The Effect of Natural Selection on Phylogeny Reconstruction Algorithms

Dehua Hang[1], Charles Ofria[1], Thomas M. Schmidt[2], and Eric Torng[1]

[1]Department of Computer Science & Engineering
Michigan State University, East Lansing, MI 48824 USA
[2]Department of Microbiology and Molecular Genetics
Michigan State University, East Lansing, MI 48824 USA
{hangdehu, ofria, tschmidt, torng}@msu.edu

Abstract. We study the effect of natural selection on the performance of phylogeny reconstruction algorithms using Avida, a software platform that maintains a population of digital organisms (self-replicating computer programs) that evolve subject to natural selection, mutation, and drift. We compare the performance of neighbor-joining and maximum parsimony algorithms on these Avida populations to the performance of the same algorithms on randomly generated data that evolve subject only to mutation and drift. Our results show that natural selection has several specific effects on the sequences of the resulting populations, and that these effects lead to improved performance for neighbor-joining and maximum parsimony in some settings. We then show that the effects of natural selection can be partially achieved by using a non-uniform probability distribution for the location of mutations in randomly generated genomes.

1 Introduction

As researchers try to understand the biological world, it has become clear that knowledge of the evolutionary relationships and histories of species would be an invaluable asset. Unfortunately, nature does not directly track such changes, and so such information must be inferred by studying extant organisms. Many algorithms have been crafted to reconstruct phylogenetic trees - dendrograms in which species are arranged at the tips of branches, which are then linked successively according to common evolutionary ancestors. The input to these algorithms are typically traits of extant organisms such as gene sequences. Often, however, the phylogenetic trees produced by distinct reconstruction algorithms are different, and there is no way of knowing which, if any, is correct.

In order to determine which reconstruction algorithms work best, methods for evaluating these algorithms need to be developed. As documented by Hillis [1], four principal methods have been used for assessing phylogenetic accuracy: working with real lineages with known phylogenies, generating artificial data using computer simulations, statistical analyses, and congruence studies. These last two methods tend to focus on specific phylogenetic estimates; that is, they attempt to provide independent confirmations or probabilistic assurances for a specific result rather than

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2723, pp. 13–24, 2003.

evaluate the general effectiveness of an algorithm. We focus on the first two methods, which are typically used to evaluate the general effectiveness of a reconstruction algorithm: computer simulations [2] and working with lineages with known phylogenies [3].

In computer simulations, data is generated according to a specific model of nucleotide or amino acid evolution. The primary advantages of the computer simulation technique are that the correct phylogeny is known, data can be collected with complete accuracy and precision, and vast amounts of data can be generated quickly. One commonly used computer simulation program is *seq-gen* [4]. Roughly speaking, seq-gen takes as input an ancestral organism, a model phylogeny, and a nucleotide substitution model and outputs a set of taxa that conforms to the inputs. Because the substitution model and the model phylogeny can be easily changed, computer simulations can generate data to test the effectiveness of reconstruction algorithms under a wide range of conditions.

Despite the many advantages of computer simulations, this technique suffers from a "credibility gap'' due to the fact that the data is generated by an artificial process. That is, the sequences are never expressed and thus have no associated function. All genomic changes in such a model are the result of mutation and genetic drift; natural selection does not determine which position changes are accepted and which changes are rejected. Natural selection is only present via secondary relationships such as the use of a model phylogeny that corresponds to real data. For this reason, many biologists disregard computer simulation results.

Another commonly used evaluation method is to use lineages with known phylogenies. These are typically agricultural or laboratory lineages for which records have been kept or experimental phylogenies generated specifically to test phylogenetic methods. Known phylogenies overcome the limitation of computer simulations in that all sequences are real and do have a relation to function. However, working with known phylogenies also has its limitations. As Hillis states, "Historic records of cultivated organisms are severely limited, and such organisms typically have undergone many reticulations and relatively little genetic divergence." [1]. Thus, working with these lineages only allows the testing of reconstructions of phylogenies of closely related organisms. Experimentally generated phylogenies were created to overcome this difficulty by utilizing organisms such as viruses and bacteria that reproduce very rapidly.

However, even research with experimentally generated lineages has its shortcomings. First, while the organisms are natural and evolving, several artificial manipulations are required in order to gather interesting data. For example, the mutation rate must be artificially increased to produce divergence and branches are forced by explicit artificial events such as taking organisms out of one petri dish and placing them into two others. Second, while the overall phylogeny may be known, the data captured is neither as precise nor complete as that with computer simulations. That is, in computer simulations, every single mutation can be recorded whereas with experimental phylogenies, only the major, artificially induced phylogenetic branch events can be recorded. Finally, even when working with rapidly reproducing organisms, significant time is required to generate a large amount of test data; far more time than when working with computer simulations.

Because of the limitations of previous evaluation methods, important questions about the effectiveness of phylogeny reconstruction algorithms have been ignored in

the past. One important question is the following: What is the effect of natural selection on the accuracy of phylogeny reconstruction algorithms?

Here, we initiate a systematic study of this question. We begin by generating two related data sets. In the first, we use a computer program that has the accuracy and speed of previous models, but also incorporates natural selection. In this system, a mutation only has the possibility of persisting if natural selection does not reject it. The second data set is generated with the same known phylogenetic tree structure as was found in the first, but this time all mutations are accepted regardless of the effect on the fitness of the resulting sequence (to mimic the more traditional evaluation methodologies). We then apply phylogeny reconstruction algorithms to the final genetic sequences in both data sets and compare the results to determine the effect of natural selection.

To generate our first data set, we use Avida, a digital life platform that maintains a population of digital organisms (i.e. programs) that evolve subject to mutation, drift, and natural selection. The true phylogeny is known because the evolution occurs in a computer in which all mutation events are recorded. On the other hand, even though Avida populations exist in a computer rather than in a petri dish or in nature, they are not simulations but rather are experiments with digital organisms that are analogous to experiments with biological organisms. We describe the Avida system in more detail in our methods section.

2 Methods

2.1 The Avida Platform [5]

The major difficulty in our proposed study is generating sequences under a variety of conditions where we know the complete history of all changes and the sequences evolve subject to natural selection, not just mutation and drift. We use the Avida system, an auto-adaptive genetic system designed for use as a platform in digital/artificial life research, for this purpose.

A typical Avida experiment proceeds as follows. A population of digital organisms (self-replicating computer programs with a Turing-complete genetic basis) is placed into a computational environment. As each organism executes, it can interact with the environment by reading inputs and writing outputs. The organisms reproduce by allocating memory to double their size, explicitly copying their genome (program) into the new space, and then executing a divide command that places the new copy onto one of the CPU's in the environment "killing" the organism that used to occupy that CPU. Mutations are introduced in a variety of ways. Here, we make the copy command probabilistic; that is, we can set a probability that the copy command fails by writing an arbitrary instruction rather than the intended instruction.

The crucial point is that during an Avida experiment, the population evolves subject to selective pressures. For example, in every Avida experiment, there is a selective pressure to reproduce quickly in order to propagate before being overwritten by another organism. We also introduce other selective pressures into the environment by rewarding organisms that perform specific computations by increasing the speed at which they can execute the instructions in their genome. For example, if the outputs produced by an organism demonstrate that the organism can

perform a Boolean logic operation such as “exclusive-or” on its inputs, then the organism and its immediate descendants will execute their genomes at twice their current rate. Thus there is selective pressure to adapt to perform environment-specific computations. Note that the rewards are not based on how the computation is performed; only the end product is examined. This leads to open-ended evolution where organisms evolve functionality in unanticipated ways.

2.2 Natural Selection and Avida

Digital organisms are used to study evolutionary biology as an independent form of life that shares no ancestry with carbon-based life. This approach allows general principles of evolution to be distinguished from historical accidents that are particular to biochemical life. As Wilke and Adami state, “In terms of the complexity of their evolutionary dynamics, digital organisms can be compared with biochemical viruses and bacteria”, and “Digital organisms have reached a level of sophistication that is comparable to that of experiments with bacteria or viruses” [6].

The limitation of working with digital organisms is that they live in an artificial world, so the conclusions from digital organism experiments are potentially an artifact of the particular choices of that digital world. But by comparing the results across wide ranges of parameter settings, as well as results from biochemical organisms and from mathematical theories, general principles can still be disentangled. Many important topics in evolutionary biology have been addressed by using digital organisms including the origins of biological complexity [7], and quasi-species dynamics and the importance of neutrality [8]. Some work has also compared biological systems with those of digital organisms, such as a study on the distribution of epistemic interactions among mutations [9], which was modeled on an earlier experiment with E. coli [10], and the similarity of the results were striking, supporting the theory that many aspects of evolving systems are governed by universal principles.

Avida is a well-developed digital organism platform. Avida organisms are self-replicating computer programs that live in, and adapt to, a controlled environment. Unlike other computational approaches to studying evolution (such as genetic algorithms or numerical simulations), Avida organisms must explicitly create a copy of their own genome to reproduce, and no particular genomic sequence is designated as the target or optimal sequence. Explicit and implicit mutations occur in Avida. Explicit mutations include point mutations incurred during the copy process and the random insertions and/or deletions of single instructions. Implicit mutations are the result of flawed copy algorithms. For example, an Avida organism might skip part of its genome during the replication, or replicate part of its genome more than once. The rates of explicit mutations can be controlled during the setup process, whereas implicit mutations cannot typically be controlled. Selection occurs because the environment in which the Avida organisms live is space limited. When a new organism is born, an older one is removed from the population.

2.3 Determining Correctness of a Phylogeny Reconstruction: The Four Taxa Case

Even when we know the correct phylogeny, it is not easy to measure the quality of a specific phylogeny reconstruction. A phylogeny can be thought of as an edge-weighted tree (or, more generally, an edge-weighted graph) where the edge weights correspond to evolutionary time or distance. Thus, a reconstruction algorithm should not only generate the correct topology or structure but also must generate the correct evolutionary distances.

Like many other studies, we simplify the problem by ignoring the edge weights and focus only on topology [11]. Even with this simplification, measuring correctness is not an easy problem. If the reconstructed topology is identical to the correct topology, then the reconstruction is correct. However, if the reconstructed topology is not identical, which will often be the case, it is not sufficient to say that the reconstruction is incorrect. There are gradations of correctness, and it is difficult to state that one topology is closer to the correct topology than a second one in many cases.

We simplify this problem so that there is an easy answer of right and wrong. We focus on reconstructing topologies based on populations with four taxa. With only four taxa, there really is only one decision to be made: Is A closest to B, C, or D? See the following diagram for an illustration of the three possibilities. Focusing on situations with only four taxa is a common technique used in the evaluation of phylogeny reconstruction algorithms [2,11,12].

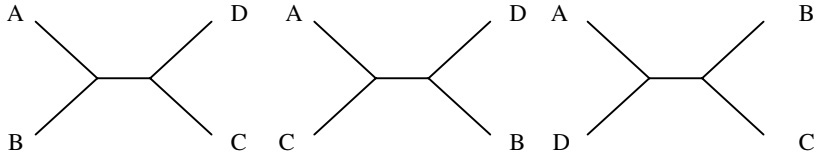


Fig. 1. Three possible topologies under four taxa model tree.

2.4 Generation of Avida Data

We generated Avida data in the following manner. First, we took a hand-made ancestor S1 and injected it into an environment E1 in which four simple computations were rewarded. The ancestor had a short copy loop and its genome was padded out to length 100 (from a simple 15-line self-replicator) with inert no-op instructions. The only mutations we allowed during the experiments were copy mutations and all size changes due to mis-copies were rejected; thus the lengths of all genome sequences throughout the execution are length 100. We chose to fix the length of sequences in order to eliminate the issue of aligning sequences. The specific length 100 is somewhat arbitrary. The key property is that it is enough to provide space for mutations and adaptations to occur given that we have disallowed insertions. All environments were limited to a population size of 3600. Previous work with avida (e.g. [16]) has shown that 3600 is large enough to allow for diversity while making large experiments practical.

After running for L1 updates, we chose the most abundant genotype S2 and placed S2 into a new environment E2 that rewarded more complex computations. Two computations overlapped with those rewarded by E1 so that S2 retained some of its fitness, but new computations were also rewarded to promote continued evolution. We executed two parallel experiments of S2 in E2 for 1.08×10^{10} cycles, which is approximately 10^4 generations. In each of the two experiments, we then sampled genotypes at a variety of times L2 along the line of descent from S2 to the most abundant genotype at the end of the execution. Let S3a-x denote the sampled descendant in the first experiment for L2 = x while S3b-x denotes the same descendant in the second experiment.

Then, for each value x of L2, we took S3a-x and S3b-x and put them each into a new environment E3 that rewards five complex operations. Again, two rewarded computations overlapped with the computations rewarded by E2 (and there was no overlap with E1), and again, we executed two parallel experiments for each organism for a long time. In each of the four experiments, we then sampled genotypes at a variety of times L3 along the line of descent from S3a-x or S3b-x to the most abundant genotype at the end of the execution. For each value of L3, four taxa A, B, C and D were used for reconstruction. This experimental procedure is illustrated in the following diagram. Organisms A and B share the same ancestor S3a-x while organisms C and D share the same ancestor S3b-x.

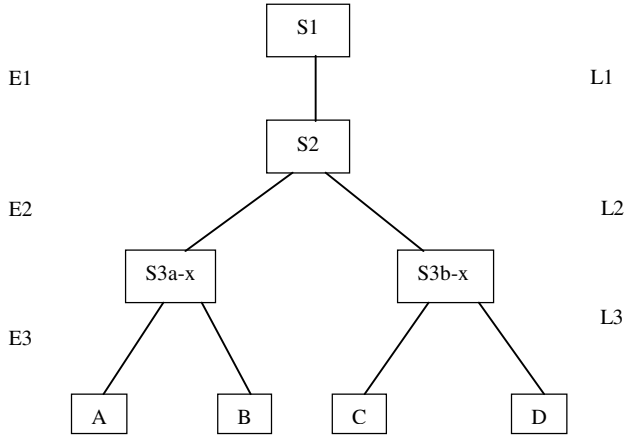


Fig. 2. Experimental procedure diagram.

We varied our data by varying the sizes of L2 and L3. For L2, we used values 3, 6, 10, 25, 50, and 100. For L3, we used values 3, 6, 10, 25, 100, 150, 200, 250, 300, 400, and 800. We repeated the experimental procedure 10 times.

The *tree structures* that we used for reconstruction were symmetric (they have the shape implied by Fig. 1). The *internal edge length* of any tree structure is twice the value of L2. The *external edge length* of any tree structure is simply L3. With six values of L2 and eleven values of L3, we used 66 different tree structures with 10 distinct copies of each tree structure.

2.5 Generation of Random Data

We developed a random data generator similar to seq-gen in order to produce data that had the same phylogenetic topology as the Avida data, but where the evolution occurred without any natural selection. Specifically, the generator took as input the known phylogeny of the corresponding Avida experiment, including how many mutations occurred along each branch of the phylogenetic tree, as well as the ancestral organism S2 (we ignored environment E1 as its sole purpose was to distance ourselves from the hand-written ancestral organism S1). The mutation process was then simulated starting from S2 and proceeding down the tree so that the number of mutations between each ancestor/descendant is identical to that in the corresponding *Avida* phylogenetic tree. The mutations, however, were random (no natural selection) as the position of the mutation was chosen according to a fixed probability distribution, henceforth referred to as the *location probability distribution*, and the replacement character was chosen uniformly at random from all different characters. In different experiments, we employed three distinct location probability distributions. We explain these three different distributions and our rationale for choosing them in Section 3.3. We generated 100 copies of each tree structure in our experiments.

2.6 Two Phylogeny Reconstruction Techniques (NJ, MP)

We consider two phylogeny reconstruction techniques in this study.

Neighbor-Joining. *Neighbor-joining (NJ)* [13,14] was first presented in 1987 and is popular primarily because it is a polynomial-time algorithm, which means it runs reasonably quickly even on large data sets. *NJ* is a distance-based method that implements a greedy strategy of repeatedly clustering the two closest clusters (at first, a pair of leaves; thereafter entire subtrees) with some optimizations designed to handle non-ultrametric data.

Maximum Parsimony. *Maximum parsimony (MP)* [15] is a character-based method for reconstructing evolutionary trees that is based on the following principle. Of all possible trees, the most parsimonious tree is the one that requires the fewest number of mutations. The problem of finding an *MP* tree for a collection of sequences is NP-hard and is a special case of the Steiner problem in graph theory. Fortunately, with only four taxa, computing the most parsimonious tree can be done rapidly.

2.7 Data Collection

We assess the performance of *NJ* and *MP* as follows. If *NJ* produces the same tree topology as the correct topology, it receives a score of 1 for that experiment. For each tree structure, we summed together the scores obtained by *NJ* on all copies (10 for Avida data, 100 for randomly generated data) to get *NJ*'s score for that tree structure.

Performance assessment was more complicated for *MP* because there are cases where multiple trees are equally parsimonious. In such cases, *MP* will output all of the most parsimonious trees. If *MP* outputs one of the three possible tree topologies (given that we are using four taxa for this evaluation) and it is correct, then *MP* gets a

score of 1 for that experiment. If *MP* outputs two tree topologies and one of them is correct, then *MP* gets a score of 1/2 for that experiment. If *MP* outputs all three topologies, then *MP* gets a score of 1/3 for that experiment. If *MP* fails to output the correct topology, then *MP* gets a score of 0 for that experiment. Again, we summed together the scores obtained by MP on all copies of the same tree structure (10 for Avida data, 100 for random data) to get *MP*'s score on that tree structure.

3 Results and Discussions

3.1. Natural Selection and Its Effect on Genome Sequences

Before we can assess the effect of natural selection on phylogeny reconstruction algorithms, we need to understand what kind of effect natural selection will have on the sequences themselves. We show two specific effects of natural selection.

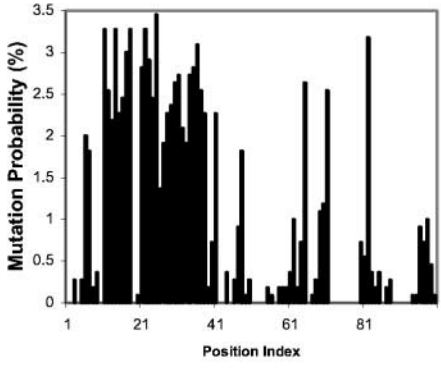


Fig. 3. Location probability distribution from one Avida run (length 100). Probability data are normalized to their percentage.

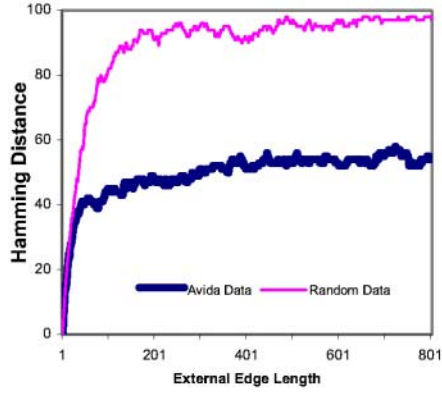


Fig. 4. Hamming distances between branch A and B from Avida data and randomly generated data. Internal edge length is 50.

We first show that the location probability distribution becomes non-uniform when the population evolves with natural selection. In a purely random model, each position is equally likely to mutate. However, with natural selection, some positions in the genome are less subject to accepted mutations than others. For example, mutations in positions involved in the copy loop of an Avida organism are typically detrimental and often lethal. Thus, accepted mutations in these positions are relatively rare compared to other positions. Fig. 2 shows the non-uniform position mutation probability distribution from a typical Avida experiment. This data captures the frequency of mutations by position in the line of descent from the ancestor to the most abundant genotype at the end of the experiment. While this is only one experiment, similar results apply for all of our experiments. In general, we found roughly three types of positions: fixed positions with no accepted mutations in the population

(accepted mutation rate = 0%); stable positions with a low rate of accepted mutations in the population (accepted mutation rate < 1%), and volatile positions with a high rate of accepted mutations (accepted mutation rate > 1%).

Because some positions are stable, we also see that the average hamming distance between sequences in populations is much smaller when the population evolves with natural selection. For example, in Fig. 4, we show that the hamming distance between two specific branches in our tree structure nears 96 (almost completely different) when there is no natural selection while the hamming distance asymptotes to approximately 57 when there is natural selection. While this is only data from one experiment, all our experiments show similar trends.

3.2 Natural Selection and Its Effect on Phylogeny Reconstruction

The question now is, will natural selection have any impact, harmful or beneficial, on the effectiveness of phylogeny reconstruction algorithms. Our hypothesis is that natural selection will improve the performance of phylogeny reconstruction algorithms. Specifically, for the symmetric tree structures that we study, we predict that phylogeny reconstruction algorithms will do better when at least one of the two intermediate ancestors will have incorporated some mutations that significantly improve its fitness. The resulting structures in the genome are likely to be preserved in some fashion in the two descendant organisms making their pairing more likely. Since the likelihood of this occurring increases as the internal edge length in our symmetric tree structure increases, we expect to see the performance difference of algorithms increase as the internal edge length increases.

The results from our experiments support our hypothesis. In Fig. 5, we show that *MP* does no better on the Avida data than the random data when the internal edge length is 6. MP does somewhat better on the Avida data than the random data when the internal edge length grows to 50. Finally *MP* does significantly better on the Avida data than the random data when the internal edge length grows to 200.

3.3 Natural Selection via Location Probability Distributions

Is it possible to simulate the effects of natural selection we have observed by the random data generator? In part 1, we observed that natural selection does have some effect on the genome sequences. For example, mutations are frequently observed only on part of the genome. If we tune the random data generator to use non-uniform location probability distributions, is it possible to simulate the effects of natural selection?

To answer this question, we collected data from 20 Avida experiments to determine what the location probability distribution looks like with natural selection. We first looked at how many positions typically are fixed (no mutations). Averaging the data from the 20 Avida experiments, we saw that 21 % are fixed in a typical run. We then looked further to see how many positions were stable (mutation rate <= 1%) and how many positions were volatile (mutation rate > 1%) in a typical experiment. Our results show that 35% of the positions are stable, and 44% of the positions are volatile.

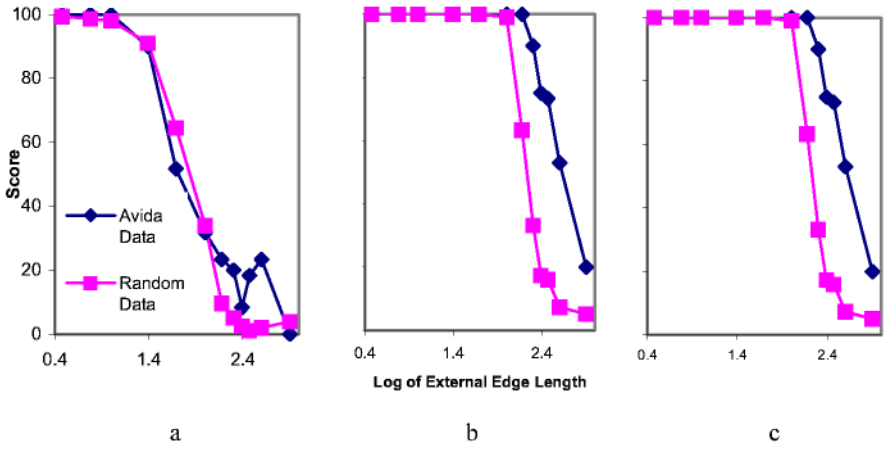


Fig. 5. *MP* scores *vs* log of external edge length. The internal edge lengths of a, b and c are 6, 50 and 200.

From these findings, we set up our random data generator with three different location probability distributions. The first is the uniform distribution. The second is a two-tiered distribution where 20 of the positions are fixed (no mutations) and the remaining 80 positions are equally likely. Finally, the third is a three-tiered distribution where 21 of the positions were fixed, 35 were stable (mutation rates of 0.296%), and 44 were volatile (mutation rates of 2.04%).

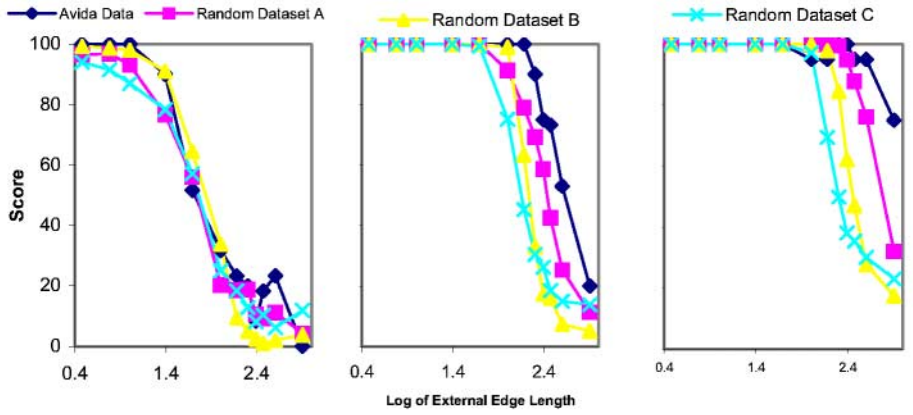
Results from using these three different location probability distributions are shown in Fig. 6. Random dataset A uses the three-tier location probability distribution. Random dataset B uses the uniform location probability distribution. Random dataset C uses the two-tier location probability distribution. We can see that MP exhibits similar performance on the Avida data and the random data with the three-tier location probability distribution.

Why does the three-tier location probability distribution seem to work so well? We believe it is because of the introduction of the stable positions (low mutation rates). Stable positions with a low probability are more likely to remain identical in the two final descendants that will make their final pairing more likely.

4 Future Work

While we feel that this preliminary work shows the effectiveness of using Avida to evaluate the effect of natural selection on phylogeny reconstruction, there are several important extensions that we plan to pursue in future work.

1. Our symmetric tree structure has only four taxa. Thus, there is only one internal edge and one bipartition. While this simplified the problem of determining if a reconstruction was correct or not, the scenario is not challenging and the full power



a b c

Fig. 6. *MP* scores from Avida data and 3 random datasets. The internal edge lengths of a, b and c are 6, 50 and 200.

of algorithms such as maximum parsimony could not be applied. In future work, we plan to examine larger data sets. To do so, we must determine a good method for evaluating partially correct reconstructions.

2. We artificially introduced branching events. We plan to avoid this in the future. To do so, we must determine a method for generating large data sets with similar characteristics in order to derive statistically significant results.

3. We used a fixed-length genome, which eliminates the need to align sequences before applying a phylogeny reconstruction algorithm. In our future work, we plan to perform experiments without fixed length, and we will then need to evaluate sequence alignment algorithms as well.

4. Finally, our environments were simple single niche environments. We plan to use more complex environments that can support multiple species that evolve independently.

Acknowledgements. The authors would like to thank James Vanderhyde for implementing some of the tools used in this work, and Dr. Richard Lenski for useful discussions. This work has been supported by National Science Foundation grant numbers EIA-0219229 and DEB-9981397 and the Center for Biological Modeling at Michigan State University.

References

1. Hillis D.M.: Approaches for Assessing Phylogenetic Accuracy, Syst. Biol. 44(1) (1995) 3–16
2. Huelsenbeck J.P.: Performance of Phylogenetic Methods in Simulation, Syst. Biol. 44(1) (1995) 17–48

3. Hillis D., Bull J.J., White M.E., Badgett M.R., Molineux L.J.: Experimental Phylogenetics: Generation of a Known Phylogeny. Science 255 (1992) 589–592
4. Ramnaut A. and Grassly N. C.: Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Comput. Appl. Biosci. 13 (1997) 235–238
5. Ofria C., Brown C.T., and Adami C.: The Avida User‘s Manual, 297–350 (1998)
6. Wilke C.O., Adami C.: The biology of digital organisms. TRENDS in Ecology and Evolution, 17:11 (2002) 528–532
7. Adami C., Ofria C., and Collier T.C.: Evolution of Biological Complexity. Proc. Natl. Acad. Sci. USA 97 (2000) 4463–4468
8. Wilke C.O., et. al.: Evolution of Digital Organisms at High Mutation Rates Leads to Survival of the Flattest. Nature, 412 (2001) 331–333
9. Lenski R.E., et. al.: Genome Complexity, Robustness, and Genetic Interactions in Digital Organisms. Nature 400 (1999) 661–664
10. Elena S.F. and Lenski, R.E.: Test of Synergistic Interactions Among Deleterious Mutations in Bacteria. Nature 390 (1997) 395–398
11. Gaut B.S. and Lewis P.O.: Success of Maximum Likelihood Phylogeny Inference in the Four-Taxon Case, Mol. Biol. Evol 12(1) (1995) 152–162
12. Tateno Y., Takezaki N., and Nei M.: Relative Efficiencies of the Maximum-Likelihood, Neighbor-joining, and Maximum Parsimony Methods When Substitution Rate Varies with Site, Mol. Biol. Evol. 11(2) (1994) 261–277
13. Saitou N. and Nei M.,: The Neighbor-Joining Method: A New Method for Reconstructing Phylogenetic Trees, Mol. Biol. Evol. 4 (1987) 406–425
14. Studier J. and Keppler K.: A Note on the Neighbor-Joining Algorithm of Saitou and Nei, Mol. Biol. Evol. 5 (1988) 729–731
15. Fitch W.: Toward Defining the Course of Evolution: Minimum Change for a Specified Tree Topology, Systematic Zoology, 20 (1971) 406–416
16. Lenski E., Ofria C., Collier C. and Adami C.: Genome Complexity, Robustness and Genetic Interactions in Digital Organisms, Nature, 400 (1999) 661–664