in the third codon position receive a lower weight. Researchers assume that the sites with higher mutation rates provide a high level of homoplasy, which reduces reconstruction accuracy. The problem with this assumption is that, while the highly variable sites could confuse the reconstruction of deeper relationships, they will contain the most useful information for shallower branch points. Our approach of weighting characters to target a specific bifurcation seems unique.

The concept behind the character-state weighting technique is to assign weights based on state transformations within a site. Nucleotide transitions are frequently down-weighted relative to transversions in phylogenetic analysis. This is based on the assumption that transitions, by virtue of their greater evolutionary rate, exhibit relatively more homoplasy and are therefore less reliable as phylogenetic characters. Six parameter parsimony is another method of character-state weighting, where a cost is assigned to the transformation from any character state to any other [*Williams and Fitch*, 1990] based on observed frequencies in the data set, as reconstructed on an initial tree (obtained from unweighed parsimony). The object of this weighting is that frequent changes are considered more likely than rare changes to have experienced homoplasy.

Milosavljević and Jurka and others [1989; 1993] generalized parsimony and com-

patibility approaches by a minimal length encoding principle to develop phylogeny reconstruction approaches that can be used to identify bifurcations. Specifically, if introducing a bifurcation reduces the number of bits needed to encode the phylogeny, then this provides support for the existence of that bifurcation. They applied this approach to identify new subfamilies of Alu sequences [*Milosavljević and Jurka*, 1993]. Our technique does not explicitly compare different phylogenies but simply provides a rating to characters based on the distribution of their character states.

5.4 Results and Discussion

5.4.1 Results with Default Settings

As noted in our methods section, our default settings are 32 leaves, 4 character states, 200 characters, and a symmetric tree topology. We did 1000 replicates for this setting. The distribution of sites with rating t is plotted in the gray bar graph in Figure 5.4. The graph appears to have a normal distribution with a slight right-skew. Since we are working with 32 sequences, the theoretical range for this rating is from 0 (if the site is identical across all of the sequences) to 16 (if the top two abundances are both 16) with 8 being the expected value if all four possible symbols are found with identical frequency. In the distribution here, there tend to be few sites that have a rating greater than 10 or less than 3.

For our method, we double the weight of all sites that have a rating above a certain threshold, but we still need to specify what that threshold should be. Ideally we want to choose a threshold that will maximally improve our ability to correctly reconstruct the tree. For our default tree (with 32 leaves generated using length 200 sequences from an alphabet size of 4) the gray bar graph in Figure 5.4 displays the number of sites for each rating (i.e., this bar graph shows the number of sites that have a given abundance of the second most frequent symbol).

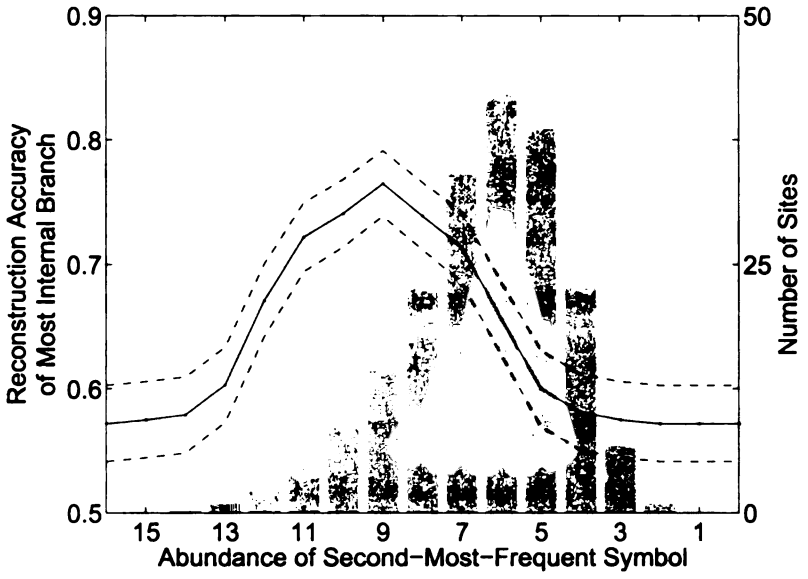

Figure 5.4: Measurements on the default tree. The bar graph indicates the number of sequence positions out of 200 that are assigned each rating (scale on right side of figure); the solid line displays the average reconstruction accuracy of the most internal bifurcation (scale on left side of figure); the dashed lines define the 95% confidence interval around our accuracy tests.

The resulting reconstruction accuracy of the most internal bifurcation when we utilize our $(t, 2)$ weighting scheme is displayed as the black line in Figure 5.4. There are no positions that have exactly 16 of one symbol and 16 of another symbol so the start point (57.2%) actually represents the reconstruction accuracy of most internal bifurcation with a uniform weighting. The accuracy steadily improves as we lower our threshold t, hitting a maximum of 76.5% before falling to the original accuracy when weights of all sites are doubled. The largest improvement occurs when we set our threshold to 9. The fact that the accuracy of our scheme with $(t, 2)$ weighting is *always* greater than the accuracy with uniform weighting even when t is small suggests that our rating of sites with respect to their information content for the deepest bifurcation is accurate. In particular, the weight given to the highest rated sites overcomes any penalty given to the low rated sites.

5.4.2 The Effect of Tree Size

To test the effect of tree size on our method, we compared our experimental results from the 16 leaf tree to the 32 leaf tree to the 64 leaf tree. The results for the 16 leaf tree are displayed in Figure 5.5, and the results for the 64 leaf tree are depicted in Figure 5.6. In the 16 leaf tree, the uniform weighted accuracy is 62.8% and the maximum accuracy occurs when $t = 5$ at 71.5% for a maximum improvement of 8.7%. In the 64 leaf tree, the uniform weighted accuracy is 59.0% and the maximum accuracy occurs when $t = 17$ at 79.3% for a maximum improvement of 20.3%.

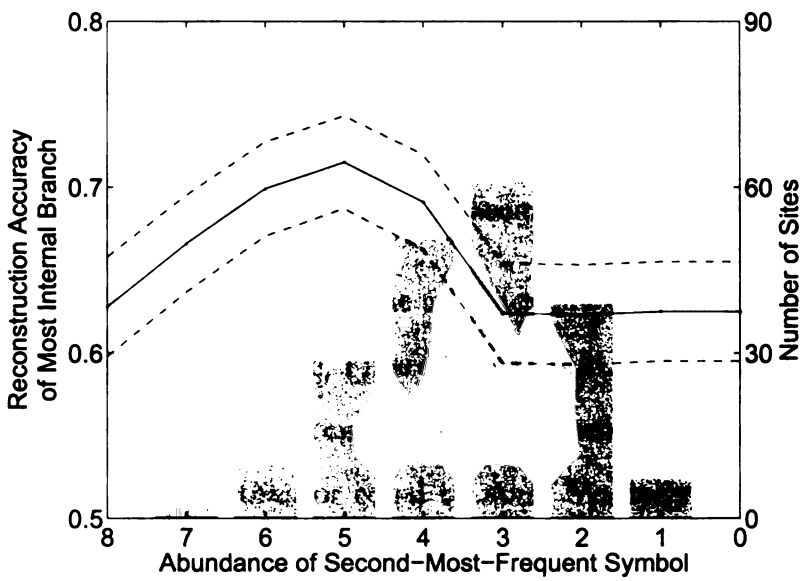

Figure 5.5: Measurements on the tree with 16 leaves. The bar graph indicates the number of sequence positions that are assigned each rating; the solid line displays the average reconstruction accuracy of the most internal bifurcation; the dashed lines define the 95% confidence interval around our accuracy tests.

As can be seen from these figures, the same general pattern holds for all tree sizes. The improvement, though, increases as we increase tree size. For example, the maximum improvement for the 16 leaf tree is only 8.7% as compared to 19.3% for the 32 leaf tree and 20.3% for the 64 leaf tree.

We also can infer some information on the optimal threshold t to use. The optimal maximum accuracy in all three figures always occurs at a threshold t that is larger

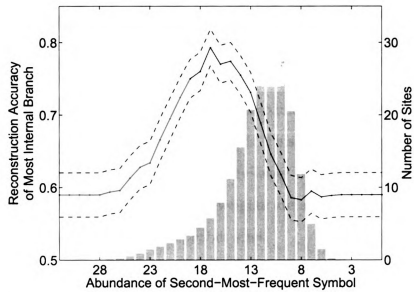

Figure 5.6: Measurements on the tree with 64 leaves. The bar graph indicates the number of sequence positions that are assigned each rating; the solid line displays the average reconstruction accuracy of the most internal bifurcation; the dashed lines define the 95% confidence interval around our accuracy tests.

than the most common rating; that is, the accuracy always peaks before the highest point of the bar graph. For example, Figure 5.4 shows that the most common rating for sites is *six*, while the greatest accuracy is reached when we double weights of all positions with rating greater than or equal to *nine*. Similarly in Figure 5.6, *eleven* is the most common rating, while the maximum accuracy occurs at a threshold of *seventeen*. In fact, in all three cases, the maximum reconstruction accuracy occurs when the threshold $t = n/4 + 1$ where n is the number of leaves in the tree, though we do not have a good explanation for this phenomenon.

5.4.3 The Effect of Alphabet Size

The nature of tree reconstruction will change when we use a different alphabet. When we have a large alphabet, the tree will typically have fewer homoplasies, since the probability of randomly mutating into the same state is reduced. To compensate for this effect, we extend the external branch lengths on our tree with 32 leaves from

80 to 140 when using an alphabet size of 20, the number of distinct amino acids in protein sequences. The bar graph and the reconstruction accuracy of the most internal bifurcation after weighting are shown in Figure 5.7. For this alphabet size, the uniform weighted accuracy is 53.0% and the maximum accuracy occurs when $t = 5$ at 79.2% for a maximum improvement of 26.2%. This improvement is quite striking because it starts from a reconstruction accuracy lower than that in our default setup, but rises to an even higher value after the weights are applied. Thus, this technique seems even more effective on larger alphabet sizes.

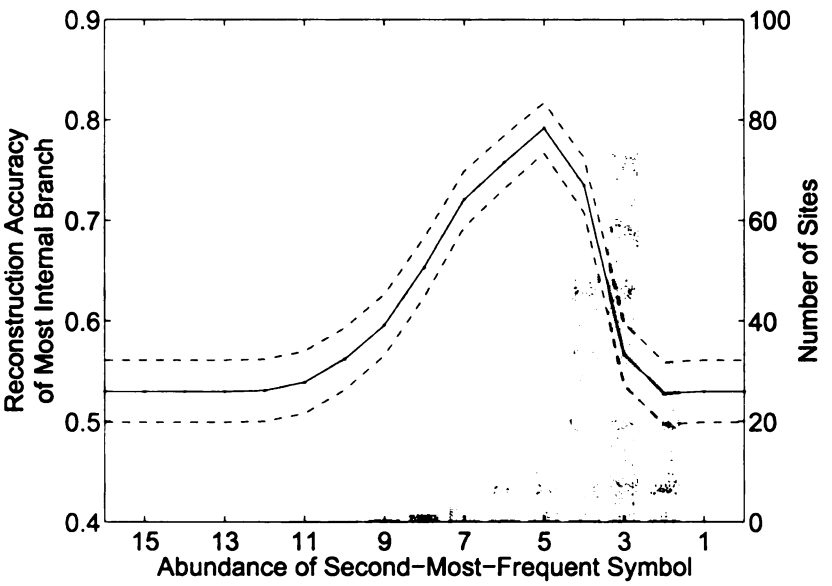

Figure 5.7: Measurements on the tree constructed using sequences with a larger alphabet size of 20. The bar graph indicates the number of sequence positions that are assigned each rating; the solid line displays the average reconstruction accuracy of the most internal bifurcation; the dashed lines define the 95% confidence interval around our accuracy tests.

Comparing the results from the two groups of experiments, we see that the bar graph in the experiments with a 4-character alphabet (Figure 5.4) has a peak at a higher rating than the bar graph in the experiments with an alphabet size of 20. This effect is due to the fact that a larger alphabet increases the chance that a site will be homoplasy-free. As in previous experiments, the accuracy always peaks at a higher threshold than the most common rating, so it is not strange to see the reconstruction

accuracy plot for an alphabet size of four to hit its peak earlier than the one with an alphabet size of 20.

5.4.4 The Effect of Asymmetry

In all of our experiments so far, we used a symmetric tree, which meant that the deepest bifurcation would always be an equal bifurcation. We now consider the effect when no symmetric bifurcation exists in the tree. As noted earlier, there will always exist a bifurcation (though not necessarily the deepest) that partitions the tree so that the smaller group will contain at least 1/3 of the leaves. See appendix for a proof of this assertion.

To study the reconstruction potential of a tree with this worst-case split, we evaluate our method on a 32 leaf tree where the deepest bifurcation does indeed partition the leaves into a set of 12 and a set of 20. This tree is depicted in Figure 5.3. The internal branch lengths are proportional to the lengths shown in the graph, with shortest ones equal to 5, the medium ones equal to 10 and longest ones equal to 20. All of the external branch lengths are equal to 80. For alphabet size 4, reconstructed accuracy of the deepest bifurcation is 57.3% with a uniform weighting and peaks with $t = 8$ at 73.5%, as shown in Figure 5.8.

Our explanation for why our method still functions correctly is as follows: if all three subpopulations maintain the same character state as in the ancestral state, this character is not informative at all. If there is a mutation along the most left internal branch and it persists all the way down to leaf nodes of cluster A, but there is no such mutation at this position in clusters B or C then a doubling of weight at this position will elongate the distance between this subpopulation A and all other leaves. In the ideal case, we would expect such a position to have a rating of 12 (with the most abundant symbol having a frequency of 20). Even adding some noise to this process, a rating of nine (which is the optimal rating threshold) is very likely for these

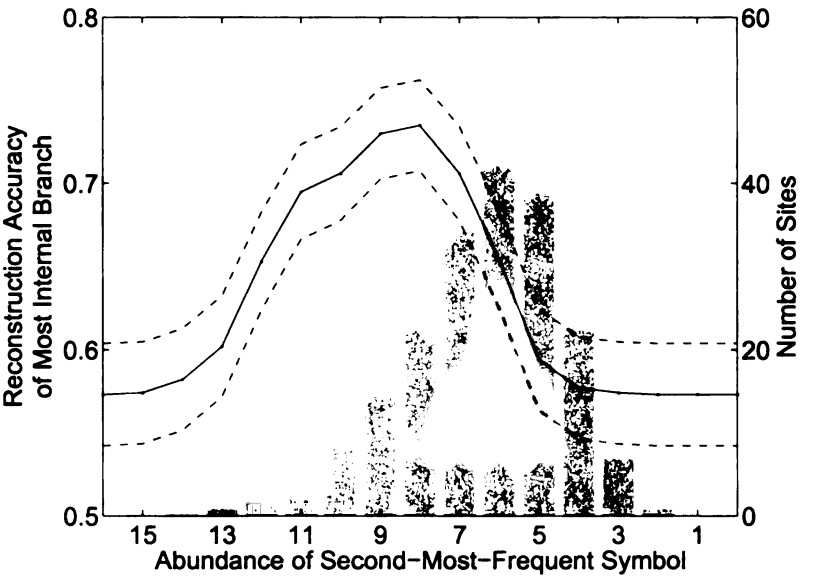

Figure 5.8: Measurements on the asymmetric tree. The bar graph indicates the number of sequence positions that are assigned each rating; the solid line displays the average reconstruction accuracy of the most internal bifurcation; the dashed lines define the 95% confidence interval around our accuracy tests.

informative sites.

A parallel argument can be made for the branch that separates B from A and C, as well as the one that separates C from A and B. Although reinforcing the reconstruction of these branches does not directly elongate our test-branch, the doubling of some sites increases the length of the branch to subtree B while the doubling of others increase the length of the branch to subtree C. The more accurate classification of subtrees B and C will also decrease the likelihood that one of their leaves is directly grouped into subtree A.

5.5 Conclusions

Our experimental results demonstrate that increasing the weight of specific sites consistently improves the reconstruction accuracy of deep bifurcations under a variety of conditions. Furthermore, it proves to be more effective as tree size or alphabet size increases. This technique can be used as the basis of phylogeny reconstruction

algorithms that apply different weights to sequence positions depending on the portion of the tree currently being reconstructed. In particular, we plan for adapting our technique of identifying deep bifurcations into a recursive top-down phylogeny reconstruction algorithm.

Chapter 6

FUTURE WORK

In this thesis, I have focused on designing methods to analyze complexity in living system, using digital organisms as my test case. The next step of my research will involve the refinement of these methods. For organism complexity, I will examine multiple sites at once in an attempt to decipher epistatic interactions and identify redundant information or correlated sites in a genome. Genetically, the adjacent sites have the highest probability to be correlated with each other and generate linkage disequilibrium; not all combinations of symbols have equal probability. Calculating the information in adjacent sites should significantly improve the accuracy of this complexity measurement. With current computer speeds, we can calculate the distribution of the combinations of instructions for three sites, which would require testing 17,576 mutants to calculate the information in a single line of a genome. Given a genome of length 100, this procedure would take about one hour.

In addition to devising methods of more accurately calculating the information content of a genome, we are also interested in designing good approximations that can be computed quickly. One such way is to make a binary decision on the information content of each site. We knock out the site (replacing it with an inert NULL instruction) and if the organism's fitness is reduced, we count this line as 1 CU. If this knockout had no effect on the organism we assume that no information was being stored at this position. Such knockout tests provide a dramatic speed increase since they only require us to test a single mutant per position in the genome. Initial tests have indicated that they provide only a minor loss of accuracy.

Knockout tests also have an additional advantage. Sometimes an instruction is not functional for a genome, but when we generate all possible mutations, other instruction at that position may influence how the genome functions. This may create an illusion of information at that site, resulting in a positive skew to our complexity calculation. Knockout tests, however, do not suffer from this. Additionally, multiple knockouts are not expensive to perform and can be used to detect redundancy. For example, if we find two sites that can be knocked out independently without harmful effect to the organism, but in combination they result in the loss of a trait, then clearly these sites were involved in the redundant encoding of that trait.

To improve our measurements of community complexity, I will preprocess each organism to determine which sites contain information. When determining if organism A contains information about B, I will only need to test those sites that are individually informative in both to determine if this is the same information. Overall, this should improve both the speed of the algorithm and its accuracy—we'll no longer have false correlations adding to our community complexity where no real information exists. This speed increase will also allow us to compare many organisms at once. Each time a new organism's unique information is calculated, we can compare it to more than just one previously tested organism.

Next, I plan to use the community complexity measure to better understand the evolution of complexity in natural systems. I hope to collaborate with the people from the Department of Microbiology and Molecular Genetics at Michigan State University and measure the complexity of the bacterial communities from different environments. The goal of this project is to evaluate the environmental impact on the information content of the genomes.

For the character-weighting technique in Chapter 5, several directions remain to be explored before the full power of our methodology can be realized. First, I will examine how well this methodology applies to other reconstruction techniques besides

Neighbor Joining. I see no reason why it should not, but this needs to be verified experimentally. Second, given our $(t, 2)$ weighting scheme, I would like a reliable and automated method for determining the optimal threshold t. Alternatively, I also plan to consider continuous schemes where a higher rated site is given strictly higher weight than any lower rated site.

I am planning to adapt our technique to turn any phylogeny reconstruction technique such as Neighbor Joining into a top-down recursive phylogeny reconstruction algorithm. At the beginning of each recursive call, I use our technique to weight each site of the sequences being reconstructed. Given these weights, I then use a phylogeny reconstruction algorithm such as Neighbor Joining to identify a deep bifurcation. Given this bifurcation, I partition the sequences into two distinct subpopulations and recursively apply our technique on each set of sequences. The unique aspect of this approach is that it focuses on reconstructing the tree one bifurcation at a time starting with deep bifurcations. Since the informative sites for each bifurcation are different, the weights generated by our algorithm will change with each recursive call. The main challenge for applying this technique is that when the subtrees are rebuilt, there is not an easy way to identify the attachment points for the branch that connects them back together. I plan to address this issue by exploring methods used in rooting unrooted trees. For example, I will consider using samples from the other subtree as an outgroup.

Appendix A

ENVIRONMENT SETUP

A.1 Environment I

```
REACTION NOT   not  process:value=1.0:type=pow
                    requisite:max_count=1
REACTION NAND  nand process:value=1.0:type=pow
                    requisite:max_count=1
REACTION AND   and  process:value=2.0:type=pow
                    requisite:max_count=1
REACTION ORN   orn  process:value=2.0:type=pow
                    requisite:max_count=1
REACTION OR    or   process:value=3.0:type=pow
                    requisite:max_count=1
REACTION ANDN  andn process:value=3.0:type=pow
                    requisite:max_count=1
REACTION NOR   nor  process:value=4.0:type=pow
                    requisite:max_count=1
REACTION XOR   xor  process:value=4.0:type=pow
                    requisite:max_count=1
REACTION EQU   equ  process:value=5.0:type=pow
                    requisite:max_count=1
```

A.2 Environment II

```
RESOURCE resNOT:inflow=100:outflow=0.01
RESOURCE resNAND:inflow=100:outflow=0.01
RESOURCE resAND:inflow=100:outflow=0.01
RESOURCE resORN:inflow=100:outflow=0.01
RESOURCE resOR:inflow=100:outflow=0.01
RESOURCE resANDN:inflow=100:outflow=0.01
RESOURCE resNOR:inflow=100:outflow=0.01
RESOURCE resXOR:inflow=100:outflow=0.01
RESOURCE resEQU:inflow=100:outflow=0.01

REACTION NOT  not  process:resource=resNOT:
                   value=1.0:frac=0.0025:max=25
REACTION NAND nand process:resource=resNAND:
                   value=1.0:frac=0.0025:max=25
REACTION AND  and  process:resource=resAND:
                   value=2.0:frac=0.0025:max=25
REACTION ORN  orn  process:resource=resORN:
                   value=2.0:frac=0.0025:max=25
REACTION OR   or   process:resource=resOR:
                   value=4.0:frac=0.0025:max=25
REACTION ANDN andn process:resource=resANDN:
                   value=4.0:frac=0.0025:max=25
REACTION NOR  nor  process:resource=resNOR:
                   value=8.0:frac=0.0025:max=25
REACTION XOR  xor  process:resource=resXOR:
                   value=8.0:frac=0.0025:max=25
```

REACTION EQU equ process:resource=resEQU:
value=16.0:frac=0.0025:max=25

A.3 Environment III

RESOURCE resNOT:inflow=100:outflow=0.01

REACTION NOT not process:resource=resNOT:
value=1.0:frac=0.0025:max=25

Appendix B

PROOF OF A NEARLY-BALANCED BIFURCATION

We assert that in any binary tree there exists a bifurcation that will split the tree into two subtrees where the smaller of the subtrees has at least 1/3 of the leaves of the full tree. We prove this assertion by contradiction.

Consider an arbitrary binary tree T with N leaves. Every bifurcation B partitions the tree into two subtrees T_1 and T_2 with N_1 and N_2 leaves respectively, where, without loss of generality, $N_1 \leq N_2$.

Assume our assertion is false. This means that for any bifurcation B in T, $N_1 < 1/3N$. Consider the bifurcation B_{max} that has the largest number of leaves in the smaller subtree, breaking ties arbitrarily. The root of the subtree T_2 has two other bifurcations connected to it that divide subtree T_2 into two smaller subtrees T_3 and T_4 with N_3 and N_4 leaves respectively, where $1 \leq N_3 \leq N_4$. Let B' be the bifurcation that separates subtree T_4 from the rest of the tree. It must be the case that $N_4 > 2/3N$ or else B' would be such a bifurcation. This contradicts our assumption that B_{max} is the bifurcation with the largest leave size for smaller subtree since the bifurcation B' has leave size $N_1 + N_3$ for smaller subtree, which is strictly larger than the size of B_{max} which is N_1. Thus our assertion is true.

BIBLIOGRAPHY

Adami, C., and N. Cerf, Physical complexity of symbolic sequences, *Physica D*, *137*, 62–69, 2000.

Adami, C., C. Ofria, and T. C. Collier, Evolution of biological complexity, *Proc. Nat. Acad. Sci*, *97*, 4463–4468, 2000.

Aguinaldo, A. M. A., J. M. Turbeville, L. S. Linford, M. C. Rivera, J. R. Garey, R. A. Raff, and J. A. Lake, Evidence for a clade of nematodes, arthropods and other moulting animals, *Nature*, *387*, 489–493, 1997.

Anand, M., and L. Orlóci, Complexity in plant communities: the notion and quantification, *J. Theor. Biol.*, *179*, 179–186, 1996.

Bennett, C. H., Logical depth and physical complexity, in *The Universal Turing Machine, A Half-Century Survey*, edited by R. Herken, pp. 227–257, Oxford University Press, Oxford, 1988.

Carpenter, J. M., Successive weighting, reliability and evidence, *Cladistics*, *10*, 215–220, 1994.

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

Cooper, T. F., and C. Ofria, Evolution of stable ecosystems in populations of digital organisms, in *Eighth International Conference on Artificial Life*, edited by M. B. RK Standish and H. Abbass, vol. 119, pp. 227–232, MIT Press, Boston, MA, 2002.

Cover, T. M., and J. A. Thomas, *Elements of Information Theory*, Sinauer Associates, 1991.

Darwin, C., *On the Origin of Species by Means of Natural Selection*, Murray, London, 1859.

Dumpis, U., et al., An outbreak of hbv and hcv infection in a paediatric oncology ward: Epidemiological investigations and prevention of further spread, *J. Med. Virol.*, *69*, 331–338, 2003.

Farris, J. S., A successive approximations approach to character weighting, *Syst. Zool.*, *18*, 374–385, 1969.

Goertzel, B., *The Evolving Mind*, Routledge, 1993.

Goldsmith, T., Optimization, constraint, and history in the evolution of eyes, *Quart. Rev. Biol.*, *65*, 281–322, 1990.

Gould, S. J., *Wonderful Life: The Burgess Shale and the Nature of History*, Penguin, 1989.

Gusfield, D., Efficient algorithms for inferring evolutionary trees, *Networks*, *21*, 19–28, 1991.

Hennig, W., D. D. Davis, and R. Zangeri, *Phylogenetic Systematics*, University of Illinois Press, Champaign, Illinois, 1979.

Heywood, V. H. (Ed.), *Global biodiversity assessment*, p. 8, Cambridge University Press, Cambridge, UK, 1995.

Hinegardner, R., and J. Engelberg, Biological complexity, *Journal of Theoretical Biology*, *104*, 7–20, 1983.

Lenski, R. E., C. Ofria, R. T. Pennock, and C. Adami, The evolutionary origin of complex features, pp. 139–144, 2003.

Li, M., and P. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications*, Springer, 1997.

Magurran, A. E., *Measuring biological diversity*, Blackwell Publishing, 2003.

Meléndez-Hevia, E., T. G. Waddell, and M. Cascante, The puzzle of the krebs citric acid cycle: assembling the pieces of chemically feasible reactions, and opportunism in the design of metabolic pathways during evolution, *J. Mol. Evol.*, *43*, 293–303, 1996.

Milosavljević, A., and J. Jurka, Discovery by minimal length encoding: A case study in molecular evolution, *Machine Learning*, *12*, 69–87, 1993.

Milosavljević, A., D. Haussler, and J. Jurka, Informed parsimonious inference of prototypical genetic sequences, in *Proceedings of the Workshop on Computational Learning Theory (COLT)*, pp. 102–117, 1989.

Mivart, S. G. J., *On the genesis of species*, D. Appleton and Co, Macmillan, London, 1871.

Newcomb, R. D., P. M. Campbell, D. L. Ollis, E. Cheah, R. J. Russell, and J. G. Oakeshott, A single amino acid substitution converts a carboxylesterase to an organophosphorus hydrolase and confers insecticide resistance on a blowfly, *Proc. Natl. Acad. Sci. USA*, *94*, 7464–7468, 1997.

Nilsson, D., and S. Pelger, A pessimistic estimate of the time required for an eye to evolve., *Proc. Roy. Soc. London*, *256*, 53–58, 1994.

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

Ofria, C., C. Adami, and T. Collier, Selective pressures on genomes in molecular evolution, *Journal of theoretical biology*, *222*, 477–483, 2003.

Pielou, E. C., *An introduction to mathematical ecology*, chap. Ecological diversity and its measurement, John Wiley & Sons Inc, 1969.

Rambaut, A., and N. C. Grassly, Seq-gen: An application for the monte carlo simulation of dna sequence evolution along phylogenetic trees, *Comput. Appl. Biosci.*, *13*, 235–238, 1997.

Ray, T. S., Evolution, complexity, entropy, and artificial reality., *Physica D*, *75*, 239–263, 1994.

Saitou, N., and M. Nei, The neighbor-joining method: a new method for reconstructing phylogenetic trees, *Mol. Biol. Evol.*, *4*, 406–25, 1987.

Salvini-Plawen, L., and E. Mayr, On the evolution of photoreceptors and eyes, *Evol. Biol*, *10*, 207–263, 1977.

Schneider, T., Evolution of biological information, *Nucleic Acids Research*, *28*, 2794–2799, 2000.

Semple, C., and M. Steel, Tree reconstruction from multi-state characters, *Adv. Appl. Math.*, *28*, 169–184, 2002.

Sen, T. Z., et al., Predicting binding sites of hydrolase-inhibitor complexes by combining several methods, *BMC Bioinformatics*, *5*, 205, 2004.

Shannon, C. E., A mathematical theory of communication, *The Bell System Technical Journal*, *27*, 379–423,623–656, 1948.

Smith, J. M., Byte-sized evolution, *Nature*, *355*, 772–773, 1992.

Soltis, D. E., and P. S. Soltis, The role of phylogenetics in comparative genetics, *Plant Physiol.*, *132*, 1790–1800, 2003.

Williams, P. L., and W. M. Fitch, Phylogeny determination using dynamically weighted parsimony method, *Method Enzymol.*, *183*, 616 - 626, 1990.

Williforda, A., B. Staya, and D. Bhattacharya, Evolution of a novel function: nutritive milk in the viviparous cockroach, diploptera punctata, *Evolution & Development*, *6*, 67–77, 2004.

Woese, C. R., O. Kandler, and M. L. Wheelis, Towards a natural system of organisms: Proposal for the domains archaea, bacteria, and eucarya., *P. Natl. Acad. Sci. USA*, *87*, 4576–4579, 1990.