INSTITUT FÜR INFORMATIK
Algorithmische Bioinformatik

Universitätsstr. 1 D–40225 Düsseldorf

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Lagrangian Relaxation for the Generalized Robinson-Foulds Metric

Laura Christine Kühle

Bachelorarbeit

Beginn der Arbeit:	26. November 2018
Abgabe der Arbeit:	26. Februar 2019
Gutachter:	Univ.-Prof. Dr. Gunnar Klau
	Univ.-Prof. Dr. Martin Lercher

Erklärung

Hiermit versichere ich, dass ich diese Bachelorarbeit selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Düsseldorf, den 26. Februar 2019

Laura Christine Kühle

Abstract

The phylogenetic tree structure is the prevalent way of portraying ancestry in biology. It displays hierarchically how different organisms (leaves of the tree) presumably evolved from a shared ancestor (root of the tree). Since even the latest research has not been able to predict ancestry with certainty yet, the development of new methods to resolve the differences between predictions made using molecular data and morphological data requires a way to compare such phylogenetic trees. The most commonly used distance indicator is still the Robinson-Foulds (RF) metric which gives the number of clades (clade = list of organisms with a shared ancestor) found in only one of the trees. Despite its simplicity, it gives a good measurement of the similarity, yet unfortunately taking only identical clades into account and completely disregarding merely similar ones. Therefore, the metric can yield inaccurate results. A good example is the comparison of a tree with ten leaves and the same tree but with one outer leaf moved to the other side. The RF metric categorizes those two trees with a normalized distance of 1 meaning the trees show no similarity at all, despite its many overlaps in the structure. A metric that still maintains the advantages of the RF metric but also gives less conservative distance values is the Generalized Robinson-Foulds (GRF) metric. It utilizes the similarity between clades by matching them to each other but still preserves the structure of the tree by prohibiting matches which disregard it. The only disadvantage is that such a problem is NP-hard, but an Integer Linear Program (ILP) approach can avert this. I implemented a heuristic which solves the ILP using Lagrangian Relaxation to approximate the GRF distance and analyzed its usefulness with regard to the example trees from above and in context of a data set comparing the differences between phylogenetic trees procured from molecular and morphological data from fossils provided by Beck et al. While the algorithm is relatively time-consuming, the result values were consistent with the trends which become apparent using different metrics. Moreover, the GRF metric yields continuously smaller distances than the RF metric and seems to be consistent with other finer distance indicators.

Contents

1 Introduction

In biology, phylogenetic trees are commonly used to show possible relations between all kinds of living organisms. A rooted phylogenetic tree displays evolution beginning with a shared ancestor from which all creatures in the tree originate (the root of the tree) via the different forms they had passed in the development to their current evolutionary stage (leaves of the tree). The organisms are represented by so-called taxa. The definition of a taxon is: "A taxonomic unit, whether named or not: i.e. a population, or group of populations of organisms which are usually inferred to be phylogenetically related and which have characters in common which differentiate (q.v.) the unit (e.g. a geographic population, a genus, a family, an order) from other such units. A taxon encompasses all included taxa of lower rank (q.v.) and individual organisms. []" (Zoological Nomenclature. et al., 1999). It is clear that the closer two taxa are in the tree, the closer they are related presumably, i.e. they share more physical or genetic similarities. Even though that all living creatures on earth probably share the same ancestor, it is still unclear how exactly every species evolved. Therefore, biologists try to reconstruct ancestry based on either molecular or morphological data. Unfortunately, there are still significant differences between the trees derived from these two approaches. Hence, comparing phylogenetic trees can be of great importance for the development of new methods or evaluating results pertaining to ancestry or relations.

Motivation for this Bachelor thesis stems from an inquiry of Beck et al. who recently proposed a new approach to resolve this conflict between morphological and molecular estimates of mammal phylogeny (Beck et al., 2018): They inferred the fossil ancestors of 46 different mammals under either maximum parsimony (MP) or maximum likelihood (ML) for the skeletal, craniodental, dental, maximum preservation, typical preservation or all character data. They are thus choosing a morphological approach.

For their evaluation, Beck et al. wanted to use several metrics to compare such phylogenetic trees checking for similarity between their generated trees and a tree created by Meredith et al. based on molecular data. In their work, they used different methods for comparison: the normalized Robinson-Foulds metric, the Subtree Prune and Regraft (SPR) distance which gives the minimum of subtrees rearranged to transform the structure of one tree to the another's (Hein, 1990), and the distortion coefficient. They were interested in seeing how the Generalized Robinson-Foulds distance, applied using Jaccard-Robinson-Foulds metric, as introduced by Böcker et al. would fair in comparison to the other approaches. Hence, in the following I will introduce an implementation of a tool to compare phylogenetic trees using said metric and discuss its application to the data received from Beck et al.

2 Preliminaries

In this section, definitions which are critical to understanding the content of this work are introduced.

2.1 Tree structure

Definition 2.1. *A **vertex** is an object in a graph.*

Definition 2.2. *Let v, w be vertices. A **directed edge** is a pair (v, w) which represents a connection from v to w in a graph.*

Definition 2.3. *Let E be a set of directed edges and V be a set of vertices. A **graph** G is defined as a pair (V, E).*

Definition 2.4. *Let $G = (V, E)$ be a graph and $v, w \in V$. v is a **parent** of w if $\exists (v, w) \in E$.*

Definition 2.5. *Let $G = (V, E)$ be a graph and $v, w \in V$. w is a **child** of v if $\exists (v, w) \in E$.*

Definition 2.6. *Let $G = (V, E)$ be a graph. G is **acyclic** if $\nexists v \in V : v$ is a (transitive) child of v.*

Definition 2.7. *Let $G = (V, E)$ be a graph. A **root** is a vertex $w \in V$ with no parents, i.e. $\nexists v \in V : (v, w) \in E$.*

Definition 2.8. *Let $G = (V, E)$ be a graph. A **leaf** is a vertex $v \in V$ with no children, i.e. $\nexists w \in V : (v, w) \in E$.*

Definition 2.9. *A **tree** T is an acyclic graph $G = (V, E)$. A tree is called **rooted** if it has one root.*

Definition 2.10. *Let $T = (V, E)$ be a tree. A tree $T^{'} = (V^{'}, E^{'})$ is called a **subtree** of G if $V^{'} \in V$ and $E^{'} \in E$.*

From now on, let assume that all (phylogenetic) trees are rooted and store information only in the leaves (comp. Figure 1) if not stated otherwise.

2.2 Clade

Definition 2.11. *Let $T = (V, E)$ be a tree and $v \in V$. A **clade** C (associated to a vertex v) is a set of entries procured from all leaves that are children (direct or transitive) of v in T. An alternate denotation is $C(v)$.*

Definition 2.12. *Let $T = (V, E)$ be a tree and $v \in V$. A clade associated to v is **non-trivial** if v is not the root or a leaf, i.e. $\exists w_1, w_2 \in V : (w_1, v), (v, w_2) \in E$. Each set of clades is called **non-trivial** if it contains only non-trivial clades.*

Definition 2.13. *Let $T = (V, E)$ be a tree. A set of non-trivial clades for T is **complete** if and only if contains a clade for every vertex except the root and every leaf. The complete non-trivial set of clades for T is denoted by $\mathcal{C}(T)$.*

Definition 2.11 varies from the usual definition of clades used for example in the *Biopython* package (*Biopython Class Clade* 2019): For a tree $T = (V, E)$ and $v \in V$, a clade $C(v)$ is a subtree with the root v. Note that I will not be using this definition but it should be kept in mind when looking at the code. Clades according to the definition used in *Biopython* will be referred to as *tree clades*.

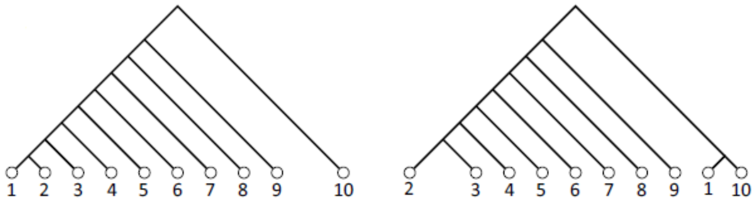

Figure 1: Two phylogenetic trees with ten leaves each (cf. Böcker et al. (2013)) over the set of entries S = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. The only difference is the position of the leaf with the label "1".

3 Generalized Robinson-Foulds Metric

Note, that this section is based on Böcker et al. (2013).

3.1 Robinson-Foulds Metric

The Robinson-Foulds metric was introduced in 1981 (Robinson et al., 1981). While being simple, it gives a general idea of the similarity between two phylogenetic trees. The approach is as follows: Given two trees $T_1 = (V_1, E_1), T_2 = (V_2, E_2)$, the complete non-trivial set of clades is acquired for each of them and the distance is calculated using the following formula:

$$dist_{RF} = \sum_{C_1 \in \mathcal{C}(T_1)} \delta_1(C_1) + \sum_{C_2 \in \mathcal{C}(T_2)} \delta_2(C_2) \tag{1}$$

with

$$\delta_i(C) = \begin{cases} 1 & , C \notin \mathcal{C}(T_j) \\ 0 & , C \in \mathcal{C}(T_j) \end{cases} \tag{2}$$

for $i \neq j, i = 1, 2$.
Obviously, a higher score indicates a greater distance between the trees.

The problem with this metric is that it only searches for identical parts of the trees and not for merely similar ones. To illustrate this, Figure 1 shows two trees in which the only difference is a movement of the leaf with the label "1" from the left side to the right of the tree. One would assume that the trees are fairly similar, however, comparing these trees with the Robinson-Foulds metric yields a score of 16, the maximum score for trees of this size. Thence, the metric is, for at least some trees, not incredibly accurate and not necessarily the best choice in comparing trees such as Beck et al.'s.

3.2 Generalized Robinson-Foulds Metric

An approach that is less conservative than the original Robinson-Foulds metric, yet still maintains the arboreal structure, was introduced by Böcker et al. (2013). The so-called Generalized Robinson-Foulds (GRF) metric operates differently in the sense that it matches not only identical, but also similar clades according to the following rules:

Define a cost function

$$\delta : (\mathcal{P}(X) \cup \{-\}) \times (\mathcal{P}(X) \cup \{-\}) \to \mathbb{R}_{\geq 0} \cup \{\infty\}\,, \tag{3}$$

with $'-'$ being the gap symbol and $\mathcal{P}(X)$ being the power set of a set of taxa X over which the trees are defined. Thus, $\delta(Y_1, Y_2)$ is a way to measure the similarity of two clades $Y_1, Y_2 \in X$ and both $\delta(Y_1, -)$, $\delta(-, Y_2)$ give the cost of not matching a clade from one of the trees.

For my purposes, I, like Böcker et al., decided to use the Jaccard-Robinson-Foulds (JRF) metric of order k,

$$\delta(Y_1, Y_2) = 2 - 2 \cdot \left(\frac{|Y_1 \cap Y_2|}{|Y_1 \cup Y_2|} \right)^k \tag{4}$$

$$\delta(Y_1, -) = 1 \tag{5}$$

$$\delta(-, Y_2) = 1 \tag{6}$$

$$\delta(\emptyset, \emptyset) = 0\,. \tag{7}$$

To solve this minimum cost problem, let a complete bipartite graph G with vertex set $\mathcal{C}_1 \cup \mathcal{C}_2$ be defined and the weight of an edge $(C_1, C_2), C_1 \in \mathcal{C}_1, C_2 \in \mathcal{C}_2$ be given by

$$w(C_1, C_2) := \delta(C_1, -) + \delta(-, C_2) - \delta(C_1, C_2)\,. \tag{8}$$

Reduces the minimum cost problem to a maximum matching over the graph G.

However, another complication arises as a result: the arboreal structure of the trees could be violated for matches in conflict. To prohibit this, let an *arboreal matching* be defined as a matching for which no matched clades are incompatible. Two matches are *incompatible* if a *conflict* occurs, i.e. when none of these cases holds:

$$Y_1 \subseteq Z_1 \wedge Y_2 \subseteq Z_2 \tag{9}$$

$$Y_1 \supseteq Z_1 \wedge Y_2 \supseteq Z_2 \tag{10}$$

$$Y_1 \cap Z_1 = \emptyset \wedge Y_2 \cap Z_2 = \emptyset\,, \tag{11}$$

for any matches $(Y_1, Y_2), (Z_1, Z_2) \in \mathcal{M}$.

From now on, each matching that is not arboreal is referred to as a *normal* matching.

4 ILP and Lagrangian Relaxation

Unfortunately, this maximum arboreal matching for a complete bipartite graph G is NP-complete as shown in Böcker et al. (2013). Thus, I try to gain a good approximation with the help of an Integer Linear Program and Lagrangian Relaxation. These will be introduced in this section which is based on Bauer (2008) and Andreotti (2014).

4.1 Integer Linear Program

Given two rooted phylogenetic trees $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ with complete non-trivial clade sets $\mathcal{C}(T_1)$ and $\mathcal{C}(T2)$, I now want to introduce an Integer Linear Program (ILP) formulation on which the tool can be based.

First, let clades of T_1 be numbered with C_i, $i = 1, \cdots, |V_1|$ and clades in T_2 with $\tilde{C}_j$, $j = 1, \cdots, |V_2|$. Then let the set of all possible matches $\mathcal{M} = \{(C_1, C_2) : C_1 \in \mathcal{C}(T_1), C_2 \in \mathcal{C}(T_2)\}$ be define and an indicator variable introduced

$$x_{i,j} = \begin{cases} 1 & ,(C_i, \tilde{C}_j) \in \mathcal{M} \\ 0 & ,else\,. \end{cases} \tag{12}$$

Additionally, $\mathcal{I}$ is a set which contains all pairs of incompatible matches $\{(i,j)(k,l)\}, (i,j), (k,l) \in \mathcal{M}$.

Now the weight function (8) can be used to define the ILP:

$$\max \sum_{i=1}^{|V_1|} \sum_{j=1}^{|V_2|} w(C_i, \tilde{C}_j) x_{i,j} \tag{13}$$

$$s.t. \sum_{j=1}^{|V_2|} x_{i,j} \leq 1 \qquad \forall i = 1, \cdots, |V_1|, \tag{14}$$

$$\sum_{i=1}^{|V_1|} x_{i,j} \leq 1 \qquad \forall j = 1, \cdots, |V_2|, \tag{15}$$

$$x_{i,j} + x_{k,l} \leq 1 \qquad \forall \{(i,j), (k.l)\} \in \mathcal{I}, \tag{16}$$

$$x_{i,j} \in \{0, 1\}\,. \tag{17}$$

As it is easily provable that a maximum matching for complete bipartite graphs is fairly quickly solvable, the part that makes this problem NP-complete has to be the condition (16). To eliminate this, I now try to get an approximation of the maximum using Lagrangian Relaxation.

However, first a slight modification of the ILP is necessary first: Replace the two sums in the maximum term with a vector

$$\omega = [w(C_1, \tilde{C}_1), w(C_1, \tilde{C}_2), \cdots, w(C_1, \tilde{C}_{|V_2|}), w(C_2, \tilde{C}_1), \cdots, w(C_{|V_1|}, \tilde{C}_{|V_2|}),]^T \tag{18}$$

times a vector

$$x = [x_{1,1}, x_{1,2}, \cdots, x_{|V_1|,|V_2|}]^T \tag{19}$$

and write (16) as inequality between a vector where every entry equals 1 (here denoted as $\mathbb{1}$) and the product of a new matrix D with dimension $|\mathcal{I}| \times (|V_1| \cdot |V_2|)$, which contains the conflict pairs and vector x.

I get the following modified ILP:

$$\max \omega^T x \tag{20}$$

$$s.t. \sum_{j=1}^{|V_2|} x_{i,j} \leq 1 \qquad \forall i = 1, \cdots, |V_1|, \tag{21}$$

$$\sum_{i=1}^{|V_1|} x_{i,j} \leq 1 \qquad \forall j = 1, \cdots, |V_2|, \tag{22}$$

$$Dx \leq \mathbb{1}, \tag{23}$$

$$x_{i,j} \in \{0, 1\}. \tag{24}$$

4.2 Lagrangian Relaxation

Lagrangian Relaxation is a technique to simplify ILP problems by dropping severe constraints by putting them in the maximum term and introducing a new factor λ, the *Lagrangian multiplier*. In case of the ILP for the Generalized Robinson Foulds metric, the severe constraint is (23) and I get:

$$\max_x \omega^T x - \lambda^T(\mathbb{1} - Dx) \tag{25}$$

$$s.t. \sum_{j=1}^{|V_2|} x_{i,j} \leq 1 \qquad \forall i = 1, \cdots, |V_1|, \tag{26}$$

$$\sum_{i=1}^{|V_1|} x_{i,j} \leq 1 \qquad \forall j = 1, \cdots, |V_2|, \tag{27}$$

$$x_{i,j} \in \{0, 1\}. \tag{28}$$

This is now easily solvable for specific values of λ.

Wanting to find an approximation which is as close as possible to the actual solution of the original ILP, the problem has to be minimized over λ. This results in a new ILP which will be called $LR(\lambda)$:

$$\min_{\lambda \geq 0} \max_x \omega^T x - \lambda^T(\mathbb{1} - Dx) \tag{29}$$

$$s.t. \sum_{j=1}^{|V_2|} x_{i,j} \leq 1 \qquad \forall i = 1, \cdots, |V_1|, \tag{30}$$

$$\sum_{i=1}^{|V_1|} x_{i,j} \leq 1 \qquad \forall j = 1, \cdots, |V_2|, \tag{31}$$

$$x_{i,j} \in \{0, 1\}. \tag{32}$$

That is the one for which will be solved with the implemented tool.

Algorithm 1: Main steps of the Subgradient Method

```
Initialise $\lambda^0$, set $t = 0$;
while stopping criterion not met do
    Choose a subgradient $s_t \in \partial LR(\lambda^t)$ based on $\lambda^t$;
    if $s_t == 0$ then
        stop;
    end
    $\lambda^{t+1} = \lambda^t + \mu_t s_t$;
end
```

Figure 2: Algorithm for the general Subgradient Method as described by Bauer (2008).

4.3 Subgradient Method

To solve $LR(\lambda)$, it was decided to use the Subgradient Method.
To do this, note that $LR(\lambda)$ is piecewise linear and convex meaning that the following lemma is applicable.

Lemma 4.1. *Let $\theta : \mathbb{R}^n \to \mathbb{R}$ be a convex function. A vector x^* minimizes θ over $\mathbb{R}^n$ if and only if $0 \in \partial\theta(x^*)$.*

Thus, I want to find a subgradient $s \in \partial LR(\lambda)$ which equals 0 for one λ^* and the corresponding x^*. The general Subgradient Method is described in Algorithm 1 (Figure 2).

According to Andreotti (2014), it is possible to show that $(\mathbb{1} - Dx^*)$ defines a subgradient of $LR(\lambda)$. I will use this to adjust λ in each step with

$$\lambda^{n+1} = \max\{\lambda^n + \mu_n(\mathbb{1} - Dx^n), 0\} \,. \tag{33}$$

μ is thereby the step size. Theoretically, each step size that satisfies

$$\sum_{n}^{\infty} \mu_n = \infty, \tag{34}$$

$$\lim_{n\to\infty} \mu_n = 0 \tag{35}$$

converges to an optimum solution but the convergence rates are usually quite poor. A widely used choice of step size is given by Held et al. (1971),

$$\mu_n = \frac{\gamma_n(LR(\lambda^n) - LR(\bar{x}))}{\|\mathbb{1} - Dx^n\|} \,. \tag{36}$$

$\bar{x}$ is thereby an estimate for the optimum value, and the factor γ_n is an arbitrary scalar which gets adjusted after a number of iterations without change of the bounds. Note that $LR(\lambda^n)$ is also dependable on x^n but since x is already dependable on λ it is not necessary to denote $LR(\lambda)$ as explicitly dependant on x. $LR(\bar{x})$ is the solution of $LR(\lambda)$ for λ equals zero in every argument and $\bar{x}$, and functions as a lower bound for our approximation. Additionally, it is necessary to have a variable for the upper bound, LR_{best} which gets adjusted when $LR(\lambda^n)$ is smaller than the current value.

5 Tool Implementation

The tool is available at *Gitlab Repository of GRF with Lagrangian Relaxation* (2019).

5.1 Requirements

First of all, a few conditions have to be fulfilled to make the tool work:

1. One has to have a functioning Python environment which includes the default packages, especially *numpy*, *scipy*, *argparse* and *math*. Recommended are the Anaconda or Miniconda packages which are freely available online.

2. The external package *Biopython* has to be installed in the Python environment. It is also freely available online.

3. The comparison trees have to be saved in one file in one of the common tree formats (Nexus and Newick are probably the most common, but any type that is supported by *Biopython* is sufficient). Alternatively, if one chooses to bypass the main function and access the method to calculate the Generalized Robinson-Foulds (GRF) metric directly, it is possible to do this by converting the chosen trees manually in the *Biopython* tree data type and then using the function *LagrangianRelaxation.calculateDistanceBetweenTwoTrees()*.
   The first approach is strongly recommended.

Further knowledge of either Python or *Biopython* is not necessary.

5.2 How to use the tool?

There are two possible ways to use the implementation of the method introduced in Section 4.3:
Either, one integrates the method in their own program by importing the files *LagrangianRelaxation.py*, *Output.py*, and *CladeConverter.py*, or one uses *Main.py*, a method that allows to easily compare two trees without further knowledge of the code.

Main.py can be used via the terminal as it is implemented using the *argparse* package which allows the use of input arguments as explained in the following:

1. file (file):
   is a string which gives the file name (with the directory) in which the (at least) two phylogenetic trees are saved. If the file contains more than two trees, the remaining are ignored for the calculation. This parameter is not optional.

2. file format (fileFormat):
   is a string giving the format the trees are saved in. Any format available in *Biopython* will be accepted. This parameter is not optional.

3. iteration (-iter, - -iterations):
   is an integer giving the maximum number of iterations the method will be trying to approximate the exact GRF. This parameter is optional, and the default value is 1000 iterations.

4. order of the JRF metric (-k, - -JRForder):
   is a positive integer which gives the order of the Jaccard-Robinson-Foulds metric. This parameter is optional, and the default value is 1.

5. stagnation limit (-stag, - -stagnationLimit):
   is an integer value for the limit of iteration after which the value of γ is adjusted according to the amplification factor. This parameter is optional, and the default value is the maximum of 5 and a tenth of the iteration number in total. The highest amount permitted is 20.

6. amplification factor (-amp, - -amplificationFactor):
   is a float value for adjusting γ. This parameter is optional, and the default value is 1.1.

7. error term (-err, - -errorTerm):
   is a float value which allows the accuracy to be adjusted. This parameter is optional, and the default value is 1E-15.

8. plot (-p, - -plot):
   is a flag which, if set, generates a .pdf and a .png file each containing a plot of the approximation progression.

9. matching (-m, - -matching):
   is a flag which, if set, generates two files, one containing the normal and the other the arboreal matching in the last iteration.

An incorrect access attempt on *Main.py* will yield the help menu as a result which also includes all information about input arguments listed above and is, as usual, accessible via -h or - -help.

5.3 Structure

The tool consists of four parts:

CladeConverter.py which gives methods to convert the tree clade data type from *Biopython* to a clade as defined in Section 2.1, *LagrangianRelaxation.py* which includes the actual calculation of the GRF with Lagrangian Relaxation, *Output.py* which contains methods responsible for any output, and *Main.py*, a handy main function to access the tool easily via the terminal.

CladeConverter.py goes through the tree recursively and creates a list of entries in the leaves which are (transitive) children (ergo a clade) for every single vertex. The methods used are pretty straight-forward and will not be further explained here.

Main.py includes the main function *Main.Main()* which takes a file, reads the trees using the parser from *Biopython*, calls the method *LagrangianRelaxation.calculateDistanceBetweenTwoTrees()*, and prints the returned output. Additionally, a simplified call via the terminal is implemented (see above).

Output.py generates files for the matching or the plot (if the flags are set accordingly). Moreover, it contains a method which returns the output in the correct format as it converts the results from the solved maximum matching problem, to the desired minimum distance problem.

LagrangianRelaxation.py is the most critical part of the tool as it contains the calculation of the GRF metric itself. Getting two trees (tree data type from *Biopython*) as input, it generates a list of clades for each of them and then starts the actual calculation by first creating a matrix containing the incompatibilities and a vector with the weighting for the JRF metric. γ is set to 2 to begin with. Then the lower and upper bound get approximated with the desired parameters. In each iteration, the normal matching is calculated by an external (i.e. not implemented in the tool) solver. Then, an arboreal matching is procured by checking for each match if it is incompatible with another match. If this is the case, the match with the lower score from the pair is deleted. Next, the distance is calculated with the normal matching, an adjustment to the upper bound is made if necessary, and the lower bound is calculated using the arboreal matching. If the change in upper bound is greater than 1E-2, a stagnation counter is reset; otherwise it increases until the stagnation limit is reached and γ is adjusted according to the amplification factor. The calculation stops if either the desired accuracy or the maximum amount of iteration as specified via the input parameter is reached. In both cases, the function returns a tuple which contains the number of iterations until the algorithm terminated, the lower bound and upper bound values (normalized and non-normalized) and the number of clades in the two trees combined.

5.4 External resources

As already mentioned, the tool uses a parser and a solver which I did not implement myself.

The parser is the parser of *Biopython*) chosen because *Biopython*) is already designed for use in bioinformatics and it offers excellent output options should the user desire to print one of these trees. Furthermore, I imagine users of the tool will use *Biopython*) already, so it would be redundant to use another package.

The solver is from the *scipy* package of the Python library and therefore easily accessible for nearly any Python user. However, most importantly, it is the only solver I could find which is designed to solve a maximum bipartite matching problem specifically.

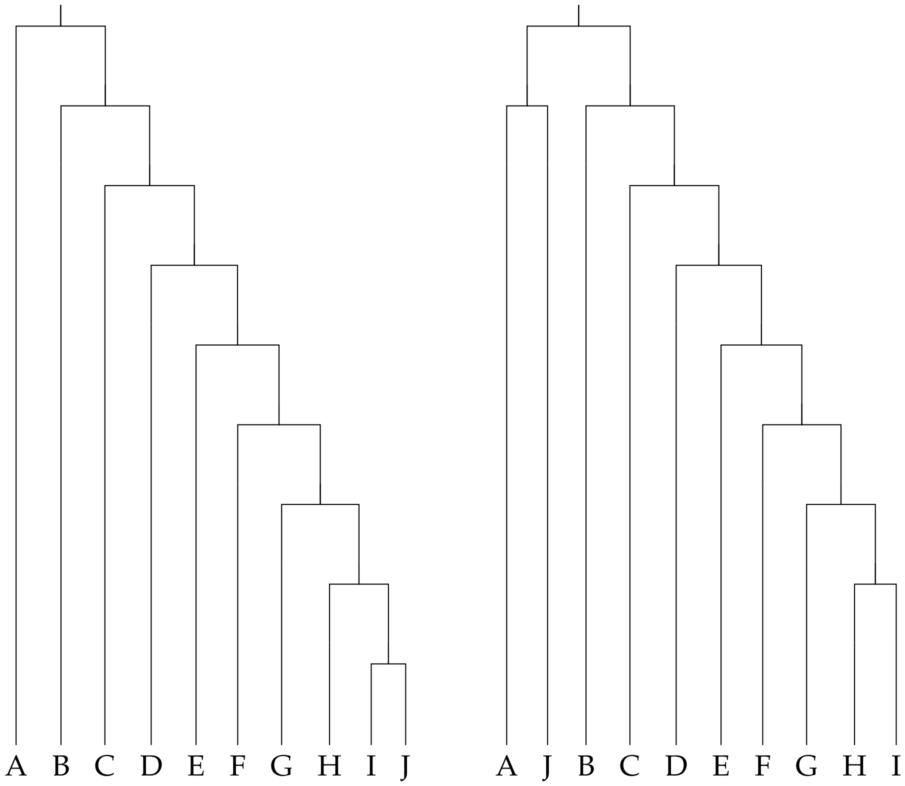

Figure 3: Two rooted phylogenetic trees T_1 (left) and T_2 (right) with ten leaves each over the set of entries S = {A, B, C, D, E, F, G, H, I, J}. The trees are analogous to the ones in Figure 1.

6 Evaluation

During the evaluation, the focus will be on three aspects: choice of the amplification factor, accuracy of the approximation, and a comparison to the metrics used by Beck et al. in their paper.

As a single comparison from Beck et al.'s data (88 clades in total) takes about 30 minutes for 100 iterations, the running time will not be further analyzed. It can be assumed to be pretty large for trees with a moderately high clade amount.

6.1 Example Data

To determine how accurate the approximation is for different amplification factors, data procured by running the code for two trees with ten leaves each, which have the same structure as those in Figure 1, will be used. The numbers in the trees from Figure 1 are replaced by letters to help with understanding (Figure 3).

That results in T_1 having the complete non-trivial clade set:

- {I, J}
- {H, I, J}
- {G, H, I, J}
- {F, G, H, I, J}
- {E, F, G, H, I, J}
- {D, E, F, G, H, I, J}
- {C, D, E, F, G, H, I, J}
- {B, C, D, E, F, G, H, I, J}

T_2's complete non-trivial clade set is:

- {A, J}
- {H, I}
- {G, H, I}
- {F, G, H, I}
- {E, F, G, H, I}
- {D, E, F, G, H, I}
- {C, D, E, F, G, H, I}
- {B, C, D, E, F, G, H, I}

In the first iteration the algorithm matches according to the JRF metric and gives the following matching:

- {I, J} – {A, J}
- {H, I, J} – {H, I}
- {G, H, I, J} – {G, H, I}
- {F, G, H, I, J} – {F, G, H, I}
- {E, F, G, H, I, J} – {E, F, G, H, I}
- {D, E, F, G, H, I, J} – {D, E, F, G, H, I}
- {C, D, E, F, G, H, I, J} – {C, D, E, F, G, H, I}
- {B, C, D, E, F, G, H, I, J} – {B, C, D, E, F, G, H, I}

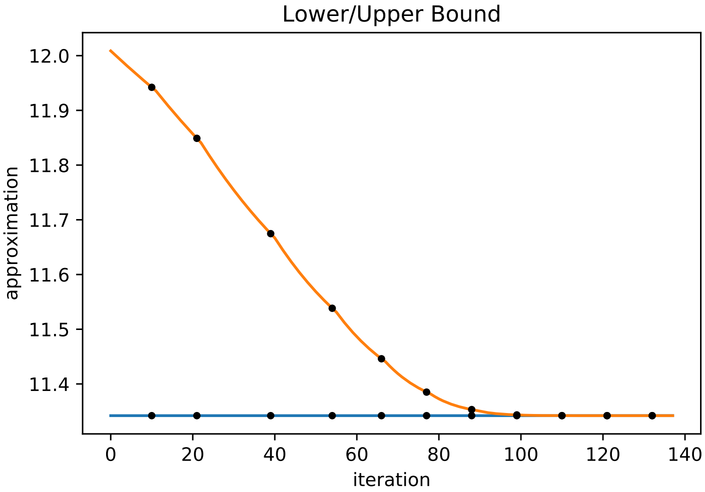

Figure 4: Lower and upper bound of the non-normalized result for the ILP for example trees with ten leaves each for 1000 iterations, stagnation limit 10, k = 1, amplification factor 1.5 and an allowed error of 1E-15.

Thereby I, J – A, J is in conflict with every other match since I, J is a proper subset of every other matched clade of T_1 but A, J is disjoint from every other matched clade of T_2. Therefore, this match is deleted to get an arboreal matching.

Fortunately, this is already the matching which gives the optimal score as a lower bound, so that this value only has to be approached by the upper bound. In Figure 4 the progression of this approximation is illustrated with the dots marking the points at which the stagnation limit is reached, so that γ gets adjusted.

Table 1 shows in which way the choice of the amplification factor influences the efficiency of the calculation: Choosing a factor of 1.0 implies no change of γ for the entire calculation. For this, I get an approximation of the value to a precision of 10^{-4} for 1000 iterations which is, while sufficient for most purposes, still not exact. To save computation time, the process can be sped up by increasing the amplification factor. Even an increase of only 10 percent yields not only a termination, but also one after merely 389 iterations instead of 1000 from before. Increasing the factor further lets the tool terminate even faster, e.g. after just 140 for a factor of 1.5. Nevertheless, the improvement is not infinite: setting the amplification factor too high will make the algorithm unstable; hence it will produce extremely high values in each step ($10^p, p > 100$) until Python is not able to process them any more; this will result in a runtime error. For the example trees, a factor unsuited in such a way is 1.6. However, decreasing γ (amplification factor < 1), will slow the process down and give relatively large intervals between the lower and upper bound.

As I have no information about which trees will be compared in future uses of this tool, I propose to set the default value for the amplification factor to 1.1: On the one hand, it is close enough to 1 that it is fair to assume that the stability is still given, and on the other hand a significant increase in both accuracy and running time is notable in comparison to a value of 1.

6.2 Data from Beck et al. (2018)

To analyze the data from Beck et al. (2018), it is prudent to first choose an amplification factor which will give good results. Because of the aforementioned running time issue, all evaluating will be done with 100 iterations. The exact solution for the JRF metric is obtained using a tool which calculates it without the Lagrangian Relaxation. It is available under *Github Repository of JRF* 2015.

A look at Table 2 shows that, as was already concluded in Section 6.1, the accuracy is greater for higher amplification factors. Notably, the tool is still working for an amplification factor of 2.5. Therefore it is possible to choose a quite high one. As I was unsure whether this holds true for all trees in the data set, I chose 2.0 as the amplification factor for the comparison between my results and the ones gained by Beck et al. in their paper.

Firstly, it is notable that upper and lower bound are fairly far from each other indicating that 100 iterations are not enough to get a reasonable approximation for this data set. However, the exact JRF distance is still mostly in the given interval. Exceptions are merely the "skeletal only" predicted ancestors (PA) and the "maximum preservation" PA for maximum likelihood (ML). Strangely, those two trees are virtually identical as they share the same clade set, but in Beck et al.'s analyzis they differ in normalized Robinson-Foulds distance (0.25 to 0.27; Table 3) as well as distribution coefficient (0.94 to 0.93; Table 3).

Generally, the lower and upper bound create an interval with a length of ca. 0.1 for trees obtained under ML or maximum parsimony (MP) with most parsimonious trees (MPT). MP with strict consensus, on the other hand, gives incredible large intervals of ca. 0.3; the algorithm seems to terminate much later for those. On a scale from 0 to 1, a divergence of 0.3 makes our approximation basically useless as it is not possible to make an accurate prediction of the distance. The reason behind this could be that the structure of said trees is not strictly binary as the other trees but include vertices with more than two children.

To illustrate how fast the approximation is, Figure 5 shows the lower and upper bound of the molecular tree from Meredith et al. (2011) and the first most parsimonious tree obtained under maximum parsimony for all characters. As seen, the lower bound does not change while the upper bound converges to the lower bound. Unfortunately, this lower bound does not correspond to the exact value of the JRF metric which means that the tool's approximation may be inaccurate for some trees. It is presumably because of the choice of estimation value. In further analyzis, the exact value of the JRF will be used as a stand-in for an exact value obtained via GRF metric.

Despite the suboptimal approximation, the results are generally speaking not bad: As desired the similarity indicated by the GRF is higher than the one indicated by the RF. All in all, it seems to be closest related to the SPR distance with values that are mostly about as close to 0 as the SPR distance is to 1 (Table 3).

Still, the general tendencies of the metrics are also reflected in the results with the GRF/JRF metric: None of the values could be described as being abnormally out of proportions compared to its counterparts since that they all show all character, skeletal only

Amplification Factor	Iterations	LB (value)	UB (value)	LB (normalized)	UB (normalized)
0.1	1000	4.077854793306871	4.657936507936508	0.25486592458167945	0.29112103174603177
0.3	1000	4.098713012075608	4.657936507936508	0.2561695632547255	0.29112103174603177
0.5	1000	4.134409642718156	4.657936507936508	0.25840060266988474	0.29112103174603177
0.8	1000	4.2876919715849375	4.657936507936508	0.2679807482240586	0.29112103174603177
1.0	1000	4.6579184396391415	4.657936507936508	0.29111990247744635	0.29112103174603177
1.1	389	4.657936507936508	4.657936507936508	0.29112103174603177	0.29112103174603177
1.2	241	4.657936507936508	4.657936507936508	0.29112103174603177	0.29112103174603177
1.3	182	4.657936507936508	4.657936507936508	0.29112103174603177	0.29112103174603177
1.4	156	4.657936507936508	4.657936507936508	0.29112103174603177	0.29112103174603177
1.5	140	4.657936507936508	4.657936507936508	0.29112103174603177	0.29112103174603177
1.6	-	-	-	-	-

Table 1: Example data results for different amplification factors with stagnation limit 10, k = 1, a maximum of 1000 iterations and an allowed error of 1E-15. ’Iterations’ gives the number of iterations before termination, i.e. accuracy to the 15th decimal point or 1000 iterations is achieved. Given are the actual and normalized of the lower (LB) and upper bound (UB).

Amplification Factor	Lower Bound (value)	Upper Bound (value)	Lower Bound (normalized)	Upper Bound (normalized)
0.1	10.8770531124283	17.400000000000006	0.12360287627759431	0.19772727272727278
0.3	10.89056009431873	17.400000000000006	0.12375636470816738	0.19772727272727278
0.5	10.914626745574694	17.400000000000006	0.12402984938153061	0.19772727272727278
0.8	11.012061437075545	17.400000000000006	0.12513706178494938	0.19772727272727278
1.0	11.20096670951311	17.400000000000006	0.12728371260810353	0.19772727272727278
1.1	11.379645898072766	17.400000000000006	0.12931415793264509	0.19772727272727278
1.2	11.582092960871407	17.400000000000006	0.13161469273717508	0.19772727272727278
1.3	11.648174045012297	17.400000000000006	0.13236561414786702	0.19772727272727278
1.4	11.76012524611889	17.400000000000006	0.13363778688771466	0.19772727272727278
1.5	11.80876429272692	17.400000000000006	0.13419050332644228	0.19772727272727278
1.6	11.977055855241687	17.400000000000006	0.13610290744592826	0.19772727272727278
1.7	11.761368750283083	17.400000000000006	0.13365191761685322	0.19772727272727278
1.8	11.826719951102646	17.400000000000006	0.13439454489889371	0.19772727272727278
1.9	11.92274490049897	17.400000000000006	0.1354857375056701	0.19772727272727278
2.0	12.022001540962762	17.400000000000006	0.13661365387457683	0.19772727272727278
2.1	12.12425607823252	17.400000000000006	0.13777563725264227	0.19772727272727278
2.2	12.22927094662829	17.400000000000006	0.13896898802986693	0.19772727272727278
2.3	12.336805711551392	17.400000000000006	0.14019097399490218	0.19772727272727278
2.4	12.44661796802383	17.400000000000006	0.14143884054572534	0.19772727272727278
2.5	12.558464230670609	17.400000000000006	0.1427098208030751	0.19772727272727278

Table 2: Comparison of the tree of Meredith and the first tree of Beck et al. (maximum parsimony, all characters, most parsimonious trees) for different amplification factors with stagnation limit 10, k = 1, an allowed error of 1E-15 and a maximum of 100 iterations.

MP	nRf	SPRd	DC	LB	UB	JRF
"all character" PAs MPTs	0.23	0.86	0.94	0.14	0.20	0.15
"all character" PAs strict	0.24	0.88	0.93	0.16	0.47	0.18
"skeletal only" PAs MPTs	0.23	0.86	0.94	0.14	0.20	0.15
"skeletal only" PAs strict	0.24	0.88	0.93	0.16	0.47	0.18
"craniodental only" PAs MPTs	0.46	0.70	0.85	0.25	0.39	0.34
"craniodental only" PAs strict	0.47	0.70	0.84	0.30	0.63	0.36
"dental only" PAs MPTs	0.54	0.63	0.76	0.34	0.52	0.45
"dental only" PAs strict	0.46	0.84	0.93	0.35	0.58	0.45
"max preservation" PAs MPTs	0.23	0.86	0.94	0.14	0.20	0.15
"max preservation" PAs strict	0.24	0.88	0.93	0.16	0.47	0.18
"typical preservation" PAs MPTs	0.46	0.72	0.86	0.23	0.36	0.32
"typical preservation" PAs strict	0.48	0.72	0.85	0.27	0.60	0.35

ML	nRf	SPRd	DC	LB	UB	JRF
"all character" PAs	0.21	0.81	0.95	0.16	0.25	0.20
"skeletal only" PAs	0.25	0.81	0.94	0.17	0.21	0.23
"craniodental only" PAs	0.32	0.81	0.91	0.23	0.35	0.31
"dental only" PAs	0.41	0.74	0.88	0.31	0.44	0.42
"max preservation" PAs	0.27	0.81	0.93	0.17	0.21	0.23
"typical preservation" PAs	0.43	0.74	0.85	0.24	0.38	0.34

Table 3: Summary of results from comparing the tree of Meredith et al. (2011) and trees of Beck et al. (2018) obtained under maximum parsimony (MP) and maximum likelihood (ML). The tool was used with a stagnation limit of 10, k being 1, an amplification factor 2, an allowed error of 1E-15 and a maximum of 100 iterations. The results are normalized with the Robinson-Foulds metric (nRF), the lower (LB) and upper bounds (UB) of the tool's calculation and the exact Jaccard-Robinson-Foulds metric (JRF) displaying more similarity for values closer to 0 whereas for the Subtree Prune and Regraft distance (SPRd) and the distortion coefficient (DC) values closer to 1 represent higher similarity. For MP the predicted ancestors (PA) are differentiated by trees recovered by strict consensus (strict) and most parsimonious trees (MPTs). The values for different MPTs of one method do not differ, therefore displayed once.

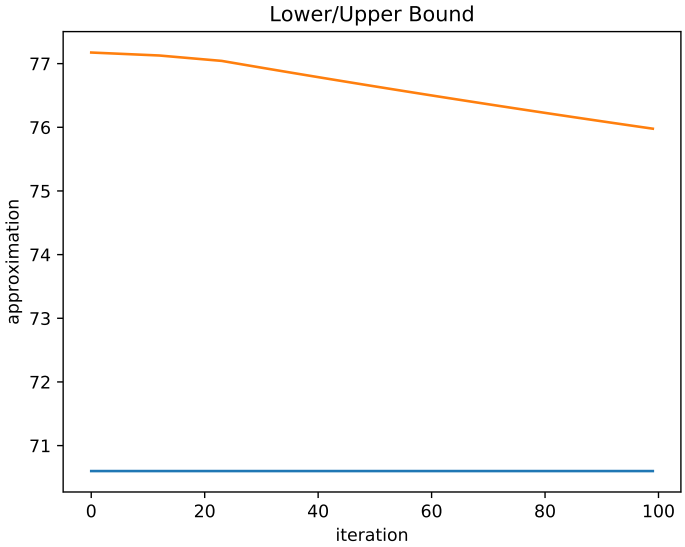

Figure 5: Lower and upper bound of the non-normalized result for the ILP for the molecular tree from Meredith et al. (2011) and the first most parsimonious tree obtained under maximum parsimony for all characters from Beck et al. (2018) for 100 iterations, k = 1, amplification factor = 2, stagnation limit = 10 and an allowed error of 1E-15.

and maximum preservation predicted ancestors to be the same for maximum parsimony (giving the highest similarity) with craniodental only and typical preservation PAs being quite close to each other in score but about double as bad as the former. For MP dental only gives for all metrics the lowest similarity (Figure 3). For maximum likelihood, the GRF/JRF metric deviates only about 0.02 from the normalized Robinson-Foulds metric for all PAs except maximum and typical preservation, but even these show similar trends.

6.3 Further expansion of the topic

To expand the tool further, it is possible to choose a different way of obtaining the estimation of an optimal score. As already discussed, the tool as it is takes the matching and deletes for every incompatible pair the match which yields a lower score. This method does not guarantee an optimal arboreal matching of the corresponding matching. For illustrating purposes, let assume the following example:

Given is a match A which is in conflict with two other matches B and C. A has a higher score than both B and C, but a lower one than B and C combined. According to the current implementation, both B and C will be deleted from the matching to gain an arboreal matching, even though deletion of A would yield a better result.

To resolve this, one could solve the selection of an arboreal matching from the given matching as a minimum vertex cover in which the vertices represent the matches and the edges the conflicts. The optimal solution of this minimum vertex cover is then corresponding to the matches which have to be deleted for an optimal arboreal matching. Problematic is only, that minimum vertex covers are NP-hard (Garey et al., 1979) which is precisely what was to be avoided. Nevertheless, it could be possible that the given tree structure reduces the cost of the minimum vertex cover so that it is applicable for our purposes. This would be a focal point for further research.

Furthermore, it is possible to use the Lagrangian Relaxation not in an Integer Linear Program but in a Branch and Bound algorithm as was proposed in Fisher (2004).

7 Conclusion

I have introduced a tool to compare rooted phylogenetic trees using the Generalized Robinson-Foulds metric with Lagrangian Relaxation. The implementation uses the Jaccard-Robinson-Foulds metric as a base sample and the Subgradient Method for the approximation. Different from the original Robinson-Foulds metric, it focuses not only on sameness, but also similarity to get a better understanding of the distance between trees. Nevertheless, it still preserves the tree structure. It is easily applicable as the necessary imports are quite straight-forward and do not require much knowledge of either coding, as well as the packages used. Furthermore, the intake enables a wide range of tree formats so that the user does not have to convert their data first. Those are significant advantages since the intended target group for the tool consists mainly of biologists who presumably do not have extensive skills in the field of computer science.

The results from the analyzis of the data given by Beck et al. show that while the tool is relatively slow for trees not pertaining a binary structure, the overall approximation is acceptable and records the same inclinations as the other metrics which were applied to the trees. Furthermore, the Generalized Robinson-Foulds metric yields shorter distances than the Robinson-Foulds metric in any case indicating a better detection of similarity.

It could also be interesting to see if any regularity is identifiable for employable amplification factors with regard to the stagnation limit and the number of clades.

Still unanswered is the question of how to improve the running time to make the tool less inconvenient as it does currently take an impractical amount of time to solve the problem for larger tree structures. One possible approach could be to try optimizing the selection of the estimated optimal value as this is not always accurate at the moment like it was discussed in Section 6.3. For this one would have to find a way without having to solve an NP-hard problem.

References

Sandro Andreotti (Sept. 2014). "Linear Programming and Integer Linear Programming in Bioinformatics". PhD thesis. Freie Universität Berlin.

Markus Johann Bauer (Apr. 2008). "A Combinatorial Approach To RNA Sequence-Structure Alignments". PhD thesis. Freie Universität Berlin.

Robin M. D. Beck and Charles Baillie (Dec. 2018). "Improvements in the fossil record may largely resolve current conflicts between morphological and molecular estimates of mammal phylogeny". In: *Proceedings of the Royal Society B: Biological Sciences* 285.1893, pp. 20181632–8.

Biopython Class Clade (2019). URL: http://biopython.org/DIST/docs/api/Bio.Phylo.BaseTree.Clade-class.htm (visited on 02/26/2019).

S. Böcker, S. Canzar, and G.W. Klau (2013). "The Generalized Robinson-Foulds Metric". English. In: *Lecture Notes in Computer Science* 8126. Proceedings title: 13th Workshop on Algorithms in Bioinformatics (WABI) Publisher: Springer Editors: A. Darling, J. Stoye, pp. 156–169.

M. L. Fisher (Dec. 2004). "The Lagrangian relaxation method for solving integer programming problems". In: *Management Science* 50.12 supplement, pp. 1861–1871.

Michael R. Garey and David S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co.

Github Repository of JRF (2015). URL: https://github.com/ls-cwi/JRF (visited on 02/26/2019).

Gitlab Repository of GRF with Lagrangian Relaxation (2019). URL: https://gitlab.cs.uni-duesseldorf.de/klau/bsc-thesis-lagrange-jrf/tree/master (visited on 02/26/2019).

Jotun Hein (Apr. 1990). "Reconstructing Evolution of Sequences Subject to Recombination Using Parsimony". In: *Mathematical biosciences* 98, pp. 185–200.

Michael Held and Richard M. Karp (Dec. 1971). "The Traveling-salesman Problem and Minimum Spanning Trees: Part II". In: *Math. Program.* 1.1, pp. 6–25.

Robert Meredith, Jan Janečka, John Gatesy, Oliver Ryder, Colleen A Fisher, Emma Teeling, Alisha Goodbla, Eduardo Eizirik, Taiz Simao, Tanja Stadler, Daniel L Rabosky, Rodney Honeycutt, John J Flynn, Colleen Ingram, Cynthia Steiner, Tiffani L Williams, Terence J Robinson, Angela Burk-Herrick, Michael Westerman, and William Murphy (Sept. 2011). "Impacts of the Cretaceous Terrestrial Revolution and KPg Extinction on Mammal Diversification". In: *Science (New York, N.Y.)* 334, pp. 521–4.

D.F. Robinson and L.R. Foulds (1981). "Comparison of phylogenetic trees". In: *Mathematical Biosciences* 53.1, pp. 131–147.

International Commission on Zoological Nomenclature., W. D. L. Ride, International Trust for Zoological Nomenclature., International Union of Biological Sciences. General Assembly, and England) Natural History Museum (London (1999). *International code of zoological nomenclature = Code international de nomenclature zoologique*. Vol. 1999. London :International Trust for Zoological Nomenclature, c/o Natural History Museum, p. 344.

List of Figures

List of Tables