PHYLOGENETIC SUPERTREE CONSTRUCTION USING TRIPLETS

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

SATYA GOVINDU AMUJURU

M.G. CENTRAL LIBRARY
ACC No. G20937
Date 5/9/11
I.I.T. ROORKEE

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JUNE, 2011

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation entitled **"PHYLOGENETIC SUPERTREE CONSTRUCTION USING TRIPLETS"** towards the partial fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science and Engineering** submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand (India) is an authentic record of my own work carried out during the period from July 2010 to June 2011, under the guidance of **Dr. Rajdeep Niyogi, Assistant Professor,** Department of Electronics and Computer Engineering, IIT Roorkee.

The matter presented in this dissertation has not been submitted by me for the award of any other degree of this or any other Institute.

Date: 29\06\2011
Place: Roorkee

A. Satya Govindu
(SATYA GOVINDU AMUJURU)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 29\06\2011
Place: Roorkee

(Dr. Rajdeep Niyogi)
Assistant Professor
Department of Electronics and Computer Engineering
IIT Roorkee.

ACKNOWLEDGEMENTS

First and foremost, I would like to extend my heartfelt gratitude to my guide and mentor **Dr. Rajdeep Niyogi**, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for his invaluable advices, guidance, encouragement and for sharing his broad knowledge. His wisdom, knowledge and commitment to the highest standards inspired and motivated me. He has been very generous in providing the necessary resources to carry out my research. He is an inspiring teacher, a great advisor, and most importantly a nice person.

I am greatly indebted to all my friends, who have graciously applied themselves to the task of helping me with ample moral supports and valuable suggestions.

On a personal note, I owe everything to the Almighty and my parents. The support which I enjoyed from my father, mother and other family members provided me the mental support I needed.

SATYA GOVINDU AMUJURU

ABSTRACT

Phylogenetics deals with evolutionary relatedness between various species. Origin of species can be established by studying this relationship. The central task in phylogenetics is to infer this relationship among a given set of species. These relationships are usually represented by a phylogenetic tree with the species of interest at the leaves and where the internal vertices of the tree represent ancestral species. The amount of available molecular data is increasing exponentially and, given the continual advances in sequencing techniques and throughput, this explosive growth will likely to continue.

Biologists assemble large multi-gene data set from available vast amount of data for use in phylogenetic analyses which imposes distinct computational challenges. As this biological data is vast it is infeasible to construct a large tree for analysis. As a result, supertree methods have been the focus of much research. Supertree methods comprise one approach to reconstructing large phylogenies, given estimated trees for overlapping subsets of the entire set of taxa, using various algorithmic techniques.

Several supertree methods have previously been developed. In this report implementation of Triplet Inference and Local Inconsistency (TILI) and Triplet Supertree Heuristic algorithms has been done. Comparative study is carried out on the basis of execution time and Accuracy measurement. The obtained results infer that Triplet Supertree Heuristic algorithm is faster and accurate than the TILI.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Introduction of Phylogenetics

A fundamental concept of the theory of evolution, independently developed by Charles Robert Darwin and Alfred Russel Wallace is that species share a common origin and have subsequently diverged through time. Interestingly, both men came to use the metaphor of a great tree to illustrate this notion of descent with modification; ever since, biologists have been using treelike diagrams to describe the pattern and timing of events that gave rise to the earth's biodiversity.

According to Pennisi today biologists have catalogued about 1.7 million of species and they estimates of the total number of species ranges from 4 to 100 million [1]. With this explosion in the amount of data in taxonomy it is no longer possible to analyze and build trees by hand. The fact that all life on Earth is genetically related is one of the most profound scientific observations of all time.

Scientists have observed that genealogical relationships amongliving things can (generally) be represented by a vast evolutionary tree. Constructing this tree of all life is a fundamental scientific problem facing human kind today. Tree Of Life (TOL) is a project (refer http://www.tolweb.org/tree) todiscover the relationships among all the species on Earth. This project presents oneof the greatest challenges of science in reconstructing the evolutionary history ofevery living organism on the earth.

Under TOL many smaller projects such as"The Green Tree of Life", and "Assembling the Fungal Tree of Life (AFTOL)"are also proposed. Huge support is also provided from different funding agenciesand countries to the development of computational methods to reconstruct thetree of life (refer http://www.phylo.org).The fieldresponsible for this undertaking is phylogenetics.Trees that depictevolutionary relationships between species, or other entities, are called phylogenetic trees orphylogenies.

In earlier days,phylogenies were built using morphological data i.e. for classifying seabirds mightinvolve comparison of beak shapes or other distinct physical attributes.Withthe discovery

of DNA and the design of sequencingtechniques, a new kind of information became available: molecular data.The increase in the availability ofDNA and protein sequence data has increased interest in molecular phylogeneticsand classification.Molecular phylogenetics overcomes limitations of morphological phylogenetics such as, convergent evolution, finding the relationship amongbacteria and the comparison of distinctly related organisms [2].

The successes of molecular biology and genetics have yielded large amounts of datathat are increasingly quantitative in nature, both in number oftaxa as well as sequence length, available for phylogeneticconstruction.Most of the inferences in comparative biology dependon accurate estimates of evolutionary relationships [3, 4].Consequently there is a growing need for new techniques to speed up this process accurately.

Reconstruction of ancestral relationships from contemporary data is widely used to provide evolutionary and functional insights into biological system. These insights are largely responsible for the development of new crops in agriculture, drug design and in understanding the ancestors of different species.

The other applications of phylogenetics include:

- In tracking the origins and development of humans over time [5]
- Understanding of the process of evolution of different species, constraints,behavior and evolutionary time [6, 7].
- Performing functional prediction of genes. Most of the similarly looking genes show the similar functionality, looking at the evolutionary history of the newly sequenced gene will help in predicting its functionality [8].
- Development of vaccines, antimicrobial and herbicides [9, 10].
- Forensics studies [11].

In a typical molecular phylogenetic analysis, a tree on aparticular set of species is constructed by first collecting the DNA or protein sequences for a homologous gene in each species, and then using those sequences toconstruct a tree on that set of species. (Homology is shared evolutionary history,and genes are called homologous if they descended from a common ancestor gene.)Using this sort of process to construct the Tree of Life is not feasible.

There are two main approaches for analyzing large number of gene datasets: constructing a tree by analyzing the entire dataset as a whole, or constructing a tree on each gene dataset separately and then combining those trees intoa single tree on the entire set of species. The first one is called the combined analysis, "total evidence" approach, and the second one is called the supertreeapproach, and a method that combines trees with overlapping leaf sets into a singletree is called a supertree method.

Supertrees are meant to contain the relations between all species in aset of smaller trees, which are combined using a supertree method. Thereare several advantages to supertree methods. The first is that whenever summarizing the results of available studies on (subsetsof) a particular group of interest when access to the sequences is not possible. Second, producing a tree from disparate data-types, such as molecular, morphological, andgene-order data, that requires independent types of analysis (and therefore prohibits acombined analysis). Butsupertrees can be created from those small trees using any one of the supertree method. Since how the input trees have been made isnot important to a supertree method. Third is that if it is infeasible toinfer a tree for a large set of species in one go, it might be possible to infertrees for subsets of the species.These can then be combined in a supertreein manageable time.

The use of supertree methodsto solve above such problems has received much criticism [12]. For example, many supertreemethods have been shown to have a size bias, favoring the relationships supportedby larger input trees over smaller ones. some have also beenshown to have a shape bias, producing either balanced or unbalanced trees moreoften [13], and the input trees on which supertrees are constructed may not be completely independent (some primary data may have beenused in the inference of more than one input tree). However, given the limitations of current sequence-based phylogenetic reconstructionmethods, supertrees are presently a necessary tool for many phylogenetic problems [14]. In fact, most, if not all, detailed estimates of the Tree of Life to date has been constructed using a supertree approach: using informal supertreemethods, splicing together various phylogenetic trees and tree representations oftaxonomies, to achieve a single tree on all currently known species.

1.2 Motivation

The central task in phylogenetics is to infer the evolutionary relationshipsamong a given set of species. These relationships are usually represented by aphylogenetic tree with the species of interest at the leaves and where the internalvertices of the tree represent ancestral species.

The large applications and huge available sequence data, reflect the need fordeveloping fast and accurate methods for constructing the phylogenies. Mostof the existing methods accept molecular sequence data or distance data forthe construction of the phylogenetic history.

Advances in molecular biology and genomic lead to the large amounts of data,both in number of taxa as well as sequence length, available for phylogeneticconstruction. Moreover, most of the inferences in comparative biology dependon accurate estimates of evolutionary relationships. In this scenariomethods, such as maximum likelihood, parsimony search, orquartet puzzling, play an important role. But they are very slow and cannot scale up to the size of the genetic data available. These methods compare treesaccording to a specific criterion. Ideally, all possible trees should be compared. But they are very slow and cannotscale up to the size of the genetic data available.

After generating the genomic data of thousands of living organisms, nowcomes the task of classifying them and making a single tree of life. This is achallenging task and till date no algorithm exists that can compute most accuratetree of life. The existing phylogenetic reconstruction algorithms cannot be usedfor this task as they suffer with poor efficiency and computational hardnessproblems.

The problem with the phylogenetic tree construction approach is thatthey cannot be used for the classification of 1.7 million described species on theearth. The existing methods cannot be used for constructing the tree of life dueto aforementioned shortcomings. However, given the limitations of current phylogenetic reconstructions methods, supertrees are presently a necessary tool for such phylogenetic problems.In fact, most, if not all, detailed estimates of the Treeof Life can only be done using phylogenetic supertree methods.

1.3 Problem statement

The aim of this thesis is to implement the construction of phylogenetic tree using triplet tree algorithms.

1. Implementation of Phylogenetic Supertree using Triplet (Inference and) Local Inconsistency.
2. Implementation of phylogenetic supertree using Triplet supertree heuristics.

1.4 Organization of the Report

The report comprises of five chapters including this chapter introduces the topic and states the problem. The rest of the dissertation report organised as follows

Chapter 2 gives an overview of basic trees, phylogenetic trees and supertrees and the process of phylogenetic study. Finally it gives a brief literature review of the related work.

Chapter 3 gives details about terminology of the triplets. It discusses about brief overview and implementation of the triplet inference and local inconsistency algorithm.

Chapter 4 gives details about hill climbing heuristics, operations that can be done on the trees. It discusses about approach and implementation of Algorithm.

Chapter 5 concludes the dissertation work and gives suggestions for future work.

Chapter 2

Background and Literature Review

The biological discipline dedicated to reconstructing organismal phylogenies is called phylogenetics. The term phylogenetics derives from the Greek, phylon means race or old family, and genos means birth or origin, gennetikos means related to generation or the genesis of something.The branching pattern of the tree represents the splitting of biological lineages, and the lengths of the branches can be used to signify the age of those events. Today, biologists call these treelike diagrams phylogenies. Reconstruction of these phylogenies can provide an evolutionary framework for studying a variety of problems led to their application in almost every other sub discipline of biology. Since evolutionary and this thesis is about phylogenetic trees, it is therefore appropriate to start by defining from a tree.

2.1 Basic Definitions

2.1.1 Trees

A treeconsists of internal nodes and leaf nodes, and they are connected by edgeswhich can have edge lengths. A tree can be a binary or non-binary tree. One node may be designated as a root node, inwhich case the tree is rooted otherwise it is unrooted.

A binary rooted tree can be treated asdirected acyclic connected graph having exactly one internal vertex of degree two and remaining internal vertices are having degree more than two. The leaves are vertices of degree one. In rooted trees there are directed edges which are directed away from the root.All nodes except the root have a parent, and all nodes except the leaf nodeshave a number of child nodes. The nodes on the path from a node x to the root are called ancestors of x. A descendant of x is any node which has x as an ancestor.

A binary unrooted tree is a tree with no special node with degree two and it can be treated as an acyclic connected graph having no internal vertices of degree two and leaves are vertices of degree one. An unrooted tree may be rooted by selecting any node as the root. The setof nodes directly connected to a node x by edges in any direction is calledthe adjacent of x.

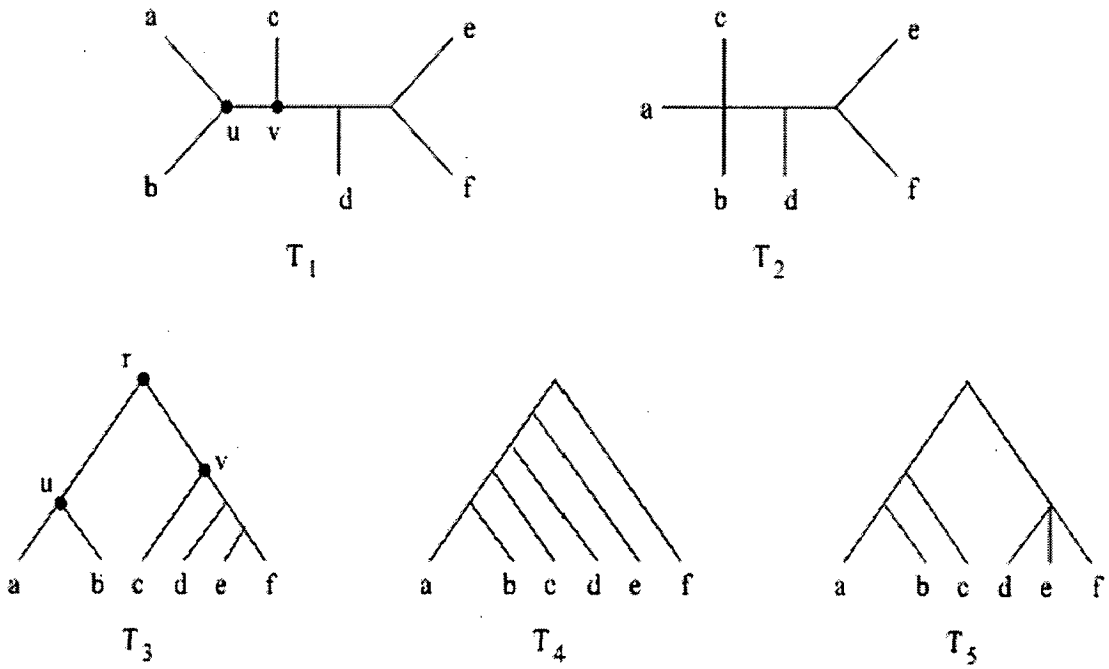

Figure 2.1: An example of rooted and unrooted trees

In Figure 2.1, T1 and T2 areunrooted, T3, T4, and T5 are rooted, T1, T3, and T4 are binary, and T2 and T5 arenon-binary. Notice that T1 can be obtained from T3 or T4 by suppressing the rootvertex of either of these rooted trees. Equivalently, T3 and T4 can be obtained fromT1 by subdividing an edge in T1 with a root vertex. For tree T3 internal nodes are {r, u, v}, leaves are {a, b, c, d, e, f}, and adjacent nodes of v are {r, c, d, e, f}.

In a fully resolved binary tree with n leaf nodes there are n-1 internal nodes. In Figure 2.1, T3 and T4 are fully resolved binary trees. The topology of a tree is the arrangement of the nodes and edges, ignoringedge lengths and symmetrical differences (i.e. it is arbitrary in which orderthe children of a node appear). If all leaf nodes are connected to a singleinternal node, then the tree is called a star tree because of its shape.

2.1.2 Subtrees

Let T be a rooted tree and choose a vertex v in T. If we remove the edge between v and the parent of v, say p, we get two connected subgraphs. Then let v be the root of the subgraphs containing v, then this is called the subtree of T rooted at v. Briefly a subtree T' is a tree whose

vertices and edges form the subsets of the vertices and edges of a given tree T. An example of a subtree is shown in Figure 2.3 in which tree T1 is a subtree of tree T.

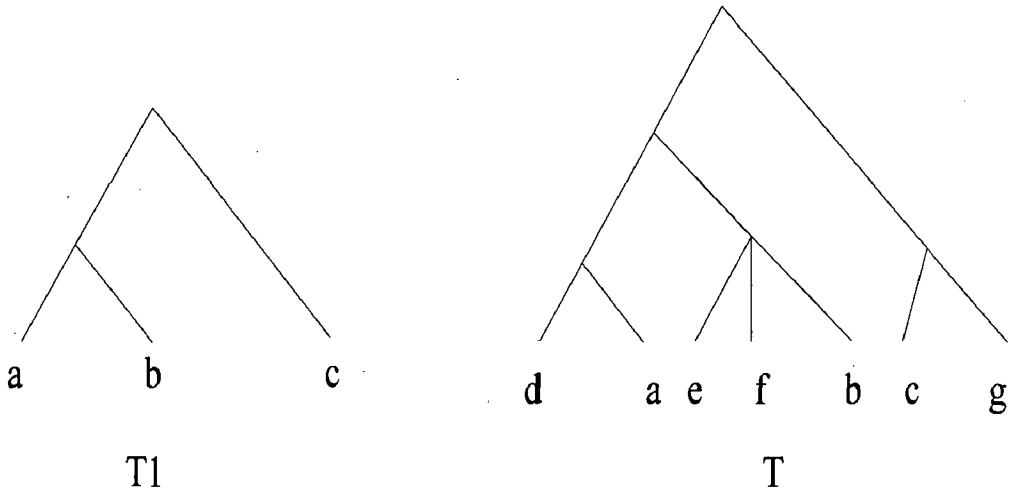

Figure 2.2: An Example of a subtree

2.1.3Clusters and Bipartitions

Each vertex in a rooted tree defines a group of taxa that are more closely relatedto each other than they are to any other taxon in the tree; such a group is called aclade. The clade defined by a vertex v is the set C(v) containing all the taxa thatcan be reached from the root via a path containing the vertex v.

In phylogeny, such a group is often called a cluster. The set of all clusters definedby some vertex in a tree T is denoted H(T); i.e, H(T) = {C(v) : v ∈ V (T)}.Notice that H(T) is a hierarchy, which means that for every two members C1,C2of H(T), either C1 ⊆ C2, C2 ⊆ C1, or C1 ∩ C2 = ∅. The trivial clusters, thosecontained in every tree with leaf set X, are X (the leaf set itself) and the singletonsets {x} where x ∈ X. The vertex v in T3 of Figure 2.1, for example, defines thecluster {c, d, e, f}, and the set of non-trivial clusters defined by the vertices of T3is {{a, b}, {e, f}, {d, e, f}, {c, d, e, f}}. We say that the rooted tree T'refines T ifH(T) ⊂ H(T'); this is denoted T ≤ T'.

The analogous relationships in unrooted trees are bipartitions of the taxonset, and are defined by edges rather than vertices. Each edge e in a phylogenetictree T, when deleted from T, creates two new subtrees T1 and T2. We say that theedge e induces the bipartition L(T1)|L(T2), equivalently

L(T2)|L(T1), of L(T); thisbipartition is denoted σe. The full set of induced bipartitions of a tree T is denotedΣ(T), i.e. $\Sigma(T) = \{\sigma_e : e \in E(T)\}$..

The trivial bipartitions, those contained inevery unrooted tree with leaf set X, are those involving a single taxon on one sideof the bipartition $\{x\}|X - \{x\}$ where $x \in X$. We often abuse notation, droppingthe brackets and commas when writing the sets on either side of the split. Thus, wesay that in Figure 2.1, the edge (u, v) in T1 induces the bipartition ab|cdef, and theset of non-trivial bipartitions of T1 is {ab|cdef, abc|def, abcd|ef}. Naturally, we saythat an unrooted tree T′refines T if $\Sigma(T) \subseteq \Sigma(T')$; again, this is denoted $T \leq T'$.Note that the definition of Σ(T) does not require that T is unrooted, thus Σ(T)is still defined for rooted trees. H(T), on the other hand, does not automaticallyextend to unrooted trees.

Bipartitions can be used to construct a single tree that summarizes the relationships in a collection of trees with identical leaf sets. The methods used to construct such summary trees are called consensus methods, and the trees producedconsensus trees; the most commonly used consensus methods are the strict consensus and majority consensus.

The strict consensus tree of a set of trees is thetree whose set of bipartitions are those that are present in all the trees in thegiven set. For a given set of trees {T1, T2, ..., Tk}, the strict consensus tree [15] Tis such that

$$\Sigma(T) = \{\sigma : \sigma \in \Sigma(T_i) \text{ for all } 1 \leq i \leq k\} = \bigcap_{i=1}^{k} \Sigma(T_i) \qquad (2.1)$$

The majority consensus tree of a set of trees is the tree whose bipartition set containsthose bipartitions present in more than half of the trees in the given set. For agiven set of trees {T1, T2, ..., Tk}, the majority consensus tree [16,17] T is such that

$$\Sigma(T) = \{\sigma : \text{the number of trees} T_i \text{such that } \sigma \in \Sigma(T_i) > \frac{k}{2} \qquad (2.2)$$

2.1.4 Triplet and Quartet trees

For every three leaves, a, b, c, there are four possible rooted treeswith leaf set {a, b, c}, see Figure 2.3. A rooted binary tree on three leaves is called atriplet. A triplet with leaf set {a, b, c} that contains the cluster {a, b} isdenoted ab|c (equivalently, ba|c). The set of triplets displayed by a rooted treeT is denoted r(T); formally, r(T) = {ab|c : a, b, c ∈ L(T) and {a, b} ∈ H(T|{a,b,c})}.

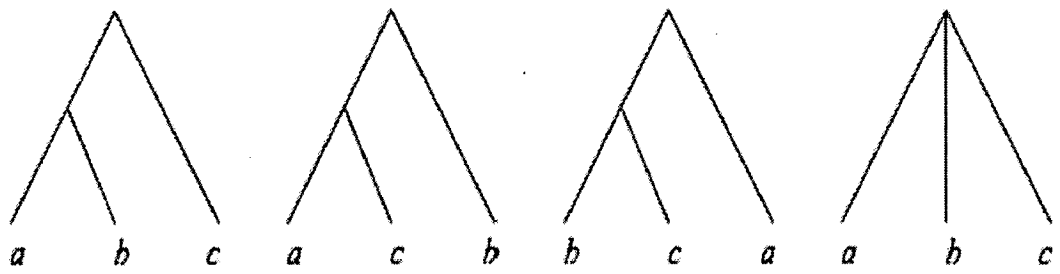

Figure 2.3: The four rooted trees with three leaves.

For every four leaves, a, b, c, d, there are four possible unrooted trees with leaf set {a, b, c, d}, Figure 2.4. An unrooted binary tree with four leavesis called a quartet tree.

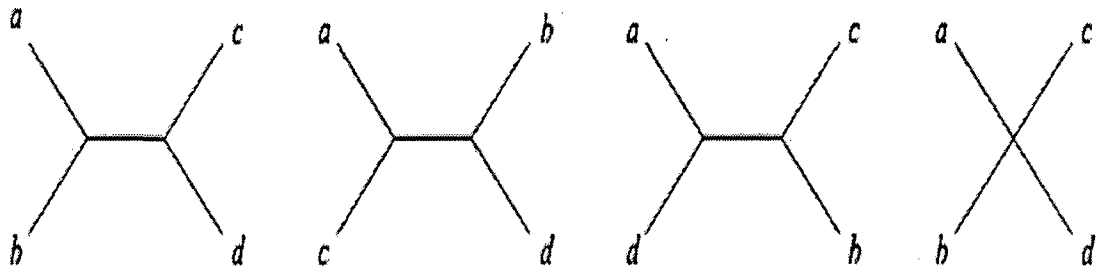

Figure 2.4: The four unrooted trees with four leaves.

A quartet tree with leaf set {a, b, c, d} and non-trivialbipartition ab|cd is denoted by that bipartition (equivalently, ba|cd, ab|dc, ba|dc, orcd|ab, etc.). The set of quartet trees displayed by an unrooted tree T is denotedq(T); formally, q(T) = {ab|cd : a, b, c, d ∈ L(T) and ab|cd = T|{a,b,c,d}}.

2.2 Phylogenetic Trees

In studying evolutionary histories, we will refer to particular entities of interest, bethey species, genes, molecular sequences, or languages, as taxa (singular taxon).This is a generalization of the use of the term taxa in biology where a taxon isany of the taxonomic categories such as phylum, order, or species in the Linnean system.

In general, the taxa have evolved via a process that can be represented by a rooted tree, a directed acyclic graph, in the graph theoretic sensewhose root represents the most recent common ancestor of all the taxa representedin the tree. At the leaves of the tree are extant taxa (or possibly extinct taxa forwhich we have fossil records), the internal vertices are ancestral taxa, and each edgerepresents an ancestor-descendant relationship with the ancestor being the taxonrepresented by the vertex closer to the root of the tree.

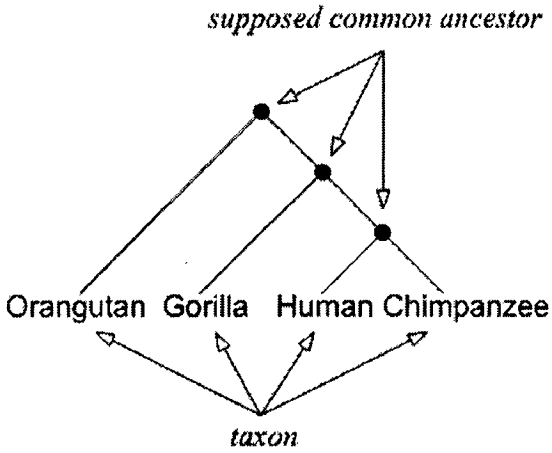

Figure 2.5: Rooted Phylogenetic Tree

Naturally there are times when the assumptions do not hold or are not used,but those are the exceptions. For instance, it might be possible to extractDNA from a specimen that has been preserved somehow, so that some ofthe taxa might be extinct.Let us take a rooted tree, as like one in Figure 2.5, as an example,systematically giving meaning to the various parts of it. Hopefully it willbe clear why rooted trees are ideal for displaying phylogenies, instead ofunrooted trees.

- **Root:**The root may be thought of as the (hypothetical) common ancestor of all the taxa.

- **Edges:**An edge corresponds to an ancestor/descendant relationship between the connected nodes: If a node d is a descendant of a, then dhas evolved from a somewhere along the line. The length of the edge can be made proportional to the time it has taken to evolve the child from the parent, or some other measure of evolutionary distance. Alternatively, if time is not important, all the edges can have the same length. Edges are directed from ancestor to descendant in a rooted tree, so they can be thought of as time lines.
- **Leaf Nodes:**Leaf nodes represent the existing taxa from which data is available.
- **Internal Nodes:**Each internal node i corresponds to a supposed evolutionary common ancestor of the taxa (leaf nodes) in the subtree rooted by i. Another way to look at it is that i represent a point where a population split up, creating new taxa, which in the tree are represented by the children of i. In any case, the internal nodes are almost always inferred from the biological data of the existing taxa, so we have no real guarantee that such common ancestors ever actually existed in the past.

2.3 Phylogenetic supertrees

When researchers publish phylogenetic trees, they are usually the result ofa narrowly focused study of some limited taxonomic group. For instance,in several studies of the mammalian order Carnivora, the published phylogenies were rarely larger than 30 taxa. It is simply not viable for a single research team to create large trees.This creates a demand forsupertree methods which make it possible to combine the efforts of differentteams to create larger phylogenetic trees.

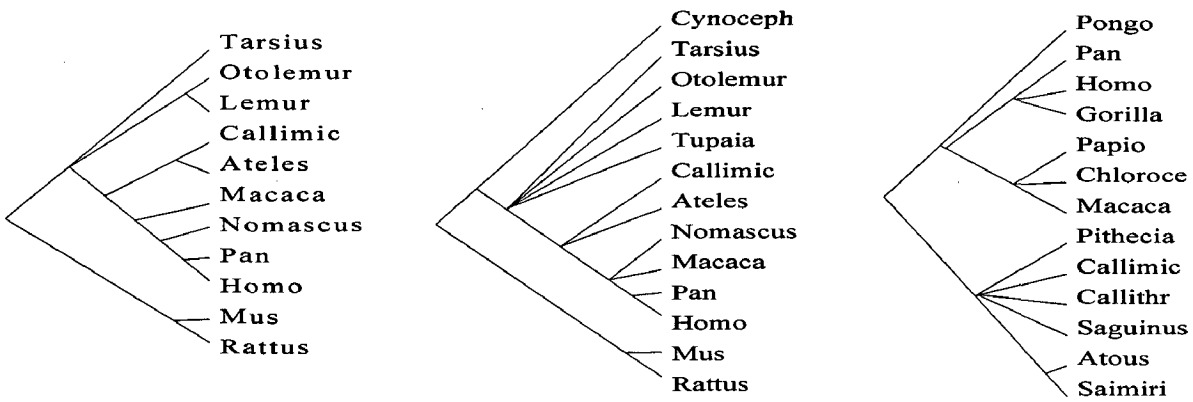

Figure 2.6: An example of input source trees.

There is no need for the teams to use the same methodologies whencreating their trees, just as there is no need for them to work on them at thesame time. In fact, phylogenetic trees that were published a long time ago,before current supertree methods were invented, can still be used as if theyhad been published for that very purpose today. Supertrees independenceof the method used to create any given phylogenetic tree is a great strength.

Supertrees are themselves phylogenetic trees, but they are built by combining a set of smaller phylogenetic trees. Such a set is referred to as a forest(see Figure 2.6), and the trees in the forest are called source trees.

Regularphylogenetic trees are based on biological data, and this result in the needfor a distinction between primary data, which comes directly from the taxa,and secondary data, which is derived from primary data. The source treesmust have partially overlapping taxa, or it makes no sense to combine themin a non-trivial fashion. If the source trees are the set T = {T1, . . . , Tk} thenthe leaf set of the supertree S is union of all leafs.

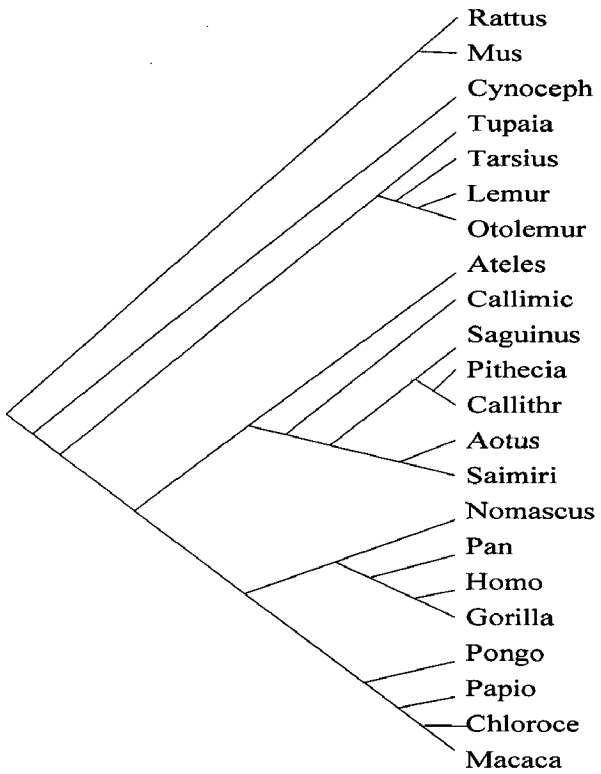

Figure 2.7: A possible supertree for the set of input trees in Figure 2.6.

2.4The Process of a Phylogenetic Study

When conducting a molecular phylogenetic analysis, biologist's first determine thescope of their study, choosing which phylogenetic group they will study, which particular taxa they will include in the study, and what genetic marker or markers theywill sequence to conduct their study. (Some studies may, instead, use morphologicalcharacters of the taxa, such as physical characteristics or biological functions, but wewill focus on studies that use molecular data.) The limitations that govern whichtaxa are included in the study include availability of tissue, location of specimens,time, funding, etc., and the ability to amplify and sequence.

Phase 1: Data collection

Biologists go out into the field and collect specimens (organisms or tissues) of thetaxa they have decided to study and make use of museum and herbarium specimens.The goal of the data collection phase is to assemble, for each taxon, a homologoussequence or sequences of characters that will be used to construct a phylogenetic treeon the set of taxa they have chosen to study. For studies using molecular characters,these data are molecular sequences for the particular marker they have decided touse in the study. To obtain these sequences, they extract DNA (or RNA) from theirspecimens and sequence the desired genetic marker. Following sequence assembly,the remainder of the phylogenetic analysis is primarily computational.

Phase 2: Sequence Alignment

An alignment is constructed taking the putatively homologous sequences and puttingthem in a matrix such that the sequence from each individual taxon occupies itsown row, and each column of nucleotides have ideally evolved from a common ancestral nucleotide. This object property of the columns is called positional homology.Positional homology is estimated by inserting gaps into each sequence in the appropriate places, and padding the beginning and end of each sequence with placeholders that represent unknown nucleotides, so as to obtain sequences having the same length. For most phylogenetic reconstruction methods, positional homology isof critical importance because each column of the alignment is assumed to representthe independent evolutionary history of the site. An incorrect alignment will havetwo or more sites that do not reflect positional homology. Therefore, at least some ofthe data used to reconstruct the phylogeny will suggest a false evolutionary historyof the sequences. To obtain an alignment,

most researchers use some software, suchas ClustalW [18], but then often adjust that alignment by eyeto create the final alignment.

Phase 3:TreeInference

A phylogenetic reconstruction method, such as maximum parsimony, maximum likelihood, Bayesian analysis, or a distance-based method [19], is applied to the aligned sequences. Here we briefly describe maximumparsimony and maximum likelihood:

Maximum Parsimony is an optimization problem based on the minimum evolution principal. A maximum parsimony tree is one that minimizesthe number of changes over the edges of the tree needed to "explain" thesequences at its leaves. We formalize this goal here by first defining the parsimony score of a tree.

Given a tree T that is leaf-labeled by a set of sequences S each of length l, anextension f of S on T is a labeling of all vertices of T by sequences of lengthl that maintains the original leaf-labeling by S. For two sequences x and y, letH(x, y) be the Hamming distance between x and y. (For two sequences $x = x_1x_2...x_l$ and $y = y_1y_2...y_l$, $H(x, y) = |\{i : x_i = y_i\}|$.) Then the parsimonyscore of an extension f, denoted score(f, T), is $\Sigma_{(u,v)\in E(T)}H(f(u), f(v))$. Aminimal extension is one that minimizes the parsimony score, and the parsimony score of a minimal extension is the parsimony length of the tree T.Computing the parsimony length of a fixed leaf-labeled tree is O(nrl), wheren is the number of sequences, l is the length of the sequences in S, and r isthe number of states observed in S. In the caseof DNA, r = 4.

A maximum parsimony tree for a given set of sequences S is a tree that has thesmallest possible parsimony length of any tree leaf labeled by S. While computing the score of a single tree can be done in linear time, finding a minimumparsimony tree is equivalent to the Hamming-distance Steiner tree problemwhich is known to be NP-hard [20]. An exhaustivesearch for a maximum parsimony tree on n sequences would require lookingat all $(2n - 5)(2n - 7) \ldots 1$ possible topologies of trees with the given leaf-labeling. Branch-and-bound algorithms can reduce the search space, but donot reduce the asymptotic running time. Using branch-and-bound algorithmsresearchers can compute all maximum parsimony trees for datasets of up to17about 25 taxa before the running time becomes prohibitive. Thus, for largerdatasets, heuristic searches are performed whereby the tree-space is exploreduntil a local optimum is found.

Maximum Likelihoodis also an optimization problem, and seeks themost probable tree according to some specified model of evolution. Given amodel of sequence evolution, a maximum likelihood tree is a tree topology,along with a set of branch lengths and parameters for the given model of evolution, that maximize the conditional probability of observing the sequencesat the leaves given the assumed model. Again let S be a set of sequencesof equal length. The maximum likelihood solution for the set S is an edge-weighted tree (T,w), together with a set of model parameters M, that minimizes Pr(S|T,w,M). As with MP, ML is NP-hard [21],and heuristic searches are used to compute ML trees.

Phase 4: Post-processing

In the post-processing stage, scientists summarize the results obtained in the treeinference stage. This typically involves computing a consensus tree for tree inferencemethods, such as MP, that return multiple trees, and assessing the supportfor branches in the inferred tree(s).

2.5 Literature Review

UPGMA is asequential clustering algorithm, and perhaps the simplest method oftree construction. It constructs a rooted tree from a distance matrix M. First it creates a pool of clusters, each containing a singletaxon. Then it combines the two closest clusters in a new cluster C.M is updated so that the distance from C to another cluster X is thearithmetic mean of the distances from the taxa in C to X. This isrepeated until there is only one cluster left, at which point the treecan be constructed: Each cluster C corresponds to a node n, and thetwo clusters which comprise C correspond to the children of n. Thealgorithm is criticized for only constructing trees correctly if certainproperties are true for M, namely that M is ultra-metric.

Neighbour Joining method [22] addresses the main shortcoming of UPGMA by relaxing the requirements for the distance matrix, but it is a linear factorslower and produces unrooted trees. It starts by placing all taxa in astar tree. A modified distance matrix is maintained in which the distance between each pair of nodes is adjusted according to the averagedistance from all other nodes. In each iteration, the two least distantnodes are linked and removed from the tree, while their common ancestor is added to the tree. The tree is reduced in this way until onlytwo nodes remain, connected by an edge. All the taxa are at this pointlinked under those nodes, and the tree has been built.

Maximum Parsimony works on the principle that simpler solutions arepreferred over more complex ones. For a phylogeny, this means thata tree which can explain the input character data with fewer evolutionary events is preferable over a tree which contains more events.By minimizing the number of character state changes along all pathsin the tree, the most parsimonious tree is obtained. Different matrixencoding schemes exist. The original, due to [23, 24], uses 1 to indicate that the trait is present and 0 to indicatethat it is not. One of the disadvantages of the method, apart fromits NP-hard time complexity, is that no branch lengths are computed:Only the order in which the evolutionary splits occur.

Maximum Likelihood takes as input a model of sequence evolution, atree providing a topology and branch lengths, and a data matrix. Itthen computes the likelihood of the tree giving rise to the data matrix,given the specified model of sequence evolution. By searching throughthe tree space, the trees with the maximum likelihood of being correctcan be found. The result depends on the model, but the method isstatistically well founded and robust to violations of the rules presentin the evolutionary model.

The above methods are used to create phylogenetic trees from primary data, which comes directly from the taxa whereas the following section is about supertree methods which will base on secondary data discussed in section 2.3. Supertree methods summarize the collection of trees without losing branch information.There are different approaches to solve this problem. Some methods solve it using graph theoretic, others are subtree based and yet others are recoding based approach.

Thus supertree methods need to solve the compatibility problem that is to decide if theinput trees are compatible. While the compatibility problem can be solved for rooted inputtrees in polynomial time using the Build algorithm from Aho et al. [25], the problem isNP-complete for unrooted input trees. Thus, the time complexity of supertreeproblems for rooted and unrooted trees are clearly different. An even stronger separationbetween supertree methods for rooted and unrooted input trees was made by Steel etal. [26]. Steel et al. showed that in contrast to supertree methods for rooted input trees, nosupertree method for unrooted input trees exists that satisfies a subset of the fundamentalsupertree properties. From this result Semple and Steel [26] conclude that there exists no'reasonable' supertree method for unrooted input trees.

For any compatible set of triplets and the set of leaf nodes Build algorithm [25] can be used to find the local consensus tree. A fundamental question in biological classification is how a supertree method should as semple input trees to optimally represent their branching information. Probably the oldestsupertree algorithm for rooted input trees is the Build algorithm, introduced by Aho etal. [25], and that can be used to solve the compatibility problem in polynomial time.While other supertree problems are typically NP-complete, there are several modificationsof the Build algorithm that still run in polynomial time. Semple and Steel [26] introduceda modified version of the Build algorithm, the MinCut Supertree algorithm.

The Build algorithm constructs a rooted tree that is consistent with a given set of lineageconstrains. If no such tree exists the empty tree is returned. Given nlineage constrains the Build algorithm runs in $O(n^2)$ time[25].Toconstructaparenttreefor a given profile, the Build algorithm is executed on the set of all lineage constrains thatare consistent with a tree in the profile.

Mincut Algorithmis a subtree based algorithm and has all the advantages of using triplets and quartets. Mincut [26] method is one of the very few methods that satisfyall the desirable properties of supertree. This method is an extension of the Build. Build method reports an error when the input trees are incompatible. Semple modified the Bduild algorithm in such a way that if theinput trees are incompatible then the some of the edges are removed from thetrees based on the minimum loss criteria. The minimum cost is identified usingthe edges which have minimum weights and when removed will result in a disconnected modified graphs. Therefore the algorithm results in a single tree evenif the input trees are not compatible.

The mincut algorithm of [26] was the first to cope withthis problem of inconsistent input triplets. Their algorithmhandles rooted subtrees of arbitrary size, however, it alsospecializes in rooted triplets. Page [27] extended this idea tomaintain all subtrees that are not in disagreement with anyinput subtree. The algorithm, denoted modified min cut, applies the min cut criterion but on a somewhatdifferent graph than that of [26].

Modified Mincut method is is also a subtree based method. Mincut method results in highly unresolved trees when applied to the input tree representing polytomy. Page [27] modified the Mincut algorithm to work for a general case. The basic idea wasto divide the edges of the

modified graph into three categories: unanimous, uncontradicted and contradicted. Whenever the modified graph is fully connected,instead of removing the edges from the minimal cut set of the graph, the contradicted edges are removed. As it is an extension of the Mincut method, it alsosatisfies the properties that Mincut satisfies.

Indirect supertree construction uses some form of matrix representation(Ponstein 1966, Ragan 1992) to encode individual source tree topologies as matrices that are then combined and analyzed using an optimization criterion. This is the most widely used phylogenetic supertree method defined by Baum[23] and Ragan [24] independently. MRP represented a universally applicablemethod that could combine the even incompatible set of input trees. In thismethod binary coding of the components of each input tree is used to generatea pseudo character matrix representation of the trees.The character matrices of all the trees are combined, the leaves that arenot present in a given tree are marked as '?', i.e, missing. The final matrix is then analyzed with parsimony procedure [28] to produce one or moreparsimonious trees.

Supertree methods developed by Chen et al. [29] and Eulenstein [30] aremotivated by the notion of error correction. In MRP taxa present in the samecluster are scored as 1, those are absent are scored as 0, and those which arenot sampled are marked '?'. One notion of error in suchcluster system is the presence of an incorrect label in a cluster or the absence ofone that should be present. This type of errors is called flips $0 \rightarrow 1$ or $1 \rightarrow 0$.This leads to the optimization problem of finding the minimum number of flipsthat converts the matrix into a matrix which is consistent with a phylogenetictree. This is an NP-hard problem and [29, 30] gave approximation algorithm forit.Another approach to the matrix representation is to employ the distancesbetween the taxa of each tree and finally combine them into a single averagematrix, called as average consensus method [29]. Any distance based tree construction method can then be used construct the supertree.

Chapter 3

Phylogenetic Supertree Construction using Triplet (Inference and) Local Inconsistency

3.1 Triplet Terminology

For every three leaves let us name, a, b, c, there are four possible rooted phylogenetic treeswith leaf set {a, b, c}, see Figure 3.1. A rooted binary tree on three leaves is called arooted triple. A rooted triple with leaf set {a, b, c} that contains the cluster {a, b} isdenoted ab|c (equivalently, ba|c). The set of rooted triples displayed by a rooted tree T is denoted r(T).

Formally,r(T) = {ab|c : a, b, c ∈ L(T) and {a, b} ∈ H(T|{a,b,c})}.

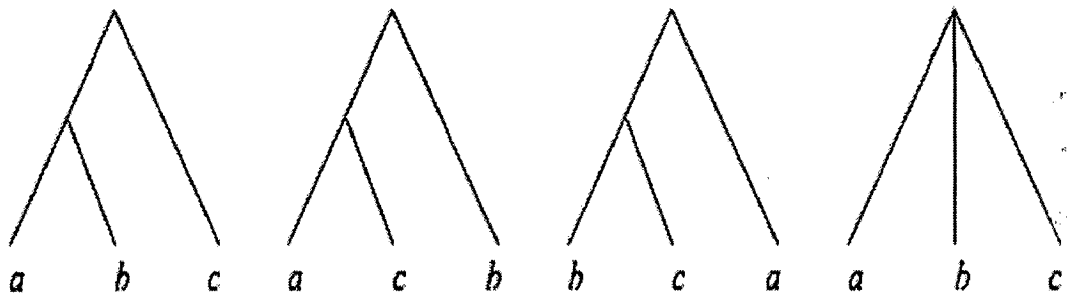

Figure 3.1: Triplet with four non trivial topologies.

In Figure 3.1 for the triplet {a, b, c}, and such a tree is called a triplet tree.A triplet tree is defined uniquely by its ancestral relationships, giving thenotation a|bc, b|ac, and c|ab for the triplet trees displayed in Figure 3.1 fromleft to right (we ignore the non-binary trifork tree which is unresolved). The lowest common ancestor oftwo leaf nodes a and b is the node c for which c is an ancestor of a and b,and where no other node with that property is a descendant of c. We writec = lca(a, b), and for a triplet tree a|bc we have that lca(b, c) is a descendantof lca(a, b) and lca(a, c).

Triplet tree space is defines as

$$(L) = U_{M \subseteq L, |M|=3} \{t : t \text{ is a triplet tree on } M\} \text{on a leaf set L.}$$

The weighting function w: $T^*(L) \rightarrow R+$,and werefer to the triplet trees on a tripletM as t_1^M, t_2^M, and t_3^M such that$w(t_1^M) \leq w(t_2^M) \leq w(t_3^M)$.

3.2 Triplet(Inference and)Local Inconsistency Algorithm[31]

The algorithm generates a supertree from input forest of trees by using three steps. In the first step it will decompose the given forest input into set of triplets, second step is an optional one which will be used to infer the missed triplets from the existing triplets and, third will generate a supertree by adding one leaf at a time to the final tree.

```
Algorithm : Decompose [ 31]
Input : forest with leaf set L
Output: a weighting function w : T*(L) → R+
Foreach tree in forest do
        root ← root(tree);
        DecorateDown(root);
        DecorateUp(root);
        CountTriplets(root);
end
foreach triplet tree t ∈ T*(L) do
if frequency(t) > 0 then
set w(t) = −log frequency(t);
else
set w(t) = ∞;
end
```

Figure 3.2 Algorithm for Decomposing Forest into Triplets

As said earlier the tree isn't actually reduced to a set of triplet trees, rather it constructs a weighting function $w : T^*(L) \rightarrow R+$ for all the triplet trees in the triplet tree space of the leaves of the inputforest. We have to count the occurrences of each triplet tree in all the input trees inorder to determine their frequencies and thus their weights, which will beused to create the weighting function. The decompose algorithm iterates throughthe trees of the forest, first decorating them, and then counting the triplet treesin the decorated trees. The algorithms used for each task are all recursive,and their starting point is always the root of the current tree. Once all thetrees have had their displayed triplet trees counted, the weighting function isconstructed in the above algorithm.

DecorateDown is a recursive algorithm which the first decoration step and it must be carried out before proceeding, since its results are required for the next decoration step. Each internal node is decorated with a set D which is to contain the leaf set ofthe subtree rooted by the node. We shall let Dx denote the D set belongingto the node x. For the input node, first D is initialized with the empty set.Then all children that are leaf nodes are added to D, and the D sets of allthe other children, which are internal nodes, are computed recursively andadded. Thus, D contains the leaf nodes directly under the input node, aswell as the leaf nodes of the subtrees rooted by its children.

For a nodeit simply subtracts the leaf nodes in Dnode from the leaf set of the entiretree, and assigns the result to the set Unode. Then it recuses on those ofthe children that are internal nodes. The purpose of Unode is to contain allthe leaf nodes which can be reached from the node by traveling at least oneedge up, and then an arbitrary path down.

It is possible to create a mapping from the triplets that are resolved ina tree to the tree's internal nodes:

$$\psi: \{M: M \subseteq \text{leaves(tree)}, |M| = 3, \text{tree}|M \text{ resolved}\} \rightarrow \text{internals(tree)} \quad (4.1)$$

The mapping is defined as follows: For the triplet {a, b, c}, find the lowestcommon ancestors lca(a, b), lca(a, c), and lca(b, c), which are all internal nodes. Triplet {a, b, c} is mapped to the node which is thedescendant of the others. The node exists because the triplet is resolved otherwise all the lowest common ancestors would be the same.

The mapping tells us that we can count all triplets resolved in a treeby counting the triplets mapped to each internal node, and adding up theresults. Since every triplet is mapped to one node, and one node only, notriplet is counted twice, but all triplets are counted.

CountTriplet algorithm receives as input a node. The tree containing node is repeatedly divided into three zones with node in the center.One division corresponds to one pair of node's children (so if the tree is binary, there is only one possible division). The zones correspond to the partsof the tree that contain Dx, Dy, and Unode for the pair x, y $\in$ children(node).Unode contains the

candidate nodes for the ancestor leaf nodes in the associated triplet trees, while Dx and Dy contain the candidate nodes for theother two leaf nodes.

By picking three leaf nodes, one from each zone, we obtain a triplet.We decide which triplet tree is displayed for the triplet and increment itscount. This is repeated for every combination of leaf nodes from the threezones. When all triplet trees have been counted, the algorithm recurses onthe children of node that are internal nodes, and thus the whole trees iscovered eventually. The time complexity for the above entire procedure is O (n^3).

```
Algorithm : Inference [31]
Input : a weighting function w : (L) → R+ for a leaf set L
Output: a modified weighting function w : T*(L) → R+
Data: sets S, R with supported and rejected triplet trees
(S,R) ← SortTriplets(w);
Foreach triplet M ⊆ L do
Foreach triplet tree t ∈T*(M) do
Foreach quartet tree q that displays t do
For each triplet tree p = t displayed in q do
        if p ∈ R then
        demerit t;
        end
        if d ∈ S for some d ∈T*(leaves(p)) \ {p} then
        demerit t;
        end
      end
    end
  end
if a triplet tree t ∈ T*(M) has fewer demerits than the othersthen
S ← S \ T*(M);
R ← R \ T*(M);
S ← S ∪ {t};
end
end
w ← AdjustWeights(S,R);
```

Figure 3.3 Algorithm for inferring missed triplets

SortTriplets function will sort the triplet trees in T^*(L) into a set of supported and rejected triplet trees called S and R, respectively.Once the triplet trees are sorted, the inference algorithm First, each triplet tree is assigneda demerit counter, which is initially set to zero. Then, the outermost loop iterates through all the triplets M ⊆ L. In the loop, each triplet tree t for M is examined in

turn to determine its demerits. We can expand t to a quartet tree by adding a leaf node fromL \M. When expanding a triplet tree,we have four edges where an expansion can take place shown in figure 3.4. A resulting quartet tree displays in 4 Triplets shown in figure 3.5.

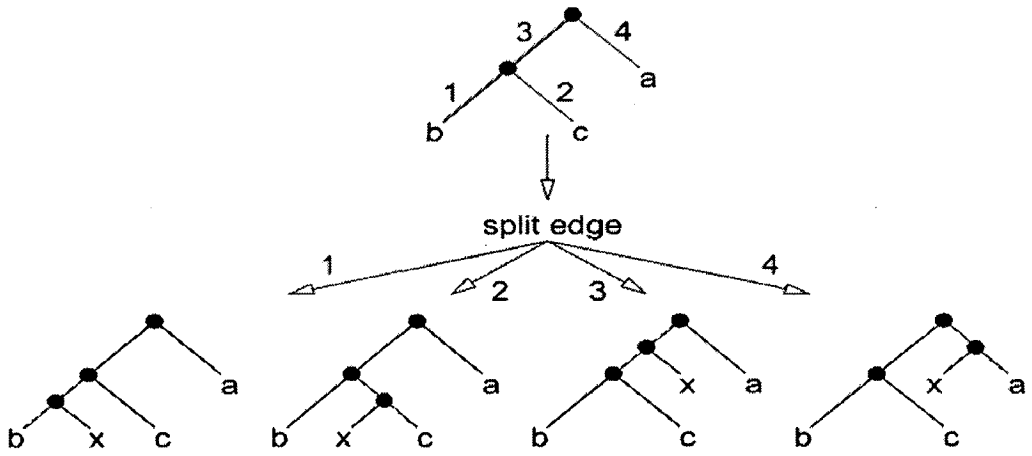

Figure 3.4: An extension of Triplet to a quartet by adding a leaf.

In this case the leaf x is added in each possiblelocation so there are four possible quartets possible.

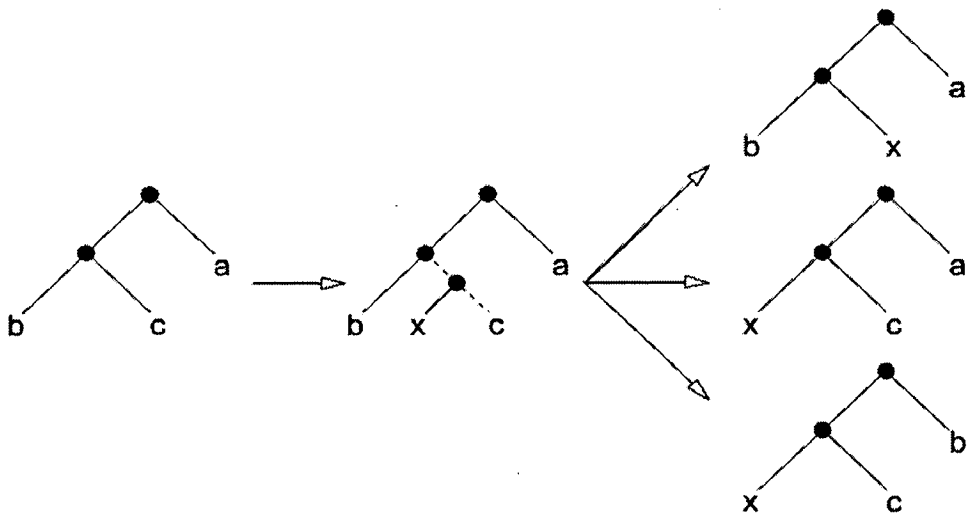

Figure 3.5: A triplet that has been expanded displays three other triplets.

Here a|bc has been expandedby splitting the edge leading to c, and the resulting triplet tree displays theadditional triplet trees a|bx, a|xc, and b|xc.The time complexity of the inference step for triplets is $O(n^4)$.

```
Algorithm : Compose [31]
Input : a weighting function w : T*(L) → R+ on the leaf set L
Output: a supertree
T ← t1^M such that t1^M maximizes w(t2^M) − w(t1^M) over all M ⊆ L;
L ← L \ leaves(T);
while L = ∅ do
foreach leaf ∈ L do
foreach tree expanded from T by adding leaf do
if LIw(tree, leaf) is the lowest so far then
T2^leaf ← T1^leaf;
T1^leaf ← tree;
            end
        end
    end
   choose leaf s.t. LIw(T2^leaf, leaf) − LIw(T1^leaf, leaf) is maximized
and secondarily s.t. LIw(T1^leaf, leaf) is minimized;
T ← T1^leaf;
   L ← L \ {leaf};
end
```

Figure 3.6Algorithm for composing triplets to a supertree

To determine topology we can use two different approaches, one using properties of lowest common ancestors, and oneusing a distance matrix. The difference is in their complexities.

Lowest Common Ancestor (LCA)

Attempt at an algorithm for triplets used the property shown in above, namely that the topologycan be detected by comparing lowest common ancestors shown in figure 3.6. First, lca(a, b),lca(a, c), and lca(b, c) are computed. Then we have

$$T|\{a,b,c\} = \begin{cases} a|bc & if\ lca(a,b) = lca(a,c) \\ b|ac & if (lca(a,b) = lca(b,c) \\ c|ab & if\ lca(a,c) = lca(b,c) \\ abc & lca(a,b) = lca(a,c) = lca(b,c) \end{cases}$$

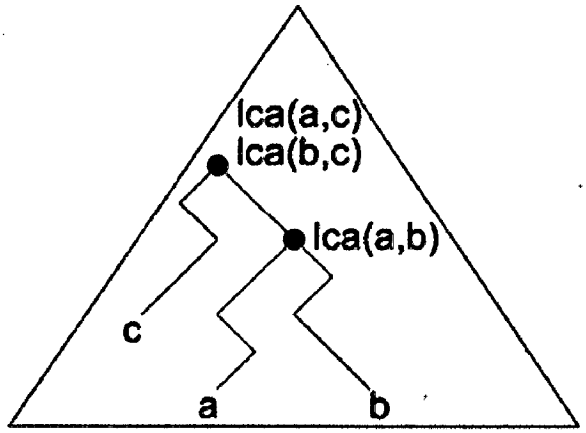

Figure 3.7 the connecting paths between the leaf nodes of a resolved triplet in a tree.

Distance Matrix (DM)

Distance matrix was based on the idea thatthe root r could be added to a triplet as a fourth leaf node. Calculate the path lengths Da|bc = |ra|+|bc|,Db|ac = |rb|+|ac|,and Dc|ab = |rc| + |ab| for each possible topology, and pick the topologywhich has the shortest one.

we must define excess, inconsistency, andlocal inconsistency in the following way for a tree T, a triplet M, and aweighting function w : $T^*(L) \rightarrow R+$:

$$excess_w(T, M) = \begin{cases} w(T|M) - w(t_2^M) \; if \; T|M = t_1^M \\ w(T|M) - w(t_1^M) \; otherwise \end{cases} \quad (4.2)$$

$$I_w(T) = \max \{ excess\,(T, M) : M \subseteq leaves(T), |M| = 3\} \quad (4.3)$$

$LI_w(T, x) = \max \{excess(T,M) : M \subseteq leaves(T), |M| = 3, x \in M\}$.

The time complexity for the above algorithm is $O(n^5)$.

M.G. CENTRAL LIBRARY
ACC No.
Date
I.I.T. ROORKEE

4.3 Implementation and Results

The algorithm Triplet Inference and Inconsistency is implemented in java using MyEclispse IDE. All executions done on a computer with the following specifications

Hardware specifications:

4 GB RAM

Core2Duo processor

Software specifications:

MyEclipse software.

Results:

The running times are shown in Table 3.1 for TLI using LCA and DM topology detection and TILI using DM topology detection. The executions are on a different forest of the same size. There is an entry for each forest size from 20 to 60 taxa, in increments of ten.

Taxa	TLI(DM)	TLI(LCA)	TILI
20	0.48	0.49	0.78
30	1.79	2.10	3.39
40	6.87	8.50	12.34
50	22.56	29.82	36.62
60	56.27	77.90	85.72

Table 3.1 Time in seconds required to build supertrees from forests of varying sizes.

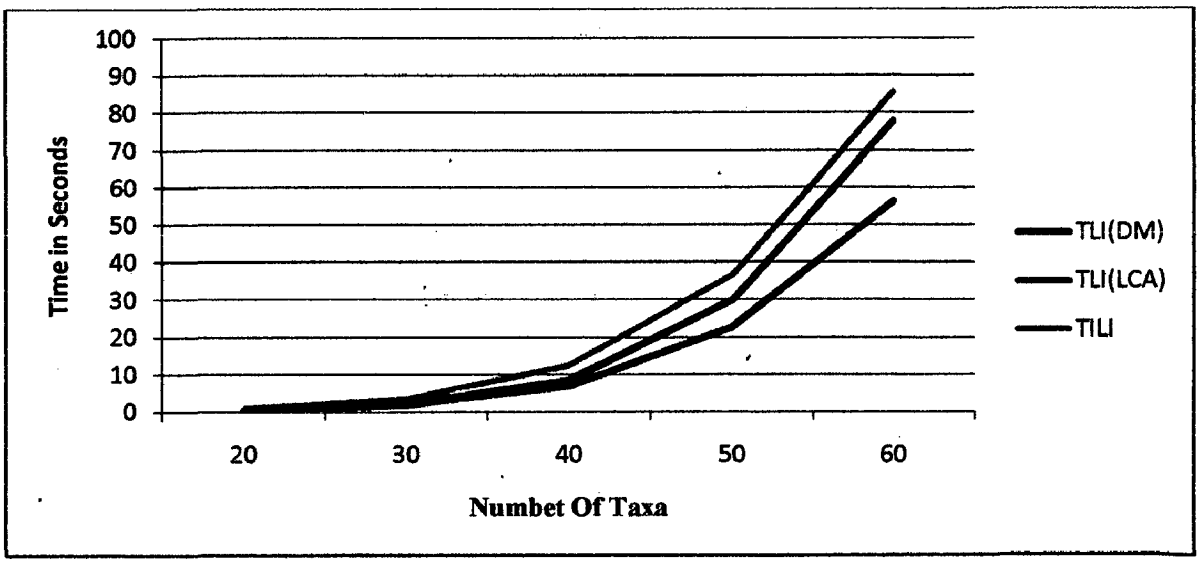

Figure 3.3 Runtime comparison between TLI using both DM and LCA and TILI

From the above we can understand that TLI using DM outperforms the executions times than TLI and TILI. If we use the LCA for computing topology the triplet is taking more time in the excess calculation shown in eq.4.2. Since using LCA for topology detection adds some more time computation for the excess calculation. But the time complexity for computing the topology of a triplet is constant using DM therefore the excess calculation can be done in constant time.For the TILI even using DM it take more time since it involves in computation of the inferring the missing triplets.

Chapter 4

Heuristic approach for Phylogenetic Supertree

Hill-climbing heuristics have been successfully applied to several intrinsically complex supertree problems [23,32, 33]. They search the space of all possiblesupertrees guided by a series of exact solutions toinstances of a local search problem. The local search problem is to find an optimal phylogenetic tree that attains the optimal criteria in the neighborhood of a given tree. The neighborhood is the set of allphylogenetic trees into which the given tree can be transformed by applying a tree edit operation explained in section 4.2.

In these heuristics a tree graph is definedfor the given set of input trees and some, typically symmetric, tree-edit operation. The nodes in the tree graph arethe phylogenetic trees over the overall taxon set of theinput trees. An edge adjoins two nodes exactly if the corresponding trees can be transformed into each other bythe tree edit operation. The cost of a node in the graph isthe measurement from the input trees to the tree represented by the node under the particular supertree problems optimization measurement. For the triplet supertreeproblem defined in later paragraph, the cost of a node in the graph is the triplet-similarity from the input trees to the tree represented by thenode. Given a starting node in the tree graph, the heuristic's task is to find a maximal-length path of steepestascent in the cost of its nodes and to return the last nodeon such a path. This path is found by solving the localsearch problem for every node along the path. The localsearch problem is to find a node with the maximum costin the neighborhood (all adjacent nodes) of a given node.

Every rooted tree can be equivalently represented by a set of triplet trees [34]. By using this fact we can solve triplet supertree problem i.e. finding a supertree for a given an input profile of n species trees (T1,..., Tn) such that it maximizes the triplet-similarity score which is defined in eq.4.1.

A triplet-similarity measure can be defined between two rooted trees that is the cardinality of the intersection of their triplet presentations. This measure can be extended to measure the similarity from a collection of rooted input trees to a rooted supertree, by summing up the triplet

similarities for each input tree and the supertree.Triplet similarity score between a given profile of trees P = (T1,...,Tn) and a supertree T* of P can be defined as

$$S(P,T^*) = \sum_{i=1}^{n} |S(T_i,T^*)| \qquad (4.1)$$

The triplet supertree problem is that for a given profile P, find a supertree T* that maximizes S(P, T*). We call any such T* a triplet supertree.

4.1 Preliminaries

Let T be a rooted tree. We define $\leq_T$ to be the partial order on V(T) where $x \leq_T y$ if y is a node on the path between Ro(T) and x. If $x \leq_T y$ we call x a descendant of y, and y an ancestor of x. We also define $x <_T y$ if $x \leq_T y$ and $x \neq y$, in this case we call x a proper descendant of y, and y is a proper ancestor of x.

The set of minima under $\leq_T$ is denoted by Le(T) and its elements are called leaves. If $\{x, y\} \in E(T)$ and $x \leq_T y$ then we call y the parent of x denoted by $Pa_T(x)$ and we call x a child of y. The set of all children of y is denoted by $Ch_T(y)$. If two nodes in T have the same parent, they are called siblings. The least common ancestor of a non-empty subset $L \subseteq V(T)$, denoted as $lca_T(L)$, is the unique smallest upper bound of L under $\leq_T$.

If $e \in E(T)$, we define T/e to be the tree obtained from T by identifying the ends of e and then deleting e. T/e is said to be obtained from T by contracting e. If v is a vertex of T with degree one or two, and e is an edge incident with v, the tree T/e is said to be obtained from T by suppressing v. The restricted subtree of T induced by a non-empty subset $L \subseteq V(T)$, denoted as T|L, is the tree induced by L where all internal nodes with degree two are suppressed, with the exception of the root node. The subtree of T rooted at node $y \in V(T)$, denoted as Ty, is the restricted subtree induced by $\{x \in V(T): x \leq_T y\}$.

Let T be a tree and $v \in V(T)$, an immediate triplet induced by v, denoted as $yz \blacktriangleright \triangleleft v$, is a triplet yz | v where there exists nodes a, $b \in V(T)$ such that $Pa_T(y) = Pa_T(z) = b$ and $Pa_T(b) = Pa_T(v) = a$.

4.2 Tree Edit Operations

A variety of different tree edit operations have been proposed [35],and two of them, rooted Subtree Pruning and Regrafting (SPR) and Tree Bisection and Reconnection (TBR), have shown much promise for phylogenetic studies.

4.2.1 ReRoot operation

Let T be a tree and x ∈ V (T). RRT(x) is defined to be the tree T if x = Ro(T). Otherwise, RRT(x) is the tree that is obtained from T by (i) suppressing Ro(T), and (ii) subdividing the edge $\{Pa_T(x), x\}$ by a new root node.

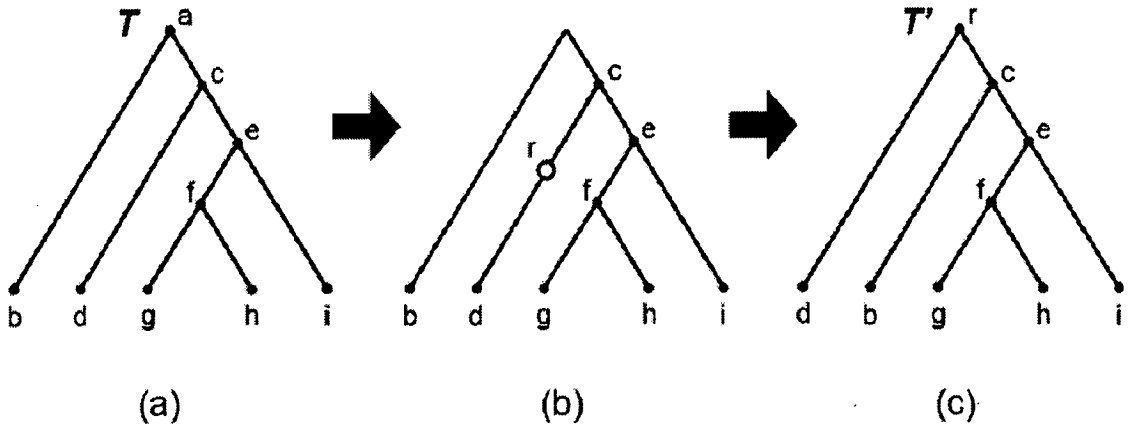

Figure 4.1 Reroot operation

The original tree T is shown in (a). In (b), we first suppress the root node, and then introduce the new root node r above d. Finally we rearrange the tree so that r is at root, as in (c).

The extension for the RR operation can be defined as $RR_T = \cup x \in V(T)\{RR_T(x)\}$.

Let $x \leq_T v$, we also define a partial RR operation RRT(v, x) by replacing Tv with (x). $RR_T v$

4.2.2 Tree Bisection and Reconnection

The planted tree for a tree T is denoted by Pl(T) and obtained by adding an additional edge, called root edge, {r, Ro(T)} to E(T).

Let T be a tree, e = (u, v) ∈ E(T), and X, Y be the connected components that are obtained by removing edge e from T where v ∈ X and u ∈ Y. We define $TBR_T(v, x, y)$ for x ∈ X and y ∈ Y

to be the tree that is obtained from Pl(T) by first removing edge e, then replacing the component X by $RR_X(x)$, and then adjoining a new edge f between $x' = Ro(RR_X(x))$ and Y as follows:

1. Create a new node y' that subdivides the edge $(Pa_T(y), y)$.

2. Adjoin the edge f between nodes x' and y'.

3. Suppress the node u, and rename x' as v and y' as u.

4. Contract the root edge.

We say that the tree $TBR_T(v, x, y)$ is obtained from T by a tree bisection and reconnection (TBR) operation that bisects the tree T into the components X, Y and reconnects them above the nodes x, y.

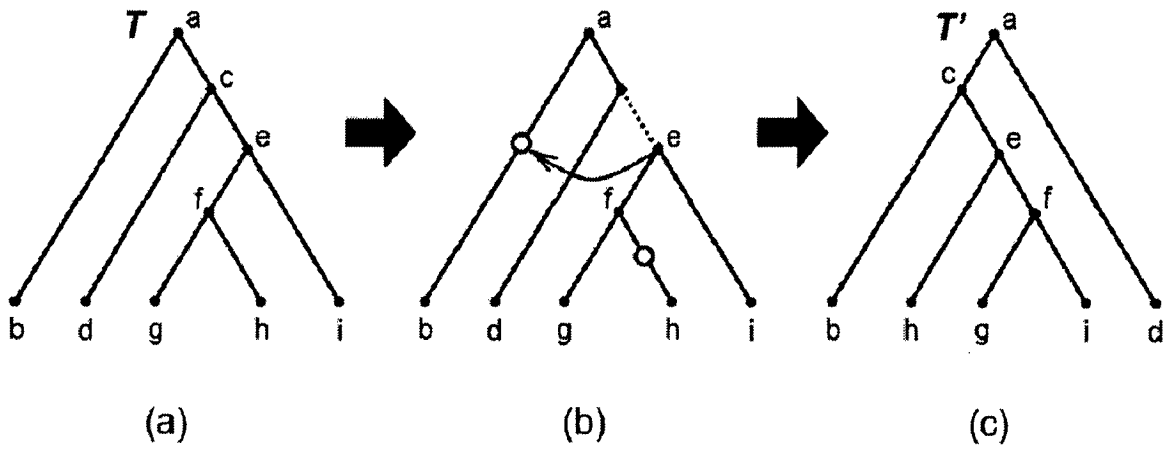

Figure 4.2 Tree Bisection and Reconnect operation.

The original tree T is shown in (a). In (b), we first remove the edge above e, that is we prune the subtree Te. Then we introduce a new node above h which will be the new root of the pruned subtree. We also introduce a new node above b this is where we will reconnect the subtree back to T. Finally we rearrange the tree and obtain the resulting tree T' as in (c).

We define the following extensions for the TBR operation:

1. $TBR_T(v, x) = \cup_{y \in Y} TBR_T(v, x, y)$

2. $TBR_T(v) = \cup_{x \in X} TBR_T(v, x)$

3. $TBR_T = \cup_{(u, v) \in E(T)} TBR_T(v)$

4.1.3 Subtree Pruning and Regrafting operation

An SPR operation for a tree T can be briefly described through the following steps: (i) prune some subtree S from T, (ii) add a root edge to the remaining tree T', (iii) regraft S into an edge of the remaining tree T', (iv) contract the root edge.

Let T be a tree, e = (u, v) ∈ E(T), and X, Y be the connected components that are obtained by removing edge e from T where v ∈ X and u ∈ Y. We define $SPR_T(v, y)$ for y ∈ Y to be $TBR_T(v, v, y)$. We say that the tree $SPR_T(v, y)$ is obtained from T by a subtree prune and regraft (SPR) operation that prunes subtree Tv and regrafts it above node y.

We define the following extensions of the SPR operation:

1. $SPR_T(v) = \cup_{y \in Y} SPR_T(v, y)$

2. $SPR_T = \cup_{(u, v) \in E(T)} SPR_T(v)$

4.2 Triplet Supertree Heuristic Algorithm [36]

SPR Scoring (SPR-S) for a given a profile P is, a supertree T of P, find a tree T* ∈ SPR_T such that

$$S(P, T^*) = max_{T' \epsilon SPR} S(P, T') \qquad (4.2)$$

SPR-Restricted Scoring (SPR-RS) for a given a profile P is , a supertree T of P, and (u, v) ∈ E(T), find a tree T* ∈ $SPR_T(v)$ such that

$$S(P, T^*) = max_{T' \epsilon SPR_T} S(P, T') \qquad (4.3)$$

Algorithm SRR-S problem (P,T) [36]
Input: A profile P = (T1,..., Tn), a supertree T of P
Output: T* ∈ SPR_T, and $\Delta_P(T, T^*)$
for all (u, v) ∈ E(T) do
Store the value of SPR-RS(P, T, (u, v))
end for
(T*, d) ← the stored value of SPR-RS calls that has the maximum
score increase by traversing the tree T in post order

```
    return (T*, d)
 end procedure

Algorithm for the SPR-RS problem
Input: A profile P = (T1,...,Tn), a supertree T of P, and (u, v) ∈ E(T)
Output: T* ∈ SPR_T(v), and Δ_P(T, T*)
       r ← Ro(T)
T̂ ← SPR_T(v, r)
    Call MovedownAndCompute(P, T̂, v)
      Traverse the tree Tr in pre-order to compute Δ_P(T, T') for each
      T' ∈ SPRT(v) using the values computed by MovedownAndCompute
      T* ← T' ∈ SPR_T(v) such that Δ_P(T,T')= max_{T''SPRT} Δ_P(T̂, T*)
      d ← ΔP(T̂, T*) - ΔP(T̂, T)
      return (T*, d)
     end procedure

procedure MovedownAndCompute(P, T, v)
     Input: A profile P, a tree T, and v ∈ V(T)
     yz ▶ ◁ v ← The immediate triplet induced by v in T
     for all t ∈ {y, z} do
       T' ← SPRT(v, t)
       Compute and store Δ_P(T, T')
       Call MovedownAndCompute(P, T', v)
     end for
 end procedure
```

Figure 4.3 Algorithm for SPR-S problem

Above algorithm is solving the SPR-S problem by using sub algorithm SPR-RS. The SPR-RS algorithm is solving for a given a profile P, a supertree T of P, and (u, v) ∈E(T), we compute $\Delta_P(T, T')$ for each $T' \in SPR_T(v)$ by first pruning and regrafting Tv to Ro(T) and compute the score differences for each "move-down" operation, then traverse T in preorder to obtain the tree that gives the maximum score difference.

```
procedure PreprocessTripletSum(P) [ 36]
  Input: A profile P = (T1,..., Tn)
  Initialize all values of σP to 0
  for i = 1 to n do
    for all u ∈ V(Ti) in post-order, u ∉ Le(Ti) do
      {v, w} ← ChT(u)
      for all {x, y} ∈ Le(Tv), z ∈ Le(Tw) do
        Increment σP(xy|z)
      end for
      for all {x, y} ∈ Le(Tw), z ∈ Le(Tv) do
        Increment σP(xy|z)
      end for
    end for
  end for
procedure PreprocessExtendedTripletSum(P, T)
  Input: A profile P = (T1,..., Tn), a supertree T of P
  for all u ∈ V(T) in post-order do
    for all v ∈ V(T) in post-order after u, lcaT({u, v}) ∉{u, v} do
      for all w ∈ V(T) in post-order after v, lcaT({u, v, w})∉ {u, v, w} do
        if w ∈ Le(T) then
         if v ∈ Le(T) then
          if u ∈ Le(T) then
            σP, T(uv|w) ← σP(uv|w)
            σP, T(uw|v) ← σP(uw|v)
            σP, T(vw|u) ← σP(vw|u)
          else {u1, u2} ← ChT(u)
              σP, T(uv|w) ← σP, T(u1v|w) + σP, T(u2v|w)
              σP, T(uw|v) ← σP, T(u1w|v) + σP, T(u2w|v)
              σP, T(vw|u) ← σP, T(vw|u1) + σP, T(vw|u2)
        end if
        else {v1, v2} ← ChT(v)
            σP, T(uv|w) ← σP, T(uv1|w) + σP, T(uv2|w)
            σP, T(uw|v) ← σP, T(uw|v1) + σP, T(uw|v2)
            σP, T(vw|u) ← σP, T(v1w|u) + σP, T(v2w|u)
        end if
         else {w1, w2} ← ChT(w)
            σP, T(uv|w) ← σP, T(uv|w1) + σP, T(uv|w2)
            σP, T(uw|v) ← σP, T(uw1|v) + σP, T(uw2|v)
            σP, T(vw|u) ← σP, T(vw1|u) + σP, T(vw2|u)
          end if
      end for
    end for
end for
```

Figure 4.4 Algorithms for Preprocess TripletSum and Extended TripletSum

Above two algorithms are used to preprocess tripletsums. In the preprocess tripletsum the value is assigned to each triplet by the total number of times it present in each tree. For each SPR-S problem Preprocess Extend Tripletsum is called once. The overall time complexity is $O(n^3)$.With an expense of initial preprocessing of $o(kn^3)$, where k is number of input trees, and n is number of taxa in the input trees.

4.2 Implementation and Results

The algorithm Triplet Heuristics is implemented in java using MyEclispse IDE. All executions done on a computer with the following specifications

Hardware specifications:

4 GB RAM

Core2Duo processor

Software specifications:

MyEclipse software.

Results

The running times are shown in Table 4.1 for TILI using DM topology detection and TH . The executions are on a different forest of the same size. There is an entry for each forest size from 20 to 60 taxa, in increments of ten.

Taxa	TILI	TH
20	0.78	0.52
30	3.39	2.18
40	12.34	8.89
50	36.62	20.28
60	85.72	60.25

Table 4.1 Time in seconds required by TILI and TH for constructing supertree.

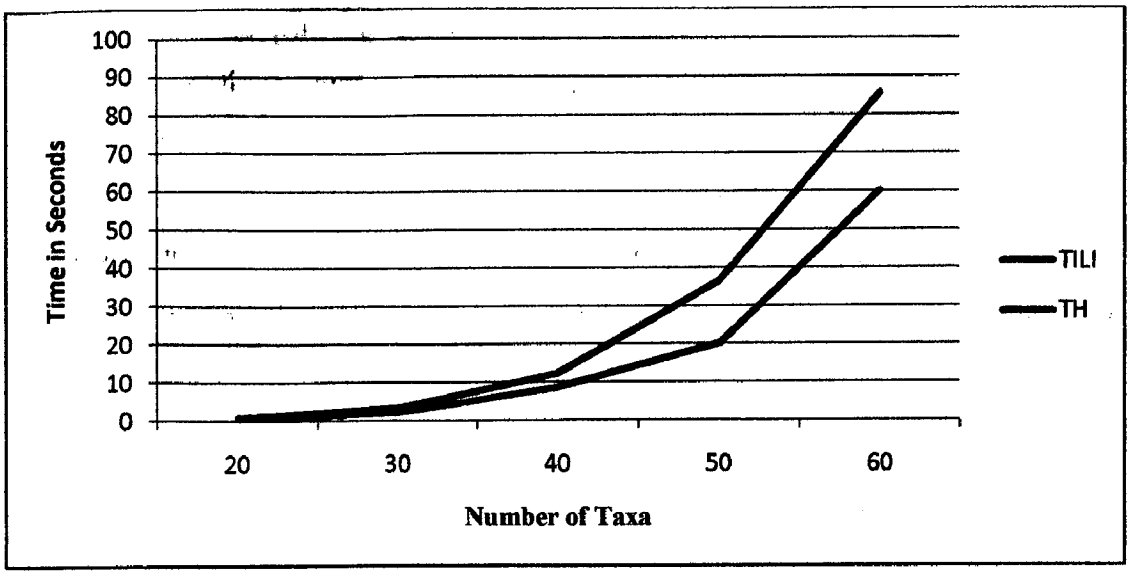

Figure 4.3 Runtime comparisons between TILI and TH

The metric we measure for these two algorithms is run time. From the above we can understand that the running time for construction of supertree for the algorithm TH has less executiontime than the algorithm using TILI for the same input set. TILI is involving in inferring missing triplets which results in requiring more execution time.

The other metric we measure is triplet similarity score which is the number of triplets present for the given input profile of trees and the constructed supertree.

Dataset (No .of Taxa)	Method	Triplet similarity
20	TILI TH	78.12 97.15
30	TILI TH	79.24 97.15
40	TILI TH	72.15 98.10
50	TILI TH	72.15 98.10
60	TILI TH	70.17 98.20

Table 4.2 Triplet Similarity Score between TILI and TH

From the table we can understand that TH algorithm is having more triplet similarity score than TILI algorithm. The triplet similarity for TILI algorithm is reducing as the number of taxa is increasing in the input data set since for each time as the number of taxa increases the input profile of trees is also increases which results in decrease of the triplet similarity for the TILI algorithm. But the TH algorithm is always calculates the similarity score in every step of supertree construction so it is almost constant for it even though the input size increases.

Chapter 5
Conclusion and Future Work

5.1 Conclusion

The objective of this thesis is implementation of triplet supertree algorithms for construction ofphylogenetic supertrees. The following sections describe what has been done:

In this thesis, the requirements for the supertree algorithms are explained. Then the process of phylogenetic study is presented. In chapter 2 the various supertree construction algorithms like Mincut supertree, Modified Mincut algorithm, Matrix Representation with Parsimony are studied. In Matrix Representation with parsimony method, though the solution is guaranteed, the time taken to execute these algorithms will be more compared to both Triplet Inference and Local Inconsistency algorithm (TILI) and Triplet supertree heuristic algorithm (TH).

Later TILI algorithm is implemented in Java +6 SDK and results are obtained under various input data of having different number of taxa. Likewise, TH algorithm is also implemented and executed under same input data. Comparative study based on execution time and accuracy measurement has been done for both the algorithms. The study shows execution time of TH algorithm is less compared to the other one, and is more accurate. From this it can be concluded that TH algorithm is superior to TILI.

5.2 Future Work

Drawing inference is the vital phase in TILI algorithm which in turnaffects the accuracy. This can be improved by considering missing triplets through new approaches. This can be further improved by following better weight model.

REFERENCES

[1] E. Pennisi. *"Modernizing the Tree of Life"*.Science, vol. 300, issue 5626, pp. 1692-1697, 2003.

[2] T. A. Brown and K. A. Brown. *"Ancient DNA: Using molecular biology to explore the pas"t.* Bioassays,Vol.16, issue 10, pp.719–726,1994.

[3] B. Kolaczkowski and J. W. Thornton. *"Performance of maximum parsimony and likelihood phylogenetics when evolution is heterogeneous"*.Nature, vol.431, pp. 980-984, 2004.

[4] B. Snel. *"Genome trees and the nature of genome evolution"*. Annual Reviewof Microbiology, vol.59, pp.191-209, 2004.

[5] A. R. Templeton. *"Human origins and analysis of mitochondrial DNA sequences"*.Science, vol.255, issue 5045, pp.737–737, 1992.

[6] J. Felsenstein. *"Phylogenies and the comparative method.American Naturalist"*,vol.125, pp.1–15, 1985.

[7] D. A. Liberles, D. R. Schreiber, S. Govindarajan, S. G. Chamberlin, and S. A. Benner. *"The adaptive evolution database (TAED)"*. Genome Biology, vol.2, issue 8,2001.

[8] J. A. Eisen. *"Phylogenomics: Improving functional predictions for uncharacterized genes by evolutionary analysis"*. Genome Research, vol.8, issue 3, pp.163–167,1998.

[9] J. R. Brown and P. V. Warren. *"Antibiotic discovery: Is it in the genes? Drug Discovery"*Today, vol.3, pp.564–566, 1998.

[10] R. Overbeek , N. Larsen, and G. D. Pusch. *"WIT: integrated system for high throughput genome sequence analysis and metabolicreconstructio."*. Nucleic AcidsResearch, vol. 28, issue 1, pp.123–125, 2000.

[11] F. G. Candelas, M. A. Bracho, and A. *"Moya.Molecular epidemiology and forensic genetics: Application to a hepatitis c virus transmission event at a hemodialysis unit"*. Journal of Infectious Diseases, vol. 187,issue 3, pp.352–358, 2003.

[12] J. Gatesy and M. S. Springer, "A critique of matrix representation with parsimony supertrees," *in Phylogenetic Supertrees.Combining Information to Reveal the Tree of Life Computational Biology*, O. R. P. Binnida-Emonds, Ed., vol. 3, pp.369–388, Kluwer Academic Publishers, Dordrecth, The Netherlands, 2004.

[13] M. Wilkinson, J. A. Cotton, and J. L. Thorley. *"The information content of trees and their matrix representations"*. Systematic Biology,vol.53, issue 6, pp.989–1001, 2004.

[14] O. R. P. Bininda-Emonds. *"MRP supertree construction in the consensus setting"*. InBioconsensus, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol.61, pp. 231–242. American Mathematical Society-DIMACS,Providence, Rhode Island, 2003.

[15] F. R. McMorris, D. B. Meronk, and D. A. Neumann. *"A view of some consensus methods for trees"*.In J. Felsenstein, editor, Numerical Taxonomy,pp 122–125. Springer-Verlag, 1983.

[16] D. L. Swofford. *"When are phylogeny estimates from molecular and morphological data incongruent?"*In M. M. Miyamoto and J. Cracraft, editors, Phylogenetic analysis of DNA sequences, pp.295–333. Oxford UniversityPress, 1991.

[17] J. P. Barthelemy and F. R. McMorris. *"The median procedure for n-trees"*.J. Classif., vol 3, pp.329–334, 1986.

[18] J. D. Thompson, D. G. Higgins, and T. J. Gibson. *"ClustalW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice*". Nucleic Acids Research,vol 22,pp.4673–4680, 1994.

[19] D. L. Swofford, G. J. Olson, P. J. Waddell, and D. M. Hillis. *"Phylogenetic Inference"*, pp 407–425. Sinauer Associates, Sunderland, Massachusetts, 2nd editionedition, 1996.

[20] L. R. Foulds and R. L. Graham. *"The steiner problem in phylogeny is NP-complete"*.Advances in Applied Mathematics, vol. 3, issue 1, pp.43-49, 1982.

[21] B. Chor and T. Tuller. *"Maximum likelihood of evolutionary trees: hardness and approximation"*. Bioinformatics, vol21, issue 1, pp.97–106, 2005.

[22] N. Saitou and M. Nei. *"The neighbor-joining method: a new method for reconstructing phylogenetic trees"*. Molecular Biology and Evolution, vol 4, pp.406–425, 1987.

[23] B. R. Baum. *"Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees"*. Taxon, vol.41,pp.3–10, 1992.

[24] M. A. Ragan. *"Matrix representation in reconstructing phylogenetic relationships among the eukaryotes"*.Biosystems, vol.28, pp.47–55, 1992.

[25] A. V. Aho, T. G. Sagiv, T. G. Szymanski, and J. D. Ullman. *"Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions"*. SIAM Journal on Computing, vol 10, pp.405–421, 1981.

[26] C. Semple and M. Steel.*"A supertree method for rooted trees"*.Discrete Applied Mathematics, vol 105, pp.147–148, 2000.

[27] R. D. M. Page. *"Modified mincut supertrees"*.WABI 2002 Lecture Notes in ComputerScience, LNCS-2452, pp.537–551, 2002.

[28] X. Gu. *"Maximum-likelihood approach for gene family evolution under functional divergence"*.Molecular Biology and Evolution., vol.18, pp.453–464, 2001.

[29] D. Chen, L. Diao, O. Eulenstein, D. Fenndez-Baca, and M. J. Sanderson. *"Flipping: A supertree construction method"*.In Bioconsensus, pp.135–160. DIMACS series in discrete mathematics and theoretical computerscience, American Mathematical Society, Providence, 2003.

[30] O. Eulenstein, D. Chen, J. G. Burleigh, D. Fernandez-Baca, and M. J.Sanderson. *"Performance of flip-supertree construction with a heuristic algorithm"*. Systematic Biology, vol .53,issue 2,pp299–308, 2004.

[31] C.Mosses. *"Triplet supertrees"*.PhD Thesis, University of Aarhus, Aarhus,Denmark, 2005.

[32] D. Chen, O. Eulenstein, D. Fernandez-Baca, J. G. Burleigh:*"Improved heuristics for minimum-flip supertree construction"*. Evolutionary Bioinformatics, vol.2,pp.401-410,2006.

[33] M. Bansal, O. Eulenstein: *"The Gene-Duplication Problem: NearLinear Time Algorithms for NNI Based Local Searches"*. Bioinformatics Research and Applications pp.14-25,2008.

[34] Bryant D: *"Building Trees, Hunting for Trees, and Comparing Trees – Theory and Methods in Phylogenetic Analysis"*. In PhDThesis University of Canterbury; 1997.

[35] Semple C, Steel M: *"Phylogenetics"*, Oxford University Press; 2003.

[36] H. T. Lin, J. G. Burleigh , O. Eulenstein: *"Triplet supertree heuristics for the tree of life"*. BMC Bioinformatics,vol.10, issue 1,pages:S8,2009.