universität
wien

DISSERTATION

Titel der Dissertation

„Phylogenomics: theory, algorithms and applications“

Verfasserin

Olga Chernomor

angestrebter akademischer Grad

Doctor of Philosophy (PhD)

Wien, 2015

Studienkennzahl lt. Studienblatt:	A 794 685 490
Dissertationsgebiet lt. Studienblatt:	Molekulare Biologie, IK: Computational Sciences
Betreuer:	Univ.-Prof.Dr. Arndt von Haeseler

PHYLOGENOMICS:
THEORY, ALGORITHMS AND APPLICATIONS

- Phylogenetic partial terraces and their detection
- Terrace aware data structure for phylogenomic inference from supermatrices
- Phylogenomics in conservation prioritization: extensions to viable taxon selection problem

Abstract

In the recent years phylogenomics opened a new chapter in evolutionary biology. The analysis of the genome-scale data sets has the potential of answering the most difficult and intriguing questions for evolutionary histories. However, such perspectives come with a higher complexity and difficulties for the phylogenomic inference. The focus of this thesis is exploring and providing insights into some of the questions arisen from theory, algorithms and applications of phylogenomics.

The main contributions of the thesis deal with *phylogenetic terraces*, which represent sets of species trees in tree space with identical score (likelihood or parsimony).

Firstly, we provide the rules to detect terraces during the tree search. To this end we study the *induced partition trees* and how topological rearrangements on species tree drive changes on partition trees. If the tree rearrangement operation applied to the current species tree does not change any of its associated induced partition trees, then the current and a new species trees belong to one terrace. We prove three propositions defining the rules when Nearest Neighbour Interchange (NNI), Subtree Pruning and Regrafting (SPR) and Tree Bisection and Reconnection (TBR) operations change the induced partition trees. We further generalize the concept of terraces to *partial terraces* and study their occurrence for real alignments using NNI neighbourhoods.

Secondly, we provide a phylogenetic terrace aware data structure (PTA) for the efficient analysis of concatenated multiple alignments. Using PTA and the rules developed to detect (partial) terraces in the presence of missing data one saves computational time by avoiding unnecessary recomputations. We implemented PTA in IQ-TREE and tested its performance on 11 real alignments. Identification of partial terraces speeded up the tree search with IQ-TREE for up to 5 and 6 times compared to the standard implementation (terrace-unaware) and RAxML, respectively. PTA is suitable for the use with all partition models and all common topological rearrangement operations, such as NNI, SPR and TBR.

Finally, we develop methods for conservation biology and ecology, where phylogenomics is used to quantify the evolutionary diversity of the species. We discuss the viable taxon selection problem, which incorporates predator-prey interactions to define viability constraints. First, we extend the problem to account for Split Diversity (SD), a biodiversity measure, which is based on the evolutionary distances between species on split networks. Second, to make the viability constraints more realistic we extend the viability definition to account for the diet composition of predators. SD with the viability constraints is used to prioritize species for the conservation actions. Though such optimization problems fall into the area of NP-hard problems, it is possible to solve them within reasonable amount of time using Integer Linear Programming (ILP), a well-known method for the decision-making problems. We provide the ILP formulations for all the discussed problems and implement them in the PDA software package. To

exemplify the discussed methods we apply them to a real case study – the Caribbean Coral Reef community.

Parts of this thesis were published in the following articles

(i). Chernomor, O., Minh, B.Q. and von Haeseler, A. (2015) Consequences of Common Topological Rearrangements for Partition Trees in Phylogenomic Inference, *Journal of Computational Biology*. (DOI: 10.1089/cmb.2015.0146)

(ii). Chernomor, O*., et al.* (2015) Split diversity in constrained conservation prioritization using integer linear programming, *Methods in ecology and evolution / British Ecological Society*, **6**, 83-91. (DOI: 10.1111/2041-210X.12299)

and in a book chapter

(iii). Chernomor, O*., et al.* (2016) Split diversity: measuring and optimizing biodiversity using phylogenetic split networks. In Pellens, R. and Grandcolas, P. (eds), *Biodiversity Conservation and Phylogenetic Systematics* Springer International Publishing, in press.

Submitted

(i). Chernomor, O., von Haeseler, A. and Minh, B.Q. Terrace Aware Data Structure for Phylogenomic Inference from Supermatrices.

Zusammenfassung

In der Evolutionsbiologie wurde in den letzten Jahren durch Aufkommen der Phylogenomik ein neues Kapitel aufgeschlagen. Mit Hilfe genomweiter Analysen wird es zunehmend möglich, schwierige Fragestellungen zur Evolutionsgeschichten zu untersuchen, doch diese neuen Perspektiven sind gleichzeitig verbunden mit erhöhter Komplexität und anderen Schwierigkeiten bei der Stammbaumberechnung. Der Fokus dieser Dissertation ist es, Einblicke in Fragen aus der Theorie der Baumrekonstruktion sowie zu Algorithmen und Anwendungen der Phylogenomik zu geben.

Ein Hauptaugenmerk liegt hierbei auf der Untersuchung phylogenetischer Terrassen, dies Speziesbäume mit gleichen Maximum Likelihood oder Maximum Parsimonie Werten. Ein Problem bei der Suche nach optimalen Bäumen ist die Größe der Terrasse. Die Größe der Terrassen hängt dabei entscheidend von dem untersuchten multiple Sequenz Alignment ab.

Im Rahmen dieser Arbeit wird zunächst erklärt wie man Terrassen während der Baumsuche detektiert. Dazu studieren wir durch Partitionen induzierte Bäume und in welcher Weise topologische Umformungen am Speziesbaum Änderungen an den Partitionsbäumen bedingen. Sollte eine Änderung der Verzweigungsstruktur eines Speziesbaum keine Änderung an den mit ihm assoziierten induzierten Partitionsbäumen hervorrufen, gehören sowohl der gegenwärtige als auch der geänderte neue Speziesbaum zur selben Terrasse.

Es werden drei Propositionen bewiesen, die Kriterien definieren nach welchen verschiedene Umformungsoperationen, namentlich NNI (Nearest Neighbour Interchange), SPR (Subtree Pruning and Regrafting) und TBR (Tree Bisection and Reconnection), sich die induzierten Partitionsbäume verändern. Weiters wird das Konzept von Terrassen erweitert, in dem partielle Terrassen definiert werden und ihr Auftreten für echte Alignments unter NNI Umformungsoperationen untersucht wird.

Im zweiten Teil wird die Datenstruktur, PTA, (phylogenetic terrace aware data structure) vorgestellt, die eine effiziente Analyse verknüpfter multipler Alignments unter Berücksichtigung phylogenetischer Terrassen ermöglicht. Mit Hilfe von PTA und den Kriterien zur Erfassung (partieller) Terrassen ist es möglich, überflüssige Neuberechnungen der Maximum Likelihood oder Maximum Parsimonie Werte zu vermeiden und so die für die Baumsuche benötigte Rechenzeit zu verringern. Durch die Identifizierung partieller Terrassen wird im Vergleich zur Standardimplementierung eine bis zu 5-fache Beschleunigung von IQ-TREE festgestellt und nach der Implementierung der Terrassenidentifikation ist IQ-TREE in der Lage bis zu 6 Mal schneller Maximum-Likelihood-Bäume zu finden als RAxML. Die Datenstruktur PTA eignet sich für den Einsatz mit allen Partitionsmodellen und für alle üblichen topologischen Umformungen wie NNI, SPR und TBR.

Im Schlussteil dieser Arbeit werden Methoden für den Einsatz in Naturschutzbiologie und -ökologie eingeführt und diskutiert, wobei Phylogenomik herangezogen wird, um die evolutionäre Diversität verschiedener Spezies zu quantifizieren. Wir diskutieren die Aufgabe der Auswahl überlebensfähiger Taxa, ein Optimierungsproblem unter Einbeziehung von Räuber-Beute-Interaktionen. Zuerst wird dabei der Rahmen der Aufgabenstellung erweitert, um auch die Splitdiversität (SD) zu erfassen, ein Biodiversitätsmaß welches auf der evolutionären Distanz zwischen verschiedenen Spezies in Splitnetzwerken basiert. Danach erweitern wir die Definition von Lebensfähigkeit um die Nahrungszusammensetzung des Räubers miteinzubeziehen und so die Modellierung realistischer zu gestalten. Mit Hilfe der SD und unter Berücksichtigung eines realistischen Modells werden Spezies für Naturschutzmaßnahmen priorisiert. Obwohl derartige Optimierungsaufgaben in den Bereich NP-schwerer Probleme fallen, zeige ich, dass sie mit Hilfe von ILP (Integer Linear Programming) in überschaubarer Zeit gelöst werden können. In dieser Arbeit werden ILP-Ansätze für alle darin diskutierten Problemstellungen beschrieben sowie eine Implementierung im Software Paket PDA bereitgestellt.

Teile dieser Arbeit wurden in den folgenden Artikeln publiziert

(i). Chernomor, O., Minh, B.Q. and von Haeseler, A. (2015) Consequences of Common Topological Rearrangements for Partition Trees in Phylogenomic Inference, *Journal of Computational Biology*. (DOI: 10.1089/cmb.2015.0146)

(ii). Chernomor, O., *et al.* (2015) Split diversity in constrained conservation prioritization using integer linear programming, *Methods in ecology and evolution / British Ecological Society*, **6**, 83-91. (DOI: 10.1111/2041-210X.12299)

und in einem Buchkapitel

(iii). Chernomor, O., *et al.* (2016) Split diversity: measuring and optimizing biodiversity using phylogenetic split networks. In Pellens, R. and Grandcolas, P. (eds), *Biodiversity Conservation and Phylogenetic Systematics* Springer International Publishing, im Druck.

Eingereicht

(i). Chernomor, O., von Haeseler, A. and Minh, B.Q. Terrace Aware Data Structure for Phylogenomic Inference from Supermatrices.

Contents

Chapter 1

Introduction

One of the fundamental questions in evolutionary biology is the reconstruction of evolutionary relationships of a set of species. To solve this problem molecular phylogenetics estimates evolutionary history from genetic sequences arranged in the so-called multiple sequence alignment. The input for traditional analysis is the alignment containing sequences for a single gene. But nowadays the gigantic amount of sequences available in the databases has transitioned phylogenetics to a genome scale, initiating by this the era of *phylogenomics* (Delsuc, et al. 2005; Liu, et al. 2015; Rannala and Yang 2008).

In general, phylogenomics denotes the inference of evolutionary relationships from more than one locus. The extensions of phylogenetic tree reconstruction approaches to phylogenomic alignments are jointly called *sequence-based methods* (Fig. 1.1). These are the *supertree* and the *supermatrix* approaches. The supertree methods are typically divided into one- and two-steps approaches. In the former one the species tree and the gene trees are estimated concurrently, which involves the use of multispecies coalescent models (Liu, et al. 2009). In the two-steps methods one first separately reconstructs gene trees using traditional phylogenetic methods (distance, parsimony or likelihood-based methods (Yang 2006)) and then combines them into one species tree (Sanderson, et al. 1998). A different strategy is used in the supermatrix method. Here, the gene alignments are first concatenated into one supermatrix, which then serves as an input for traditional tree reconstruction inferences (de Queiroz and Gatesy 2007).

The extension of well-developed techniques to larger data sets is a natural step. Nevertheless, the genome scale of phylogenomics provides a base for the development of new techniques, which infer evolutionary relationship based on the *whole-genome features* (e.g. gene order, gene content, distributions of DNA strings across genome or intron positions, for review see Delsuc, et al. (2005)). Such techniques are still at an immature state, but have a lot of potential to provide the resolution of molecular evolution on a different scale.

In this thesis we mainly focus on the supermatrix method, which nowadays remains the most widely used technique for the phylogenomic analysis of large data sets (e.g., Burleigh, et al. 2015; Dell'Ampio, et al. 2014; Fabre, et al. 2009; Goodheart, et al. 2015; Hedtke, et al. 2013; Jonsson, et al. 2015; Soltis, et al. 2013; von Haeseler 2012).

To discuss difficulties in the analysis of supermatrices, it is worth to mention, that the application of traditional phylogenetic methods to concatenated alignments unavoidably transfers all the typical reconstruction pitfalls to a larger scale. Among those are the weak and non-phylogenetic signals, which cannot be overcomed by simply analysing larger alignments (Philippe, et al. 2011). One of the most well known artefacts misleading all phylogenetic methods is the long branch attraction (Felsenstein 1978). Also the short internal branches, which are caused by the weak signal, are also problematic for the tree reconstruction (e.g., see Saitou and Nei (1986)). These artefacts will also affect phylogenomic inference and therefore require further theoretical understanding.

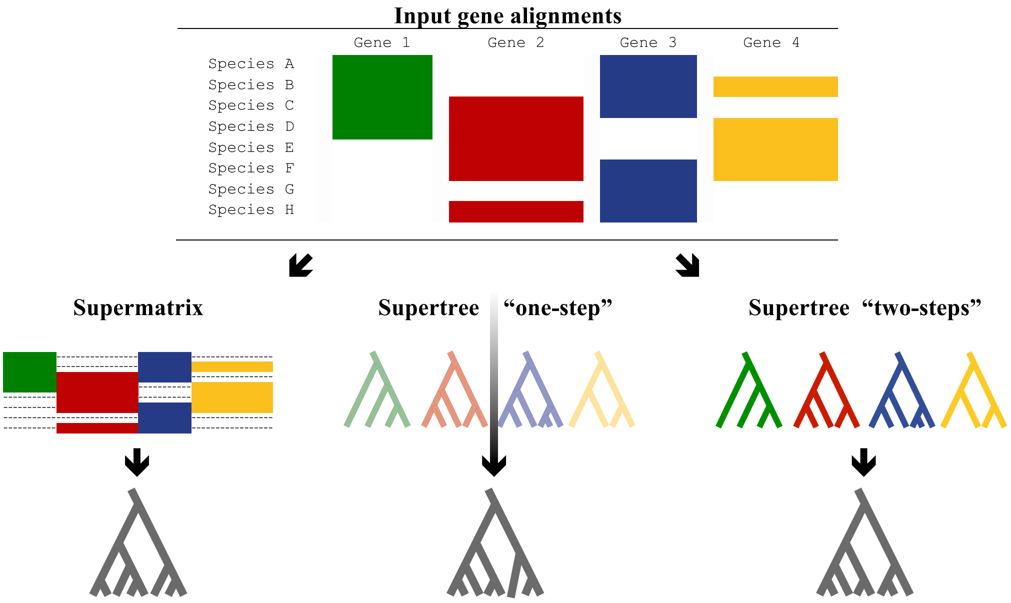


Figure 1.1. The schematic flow of sequence-based methods for phylogenomic inference. The input is the gene alignments (top panel). The white fields denote missing sequences. In the supermatrix method (bottom left) gene alignments are first concatenated into one "supergene". If some sequences are missing, they are substituted by gaps. The concatenated alignment is then used to reconstruct the species tree using typical phylogenetic methods. The one-step supertree methods (bottom center) reconstruct concurrently gene trees and species tree. On the bottom right is indicated the two-steps supertree methods, where one first infers the gene trees and then the information from them is combined to reconstruct the species tree.

Several issues concern statistical models. The mutations of sequences such as replacement of one character (nucleotide or amino acid) by another are modelled by the so-called *substitution models* (Yang 2006). The assumption of the correct substitution model is essential for the likelihood-based methods. If the model cannot capture the underlying evolutionary process, increasing the alignment length cannot correct for this (e.g., see Kumar, et al. (2012)).

The choice of a substitution model for supermatrices is not trivial. Unless it is known, that all genes in the concatenated alignment evolve similarly, in general, there is no reason to assume that a single substitution model can describe evolutionary histories of every gene. A natural solution for this problem on supermatrices was obtained by the development of *partition models* (Yang 1996), which allow each gene to have its own substitution model. In such a way partition models, for example, are able to account for heterogeneous evolution caused by heterotachy (Kolaczkowski and Thornton 2008), i.e. when the evolutionary rates vary over time (Lopez, et al. 2002). Thus, partition models are more realistic for multi-gene alignments.

The supermatrix method makes one general assumption, that the evolutionary process of multiple genes can be well described by only one tree. Therefore, using a supermatrix for the datasets that clearly violate this assumption might lead to inconsistent estimates of the species tree (Kubatko and Degnan 2007; Roch and Steel 2015).

In general, a tree can be viewed as a good and useful approximation of the evolutionary past. The rational for this is the fact that a vertical evolution, that is the flow of genetic material to an organism from its ancestor, is the primary mechanism for the inheritance of genetic information. However, one tree cannot perfectly display all complex evolutionary scenarios of multi-gene data sets. Even for a single gene it is questionable whether a tree is a good representation of the evolution (Morrison 2014). There is a lot of evidence of non-tree like *reticulate evolution* (Sneath 1975) (e.g., hybridization, horizontal or lateral gene transfer), especially among prokaryotes (Bapteste, et al. 2009). It is therefore suggested, that phylogenetic networks (Huson and Bryant 2006) could better resemble evolutionary process even on a single-gene scale.

Apart from the afore-mentioned theoretical complications related to the phylogenomic analysis, nowadays multi-gene data sets pose also a complex computational problem. The drastic increase of the input size challenges the state-of-the-art phylogenetic software from the memory usage and computational time. The data sets with more than thousands of genes are no longer exceptions in contemporary analysis (e.g., Dell'Ampio, et al. 2014; Gonzalez, et al. 2015; Goodheart, et al. 2015). Therefore, more than ever it is important to have highly optimized programs, which take into account intrinsic features of particular data sets to speed up the computation and to reduce the memory consumption.

Further development of theoretical methods and subsequent improvement of existing algorithms and software are of paramount importance for future phylogenomic applications. Ranging from forensics and pharmaceutics to studies of music, tale and language evolution, phylogenetics has a variety of successful applications also outside of the common molecular evolution framework (e.g., Brown, et al. 2014; Chambers, et al. 2000; Nakhleh, et al. 2005; Tehrani 2013).

One of the interesting applications is found in conservation biology and ecology. Here, evolutionary history is used to study and quantify the diversity of species (Faith 1992). Diversity measures are further used to prioritize species for conservation actions, which is an important problem in the world of limited resources (Monastersky 2014; Tollefson and Gilbert 2012). Advances in molecular evolution and computational techniques open new opportunities for decision-making problems in such applications (Pellens and Grandcolas 2016; Purvis, et al. 2005).

In this thesis we provide insights into several questions arising from different aspects of phylogenomic inference, covering particular problems from theory, algorithms and applications. Chapter 2 contributes to the theoretical understanding of phylogenomic inference and discusses how to detect phylogenetic full and partial terraces for all the common topological rearrangements. In Chapter 3 we show how the particular features of the tree space can be used for the efficient supermatrix inference in the presence of missing data. Chapter 4 discusses several problems from conservation biology to promote the application of phylogenomics in ecology and biodiversity studies.

1.1 Background

In this section we briefly introduce the basic concepts and definitions important for the following chapters.

1.1.1 Representation of evolution

In this thesis we mainly work with phylogenetic trees. However, the methods developed for the application of phylogenomics in conservation biology (Chapter 4) also involve phylogenetic networks. Here, we provide the basic ideas of evolutionary trees and networks.

It is common to represent evolutionary history by trees, where the tree leaves denote contemporary species and the internal nodes are the ancestors. The branching order reflects the speciation events and it is typically assumed that the speciation results in two daughter species (i.e. the degree of inner nodes is 3; such trees are called *bifurcating*). The edges reflect the descent with modification and the edge length is the expected number of substitutions per site. Evolutionary trees can be rooted or unrooted. In the following we only discuss bifurcating unrooted trees.

When trees for the same species set exhibit different topologies (i.e. the trees are incongruent), phylogenetic signals from these trees can be represented by a *split system*. Each edge on the tree defines a split of taxa in two non-overlapping non-empty subsets. A split system is a union of all such bipartitions from corresponding incongruent trees.

A *split network* is a representation of the split system (Fig. 1.2). The conflicting signals are represented in the split network by the parallelograms and cubes. To obtain a bipartition from the network one cuts either a single edge or parallel edges. Instead of edge lengths, split networks use split weights. They can be computed, for example, as the sum of corresponding tree edge lengths weighted by the alignment lengths used to reconstruct these trees.

When the trees have exactly the same species sets, no particular algorithm is needed to reconstruct a split system from them. One simply collects all the splits from these trees. For the algorithm to construct a split system and its corresponding network from the trees with overlapping species sets see Huson, et al. (2004).

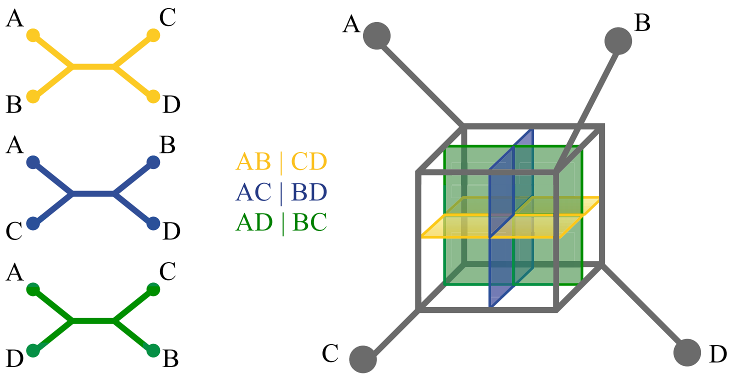


Figure 1.2. The information from all alternative tree topologies for species A, B, C, D (yellow, blue and green trees) can be displayed on one split network (right panel). We denote a bipartition of species by a vertical bar "|". The central edges on the trees define incompatible bipartitions $AB|CD$, $AC|BD$, and $AD|BC$, while the remaining trivial bipartitions are common for all trees: $A|BCD$, $B|ACD$, $C|ABD$, and $D|ABC$. The split network combines all bipartitions from these trees. By cutting along planes inside the cube of a network one retrieves back three incongruent splits. The trivial splits, separating one species from the rest, are obtained from the network by cutting edges incident to species A, B, C, D.

1.1.2 Mutations and their modelling

The genetic sequences are prone to different types of mutations. In this section we discuss the important types of one-character mutations and start with *substitutions*.

Models of evolution are essential for likelihood-based phylogenetic methods. *Substitution models* describe the rates at which one character substitutes another one during evolution (for an example of nucleotide substitutions see Figure 1.3). The commonly used substitution models make several assumptions. The first assumption is that all the sites in the alignment evolve according to the same Markov process. The Markov property states that the future is determined by the current state, while no assumption is made about the past. It is also assumed that character frequencies and the evolutionary model are constant through time and across sites of the alignment.

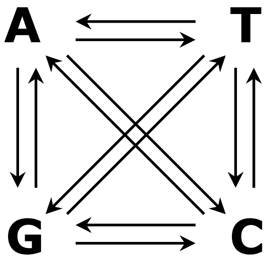


Figure 1.3. Nucleotide substitutions. Transitions are the substitutions between the two pyrimidines (T↔C) or between the two purines (A↔G). Transversions are the substitutions between a pyrimidine and a purine.

For mathematical tractability, the most widely used substitution models are time-reversible. Which means that at equilibrium the expected amount of change from character c_1 to c_2 is the same as the amount of change from c_2 to c_1. Note, though, that the mutation rates from c_1 to c_2 and from c_2 to c_1 may differ. If the model is time-reversible the likelihood of the tree does not depend on the placement of the root. Therefore, under reversibility the likelihood of the rooted tree is the same as for the unrooted one. This is called a pulley principle (Felsenstein 1981). In such a way, time-reversibility substantially reduces the computational effort, since one only has to explore the space of unrooted trees.

The nucleotide models differ from each other in the assumptions about stationary base frequencies and substitution rates. For example, the simplest time-reversible nucleotide model, Jukes-Cantor (JC69; Jukes and Cantor 1969), assumes that nucleotides have uniform stationary frequencies (0.25) and the same substitution rate, μ. In contrast, the most general time-reversible (GTR) model of nucleotide substitution (Tavare 1986) assumes different frequencies of nucleotide occurrence and different substitution rates for each pair of nucleotides. The nucleotide frequencies for GTR are typically computed from the input alignment and are equal to the observed base frequencies. The substitution rates for all nucleotide models in maximum likelihood are optimized during the tree reconstruction. In Bayesian inference they are sampled from parameter distributions.

For protein models the evolutionary rates are pre-estimated from the empirical data (for example, see Whelan and Goldman (2001)), while the amino acid frequencies can either be set to $\frac{1}{20}$ or computed from the input alignment.

Additionally substitution models can be augmented to account for the site rate heterogeneity by using the invariable-sites plus a discrete gamma distribution of rate variation (Gu, et al. 1995; Yang 1994).

Apart of substitutions, evolutionary process also exhibits such mutations as insertion and deletion of characters ("*indels*"). Therefore, multiple sequence alignments also contain *gaps*, denoted by "-", which are commonly treated as missing data. In likelihood-based inferences one simply sums up the likelihood over all possible states for each gap.

1.1.3 Maximum likelihood (ML) inference

All the phylogenomic analyses presented in this thesis were carried out with ML. In this section we provide its general description.

ML is a well-established statistical method introduced to phylogenetics by Joseph Felsenstein (1981). ML aims to find a tree and model parameters that best explain the data – multiple sequence alignment. To this end, it maximizes the log-likelihood function

$$L(D|T,M),$$

where D is the alignment, T denotes the tree with its edge lengths and M is the substitution model. Since in phylogenetics the sites of the alignment are generally assumed to evolve independently, the log-likelihood is equal to the sum of per-site log-likelihoods

$$L(D|T,M) = \sum_{i=1}^{n} L(D_i|T,M),$$

where n is the number of sites and D_i denotes the i^{th} site of D. Given the alignment and the substitution model, the tree search is carried out by moving in tree space from one species tree to another via tree rearrangement operations applied to the current tree. Next, the edge lengths of the new tree are optimized. Finally, one computes whether the log-likelihood of the new tree is larger than the log-likelihood of the old tree. This process is repeated till no better tree can be found.

Under the assumption of the true model ML was shown to be statistically consistent (Rogers 1997), which means that as the alignment length tends to infinity the probability of finding the true parameter values (tree topology, edge lengths and model parameters) tends to one.

1.1.4 Species trees, induced partition trees and phylogenetic terraces

Two main contributions of this thesis are related to the concept of phylogenetic terraces (Sanderson, et al. 2011) and the induced partition trees. Here, we briefly introduce their definitions.

Phylogenomics aims to reconstruct a species tree from many genes, loci or, more generally, partitions. In the supermatrix method, where all gene alignments are first concatenated into one supergene, if a sequence is not available for some species, it is substituted by *gaps* (Fig. 1.1), which constitute missing data. With each partition there is an associated *induced partition tree*. If the partition contains sequences for all the species from the set, then the induced partition tree is the same as the species tree. Otherwise, it is obtained from the species tree by removing taxa (and corresponding edges) with no available sequence for a partition of interest (Fig. 1.4).

Supermatrix

	Partition 1	Partition 2
Species A	CAGTAAGCTGAGCTGCA	ATTGACATGAATTGAGA
Species B	CAGCCAGCTGAGCTGCG	-----------------
Species C	CTGTAAGCTACATAACC	ACATCCATGAATTGAGC
Species D	CAGTGCTCAGAGCTGCG	-----------------
Species E	CAGTAAGCTGAGGCTCA	ATACTTTTCATACGAGC

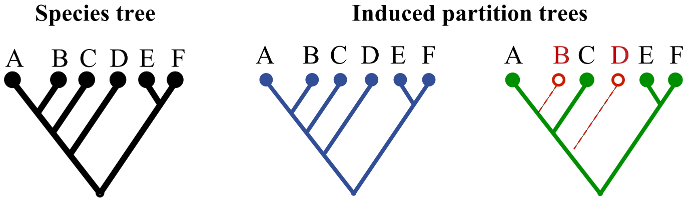


Figure 1.4. An example of induced partition trees. Partition 1 has sequences for all the species in the set. Therefore, its induced partition tree is identical to the species tree. While for partition 2 sequences for species B and D are not available. To obtain its induced partition tree we delete the corresponding species and their incident edges from the species tree. In general, one deletes all the edges below the missing species up to a common ancestor with the available species. Here, according to the species tree, species B has a common ancestor with A, while species D has a common ancestor with A, B, and C. Therefore, we only have to delete the edges incident to species B and D.

Sanderson, et al. (2011) discovered that in the presence of missing data under partition models the tree space might contain such structures as *phylogenetic terraces*. A phylogenetic terrace is a collection of species trees that have exactly the same sets of induced partition trees and as a result have identical score (likelihood or parsimony). The number of trees on one terrace as well as the number of terraces for sparse supermatrices can be very large (Sanderson, et al. 2011).

1.1.5 Evolutionary measures of biodiversity

In Chapter 4 we develop methods for phylogenomic application in conservation biology. This application is related to the biodiversity and its evolutionary measures. Here, we formally provide their definitions.

Biodiversity refers to a variety of life, including variations in genes, species and functional traits. One way to quantify it is to use the evolutionary distances among species. *Phylogenetic Diversity* (PD; Faith 1992) is computed on phylogenetic tree and is equal to the sum of the edge lengths of the (sub)tree spanned by the species of interest.

Another diversity measure, called *Split Diversity* (SD; Minh, et al. 2009), is computed from phylogenetic split systems and is equal to the sum of split weights separating at least two taxa from the species set of interest. SD generalizes PD to account for incongruences in phylogenetic signals from multiple genes.

1.2 Main contributions of the thesis

1.2.1 Theoretical insights into analysis of supermatrices

In Chapter 2 we answer the question of how to quickly detect terraces during tree search. To this end we prove three conditions (Propositions 2.1-2.3) under which most common topological rearrangements, such as Nearest Neighbour Interchange (NNI), Subtree Pruning and Regrafting and Tree Bisection and Reconnection (Felsenstein 2004), change the topology of the corresponding induced partition trees. We also discuss the specific features of different partition models in ML and how they influence edge length optimization of induced partition trees. Moreover, we generalize the concept of terraces to *partial terraces* and study their occurrence for the real alignments using NNI neighbourhoods. Overall, our results provide theoretical insights into the structure of tree spaces, imposed by the missing data, as well as how to identify problematic cases like (partial) terraces during tree search.

1.2.2 From theory to algorithms: efficient data structure for phylogenomic inference from supermatrices

Contributing to the algorithms for phylogenomics, in Chapter 3 we introduce a phylogenetic terrace aware data structure (PTA) for the efficient phylogenetic inference from supermatrices. This data structure can be used in combination with the rules developed in Chapter 2 to speed up the tree search algorithms in the presence of missing data for all types of partition models. We implemented PTA data structure in IQ-TREE (Nguyen, et al. 2015) and tested its performance on 11 real alignments. Accounting for (partial) terraces PTA speeded up the tree reconstruction for all types of partition models with a maximum speed up of 5. We also compared the results with RAxML (Stamatakis 2014) and observed that PTA implementation outperformed RAxML for the majority of the alignments with a maximum speed up of 6. PTA data structure is a valuable contribution for phylogenomic inferences from supermatrices under partition models.

1.2.3 The application of phylogenomics: the development of computational techniques for conservation prioritization

We promote the application of phylogenomics in conservation biology by providing the easy to use techniques for *viable taxon selection* (Moulton, et al. 2007) problem accompanied by Split Diversity. Viable taxon selection aims at prioritizing subsets of species for conservation actions accounting for their predator-prey relationships. We modelled this problem in terms of Integer Linear Programing (ILP; Gomory 1958), a well known method for decision-making problems. To make the concept of viability more realistic we generalized it to account also for the species diet composition. We implemented both problems in PDA (Minh, et al. 2009), which uses GUROBI (2012) library to solve the corresponding ILP problems. To exemplify the use of viable taxon selection and its generalization, we applied these problems to a real case study: Caribbean coral reef community. The combination of molecular evolution and predator-prey interactions as well as the use of flexible ILP method makes PDA software a good complementary to the existing conservation prioritization tools.

CHAPTER 2

Consequences of Common Topological Rearrangements for Partition Trees in Phylogenomic Inference

In phylogenomic analysis the collection of trees with identical score (maximum likelihood or parsimony score) may hamper tree search algorithms. Such collections are coined phylogenetic terraces. For sparse supermatrices with a lot of missing data the number of terraces and the number of trees on the terraces can be very large. If terraces are not taken into account, a lot of computation time might be unnecessarily spent to evaluate many trees that in fact have identical score. To save computation time during the tree search it is worthwhile to quickly identify such cases.

The score of a species tree is the sum of scores for all the so-called induced partition trees. Therefore, if the topological rearrangement applied to a species tree does not change the induced partition trees, the score of these partition trees is unchanged.

Here, we provide the conditions under which the three most widely used topological rearrangements (Nearest Neighbouring Interchange, Subtree Pruning and Regrafting, and Tree Bisection and Reconnection) change the topologies of induced partition trees. During tree search these conditions allow us to quickly identify whether we can save computation time on the evaluation of newly encountered trees. We also introduce the concept of partial terraces and demonstrate that they occur more frequently than the original “full” terrace. Hence, partial terrace is the more important factor of timesaving compared to full terrace.

Therefore, taking into account the above conditions and the partial terrace concept will help to speed up the tree search in phylogenomic inference.

2.1 Introduction

In phylogenomics one aims to reconstruct a phylogenetic *species* tree from multiple genes. One popular approach is to infer the trees from the concatenated gene alignment, the so-called supermatrix (de Queiroz and Gatesy 2007; Sanderson, et al. 1998). Here, if a gene sequence is not available for some taxon, it is represented by the sequence of unknown characters and is referred to as missing data. Several studies (Hedtke, et al. 2013; Nyakatura and Bininda-Emonds 2012; Pyron, et al. 2011; Pyron and Wiens 2011; Springer, et al. 2012; van der Linde, et al. 2010) use quite sparse supermatrices in their analysis and the percentage of missing data sometimes constitutes up to 95% (Peters, et al. 2011).

Recently it has been shown that missing data can hamper the tree search via existence of phylogenetic terraces (Sanderson, et al. 2011), a collection of trees with exactly the same likelihood or parsimony score. Terraces occur in the analysis with *partitioned data*, i.e. when distinct blocks of a supermatrix are treated differently (for example, when each gene corresponding to one block evolves under its own evolutionary model). Two trees are said to belong to one terrace if the collections of their *induced partition trees* are exactly the same. Here, the induced partition tree is obtained by pruning the taxa on species tree, which have no sequence for the corresponding partition block.

Since the number of trees on one terrace can be quite large (Sanderson, et al. 2011), accounting for terraces in tree search algorithms can potentially save a lot of computation time. During the tree search one explores the tree space by moving from one candidate tree to another by means of topological rearrangements. If the topological rearrangement does not change any of the induced partition trees, then the two trees belong to the same terrace and a recomputation of objective function (maximum likelihood or maximum parsimony) used in the tree search is not necessary in order to evaluate a new tree.

Here, we first specify the conditions under which the topological rearrangements applied to the species tree change the corresponding induced partition trees. Using these conditions one can quickly identify whether it is necessary to recompute the objective function for a given partition or not as a consequence of one of the three widely used rearrangements: Nearest Neighbour Interchange (NNI), Subtree Pruning and Regrafting (SPR) and Tree Bisection and Reconnection (TBR) (Felsenstein 2004).

We further generalize the concept of terrace to *partial terrace*, which is even more useful in practical phylogenetic analysis. We analyse several published alignments by examining NNI neighbourhoods of random trees and trees encountered during the tree search using IQ-TREE (Nguyen, et al. 2015). We show that for large number of taxa partial terraces are mainly determined by the missing data and less dependent on the actual tree topology analysed. By taking into account partial terraces it will be possible to speed up the tree search algorithms even in the absence of terraces.

The outline of the chapter is the following. We first introduce the notations and then discuss the important features of NNI, SPR and TBR. Next, we specify the conditions when these topological rearrangements do not change the topology of induced partition trees. We further elucidate why such conditions are helpful even in the absence of terraces and define the concept of partial terrace. We analyse several published alignments to point out that partial terraces do occur in practice. Finally, we discuss the additional practical advantages of using induced partition trees in the maximum likelihood framework.

2.2 Background

2.2.1 Basic definitions and Notations

In this section we provide basic definitions and notations used throughout the chapter. For a complete overview see chapters 2, 3 and 6 in Semple and Steel (2003).

Definition 2.1: Let X be a taxon set. A *phylogenetic tree* T of X is a leaf-labelled tree with a bijection map from X into the set of leaves of T.

In the following we work only with bifurcating phylogenetic trees, i.e. all internal nodes have exactly three adjacent edges.

Definition 2.2: A *split*, denoted by $A|B$, is a bipartition of X into two non-empty, non-overlapping sets A and B, where $A \cup B = X$.

Note that $A|B$ and $B|A$ are equivalent. Every edge of T is associated with a split. When cutting an edge e of T we obtain two subtrees with leaf labels X_1 and X_2, then a split corresponding to e is defined as $X_1|X_2$. We denote this with $e = X_1|X_2$. We denote by $\Sigma(T)$ a collection of all splits corresponding to edges of T.

The *symmetric difference* of two sets A and B, denoted $A\Delta B$, is given by $(A\backslash B) \cup (B\backslash A)$, or the union of taxa present in A but not B, and vice versa.

Definition 2.3: Let T_1 and T_2 be the two leaf-labelled trees with the same label set X and $\Sigma(T_1)$ and $\Sigma(T_2)$ be the collections of splits of T_1 and T_2, respectively. Then the *Robinson–Foulds (RF) distance (Robinson and Foulds 1981)* between T_1 and T_2 is equal to $|\Sigma(T_1)\Delta\Sigma(T_2)|$.

If for two trees the RF-distance between them is 0, then they have the same collection of splits, and from Splits-Equivalence theorem (Semple and Steel 2003; p.43) the trees are *equivalent*.

Definition 2.4. Let Y be a subset of X. An *induced subtree* of T, denoted by $T|Y$, is a leaf-labelled tree with the following collection of splits

$$\Sigma(T|Y) = \{A \cap Y|B \cap Y : A|B \in \Sigma(T) \text{ and } A \cap Y \neq \emptyset, B \cap Y \neq \emptyset\}.$$

For a species tree T and a given partition with taxon set Y a *partition tree* is an induced subtree $T|Y$.

2.2.2 Topological rearrangement operations

In this section we introduce the topological rearrangements on trees commonly used in phylogenetic inference.

- *Nearest Neighbour Interchange (NNI)*

The simplest possible operation that changes only one split on a tree is an *NNI*. It can only be applied to interior edges of the tree, since it requires the so-called quartet structure with an interior edge being the central edge of this structure (Fig. 2.1).

Let e be an interior edge of T and e_1, e_2, e_3, e_4 its four incident edges with A, B, C, D being the taxon sets leading from them respectively (Fig. 2.1). An NNI on T around e is obtained by exchanging the subtrees below two non-incident edges from e_1, e_2, e_3, e_4. We denote a new tree by T_{NNI}.

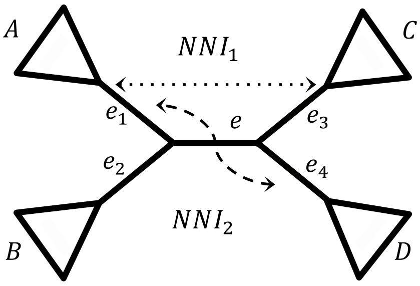


Figure 2.1. Visualization of NNI. Species tree T and the two NNIs around central edge e. NNI_1 is obtained by exchanging subtrees below edges e_1 and e_3, while NNI_2 by exchanging subtrees e_1 and e_4.

For each interior edge e there are two possible NNIs obtained by exchanging a subtree below e_1 with a subtree below either e_3 or e_4 (note, that this is equivalent to swapping the subtree below e_2 with either e_4 or e_3, respectively).

Let us assume that the NNI is applied to edge e by swapping e_1 and e_3. The splits corresponding to e_1, e_2, e_3 and e_4 stay unchanged

$$\begin{aligned} e_1 &= A|B \cup C \cup D, \\ e_2 &= B|A \cup C \cup D, \\ e_3 &= C|A \cup B \cup D, \\ e_4 &= D|A \cup B \cup C. \end{aligned}$$

This also holds true for the edges belonging to subtrees below e_1, e_2, e_3 and e_4 (Fig. 2.1). Here, if $e_1 = A|B \cup C \cup D$, *the subtree below* e_1 is a subtree with a leaf set A and not the union of sets. Hence, the splits corresponding to e_1, e_2, e_3, e_4 and edges below them will be shared by T and T_{NNI}. The central edge e in terms of splits will be changed by the NNI from $A \cup B|C \cup D$ to $e^{NNI} = A \cup D|B \cup C$.

It follows from above that T and T_{NNI} are different only in one split, i.e.

$$\Sigma(T)\,\Delta\,\Sigma(T_{NNI}) = \{A \cup B|C \cup D, A \cup D|B \cup C\}$$

and the RF-distance between T and T_{NNI} is 2.

- *Subtree Pruning and Regrafting (SPR)*

We now discuss *SPR*, a more general topological rearrangement that changes one or more splits of the tree.

An SPR on T is represented in Figure 2.2 (see also Hordijk and Gascuel (2005)). A new tree T_{SPR} is obtained from T by pruning the subtree below edge a and regrafting it onto edge b_n (we sometimes refer to such SPR as n-SPR). Note, that n is at least 3 and if $n = 3$, an SPR is equivalent to an NNI obtained by swapping subtrees belonging to edges a and b_2. Let $A, B_1, \dots, B_n$ denote the corresponding taxon sets leading from $a, b_1, \dots, b_n$ respectively (Fig. 2.2).

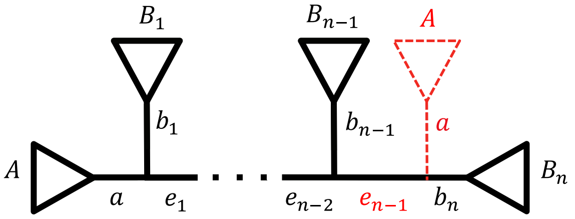


Figure 2.2. Visualization of SPR. A new tree T_{SPR} is obtained by pruning the subtree A below edge a and regrafting it onto edge b_n (dashed red subtree). After SPR is applied, edges b_1 and e_1 are joined and edge b_n is split into e_{n-1} and b_n.

An SPR on T changes only the splits of the path edges, namely: for $\forall x \in \{1, \dots, n-2\}$

$$e_x = A \cup B_1 \cup \dots \cup B_x|B_{x+1} \cup \dots \cup B_n$$

is changed to

$$e_x^{SPR} = B_1 \cup \dots \cup B_x|B_{x+1} \cup \dots \cup B_n \cup A,$$

where e_x^{SPR} is an edge that corresponds to e_x on a new tree T_{SPR}. Also a new edge appears $e_{n-1} = B_1 \cup \dots \cup B_{n-1}|A \cup B_n$. The rest of splits remain unchanged and are shared by both trees. Hence, for T and T_{SPR} the symmetric difference $\Sigma(T)\,\Delta\,\Sigma(T_{SPR})$ consists of the following splits

$$\begin{array}{ll} A \cup B_1 \cup \dots \cup B_x|B_{x+1} \cup \dots \cup B_n, & \forall x \in \{1, \dots, n-2\}, \\ B_1 \cup \dots \cup B_x|B_{x+1} \cup \dots \cup B_n \cup A, & \forall x \in \{2, \dots, n-1\}. \end{array}$$

The RF-distance between T and T_{SPR} is equal to $2(n-2)$.

- *Tree Bisection and Reconnection (TBR)*

The last topological rearrangement we are going to discuss is the *TBR*. A TBR on T is shown in Figure 2.3, where a new tree T_{TBR} is obtained from T (Fig. 2.3, in black) by cutting edge e and reconnecting edges b_n and c_m with a new edge e^{TBR} (Fig. 2.3, red dashed line). Note that n or m must be greater than 2. W.l.o.g. assume that $m \leq n$. If $n = 3$ and $m = 2$, then a TBR corresponds to an NNI around edge e_1 by swapping subtrees below e and b_2. If $n > 3$ and $m = 2$, then a TBR corresponds to an SPR.

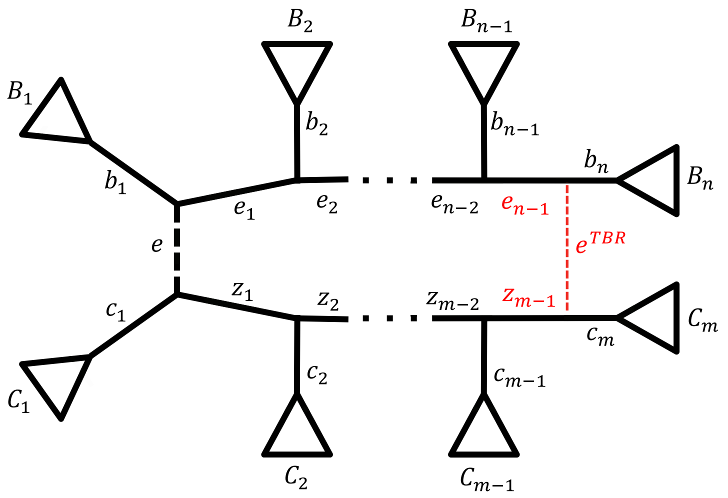


Figure 2.3. Visualization of TBR. To obtain T_{TBR} species tree T is cut into two parts (by removing edge e), which are further reconnected by joining edges b_n and c_m with e^{TBR}. Edge b_n is split into e_{n-1} and b_n, while c_m is split into z_{m-1} and c_m. Edges b_1 and e_1 are joined, as well as c_1 and z_1.

TBR only changes the splits corresponding to all path edges (e_i and z_j), but e. Namely:

$$e = B_1 \cup \dots \cup B_n | C_1 \cup \dots \cup C_m = e^{TBR},$$

while for $\forall x \in \{1, \dots, n-2\}$

$$e_x = C_1 \cup \dots \cup C_m \cup B_1 \cup \dots \cup B_x | B_{x+1} \cup \dots \cup B_n$$

is changed to

$$e_x^{TBR} = B_1 \cup \dots \cup B_x | B_{x+1} \cup \dots \cup B_n \cup C_1 \cup \dots \cup C_m$$

and for $\forall y \in \{1, \dots, m-2\}$

$$z_y = B_1 \cup \dots \cup B_n \cup C_1 \cup \dots \cup C_y | C_{y+1} \cup \dots \cup C_m$$

is changed to

$$z_y^{TBR} = C_1 \cup \dots \cup C_y | C_{y+1} \cup \dots \cup C_m \cup B_1 \cup \dots \cup B_n.$$

Also two new edges appear

$$e_{n-1} = B_1 \cup \dots \cup B_{n-1} | B_n \cup C_1 \cup \dots \cup C_m,$$

$$z_{m-1} = C_1 \cup \dots \cup C_{m-1} | C_m \cup B_1 \cup \dots \cup B_n.$$

The remaining splits stay unchanged. Hence, for T and T_{TBR} the symmetric difference $\Sigma(T) \,\Delta\, \Sigma(T_{TBR})$ is a set consisting of the following splits

$$\begin{array}{ll} C_1 \cup \dots \cup C_m \cup B_1 \cup \dots \cup B_x | B_{x+1} \cup \dots \cup B_n, & \forall x \in \{1, \dots, n-2\}, \\ B_1 \cup \dots \cup B_x | B_{x+1} \cup \dots \cup B_n \cup C_1 \cup \dots \cup C_m, & \forall x \in \{2, \dots, n-1\}, \\ B_1 \cup \dots \cup B_n \cup C_1 \cup \dots \cup C_y | C_{y+1} \cup \dots \cup C_m, & \forall y \in \{1, \dots, m-2\}, \\ C_1 \cup \dots \cup C_y | C_{y+1} \cup \dots \cup C_m \cup B_1 \cup \dots \cup B_n, & \forall y \in \{2, \dots, m-1\}. \end{array}$$

Therefore, the RF-distance between T and T_{TBR} is $2(n + m - 4)$.

2.3 Consequences of topological rearrangements applied to a species tree

In the following we discuss how the topological rearrangement of the species tree T influences the topology of the partition trees and start with the simplest operation, an NNI.

Proposition 1. Let e be an interior edge and e_1, e_2, e_3, e_4 the four edges adjacent to e with A, B, C, D being the taxon sets leading from the corresponding edges (Fig. 2.1). Let a new tree T_{NNI} be obtained from T via NNI. For a partition with a taxon set Y the topologies of $T|Y$ and $T_{NNI}|Y$ are different iff Y has at least one representative taxon in each subset A, B, C, D.

Proof. W.l.o.g. assume that T_{NNI} is obtained from T via swapping of subtrees below e_1 and e_3. Then $\Sigma(T)\,\Delta\,\Sigma(T_{NNI}) = \{A \cup B|C \cup D, A \cup D|B \cup C\}$ and as a consequence for corresponding partition trees we have

$$\Sigma(T|Y)\,\Delta\,\Sigma(T_{NNI}|Y) = \{(A \cup B) \cap Y|(C \cup D) \cap Y, (A \cup D) \cap Y|(B \cup C) \cap Y\}.$$

It is easy to show, that if at least one set from $A \cap Y, B \cap Y, C \cap Y, D \cap Y$ were empty, then both splits $(A \cup B) \cap Y|(C \cup D) \cap Y$ and $(A \cup D) \cap Y|(B \cup C) \cap Y$ coincide with splits shared by $T|Y$ and $T_{NNI}|Y$ (for example, see Fig. 2.4). Hence, $\Sigma(T|Y)\,\Delta\,\Sigma(T_{NNI}|Y) = \emptyset$ and the RF-distance between these trees would be 0. Therefore, for $T|Y$ and $T_{NNI}|Y$ to have different topologies, all $A \cap Y, B \cap Y, C \cap Y, D \cap Y$ must be non-empty, meaning that Y has to have at least one representative in each subset A, B, C, D. ■

In simple words, if some intersections of A, B, C, D with Y are empty, then a partition tree does not have a corresponding quartet structure for the NNI to be applied to and edge e loses its centrality or interior feature (see for example Fig. 2.4). When this happens, the topology of the partition tree $T|Y$ is not affected by the NNI applied to e on the species tree T.

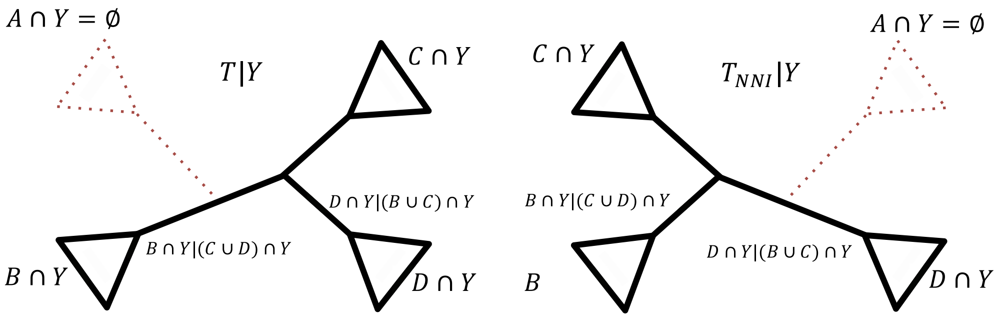


Figure 2.4. An example when an NNI on T does not change the topology of $T|Y$. Solid lines correspond to two induced partition trees before ($T|Y$) and after ($T_{NNI}|Y$) the NNI was applied to edge e on T by swapping the subtrees below e_1 and e_3 (Fig. 2.1). In this case Y does not have a representative in A (i.e. $A \cap Y = \emptyset$) therefore, $(A \cup B) \cap Y|(C \cup D) \cap Y = B \cap Y|(C \cup D) \cap Y$ and $(A \cup D) \cap Y|(B \cup C) \cap Y = D \cap Y|(B \cup C) \cap Y$. Since the splits $B \cap Y|(C \cup D) \cap Y$ and $D \cap Y|(B \cup C) \cap Y$ are shared by $T|Y$ and $T_{NNI}|Y$, then $\Sigma(T|Y)\,\Delta\,\Sigma(T_{NNI}|Y) = \emptyset$ and RF-distance between $T|Y$ and $T_{NNI}|Y$ is 0.

We next specify the condition when an SPR changes the topology of partition tree.

Proposition 2. Let tree T be in the form shown in Figure 2.2 and a new tree T_{SPR} is obtained with SPR by pruning subtree below edge a and regrafting it onto b_n. Then for a partition with a taxon set Y the following is true

(i) the topologies of $T|Y$ and $T_{SPR}|Y$ are different, if Y has at least one representative in A and in at least another three subsets from $B_1, B_2, \dots, B_n$;

(ii) this SPR will correspond to an SPR on $T|Y$ obtained by pruning the subtree below edge with a split $A \cap Y|(B_1 \cup \dots \cup B_n) \cap Y$ and regrafting it onto edge with split $B_k \cap Y|(\cup_{i\in\{1,..,n\}\setminus k} B_i \cup A) \cap Y$, where $k = \max_{1\le i\le n}\{\, i \mid B_i \cap Y \neq \emptyset\}$.

Proof. (i) The symmetric difference $\Sigma(T) \,\Delta\, \Sigma(T_{SPR})$ consists of the following splits

$$A \cup B_1 \cup \dots \cup B_x | B_{x+1} \cup \dots \cup B_n, \quad \forall x \in \{1, \dots, n-2\},$$
$$B_1 \cup \dots \cup B_x | B_{x+1} \cup \dots \cup B_n \cup A, \quad \forall x \in \{2, \dots, n-1\}.$$

As a consequence for the induced partition trees $T|Y$ and $T_{SPR}|Y$ the symmetric difference of $\Sigma(T|Y)$ and $\Sigma(T_{SPR}|Y)$ consists of

$$(A \cup B_1 \cup \dots \cup B_x) \cap Y | (B_{x+1} \cup \dots \cup B_n) \cap Y, \quad \forall x \in \{1, \dots, n-2\},$$
$$(B_1 \cup \dots \cup B_x) \cap Y | (B_{x+1} \cup \dots \cup B_n \cup A) \cap Y, \quad \forall x \in \{2, \dots, n-1\}.$$

It is easy to see, that if $A \cap Y = \emptyset$, then all these splits would be shared by both partition trees, i.e. $\Sigma(T|Y) \,\Delta\, \Sigma(T_{SPR}|Y) = \emptyset$ and the RF-distance between $T|Y$ and $T_{SPR}|Y$ would be 0. Therefore, Y must have at least one representative in A.

For $T|Y$ and $T_{SPR}|Y$ to have different topologies an SPR on T should correspond to at least an NNI on $T|Y$. Hence, $T|Y$ must have a corresponding quartet structure and together with A at least another three subsets from $B_1, B_2, \dots, B_n$ should have at least one representative in Y. W.l.o.g. assume that together with A also B_m, B_h, B_k ($1 \le m < h < k \le n$) have at least one representative in Y while $B_j \cap Y = \emptyset \;\forall j \in \{1, \dots, n\} \setminus \{m, h, k\}$ (for example see Fig. 2.5). Then

$$\Sigma(T|Y) \,\Delta\, \Sigma(T_{SPR}|Y) = \{(A \cup B_m) \cap Y|(B_h \cup B_k) \cap Y, (B_m \cup B_h) \cap Y|(B_k \cup A) \cap Y\}.$$

Thus the RF-distance between $T|Y$ and $T_{SPR}|Y$ is 2.

(ii) Let $I = \{i_1, .., i_k\}$ be the set of all indices, such that $\forall i \in I: B_i \cap Y \neq \emptyset$ and let $1 \le i_1 < \dots < i_k \le n$.

For edge $a = A|B_1 \cup \dots \cup B_n$ its corresponding split on the partition tree $T|Y$ is equal to

$$A \cap Y|(B_1 \cup \dots \cup B_n) \cap Y = A \cap Y| \cup_{i\in I} (B_i \cap Y).$$

Similarly for e_{i_k-1} its corresponding split on $T|Y$

$$\left(A \cup B_1 \cup \dots \cup B_{i_k-1}\right) \cap Y \middle| \left(B_{i_k} \cup \dots \cup B_n\right) \cap Y = \left(\cup_{i\in I\setminus i_k} B_i \cup A\right) \cap Y \middle| B_{i_k} \cap Y,$$

and for $e^{SPR}_{i_k-1}$ its corresponding split on the partition tree $T_{SPR}|Y$

$$\left(B_1 \cup \dots \cup B_{i_k-1}\right) \cap Y \middle| \left(B_{i_k} \cup \dots \cup B_n \cup A\right) \cap Y = \left(\cup_{i\in I\setminus i_k} B_i\right) \cap Y \middle| \left(B_{i_k} \cup A\right) \cap Y.$$

The above means that an edge on $T|Y$ with split $\left(\cup_{i\in I\setminus i_k} B_i \cup A\right) \cap Y | B_{i_k} \cap Y$ was divided by an edge with split $A \cap Y| \cup_{i\in I} (B_i \cap Y)$ in two edges (see also Fig. 2.5, where $I = \{i_1, i_2, i_3\}$). Therefore, regrafting onto edge b_n on T corresponds to regrafting onto edge with a split $B_{i_k} \cap Y|(\cup_{i\in\{1,..,n\}\setminus i_k} B_i \cup A) \cap Y$ on partition tree $T|Y$. And since $1 \leq i_1 < \cdots < i_k \leq n$, then $i_k = \max_{1\leq i\leq n}\{\, i \mid B_i \cap Y \neq \emptyset\}$.

■

In other words Proposition 2 states that an SPR on T changes the topology of $T|Y$ if the structure of T from Fig. 2.2 corresponds to at least a quartet structure on $T|Y$ (for example, Fig. 2.5). In this case n-SPR on T is an 3-SPR (or NNI) on $T|Y$.

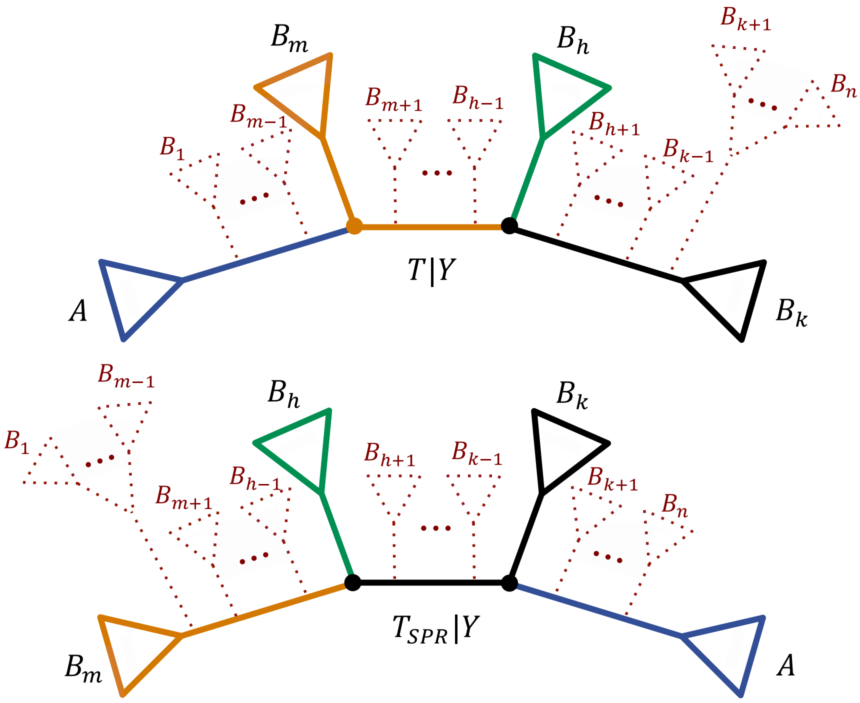


Figure 2.5. An example when n-SPR on T is a 3-SPR (or NNI) on $T|Y$. There are two induced partition trees (solid lines): before ($T|Y$) and after ($T_{SPR}|Y$) an SPR was applied on T by pruning the subtree below edge a and regrafting it onto b_n (Fig. 2.2). The three dots denote all the subtrees between the corresponding pair of subtrees on the species trees T and T_{SPR}. Here, only A, B_m, B_h and B_k have at least one representative in Y and $\forall j \in \{1, \dots, n\} \setminus \{m, h, k\}$: B_j have no taxa in common with Y.

We now discuss TBR and the topological change of a partition tree as a consequence of TBR on species tree.

Proposition 3. Let tree T be in the form shown in Figure 2.3 and a new tree T_{TBR} is obtained by cutting edge e and reconnecting b_n and c_m with a new edge. Then for a partition with a taxon set Y the following is true

(i) the topologies of $T|Y$ and $T_{TBR}|Y$ are different, if at least one of the following conditions is satisfied

– Y has at least one representative in at least one subset from $B_1, B_2, \dots, B_n$ and in at least another three subsets from $C_1, C_2, \dots, C_m$,

– Y has at least one representative in at least one subset from $C_1, C_2, \dots, C_m$ and in at least another three subsets from $B_1, B_2, \dots, B_n$;

(ii) this TBR will correspond to a TBR on $T|Y$ obtained by cutting the edge with split $(B_1 \cup \dots \cup B_n) \cap Y|(C_1 \cup \dots \cup C_m) \cap Y$ and reconnecting edges with splits $B_k \cap Y|(\cup_{i\in\{1,..,n\}\setminus k} B_i \cup C_1 \cup \dots \cup C_m) \cap Y$ and $C_h \cap Y|(\cup_{j\in\{1,..,m\}\setminus h} C_j \cup B_1 \cup \dots \cup B_n) \cap Y$, where $k = \max_{1\le i\le n}\{\, i \mid B_i \cap Y \neq \emptyset\}$ and $h = \max_{1\le j\le m}\{\, j \mid C_j \cap Y \neq \emptyset\}$.

Proof. (i) The symmetric difference $\Sigma(T)\,\Delta\,\Sigma(T_{TBR})$ consists of the following splits

$$\begin{aligned}
&C_1 \cup \dots \cup C_m \cup B_1 \cup \dots \cup B_x|B_{x+1} \cup \dots \cup B_n, && \forall x \in \{1, \dots, n-2\},\\
&B_1 \cup \dots \cup B_x|B_{x+1} \cup \dots \cup B_n \cup C_1 \cup \dots \cup C_m, && \forall x \in \{2, \dots, n-1\},\\
&B_1 \cup \dots \cup B_n \cup C_1 \cup \dots \cup C_y|C_{y+1} \cup \dots \cup C_m, && \forall y \in \{1, \dots, m-2\},\\
&C_1 \cup \dots \cup C_y|C_{y+1} \cup \dots \cup C_m \cup B_1 \cup \dots \cup B_n, && \forall y \in \{2, \dots, m-1\}.
\end{aligned}$$

As a consequence the symmetric difference $\Sigma(T|Y)\,\Delta\,\Sigma(T_{TBR}|Y)$ consists of

$$\begin{aligned}
&(C_1 \cup \dots \cup C_m \cup B_1 \cup \dots \cup B_x) \cap Y|(B_{x+1} \cup \dots \cup B_n) \cap Y, && \forall x \in \{1, \dots, n-2\},\\
&(B_1 \cup \dots \cup B_x) \cap Y|(B_{x+1} \cup \dots \cup B_n \cup C_1 \cup \dots \cup C_m) \cap Y, && \forall x \in \{2, \dots, n-1\},\\
&(B_1 \cup \dots \cup B_n \cup C_1 \cup \dots \cup C_y) \cap Y|(C_{y+1} \cup \dots \cup C_m) \cap Y, && \forall y \in \{1, \dots, m-2\},\\
&(C_1 \cup \dots \cup C_y) \cap Y|(C_{y+1} \cup \dots \cup C_m \cup B_1 \cup \dots \cup B_n) \cap Y, && \forall y \in \{2, \dots, m-1\}.
\end{aligned}$$

It is easy to see, that if $\forall i \in \{1, .., n\}$: $B_i \cap Y = \emptyset$, then all these splits would be shared by both partition trees, i.e. $\Sigma(T|Y)\,\Delta\,\Sigma(T_{TBR}|Y) = \emptyset$ and the RF-distance between $T|Y$ and $T_{TBR}|Y$ would be 0. Therefore, Y must have at least one representative in at least one from $B_1, B_2, \dots, B_n$. Similarly, Y must have at least one representative in at least one from $C_1, C_2, \dots, C_m$.

W.l.o.g. assume that $B_k \cap Y \neq \emptyset$ and $C_h \cap Y \neq \emptyset$, where $1 \le k \le n$ and $1 \le h \le m$.

Partition trees $T|Y$ and $T_{TBR}|Y$ will have different topologies, if a TBR on T corresponds to at least an NNI on $T|Y$. Hence, the partition tree $T|Y$ must have a corresponding quartet structure and together with B_k and C_h at least another two subsets from the remaining B_i and C_j should have at least one representative in Y.

W.l.o.g. assume that together with B_k and C_h also C_p, C_q $(1 \le p < q < h \le m)$ have at least one representative in Y (Fig. 2.6a). Then it is easy to show that

$$\Sigma(T|Y)\, \Delta\, \Sigma(T_{TBR}|Y) = \{(B_k \cup C_p) \cap Y|(C_q \cup C_h) \cap Y, (C_p \cup C_q) \cap Y|(C_h \cup B_k) \cap Y\}$$

and RF-distance between $T|Y$ and $T_{TBR}|Y$ is 2.

Similarly, one can show that if together with B_k and C_h also B_p, B_q $(1 \le p < q < k \le n)$ have at least one representative in Y, then RF-distance between $T|Y$ and $T_{TBR}|Y$ is also 2.

In contrast, if Y has at least one representative in B_k, C_h and also in B_p, C_q $(1 \le p < k \le n$ and $1 \le q < h \le m)$, then $\Sigma(T|Y)\, \Delta\, \Sigma(T_{TBR}|Y) = \emptyset$ and RF-distance is 0 (Fig. 2.6b).

(ii) Let $I = \{i_1, .., i_k\}$ be the set of all indices, such that $\forall i \in I$: $B_i \cap Y \neq \emptyset$ and let $1 \le i_1 < \cdots < i_k \le n$. Similarly let $J = \{j_1, .., j_h\}$ be the set of all indices, such that $\forall j \in J$: $C_j \cap Y \neq \emptyset$ and let $1 \le j_1 < \cdots < j_h \le m$. Then for edge

$$e = B_1 \cup \ldots \cup B_n | C_1 \cup \ldots \cup C_m$$

the corresponding split on $T|Y$ is

$$\cup_{i \in I} B_i \cap Y \mid \cup_{j \in J} C_j \cap Y.$$

For edge

$$e_{i_k - 1} = C_1 \cup \ldots \cup C_m \cup B_1 \cup \ldots \cup B_{i_k - 1} | B_{i_k} \cup \ldots \cup B_n$$

its corresponding split on tree $T|Y$ is

$$\left(C_1 \cup \ldots \cup C_m \cup B_1 \cup \ldots \cup B_{i_k-1}\right) \cap Y \big| \left(B_{i_k} \cup \ldots \cup B_n\right) \cap Y =$$
$$= \left(\cup_{j \in J} C_j \cap Y\right) \cup \left(\cup_{i \in I \setminus i_k} B_i \cap Y\right) \big| B_{i_k} \cap Y.$$

Similarly for the corresponding edge on T_{TBR} $e_{i_k-1}^{TBR} = B_1 \cup \ldots \cup B_{i_k-1} | B_{i_k} \cup \ldots \cup B_n \cup C_1 \cup \ldots \cup C_m$ its split on $T_{TBR}|Y$ is

$$\left(B_1 \cup \ldots \cup B_{i_k-1}\right) \cap Y \big| \left(B_{i_k} \cup \ldots \cup B_n \cup C_1 \cup \ldots \cup C_m\right) \cap Y =$$
$$= \left(\cup_{i \in I \setminus i_k} B_i \cap Y\right) \big| \left(B_{i_k} \cap Y\right) \cup \left(\cup_{j \in J} C_j \cap Y\right).$$

For edges

$$z_{j_h-1} = B_1 \cup \ldots \cup B_n \cup C_1 \cup \ldots \cup C_{j_h-1} | C_{j_h} \cup \ldots \cup C_m$$

and

$$z_{j_h-1}^{TBR} = C_1 \cup \ldots \cup C_{j_h-1} | C_{j_h} \cup \ldots \cup C_m \cup B_1 \cup \ldots \cup B_n$$

their corresponding splits on $T|Y$ and $T_{TBR}|Y$ are

$$(\cup_{i \in I} B_i \cap Y) \cup \left(\cup_{j \in J \setminus j_h} C_j \cap Y\right) \big| C_{j_h} \cap Y$$

and

$$\big(\cup_{j\in J\setminus j_h} C_j \cap Y\big)|\big(C_{j_h} \cap Y\big) \cup (\cup_{i\in I} B_i \cap Y)$$

respectively. The above means that edges on $T|Y$ with corresponding splits

$$\big(\cup_{j\in J} C_j \cap Y\big) \cup \big(\cup_{i\in I\setminus i_k} B_i \cap Y\big)|B_{i_k} \cap Y$$

and

$$(\cup_{i\in I} B_i \cap Y) \cup \big(\cup_{j\in J\setminus j_h} C_j \cap Y\big)|C_{j_h} \cap Y$$

were reconnected on $T_{TBR}|Y$ by $\cup_{i\in I} B_i \cap Y \mid \cup_{j\in J} C_j \cap Y$. Since $1 \le i_1 < \cdots < i_k \le n$ and $1 \le j_1 < \cdots < j_h \le m$, then $i_k = \max_{1\le i\le n}\{\, i \mid B_i \cap Y \neq \emptyset\}$ and $j_h = \max_{1\le j\le m}\{\, j \mid C_j \cap Y \neq \emptyset\}$. ■

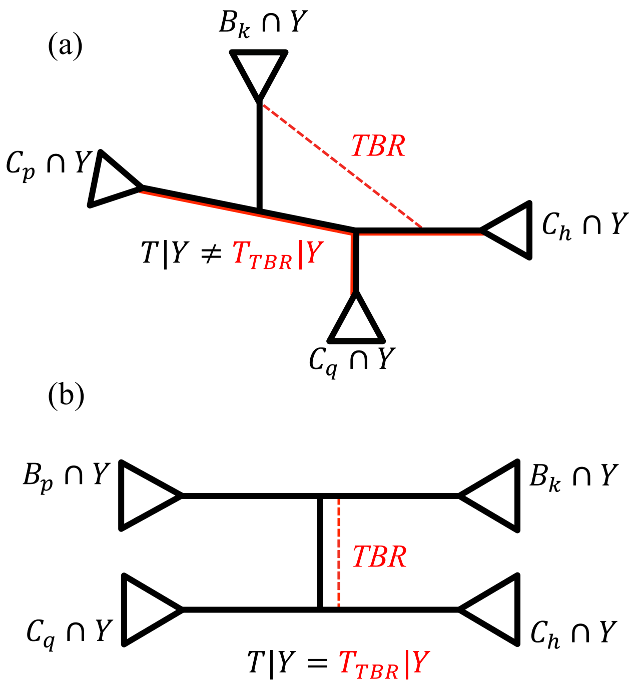


Figure 2.6. Examples of corresponding TBRs on partition trees. Two partition trees with topologies before ($T|Y$, in black) and after ($T_{TBR}|Y$, in red) the TBR was applied to the species tree. For simplicity we do not show the pruned subtrees for which $B_i \cap Y = \emptyset$ and $C_j \cap Y = \emptyset$. (a) The simplest case when the TBR changes the topology of partition tree. In this case a TBR on species tree corresponds to an NNI on partition tree. (b) An example case, when the topology of partition tree remains unchanged after TBR.

2.4 Partial Terraces

2.4.1 Definition of partial terraces

In this section we discuss *partial terraces* that generalize the terrace concept (Sanderson, et al. 2011), which we call *full terrace* for clarity. When comparing the two trees in a partitioned framework we compare the sets of their induced partition trees. If the sets are identical, then the two trees belong to one full terrace. Sanderson, et al. (2011) showed that the number of trees on one full terrace can be quite large. Large full terraces pose a problem in phylogenetic inference, since they may abort tree search prematurely or even if an optimal tree has been found, this tree is by no means unique. To reduce this problem, it is possible to reduce the terrace size by, for example, choosing a different partition scheme (Sanderson, et al. 2015) or by excluding some taxa from the analysis.

Now, if two species trees T_1 and T_2 share only a subset of identical induced partition trees, then we say that they belong to the same *partial terrace*. The log-likelihoods and parsimony scores of identical partition trees $T_1|Y_i$ and $T_2|Y_i$ are the same. Obviously, partial terraces occur more frequently than full terraces (see below). Large partial terraces can be still problematic for tree search algorithms. On the other hand, partial terraces provide the potential to reduce computation time.

2.4.2 Occurrence of partial terraces in real data

In this section we evaluate how often partial terraces occur in real alignments. By no means we intend to make a full exploration of potential computing time that may be saved since the performance of the particular software will depend on the data structures and particular implementation used for the tree space exploration.

To elucidate the occurrence of partial terraces and full terraces we analysed 7 recently published alignments (Table 2.1). Alignments have different number of taxa ranging from 69 to 404 taxa. The number of partitions (here, genes) varies from 11 to 79.

Type and ID	No. Species	No. Genes	Missing Data	Source
DNA1	128	32	30%	Stamatakis and Alachiotis (2010)
DNA2	237	74	72%	Nyakatura and Bininda-Emonds (2012)
DNA3	372	79	66%	Springer, et al. (2012)
DNA4	404	11	60%	Stamatakis and Alachiotis (2010)
AA1	69	31	35%	
AA2	70	35	34%	De Queiroz, et al. (1995)
AA3	72	51	35%	

Table 2.1. The alignments used to study the occurrence of partial terraces during the tree search.

For each alignment we performed a maximum likelihood tree search using IQ-TREE (Nguyen, et al. 2015) under EUL partition model assuming the GTR+Γ (Lanave, et al. 1984; Yang 1994) and the LG+Γ (Le and Gascuel 2008; Yang 1994) models for all partitions in the DNA and the AA alignments, respectively. We collected all the intermediate trees encountered during the search. For each intermediate tree T we explored all trees T_{NNI} in its NNI neighbourhood. We examined partial terraces of each T_{NNI} and T by computing how many induced partition trees are shared between them.

Apart from intermediate trees collected during the tree search, we also analysed NNI neighbourhoods for 1000 random Yule-Harding (YH) trees (Harding 1971) for each tested alignment.

We defined 12 bins based on the percentage of shared induced partition trees between T and T_{NNI} (Table 2.2) and counted how many T_{NNI} trees fall into each bin.

Name	**Percentage of shared partition trees** **out of the total number of partition trees**
no Partial Terrace (PT)	= 0%, the topologies of all partition trees are pairwise different between T and T_{NNI}
PT1 PT2 PT3 ... PT9 PT10	(0%, 10%] (10%, 20%] (20%, 30%] ... (80%, 90%] (90%, 100%)
Full Terrace	=100%, T and T_{NNI} belong to one terrace

Table 2.2. Partial Terrace bins based on the percentage of the shared partition trees between T and T_{NNI}.

Table 2.3 shows the mean percentage of T_{NNI} trees that fall into corresponding bin for the intermediate trees. Figure 7 displays the boxplots for the first three alignments from Table 2.1 either for the IQ-TREE search trees (left column) or the random YH trees (right column) (see Appendix A Fig. A1 - A4 for the remaining alignments).

(%)	no PT	PT1	PT2	PT3	PT4	PT5	PT6	PT7	PT8	PT9	PT10	Full Terrace
DNA1	7.14	12.97	4.85	1.80	3.55	32.59	37.11	0	0	0	0	0
DNA2	0	0	0.02	0.63	1.82	5.07	11.31	8.69	18.19	10.77	41.75	1.75
DNA3	0	2.75	5.38	9.22	10.06	6.32	4.34	1.33	0.23	6.38	50.36	3.63
DNA4	0.35	0.26	1.88	4.06	5.20	6.56	8.77	11.34	16.48	23.37	17.68	4.05
AA1	12.11	10.64	7.47	8.10	6.35	15.42	11.28	8.20	10.50	7.18	2.76	0
AA2	8.73	11.90	6.77	9.10	11.22	9.08	16.27	10.95	11.63	2.92	1.44	0
AA3	12.25	11.62	4.07	7.15	3.04	15.47	15.43	10.40	7.55	7.92	4.85	0.26

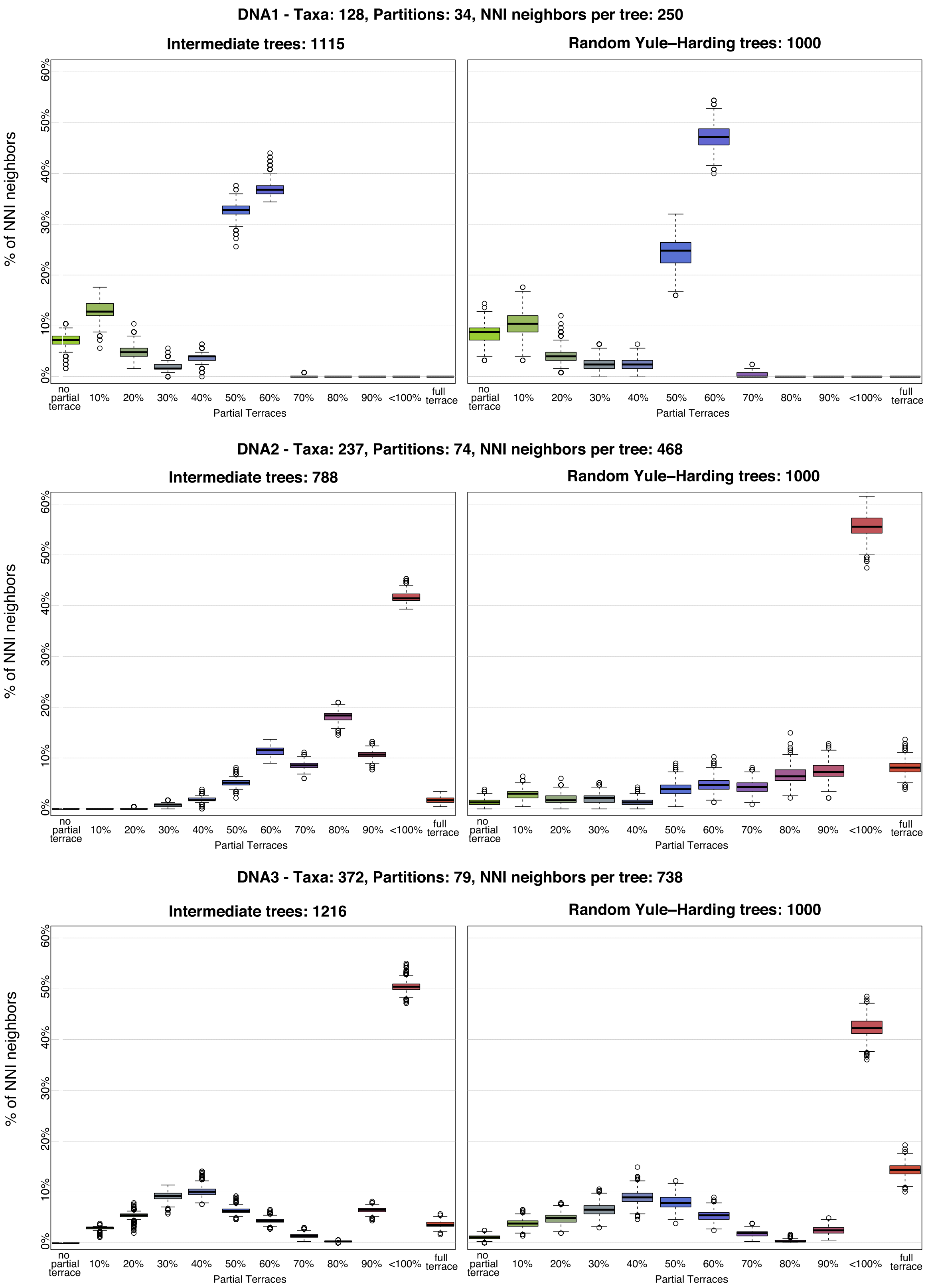
Table 2.3. The mean percentage of trees from NNI neighbourhood of intermediate trees falling into corresponding partial terrace bin.

Intermediate and random trees have similar percentages of T_{NNI} trees across different bins (Fig. 2.7, Appendix A Fig. A1-A4). This suggests that the general picture of partial terraces is mainly determined by the spread of missing data in the supermatrix and is less dependent on the actual tree topology. Moreover, increasing the number of taxa tends to decrease the variance of T_{NNI} percentage within each bin (for both intermediate and random trees).

Figure 2.8 integrates the information from Tables 2.2 and 2.3 and provides with a rough estimates of potential computational savings, if accounting for partial and full terraces. The green bars reflect the average percentage of identical induced partition trees when T_{NNI} is compared to T.

For example, for DNA1 there is no full terrace, but we observe partial terraces that may lead to a reduction of about 38% (the percentage of green bars) in computation time.

There is a full terrace for DNA2, but it consists of only 1.75% of the NNI neighbourhood. Whereas partial terraces constitute the remaining 98.25% and lead to a potential reduction of computations of about 80% (the percentage of green bars). In fact, since no T_{NNI} tree falls into “no PT” bin, we can save some computation time for all the trees encountered during tree search. Similar trend is observed for DNA3 and DNA4 with the predicted timesaving of 71% for each alignment.



DNA1 - Taxa: 128, Partitions: 34, NNI neighbors per tree: 250

Intermediate trees: 1115

Random Yule–Harding trees: 1000

DNA2 - Taxa: 237, Partitions: 74, NNI neighbors per tree: 468

Intermediate trees: 788

Random Yule–Harding trees: 1000

DNA3 - Taxa: 372, Partitions: 79, NNI neighbors per tree: 738

Intermediate trees: 1216

Random Yule–Harding trees: 1000

Figure 2.7. NNI neighbourhood analysis for alignments DNA1 (top), DNA2 (middle) and DNA3 (bottom).

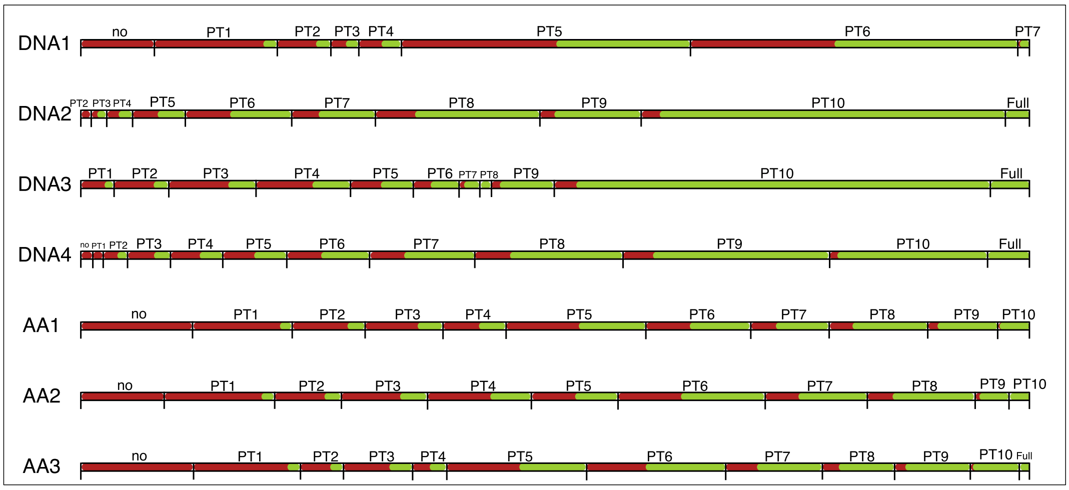


Figure 2.8. Visualization of NNI neighbourhoods and potential computational savings. Each horizontal line reflects the NNI neighbourhood for each tested alignment, i.e. 100% of T_{NNI} trees. These neighbourhoods are divided into partial terrace bins (Table 2.2) and depicted here by horizontal segments. The length of each segment corresponds to the mean percentage of T_{NNI} trees falling into the bin (Table 2.3). Each segment is composed of green and a red bar, corresponding to the fractions of partition trees that are shared and not shared between T and T_{NNI}, respectively. Basically, green bars indicate potential computational savings when accounting for partial and full terraces during the tree search.

2.5 Advantages of using induced partition trees in ML inference

In maximum likelihood inference after applying a topological rearrangement on T, one needs to optimize the edge lengths of a new tree T_{NEW}. Therefore, together with the topological changes of partition trees it is important to consider how topological rearrangement on T influences edge length optimization.

In the following we discuss two partition models commonly used in likelihood inferences, *edge-unlinked* (EUL) and *edge-linked* (EL), and the advantages of using induced partition trees for either model (Yang 1996).

We start by considering the most general partition model, EUL. Given a species tree T we first obtain the corresponding induced partition trees. Under EUL model the edge lengths of the partition trees are optimized separately. The edge lengths of T are then computed from the corresponding edge lengths inferred on the partition trees, for example, as mean edge length.

Therefore, if the topological rearrangement on T does not change the topology of a partition tree $T|Y$, no edge length optimization is necessary and as a result the optimal partition tree likelihood remains unchanged after such a topological rearrangement on T. Let T_{NEW} be a tree obtained from T by some topological rearrangement.

Under EUL partition model there is no need to optimize the edge lengths of partitioned trees shared between T and T_{NEW}. As a result the log-likelihood of the corresponding partition trees is the same.

In contrast to the EUL model, the edges between T and partition trees are linked in the EL model. That means there is only one set of edge lengths for T and partition trees with the possibility of rescaling edge lengths of each partition tree by a partition-specific evolutionary rate. Therefore, the optimization of edge lengths is done on the species tree. Even if a topological rearrangement on T does not change the topology of partition tree, it still affects the optimal partition tree likelihood via optimization of edge lengths. This is also the reason why full terraces cannot occur under the EL model (Sanderson, et al. 2015).

Theoretically, one would need to optimize each edge on the species tree, which would definitely influence the partition tree edge lengths and also the likelihood. But in practice to save computations, one only optimizes those edges in the vicinity of topological changes (Guindon, et al. 2010; Nguyen, et al. 2015; Stamatakis, et al. 2005) For example, for an NNI one only re-optimizes the five edge lengths (e, e_1, e_2, e_3, e_4) around the swap. Under EL model such particular feature of practical optimization can take an advantage when considering the induced partition trees.

Given a partition tree with taxon set Y and an edge e on T with the corresponding split $A|B$, if $A \cap Y = \emptyset$ or $B \cap Y = \emptyset$, then the optimization of e does not affect the likelihood of $T|Y$.

In this case a split $A|B$ does not have a corresponding split in $\Sigma(T|Y)$ and therefore, edge e is not linked to any edge on $T|Y$. This observation can be exploited to save computing time.

2.6 Discussion

We have shown that it is advantageous to identify and account for full and partial terraces during the tree search in phylogenomics. One main advantage is the saving of computation time. If two trees belong to the same full or partial terrace, then one needs to compute the objective function for the identical partition trees only once. The values of objective function will be the same for these partition trees. The larger the number of identical partition trees between species trees is the more computation time can be saved.

From the conditions discussed in the previous sections, the topological rearrangement that benefits the most from partial terraces is obviously NNI. It is intuitive that NNI applied to the species tree will not change the topology of partition trees more often than SPR or TBR. However, in tree searches one typically applies short SPR (e.g., RAxML), i.e., the number of edges between the pruning and the regrafting edges are much smaller than the number of taxa. The same is true for TBR. And since one also expects short SPR and short TBR to result in no change of partition trees quite often for sparse supermatrices, partial terraces are also beneficial for these rearrangements.

Moreover, the use of induced partition trees has another advantage that long SPR or TBR on a species tree T, as a result of missing data, might correspond to a much shorter SPR or TBR on $T|Y$. This leads to computation saving even if SPR or TBR change the topology of the induced partition trees.

Here, we elucidated the frequent existence of partial terraces in practice via NNI neighbourhoods, showing that partial terraces are not only a theoretical concept, but also have practical implications in phylogenomics. The predicted timesaving for the examined real alignments are only the rough estimates, since we treated the alignment lengths per partition as equal. If the length of alignment corresponding to the shared partition trees is relatively large compared to the whole supermatrix, than one expects even more speed up.

Another important factor for timesaving is the actual implementation of search strategies in the particular software. We plan to implement efficient techniques to take full advantage of partial and full terraces in IQ-TREE. A more thorough analysis of such techniques will be presented elsewhere.

CHAPTER 3

Terrace Aware Data Structure for Phylogenomic Inference from Supermatrices

In phylogenomics the analysis of concatenated gene alignments, the so-called supermatrix, is commonly accompanied by the assumption of partition models. Under such models each gene, or more generally partition, is allowed to evolve under its own evolutionary model. Though partition models provide a more comprehensive analysis of supermatrices, missing data may hamper the tree search algorithms by the existence of phylogenetic (partial) terraces.

Here we introduce the phylogenetic terrace aware (PTA) data structure for efficient analysis under partition models. In the presence of missing data PTA exploits (partial) terraces and induced partition trees to save computation time.

We show that an implementation of PTA in IQ-TREE leads to a substantial speedup of up to 5 and 6 times compared with the standard IQ-TREE and RAxML implementations, respectively. PTA is generally applicable to all types of partition models and common topological rearrangements thus can be employed by all phylogenomic inference software.

3.1 Introduction

The gigantic amount of sequence data generated by next generation sequencing technologies has spurred phylogenomics (Delsuc, et al. 2005; Eisen 1998; Kumar, et al. 2012). Here, one aims to infer the tree of life from multiple genes, loci, or even whole genomes, which provide enough phylogenetic information to resolve difficult branching orders (Bininda-Emonds, et al. 1999; Dunn, et al. 2008; Meusemann, et al. 2010; Rokas, et al. 2003).

Phylogenomic inference methods are categorized into supertree and supermatrix methods (Bininda-Emonds, et al. 2002; De Queiroz, et al. 1995; Delsuc, et al. 2005; Kupczok, et al. 2010; Sanderson, et al. 1998). Supertree methods either combine inferred (gene) trees into one "supertree" or concurrently reconstruct gene trees and species tree. Supermatrix refers to the concatenation of multiple sequence alignments from different genes. Unavailable sequences are substituted in the supermatrix by gaps and constitute the so-called *missing data*. Traditional phylogenetic methods are then used to reconstruct the species tree from the concatenated alignment.

Complex evolutionary scenarios of multi-gene data sets raise additional difficulties for phylogenetic inferences from supermatrices. For example, failure to account for heterogeneous evolution caused by heterotachy (Lopez, et al. 2002), i.e. when evolutionary rates vary over time, leads to systematic errors in phylogenetic reconstruction (Kolaczkowski and Thornton 2004; Philippe, et al. 2005).

To account for different evolutionary scenarios partition models were introduced (Yang 1996) that allow genes to evolve with different substitution models. Three types of partition models *Edge-Unlinked*, *Edge-Linked-joint* and *Edge-Linked-proportional* are implemented in many maximum likelihood (ML) software packages (Table 3.1).

Software	EL-joint	EL-proportional	EUL
MetaPIGA (Helaers and Milinkovitch 2010)	x	x	
PhyML (Guindon, et al. 2010)			
GARLI (Zwickl 2006)	x	x	
RAxML (Stamatakis 2014)	x		x
TreeFinder (Jobb, et al. 2004)	x	x	x
IQ-TREE (Nguyen, et al. 2015)	x	x	x

Table 3.1. Availability of partition models in ML tree search software.

The *Edge-Unlinked (EUL)* partition model, where each partition has its own set of edge lengths is the most general. However, due to missing data an EUL model may lead to phylogenetic terraces (Sanderson, et al. 2011), where different tree topologies have identical score (ML or parsimony). The more restrictive *Edge-Linked (EL)* partition models could be used to avoid terraces, but assuming these models in the presence of heterotachy can be misleading (Sanderson, et al. 2015).

Large phylogenetic terraces may hamper a thorough exploration of tree space by current search algorithms. When encountering a large terrace during the tree search a lot of computation time is spent on the evaluation of equally optimal trees. Therefore, it is important to detect terraces and to avoid unnecessary computations.

Recently, we generalized the concept of terraces to *partial terraces* (Chernomor, et al. 2015) and provided conditions to quickly identify their occurrences for a species tree and a supermatrix. We also predicted how much time could be saved when accounting for (partial) terraces during the tree search. However, an efficient implementation of the theoretical results was not provided.

Here, we first provide the background of phylogenetic partial terraces and the rule to quickly detect them. We then formally review the three partition models. Next, we describe a *phylogenetic terrace aware* (PTA) data structure and provide a dynamic programming algorithm to build it. We reformulate condition to quickly identify partial terraces and discuss additional timesaving features of different partition models in ML inference. We implemented PTA and the rule to detect partial terraces in IQ-TREE. We finally analyse the efficiency of PTA by examining 11 published alignments and compare the results with the standard IQ-TREE implementation and with RAxML.

3.2 Background

3.2.1 Partial and full phylogenetic terraces

Let $k \geq 2$ denote the number of genes, loci, or codon positions of protein-coding DNA in a supermatrix. In the following we use "partition" to generally refer to any subset of genomic positions. Denote by $Y_1, Y_2, \dots, Y_k$ the species sets for the k partitions and $X = Y_1 \cup Y_2 \cup \dots \cup Y_k$ the set of all species. $S_1, S_2, \dots, S_k$ denote the corresponding alignments and S is the concatenated alignment (supermatrix) of $S_1, S_2, \dots, S_k$. Stretches of unknown characters are added to S if a species has no sequence for some partition (i.e., when $Y_i \neq X$).

For a species tree T and a given partition Y_i, the associated induced partition tree, denoted $T|Y_i$, is the tree obtained from T by pruning species with no sequence for partition Y_i (i.e. missing sequences). Hence, for every species tree there is a corresponding set of k induced partition trees.

If for two species trees T_1 and T_2 there exists a set of indices $J \subseteq \{1, \dots, k\}$ such that $\forall j \in J$ the corresponding induced partition trees $T_1|Y_j$ and $T_2|Y_j$ are identical then it is said that T_1 and T_2 belong to one *partial terrace* (Chernomor, et al. 2015). In this context, a phylogenetic terrace coined in Sanderson, et al. (2011) is a special case when $J = \{1, \dots, k\}$. For clarity we call this case a *full terrace.* For $\forall j \in J$ the scores (ML or parsimony) of $T_1|Y_j$ and $T_2|Y_j$ are equal. Therefore, if during the tree search we identify full terrace, one needs to compute the score of $T_1|Y_j$ or $T_2|Y_j$ only once $\forall j \in J$.

3.2.2 How to identify partial terraces during the tree search

When searching for the optimal species tree we move from one species tree T to another T_{NEW} by means of some topological rearrangements. In phylogenetic software the most common topological rearrangements used are Nearest Neighbour Interchange (NNI), Subtree Pruning and Regrafting (SPR), Tree Bisection and Reconnection (TBR) (Felsenstein 2004).

In order to detect partial terraces during the tree search it is enough to answer the question whether the topological rearrangement applied to a species tree T changes any of its partition trees. If some partition trees remained unchanged, then T and T_{NEW} belong to one partial terrace and we only need to compute the score (ML or parsimony) of partition trees that were affected by the topological rearrangement.

We now illustrate the necessary condition presented in Chernomor, et al. (2015) for the NNI to change the topology of a given induced partition tree. Let T be a species tree and e an interior edge of T. We denote by e_1, e_2, e_3, e_4 edges adjacent to e and by A, B, C, D the taxon sets leading from them respectively (Fig. 3.1). Now, let a new tree T_{NNI} be obtained from T via NNI. Then the Proposition 1 (Chernomor, et al. 2015) states that

> *"For a partition with a taxon set Y the topologies of $T|Y$ and $T_{NNI}|Y$ are different iff Y has at least one representative taxon in each subset A, B, C, D."*

(*)

Condition (*) simply means that for an NNI to change the topology of a partition tree $T|Y$ all $A \cap Y, B \cap Y, C \cap Y$ and $D \cap Y$ must be not empty.

In the following we discuss the three partition models that we implemented in IQ-TREE and used to examine the performance of the proposed here terrace aware data structure.

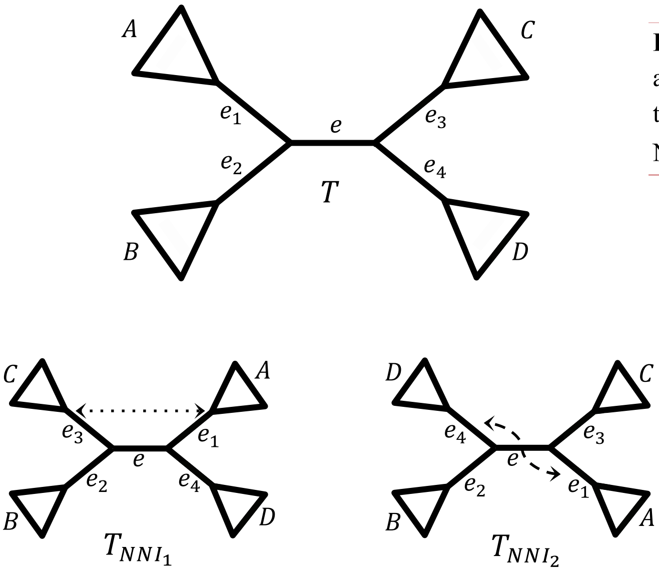


Figure 3.1. The species tree T and the two NNI neighbouring trees T_{NNI_1} and T_{NNI_2}, obtained by NNIs around the central edge *e*.

3.2.3 Partition models

Partition models allow for different evolutionary scenarios for each partition. More formally, each partition Y_i is assumed to evolve under its own substitution model M_i and the model parameters of each M_i are optimized separately on the corresponding partition tree.

The log-likelihood of a species tree T under a partition model M is the sum of partition tree log-likelihoods

$$\ell(T, M \mid S) = \sum_{i=1}^{k} \ell\,(T|Y_i, M_i \mid S_i). \quad (1)$$

Here, the log-likelihood ℓ of the partition trees depends on the topology and the edge lengths of each $T|Y_i$. The relation of edge lengths between species tree and partition trees is what distinguishes partition models.

The most general EUL model, denoted by M_{EUL}, allows each partition tree $T|Y_i$ to have its own set of edge lengths that are optimized separately per partition tree. We denote a length of e on $T|Y_i$ by $\lambda_i(e)$. The EUL model implies that the species tree T has no defined edge lengths. However, one can display the edge lengths for T, for example, as the weighted average of corresponding edges on partition trees.

In contrast to EUL, the more restrictive EL models assume relationships between edge lengths of the species tree and all partition trees. The *EL-proportional* model, denoted by $M_{EL_{prop}}$, assumes one set of edge lengths, $\lambda(.)$, for the species tree T and rescales the partition trees with specific positive rates $r_1, r_2, \ldots, r_k$ such that the weighted average rate is 1 (i.e., $\frac{\sum_{i=1}^{k} w_i r_i}{\sum_{i=1}^{k} w_i} = 1$, where w_i is the length of partition alignment S_i). The partition rates r_i are optimized separately for each partition tree.

The second EL model, *EL-joint*, denoted by $M_{EL_{joint}}$, is a special case of $M_{EL_{prop}}$ with all the partition rates equal to 1 (i.e., $r_1 = r_2 = \cdots = r_k = 1$). This simply means that the edge lengths of partition trees are equal to the lengths of corresponding edges on the species tree.

In contrast to M_{EUL}, which optimizes the edge lengths per partition tree, EL models optimize edge lengths, $\lambda(.)$, on the species tree.

3.3 Phylogenetic Terrace Aware data structure

In this section we introduce the phylogenetic terrace aware (PTA) data structure, which facilitates the detection and handling of partial terraces during the tree search and provides an efficient analysis of supermatrices. PTA consists of the species tree, the set of its induced partition trees and the set of maps, which link edges of the species tree to each induced partition tree. In the following we introduce this map and an efficient algorithm to build it.

3.3.1 Map from the species tree onto partition trees

Let E denote the set of all edges of T and E_i the edge set of $T|Y_i$. We represent each edge $e \in E$ by its split $e = A|B$, where A and B are disjoint complementary non-empty subsets of the leaf set X with $|X| = n$. For every partition Y_i we introduce the map

$$f_i : E \to E_i \cup \{\varepsilon\},$$

$$f_i(e) = \begin{cases} A \cap Y_i | B \cap Y_i, & \text{if } A \cap Y_i \neq \emptyset \text{ and } B \cap Y_i \neq \emptyset \\ \varepsilon, & \text{otherwise.} \end{cases} \quad (2)$$

In supertree terminology, $f_i(.)$ is the map from supersplits in T to subsplits (or partial splits) in $T|Y_i$ (Semple and Steel 2003; chap. 6). Every supersplit has no or exactly one corresponding subsplit, whereas a subsplit has one or more corresponding supersplits. If a supersplit has no corresponding subsplit, we equate its map to ε. Basically, $f_i(e)$, if not equal to ε, is an edge on $T|Y_i$ corresponding to e.

The collection of all maps $F = \{f_1, \dots, f_k\}$ together with the trees $\{T, T|Y_1, \dots, T|Y_k\}$ forms the PTA data structure for partition model analyses.

3.3.2 An efficient algorithm for building *F*

We now describe a dynamic programming algorithm to build F in linear time for unrooted bifurcating trees using a post-order tree traversal. It first assigns f_i for external edges and then proceeds towards internal edges of the tree once the two neighbouring edges have already been processed. More specifically, let $e = \{x\}|X \backslash \{x\} \in E$ be an external edge then

$$f_i(e) = \begin{cases} \varepsilon, & x \notin Y_i \\ \{x\}|Y_i \backslash \{x\}, & x \in Y_i \end{cases} \quad (3)$$

Obviously, Eq. (3) follows directly from Eq. (2). Now, let e be an internal edge and e, e_1, e_2 are adjacent edges. Then if e_1, e_2 have already been processed, we assign $f_i(e)$ as follows:

$$f_i(e) = \begin{cases} \varepsilon, & f_i(e_1) = f_i(e_2) \\ f_i(e_1), & f_i(e_1) \neq \varepsilon \text{ and } f_i(e_2) = \varepsilon \\ f_i(e_2), & f_i(e_1) = \varepsilon \text{ and } f_i(e_2) \neq \varepsilon \\ e_i^*, & \text{otherwise,} \end{cases} \quad (4)$$

where e_i^* is an edge on $T|Y_i$ adjacent to $f_i(e_1)$ and $f_i(e_2)$.

Proof for the correctness of Eq. (4).

Figure 3.2a illustrates T around e_1, e_2, e, where A, B, C are the three corresponding species sets. We note that $e = (A \cup B)|C$, $e_1 = A|(B \cup C)$, $e_2 = B|(A \cup C)$.

From the definition of map $f_i(.)$ it follows that

$$f_i(e) = \begin{cases} (A \cup B) \cap Y_i | C \cap Y_i, & (A \cup B) \cap Y_i \neq \emptyset \text{ and } C \cap Y_i \neq \emptyset \\ \varepsilon, & \text{otherwise} \end{cases}, \quad (5)$$

$$f_i(e_1) = \begin{cases} A \cap Y_i | (B \cup C) \cap Y_i, & A \cap Y_i \neq \emptyset \text{ and } (B \cup C) \cap Y_i \neq \emptyset \\ \varepsilon, & \text{otherwise} \end{cases}, \quad (6)$$

$$f_i(e_2) = \begin{cases} B \cap Y_i | (A \cup C) \cap Y_i, & B \cap Y_i \neq \emptyset \text{ and } (A \cup C) \cap Y_i \neq \emptyset \\ \varepsilon, & \text{otherwise} \end{cases}. \quad (7)$$

We now consider the four cases from Eq. (4):

1) If $f_i(e_1) = f_i(e_2) = \varepsilon$, then from (6) and (7) it follows that at least two of the three intersections $A \cap Y_i$, $B \cap Y_i$ and $C \cap Y_i$ are empty. Therefore from Eq. (5) we have $f_i(e) = \varepsilon$. Otherwise if $f_i(e_1) = f_i(e_2)$ and are different from ε, then we have $A \cap Y_i = (A \cup C) \cap Y_i$ and $B \cap Y_i = (B \cup C) \cap Y_i$, from which it follows that $C \cap Y_i = \emptyset$ and thus $f_i(e) = \varepsilon$.

2) $f_i(e_1) \neq \varepsilon$ and $f_i(e_2) = \varepsilon$. From $f_i(e_1) \neq \varepsilon$ it follows that $A \cap Y_i \neq \emptyset$ and $(B \cup C) \cap Y_i \neq \emptyset$, while from $f_i(e_2) = \varepsilon$, $B \cap Y_i = \emptyset$ or $(A \cup C) \cap Y_i = \emptyset$. Since $A \cap Y_i \neq \emptyset$ then $(A \cup C) \cap Y_i \neq \emptyset$, and therefore $B \cap Y_i = \emptyset$ and $C \cap Y_i \neq \emptyset$. Since sets $A \cap Y_i$ and $C \cap Y_i$ are not empty while $B \cap Y_i$ is, then

$$f_i(e) = (A \cup B) \cap Y_i | C \cap Y_i = A \cap Y_i | (B \cup C) \cap Y_i = f_i(e_1).$$

3) $f_i(e_1) = \varepsilon$ and $f_i(e_2) \neq \varepsilon$. Similar to condition 2 above, we have $f_i(e) = f_i(e_2)$.

4) If $f_i(e_1) \neq f_i(e_2)$ and are both not equal to ε. From $f_i(e_1) \neq \varepsilon$ we have that $A \cap Y_i \neq \emptyset$, from $f_i(e_2) \neq \varepsilon$ it follows that $B \cap Y_i \neq \emptyset$, and since $f_i(e_1) \neq f_i(e_2)$ then $C \cap Y_i \neq \emptyset$. Therefore, $f_i(e) = (A \cup B) \cap Y_i | C \cap Y_i \neq \varepsilon$ is an edge on subtree $T|Y_i$ incident to $f_i(e_1)$ and $f_i(e_2)$ (Fig. 3.2b).

Thus, Eq. (4) is correct.

■

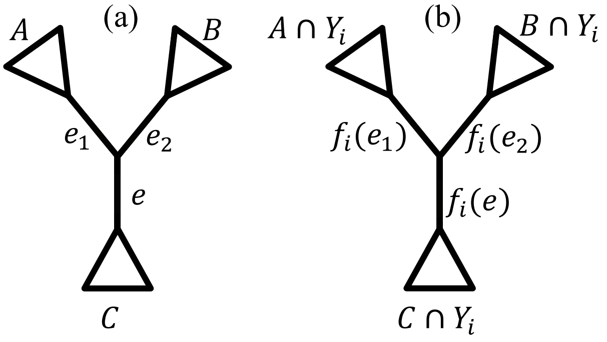


Figure 3.2. Proof for the correctness of Eq. (4). **(a)** Three adjacent edges on species tree T and **(b)** their corresponding edges on partition tree $T|Y_i$.

The described algorithm builds F in $O(nk)$ time, where n and k are the number of species and partitions respectively. Note that F has to be recomputed each time the topology of the species tree changed. In the following we present how to update the PTA when an NNI is applied to the species tree.

3.3.3 Identifying unchanged partition trees with PTA

For a species tree T and a partition Y_i condition (*) provides a check whether an NNI applied to e changes the topology of $T|Y_i$. Using the map notation, (*) is equivalent to

> *"For a partition with a taxon set Y_i the topologies of $T|Y_i$ and $T_{NNI}|Y_i$ are different iff all $f_i(e_1), f_i(e_2), f_i(e_3), f_i(e_4) \neq \varepsilon$."* (**)

Condition (**) follows directly from (*) and the definition of $f_i(.)$.

When an NNI is applied to e, one updates each partition tree $T|Y_i$ using two rules:

1. If all $f_i(e_1), f_i(e_2), f_i(e_3), f_i(e_4) \neq \varepsilon$, then $T_{NNI}|Y_i$ will result from $T|Y_i$ by swapping the corresponding edges. For example, if e_1 and e_3 are swapped on T, then $f_i(e_1)$ and $f_i(e_3)$ are swapped on $T|Y_i$.
2. Otherwise, the topologies of $T_{NNI}|Y_i$ and $T|Y_i$ are identical. Thus, we keep the tree topology of $T|Y_i$ and have to update $f_i(e)$ according to Eq. (4).

Under the EUL partition model when computing the log-likelihood of T_{NNI} we only have to compute the log-likelihood of $T_{NNI}|Y_i$ in the first case. In the second case the optimal log-likelihoods of $T|Y_i$ and $T_{NNI}|Y_i$ are equal. This advantage comes from the fact that under EUL model the edge lengths of partition trees are optimized independently. Therefore, if the topologies of $T_{NNI}|Y_i$ and $T|Y_i$ are identical, there is no need to optimize edges. Thanks to this, the computing time for the EUL model benefits a lot from partial terraces. In fact, when T_{NNI} and T belong to one full terrace, i.e. when for all partitions condition (**) is not satisfied, no recomputation is necessary at all.

More restrictive EL models require additional care. Since under EL models the edges of the species tree and partition trees are "*linked*", together with the topological changes of $T|Y_i$ we also have to account for changes of edge lengths. Using the map $f_i(.)$, the edge lengths of partition tree $T|Y_i$ are computed as

$$\lambda_i(e') = r_i \times \sum_{e \in E : f_i(e) = e'} \lambda(e), \quad \forall e' \in E_i. \qquad (8)$$

Therefore, each time map $f_i(.)$ is changed, the edge lengths on $T|Y_i$ will also change.

If the topology of $T|Y_i$ is not changed by the NNI (i.e. condition (**) is not satisfied), there are four different cases possible, which lead to varying computational timesavings (see Appendix B.2).

For EL models PTA helps to save computation time during edge length optimization. To speedup the optimization of each edge $e \in E$, in Eq. (1) one only has to sum the log-likelihoods over those partitions Y_i where $f_i(e) \neq \varepsilon$. This advantage is a result of using induced partition trees instead of complete partition trees (i.e. with a full set of species instead of Y_i).

Since NNI changes one split of the species tree, namely, the split of an edge it is applied to, F is recomputed in $O(k)$ time. Thus, the extra computations needed to maintain F are negligible compared to the expensive likelihood computations.

Although we only reformulated the condition for NNI, we note that a similar reformulation can also be applied for SPR and TBR conditions (see Appendix B.1). Thus, the PTA can be employed with all common topological rearrangements.

3.4 Performance assessment on real alignments

We denote by IQ-TREE$_{\text{PTA}}$, the IQ-TREE version 1.3.3 that implements the PTA data structure. The performance of IQ-TREE$_{\text{PTA}}$ is compared with the standard IQ-TREE implementation and with RAxML version 8.1.24. We also tested RAxML with option –U (denoted by RAxML$_{\text{optU}}$), which disregards missing data and results in memory and time saving for gappy alignments (Izquierdo-Carrasco, et al. 2011). Note that RAxML implements EUL and EL-joint models, but not the EL-proportional model. Therefore, the last model was only examined with IQ-TREE and IQ-TREE$_{\text{PTA}}$.

We analysed eleven (eight DNA and three AA) alignments (Table 3.2) with the percentages of missing data ranging from 30% to 73%.

Type ID	Taxa	Genes	Length	Missing data	Source
DNA1	128	34	29,198	30%	Stamatakis and Alachiotis (2010)
DNA2	180	15	14,912	60%	van der Linde, et al. (2010)
DNA3	237	74	43,834	72%	Nyakatura and Bininda-Emonds (2012)
DNA4	298	3	5,074	34%	Bouchenak-Khelladi, et al. (2008)
DNA5	372	79	61,199	66%	Springer, et al. (2012)
DNA6	404	11	13,158	60%	Stamatakis and Alachiotis (2010)
DNA7	435	18	16,016	73%	Hinchliff and Roalson (2013)
DNA8	767	5	5,714	59%	Pyron, et al. (2011)
AA90	69	31	8,546	35%	Dell'Ampio, et al. (2014)
AA10	70	35	11,789	34%	
AA11	72	51	12,548	35%	

Table 3.2. Benchmark Alignments

For all partition models we assumed the GTR+Γ (Lanave, et al. 1984; Yang 1994) and the LG+Γ (Le and Gascuel 2008; Yang 1994) models for all genes in the DNA and the AA alignments, respectively.

For each program we performed 10 independent tree reconstruction runs per alignment and per partition model. All computations were carried out on the Vienna Scientific Cluster 3.

3.4.1 CPU time comparison

For each partition model and each alignment we computed: (i) the average CPU time over 10 runs for each program; and (ii) the speedup of IQ-TREE$_{PTA}$ compared to other implementations: the ratio between the average CPU time of each program and that of IQ-TREE$_{PTA}$.

Table 3.3 shows the average CPU times for the EUL (panel a) and the EL-joint (panel b) models obtained by RAxML, RAxML$_{optU,}$ IQ-TREE, and IQ-TREE$_{PTA}$. The last three columns show the IQ-TREE$_{PTA}$ speedups compared to other three programs.

Under the EUL model IQ-TREE$_{PTA}$ was the fastest program for all alignments (Table 3.3a, Fig. 3.3a). On average it runs 3 times faster than the standard IQ-TREE and 3.26 times faster than RAxML and RAxML$_{optU}$ over all alignments. For three alignments (DNA4, DNA6, DNA8) IQ-TREE was much slower than RAxML and RAxML$_{optU}$ (Table 3.3a). However, IQ-TREE$_{PTA}$ achieved a substantial speedup (ranging from 3.52 to 5.12) for these alignments compared to IQ-TREE and thus bypassed the performance of RAxML.

Under the EL-joint partition model IQ-TREE$_{PTA}$ was on average 1.8 times faster than IQ-TREE (Table 3.3b, Fig. 3.3b). The least speedups correspond to alignments with the smallest percentages of missing data (DNA1, DNA4, AA9, AA10, AA11). IQ-TREE$_{PTA}$ was slower than RAxML and RAxML$_{optU}$ for DNA4 and DNA6, but was faster for the remaining nine alignments.

Finally, the EL-proportional model was only examined with IQ-TREE and IQ-TREE$_{PTA}$. Table 3.4 and Figure 3.3c show that IQ-TREE$_{PTA}$ is on average 1.9 times faster than the standard IQ-TREE.

3.4.2 Log-likelihood comparison

Under the EUL model the average log-likelihoods of the best-found trees obtained by different programs are quite similar with maximal difference of 50 (Fig. 3.4a) except for DNA1, DNA3 and DNA5, where the IQ-TREE$_{PTA}$ ML trees have log-likelihoods, which are on average 200, 350 and 100 units higher than RAxML trees, respectively. The log-likelihoods from 10 IQ-TREE$_{PTA}$ runs have consistently small variance, whereas the other programs sometimes show large variances (RAxML for DNA3, RAxML$_{optU}$ for DNA3 and DNA6, IQ-TREE for DNA8).

Under the EL-joint model the log-likelihoods of the best-found trees obtained by different implementations are similar for most alignments (Fig. 3.4b) except for DNA3. The DNA3 alignment is also characterized by the largest variance of log-likelihoods over 10 runs for all implementations with the largest variance of ±250 units for RAxML.

Under the EL-proportional model the average log-likelihoods for IQ-TREE$_{PTA}$ and IQ-TREE runs agree on nine alignments (Fig. 3.4c) while not for DNA5 and DNA8. For DNA5 IQ-TREE$_{PTA}$ found trees that have on average 300 units higher log-likelihoods than IQ-TREE, whereas for DNA8 the IQ-TREE$_{\text{PTA}}$ trees showed 250 units smaller log-likelihoods. Finally, for DNA3 and DNA5 both IQ-TREE and IQ-TREE$_{\text{PTA}}$ showed large variances in log-likelihoods obtained from 10 runs (e.g., ±350 units for IQ-TREE$_{\text{PTA}}$ on DNA5).

(a) Edge-Unlinked model								
Alignments		Average CPU time (hh:mm:ss)				Average IQ-TREE$_{\text{PTA}}$ speedup compared to		
Type ID	**Missing data**	**RAxML**	**RAxML$_{\text{optU}}$**	**IQ-TREE**	**IQ-TREE$_{\text{PTA}}$**	**RAxML**	**RAxML$_{\text{optU}}$**	**IQ-TREE**
DNA1	30%	02:12:33	02:11:09	01:04:17	00:38:24	3.45	3.41	1.67
DNA2	60%	01:02:23	01:05:58	00:55:53	00:24:41	2.53	2.67	2.26
DNA3	72%	04:19:19	04:26:17	02:50:54	00:55:15	4.69	4.82	3.09
DNA4	34%	**00:49:21**	**00:44:56**	**02:49:42**	**00:33:10**	**1.49**	**1.35**	**5.12**
DNA5	66%	15:34:02	15:19:25	07:37:15	02:48:47	5.53	5.45	2.71
DNA6	60%	**02:48:11**	**02:48:29**	**06:57:02**	**01:37:18**	**1.73**	**1.73**	**4.29**
DNA7	73%	02:42:19	02:33:19	02:45:00	01:07:42	2.40	2.26	2.44
DNA8	59%	**03:50:31**	**04:11:34**	**05:04:20**	**01:26:21**	**2.67**	**2.91**	**3.52**
AA90	35%	01:56:36	01:58:00	01:32:26	00:36:45	3.17	3.21	2.51
AA10	34%	03:22:44	03:16:02	02:09:26	00:50:06	4.05	3.91	2.58
AA11	35%	03:54:05	03:45:19	03:00:17	00:54:22	4.30	4.14	3.32
(b) Edge-Linked-joint model								
Alignments		Average CPU time (hh:mm:ss)				Average IQ-TREE$_{\text{PTA}}$ speedup compared to		
Type ID	**Missing data**	**RAxML**	**RAxML$_{\text{optU}}$**	**IQ-TREE**	**IQ-TREE$_{\text{PTA}}$**	**RAxML**	**RAxML$_{\text{optU}}$**	**IQ-TREE**
DNA1	30%	02:12:40	02:06:59	00:51:48	00:40:12	3.30	3.16	1.29
DNA2	60%	01:03:17	01:04:24	01:05:47	00:35:21	1.79	1.82	1.86
DNA3	72%	07:12:30	06:24:10	06:37:09	02:25:38	2.97	2.64	2.73
DNA4	34%	00:35:43	00:39:43	01:13:57	00:59:10	0.60	0.67	1.25
DNA5	66%	24:28:03	23:18:58	14:50:53	06:41:56	3.65	3.48	2.22
DNA6	60%	02:50:18	02:48:37	06:35:30	04:30:10	0.63	0.62	1.46
DNA7	73%	**03:29:43**	**03:35:04**	**06:34:58**	**02:16:27**	**1.54**	**1.58**	**2.89**
DNA8	59%	**04:08:45**	**03:55:08**	**08:12:19**	**03:21:24**	**1.24**	**1.17**	**2.44**
AA90	35%	02:18:13	02:19:47	01:09:34	00:53:27	2.59	2.62	1.30
AA10	34%	03:44:36	03:47:51	01:52:57	01:24:03	2.67	2.71	1.34
AA11	35%	05:01:00	05:22:13	02:18:36	01:40:09	3.01	3.22	1.38

Table 3.3. Comparison of average CPU runtimes between standard RAxML and IQ-TREE and their implementations accounting for missing data: RAxML$_{\text{optU}}$ (using -U option) and IQ-TREE$_{\text{PTA}}$. The speedup is computed as ratios of average CPU runtime of the corresponding implementation to IQ-TREE$_{\text{PTA.}}$ The numbers in bold face correspond to alignments where standard IQ-TREE was slower than RAxML, while IQ-TREE$_{\text{PTA}}$ substantially improves the runtimes for these alignments.

Edge-Linked-proportional model				
Alignments		Average CPU time (hh:mm:ss)		Average IQ-TREE$_{\mathrm{PTA}}$ speedup compared to
Type ID	**Missing data**	**IQ-TREE**	**IQ-TREE$_{\mathrm{PTA}}$**	**IQ-TREE**
DNA1	30%	00:54:13	00:42:15	1.28
DNA2	60%	01:11:22	00:27:59	2.55
DNA3	72%	07:02:35	02:12:31	3.19
DNA4	34%	01:10:33	00:48:59	1.44
DNA5	66%	15:53:28	09:49:17	1.62
DNA6	60%	07:54:52	03:55:55	2.01
DNA7	73%	07:43:49	02:45:11	2.81
DNA8	59%	08:22:36	03:47:27	2.21
AA90	35%	01:10:04	00:54:47	1.28
AA10	34%	01:56:31	01:28:32	1.32
AA11	35%	02:02:09	01:35:48	1.27

Table 3.4. Comparison of average CPU runtimes between IQ-TREE standard implementation and IQ-TREE$_{\mathrm{PTA}}$ for EL-proportional model. The speedup is computed as the ratio of average IQ-TREE runtimes to IQ-TREE$_{\mathrm{PTA.}}$

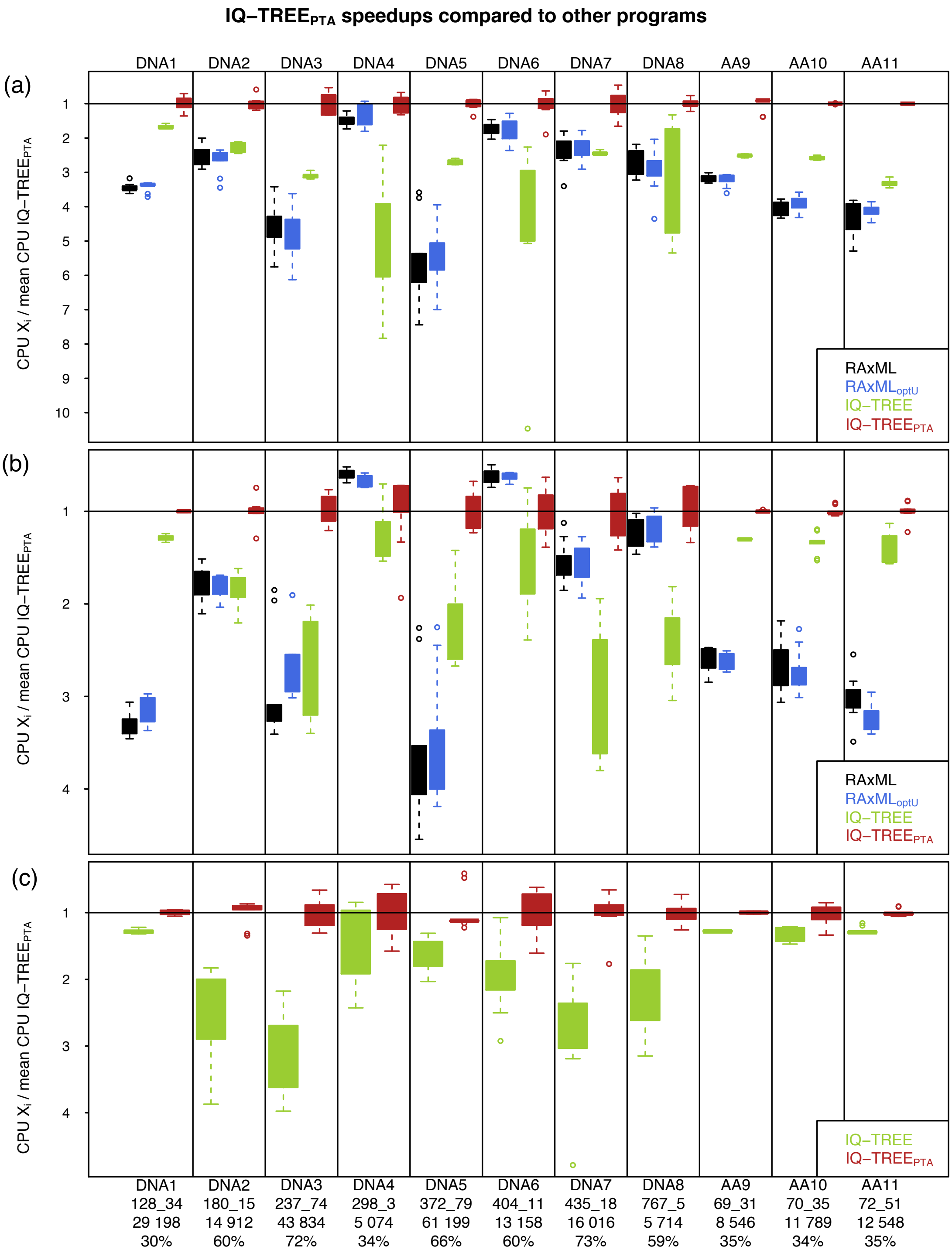


Figure 3.3. The CPU time comparison of different implementations under **(a)** EUL, **(b)** EL-joint and **(c)** EL-proportional partition models. Each boxplot shows the distribution of the runtime ratios for 10 runs between a comparing program and the mean runtime of IQ-TREE$_{PTA}$. Boxes above and below the horizontal line mean that the corresponding program is faster and slower than IQ-TREE$_{PTA}$, respectively.

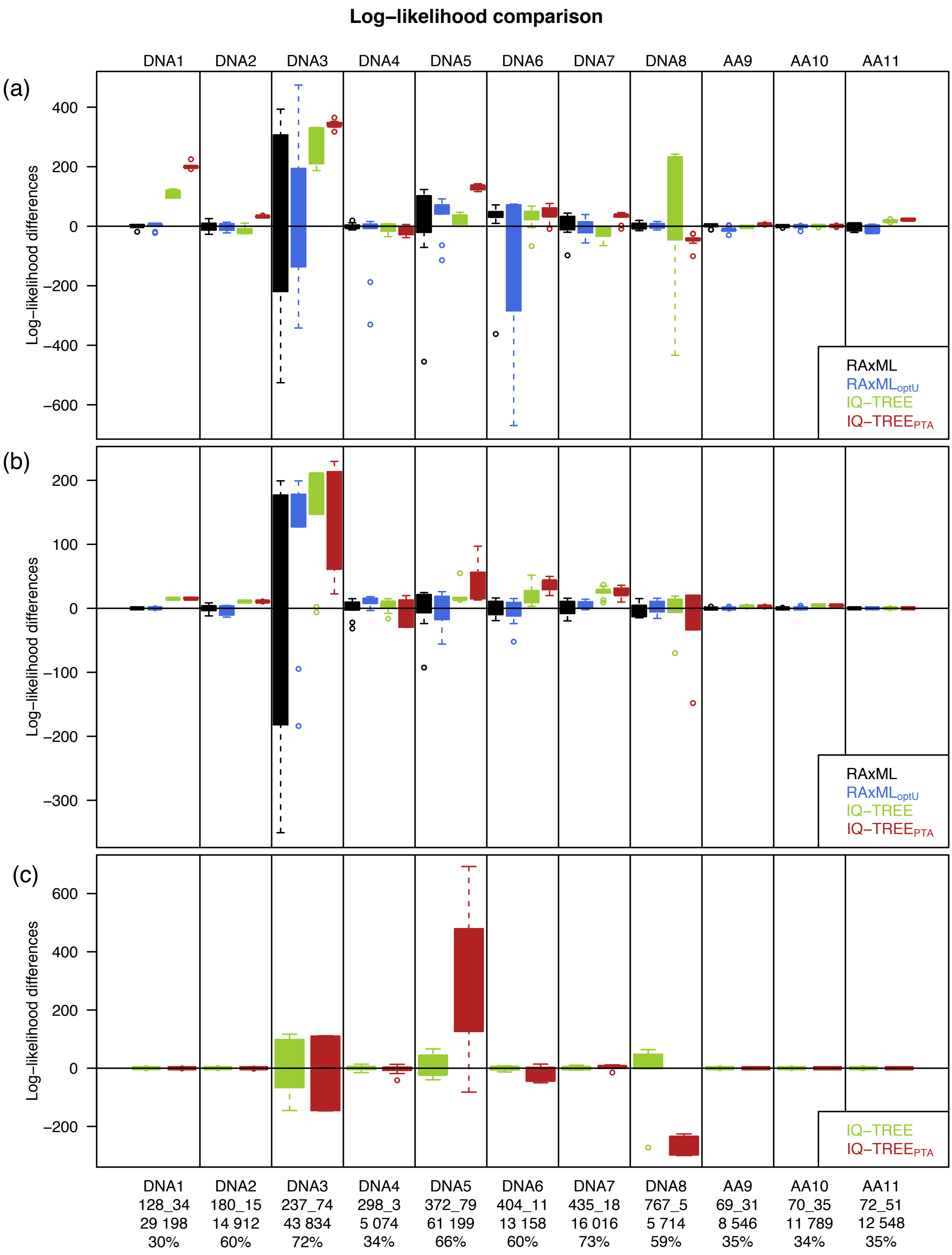


Figure 3.4. The log-likelihood comparison of different implementations under **(a)** EUL, **(b)** EL-joint and **(c)** EL-proportional partition models. Each boxplot shows the distribution of the log-likelihood differences for 10 runs between a comparing program and the mean log-likelihood of RAxML standard (panels a and b) or IQ-TREE standard (panel c). Boxes above and below the horizontal line mean that the corresponding program has higher and smaller log-likelihood than RAxML or IQ-TREE, respectively.

3.5 Conclusions

We introduced a phylogenetic terrace aware (PTA) data structure for an efficient phylogenomic inference from supermatrices. PTA consists of the species tree, the set of its induced partition trees and the set of maps, which link the edges of the species tree to the induced partition trees. This mapping enables an easy topological synchronization between the species tree and partition trees after each topological rearrangement such as NNI presented here. We note that the PTA can also be employed for SPR and TBR rearrangements following the conditions of Chernomor, et al. (2015) (see Appendix B.1 for more details). Thus, the PTA is a general data structure that can be incorporated into existing ML software packages.

In the presence of missing data, to reduce computation time PTA exploits partial terraces and avoids unnecessary likelihood recomputation. The use of induced partition trees in PTA saves time during edge lengths optimization, which is particularly helpful for the EL partition models. The overhead of maintaining the PTA mapping is negligible compared with the time-consuming likelihood computation and the observed speed up is correlated with the amount of missing data.

We implemented PTA in IQ-TREE (Nguyen, et al. 2015). Since IQ-TREE applies NNIs to search the tree space, we used the rule to identify partial terraces for NNI-based searches. Analysis on real alignments showed that accounting for partial terraces, as expected from theory, substantially speeds up the tree search under partition models. Our analysis also revealed sometimes high variances in log-likelihoods between different runs for both RAxML and IQ-TREE. This reinforces the observation (Nguyen, et al. 2015) that one should run each program as often as possible to ensure more reliable results.

As further step, we plan to implement the PTA data structure into the phylogenetic likelihood library (Flouri, et al. 2015) and to perform the analysis also with SPR.

While PTA helps to speed up tree search, it would be also interesting to derive a tree search strategy that specifically exploits the special structure of large (partial) terraces. Here, it is desirable to direct the search to “escape” large partial terraces. Otherwise, a lot of computations might be unnecessarily spent to evaluate less promising trees. Nevertheless, the PTA data structure developed here can be useful. For example, one can choose topological rearrangements on the species tree such that all partition trees are changed. The resulting species tree will most likely belong to another partial terrace, thus providing the potential to explore another region of the tree space. To the best of our knowledge, since the introduction of terrace concept (Sanderson, et al. 2011) no terrace-aware search strategy has been introduced. Such a search strategy will be essential to adequately cope with gappy phylogenomic data.

CHAPTER 4

Phylogenomics in Conservation Prioritization

In this chapter we present the application of phylogenomics to biodiversity and conservation biology.

Phylogenetic Diversity (PD) is a measure of biodiversity based on the evolutionary history of species. Here, we discuss the use of PD, and the more general measure Split Diversity (SD), in conservation prioritization.

Depending on the conservation goal and the information available about species, one can construct optimization routines that incorporate various conservation constraints. Here, we discuss the *viable taxon selection* problem, which incorporates predator-prey interactions between the species in a community to define viability constraints. First, we extend the problem to SD. Second, to make the concept of viability more realistic we extend the analysis to account for the diet composition of predators.

Despite such optimization problems falling into the area of NP-hard problems, it is possible to solve them in a reasonable amount of time using Integer Linear Programming (ILP). We apply ILP to solve viable taxon selection and its extensions, incorporating SD as an objective to prioritize candidates for the conservation actions. We implemented the discussed problems in PDA, user-friendly software available at http://www.cibiv.at/software/pda.

To exemplify the discussed problems we demonstrate their utility using data from the Caribbean coral reef community.

4.1 Introduction

Many important challenges in biodiversity conservation involve the prioritization of species, habitats or ecosystems for the allocation of limited conservation funding. These problems require techniques that allow the selection of units that maximize a quantity of interest, such as species diversity, phylogenetic diversity or ecosystem function, subject to some number of constraints (e.g., Purvis, et al. 2005). A basic approach is to focus on taxon richness (Gaston and Spicer 2004) in order to maximize the number of taxa conserved. However, the assumption that all taxa are equally valuable may make taxon richness too simplistic (May 1990).

One approach to incorporating variation among species is to use indices that take into account phylogenetic information (Crozier 1992; Faith 1992; Vanewright, et al. 1991), the most popular being phylogenetic diversity (PD; Faith 1992). PD is the amount of evolutionary history encompassed by a given number of taxa (e.g. species), and is often predictive of phenotypic diversity or the ecosystem function provided by a set of taxa (Cadotte, et al. 2012; Isaac, et al. 2007; Srivastava, et al. 2012; Winter, et al. 2013). Given a phylogenetic tree for a set of taxa, the PD of a taxon subset is calculated as the sum of branch lengths of the minimal sub-tree spanned by those taxa. PD depends on the availability of a single, reliable phylogenetic tree estimate with branch lengths, and cannot readily be calculated when one wishes to use information from multiple trees. The single tree may be a species tree reconstructed from many genes that may have different evolutionary rates (Graur and Li 2000) or even support different tree topologies (Nei 1987). One may instead wish to weigh evidence across these gene trees, or across a number of candidate trees from bootstrap samples (Felsenstein 1985) or from a Bayesian posterior distribution (Yang and Rannala 1997). To resolve this issue, Minh, et al. (2009) have recently introduced the concept of split diversity (SD), which generalizes PD by combining information from multiple trees.

Integer Linear Programming (ILP; Gomory 1958) is a widely-used technique to solve optimization problems in various scientific disciplines (e.g., Jünger, et al. 2010) with great potential for conservation decision-making. ILP solves problems by optimizing a linear objective function subject to linear constraints acting on integral variables (such as the inclusion or exclusion of species). Theoretically solving ILP is nondeterministic polynomial-time hard (NP-hard), meaning that to guarantee an optimal solution it may be necessary to evaluate exponentially many subsets of species (e.g., Karp 1972).

With the advances of state-of-the-art ILP software packages such as CPLEX (2012) and GUROBI (2012) (free of charge for academic use) many NP-hard problems can be solved within a reasonable time while ensuring optimal solutions (Jünger, et al. 2010; and references therein).

ILP was first applied to the minimum representation problem in biodiversity conservation (Cocks and Baird 1989). Subsequently, ILP has been applied to more complex topics such as minimizing the total land mass protected while maximizing the biodiversity (taxon richness or PD) given limited resources (e.g., Haight and Snyder

2009; Önal and Briers 2003; Possingham, et al. 2000; Rodrigues and Gaston 2002; Underhill 1994). Moreover, ILP also works for the more general SD measure (Minh, et al. 2009) and predator-prey relationships (Faller 2010).

Here, we further show the efficiency and flexibility of ILP to maximize SD for a more generalized conservation question, the *viable taxon selection* problem. To make the problem more realistic, we extend it to account for species' diet compositions and introduce the $d\%$-viability constraint. The approach was implemented in the PDA software package (Minh, et al. 2009) and uses the GUROBI library to solve the corresponding ILP problems. We first show on simulated data that ILP solves the viable taxon selection problem within reasonable time for a various range of input parameters. We further exemplify the use of viable taxon selection problems for a real case by analysing the data for Caribbean coral reef community.

4.2 Methods

4.2.1 Taxon selection under viability constraint

The viable taxon selection problem arises when the interactions between species are incorporated into conservation decisions. If the candidate species for prioritization depend on each other, as in a food web representing the predator-prey relationships among community members, our prioritization can account for such information. For example, we may wish to select a set of taxa S with maximal diversity under the constraint that these taxa form a *viable* food web (Moulton, et al. 2007). Here, one focuses on the bottom-up dependencies represented in food webs, so that a taxon is defined as viable in S if it is either a basal taxon in the food web (i.e. a species without prey such as a primary producer) or a predator that has at least one prey in S. S is called *viable* if all its taxa are viable. The problem is now formulated as

Problem 1 *(Viable taxon selection)*: Given a food web and a phylogenetic tree, choose a viable subset of at most k taxa, which maximizes PD.

Problem 1 has been formulated in terms of ILP by Faller (2010). Here, the question of interest is to generalize this problem to SD, such that one can account for non-tree like evolution. Such an extension of problem 1 is provided by the following statement

Problem 2 *(Viable taxon selection under SD)*: Given a food web and a split system, choose a viable subset of at most k taxa, which maximizes SD.

Let $X = \{s_1, s_2, \dots, s_n\}$ denote the set of n taxa of interest. In order to transform problem 2 into an ILP framework, we introduce for each taxon $s_i \in X$ a taxon variable v_i. Then a subset $S \subset X$ is represented by a vector $(v_1, \dots, v_n)$, where $v_i = 1$ if $s_i \in S$ and $v_i = 0$ if $s_i \notin S$.

Following the notation of Moulton, et al. (2007), let $D = (X, A)$ denote a directed acyclic graph representing the food web, where A denotes the set of arrows (directed edges) represented as a pair of taxa, s.t. $(s_i, s_j) \in A$ if taxon s_i feeds on s_j. Let C_i denote the set of preys of s_i and if $C_i = \emptyset$, s_i is called a *basal prey*.

The viability constraint for each species, that is not a basal prey, is then modelled by the following inequality

$$\sum_{i \in C_j} v_i \geq v_j, \qquad \forall j\colon C_j \neq \emptyset, \tag{1}$$

where variables v_i have to satisfy a binary constraint for all i

$$v_i \in \{0,1\}, \qquad \forall i = 1,2, \dots, n. \tag{2}$$

Viability constraint assures that $v_j \in S$, only if at least one of its prey taxa is in S. The size of S is constrained by the following inequality

$$\sum_{i=1}^{n} v_i \leq k. \tag{3}$$

To introduce SD constraints, we follow the notations of Minh, et al. (2010) by denoting an input split system as (Σ, λ), where Σ is a set of splits (bipartitions of X) and λ the split-weight function. A split $\sigma \in \Sigma$ is represented by an n-element binary vector $(\sigma_1, \sigma_2, \dots, \sigma_n)$, where n is the number of taxa and σ_i takes a value of 0 or 1 depending on the bipartition that s_i belongs to. Note that the vector $(1 - \sigma_1, 1 - \sigma_2, \dots, 1 - \sigma_n)$ represents the same split.

To compute SD of subset S of X based on the split system (Σ, λ), let introduce a so-called split variable y_σ for every split σ, where $y_\sigma = 1$ if σ separates at least two taxa of S, and $y_\sigma = 0$ otherwise. Then

$$\mathrm{SD}(S) = \sum_{\sigma \in \Sigma} \lambda_\sigma y_\sigma, \tag{4}$$

where split variables y_σ have to satisfy binary

$$y_\sigma \in \{0,1\}, \qquad \forall \sigma \in \Sigma \tag{5}$$

and split constraints

$$\sum_{i=1}^{n} \sigma_i v_i \geq y_\sigma, \quad \forall \sigma \in \Sigma, \tag{6}$$

$$\sum_{i=1}^{n} (1 - \sigma_i) v_i \geq y_\sigma, \quad \forall \sigma \in \Sigma. \tag{7}$$

The resulting solution $(v_1, \dots, v_n)$ that maximizes the objective function, Eq.(4), corresponds to a viable subset S, which is ensured by Eq.(3) to contain at most k taxa.

In case, it is important that some of the taxa are present in the solution set S irrespective of constraints, one simply sets the corresponding $v_j = 1$.

In the next section, the viable taxon selection problems 1 and 2 will be extended to a more realistic definition of viability.

4.2.2 Accounting for the diet composition and the extension to d%-viability

Problems 1 and 2 consider a predator as a viable member of a food web even if only one of its prey taxa is conserved. However, if the conserved prey taxon makes up only a small fraction of the predator's diet, the predator is unlikely to maintain sufficient food intake to be treated as a viable species. For that reason we introduce a more realistic definition of viability that considers the diet composition of predators. To this end, denote by $D = (X, W)$ a weighted food web of the taxon set X, where W is the diet composition matrix. Here, the arrow (s_j, s_i) of food web is weighted by w_{ij}, the proportion of prey s_i in the diet of predator s_j, such that the diet composition for each predator sums up to 100%, (i.e., $\sum_i w_{ij} = 1$ for every predator s_j).

Using (X, W) we compute the total diet of a predator s_j over all of its prey taxa in a set S as:

$$\delta(s_j|S) = \sum_{s_i \in S} w_{ij}. \tag{8}$$

This allows setting a constraint that each predator must have a minimum proportion of its prey composition preserved for a set of taxa to be viable. We define a subset S of taxa as *d%-viable* if every predator $s_j \in S$ has the score $\delta(s_j|S) \geq d$.

Problem 3 *(d%-viable taxon selection under SD)*: Given a weighted food web $D = (X, W)$ and a split system (Σ, λ), select a *d%*-viable subset of at most k taxa, which maximizes SD.

Problem 3 is again solved with ILP by simply modifying the viability constraint given by Eq.(1) to:

$$\sum_{i \in C_j} w_{ij} v_i \geq d v_j, \qquad \forall j : C_j \neq \emptyset. \tag{9}$$

Finalizing, the ILP formulations of problem 2 and 3 are summarized in Table 4.1.

Table 4.1: ILP formulations. Objective function and constraints of viable (problem 2) and $d\%$-viable (problem 3) taxon selection problems under SD.

	Problem 2	**Problem 3**
Maximize:	$\sum_{\sigma\in\Sigma}\lambda_\sigma y_\sigma$	
Subject to: Size constraint	$\sum_{i=1}^{n} v_i \leq k$	
Viability constraints	$\sum_{i\in C_j} v_i \geq v_j, \quad \forall j: C_j \neq \emptyset$	$\sum_{i\in C_j} w_{ij}v_i \geq dv_j, \quad \forall j: C_j \neq \emptyset$
Split constraints	$\sum_{i=1}^{n} \sigma_i v_i \geq y_\sigma, \quad \forall\sigma \in \Sigma$ $\sum_{i=1}^{n} (1-\sigma_i) v_i \geq y_\sigma, \quad \forall\sigma \in \Sigma$	
Binary constraints	$v_i \in \{0,1\}, \qquad \forall i = 1,2,\dots,n$ $y_\sigma \in \{0,1\}, \qquad \forall\sigma \in \Sigma$	

4.2.3 Methods implementation

We implemented problems 1-3 in the software package *Phylogenetic Diversity Analyser* (PDA; Minh, et al. 2009) using C++. PDA is available as a command-line program for Windows, Mac OS X, Unix, and as online web service. For the viable taxon selection problems the user inputs a phylogenetic tree or a split network and weighted or non-weighted dependency network such as food web. PDA then outputs the optimal taxon set and detailed information about the set. More information can be found at http://www.cibiv.at/software/pda.

4.3 Analysis of simulated data

4.3.1 Simulations set up

Since ILP is NP-hard (e.g., Karp 1972), to see whether ILP can solve the viable taxon selection problem in feasible time, we first tested its performance using simulated data.

We constructed split systems using splits from random Yule-Harding (Harding 1971) trees. To vary the complexity, which is determined by the number of splits in the system, we varied the number of trees used to build one split system from 1 to 4. These random trees had the same (fictional) species sets. Therefore, no algorithm is necessary to build a split system from them. We simply collected all bipartitions from t generated trees (1, 2, 3 or 4). Since the trees are random, they will exhibit a lot of incompatible splits. Therefore, a split system built from splits of several random trees will be characterized by a much larger number of splits compared to the number of edges/splits from one tree. For convenience, we sometimes refer to split system by its representation, namely, split network.

To generate random food webs we used a Cascade model (Cohen and Newman 1985). In this model, the species are ranked from 1 to N, where N is the total number of species, and the predators are only allowed to feed on the species of lower rank. The resulting food web is a non-weighted directed acyclic graph. The complexity of the food web, defined as linkage density, is measured by the connectance C

$$C = \frac{L}{N^2},$$

where L denotes the total number of links in the food web. We varied C from 0.1 to $\frac{N-1}{2N}$, where the latter corresponds to fully connected food web with $\binom{N}{2}$ links.

We also varied the number of species in the split network and food web from 10 to 250. The number 250 well resembles a maximum size of real food webs, since resolving a large food web is non-trivial. For each combination (the number of trees to construct a split system, the food web connectance and the species number) we generated 100 instances.

The size of the optimal subset, k, was varied from 10% to 90% of the total number of species.

We summarized all the settings in Table 4.2.

Parameters	Min	Max	Step	Additional values	The number of cases	Number of test sets
Species number, N	10	250	+10	-	25	100 per combination
Number of trees used to build a Split System, t	1	4	+1	-	4	
Connectance of the Food Web, C	0.1	0.4	+0.1	$\frac{N-1}{2N}$ (Fully connected)	5	
Subset size, k	10%	90%	+20%	-	5	

Table 4.2. Simulations set up

4.3.2 Results

To evaluate the performance of ILP, we recorded the CPU time spent by GUROBI on solving viable taxon selection problem. For each combination of parameters we computed the average CPU time of 100 runs.

The increase in complexity of a split system caused an increase of run time (Fig. 4.1). This is most apparent for the small subset sizes k (10% and 30%). In contrast to this, the increase in complexity of a food web led to decrease in run time, with the smallest average run time obtained for fully connected food webs (bottom plots in Fig. 4.1).

Most importantly, simulations with varying complexities of split system and food webs yielded average run times of less than 1.25 seconds on a 2.66 GHz computer, with a maximum of 9 seconds[1].

The simulations above were only used to analyse ILP performance in solving viable taxon selection problem (problem 2), but as we will see from the analysis of real data, also the $d\%$-viable taxon selection (problem 3) is solved within seconds. We do not exclude the possibility that there might be some difficult cases when ILP might become time consuming, but it is difficult to predict or describe such cases. From the different settings we tested, one could expect that with the current state-of-the-art optimizers the majority of ILP problems of viable and $d\%$-viable taxon selection will be solved within plausible time.

[1] this maximum was obtained for one test set with $k = 10\%, C = 0.1$ and a split system built from splits of 4 random Yule-Harding trees

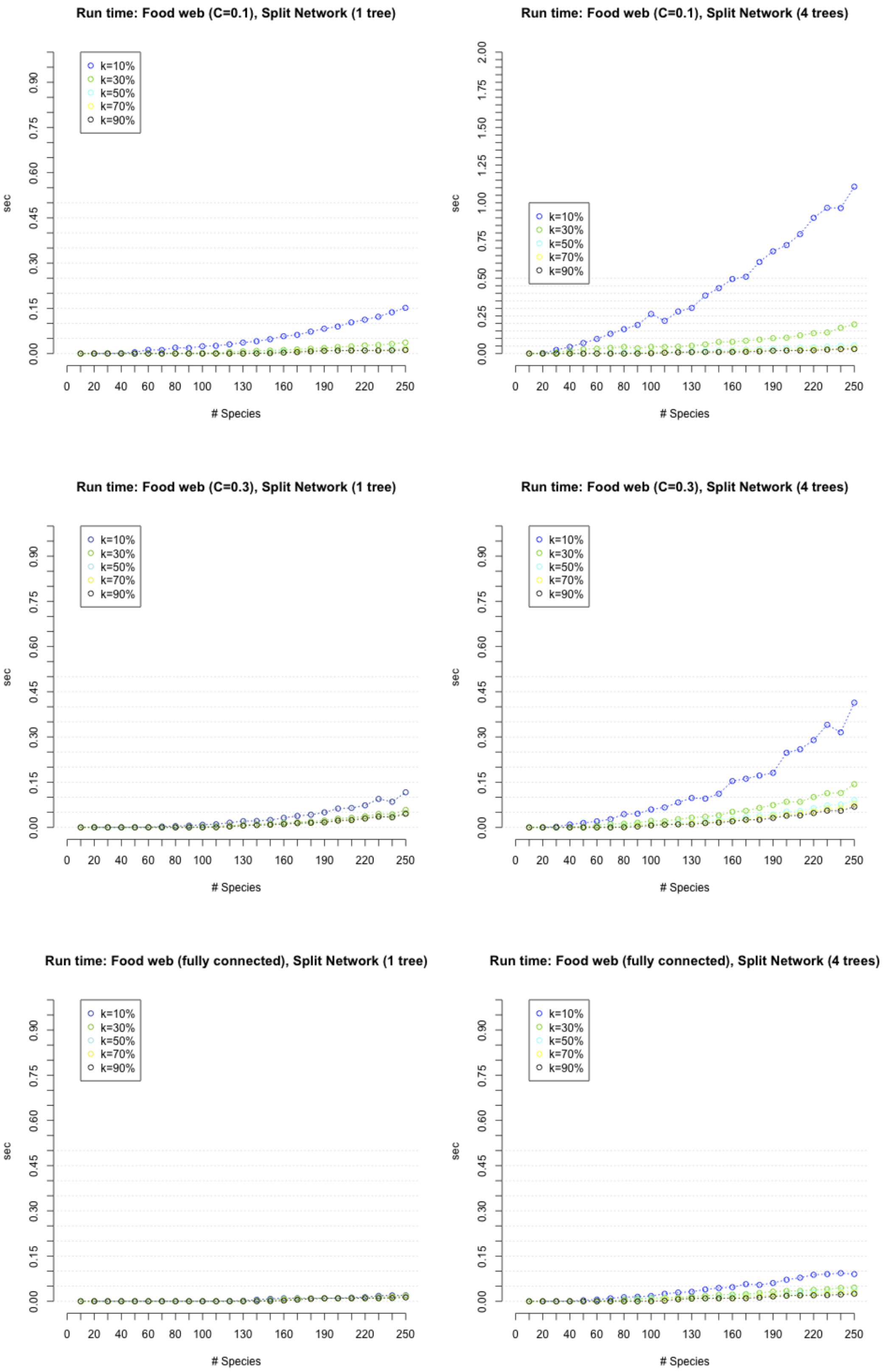


Figure 4.1. The average run times used by GUROBI to solve ILP problem of viable taxon selection. The complexity of split systems: only the simplest (1 input tree, left column) and the most complex (4 input trees, right column) cases tested are shown. The complexity of food webs increases from top to bottom: simplest case – connectance 0.1 (top), connectance 0.3 (middle), and fully connected food web (bottom) The x-axis is the total number of species. The y-axis is the run time in seconds. Note that the upper most right plot has an increased y-axis scale compared to the other plots.

4.4 Application to Caribbean coral reef community

In this section we will demonstrate on the Caribbean coral reef community data how predator-prey interactions can be incorporated in the analysis used for conservation prioritization. To our knowledge, the food web representing the relationships between species of this community is the largest available. Also, Caribbean Islands are one of the world's biodiversity hotspots (Mittermeier, et al. 2005; Myers, et al. 2000). Therefore, this case study combines complex ecological input together with a high value for biodiversity studies.

4.4.1 Data description

An input for viable taxon selection problems consists of ecological and phylogenetic components. First, lets consider the former one.

- *Food web and diet composition*

The predator-prey interactions of Caribbean coral reef community are represented by a well-resolved marine food web (Opitz 1996).

This data set consists of 250 taxonomic units including 208 fish species, a category "unidentified fishes", 35 non-fish taxa, two aggregated groups of consumers (zooplankton and microfauna) and four categories of primary producer: organic matter (including particulate organic matter, dissolved organic matter and detritus), benthic autotrophs, symbiotic algae and phytoplankton. Since for unidentified fishes phylogenetic information is not available, it was removed from the food web, as well as *Synodus synodus*, which only feeds on this group.

The food web is represented as a weighted directed graph, where the nodes in the graph correspond to the taxa and the arrows point from predators to their preys. Arrow weights are defined as the fraction a prey contributes to the diet of the predator. The food web contained 20 cycles including 11 cases of cannibalism and 9 cycles of two taxa (mutual predation). Therefore, 20 arrows were excluded to obtain a directed acyclic graph. The reduced food web has 248 nodes and 3281 arrows. Finally, the weights were rescaled such that for each predator the diet sums up to 100%.

Of the 248 nodes in the food web, all but the four basal nodes depend on consumption of at least one other taxon, and all but one (tiger shark, *Galeocerdo cuvier*) is prey for at least one other taxon. The food web is characterized by a complex structure and extensive omnivory, with food chains of as many as 25 links. Thus, this ecological network features extensive and complex dependencies among species that must be accounted for if we are to select a viable subset of taxa.

- *Phylogenomic inference*

For each of the 242 taxa we retrieved if available (Table 4.3) four ribosomal RNA genes (*12S rRNA, 16S rRNA, 18S rRNA, 28S rRNA*), cytochrome c oxidase I (*COI*) and cytochrome b (*CYTB*) from the NCBI database (Geer, et al. 2010). If the taxa corresponded to a genus or family, we chose one representative species having most sequences from this genus or family. When none of the genes were available for a given species, we substituted this species with its closest relative from the same genus (or family).

Genes	No. Sequences	No. Sites	Substitution model
12S rRNA	115	1328	GTR+G
16S rRNA	171	2680	GTR+I+G
18S rRNA	24	2887	TN+G
28S rRNA	51	4791	GTR+G
Cytochrome c oxidase I (COI)	218	1641	GTR+I+G
Cytochrome b (CYTB)	130	1173	TVM+I+G
Supermatrix	242	14500	GTR+G

Table 4.3. Sequence data collected for Caribbean coral reef community.

For each rRNA gene we aligned sequences using MAFFT L-INS-i v6.935b (Katoh and Toh 2008). For COI and CYTB we used TranslatorX (Abascal, et al. 2010) and MAFFT to align the translated amino acid sequences and then back translate into cDNA alignments.

Finally, we reconstructed a maximum-likelihood (ML) tree for each gene using IQ-TREE (Nguyen, et al. 2015), where the substitution models were selected by the Bayesian information criterion (Schwarz 1978). The gene trees served as input to infer a split system (Σ, λ) (Appendix C Fig. C.1) using SplitsTree4 (Huson and Bryant 2006; Huson, et al. 2004). We also computed the ML tree T (Appendix C Fig. C.2) from the concatenated gene alignments (supermatrix).

The tree T and the split system (Σ, λ) were used to compute PD and SD, respectively. The split system (Σ, λ) contains 558 non-trivial splits (i.e., splits that contain at least two taxa on either side), which is 2.3 times more splits than T. This indicates that the 6 gene trees are very incongruent. This incongruence has a number of potential causes, including insufficient phylogenetic information, noise in the alignments or even non-treelike evolution (Doolittle 1999; Philippe, et al. 2011).

4.4.2 Results

We now discuss the optimal taxon sets obtained under different constraints. We require that the aggregate trophic groups are always included in the optimal sets, because they are at the base of the food web and because they represent taxonomically diverse collections of organisms rather than defined taxa.

- *Taxon selection under PD vs. SD*

We first compare the results of taxon selection under PD and SD measures without taking into account the food web. Such problems are just the simplifications of problem 1 and 2 obtained by ignoring viability constraint. In such a way one can observe the differences and the impact of using either measure on the final solution.

Let S_{PD} denote a set of k taxa having maximal PD and S_{SD} a set of k taxa having maximal SD. We obtained S_{PD} by the greedy algorithm (Minh, et al. 2006), and S_{SD} using ILP (Minh, et al. 2010). For $k = 24$ (10% of the taxa), S_{PD} (Fig. 4.2: red and blue nodes, food web restricted to S_{PD} contains red and blue arrows) has a relative PD of 36.12% of the total branch lengths of phylogenetic tree. That is, 10% of the species in Caribbean food web conserves 36.12% of its evolutionary history based on phylogenetic tree.

In the split system the optimal subset S_{SD} (Fig. 4.2: green and blue nodes, food web restricted to S_{SD} contains green and blue arrows) with 24 taxa conserves 57.89% of the total SD (i.e. the sum of the all split weights).

S_{PD} and S_{SD} only share 11 taxa (Fig. 4.2: blue nodes). The other 13 species in each set are different and belong to different genera (Fig. 4.2: red and green nodes for species exclusively in S_{PD} and S_{SD}, respectively).

Note that *Synodus foetens* (in S_{PD}) and *Antennarius striatus* (in both sets S_{PD} and S_{SD}) have no outgoing arrows in Fig. 4.2 (also indicated by 0 entries for corresponding taxa in Table 4.4). This renders S_{PD} and S_{SD} inviable, and illustrates the need to include information about ecological dependencies in conservation decisions.

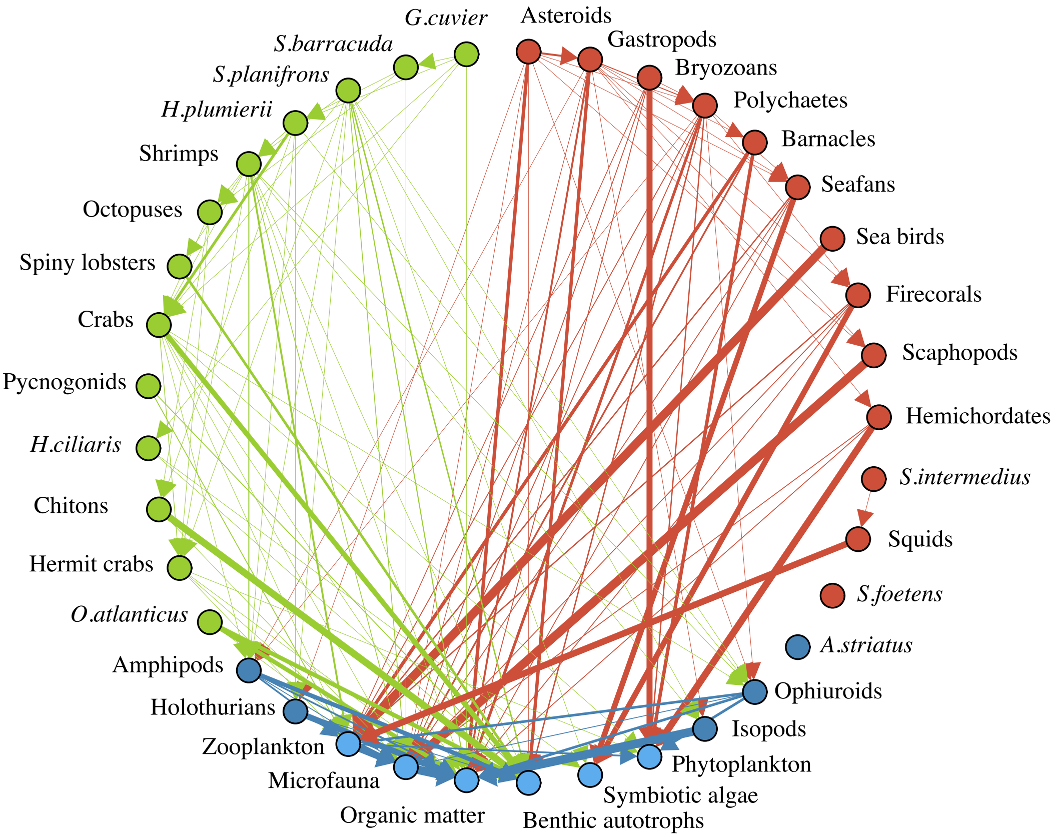


Figure 4.2. Food web restricted to only those taxa present in S_{PD} or S_{SD} (see main text and Table 4.4). Red, green, and blue nodes depict the taxa present exclusively in S_{PD}, exclusively in S_{SD}, and in both sets, respectively. Light blue nodes correspond to aggregated groups. Arrows connect from predators to their preys with thickness reflecting the prey proportion in the predator diet. Arrows pointing to or from green and red nodes are coloured green and red respectively. Arrows between blue nodes are coloured blue. Note that the arrows between green and red nodes are ignored.

Table 4.4. List of taxa in optimal sets S_{PD}, S_{SD}, S_1, S_2 to preserve 24 (10%) taxa. The numbers in parentheses depict the diversity percentages of the set compared to the total diversity. Table entries show the sum of diet proportions for each taxon over other taxa in the same set. Minus signs indicate the absence of the taxa from the corresponding set. The entries in red and orange indicate non-viable taxa or taxa with small diet composition preserved by their preys in the corresponding set.

	Taxon name	S_{PD} (36.12%)	S_{SD} (57.89%)	S_1 (57.67%)	S_2 (56.36%)
Required to be included in all optimal sets (see main text).	Organic matter	0	0	0	0
	Benthic autotrophs	0	0	0	0
	Symbiotic algae	0	0	0	0
	Phytoplankton	0	0	0	0
	Zooplankton	1	1	1	1
	Microfauna	1	1	1	1
Taxa selected by the greedy algorithm (s_{PD}) or ILP (all the remaining sets).	*Ophioblennius atlanticus*	-	1	1	1
	Acanthopleura granulata	-	0.84	0.84	-
	Quadrimaera pacifica	1	1	1	1
	Haemulon plumierii	-	0.5	0.5	0.5
	Holothuria floridana	1	1	1	1
	Achelia sawayai	-	0.2	0.2	0.35
	Sphyraena barracuda	-	0.06	0.06	-
	Galeocerdo cuvier	-	0.11	0.11	-
	Stegastes planifrons	-	0.58	0.58	0.58
	Elacatinus evelynae	-	-	1	-
	Holacanthus ciliaris	-	0.02	0.02	0.98
	Panulirus argus	-	0.3	0.3	0.7
	Stenopus hispidus	-	0.66	0.66	0.67
	Ophiocoma echinata	1	1	1	1
	Octopus vulgaris	-	0.07	0.07	-
	Idotea baltica	1	1	1	1
	Pagurus longicarpus	-	0.2	0.2	1
	Callinectes sapidus	-	0.81	0.81	0.86
	Carcharhinus leucas	-	-	-	0.3
	Sphyraena picudilla	-	-	-	0.69
	Pinctada radiata	-	-	-	1
	Iotrochota birotulata	-	-	-	1
	Sepioteuthis sepioidea	0.75	-	-	1
	Reteporella beaniana	1	-	-	-
	Synodus foetens	0	-	-	-
	Fregata magnificens	1	-	-	-
	Cittarium pica	0.86	-	-	-
	Synodus intermedius	0.05	-	-	-
	Loimia medusa	0.93	-	-	-
	Millepora sp. AMN-2008	1	-	-	-
	Cephalodiscus gracilis	1	-	-	-
	Megabalanus californicus	1	-	-	-
	Oreaster reticulatus	0.83	-	-	-
	Graptacme eborea	1	-	-	-
	Gorgonia flabellum	1	-	-	-
	Antennarius striatus	0	0	-	-

- *Taxon selection under SD vs. viable taxon selection under SD*

Maximizing PD or SD without taking into account the food web leads to inviable sets, where *Synodus foetens* and *Antennarius striatus* do not find prey (nodes with no outgoing arrows in Fig. 4.2). Therefore, in the following we require that the optimal SD set is viable (i.e. each predator must have at least one prey in the set, problem 2).

The resulting optimal set denoted S_1 (Fig. 4.3: red and blue nodes) containing 10% of the taxa has a relative SD of 57.67% compared with the total SD of all taxa (i.e., only 0.22% less than the taxon-set chosen solely on SD). This loss in relative diversity is due to the replacement of *Antennarius striatus* (the "non-viable" species in S_{SD}) with *Elacatinus evelynae.* Therefore S_1 "repairs" the viability with negligible loss of diversity.

Lets now look in more detail at the diet composition of the species in S_1. For each predator $s_j \in S_1$ we compute its proportion of diet conserved in S_1: $\delta(s_j|S_1) = \sum_{s_i \in S_1} w_{ij}$, where w_{ij} is the proportion of prey s_i in the diet of s_j (see Methods, section 2). For the taxa *Galeocerdo cuvier*, octopuses, *Sphyraena barracuda* and *Holacanthus ciliaris* (Fig. 4.3: four red nodes) only 11%, 7%, 6% and 2% of their diet is conserved (Table 4.4, entries in red), while for pycnogonids, hermit crabs, spiny lobsters and *Haemulon plumierii* the diet proportion conserved ranges between 20% and 50% (Table 4.4, entries in orange).

- *Viable vs. d%-viable taxon selection under SD*

The above observations indicate that the simple viability constraint (conserving at least one prey per predator) might result in some predators having an insufficient availability and variety of prey. To address this, we applied the d%-viability constraint, which requires that every taxon in the optimal SD set must have at least d% of its diet composition conserved (Methods, problem 3). Note that the d%-viability constraint reduces to the simple viability constraint if we set $d = \epsilon$ (i.e., a very small number). Despite this additional constraint, the problem of d%-viable taxon selection can still be solved by ILP.

As an illustration, for $k = 24$ and $d = 30\%$, the optimal set S_2 (Fig. 4.3: green and blue nodes) has a relative SD of 56.36%, a reduction of 1.31% compared with S_1.

Set S_2 has five species (Fig. 4.3: green nodes) not present in S_1. At the same time five taxa (Fig. 4.3: red nodes) are not present in S_2, of which *G. cuvier,* octopuses, and *S. barracuda* have less than 30% diet conserved by their preys in S_1 (Table 4.4). On the other hand, pycnogonids, *H. ciliaris,* and hermit crabs, which have less than 30% diet conserved in S_1, now become 30%-viable in S_2 (Table 4.4). This is because the newly added taxon (bivalves; Fig. 4.3: green node) is a prey of hermit crabs, contributing 80% to their diet. Another new taxon (sponges) being a prey of pycnogonids and *H. ciliaris* contributes to their diets 15% and *97%* respectively, making them 30% -viable (Table 4.4).

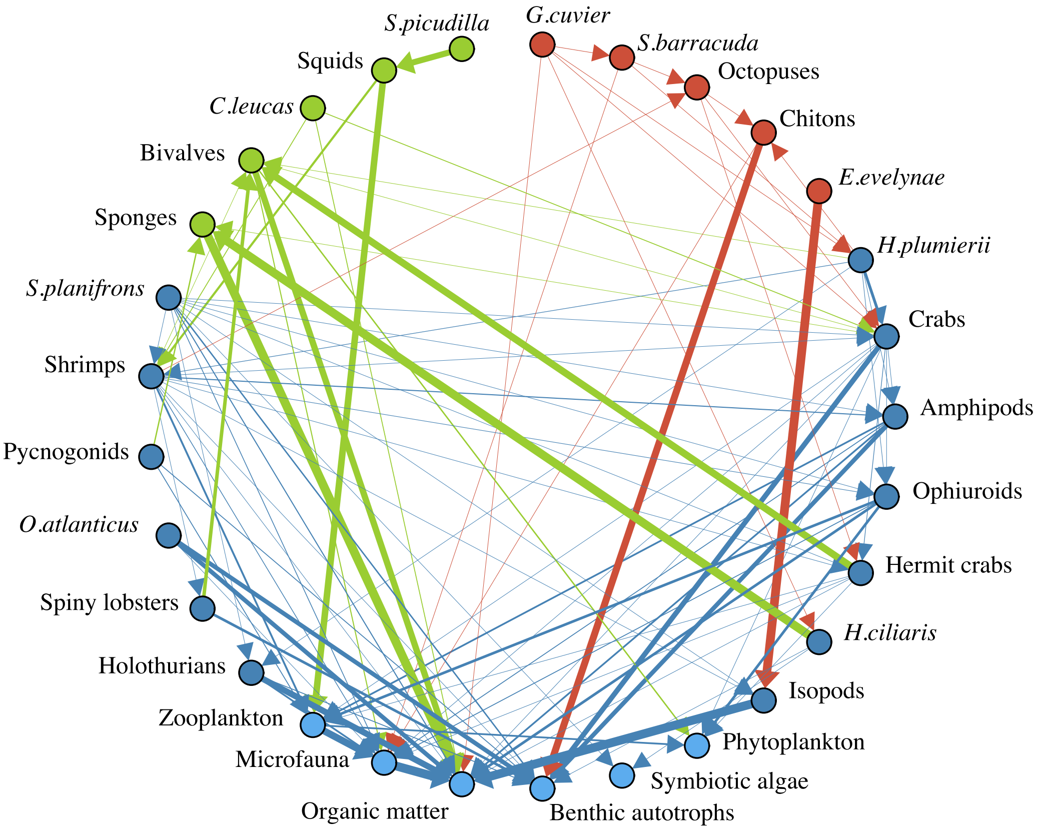


Figure 4.3. Food web restricted to only those taxa present in S_1 or S_2 (see main text and Table 4.4). Red, green, and blue nodes depict the taxa present exclusively in S_1, exclusively in S_2, and in both sets, respectively. Light blue nodes correspond to aggregated groups. Arrows connect from predators to their preys with thickness reflecting the prey proportion in the predator diet. Arrows pointing to or from green and red nodes are coloured green and red respectively. Arrows between blue nodes are coloured blue. Note that the arrows between green and red nodes are ignored.

- *Analysis of SD reduction for increasing d*

As it was discussed above the SD of S_2 has a reduction of only 1.31% compared with S_1. To see the influence of parameter d on the SD of optimal set, we exhaustively computed all optimal SD sets $S_{max}(k, d)$ for $k \in \{10\%, 20\%, \ldots, 90\%\}$ and $d \in \{0\%, 1\%, \ldots, 90\%\}$. Figure 4.4 shows the optimal SD of all these sets. For a fixed k the changes in the SD of $S_{max}(k, d)$ with varying d are moderate as indicated by almost horizontal lines in Figure 4.4. The largest difference is 4.63% for $k = 10\%$ and d increasing from 0% to 90%. This means, that even stringent viability constraints with high d still provide us with almost equally optimal subsets based on SD value.

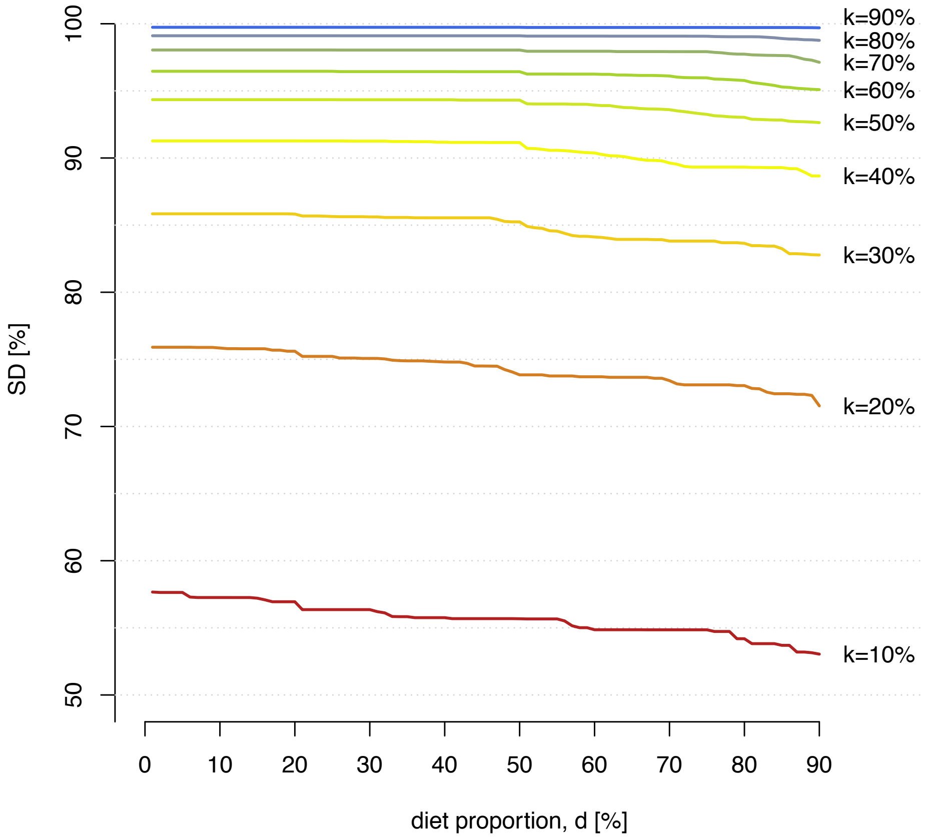


Figure 4.4. Dependence of SD on the subset size, k, and the diet portion, d. SD of the optimal set $S_{max}(k, d)$ for varying k and d.

- *Computational time*

98% of the runs with PDA software for different parameters of problems 2 and 3 applied to Caribbean coral reef community consumed less than 1 second on a 2.66 GHz computer. The maximum run time of the remaining 2% was 3 seconds.

4.5 Discussion

In this chapter we presented an application of phylogenomics in conservation prioritization. We provided ILP solutions for biodiversity optimization problems that incorporate phylogenetic information and also include ecological constraints. Here, one is interested in selecting a subset of species, which maximizes PD or SD, subject to the so-called viability constraints. The viable taxon selection problems are NP-hard. Nevertheless, thanks to the powerful GUROBI solver employed, the analyses were carried out within seconds. ILP is flexible and allows for modelling various constraints while using the same basic formulations. It is computationally efficient and guarantees optimal solution. Therefore, the PDA software provided here is complementary to existing conservation prioritization tools.

We exemplified viable taxon selection problems with a large-scale case study: the Caribbean reef community. This study demonstrates the usage of viability constraints in conservation prioritization thanks to the availability of food web and diet composition data. Such food webs allow us to analyse an entire set of species as an interaction network rather than as isolated units. We found that in the case of the Caribbean food web, including viability constraints results in only small reductions in the amount of biodiversity that can be preserved. This is explained by the fact that the most evolutionarily distant taxa are concentrated on the low trophic levels of the food web. Therefore, by maximizing PD or SD for the Caribbean community we already obtained almost viable sets. However, taxon selection based on viability also highlighted which representatives of each subclade contribute to viability of the set. In practice, incorporating viability constraints has the potential to prevent the use of limited resources on specialist taxa unless a sufficient resource base to support them is also preserved.

While the incorporation of predator-prey links and diet composition gets us closer to ecological realism, there are nonetheless many factors that are not accounted for in the examples described here. First, we are only considering predator-prey relationships, and not other interaction types such as mutualism, facilitation or interference competition (Kefi, et al. 2012). This framework should be applicable to other types of interaction networks, such as mutualism networks, that allow viability criteria to be specified. For example, a viable taxon may require the preservation of at least one mutualist partner (i.e., partners sufficient to contribute a certain fraction of mutualist benefit). We also consider only the bottom-up dependencies within food webs, not top-down effects of predators on their prey (e.g. apparent competition, trophic cascades). The proper incorporation of the complexity of interactions that result from top-down effects may require a move from a static representation of a food web to a population-dynamic model that explicitly includes extinction due to population decline (Ebenman, et al. 2004). However, this is beyond the scope of this work.

One may also need to consider how and if it is appropriate to incorporate diet composition to ensure that each taxon has at least *d*% of its food base preserved. Most published food webs contain only a topological representation of predator-prey relationships, and large food webs such as the Caribbean data set that include weights representing energy flow or diet composition are rare. However, even in the absence of diet composition data, one has the option of assigning the links between a predator and each of its *n* prey a weight of $1/n$, assuming that they are of equal importance. This allows the application of additional criteria; for example, that a viable predator must have access to at least 50% of its prey taxa. Where diet composition data are available, they provide a means of indirectly considering taxon abundances, as more abundant taxa will generally make up a greater proportion of their predators' diets. Further, if some prey types are only available during certain seasons, one could devise "seasonal constraints" ensuring that some prey taxa are present for every season. Finally, one could consider contributions from prey that do not appear in the taxon set (e.g. prey consumed outside the spatial area covered by the food web data) by crediting these predators with some proportion of their prey intake regardless of the taxon set selected. Such additional constraints can be easily formulated under the ILP framework.

The set of taxa returned by the optimization procedure is a starting point for conservation planning, but should be followed by consideration of the biology of the selected taxa. A food web is a simplified representation of a community or meta-community, and lacks information that might bear on the suitability of the taxon set. For example, it should be confirmed that the prey taxa predicted to support each predator are sufficiently abundant and widespread to do so, or that they can reasonably be expected to become more common as a result of conservation action. If the food web contains errors, such as a link between taxa that no longer co-occur or the omission of an important link, it might lead to sub-optimal taxon selection. Further, taxa may be subject to additional constraints that may be difficult to capture in the ILP, so at times it may be necessary to reconsider the taxa targeted for conservation action in light of additional biological or societal information.

The importance of preserving the diversity of life is widely recognized and understood. In an ideal world we could ensure the persistence of all levels of biodiversity, but with limited resources the prioritization of some taxa or ecosystems is unavoidable. We thus need good criteria with which to apply triage, to prioritize the allocation of these resources to maximize conservation return under budget constraints (Bottrill, et al. 2008). We have demonstrated the utility of the ILP approach to show how sensible and objective conservation decisions can be made while accounting for the ecological constraints. The evaluation of different future scenarios with the aid of the ILP approach presented here will certainly prove to be a valuable contribution to conservation planning in a changing world.

CHAPTER 5

Conclusions

The complexity of phylogenomics raises questions on many different levels of phylogenetic inference. A better theoretical understanding as well as the improvement of existing algorithms is essential for the accurate and efficient analysis of multi-gene data sets.

Here, we first focus on developing the rules to detect problematic structures like *phylogenetic terraces* (Sanderson, et al. 2011), which are caused by missing data. To be useful in phylogenetic inference tools the identification of terraces should be quick. Therefore, the naïve pair-wise comparison of all the induced partition trees associated with the two compared species trees is not efficient to be applicable in the tree search algorithms. Instead we proposed to detect the changes of induced partition trees as the consequences of topological rearrangements applied to a species tree. We concentrated on three most common topological operations: NNI, SPR and TBR.

Two trees are said to be equivalent if they share all the splits (bipartitions) defined by their edges. Therefore, the key point in the derivation of necessary conditions is the analysis of different splits between two *consecutive* species trees. Here, *consecutive* refers to two trees, where one tree is obtained from another by only one topological rearrangement. The splits of species tree (*supersplits*) correspond to induced *subsplits* on associated induced partition trees. The idea is to find the conditions, under which the supersplits that are different between two species trees correspond to subsplits that are shared by the two associated induced partition trees. In this case, the two partition trees are the same, i.e. the rearrangement applied to a species tree did not change the induced partition tree. If all the induced partition trees remain unchanged, then two consecutive species trees belong to one terrace.

In such a way, we provided the rules to quickly identify terraces for NNI, SPR and TBR-based tree searches.

Moreover, we generalized the concept of terraces to *partial terraces*. As we showed with real alignments partial terraces occur more often than *full* terraces. Therefore, accounting for partial terraces is an important aspect for the efficient phylogenomic inference in the presence of missing data.

Next, we proposed a phylogenetic terrace aware (PTA) data structure, which facilitates the detection of (partial) terraces during tree search. PTA consists of the species tree, a set of induced partition trees and the maps from the species tree edges into the edges of each partition tree. We provided a dynamic programming algorithm to build this data structure. The algorithm is linear in the number of species and the number of partitions in the alignment. First it maps all the external edges of the species tree and then proceeds towards internal edges. The update and maintenance of PTA requires negligible extra computations.

In order to use the rules developed for the detection of partial terraces they had to be reformulated in terms of PTA maps.

We implemented PTA in IQ-TREE together with the reformulated rule to detect partial terraces for NNI-based tree searches. We analysed 11 real alignments with varying amount of missing data. The comparison with the standard implementation and RAxML showed that accounting for partial terraces and the usage of induced partition trees led to a speedup of up to 5 and 6 times, respectively. PTA can be employed with all the partition models (Edge-Linked joint and proportional, and Edge-Unlinked) and all common topological rearrangements (NNI, SPR, and TBR). It is therefore a versatile and general structure for the efficient supermatrix analysis.

The implementation of PTA and the corresponding rules to detect partial terraces into the *phylogenetic likelihood library* (Flouri, et al. 2015) could be one of the next steps in this project. Further work could also focus on the choice of topological rearrangements to be applied for the controlled exploration of the tree space. During tree search one can choose topological rearrangements that perturb the species tree the most, such that all partition trees are affected by the corresponding operations applied to the species tree. From another side, one could be also interested in topological rearrangements applied to one or several partition trees with no effect on all the other partition trees. Different strategies could potentially improve the exploration of the tree space.

The last contribution of this thesis was the development of methods for phylogenomic application in conservation biology. We discussed the viable taxon selection problem and provided Integer Linear Programming (ILP) formulations for two extensions of this problem: the generalization to Split Diversity and the extension to account for the diet composition of predators. We showed that ILP solves the problems within seconds thanks to the powerful GUROBI library. We also analysed the Caribbean Coral Reef community data, which provided the largest resolved food web.

We note that the ILP framework is extensible to other diversity measures provided that the measures can be expressed as a linear function. A possible extension to the viable taxon selection problems is to choose species under budgetary constraints. Here, each species has a conservation cost and the inclusion of the taxon is constraint by the budget. Close collaboration between conservation biologists and mathematicians is recommended to convert complex conservation problems into an ILP framework.

In summary, the work presented in this thesis contributes to

(i). A better theoretical understanding of partial terraces and their detection based on the topological rearrangements applied to a species tree;

(ii). The improvement of phylogenetic inference from supermatrices, by using the intrinsic characteristics of the tree space structure imposed by the missing data to save computation time;

(iii). Phylogenomics application in conservation prioritization, by providing the solutions to several viable taxon selection problems and their implementation in PDA software package.

Acknowledgements

Firstly, I would like to thank my supervisor Arndt von Haeseler for this great opportunity to challenge myself in fascinating field of bioinformatics. I am grateful for his constant support and encouragements, for all the ideas and discussions that influenced my work and made this thesis possible. I also would like to thank Arndt for all the advices that made my thoughts and writing less chaotic, improved my presentation and discussion skills and helped to slow down the speed of my talks.

I am taking this opportunity to thank Monika Henzinger for all the discussions, suggestions and helpful feedbacks on my projects. I am also grateful to Ivo Hofacker for his interest and comments on my work throughout IK meetings.

I would like to thank my *subboss* and a great friend Minh, who always had an answer to all my questions, who introduced me to the world of IQ-TREE code, spending his time helping me to debug it, and who always could rephrase my way too long sentences to make them clear and concise. Many thanks for all your help, support and motivation.

Financial support from the University of Vienna (Initiativkolleg Computational Sciences) and the Austrian Science Fund – FWF is greatly appreciated.

I am grateful to all CIBIV members for a friendly atmosphere and helpful feedbacks during our Journal Clubs and Monday Seminars.

I thank Heiko for the fruitful discussions and feedbacks, Jana for the help with Zusammenfassung, Tung for being a great office-mate, for his help with IQ-TREE and long life talks, Milica, Celine, Luis, Connie and Florian for all the chats, dancing and funny moments, Philipp for all the discussions of happiness and running events, Misha and Steven for the awesome game evenings and your encouragements to speak German. Thank you guys for all the talks, parties, sports and cocktail tours we had!

I would like to thank Alessandra and Olga for our nice chats over a cup of coffee (or wine), for yoga classes, dinners out and crazy dancing during MFPL happy hours. I appreciate the time we spent together.

A huge thank you to Masya, Slon, Nata and Bezz for all the fun and adventures that helped to reboot my brain for further work.

Finally, I would like to thank my family: my Grandparents, Mum, Dad, Yulia, Liudmyla, Yuriy, Jaroslav and Marina for their encouragements and belief in me. A special thank you to Ievgen for his love and constant support, for being there to encourage, motivate, to give an advice and simply to cheer me up, when I was sad or to share my happiness on sunny days.

Curriculum Vitae

Olga Chernomor

Center for Integrative Bioinformatics Vienna (CIBIV)
Max F. Perutz Laboratories
Dr. Bohr Gasse 9
1030 Vienna, Austria
Phone: +43 1 4277 74325
Email: olga.chernomor(at)univie.ac.at

Education

- **2009-2011:** Master in Mathematical Engineering in Life Sciences, University of L'Aquila (Italy) / University of Nice - Sophia Antipolis (France)
- **2008-2009:** Master in Mathematics, Oles Honchar Dnipropetrovsk National University (Ukraine)
- **2004-2008:** Bachelor in Mathematics and basics of Economics, Oles Honchar Dnipropetrovsk National University (Ukraine)

Research Experience

- **10/2011-present:** PhD student at the Center for Integrative Bioinformatics Vienna (CIBIV)
- **04/2011-07/2011:** Collaboration with group of Computational Biophysics, Biochemistry and Chemistry (CBBC), La Sapienza-University of Rome

Publications

(i) Chernomor, O., Minh, B.Q. and von Haeseler, A. (2015) Consequences of Common Topological Rearrangements for Partition Trees in Phylogenomic Inference, *Journal of Computational Biology*. (DOI: 10.1089/cmb.2015.0146)

(ii) Chernomor, O., *et al.* (2015) Split diversity in constrained conservation prioritization using integer linear programming, *Methods in ecology and evolution / British Ecological Society*, **6**, 83-91. (DOI: 10.1111/2041-210X.12299)

(iii) Chernomor, O., *et al.* (2016) Split diversity: measuring and optimizing biodiversity using phylogenetic split networks. In Pellens, R. and Grandcolas, P. (eds), *Biodiversity Conservation and Phylogenetic Systematics*, Series: Topics in Biodiversity and Conservation, Springer International Publishing, in press.

(iv) Coccia, E., Chernomor, O., Barborini, M., Sorella, S., and Guidoni, L.. (2012) Molecular Electrical Properties from Quantum Monte Carlo Calculations: Application to Ethyne. *J. Chem. Theory Comput.,* 8(6), pp 1952-1962. (DOI: 10.1021/ct300171q)

Bibliography

Abascal F, Zardoya R, Telford MJ. 2010. TranslatorX: multiple alignment of nucleotide sequences guided by amino acid translations. Nucleic Acids Research 38:W7-W13.

Bapteste E, O'Malley MA, Beiko RG, et al. 2009. Prokaryotic evolution and the tree of life are two different things. Biol Direct 4:34.

Bininda-Emonds OR, Gittleman JL, Purvis A. 1999. Building large trees by combining phylogenetic information: a complete phylogeny of the extant Carnivora (Mammalia). Biol Rev Camb Philos Soc 74(2):143-75.

Bininda-Emonds ORP, Gittleman JL, Steel MA. 2002. The (Super)tree of life: Procedures, problems, and prospects. Annual Review of Ecology and Systematics 33:265-289.

Bottrill MC, Joseph LN, Carwardine J, et al. 2008. Is conservation triage just smart decision making? Trends in Ecology & Evolution 23(12):649-54.

Bouchenak-Khelladi Y, Salamin N, Savolainen V, et al. 2008. Large multi-gene phylogenetic trees of the grasses (Poaceae): progress towards complete tribal and generic level sampling. Mol Phylogenet Evol 47(2):488-505.

Brown S, Savage PE, Ko AM, et al. 2014. Correlations in the population structure of music, genes and language. Proc Biol Sci 281(1774):20132072.

Burleigh JG, Kimball RT, Braun EL. 2015. Building the avian tree of life using a large-scale, sparse supermatrix. Molecular Phylogenetics and Evolution 84:53-63.

Cadotte MW, Dinnage R, Tilman D. 2012. Phylogenetic diversity promotes ecosystem stability. Ecology 93(8):S223-S233.

Chambers JK, Macdonald LE, Sarau HM, et al. 2000. A G protein-coupled receptor for UDP-glucose. J Biol Chem 275(15):10767-71.

Chernomor O, Minh BQ, von Haeseler A. 2015. Consequences of Common Topological Rearrangements for Partition Trees in Phylogenomic Inference. Journal of Computational Biology:accepted.

Cocks KD, Baird IA. 1989. Using mathematical-programming to address the multiple reserve selection problem - An example from the Eyre Peninsula, South-Australia. Biological Conservation 49(2):113-130.

Cohen JE, Newman CM. 1985. A Stochastic-Theory of Community Food Webs .1. Models and Aggregated Data. Proceedings of the Royal Society Series B-Biological Sciences 224(1237):421-448.

CPLEX. IBM ILOG CPLEX Optimizer [Internet]. Available from: http://www.ibm.com/software/integration/optimization/cplex-optimizer/

Crozier RH. 1992. Genetic diversity and the agony of choice. Biological Conservation 61(1):11-15.

De Queiroz A, Donoghue MJ, Kim J. 1995. Separate Versus Combined Analysis of Phylogenetic Evidence. Annual Review of Ecology and Systematics 26:657-681.

de Queiroz A, Gatesy J. 2007. The supermatrix approach to systematics. Trends Ecol Evol 22(1):34-41.

Dell'Ampio E, Meusemann K, Szucsich NU, et al. 2014. Decisive data sets in phylogenomics: lessons from studies on the phylogenetic relationships of primarily wingless insects. Mol Biol Evol 31(1):239-49.

Delsuc F, Brinkmann H, Philippe H. 2005. Phylogenomics and the reconstruction of the tree of life. Nat Rev Genet 6(5):361-75.

Doolittle WF. 1999. Phylogenetic classification and the universal tree. Science 284(5423):2124-2128.

Dunn CW, Hejnol A, Matus DQ, et al. 2008. Broad phylogenomic sampling improves resolution of the animal tree of life. Nature 452(7188):745-9.

Ebenman B, Law R, Borrvall C. 2004. Community viability analysis: The response of ecological communities to species loss. Ecology 85(9):2591-2600.

Eisen JA. 1998. Phylogenomics: Improving functional predictions for uncharacterized genes by evolutionary analysis. Genome Research 8(3):163-167.

Fabre PH, Rodrigues A, Douzery EJ. 2009. Patterns of macroevolution among Primates inferred from a supermatrix of mitochondrial and nuclear DNA. Mol Phylogenet Evol 53(3):808-25.

Faith DP. 1992. Conservation Evaluation and Phylogenetic Diversity. Biological Conservation 61(1):1-10.

Faller B. 2010. Combinatorial and Probabilistic Methods in Biodiversity Theory. University of Canterbury.

Felsenstein J. 1978. Cases in which Parsimony or Compatibility Methods Will be Positively Misleading. Systematic Zoology 27(4):401-410.

Felsenstein J. 1981. Evolutionary Trees from DNA-Sequences - a Maximum-Likelihood Approach. Journal of Molecular Evolution 17(6):368-376.

Felsenstein J. 1985. Confidence-Limits on Phylogenies - an Approach Using the Bootstrap. Evolution 39(4):783-791.

Felsenstein J. 2004. Inferring Phylogenies. Sunderland, Massachusetts: Sinauer Associates.

Flouri T, Izquierdo-Carrasco F, Darriba D, et al. 2015. The Phylogenetic Likelihood Library. Systematic Biology 64(2):356-362.

Gaston KJ, Spicer JI. 2004. Biodiversity: An Introduction. Oxford: Blackwell Publishing.

Geer LY, Marchler-Bauer A, Geer RC, et al. 2010. The NCBI BioSystems database. Nucleic Acids Research 38:D492-D496.

Gomory RE. 1958. Outline of an algorithm for integer solutions to linear programs. Bulletin of the American Mathematical Society 64(5):275-278.

Gonzalez VL, Andrade SC, Bieler R, et al. 2015. A phylogenetic backbone for Bivalvia: an RNA-seq approach. Proc Biol Sci 282(1801):20142332.

Goodheart JA, Bazinet AL, Collins AG, et al. 2015. Relationships within Cladobranchia (Gastropoda: Nudibranchia) based on RNA-Seq data: an initial investigation. Royal Society Open Science 2(9).

Graur D, Li W-H. 2000. Fundamentals of Molecular Evolution. Sunderland, Massachusetts: Sinauer Associates.

Gu X, Fu YX, Li WH. 1995. Maximum likelihood estimation of the heterogeneity of substitution rate among nucleotide sites. Mol Biol Evol 12(4):546-57.

Guindon S, Dufayard JF, Lefort V, et al. 2010. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. Syst Biol 59(3):307-21.

GUROBI. 2012. Gurobi optimizer reference manual. 3.0 ed.

Haight RG, Snyder SA. 2009. Integer programming methods for reserve selection and design. In: Moilanen A, Wilson KA, Possingham HP, editors. Spatial Conservation Prioritization: Quantitative Methods and Computational Tools. New York: Oxford University Press. p. 28-42.

Harding EF. 1971. The probabilities of rooted tree shapes generated by random bifurcation. Advances in Applied Probability 3:44–77.

Hedtke SM, Patiny S, Danforth BN. 2013. The bee tree of life: a supermatrix approach to apoid phylogeny and biogeography. Bmc Evolutionary Biology 13.

Helaers R, Milinkovitch MC. 2010. MetaPIGA v2.0: maximum likelihood large phylogeny estimation using the metapopulation genetic algorithm and other stochastic heuristics. BMC Bioinformatics 11:379.

Hinchliff CE, Roalson EH. 2013. Using Supermatrices for Phylogenetic Inquiry: An Example Using the Sedges. Systematic Biology 62(2):205-219.

Hordijk W, Gascuel O. 2005. Improving the efficiency of SPR moves in phylogenetic tree search methods based on maximum likelihood. Bioinformatics 21(24):4338-47.

Huson DH, Bryant D. 2006. Application of phylogenetic networks in evolutionary studies. Mol Biol Evol 23(2):254-67.

Huson DH, Dezulian T, Klopper T, et al. 2004. Phylogenetic super-networks from partial trees. Ieee-Acm Transactions on Computational Biology and Bioinformatics 1(4):151-158.

Isaac NJB, Turvey ST, Collen B, et al. 2007. Mammals on the EDGE: Conservation Priorities Based on Threat and Phylogeny. Plos One 2(3).

Izquierdo-Carrasco F, Smith SA, Stamatakis A. 2011. Algorithms, data structures, and numerics for likelihood-based phylogenetic inference of huge trees. Bmc Bioinformatics 12.

Jobb G, von Haeseler A, Strimmer K. 2004. TREEFINDER: a powerful graphical analysis environment for molecular phylogenetics. Bmc Evolutionary Biology 4.

Jonsson KA, Fabre PH, Kennedy JD, et al. 2015. A supermatrix phylogeny of corvoid passerine birds (Aves: Corvides). Mol Phylogenet Evol 94(Pt A):87-94.

Jukes TH, Cantor CR. 1969. Evolution of protein molecules. Mammalian protein metabolism 3:21-132.

Jünger M, Liebling TM, Naddef D, et al. 2010. 50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art. 1 ed. Heidelberg, Germany: Springer.

Karp R. 1972. Reducibility among Combinatorial Problems. Complexity of Computer Computations. New York, USA: Springer. p. 85-103.

Katoh K, Toh H. 2008. Recent developments in the MAFFT multiple sequence alignment program. Briefings in Bioinformatics 9(4):286-298.

Kefi S, Berlow EL, Wieters EA, et al. 2012. More than a meal ... integrating non-feeding interactions into food webs. Ecology Letters 15(4):291-300.

Kolaczkowski B, Thornton JW. 2004. Performance of maximum parsimony and likelihood phylogenetics when evolution is heterogeneous. Nature 431(7011):980-4.

Kolaczkowski B, Thornton JW. 2008. A mixed branch length model of heterotachy improves phylogenetic accuracy. Molecular Biology and Evolution 25(6):1054-1066.

Kubatko LS, Degnan JH. 2007. Inconsistency of phylogenetic estimates from concatenated data under coalescence. Syst Biol 56(1):17-24.

Kumar S, Filipski AJ, Battistuzzi FU, et al. 2012. Statistics and Truth in Phylogenomics. Molecular Biology and Evolution 29(2):457-472.

Kupczok A, Schmidt HA, von Haeseler A. 2010. Accuracy of phylogeny reconstruction methods combining overlapping gene data sets. Algorithms for Molecular Biology 5.

Lanave C, Preparata G, Saccone C, et al. 1984. A New Method for Calculating Evolutionary Substitution Rates. Journal of Molecular Evolution 20(1):86-93.

Le SQ, Gascuel O. 2008. An improved general amino acid replacement matrix. Mol Biol Evol 25(7):1307-20.

Liu L, Xi Z, Wu S, et al. 2015. Estimating phylogenetic trees from genome-scale data. Ann N Y Acad Sci.

Liu L, Yu L, Kubatko L, et al. 2009. Coalescent methods for estimating phylogenetic trees. Mol Phylogenet Evol 53(1):320-8.

Lopez P, Casane D, Philippe H. 2002. Heterotachy, an important process of protein evolution. Mol Biol Evol 19(1):1-7.

May RM. 1990. Taxonomy as destiny. Nature 347(6289):129-130.

Meusemann K, von Reumont BM, Simon S, et al. 2010. A phylogenomic approach to resolve the arthropod tree of life. Mol Biol Evol 27(11):2451-64.

Minh BQ, Klaere S, von Haeseler A. 2006. Phylogenetic diversity within seconds. Systematic Biology 55(5):769-73.

Minh BQ, Klaere S, von Haeseler A. 2009. Taxon selection under split diversity. Systematic Biology 58(6):586-594.

Minh BQ, Klaere S, von Haeseler A. 2010. SDA*: A simple and unifying solution to recent bioinformatic challenges for conservation genetics. In: Pham SB, Hoang TH, McKay B, Hirota K, editors. The second international conference on knowledge and systems engineering. Hanoi, Vietnam: IEEE Computer Society. p. 33-37.

Mittermeier RA, Gil PR, Hoffman M, et al. 2005. Hotspots Revisited: Earth's Biologically Richest and Most Endangered Terrestrial Ecoregions. Mexico: Conservation International.

Monastersky R. 2014. Biodiversity: Life--a status report. Nature 516(7530):158-61.

Morrison DA. 2014. Is the tree of life the best metaphor, model, or heuristic for phylogenetics? Syst Biol 63(4):628-38.

Moulton V, Semple C, Steel M. 2007. Optimizing phylogenetic diversity under constraints. Journal of Theoretical Biology 246(1):186-194.

Myers N, Mittermeier RA, Mittermeier CG, et al. 2000. Biodiversity hotspots for conservation priorities. Nature 403(6772):853-8.

Nakhleh L, Don R, Warnow T. 2005. Perfect Phylogenetic Networks: A New Methodology for Reconstructing the Evolutionary History of Natural Languages. Language 81(2):382-420.

Nei M. 1987. Molecular Evolutionary Genetics. New York, USA: Columbia University Press.

Nguyen LT, Schmidt HA, von Haeseler A, et al. 2015. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. Mol Biol Evol 32(1):268-74.

Nyakatura K, Bininda-Emonds ORP. 2012. Updating the evolutionary history of Carnivora (Mammalia): a new species-level supertree complete with divergence time estimates. Bmc Biology 10.

Önal H, Briers RA. 2003. Selection of a minimum-boundary reserve network using integer programming. Proceedings of the Royal Society B-Biological Sciences 270(1523):1487-1491.

Opitz S. 1996. Trophic interactions in Caribbean coral reefs. Makati City, Philippines: ICLARM Technical Reports.

Pellens R, Grandcolas P. 2016. Biodiversity Conservation and Phylogenetic Systematics: Preserving our evolutionary heritage in an extinction crisis. Topics in Biodiversity and Conservation. 1 ed.: Springer International Publishing. p. 363.

Peters RS, Meyer B, Krogmann L, et al. 2011. The taming of an impossible child: a standardized all-in approach to the phylogeny of Hymenoptera using public database sequences. Bmc Biology 9.

Philippe H, Brinkmann H, Lavrov DV, et al. 2011. Resolving Difficult Phylogenetic Questions: Why More Sequences Are Not Enough. Plos Biology 9(3).

Philippe H, Zhou Y, Brinkmann H, et al. 2005. Heterotachy and long-branch attraction in phylogenetics. BMC Evol Biol 5:50.

Possingham HP, Ball IR, Andelman S. 2000. Mathematical methods for identifying representative reserve networks. In: Ferson S, Burgman M, editors. Quantitative Methods for Conservation Biology. New York: Springer. p. 291-305.

Purvis A, Gittleman JL, Brooks T. 2005. Phylogeny and Conservation. Cambridge: Cambridge University Press.

Pyron RA, Burbrink FT, Colli GR, et al. 2011. The phylogeny of advanced snakes (Colubroidea), with discovery of a new subfamily and comparison of support methods for likelihood trees. Molecular Phylogenetics and Evolution 58(2):329-342.

Pyron RA, Wiens JJ. 2011. A large-scale phylogeny of Amphibia including over 2800 species, and a revised classification of extant frogs, salamanders, and caecilians. Molecular Phylogenetics and Evolution 61(2):543-583.

Rannala B, Yang ZH. 2008. Phylogenetic inference using whole Genomes. Annual Review of Genomics and Human Genetics 9:217-231.

Robinson DF, Foulds LR. 1981. Comparison of Phylogenetic Trees. Mathematical Biosciences 53(1-2):131-147.

Roch S, Steel M. 2015. Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent. Theoretical Population Biology 100:56-62.

Rodrigues ASL, Gaston KJ. 2002. Optimisation in reserve selection procedures - why not? Biological Conservation 107(1):123-129.

Rogers JS. 1997. On the consistency of maximum likelihood estimation of phylogenetic trees from nucleotide sequences. Syst Biol 46(2):354-7.

Rokas A, Williams BL, King N, et al. 2003. Genome-scale approaches to resolving incongruence in molecular phylogenies. Nature 425(6960):798-804.

Saitou N, Nei M. 1986. The number of nucleotides required to determine the branching order of three species, with special reference to the human-chimpanzee-gorilla divergence. J Mol Evol 24(1-2):189-204.

Sanderson MJ, McMahon MM, Stamatakis A, et al. 2015. Impacts of Terraces on Phylogenetic Inference. Syst Biol.

Sanderson MJ, McMahon MM, Steel M. 2011. Terraces in phylogenetic tree space. Science 333(6041):448-50.

Sanderson MJ, Purvis A, Henze C. 1998. Phylogenetic supertrees: Assembling the trees of life. Trends Ecol Evol 13(3):105-9.

Schwarz G. 1978. Estimating Dimension of a Model. Annals of Statistics 6(2):461-464.

Semple C, Steel MA. 2003. Phylogenetics. Oxford ; New York: Oxford University Press.

Sneath PHA. 1975. Cladistic Representation of Reticulate Evolution. Systematic Zoology 24(3):360-368.

Soltis DE, Mort ME, Latvis M, et al. 2013. Phylogenetic Relationships and Character Evolution Analysis of Saxifragales Using a Supermatrix Approach. American Journal of Botany 100(5):916-929.

Springer MS, Meredith RW, Gatesy J, et al. 2012. Macroevolutionary Dynamics and Historical Biogeography of Primate Diversification Inferred from a Species Supermatrix. Plos One 7(11).

Srivastava DS, Cadotte MW, MacDonald AAM, et al. 2012. Phylogenetic diversity and the functioning of ecosystems. Ecology Letters 15(7):637-648.

Stamatakis A. 2014. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics 30(9):1312-3.

Stamatakis A, Alachiotis N. 2010. Time and memory efficient likelihood-based tree searches on phylogenomic alignments with missing data. Bioinformatics 26(12):i132-i139.

Stamatakis A, Ludwig T, Meier H. 2005. RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees. Bioinformatics 21(4):456-63.

Tavare S. 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. Lecture Notes on Mathematical Modelling in the Life Sciences 17:57-86.

Tehrani JJ. 2013. The phylogeny of Little Red Riding Hood. PLoS One 8(11):e78871.

Tollefson J, Gilbert N. 2012. Earth summit: Rio report card. Nature 486(7401):20-3.

Underhill LG. 1994. Optimal and suboptimal reserve selection algorithms. Biological Conservation 70(1):85-87.

van der Linde K, Houle D, Spicer GS, et al. 2010. A supermatrix-based molecular phylogeny of the family Drosophilidae. Genetics Research 92(1):25-38.

Vanewright RI, Humphries CJ, Williams PH. 1991. What to protect - Systematics and the agony of choice. Biological Conservation 55(3):235-254.

von Haeseler A. 2012. Do we still need supertrees? BMC biology 10(1):13.

Whelan S, Goldman N. 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. Mol Biol Evol 18(5):691-9.

Winter M, Devictor V, Schweiger O. 2013. Phylogenetic diversity and nature conservation: where are we? Trends in Ecology & Evolution 28(4):199-204.

Yang Z. 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. Journal of Molecular Evolution 39(3):306-14.

Yang Z. 1996. Maximum-Likelihood Models for Combined Analyses of Multiple Sequence Data. J Mol Evol 42(5):587-96.

Yang Z. 2006. Computational Molecular Evolution. New York: Oxford University Press Inc.

Yang ZH, Rannala B. 1997. Bayesian phylogenetic inference using DNA sequences: A Markov Chain Monte Carlo method. Molecular Biology and Evolution 14(7):717-724.

Zwickl DJ. 2006. Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion. The University of Texas at Austin.

APPENDIX A

Supplementary figures to Chapter 2

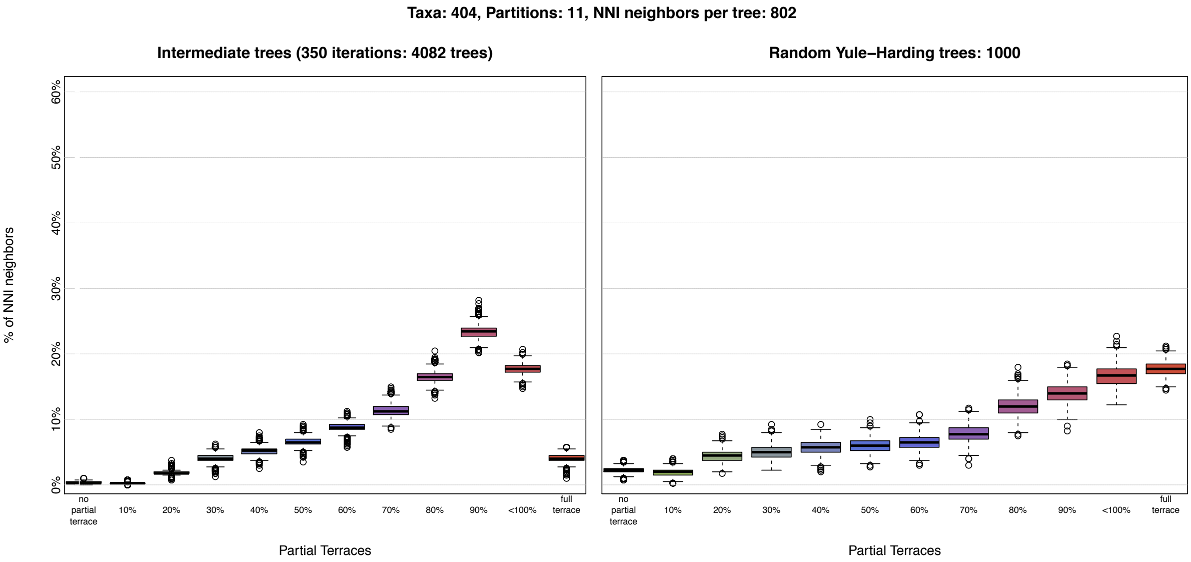


Figure A.1. Analysis of NNI neighbourhood of intermediate and random trees for DNA4

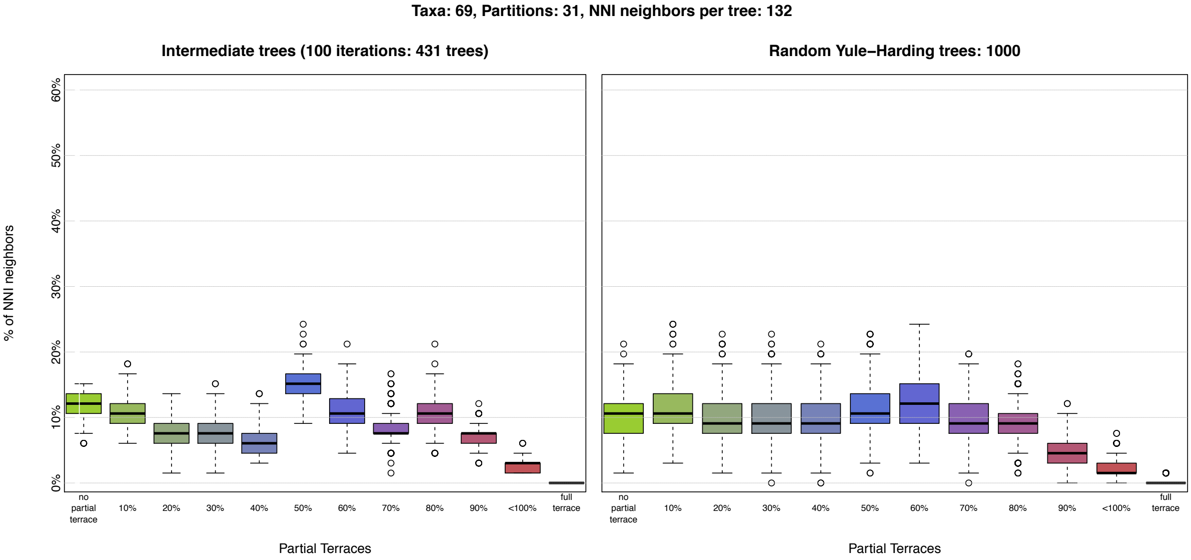


Figure A.2. Analysis of NNI neighbourhood of intermediate and random trees for AA1

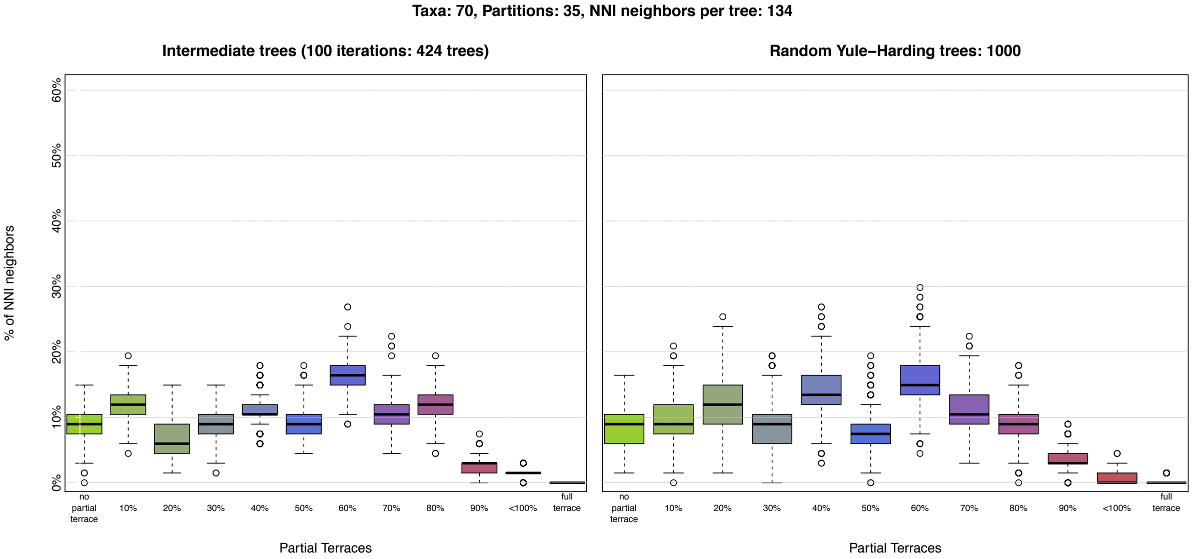


Figure A.3. Analysis of NNI neighbourhood of intermediate and random trees for AA2

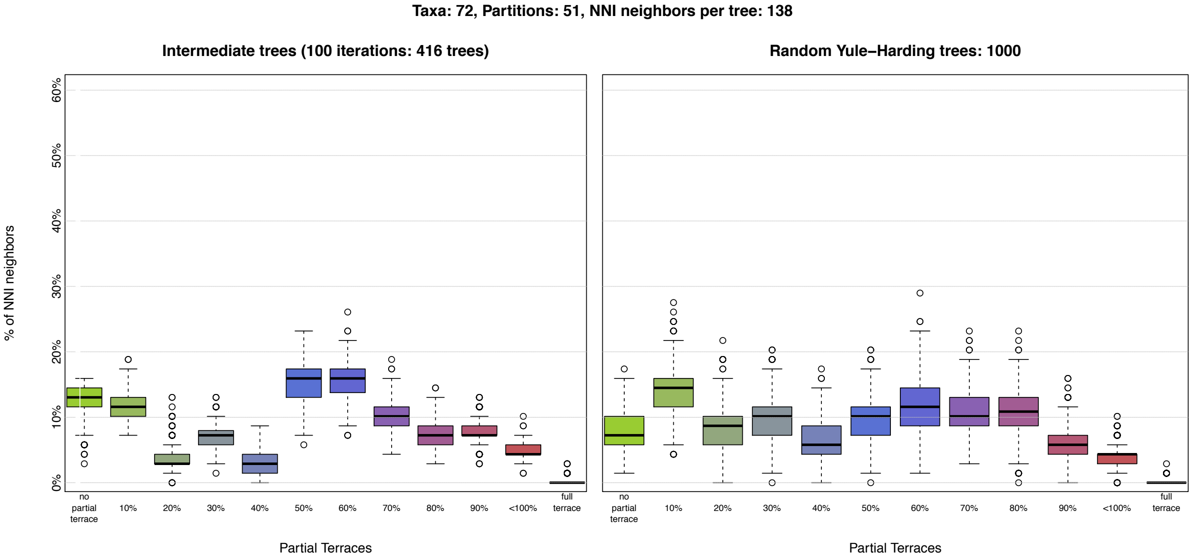


Figure A.4. Analysis of NNI neighbourhood of intermediate and random trees for AA3

APPENDIX B

Supplementary text and figures to Chapter 3

B.1 Details of using PTA with SPR and TBR

- *Complexity of updating F*

For an SPR (and similar for a TBR) the time needed to recompute F depends on the length of the move, i.e. the number of edges between the cut edge and its reinsertion. Nevertheless, since phylogenetic software mainly apply short moves (e.g., RAxML, see (Stamatakis, et al. 2005)), recomputation time of F is close to $O(k)$ also for SPR and TBR.

- *Detection of partial terraces with PTA in SPR- and TBR-based tree searches*

Here we present reformulations of Propositions 2 and 3 (Chernomor, et al. 2015) in terms of PTA maps. We follow the notations introduced in the original paper. Let T_{SPR} and T_{TBR} denote the species trees obtained from T by one SPR and one TBR, respectively. Any SPR and TBR can be represented in the forms shown in Figure B.1.

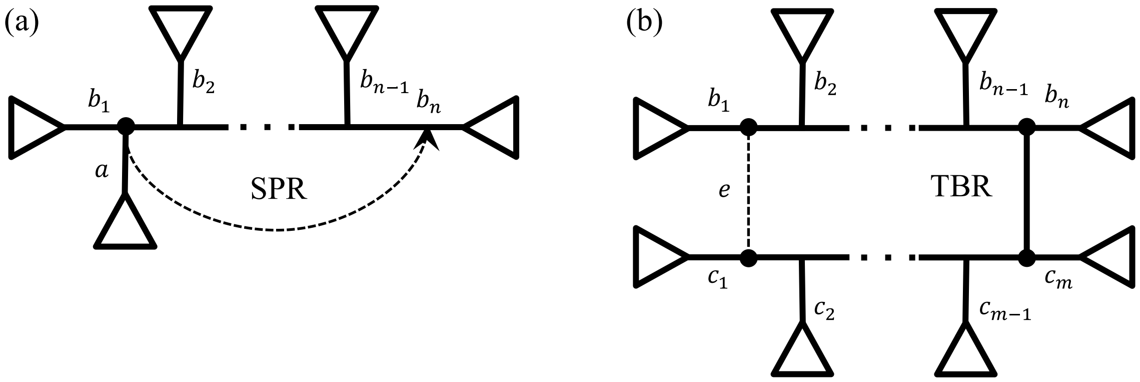


Figure B.1. General representations of (a) SPR and (b) TBR, where the triangles denote the subtrees below the corresponding edges.

Condition for SPR. If an SPR is applied to a species tree T by pruning the subtree below edge a and regrafting it onto b_n (Figure B.1, panel a), then for a partition with a taxon set Y_i the following is true

(iii) the topologies of $T|Y_i$ and $T_{SPR}|Y_i$ are different, if $f_i(a)$ and at least three from $f_i(b_1), f_i(b_2), \dots, f_i(b_n)$ are different from ε;

(iv) this SPR corresponds to an SPR on $T|Y_i$ obtained by pruning the subtree below edge $f_i(a)$ and regrafting it onto edge $f_i(b_k)$, where $k = \max_{1 \le x \le n}\{x \mid f_i(b_x) \neq \varepsilon\}$.

Condition for TBR. If a TBR is applied to a species tree T by cutting edge e and reconnecting b_n and c_m with a new edge (Figure B.1, panel b), then for a partition with a taxon set Y_i the following is true

(iii) the topologies of $T|Y_i$ and $T_{TBR}|Y_i$ are different, if at least one of the following conditions is satisfied:

- at least one from $f_i(b_1), f_i(b_2), \dots, f_i(b_n)$ and at least another three from $f_i(c_1), f_i(c_2), \dots, f_i(c_m)$ are different from ε;
- at least one from $f_i(c_1), f_i(c_2), \dots, f_i(c_m)$ and at least another three from $f_i(b_1), f_i(b_2), \dots, f_i(b_n)$ are different from ε;

(iv) this TBR corresponds to a TBR on $T|Y_i$ obtained by cutting the edge $f_i(e)$ and reconnecting edges $f_i(b_k)$ and $f_i(c_h)$, where $k = \max_{1 \le x \le n}\{x \mid f_i(b_x) \ne \varepsilon\}$ and $h = \max_{1 \le y \le n}\{y \mid f_i(c_y) \ne \varepsilon\}$.

B.2 Applying NNIs for EL partition models

Under EL models together with the topological changes we also have to consider the changes of edge lengths on partition trees after the NNI was applied to T.

If the central edge e has the same corresponding subsplit before and after NNI, the edge lengths on partition tree are not changed. Otherwise, from Eq. (8) it follows that the corresponding edge $f_i(e)$ before NNI, if not equal to ε, should have its length decreased by $r_i\lambda(e)$, and the corresponding edge $f_i(e)$ after NNI, if not equal to ε, should have its length increased by the same amount. Here, $\lambda(e)$ is the length of e on species tree T.

To save computing time one re-optimizes only edges in the vicinity of topological changes (Guindon, et al. 2010; Nguyen, et al. 2015; Stamatakis, et al. 2005), in the following we assume that for the NNI one has to optimize five edges: e and its incident edges e_1, e_2, e_3, e_4.

There are four different cases depending on the number of edges e, e_1, e_2, e_3, e_4 on T that map to ε before NNI (Figure B.2).

In case 1, when one out of $f_i(e_1), f_i(e_2), f_i(e_3), f_i(e_4)$ is equal to ε, the two NNIs change the corresponding edge lengths and the resulting partition trees, $T_{NNI1}|Y_i$ and $T_{NNI2}|Y_i$, have different edge lengths. In case 2, when the map of two non-incident edges is equal to ε, only one NNI changes the edge lengths. For example, in Figure B.2, where $f_i(e_1) = f_i(e_3) = \varepsilon$, NNI1 does not change the edge lengths, because the $f_i(.)$ of e, e_1, e_2, e_3, e_4 is not changed, while after NNI2 $f_i(e)$ is equal to ε. In case 3, when the map of two incident edges and as a result also $f_i(e)$ are equal to ε, both NNIs change the edge lengths, but the induced partition trees resulting from the NNIs have the same edge lengths.

In case 4, when at least any three out of $f_i(e_1), f_i(e_2), f_i(e_3), f_i(e_4)$ are equal to ε, the map $f_i(.)$ of all e, e_1, e_2, e_3, e_4 is not affected by the NNI and is equal to ε before and after an NNI is applied to the species tree T. Therefore, together with the topology also the edge lengths remain unchanged and as a result no recomputation is necessary for such partition at all.

Though for cases 1-3 the edge lengths of partition trees are affected by the topological rearrangement and also have to undergo optimization, these computations are still less demanding as if the topology of partition tree would be changed. Therefore, taking into account this information during the tree search still leads to speedups as shown in the results section for EL models.

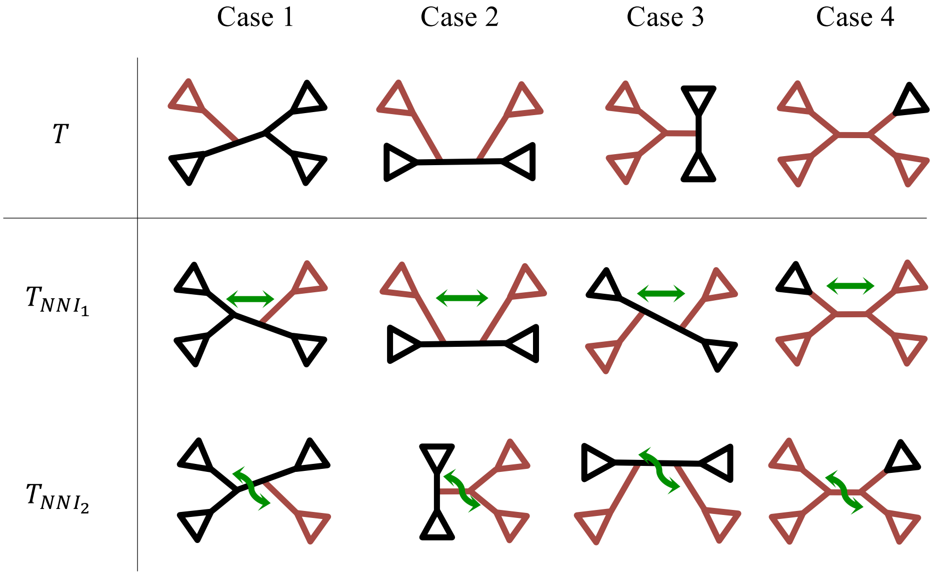


Figure B.2. Cases that do not change the topology of partition tree under EL models. Each tree is a species tree: before NNI (T, first row) and after NNI (T_{NNI}, second and third rows). The edges and the species sets leading to them are the same as in Fig. 3.1 (for example, on T the upper left edge is e_1 with the species set A and so on). Here, $f_i(.)$ of red colored edges is equal to ε and red triangles correspond to taxa sets, which are absent on the considered partition tree: $T|Y_i, T_{NNI_1}|Y_i$ or $T_{NNI_2}|Y_i$. The black colored parts correspond to the topologies of these induced partition trees. The arrows show edges that were swapped during NNI around the central edge e on T.

APPENDIX C

Supplementary figures to Chapter 4

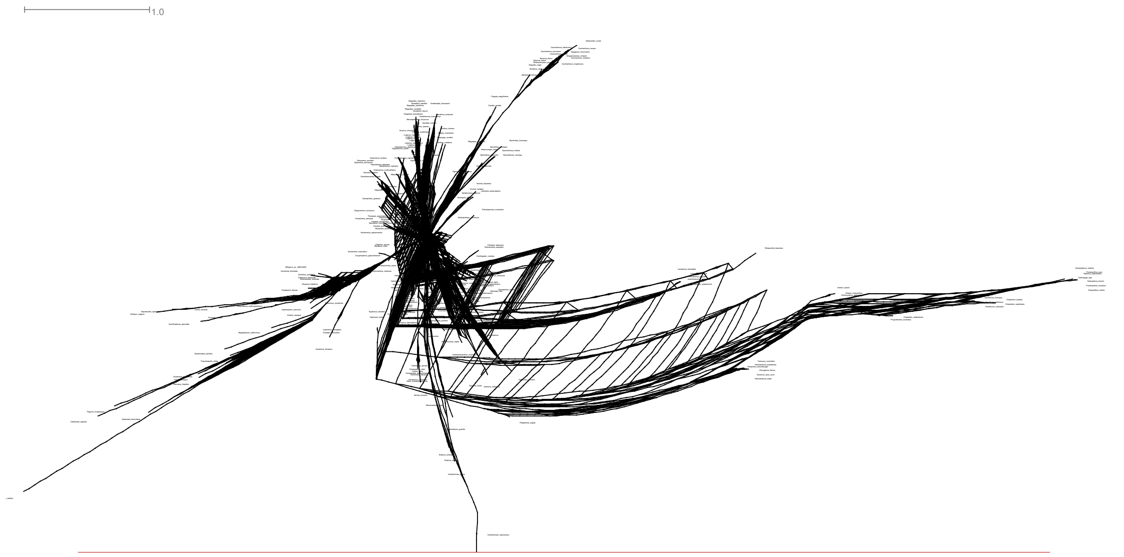


Figure C.1. The split network of six gene trees

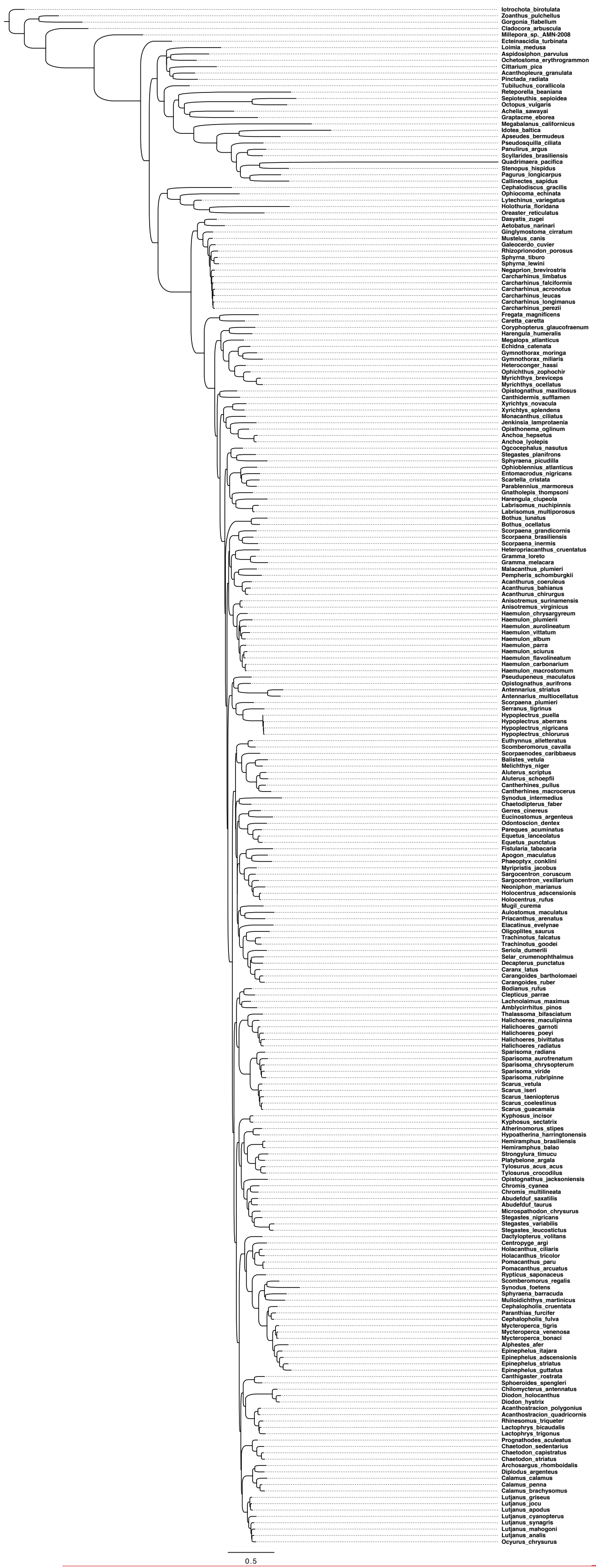


Figure C.2. The tree reconstructed from the supermatrix of six genes