

# Phylogenetic Trees Using Evolutionary Search: Initial Progress in Extending Gaphyl to Work with Genetic Data

Clare Bates Congdon

Department of Computer Science  
Colby College  
5846 Mayflower Hill Drive  
Waterville, ME 04901  
ccongdon@colby.edu

Kevin J. Septor

Department of Computer Science  
Colby College  
7424 Mayflower Hill Drive  
Waterville, ME 04901  
kjseptor@colby.edu

## Abstract-

Gaphyl is an application of evolutionary algorithms (EA's) to phylogenetics, an approach used by biologists to investigate evolutionary relationships among organisms. For datasets larger than 20-30 species, exhaustive search is not practical in this domain. Gaphyl uses an evolutionary search mechanism to search the space of possible phylogenetic trees, in an attempt to find the most plausible evolutionary hypotheses, while typical phylogenetic software packages use heuristic search methods. In previous work, Gaphyl has been shown to be a promising approach for searching for phylogenetic trees using data with binary attributes and Wagner parsimony to evaluate the trees. In the work reported here, Gaphyl is extended to work with genetic data. Initial results with this extension further suggest that evolutionary search is a promising approach for phylogenetic work.

## 1 Introduction

Phylogenetics [9] is a method widely used by biologists to investigate evolutionary pathways followed by organisms currently or previously inhabiting the Earth. Our work is an application of evolutionary algorithms as the search mechanism for discovering these evolutionary hypotheses. In previous work [1], we reported on Gaphyl, an evolutionary algorithms system for phylogenetics. The original version of Gaphyl worked only with binary-valued phenotypic traits; here, we describe an extension to Gaphyl to work with genetic data.

## 2 Background on Phylogenetics

Given a dataset that contains a number of different species, each with a number of (phenotypic or genetic) attribute-values, phylogenetics software constructs phylogenies, which are representations of the possible evolutionary relationships among the given species. A phylogeny is a tree structure: The root of a tree can be viewed as the common ancestor, the leaves are the species, and subtrees are subsets of species that share a common ancestor. Each branching of a parent node into offspring represents a divergence in one or more attribute-values of the species within the two subtrees.

Phylogenies are constructed and evaluated using metrics

|           | sites |   |   |   |   |   |   |   |
|-----------|-------|---|---|---|---|---|---|---|
|           | 1     | 2 | 3 | 4 | 5 | 6 | 7 |   |
| sequences | 1     | T | T | A | T | T | A | A |
|           | 2     | A | A | T | T | T | A | A |
|           | 3     | A | A | A | A | A | T | A |
|           | 4     | A | A | A | A | A | A | T |

Figure 1: Four hypothetical seven-base sequences, from [9].

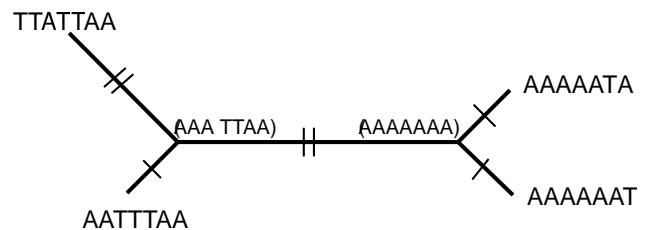


Figure 2: One possible phylogenetic tree relating the sequences in Figure 1. The sequences in parentheses are hypothesized ancestors; the hash marks indicate the number of base changes connecting adjacent sequences.

such as parsimony, in which a tree with fewer changes in the evolutionary path is considered better than one with more changes. In general, the metric used to evaluate the phylogenies embodies hypotheses about the likely paths of evolutionary change; for example, some metrics embody the assumption that species will grow only more complex via evolution – that features will be gained, but not lost, in the evolutionary process. Thus, there are several different metrics used within the phylogenetics community.

For phylogenetic models that do not incorporate assumptions about the directions of evolutionary change (the phylogenies do not model whether species A evolved into species B or vice versa), the “trees” do not have an explicit “root”, which would correspond to a hypothetical ancestor. In this case, the trees are considered “unrooted” and are often drawn as networks (although they are typically still called “trees”).

A hypothetical dataset (from [9]) is shown in Figure 1. In this example, there are four seven-base sequences (for example, representing samples from four related species); the task is to find one or more most parsimonious trees that connects these. That is, we are searching for possible descriptions of the evolutionary relationships between these species that minimize the number of changes in the genetic sequence.

Figure 2 shows one possible tree for the data in Figure 1. The tree contains four observed sequences (the leaf nodes) and the sequences for two hypothesized ancestors (interior nodes). Hash marks indicate the number of base changes between adjacent species. The tree shown is a “most parsimonious” tree for this data, requiring six base changes to relate the four species. The tree is unrooted, so drawn as a network.

Note that it is possible to have multiple “best” solutions that are equally parsimonious. With four species, as in the example above, there are only three distinct networks possible, corresponding to pairing different species in the subtrees. In list form, these trees can be represented as  $((1,2),(3,4))$ ,  $((1,3),(2,4))$ , and  $((1,4),(2,3))$ ; these trees have parsimonies of 7, 9, and 9 respectively. (Note that swapping left and right subtrees does not constitute a distinct network. For example,  $((1, 2),(3, 4))$  and  $((4,3),(1,2))$  are the same.) In this example, there is only one “most parsimonious” solution. However, in general, there may be multiple phylogenies that are “most parsimonious”, and one would like to find all of these when searching, as they correspond to equally plausible hypotheses.

A typical phylogenetics approach uses a deterministic hillclimbing methodology to find phylogenies for a given dataset, saving one or more “best” trees as the result of the process, where “best” is defined by the specific metric used (in this work, we are using parsimony). The tree-building approach adds each species into the tree in sequence, searching for the best place to add the new species. The search process is deterministic, but different trees may be found by running the search with different random orderings of the species in the dataset.

### 3 System Design

Gaphyl uses an evolutionary algorithms approach to search the space of possible phylogenies. It is constructed from existing phylogenetics software, to which the search mechanism has been replaced with evolutionary search. The system is illustrated in Figure 3.

Phylip [4] is a phylogenetics system widely used by biologists, and serves two roles in this work. On the one hand, it was used as a point of comparison to the evolutionary search approach; on the other hand, it served as the base code, to which an evolutionary mechanism was added for search. (Using the Phylip source code rather than writing our own tree-evaluation modules also helps to ensure that our trees are properly comparable to the Phylip trees.) Phylip was selected for the base code because it is well known to biologists and its source code is freely available.

Genesis [5] was used as the basis for the evolutionary mechanisms added. These mechanisms were changed considerably, as Gaphyl evolves tree structures, not bit strings. The code from Phylip is used only to evaluate the trees (using parsimony), and the EA process is used to construct the trees. In other words, the Phylip code provides the fitness function for the EA search.

Gaphyl begins with an initial population of randomly generated trees, which are passed to the Phylip routines for

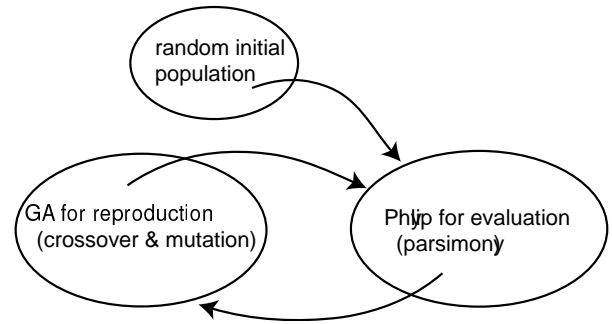


Figure 3: Gaphyl uses EA processes for searching through the space of possible phylogenies and Phylip to evaluate the phylogenies.

evaluation. Each tree is assigned a fitness corresponding to the parsimony of the tree. Trees with lower parsimony are considered more fit and are more likely to be selected as parents for the next generation. Note that for this task, we have a large search space and well defined fitness function. However, appropriate genetic operators need to be designed to facilitate the search process.

#### 3.1 The Evolutionary Search Component

A typical EA approach to doing “crossover” with two parent solutions with a tree representation is to pick a subtree (an interior or root node) in both parents at random and then swap the subtrees to form the offspring solution. A typical mutation operator would select a point in the tree and mutate it to any one of the possible legal values (here, any one of the species). However, these approaches do not work with the phylogenies because each species must be represented in the tree exactly once.

Operators designed specifically for this task are described in the following sections.

##### 3.1.1 Elitism and Canonical Form

In order to save the best solutions found while searching, both as potential parents for future generations and so that the best solutions are present at the end of the process, elitism was added to Gaphyl. A parameter specifies a percentage of the population to save from one generation to the next. These best solutions may be parents to the next generation, but will be retained unless more elite solutions are discovered.

Trees are put into a canonical form when saving the best trees found in each generation, to ensure that no equivalent trees are saved among the best ones. Canonical form is illustrated in Figure 4.

Trees are saved in their original form, which is used for reproduction. The canonical form is used primarily for elitism, and is also helpful at the end of a run, so that if there are multiple best trees, we know they are distinct solutions.

##### 3.1.2 Crossover Operator

The needs for our crossover operator bear some similarity to traveling salesperson problems (TSP’s), where each city

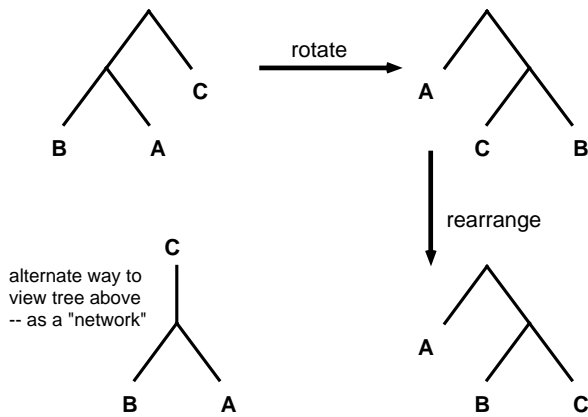


Figure 4: An illustration of putting a tree into canonical form. The tree starts as in the top left; an alternate representation of the tree as a “network” is shown at the bottom left. First, the tree is rotated, so that the first species in the dataset is an offspring of the root. Second, subtrees are rearranged so that smaller trees are on the left and alphabetically lower species are on the left.

is to be visited exactly once on a tour. There are several approaches in the literature for working on this type of problem with a EA, however, the TSP naturally calls for a string representation, not a tree. In designing our own operator, we studied TSP approaches for inspiration, but ultimately devised our own. We wanted our operator to attempt to preserve some of the species relationships from the parents. In other words, a given tree contains species in a particular relationship to each other, and we would like to retain a large degree of this structure via the crossover process.

Our crossover operator proceeds as follows:

1. Choose a species at random from one of the parent trees. Select a subtree at random that includes this node, excluding the subtree that is only the leaf node and the subtree that is the entire tree. (The exclusions prevent crossovers where no information is gained from the operation.)
2. In the second parent tree, find the smallest subtree containing all the species from the first parent’s subtree.
3. To form an offspring tree, replace the subtree from the first parent with the subtree from the second parent. The offspring must then be pruned (from the “older” branches) to remove any duplicate species.
4. Repeat the process using the other parent as the starting point, so that this process results in two offspring trees from two parent trees.

This process results in offspring trees that retain some of the species relationships from the two parents, and combine them in new ways.

An example crossover is illustrated in Figures 5 and 6.

### 3.1.3 Mutation Operators

One of our mutation operators selects two leaf nodes (species) at random, and swaps their positions in the tree.

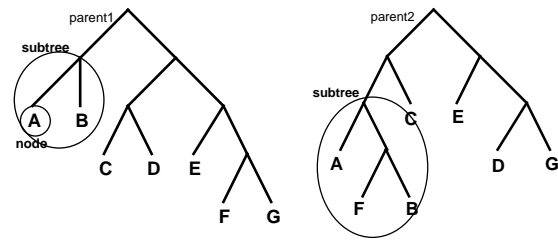


Figure 5: Two example parent trees for a phylogenetics problem with seven species. A subtree for crossover has been identified for each tree.

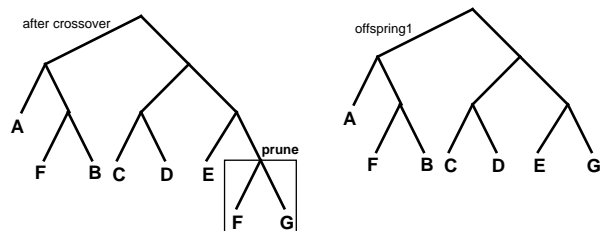


Figure 6: At the left, the offspring initially formed by replacing the subtree from parent1 with the subtree from parent2; on the right, the offspring tree has been pruned to remove the duplicate species F.

This operator allows the EA to investigate slight variations on a parent tree.

A second mutation operator picks a random subtree and a random species within the subtree. The subtree is rotated to have the species as the left child of the root and reconnected to the parent. The idea behind this operator is that within a subtree, the species might be connected to each other in a promising manner, but not well connected to the rest of the tree. The mechanics are similar to the rotation for canonical form, illustrated in Figure 4.

### 3.1.4 Immigration

Initial experiments with Gaphyl suggested that the evolutionary process seemed to be converging too rapidly — losing the diversity across individuals that enables the crossover operator to find stronger solutions. “Premature convergence” is a known problem in the EA community, and there are a number of good approaches for combatting it. In Gaphyl, we opted to implement parallel populations with immigration.

The population is subdivided into a specified number of subpopulations which, in most generations, are distinct from each other (crossovers happen only within a given subpopulation). After a number of generations have passed, each population migrates a number of its individuals into other populations; the emigrants are chosen at random and each emigrant determines at random which population it will move to and which tree within that population it will uproot. The uprooted tree replaces the emigrant in the emigrant’s original population. The number of populations, the number of generations to pass between migrations, and the number of individuals from each population to migrate at

each migration event are determined by parameters to the system. Immigration was added due to problems with premature convergence identified in early stages of development.

## 4 The Evolution of Gaphyl

Previous work, e.g., [1], used the Wagner parsimony component of Phylip, called “mix” in the source code. (This version is so called because it allows for a mixture of parsimony metrics; only Wagner parsimony was used for Gaphyl.) This work had two main thrusts:

1. To explore the viability of using an evolutionary approach to search for phylogenetic trees.
2. Once the approach was demonstrated to be viable, to investigate the effects of variations of operators and parameter settings.

As an exploration of the viability of an evolutionary approach to searching for phylogenetic trees, Gaphyl was originally designed to work with binary datasets, allowing only attributes with 0/1 values. The system was developed and evaluated using two primary datasets. The first contained 23 species and 29 attributes; the second contained 49 species and 61 attributes. The first dataset is in the realm of exhaustive search. It is therefore not surprising that Gaphyl and Phylip were able to find the same number of best trees (45 trees with parsimony of 72) in comparable search time (20 minutes). In working with the second dataset, however, Gaphyl was able to find more of the most parsimonious trees in the data, indicating that the evolutionary approach may be an advantage for some datasets. (Gaphyl found 250 trees with parsimony 279, while Phylip found 75 trees, in 24 hours of runtime; Phylip was not able to find more than 95 different trees in three days of runtime.)

The previous work with Gaphyl required development and exploration of suitable crossover and mutation operators for the task. The work here focuses on the fitness function, and does not alter the search mechanisms of the system.

### The Phylogenetics Component

The intention of previous work was to investigate the utility of evolutionary algorithms as the search method for finding phylogenies. Thus, the task used was intentionally a simple form of phylogenetics. While 0/1 character states provide a good development task, the primary interest in phylogenetics in current research is in working with genetic data. Building on the success of the initial exploration, we have extended the system to work with genetic data.

In extending Gaphyl to work with DNA data, the Phylip component called “dnaps” was incorporated as the fitness function. This module uses parsimony with DNA sequence data and is identified in the Phylip documentation as having analogous assumptions to mix:

According to the Phylip documentation [4], the following assumptions are incorporated:

1. Each site (DNA base) evolves independently.
2. Different lineages (subtrees) evolve independently.
3. The probability of a base substitution at a given site is small.
4. The expected amounts of change in different branches of the phylogeny do not vary by so much that two changes in a high-rate branch are more probable than one change in a low-rate branch.
5. The expected amounts of change do not vary enough among sites that two changes in one site are more probable than one change in another.

Note that in adding the dnaps code to the existing Gaphyl system, the evolutionary search process does not change. The primary change amounts to an alteration of the fitness function. Thus, all the work done to develop useful operators does not need to be redone, although it would be well to reevaluate the effects of operators when using qualitatively different data.

Although it is expected that favorable parameter settings might vary with specific forms of data, no changes need to be made to the EA search process. Thus, the expectation is that Gaphyl will also be a successful approach for searching for phylogenies for DNA data.

## 5 Materials and Methods

### 5.1 Datasets

The research described here was conducted using published datasets available over the internet from the TreeBase archive [2].

Recall that there is little reason to consider evolutionary search for datasets with fewer than 20 species because a problem that size can almost be exhaustively searched. Consequently, in selecting datasets, we looked for some that contained at least 30 species. Four datasets have been investigated so far, chosen because they were available in TreeBase, used DNA sequence data, and had an appropriate number of species in the dataset.

The following datasets were used for the work reported here:

1. Matrix accession number M184c3x27x98c11c12c31 in TreeBase, a study of angiosperms [3] consisting of 30 species and 3264 nucleic acid characters. (This dataset is abbreviated as M184 in the text.)
2. Matrix accession number M608 in TreeBase, a study of angiosperms [8] consisting of 50 species and 1104 nucleic acid characters.
3. Matrix accession number M194c3x30x98c09c57c27 in TreeBase, a study of polyporaceae [6] consisting of 63 species and 1588 nucleic acid characters. (This dataset is abbreviated as M194 in the text.)
4. Matrix accession number M176c9x26x97c18c16c54 in TreeBase, a study of gilled mushrooms and puffballs [7] consisting of 85 species and 3487 nucleic acid characters. (This dataset is abbreviated as M176 in the text.)

Based on previous work with Gaphyl, it appears that when the number of species is small, the hillclimbing method in Phylip is sufficient to find the best solutions and no gain is achieved through the evolutionary search processes in Gaphyl. The hypothesis is that as the number of species in the dataset increases, Gaphyl may be more likely to show a gain over the Phylip search process, perhaps by sidestepping local optima that Phylip gets stuck on.

The work done to date illustrates that Gaphyl is a promising approach for phylogenetics work, as Gaphyl finds a wider variety of trees on the 49-species binary task than Phylip does. This result suggests that Gaphyl may be able to find solutions better than those Phylip is able to find on datasets with a larger number of species and attributes, because it appears to be searching more successful regions of the search space. Thus, with the DNA version of Gaphyl, we expected in general to see Gaphyl do increasingly well as the number of species increased in the dataset.

## 5.2 Comparing Gaphyl and Phylip

It is difficult to know how to fairly compare the work done by the two systems in order to assert that one is “better” than another for a given dataset.

Both systems are far from optimized, so strong conclusions cannot be drawn from runtime alone. Recall that Gaphyl was constructed from existing systems, neither one of which was optimized for speed. In particular, Genesis was designed to simplify GA experimentation and modifications (much like the project here). It is possible to make some comparisons of operations done by the two systems in their search, but these are apples and oranges, since the work done to get from one tree to the next varies between the systems.

In previous work with the 49-species binary dataset, Phylip and Gaphyl both ran for 24 hours. In Phylip, each jumble corresponds to a hillclimbing search, which (with the 49 species) investigates on the order of 10,000 trees for each random ordering of the species list, and 40,000 jumbles in the 24 hours, or on the order of 400 million trees. In Gaphyl, 10 experiments (using different seeds to the random number generator) with a population size of 250 and 2000 generations investigates on the order of 50 million trees in 24 hours, although the number should be halved due to the 50% elitism. Thus, Gaphyl evaluates 25 million trees in the same time that Phylip evaluates 400 million trees. Gaphyl is much slower to produce the trees, but is perhaps more effective in that it searches a large space more selectively.

It is not clear that either runtime or number of trees evaluated are the proper metrics to use in comparing the two systems; neither seems quite appropriate. However, runtime is most often used by the authors as it is a practical concern when the searches must run for days.

## 6 Initial Results

With the first three datasets listed above, Gaphyl and Phylip are able to find the same set of best solutions in short runtimes; with the smaller datasets, Phylip is more expedient.

In the Gaphyl experiments, parameters were kept fixed, except for the population size, number of generations, and number of parallel populations. Based on the previous work with binary data, the experiments here were run with a crossover rate of 100%, first mutation rate of 10%, second mutation rate of 100%, and 50% elitism. When there is more than one population, five percent of the population migrates after 25%, 50%, and 75% of the generations have completed.

With dataset M184 (30 species and 3264 bases), both systems are able to find two trees with parsimony 18676. Phylip is able to find these two trees with 100 jumbles, which take about five minutes to complete. Example parameter settings for Gaphyl are population size 500, 400 generations, and one population; Gaphyl is able to find the two trees in 23 minutes.

With dataset M608 (50 species and 1104 bases), both systems are able to find one tree with parsimony 7671. Phylip is able to find this tree with 100 jumbles, which take about nine minutes to complete. Example parameter settings for Gaphyl are population size 250, 2000 generations, and 4 populations; Gaphyl requires about 3 hours to find the best tree.

With dataset M194 (63 species and 1588 bases), both systems are able to find four trees with parsimony 7497. In this case, 10,000 Phylip jumbles are needed to find the four trees and take 16 hours to complete. Example parameter settings for Gaphyl are population size 250, 2000 generations, and 2 populations; Gaphyl is able to find the four trees in 12.5 hours.

Although it is impossible to know for certain without doing exhaustive search, we have tried these three datasets with a variety of parameter settings and random seeds on both systems, and believe these to be the best solutions possible for these datasets.

These three datasets exhibit a pattern that has been observed when comparing Gaphyl and Phylip before in that smaller datasets have been observed to favor Phylip in runtime for finding the same best solutions. The performance is different from previous results, however, in that we had thought that 50 species was approximately the threshold beyond which Gaphyl might be the better performing system. In the examples above, Gaphyl is markedly slower than Phylip with 50 species, while showing a modest gain in speed with 63 species.

It seems possible that the difference from previous observations can be explained by differences in the datasets. In the datasets used previously, there were many equally best solutions. In general, Phylip seemed able to find one of the best solutions more quickly than Gaphyl was able to find one, but then Gaphyl was able to find more of the multiple bests than Phylip. Since the datasets used here seem to have only a small number of best solutions, they may be more amenable to Phylip search. In particular, the M608 dataset appears to have only one best solution and it is therefore not anomalous that Phylip is able to best Gaphyl in finding this one tree in less runtime. Thus, differences from previous observations may reflect the nature of a specific dataset and

not the difference between binary and genetic data.

However, it is also possible that there is an important difference between the binary data and the genetic data. For example, multiple equally best trees might be more common in data with binary character states, which tends to have fewer attributes and has only the two values for each attribute. The effect of this could tend to fewer differences in the observed values among species, and this would tend to result in subtrees where two species could be swapped without altering the parsimony. When two leaf nodes can be swapped without affecting the parsimony (and these are not sibling nodes), two different equally parsimonious solutions will necessarily result.

### 6.1 A new dataset

We have done little work so far with dataset M176 (85 species and 3487 bases), but will discuss it here because it appears to be a promising task for better understanding the differing capabilities of Gaphyl and Phylip.

This dataset seems complex enough that we do not believe we yet know that we have found the definitive best solutions. 1000 Phylip jumbles require 5 hours and find 6 best trees with parsimony 15876. 10,000 jumbles, taking approximately 50 hours, must be completed to see if Phylip can do better.

With parameter settings tried thus far, Gaphyl has not yet directly found trees of this fitness. For example, a population size 250, 2000 generations, and 2 populations takes 40 hours and finds a best solution of 16005; four such solutions are found. However, this experiment must be repeated with twice as many generations, because there is not evidence that the search process has converged in this time.

However, we have tried a two-step running process that was informative when developing the original version of Gaphyl. The best trees from these initial runs are harvested and used to seed a new run. Using this strategy, Gaphyl is able to find 12 trees at 15876 fitness with an additional 12 hours of runtime.

The benefit of the two-step process suggests that parallel populations could be used to a better effect in Gaphyl with this task. More work must be done to evaluate this and whether there is something different about this dataset (beyond the number of species) that can explain why multiple populations seem to be helpful here in particular.

## 7 Conclusions and Future Work

We are pleased to have constructed the DNA version of Gaphyl and have the means to now further compare the search process of Gaphyl to that of Phylip on DNA data. While the possible advantages of Gaphyl over Phylip as illustrated in this work are modest, we also have gained a few insights into the nature of problems that might be better suited to the evolutionary search mechanisms, and have more questions to pursue in the future.

So far, it seems that Gaphyl is able to find what Phylip is able to find, though sometimes requiring longer runtimes to do so. More runs with Phylip are still needed to determine

whether the converse can be said in the cases where Gaphyl may be able to find more trees.

Further work must be done in understanding the fitness landscape of phylogenetics tasks in general, and how a given dataset challenges or simplifies the search process for either one of the two systems. For example, is it possible to determine ahead of time whether Gaphyl or Phylip is a better approach for a given dataset? Working with artificial data may help to pursue these questions.

The systems must be evaluated more fully on the datasets presented here as well as a much broader range of datasets. This work should include further investigation of the current operators in Gaphyl, and whether appropriate parameter settings are different for different tasks. Also, it may prove appropriate to design new operators for the DNA search process or to extend existing mechanisms, such as the immigration process.

In previous work, the notion of finding an initial set of trees from Phylip and then continuing the process through Gaphyl was explored and found not to be beneficial for the dataset it was investigated on. This idea still appears to hold merit, and it would be good to explore it with more datasets.

## Bibliography

- [1] C. B. Congdon. Gaphyl: An evolutionary algorithms approach for the study of natural evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, pages 1057–1064, San Francisco, CA, 2002. Morgan Kaufmann.
- [2] M. J. Donoghue. Treebase: A database of phylogenetic knowledge. web-based data repository, 2000. <http://www.treebase.org>.
- [3] M. J. Donoghue and S. Mathews. Duplicate genes and the root of the angiosperms, with an example using phytochrome sequences. *Phylogenetics and Evolution*, 1998.
- [4] J. Felsenstein. Phylip source code and documentation, 1995. Available via the web at <http://evolution.genetics.washington.edu/phylip.html>.
- [5] J. J. Grefenstette. A user's guide to GENESIS. Technical report, Navy Center for Applied Research in AI, Washington, DC, 1987. Source code updated 1990; available at <http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/genetic/ga/systems/genesis/>.
- [6] D. S. Hibbett and M. J. Donoghue. Progress toward a phylogenetic classification of the polyporaceae through parsimony analysis of mitochondrial ribosomal dna sequences. *Can. J. Bot.*, 73:s853–S861, 1995.
- [7] D. S. Hibbett, E. M. Pine, E. Langer, G. Langer, and M. J. Donoghue. Evolution of gilled mushrooms and puffballs inferred from ribosomal dna sequences. *PNAS*, 94, 1997.
- [8] S. Mathews and M. J. Donoghue. The root of angiosperm phylogeny inferred from duplicate phytochrome genes. *Science*, 286(5441):947–950, 1999.
- [9] R. D. M. Page and E. C. Holmes. *Molecular Evolution: A Phylogenetic Approach*. Blackwell Science Ltd., Oxford, 1998.