
Jumpstarting phylogenetic analysis

Jesse Mecham*, Mark Clement, Quinn Snell,
Todd Freestone and Kevin Seppi

Department of Computer Science,
Brigham Young University, Provo UT 84602, USA
E-mail: jesse_mecham@byu.edu E-mail: clement@cs.byu.edu
E-mail: snell@cs.byu.edu E-mail: tfreestone@gmail.com
E-mail: kseppi@cs.byu.edu
*Corresponding author

Keith Crandall

Department of Integrative Biology,
Brigham Young University, Provo UT 84602, USA
E-mail: keith_crandall@byu.edu

Abstract: Phylogenetic analysis is a central tool in studies of comparative genomics. When a new region of DNA is isolated and sequenced, researchers are often forced to throw away months of computation on an existing phylogeny of homologous sequences in order to incorporate this new sequence. The previously constructed trees are often discarded, and the researcher begins the search again from scratch. The jumpstarting algorithm uses trees from the prior search as a starting point for a new phylogenetic search. This technique drastically decreases search time for large data sets. This kind of analysis is necessary as researchers analyse tree of life size data sets.

Keywords: phylogenetics; jumpstart; alignment.

Reference to this paper should be made as follows: Mecham, J., Clement, M., Snell, Q., Freestone, T., Seppi, K. and Crandall, K. (2006) 'Jumpstarting phylogenetic analysis', *Int. J. Bioinformatics Research and Applications*, Vol. 2, No. 1, pp.19–35.

Biographical notes: Jesse Mecham is a Master's student in the Department of Computer Science at Brigham Young University. His current research interests are in search algorithms and molecular evolution, with particular interest in intron regulation.

Mark Clement is an Associate Professor in the Department of Computer Science at Brigham Young University. He has worked with parallel-processing solutions to bioinformatics problems and has focused on sequence alignment and phylogenetic analysis.

Quinn Snell is an Associate Professor in the Department of Computer Science at Brigham Young University. His research areas are parallel processing and bioinformatics. His recent focus is on phylogenetic analysis and optimisation alignment.

Todd Freestone is a Bioinformatics Undergraduate in the Department of Integrative Biology.

Kevin Seppi is an Assistant Professor in the Department of Computer Science at Brigham Young University. He works in Machine Learning and Artificial Intelligence. His work includes the development of search algorithms, especially search algorithms for phylogenetic analysis.

Keith Crandall is a Professor in the Departments of Integrative Biology and Microbiology and Molecular Biology, and has a general interest in developing population genetic and phylogenetic methodology and testing and comparing such methodology through computer simulation. He is also Coordinator of the Bioinformatics Program at Brigham Young University.

1 Introduction

Phylogenetic analysis has become an integral part of many biological research programs. These include such diverse areas as human epidemiology (Clark et al., 1998; Templeton et al., 2005), viral transmission (Crandall, 1996), biogeography (DeSalle, 1995) and systematics (Hillis et al., 1996). With the advent of high throughput sequencing, an increasingly large volume of sequence data is becoming available. Scientists should be able to take advantage of these data and also of the research that others have performed. For example, when a new virus is detected, it should be possible to estimate a phylogenetic tree (an evolutionary history) containing all related viruses and the unknown variants, in order to answer questions such as:

- Where did this virus come from?
- When did this virus arrive in the human population?
- What are the related species from which we might derive ideas about appropriate antibodies for testing and remedies for treatment?
- Has this virus been genetically modified through natural or human induced recombinant technology?
- How is this virus evolving and what genetic changes occurred, which enabled it to successfully enter the human population? This allows us to gain insights into how we can prevent future outbreaks.

Unfortunately, this kind of phylogenetic search is currently computationally infeasible. The time it takes to perform a complete search using maximum likelihood exceeds several months with even a small number of sequences (on the order of 100–200). In the case of the SARS epidemic, and others like it, key information must be available in days or at most weeks in order for appropriate action to be taken. Much of the problem comes from the culture and software design for most phylogenetic software packages (Swofford, 1993; Goloboff, 1997; Felsenstein, 2002). These packages require the user to start a search from scratch every time a new sequence is added to the search (this is exactly the situation when a new antigen is observed). The software packages also do not allow users to share partial trees that could speed up the phylogenetic search process. This creates a culture where investigators see little or no benefit to collaborate in phylogenetic research. The jumpstarting algorithm investigated in this research allows

researchers to use previously generated phylogenetic trees to create better start tree for future searches.

Although jumpstarting may seem like an intuitive concept, there are many factors that must be considered if prior trees are actually going to be of benefit. Such factors include

- The number of taxa inserted into new search. Researchers will often sequence a new specimen and want to incorporate this sequence data into an existing search. We have found that when a small number of sequences are inserted, the trees from prior searches are of great benefit to the search with new sequences. When too many sequences are added, however, the effectiveness of jumpstarting diminishes.
- Alignment of previous searches. When new sequences are added to a search, a new alignment must normally be performed to incorporate the new data. The alignment used to generate prior trees also influences the impact these prior trees will have on the new search. Even when a poor alignment is used to generate prior trees, jumpstarting is able to gain some benefit from using these trees.
- Type of consensus used when merging trees from previous searches. When several overlapping trees are used from prior searches, it may be necessary to merge the trees to create a jumpstart tree with maximal overlap with new search sequences. Many trees may be available from prior searches with the same score. The way in which consensus is performed can greatly impact the efficacy of jumpstarting.

By investigating the influence of these factors, researchers can make correct decisions when utilising jumpstarting to speed up the generation of phylogenetic trees.

1.1 Phylogenetics

The branching pattern of ancestor/descendant relationships among species or their parts (e.g., genes) is a phylogeny. Researchers attempt to estimate these historical relationships by examining character evolution using a tree – a mathematical structure used to model the actual evolutionary history of species or their parts (Felsenstein, 2004). These inferred trees (historical branching relationships) can be represented as cladograms, where branch lengths are arbitrary and only the branching order is significant, or as phylograms, where the branch lengths are proportional to the amount of evolutionary change along the branch.

Phylogenies were historically used to classify organisms into natural evolutionary groups, based on these ancestor/descendant relationships. Indeed, great effort is currently being spent on estimating the ‘tree of life’, to quantify the biodiversity of our planet (Crandall and Buhay, 2004). However, phylogenies have also spread in use as the utility of the evolutionary framework for numerous other disciplines becomes increasingly obvious (Crandall and Buhay, 2004). For example, phylogenies are now being extensively used in the biomedical sciences including developmental biology, genomic biology, infectious disease, virology and human genetics.

Phylogenies have become essential tools in the study of the molecular epidemiology of disease agents (Pagel, 1999). A prime example of the troubles encountered when the phylogenetic approach is ignored comes from the outbreak of the West Nile Virus in New York City. This virus was responsible for multiple deaths in New York, yet the Centers for Disease Control and Prevention (CDC) initially misdiagnosed the causative agent as St. Louis encephalitis virus, owing to their lack of an appropriate phylogenetic

comparison (Enserink, 1999). The study of origins, spread and diversity of pathogens are clearly evolutionary questions. Only after the serological evidence was coupled with strong phylogenetic evidence was the West Nile Virus correctly identified as the etiological agent responsible for the encephalitis outbreak in New York (Crandall and Buhay, 2004).

Phylogenetic estimation is accomplished by optimising character change relative to some criterion over a tree. The tree for which the character data show the best optimisation is the preferred tree. Two of the principle optimisation criteria used by researchers are maximum parsimony and maximum likelihood. The parsimony criterion attempts to minimise the number of changes among a tree for shared-derived characters, while likelihood attempts to maximise the probability of change for all characters relative to some model of evolution. Each criterion has its own strengths and weaknesses. For example, maximum parsimony can incorporate insertion/deletion (indel) events and have asymmetric changes (e.g., a change from character A to character B is not the same as a change from character B to character A), whereas current implementations of maximum likelihood cannot accommodate these biological realities. Likewise, maximum likelihood can account for heterogeneity in evolutionary rates and multiple changes at the same character position, whereas maximum parsimony cannot. Thus there is, often times, heated discussion about appropriate methods to use to estimate phylogenetic relationships.

Another reason for which there is such a debate about phylogenetic methods is that their performance varies depending upon the type of data used, the number of sequences involved and the depth of the evolutionary relationships to be inferred. Exact searches, those that explore every possible tree topology for a given optimality criterion, are only possible for a very small number of taxa (on the order of 20–30). This limited search is because of the rapidly increasing number of possible trees with a modest increase of taxa (Felsenstein, 1978). The total number of (unrooted, strictly bifurcating) trees for T taxa is

$$B(T) = \prod_{i=3}^T (2i - 5).$$

So, for example, with only 50 sequences, there are 3×1074 possible trees. For the tree of life, there are estimated to be well over ten million species, yet for ten million sequences, there are $5 \times 1068,667,340$ possible trees. Therefore, the phylogeny problem is a particularly tough one that is well suited for distributed technology (because one performs the same calculations over different, independent, tree topologies) such as web-based systems that utilise distributed resources.

Phylogenetics has become an active field in and of itself (Lanciotti et al., 1999). It is an extremely exciting field where talents in mathematics, computer science and biology can be brought together to work on the problem of inferring historical relationships. A survey of the recent literature in many of the biomedical fields will attest to the ever-increasing applicability of phylogenetic analyses.

1.2 *Jumpstarting*

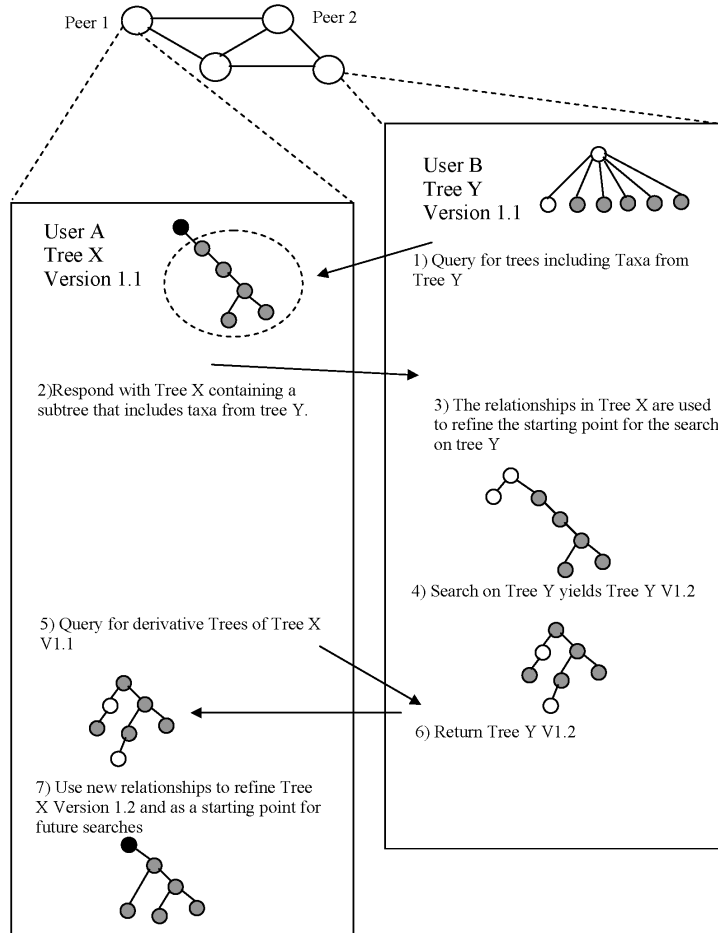
Since the phylogenetic search space is so large, it is extremely important to create search heuristics that are as efficient as possible. The jumpstarting algorithm was developed to take advantage of previous searches in order to speed up new phylogenetic computations.

Jumpstarting algorithm

- let $T = \{x \mid x \text{ is in the set of taxa involved in the new search}\}$
- query the database for prior searches with the set of taxa S_i , where at least one of the taxa in the prior search is the same as the new search x ($x \in T$ and $x \in S_i$)
- for each of these prior searches on taxa S_i , determine the intersection
- $I_i = T \cap S_i = \{x \mid x \in T \text{ and } x \in S_i\}$
- select an intersection threshold value i and select all trees from database used to create I , where $|I_i| > I$ and insert them into the set of trees N_i
- use the Newick parenthetical notation for the best tree from this maximal intersection as the base tree for the new search
- add taxa $x \in T$ where $x \notin S_i$ to all trees within N_i
- begin a normal search with the trees from N_i .

Figure 1 provides a concrete example of the jumpstart algorithm. In this example, User A on Peer 1 has performed a search resulting in Tree X version 1.1. The following steps are included in the algorithm.

- User B on Peer 2 prepares a set of taxa that will be used in a phylogenetic search and creates the data structure for Tree Y, Version 1.1. A query is sent to peer machines to determine if searches have already been performed with some of these taxa.
- Peer 1 has Tree X Version 1.1, which matches the criteria in the query. This tree is returned to Peer 2.
- Peer 2 uses Tree X Version 1.1 combined with other local taxa, to jumpstart a phylogenetic search.
- After expending significant computational resources, User B generates Tree Y Version 1.2, which refines the relationships between taxa in Tree X as well as Tree Y. This version of the tree is entered into the database.
- User A has a reference to Tree Y, since a sub-tree of Tree X was used as a jumpstart point for Tree Y. When Tree Y Version 1.2 is generated, Peer 1 can send a query for derivative trees of Tree X Version 1.1.
- Peer 2 will return Tree Y Version 1.2. User 2 may decide that all of the relationships contained in Tree Y Version 1.2 should not be made public. In this case, the sub-tree containing only the nodes originally found in Tree X would be returned.
- Peer 1 receives the refined relationships and can create Tree X version 1.2. This tree can be used for future searches.

Figure 1 Example interactions between jumpstarting peers. First graph

In this example interaction, both User A and User B have benefited from the collaboration. The tree returned from Peer 2 can be used, or discarded, depending on the value that User A places on the results. User B has been able to cut months off of his search time because of the initial jumpstart tree he/she was able to derive from Tree X Version 1.1.

2 Methods

Typical phylogenetic studies involve the collection of data, sequence alignment and phylogenetic analysis. Jumpstarting a phylogenetic analysis can improve the results in multiple stages of this process. When a new sequence is added to a data set, a new alignment must normally be computed, and the old trees that were generated in phylogenetic analysis are not normally used in further analysis. Both alignment and phylogenetic analysis are time-consuming processes (some analyses have been known to take months). When a researcher adds a sequence to the analysis, all the previous time

spent in computation is essentially thrown away because the researcher must start over again. This drastically slows the scientific process. Jumpstarting eliminates the wasted time by taking advantage of previous analyses.

2.1 Experimental set-up

In order to analyse the effectiveness of the jumpstarting algorithm, we constructed an experiment designed to simulate real phylogenetic analysis. A database was constructed to hold the phylogenetic trees returned from a search and to make them accessible to future queries. Four representative data sets were thoroughly analysed:

- *Zilla*: a 500 taxa data set, 759 nucleotide bases in length
- *Avian*: a 921 taxa data set, 1120 nucleotide bases in length
- *Three genes*: a 567 taxa data set, 2153 nucleotide bases in length
- *HIV*: a 397 taxa data set, 8583 nucleotide bases in length.

After selecting the data sets to run the experiment, we populated our database with trees based on the following algorithm:

- A for each data set D_{original} consisting of t taxa, remove n random taxa to create a new data set D_{small} consisting of $(t - n)$ taxa
- B run PAUP* (Swofford, 1993) on each D_{small} data sets for specified length of time, resulting in a set S_{small} of trees
- C insert S_{small} into the database along with all relevant information about how the trees were created.

At this point, we have a populated database that we can query for trees to use in our jumpstarting algorithm. Now we are ready to perform the jumpstarting part of our experimental analysis by:

- D taking the original n taxa that were removed in step (A) and connect them to the base of each tree in S_{small} , to create a new set of trees S_{original}
- E beginning a PAUP* (Swofford, 1993) TBR parsimony search by using the trees in S_{original} as a starting point for the search.

The above-mentioned algorithm was designed in order to simulate scenarios, where jumpstarting would be beneficial to a researcher. Some possible scenarios might include a researcher who has been conducting a ten-month phylogenetic analysis on various birds, but has just finished sequencing n new sequences that she wishes to insert into her avian data set. Another scenario might be a researcher, who has just sequenced n new HIV isolates and wants to know how these isolates are related to those already researched by colleagues at another institution. In each of these scenarios, the researcher is concerned with inserting a few new taxa into trees already constructed.

As the number of new taxa is increased, the jumpstarting algorithm should become less effective. It is important to know when jumpstarting should be abandoned and the researcher should just start a new full search. In order to answer this question, the jumpstarting algorithm was run with $n = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 50, 60, 70, 80, 90, 100\}$ for all data sets, and with

$n = \{200, 300, 400\}$ additionally for the Avian and Zilla data sets (owing to the fact that they contain a larger number of taxa). For our experiments, the Ratchet algorithm (Nixon, 1998) was used in step (2) above in order to create the S_{small} set of trees. The ratchet was allowed to run between 30 and 200 iterations and was set to weight 30% of the matrix characters. This was done to simulate the quality of trees that would be present in a researcher's database after searching on a data set for a significant amount of time.

In order to analyse the impact of alignment on jumpstarting, gaps were removed from sequences in the data set D_{original} , and the sequences were realigned with CLUSTALW using the 'fast' option (Quinn et al., 2000). This option creates a poorer alignment (which ultimately results in poorer trees). This experiment was performed to analyse the jumpstarting's ability to overcome noise introduced by poor alignment. This is important in situations where the researcher may wish to incorporate trees into a jumpstarting search from an outside source, but may not be confident in the alignment used by outside researchers.

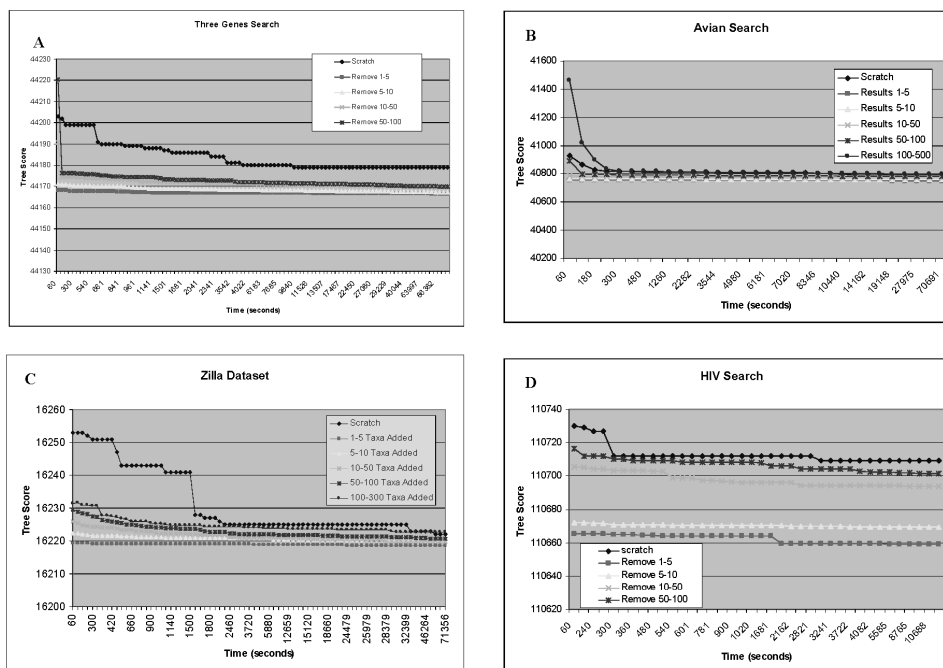
3 Results

In order to quantify the effectiveness of jumpstarting searches, we analysed the time to generate the most optimal tree using jumpstarting and compared these results with the time required to reach the most optimal tree when starting from scratch. More optimal trees have shorter length values. In all the data sets analysed, jumpstarting TBR found more optimal trees in less time than the equivalent regular TBR search. However, the time at which that score was found and the overall final score was found to be heavily dependent on a variety of factors:

- the number of taxa inserted
- the alignment quality
- the total number of taxa.

One of the greatest advantages of jumpstarting is its ability to find more optimal trees in substantially less time than a regular TBR search. In order to measure this difference, execution times were recorded for each tree found in the experiments described in Section 3.1. A 'scratch' TBR parsimony search was also performed, and the time at which all trees were discovered was recorded. By comparing the results of each experimental jumpstart run with those obtained from the scratch experiment, we are able to analyse the performance benefits of jumpstarting over regular phylogenetic searches. All scratch data sets were allowed to run for a minimum of 24 hours beyond the point they found their lowest scoring tree. In all cases, the jumpstart data sets were run for less time than the corresponding scratch data set. All experiments were run separately on a 700 MHz SGI MIPS R16000 processor using 64 GB shared memory.

The results of this analysis are shown in Figure 2. Each graph shows the results recorded from one data set. In order to make the plots more readable, results were grouped by the number of new taxa inserted (1–5, 5–10, 10–50, etc.). Each data point was then averaged with all other members of its group, and the results are plotted on the appropriate graph. The result of each respective "scratch" search is also shown for comparison.

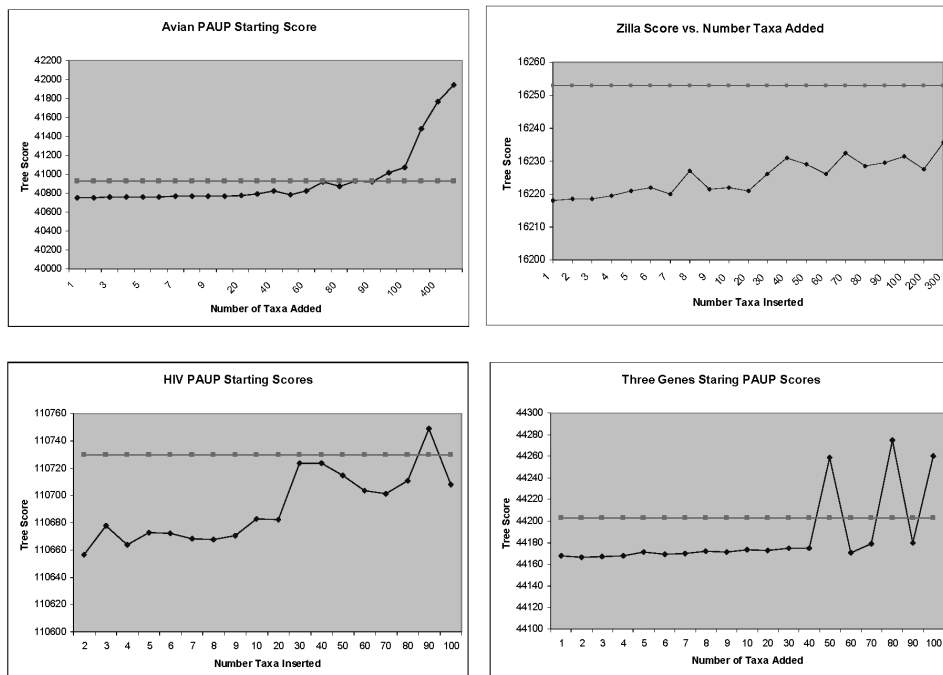
Figure 2 Timeline showing average tree score during a phylogenetic search

Each graph represents a different data set. All data series are grouped by the number of taxa inserted into the jumpstart search (1–5 taxa inserted, 5–10 taxa inserted, 10–50 taxa inserted, etc. as noted in legend of each graph). Additionally, a TBR search where no jumpstarting was used (scratch) is shown in each of the graphs. Note that for each data set, when 50 or fewer taxa were inserted, jumpstarting found better trees in significantly less time. However, it must be noted that as the number of taxa inserted increases, the effectiveness of jumpstarting decreases.

In experiments where one hundred or fewer taxa were inserted, jumpstarting returned significantly lower tree scores within the first 120 seconds than a regular TBR search was able to find over the course of the entire experiment. When ten or fewer taxa were inserted into the search, jumpstarting found a better tree in sixty seconds, than what a TBR scratch found during the entire experiment. One prime example of this is seen in the Three Genes experiment, where jumpstarting with the addition of 1–5 taxa returned a score of 44168 within sixty seconds, while the best tree using the TBR scratch method after 65 hours was returned a score of only 44179 (Figure 2). However, it should be noted that the ability of jumpstarting to quickly return low-scoring trees is inversely related to the number of new taxa inserted into the search. In other words, as the number of new taxa incorporated into a single jumpstart search increases, the effectiveness of jumpstarting will decrease.

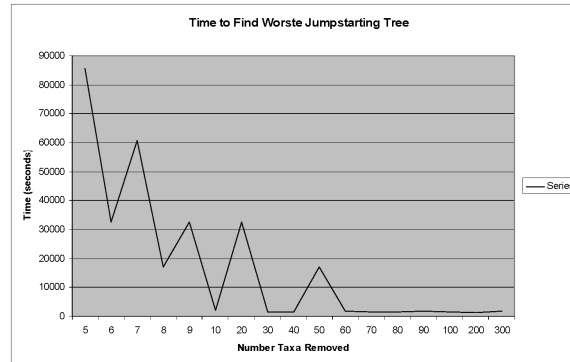
In order to understand why jumpstarting reaches its best tree score so quickly, it is helpful to observe the trends seen in the initial trees it returns. Below are the PAUP tree scores from the first sixty seconds of a jumpstart search. The scores are grouped by the number of taxa added to the jumpstart tree (Figure 3). While an observable trend shows the initial jumpstart tree score increasing with the number of new taxa incorporated, it should be noted that jumpstarting consistently begins its search with more optimal trees in each case that has less than 50 taxa incorporated into the new search. This gives us a feel for how many new taxa a researcher can insert into a jumpstarting search and still see instant measurable benefits.

Figure 3 Comparison of starting scores by number of taxa inserted into jumpstarting algorithm



Each graph represents a different data set. The horizontal line in each data set represents the starting score of a TBR search, where no jumpstarting was used. The other line in each graph represents the average starting score (y-axis) for each jumpstarting experiment, grouped by number of taxa inserted into the jumpstart search (x-axis). Note the trend that as the number of taxa inserted increases, the average starting score increases as well.

It is possible to take these results one step further and compare the speed-up jumpstarting offers over a regular TBR search by measuring the time it takes for a TBR to find the worst/initial jumpstarting tree in a corresponding experiment (Figure 4). By observing the graph, it becomes apparent that as the number of taxa incorporated into the jumpstart search increases, the overall speed-up afforded by jumpstarting decreases. When five taxa were inserted, it took regular TBR over 2.5 hours to find the same tree that jumpstarting found in 60 seconds. When more than 100 taxa were inserted, however, jumpstarting returned worse starting trees than a regular TBR search did.

Figure 4 TBR search time to find tree score equal to worst jumpstart tree score

The above chart shows the average searching time it took TBR to find a tree with the same score as the initial search tree returned from the jumpstarting algorithm. The above chart shows the scores of the Zilla data set. The plotted coordinates represent the time (x-axis) it took for a regular TBR search to achieve the average score of the initial tree for each experiment grouped by the number of taxa inserted (y-axis). Since the regular TBR search never found a tree equal to the worst jumpstarting tree when less than five taxa were inserted, only data points where five or more taxa were inserted are shown. Notice that as the number of taxa inserted increases, the time required for a regular TBR search to find an equally good tree decreases (note, make it first, not worst).

3.1 Tree quality

Although a particular algorithm may be efficient at finding a sub-optimal tree quickly, the researcher is more concerned about the final tree score than the time it took to reach sub-optimal trees. In order to analyse the quality of trees found by the jumpstarting algorithm, we compared the scores of the best tree found by using the jumpstarting algorithm with the best tree scores found by a TBR search from scratch (Figure 4).

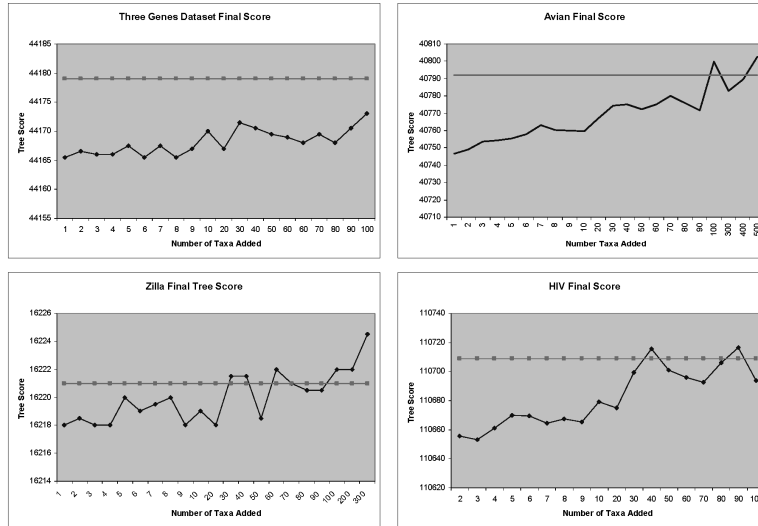
There is a general trend observed that the final tree score is inversely related to the number of taxa removed. As the number of taxa inserted into a jumpstart search increases, the average score of the best tree found during a jumpstarted phylogenetic search decreases. In all of our experiments, if fewer than thirty taxa were inserted into the search, jumpstarting found better trees than a regular TBR search.

There was a point at which the jumpstart search began to return worse trees than a regular TBR search, but it only occurred after a significant number of taxa had been added to the jumpstart search. The first occurrence of diminishing results occurred when the number of taxa inserted was greater than 6% of the total taxa included in the search (Zilla data set), while the average over all data sets was 11% of the total taxa included in the search.

3.2 Alignment quality

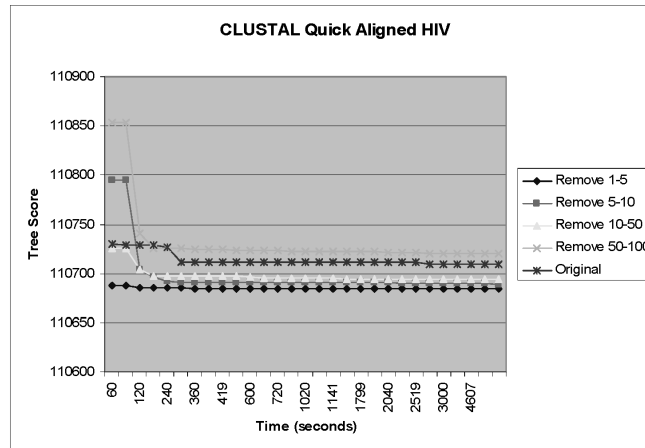
When analysing the effects of alignment on *jumpstarting*, the decision was made to test trees constructed from a sub-optimal alignments in order to assess the ability of the algorithm to eliminate noise introduced by poor alignments. Each of the experiments was analyzed based on the same metrics in other experiments: time to optimal tree and tree score (Figures 5–7).

Figure 5 Comparison of final scores by number of taxa inserted into jumpstarting algorithm



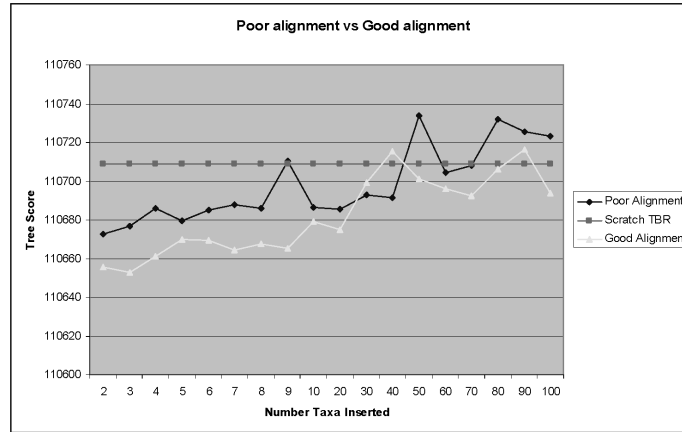
Each graph represents a different data set. The horizontal line in each data set represents the final tree score of a TBR search where no jumpstarting was used. The other line in each graph represents the average final score (y-axis) for each jumpstarting experiment, grouped by number of taxa inserted into the jumpstart search (x-axis). Note the trend that as the number of taxa inserted increases, the average starting score increases as well. It is also important to observe that jumpstarting always finds better trees, if twenty or fewer taxa are inserted.

Figure 6 Timeline showing average tree score during a TBR jumpstart search, where the jumpstart trees used were created with a poor alignment



Each graph represents a different data set. All data series are grouped by the number of taxa inserted into the jumpstart search (1–5 taxa inserted, 5–10 taxa inserted, 10–50 taxa inserted, etc. as noted in legend of each graph). Additionally, a PAUP search where no jumpstarting was used (scratch) is shown in each of the graphs. Note that for each data set, when 50 or fewer taxa were inserted, jumpstarting found better trees in less time. However, it must be noted that as the number of taxa inserted increases, the effectiveness of jumpstarting decreases.

Figure 7 Comparison of how alignment affects final scores of trees found by searching using the jumpstarting algorithm



The horizontal line in each data set represents the final tree score of a PAUP search where no jumpstarting was used. All final PAUP searches (scratch, good and poor) were done using the same final alignment. The series labelled 'poor' represents experiments where trees fed to the jumpstarting algorithm were created from a poor alignment. The poor alignment was created by using the fast option in CLUSTALW. The series labelled 'good' represents experiments where trees fed to the jumpstart algorithm were created from a good alignment (using the slow option in CLUSTALW). Note that by using jumpstart trees that were created using a poor alignment, the effectiveness of the jumpstarting algorithm decreased, yet it still outperformed a TBR PAUP search from scratch.

HIV was chosen as the example in this paper since it had the longest sequences and was most sensitive to alignment errors. In order to compare the relative change in jumpstarting performance, the results of the poorly aligned dataset were compared with the results of a good alignment (Figure 6). While the results show that jumpstarting with poorly aligned trees outperformed a 'scratch' search, the jumpstarting performance was decreased by approximately 30% when compared with trees constructed from good alignments. While there is a definite decrease in jumpstarting performance when a poor alignment is used to create the dataset, *jumpstarting* shows that it still has the robustness to outperform a regular TBR search even when the data used to feed it is not correctly aligned.

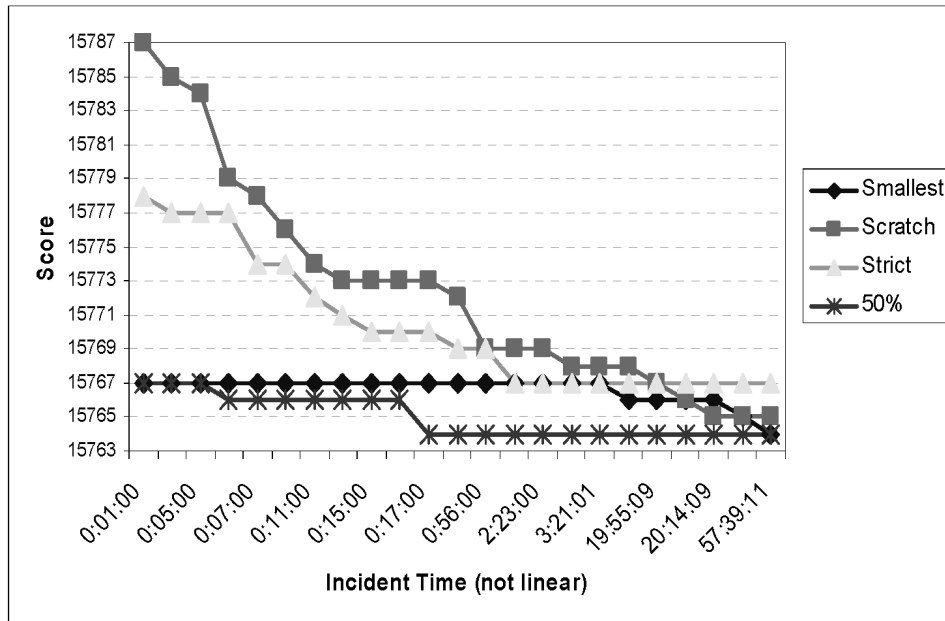
3.3 Consensus algorithms

Jumpstarting is advantageous when adding sequences to an existing analysis. However, this is not the only use for jumpstarting. More typically, a researcher may request data from the database and wish to begin computation by taking advantage of these data. A user can request all trees that contain certain taxa or sequences from the database. However, these trees may also contain extraneous taxa. One approach is to simply strip out the extraneous samples and use the resulting trees as a starting point. A consensus tree could also be used.

Figure 8 demonstrates the different jumpstarting possibilities available. A researcher may choose to start computation based on one of the most optimal (Smallest) trees

returned from the query. Optionally, a consensus tree may be created and used for jumpstarting the new search (Strict, 50%). Preliminary studies show that creating a majority rule consensus tree (50%) from the collection of most optimal trees returned by the jumpstart system seems to be the best option. Figure 8 shows that this method found the most optimal tree in 17 minutes, whereas the other choices took at least 57 hours. Increases in performance such as this are vital for advances and to apply the power of the phylogenetic approaches to studies in biomedical research.

Figure 8 Comparison of various jumpstart methodologies with the HIV data set



The graph illustrates the effect of various schemes for incorporating trees into a single jumpstarting set. Strict consensus tends to eliminate too much of the inherent structure of the jumpstarting tree, giving diminished results. However, using 50% consensus tends to show the most promising results in our experiments.

4 Conclusions

Jumpstarting is effective at improving the ability of researchers to quickly generate phylogenies. When a new epidemic strikes, it is often important to determine the relationship between the current organism and others that have been successfully treated previously. This process can take a prohibitively long period of time with current algorithms. The jumpstarting algorithm can generate superior phylogenetic relationships much more quickly than existing algorithms. These relationships can be used to make informed decisions in epidemiology and other areas.

This paper describes important factors that affect the performance of jumpstarting.

- The number of taxa inserted into new search. When an average of 30 or fewer taxa is added to a search, jumpstarting significantly outperforms searches started from scratch. The exact number is dependent on the data set and future research will define ways to define this threshold.
- Alignment of previous searches. While there is a decrease in jumpstarting performance when poor alignments are used for prior trees, jumpstarting still outperforms searches started from scratch. Future work will investigate which trees to incorporate into a search based on alignment differences.
- Type of consensus used when merging trees from previous searches. The 50% consensus shows the greatest improvement out of all consensus methods investigated. Future research will investigate additional consensus methods and their effect on jumpstarting

Altering any one of these factors can drastically affect the jumpstarting's ability to return optimal trees in minimal time. However, even in a world of imperfect data, jumpstarting can be a powerful tool used by researchers to decrease the time required to conduct phylogenetic searches and improve the optimality of the trees they produce.

Jumpstarting is an effective algorithm for improving search times and tree quality in phylogenetic analysis. It can be combined with various search algorithms to deal with important medical problems. Future work will investigate ways of mining the database and combining existing trees to provide the best starting point for future searches. We hope that these results will encourage researchers to begin collecting the trees that they have been discarding thus far, and help in developing a community which not only shares proteomic and nucleotide data, but also feels encouraged to share evolutionary data to facilitate the efforts of their colleagues.

References

- Clark, A., Weiss, K., Nickerson, D., Taylor, S., Buchanan, A., Stengard, J., Salomaa, V., Vartiainen, E., Perola, M., Boerwinkle, E. and Sing, C. (1998) 'Haplotype structure and population genetic inferences from nucleotide-sequence variation in human lipoprotein lipase', *American Journal of Human Genetics*, Vol. 63, pp.595–612.
- Crandall, K. (1996) 'Multiple interspecies transmissions of human and simian T-cell leukemia/lymphoma virus type I sequences', *Molecular Biology and Evolution*, Vol. 13, pp.115–131.
- Crandall, K.A. and Buhay, J.E. (2004) 'Genomic databases and the tree of life', *Science*, Vol. 306, pp.1144–1145.
- DeSalle, R. (1995) 'Molecular approaches to biogeographic analysis of Hawaiian Drosophilidae', in Wagner, W.L. and Funk, V.A. (Eds.): *Hawaiian Biogeography*, Smithsonian Institution Press, Washington DC, pp.72–89.
- Enserink, M. (1999) 'Groups race to sequence and identify New York virus', *Science*, Vol. 286, pp.206–207.
- Felsenstein, J. (1978) 'The number of evolutionary trees', *Systematic Zoology*, Vol. 27, pp.27–33.
- Felsenstein, J. (2002) *PHYLIP, version 3.6*, Department of Genome Sciences, University of Washington, <http://evolution.genetics.washington.edu/phylip.html>.
- Felsenstein, J. (2004) *Inferring Phylogenies*, Sinauer Associates, Sunderland, MA.

- Goloboff, P. (1997) *NONA*, Available via FTP with registration from Willi Hennig Society, http://www.cladistics.com/about_nona.htm.
- Hillis, D., Miritz, C. and Mable, B. (1996) *Molecular Systematics*, Sinauer Assc., Sunderland.
- Lanciotti, R.S., Roehrig, J.T., Deubel, V., Smith, J., Parker, M., Steele, K., Crise, B., Volpe, K.E., Crabtree, M.B., Scherret, J.H., Hall, R.A., MacKenzie, J.S., Cropp, C.B., Panigrahy, B., Ostlund, E., Schmitt, B., Malkinson, M., Banet, C., Weissman, J., Komar, N., Savage, H.M., Stone, W., McNamara, T. and Gubler, D.J. (1999) 'Origin of the West Nile virus responsible for an outbreak of encephalitis in the Northeastern United States', *Science*, Vol. 286, pp.2333–2337.
- Nixon, K. (1998) 'The parsimony ratchet: a new method for rapid parsimony analysis and broad sampling of tree islands in large data sets', *Program of the 17th Meeting of the Willi Hennig Society*, Sao Paulo, Brazil, p.59.
- Pagel, M. (1999) 'Inferring the historical patterns of biological evolution', *Nature*, Vol. 401, pp.877–884.
- Quinn, S., Whiting, M., Clement, M. and McLaughlin, D. (2000) 'Parallel phylogenetic inference', *Proceedings of Supercomputing*, Dallas, Texas, Article No. 35, ISBN: 0-7803-9802-5.
- Swofford, D. (1993) *PAUP: Phylogenetic Analysis Using Parsimony*, Smithsonian Institution, Washington DC, <http://paup.csit.fsu.edu>.
- Templeton, A.R., Maxwell, T., Posada, D., Stengard, J.H., Boerwinkle, E. and Sing, C.F. (2005) 'Tree scanning: a method for using haplotype trees in phenotype/genotype association studies', *Genetics*, Vol. 169, pp.441–453.

Bibliography

- Achard, F., Vaysseix, G. and Barillot, E. (2001) 'XML, bioinformatics and data integration', *Bioinformatics*, Vol. 17, No. 2, pp.115–125.
- Bhandarkar, M., Budescu, G., Humphrey, W., Izaguirre, J., Izrailev, S., Kale, L., Kosztin, D., Molnar, F., Phillips, J. and Schulten, K. (1999) 'BioCoRE: a collaboratory for structural biology', *Proceedings of the SCS International Conference on Web-Based Modeling and Simulation*, San Francisco, California, pp.242–251.
- BugZilla (2003) <http://bugzilla.mozilla.org/>.
- Foster, I. and Kesselman, C. (1997) 'Globus: a metacomputing infrastructure toolkit', *International Journal of Supercomputer Applications*, Vol. 11, No. 2, pp.115–128.
- Foster, I., Kesselman, C., Nick, J. and Tuecke, S. (2002) 'The physiology of the grid: an open grid services architecture for distributed systems integration', *Open Grid Service Infrastructure Working Group*, Global Grid Forum.
- GenBank (2005) *National Center for Biotechnology Information*, <http://www.ncbi.nlm.nih.gov/Genbank>.
- Gnutella (2003) <http://www.gnutella.com>.
- Grimshaw, A. and Wulf, W., (1997) 'The legion vision of a worldwide virtual computer', *Communications of the ACM*, Vol. 40, No. 1, pp.39–45.
- Judd, G., Clement, M. and Snell, Q. (1998) 'DOGMA: distributed object group metacomputing architecture', *Concurrency: Practice and Experience*, Vol. 10, No. 1, pp.1–7.
- Leitner, T. (2002) *The Molecular Epidemiology of Human Viruses*, Kluwer Academic Publishers, Norwell, Massachusetts.
- Maddison, W. and Maddison, D. (2003) *Mesquite: A Modular System for Evolutionary Analysis*, <http://mesquiteproject.org>
- Microsoft Corporation (2003) *Microsoft Netmeeting*, <http://www.microsoft.com/windows/netmeeting>.

- Morell, V. (1996) 'TreeBASE: the roots of phylogeny', *Science*, Vol. 273, p.569.
- Napster (2003) <http://www.napster.com>.
- Peterson, L. and Davie, B. (2000) *Computer Networks: A Systems Approach*, Morgan Kaufman publishers, San Francisco, California.
- Piel, W., Donoghue, M. and Sanderson, M. (1996) 'TreeBASE: a relational database of phylogenetic information', *Proceedings of the International Joint Workshop for Studies on Biodiversity*, Tsubkuba, Japan, pp.41–47.
- Semple, C. and Steel, M. (2003) *Phylogenetics*, Oxford University Press, Oxford.
- Snell, Q., Tew, K., Ekstrom, J. and Clement, M. (2002) 'An enterprise based grid resource management system', *Proceedings of the Eleventh IEEE International Symposium on High Performance Distributed Computing(HPDC-11)*, Edinburgh, Scotland, pp.83–92.
- SourceForge (2003) <http://sourceforge.net/>
- University of Cambridge (2003) *Virtual Network Computing (VNC)*, <http://www.uk.research.att.com/vnc>