Applications and Algorithms for Inference of Huge Phylogenetic Trees: a Review

Muhammad Sardaraz¹, Muhammad Tahir¹, Ataul Aziz Ikram¹, Hassan Bajwa²

¹Department of Computing and Technology, Iqra University, Islamabad, Pakistan
²Department of Electrical Engineering, University of Bridgeport, USA

Abstract  Phylogenetics enables us to use various techniques to extract evolutionary relationships from sequence analysis. Most of the phylogenetic analysis techniques produce phylogenetic trees that represent relationship between any set of species or their evolutionary history. This article presents a comprehensive survey of the applications and the algorithms for inference of huge phylogenetic trees and also gives the reader an overview of the methods currently employed for the inference of phylogenetic trees. A comprehensive comparison of the methods and algorithms is presented in this paper.

Keywords  Survey, Phylogenetic Trees, Methods for Phylogenetics

1. Introduction

Phylogenetics is the study of evolutionary relationships. It enables us to use various methods to extract necessary information from a sequence. Most of the phylogenetic analysis techniques produce phylogenetic trees. These trees represent relationships between any set of species or their evolutionary history. The most accurate tree describing the evolution of a sequence can be obtained by phylogenetic analysis. Reconstruction of phylogenetic relationship using a DNA, RNA or amino acid sequences is a hierarchal process consisting of four steps[1] i.e. sequence alignment, selection of appropriate model, tree building method and the assessment of the resulting phylogeny[2].

A variety of systems and applications[3-7] have been developed to infer phylogenetic trees. Early approaches were based on single node or single processor computers. Due to advances in high throughput sequencing technologies, publically available genomic data is increasing exponentially. This huge amount of data needs sophisticated, efficient and high throughput methods to be analysed. This requirement brings parallel and distributed computing to the field of bioinformatics. Recent systems and algorithms for sequence alignment and phylogenetics are based on parallel and distributed computing. This paper presents a comprehensive survey of the applications and algorithms for the inference of phylogenetic trees. Comparison of the methods and algorithms is also presented.

2. Tree Building Methods

Different criteria are used to classify tree building methods[8]. One way is to define them as algorithm based or criteria based. Algorithm based, as the name suggests are stepwise procedures or a series of steps, while the criteria based methods follow some criteria e.g. optimization steps. Another classification is distance methods vs. character based methods. Distance method is based on the pair wise distance using some measurement, whereas in character based method trees are derived by optimizing the distribution of the patterns for each character[1,8].

2.1. Distance Methods

Distance methods are based on pair wise distance i.e. evolutionary distance between two aligned sequences to derive a tree. To estimate the evolutionary distance an evolutionary model is applied that assumes the nature of the evolutionary changes. Pair wise distance is calculated using Maximum Likelihood (ML) estimators. The advantage of distance based method is that they are less computationally intensive than character based methods, but disadvantage is that the actual character data is discarded. Commonly used distance based methods are Un-weighted Pair Group Method with Arithmetic mean (UPGMA)[9] and Neighbour Joining (NJ)[10].

UPGMA method selects closely related pairs of sequences to build the tree. Tree branches are joined on the basis of similarity among pairs and averages of joined pairs. When divergence relates to molecular clock it builds an accurate tree topology with true branch lengths[11,12].

Neighbour joining methods use distance matrices for tree construction. It initially sums individual distances to calculate the divergence of an organism from all other organisms, and then based on this sum corrected distance matrix is evaluated. This method does not automatically yield a tree with minimum over all distance[11].
2.2. Character Based Methods

In character based methods trees are derived by optimizing the distribution of the patterns for each character. These methods use character data in all steps of the analysis and have a little similarity among each other[41]. Character based methods include Maximum Parsimony (MP)[13] and Maximum Likelihood (ML)[14].

In Maximum Parsimony method observed input sequences are explained with a minimum number of substitutions. The problem is to find a tree topology that minimizes the overall score. In many cases Parsimony method is used as they are relatively independent of nucleotide and amino acid substitutions[1,8].

Maximum Likelihood method has also been used in many systems. This method tries to find a model that has highest probability to generate the input sequence under a given evolutionary model. A likelihood of observed changes is computed and the length of the tree branch is induced by the product of likelihood. The main strength of the ML method is that we can formulate hypothesis about evolutionary relationships[1,8].

Stamatakis[11] has shown that distance based methods are faster than character based methods but are less accurate. Some studies suggest that character based methods recover the true tree or topologically closer related tree to the true tree, more frequently than the distance based method[15,16].

Maximum Likelihood has also been approximated such as Markov Chain Monte Carlo (MCMC) and its variant Metropolis-Coupled Markov Chain Monte Carlo (MC3) analysis[22]. MCMC is used to approximate the posterior probabilities of the trees. MC3 can be visualized as a set of independent searches that occasionally exchange information, which allows a search to occasionally leap a valley that would otherwise trap it on a suboptimal hill. For both types of analysis, the final output is a set of trees that the program has repeatedly visited. A majority rule consensus tree may be built from the output trees [17].

Table 1. Comparison of Phylogenetic Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighbor Joining</td>
<td>Fast</td>
<td>Due to distances information is lost. Hard in obtaining reliable estimates and the tree with minimum overall distance is not generated automatically[11,17].</td>
</tr>
<tr>
<td>UPGMA</td>
<td>Fast and Robust</td>
<td>For varied branch lengths its performance can be poor</td>
</tr>
<tr>
<td>Parsimony</td>
<td>Phylogeny under a given model is clear from likelihood</td>
<td>Depending on the search algorithm this method can be slow.</td>
</tr>
<tr>
<td>ML</td>
<td>Sometimes faster than other methods</td>
<td>Difficult in determining approximation.</td>
</tr>
</tbody>
</table>

3. Survey of the Systems and Algorithms

Olson et al.[19] presented an algorithm in which a tree consisting of n taxa is built by using a stepwise addition algorithm. A local arrangement is done in order to select the best tree from each step which then gives a search space for likely trees. For n taxa a global arrangement is invoked. The algorithm scales very well with nearly linear speedups; however a study[12] has shown that its algorithmic solutions are considered to be obsolete. A parallel version of the algorithm is presented in[20].

Swofford.D presented an algorithm, which is also based on stepwise addition algorithm[18]. In the algorithm local alignment is done for each taxon to find best tree and global arrangement is done to find final tree after n taxa are added.

Bayesian inference method is presented in[21]. Inference of the phylogeny is based on the posterior probabilities of the phylogenetic trees. It uses Markov Chain Monte Carlo (MCMC) and also its variant Metropolis-Coupled Markov Chain Monte Carlo (MC3) analysis[22]. MCMC is used to approximate the posterior probabilities of the trees. MC3 can be visualized as a set of independent searches that occasionally exchange information, which allows a search to occasionally leap a valley that would otherwise trap it on a suboptimal hill. For both types of analysis, the final output is a set of trees that the program has repeatedly visited. A majority rule consensus tree may be built from the output trees [17].

Schmidt et al.[23] introduced quartet-puzzling (QP) algorithm. The QP algorithm consists of three steps. First it finds relationship for set of four out of n sequences. The trees are then composed into an intermediate tree adding sequences. The result of this step is highly dependent on the order of sequences[17]. As a result, many intermediate trees from different input orders are constructed. From these intermediate trees a majority rule consensus tree is built in the consensus step. The algorithm is based on MPI and parallelization works efficiently. The QP algorithms have inacceptable inference time and have poor final results[11].

Guindon and Gascuel, describe an approach for phylogeny reconstruction[24]. Based on the ML method, the core of the method is an algorithm that adjusts tree topology and branch length simultaneously. Starting from an initial tree build by a fast distance based method, the algorithm modify the tree to improve the likelihood at each iteration. The algorithm is topologically accurate and fast, but introduces randomness in search due to intensive topological rearrangements.

Stamatakis et al. 2005[4] present a program to infer phylogenetic trees. It uses maximum likelihood method. It is claimed that the program allows the computation of 1000 taxan trees in less than 24 hours on single processor PC. The program uses dna pars from Phy lip[25] package to build initial parsimony tree. Simple evolutionary models are used to relate parsnaps to maximum likelihood. Then the stepwise addition of the tree is done by dna pars. The algorithm gives improved likelihood values and memory requirements are low. It is significantly less modeling flexible and does not
handle protein sequence data.

Minkher et al. [3] present a modified version of the IQPNNI [26]. The IQPNNI consists of two major steps. In the initial step the local optimum likelihood tree is obtained by combining the BIONJ[27] with the Fast Nearest Neighbour Interchange[24]. In the following step leaves are randomly removed from the tree. It reinserts these leaves by important quartet puzzling (IQP) algorithm[28]. The optimization process is repeated until no further likelihood improvements are achieved. In the modified version of the algorithm the Brent’s method to determine optimal branch length is replaced by a slight modification of Newton’s method[29]. However, Newton’s method is faster as it takes the advantage of first and second order derivatives, which can be calculated efficiently. The result returned by the Newton’s method is re-evaluated to check whether re-optimization is needed. Alternatively, re-optimization may be done using Brent’s method if necessary. As the algorithm is based on the optimization step, therefore the parallelization of optimization step is done first. This algorithm is fast and efficient but suffers from the problem of non-convergence. Also sometimes the results need to be re-optimized.

A study on the suitability of distributed computing for the analysis of large phylogenetic trees is presented in the work by Keane et al.[30]. A fully distributed cross platform to build phylogenetic tree is developed. It is based on maximum likelihood method and uses a popular library for phylogenetic analysis. The program is written in java and is independent of architecture and operating system. Hence it can be implemented in heterogeneous environment. The system does not need specialized hardware and is suitable for those researchers who cannot afford expensive hardware. The system can be implemented in university labs using spare clock cycles of the CPUs. The algorithm is based on hill climbing algorithm[20] with platform independence and distributed paradigm. The framework is easy to implement and does not need sophisticated hardware. A study, [31] has shown that the disadvantage of this system is that it takes excessive runtimes for large data sets.

The work by Yang[5] presents a package to perform phylogenetic analysis. It uses the ML method. The package has several programs i.e. BASEML, CODEML, EVOLVER, PAMP, YN00, MCMCTREE and CH2. Primitive programs for tree search are BASEML and CODEML. The trees obtained from other programs such as Phylib[25] can also be evaluated. The package has a collection of sophisticated models used for sequence evolution. The package can perform multiple functions i.e. test of phylogenetic trees, estimation of parameters likelihood ratio test (LRTs), estimation of synonymous and non-synonymous substitutions, estimation of amino acid substitution matrices, estimation of special divergence, reconstruction of ancestral sequences and generation of nucleotide codon and amino acid sequences.

Keane et al.[31] present a distributed platform for phylogenomics. It can span multiple platforms and is based on ML method. It allows the researchers to use semi-idle computers to create a virtual super computer. The platform is suitable for researchers who do not have suplicated hardware for phylogenetic analysis of large sequences. The program contains three stages i.e. model selection, tree searching and bootstrapping. The program gives the user an option to select the algorithm to analyse data sets. The algorithm uses heterogeneous non-dedicated machines and has high throughput. It lacks the features like simultaneous estimation of alignment and does not support complex models of sequence evolution.

Dereeper et al.[32] have developed a platform that transparently chains the programs to perform multiple sequence alignment, phylogenetic reconstruction and graphical representation of the inferred tree. This platform has been designed for both non-experienced as well as expert users. The program runs in three modes. In the non-specialist mode it provides a ready to use pipelining chain programs. It uses MUSCLE[33] for multiple sequence alignment, Gblocks[34] for automatic alignment curation and Phyml[24] for tree building. All parameters are present by default in this mode. In advance mode the pipeline is same as the one click mode but here the user can edit the setting according to his/her own requirements. In the ‘A la Carte’ mode the interface remain same as in advance mode except that it offers the possibility of running and testing efficiently large methods. It runs on dedicated server and have input and output limitations for some programs.

Matthews and Williams in[6] developed an algorithm by using MapReduce[35]. This algorithm uses multi-core platform. It generates a t x t Robinson Fold (RF) matrix between evolutionary trees. The platform uses MapReduce in a nonstandard way; typically in MapReduce framework the final output representation is smaller than initial input. But in the algorithm the output size is much larger than the input. In the first map stage hash table is generated by the MapReduce that is every bipartition is given a unique identifier. HashBase is updated by each mapper to create a local hash table.

A modification of Phylib package has been done by Ropelewskiet al.[36], using MPI to enable large scale phylogenetic studies of protein sequences. It is based on increasing the number of bootstraps replication that can be performed on large scale protein data sets. This system uses Maximum Parsimony, Distance Matrix and Maximum Likelihood methods. The methodology to parallelize the Phylib program has been discussed and it performance is measured. Phylib supports heuristics and algorithms as well. The package has sampling techniques such as bootstrapping, jackknifing and permutation of characters. This package is an initial parallelization of the bootstrapped calculations on various data sets. It uses distance matrix and parsimony methods for large protein families and maximum likelihood method for moderate families. The program is fast and supports different methods for tree search. It is used only for protein sequences and the file input and output also limits the code.

Tiffani. et al.[7] present a suite of web tools for molecular evolution, phylogenomics, and hypothesis testing. It inte-
igrates many applications for evolutionary analysis, format conversion, file storage, and results editing. It guides the user through the analysis for steps to follow. It provides integration of applications both to expert and non-expert users. The five major tasks performed by the suite are sequence alignment, phylogeny, evolutionary tests, pipeliner, and utilities. It uses Distance Method, Maximum Parsimony, Maximum Likelihood, and Bayesian Methods for tree reconstruction. The suite uses Phylip package for sequences and MrBayes for Bayesian phylogeny. For Maximum Likelihood Phylip, Tree-Puzzle, and PhyML packages are used.

Chen. et al.[37] present a program to construct phylogeny. Based on mixture model, the program implements a bootstrap procedure with majority rule consensus. The program uses binary sequence data to construct phylogeny. It is claimed that the program is efficient than classic methods but it is time consuming.

Tamura. et al.[38] present a program for mining online databases, sequence alignment and phylogenetic trees. The program is based on ML, Distance Method and Maximum Parsimony Method. The program offers computational efficiency and accuracy of estimates for inferring. A comparison of the approaches discussed including types of architecture, method used, as well as strengths and weaknesses of each approach is given in Table 2.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Applicable to</th>
<th>Architecture</th>
<th>Method Used</th>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>fastDNAml</td>
<td>Inference of phylogenetic trees</td>
<td>Sequential, Parallel</td>
<td>ML</td>
<td>Scales very well with Nearly linear speedups</td>
<td>Its algorithmic solutions are considered obsolete and some algorithms have outperformed it [12]</td>
</tr>
<tr>
<td>PAUP</td>
<td>Phylogenetic inference</td>
<td>Sequential</td>
<td>ML</td>
<td>Has good speed on small data sets</td>
<td>The algorithms used are obsolete.</td>
</tr>
<tr>
<td>MrBayes[21]</td>
<td>Phylogenetic inference</td>
<td>Sequential</td>
<td>Bayesian Method</td>
<td>Fast</td>
<td>Final likelihood may not be accurate</td>
</tr>
<tr>
<td>TREE-PUZZLE</td>
<td>Phylogenetic trees</td>
<td>Parallel</td>
<td>Quartet based, ML</td>
<td>Based on MPI and parallelization works efficiently</td>
<td>The QP algorithms have unacceptable inference time and have poor final results [11]</td>
</tr>
<tr>
<td>RAxML[5]</td>
<td>Inference of phylogenetic trees</td>
<td>Parallel</td>
<td>ML</td>
<td>Gives improved likelihood-values and has low memory requirements</td>
<td>It is significantly less modeling flexible It does not handle protein sequences.</td>
</tr>
<tr>
<td>pIQPNNI[4]</td>
<td>Inference of phylogenetic trees</td>
<td>Parallel</td>
<td>ML</td>
<td>Algorithm is fast and Efficient</td>
<td>Suffers from the problem of non-convergence</td>
</tr>
<tr>
<td>DPRml[30]</td>
<td>Phylogenetic analysis</td>
<td>Distributed</td>
<td>ML</td>
<td>uses idle clock cycles.</td>
<td>Takes long time for large datasets [31]</td>
</tr>
<tr>
<td>MultiPhyl [31]</td>
<td>Phylogenetics</td>
<td>Distributed</td>
<td>ML</td>
<td>High throughput. Uses heterogeneous non-dedicated machines.</td>
<td>Lacks the features like simultaneous estimation of alignment. It does not support complex models of sequence evolution</td>
</tr>
<tr>
<td>Phylogeny.fr [32]</td>
<td>MSA and phylogenetic Reconstruction</td>
<td>Web server</td>
<td>ML</td>
<td>Easy to use and runs in different modes</td>
<td>Runs on dedicated server and have input and output limitations</td>
</tr>
<tr>
<td>MrsRF[6]</td>
<td>Phylogenetic analysis</td>
<td>Parallel</td>
<td>DistanceMatrix</td>
<td>Fast</td>
<td>Hard to implement</td>
</tr>
<tr>
<td>MPI-PHYLIP</td>
<td>Phylogenetic study of protein sequences</td>
<td>Parallel</td>
<td>MP, Distance, Matrix, and ML</td>
<td>The program is fast and supports different methods for tree search</td>
<td>Used only for protein sequences and the file input and output also limits the code</td>
</tr>
<tr>
<td>Phylemon 2.0 [7]</td>
<td>Molecular evolution, phylogenetics, phylogeneomics, hypothesis testing</td>
<td>Web server</td>
<td>Distance Method, ML, and Bayesian Method</td>
<td>Provides necessary applications to both expert and non-expert users</td>
<td>Needs dedicated hardware.</td>
</tr>
</tbody>
</table>
4. Results and Discussion

Performance analysis of a program for phylogenies can be done using some qualitative and quantitative parameters. The qualitative measurements include the ability of the program to handle different types of sequences i.e. DNA or amino acid sequences, as well as the ability to read sequences in different available formats etc. The quantitative measures include parameters such as the execution time of the program, final quality of the tree, and memory requirements when the size of the phylogenies is large. It is also important to know that how to be confident of the accuracy of the results obtained. Different ways may be used to analyse performance of a program. One way to analyse performance of a program is to take a set of real world data including the best known trees and a set of parameters for various programs. Another way is to take the simulated data and use different scoring functions to analyse the performance of a program[11].

A study[17] has shown comparative results of different sequential programs, PAUP[18], TREE-PUZZLE[23], fastDNAml[19], and MrBays[21]. The survey is based on simulated data and results show that MrBays outperforms other analysed programs in terms of speed and quality[11]. Comparative studies have been done using multiple sets of programs with simulated and real world data[39]. It has been found that among traditional and Bayesian approaches for phylogenetic trees MrBays and Phyml are fastest and accurate approaches[11].

Another study[31] has shown the comparative results of MultiPhyl[31], Phyml[24], DPRml[30], IQPNNI[26], and RaxML[4]. The results are calculated using a single processor machine and are shown in Table 3. The results show that for final likelihood values both MultiPhyl and Phyml perform similar and better than other analyzed programs. In half data sets Phyml achieves high likelihood values, whereas in the remaining half MultiPhyl achieves high final likelihood values. In the remaining programs RaxML outperforms the other analyzed programs in terms of final likelihood values. The runtime results show that on smaller data sets both Phyml and MultiPhyl comparable times but for large data sets Phyml outperforms MultiPhyl. There is also a clear difference in runtime of RaxML and Phyml. These programs are based on faster heuristics and have better performance as compared to traditional Maximum Likelihood programs. It is still time consuming task to perform phylogenetic analysis of large scale data on a single processor. One solution of this problem is to use parallel processing to perform high throughput phylogenetic analysis. This solution is however expensive as dedicated hardware is required. This can be avoided if non-dedicated heterogeneous processors are used [31]. Parallel and distributed computing addresses the problem of computational complexity and speeds up the algorithms for phylogenies.

5. Conclusions

Phylogenetic analysis tells us about the relationship among different organisms and their evolutionary history. The analysis of publically available genomic data needs efficient and high throughput programs. This article explored various methods for phylogenetics and various approaches for inference of huge phylogenetic trees along with the methods and approaches used. As the genomic data is increasing exponentially; the developed approach must be computationally less intensive and must have low memory requirements. These goals can be achieved by constructing good heuristics that run on parallel systems with low memory requirements.

---

**Table 3.** Comparison of final log likelihood values and runtimes (seconds in brackets) of MultiPhyl v1.0.4, RAxML-VI, Phyml v2.4.4, DPRml and IQPNNI v3.0 for a number of previously published datasets[31]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MultiPhyl(NNI)</th>
<th>Phyml2.4.4</th>
<th>DPRml</th>
<th>MultiPhyl (SPR)</th>
<th>IQPNNI v3.0</th>
<th>RAxML-VI</th>
</tr>
</thead>
<tbody>
<tr>
<td>50SC</td>
<td>-43664 (78)</td>
<td>-43691 (66)</td>
<td>-43735 (5994)</td>
<td>-43651 (6777)</td>
<td>-43630 (443)</td>
<td>-43630 (69)</td>
</tr>
<tr>
<td>101SC</td>
<td>-73807 (163)</td>
<td>-73818 (145)</td>
<td>-73754 (82935)</td>
<td>-73790 (13674)</td>
<td>-73648 (1703)</td>
<td>-73610 (602)</td>
</tr>
<tr>
<td>150SC</td>
<td>-44178 (162)</td>
<td>-44138 (113)</td>
<td>-44082 (141211)</td>
<td>-44172 (13931)</td>
<td>-44061 (2513)</td>
<td>-44024 (342)</td>
</tr>
<tr>
<td>150ARB</td>
<td>-76472 (482)</td>
<td>-76489 (323)</td>
<td>-76571 (309434)</td>
<td>-76472 (17441)</td>
<td>-76473 (4117)</td>
<td>-76473 (814)</td>
</tr>
<tr>
<td>193V</td>
<td>-64799 (235)</td>
<td>-64532 (263)</td>
<td>n/a</td>
<td>-64802 (13411)</td>
<td>-64413 (2511)</td>
<td>-64407 (905)</td>
</tr>
<tr>
<td>200ARB</td>
<td>-103741 (1003)</td>
<td>-103789 (395)</td>
<td>n/a</td>
<td>-103740 (17147)</td>
<td>-103784 (6470)</td>
<td>-103696 (1487)</td>
</tr>
<tr>
<td>218RDPII</td>
<td>-155967 (1049)</td>
<td>155876 (535)</td>
<td>n/a</td>
<td>-155934 (17233)</td>
<td>-155607 (11508)</td>
<td>-155603 (2937)</td>
</tr>
<tr>
<td>250ARB</td>
<td>-130530 (1262)</td>
<td>-130488 (619)</td>
<td>n/a</td>
<td>-130518 (17570)</td>
<td>-130271 (15226)</td>
<td>-130287 (3575)</td>
</tr>
</tbody>
</table>

---

**REFERENCES**


