An analysis of maximum parsimony algorithms to predict parasitism in Eukaryota

using a large multiurcated phylogenetic synthesis tree

Abstract

Anmerkung: Alle Figures richtig beschriften. Generell: Alle Figure captions sollten selbsterklärend sein. Sprich, ich sollte nur die Caption (normalerweise auch unter dem Bild) lesen und nicht den Rest das Papers und dann trotzdem verstehen können, was eigentlich passiert ist und - hilfreich - auch warum das relevant ist. (Bernhard)

Contents

1 Introduction

This paper is about the further development of parsimony algorithms for non-binary trees, applied to the currently largest phylogeny synthesis tree of Open Tree Of Life, with the application to the ancestral state reconstruction of parasitism.
Anmerkung: Der erste Satz muss nochmal überarbeitet werden. Das Ziel / Ergebnis der Arbeit hat sich inzwischen geändert
Anmerkung: Wir haben mehr einen Algorithmus getestet und an unserem spezifischen Problem angewendet, als viel selbst zu entwickeln. Allerdings haben wir den Fitch Algorithmus von binär auf multinär umgeschrieben.

Anmerkung: Mein Vorschlag einer Gliederung (jeweils ca. ein Absatz) (Bernhard)

i) Motivation:

- Was ist das große Ziel?
  Das Ziel dieser Arbeit ist die Anwendung von maximum parsiomony algorithmen auf nicht binäre Bäume und auf sehr große Datensätze. Insbesondere auf das Beispiel 'Entstehung des Parasitismus' im ganzen Eukaryotischen Tree of Life.
- Was soll erreicht werden?
  Wir wollen vorhandene Algorithm (Sankoff/castor) auf diese Aufgabenstellung hin testen und ihre Vorhersagekraft abschätzen. Außerdem wollen wir den Fitch algorithmus für binäre Bäume auf unser Problem erweitern und mit dem Sankoff Algorithmus vergleichen.
- Warum ist das relevant? Was könnte man dann tun?
  predict states of species...
  TODO: !!!

ii) Hintergrund:

- Was gab es in dieser Richtung bereits als ganze Ansätze oder wenn nicht, warum nicht? Woran ist es bisher gescheitert?
  Bisher wurden vorallem Algorithmen für das binäre Problem entwickelt, da man wesentlich kleinere Teilbäume betrachtet hat, von welchen man auch alle Aufspaltungen kennt. Durch die Entwicklung von OTL, eines gesamten Baum des Lebens, ergibt sich das Problem, dass dieser bei weitem nicht binär ist.

  Researchers of the phylogenies have been dealt with the ancenstral state reconstruction in the 60s. The first methods were only brute force TODO: Quelle, siehe Fitch: Camin and Sokal 1965 . Next came a set of parsimony algorithms such as: Fitch-parsimony [1], Wagner-parsimony [2] ... TODO: weitere? .
  With more and more data, there is now the possibility to use more information to calculate the probabilities of the ancestral states. In addition to the states of the leafs, algorithms could also use branch lengths. The likelihood based algorithms came more in interest.
  Our focus came with another 'data extension'. We wanted to work with the biggest phylogenetic tree that exists at this moment, which goes over all observed species. For most TODO: most? species there is no phylogeny, but only a taxonomic classification.

- Welche Grundlagen sind notwendig:

  - open tree of life: Was ist das, warum relevant und überlegen als reine Ansätze?
    TODO: !!!

    So the biggest 'phylogenetic tree' is a synthesis of phylogenetic trees filled with a taxonomic tree given by Open Tree of Life [3]. This tree is not binary and therefore the developed algorithms are not directly applicable.

  - Algorithmen: Was gibt es? Ruhig ausführlicher als hier bereits und vor allem auch nach einer Darstellung am Ende ableiten, was für uns relevant ist. Also

beschreiben, wie Methode a, b, c funktionieren und dann abwägen, was daher für Dich am relevantesten ist.

TODO: !!!

Anmerkung: GloBI und OTL in der Einleitung vorstellen. (Emanuel)

iii) Outlook/Structure of this work

In this work, we have looked at the algorithms that are generally suited to our data, to develop them further for the not binary case, and finally to compare their usability with our sythesis tree.
We have decided to consider only parsimony algorithms since we have no information on branch lengths and no other additional information like different transition probabilities of our states.

1.1 Definitions

- Parasit - Freilebend
- Multifurkation - binär
- height (min, max, mean), depth of a tree/node (Distanz zur Wurzel vs distanz zum Blatt)
- maximum parsimony
- OTL, OTT, GloBI

2 Methods

As initiated, we would like to apply a maximum parsimony algorithm to the entire tree of life to obtain an ancestral state reconstruction of free-living versus parasite states.
So far, these reconstructions have been made mainly on binary trees with better data availability. Therefore, we decided to use a simulation to decide how to evaluate the existing algorithms and possibly adapt them to our given problem.
Accordingly, in addition to the necessary data sets (GloBI, OTL), the chosen algorithm and the evaluation of its results, this chapter also deals with the previously performed simulation and the evaluation of the various algorithms and their parameters.
Figure 2.1 briefly outlines these relationships.

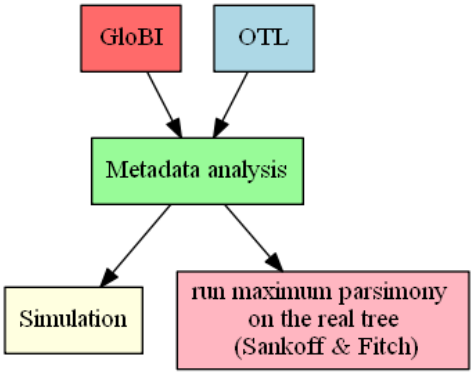
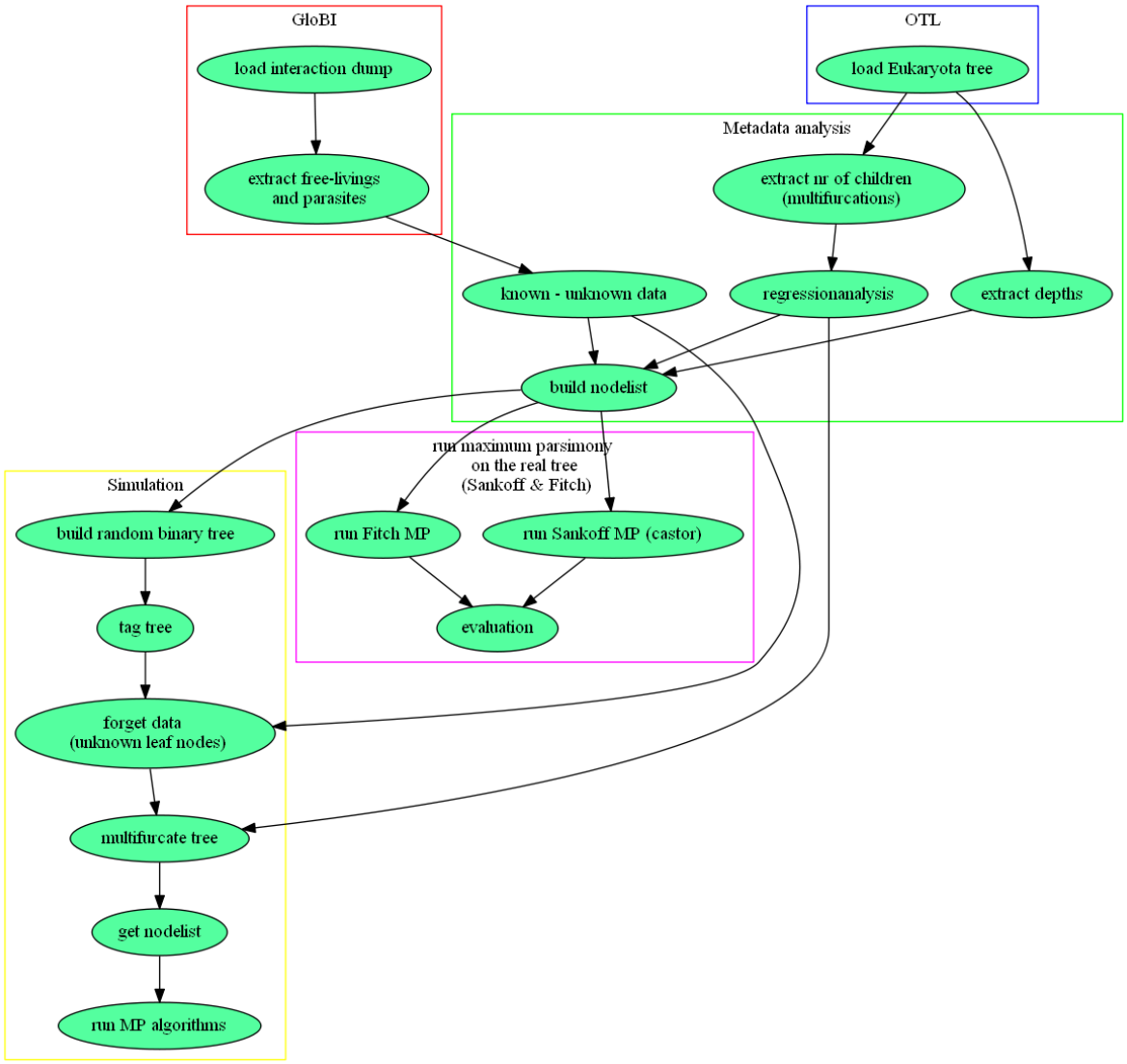


Figure 2.1: Workflow

The coming sections are thus subdivided into the following topics:
TODO: or: The resulting procedure is as follows:
(1) Get the real tree and real data for the leaf nodes → OTL, GloBI databases. (2) Get metadata of these for a realisitc the simulation. (3) Build and run the simulation. (4) Evaluation of parameters for the simulation and the real problem. (5) Run the resulted algorithm on the original data. (6) Evalute and interprete results. → Origins etc...

TODO: Metadata analysis nötig für run maximum parsimony?

Figure 2.2: Workflow



2.1 Get data - Properties of real Data

For our research we need two types of data: a tree and information about the states.
For the tree we decided to use Open Tree of Life (OTL), because it's the biggest available

synthesis tree.

TODO: hier referenzen zu nem paper das das bestätigt o.ä.

For the state information, we decided to use the Global biotic interaction database (GloBI). Also in this case, this is one of the largest databases and both OTL and GloBI support the OTT identification. OTT (open tree taxonomy) is a taxonomy that assigns to each species a unique id, both ancestor and now living species (internal and leaf nodes).

2.1.1 OTL

For our project we looked for a large database for phylogenetic trees and also for a taxonomic tree. Since we run our algorithm on the phylogenetic tree, and for the evaluation and other properties the taxonomy provides us with much more information.
OTL gives us both. A synthesis of phylogenetic trees (currently 819 trees) and a taxonomic tree. OTL also includes the large phylogenetic database TreeBASE [3].

TODO: Das steht auf der Website nicht in dem Paper...

For phylogenetic data, there are at least five big data collections, namely: ITIS (Integrated Taxonomic Information System) [4], NCBI (National Center for Biotechnology Information) [5], WORMS (World Register of Marine Species) [6], GBIF (Global Biodiversity Information Facility) [7], OTT (OpenTreeOfLife-Taxonomy) [3].
TODO: Marius: "Every dataset has it's own characteristics and downsides. ITIS is only a small set of 100% confirmed and named species. GBIF is not composed with the help of phylogeny, the same is valid for the NCBI taxonomy. The WORMS taxonomy is a way too small dataset of mostly marine species.
We choosed the taxonomy from OpenTreeOfLife because it's including most of the known taxonomies and got synthesised by preffering taxonomies that match with available phylogenetic data. At the same time the team from OTL preferred a maximum number of species [3]. This is resulting in somekind of hybrid between taxonomy and phylogeny."

Distribution of Taxa

- In our tree we can distinguish 28 different Taxa with the OTL taxonomic tree.
- The most of them are hardly represented. The major taxonomic groups are: ...
- Here you can see some characteristics of the Multifurcation of the tree.
  In a phylogeny, the taxonomic division of the tree is far too coarse, meaning that there

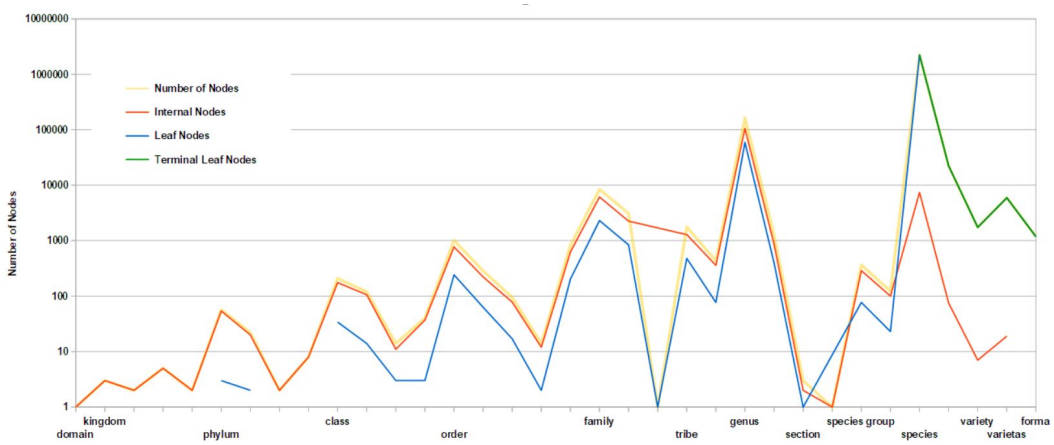


Figure 2.3: Distribution of Nodes in Rank-Cathegories

should be more subtaxa or 'unranked' nodes. But the closer we get to the root, the more the pure taxonomic tree is reflected. If the tree were binary, the taxa would have to double. But the multipliers for some are much bigger and for others much smaller, which you can see in in figure 2.3.
Please see the appendix on page 40 for a complete table 5.2 of these data.
TODO: Was hierzu ist richtig und wichtig?

Distribution of data in the taxa

Mithilfe des taxonomischen Baums von OTL haben wir die Knoten ihren Kingdoms, Phyla und Classes zugeteilt.
Please see the appendix on page 39 for a complete table 5.1 of these data.
TODO: gibt es einen Zusammenhang zwischen Anzahl
TODO: max max height zu anzahl nodes in phylum plotten? oder mean max height oder ... (mean, min

TODO: depth...

2.1.2 GloBI

TODO: Marius: "There aren't many big active interaction databases out there, most of them are offline or outdated. For example: IWDB (Interaction Web Database) [8], Webs on the Web [9], Animal Diversity Web [10] and ecoweb [11]. GloBI is including most of the known ones and is still growing actively [12]. So the question which interaction database could be used was answered rather quickly."

This database consists of entries of the form: species A (source) interacts with B (target). We appointed some interactions, where we know from the biological perspective that the species source or target has to be a parasite or a free-living species. These are the following:

- free-living source: preysOn, eats, flowersVisitedBy, hasPathogen, pollinatedBy, hasParasite, hostOf
- free-living target: preyedUponBy, parasiteOf, visitsFlowersOf, pathogenOf, hasHost
- parasite source: parasiteOf, pathogenOf
- parasite target: hasParasite, hasPathogen

TODO: Interactions nochmal prüfen! Darauf basieren unsere Ergebnisse!
We build two lists: parasites and free-livings, and add the source or targets of an interaction to these.
TODO: klar? Oder Beispiel bringen? (Katze isst Maus → Katze ist Freilebend)
TODO: einige spezies nicht mit einbezogen, da sie keine OTT id haben, hier könnte man noch verbesser
TODO: You can find all interaction types here: https://github.com/jhpoelen/eol-globi-data/blob/master

With this we got ~ 51000 (distinct) freeliving species and ~ 47000 (distinct) parasite species (see section countings) TODO: ref einfügen . But we found also ~ 57000 (not distinct) source species and ~ 810000 (not distinct) target species without OTT ids. Since we currently use only OTT ids, we could not use this information.
TODO: mehr dazu in section: unknown nodes...

2.1.3 Countings

Name	Number of
Eukaryota	241 974 internal nodes, 2 293 463 leaf nodes
interactions	5 346 414
freeliving species	51 337 (distinct)
parasite species	47 332 (distinct)
$\rightarrow$ unknown nodes	???
unused possible species	57 352 (source), 809 993 (target)
Nr of Children $== 1$	55 700 51 744 (height $= 2$) 3 956 (height > 2)

TODO: tree, subtrees? figure dazu?
$\frac{100}{2293463*(51337+47332)} \approx 4.3 \rightarrow$ only 4.3% of leaf nodes are filled with information

#interactions zu #Parasiten und #Freilebend $\rightarrow$ Wieviel gibt GloBI her? (dazu noch #unused possible species, wieviele Parasiten ohne ott haben wir gefunden?) #Parasiten und #Freilebend und #Blattknoten $\rightarrow$ Wieviele unbekannte Knoten haben wir? #Interne Knoten und #Blattknoten $\rightarrow$ Wie stark ist die Multifurkation?

Data artifacts

Zu der Frage, warum gibt es $nrChildren == 1$. Es gibt tatsächlich 55700 Knoten mit nur einem Kind. Davon ist der großteil direkt vor einem Blatt (51744), aber 3956 ($height > 2$) sind irgendwo im Baum. Ein paar Beispiele:
Nephroselmidophyceae: (class)
https://tree.opentreeoflife.org/opentree/argus/ottol@1038762
Phrynocrinidae: (family)
https://tree.opentreeoflife.org/opentree/argus/ottol@3647979
Elaeocarpus sylvestris:
https://tree.opentreeoflife.org/opentree/argus/opentree9.1@ott166969
Was bedeutet das für unsere Analyse? Ignorieren wir die $\sim$ 4000 internen und Zählen sie zu den Blättern, oder zählen wir alle zu den Internen oder puzzlen wir sie einzeln raus. D.h. nur die mit der Zusatzeigenschaft ($height == 2$) sind auch Blätter.
Es könnte aber auch Fälle geben wie: ...-O-O-O-O-O, dann könnte man auch einen Knoten mit höherer Höhe als Blatt zählen. Das müsste dann gelten $minheight = maxheight$.

```
leaf.taxa <-  all.taxa[all.taxa$nr_children==0, ]
extendedLeaf.taxa <- all.taxa[all.taxa$nr_children<2 & min.height==max.
inner.taxa <- all.taxa[all.taxa$nr_children>1 | (all.taxa$nr_children==
```

TODO: macht extendedLeaf wirklich einen Sinn?

2.2 Metadata analysis

Um eine möglichst realistische Simulation zu erzeugen haben wir auf der einen Seite einige Daten gesammelt (vorheriges Kaptiel), und außerdem beeinflussende Parameter untersucht. Wir haben zwei große Arten von Parametern:

i) Biological parameters (A result of the evolutionary process.):

- transition probabilities

ii) Distribution of the loss of information:

- Loss of topology ($\rightarrow$ mutlifurcations).
- Unknown information about the states of some leaf nodes.

We tested the influence of these parameters on our result using our simulation (Section 2.3).

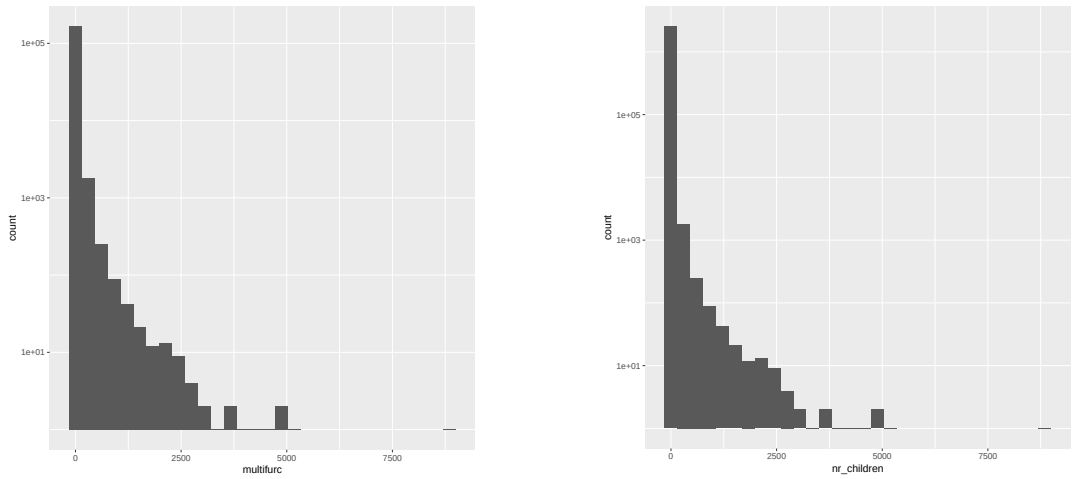
2.2.1 Transition probabilities

Anmerkung: Wir hätten aber maximal 2 beta-Verteilungen mit jeweils 2 Parametern und je einem eigenen Schwellenwert bei dem in die andere Verteilung gewechselt wird. Also 6 Parameter. Ich finde den Parameterraum den du bisher betrachtest zum einen nicht systematisch erfasst, zum anderen wahrscheinlich auch zu klein. Du kannst -Parameter sparen- indem zu z.B die beiden Verteilungen und cut-offs symmetrisch (gespiegelt an 0.5) machst. Solltest aber meines Erachtens die Schwellenwerte freier variieren. Das können wir bis zu unserem nächsten Treffen zurückstellen und dann gemeinsam diskutieren. (Emanuel)
Das mit dem symmetrisch spiegeln geht halt nur bei maximal 40-60 Verteilungen richtig gut, danach haben brauchen eventuell eine lange Austestphase, bis ein Baum dieser Verteilung entspricht. Wenn wir allerdings dafür zwei verschiedene Schwellenwerte einführen könnte das dieses Problem eventuell ausgleichen. Das müsste ich austesten.

TODO: Kapitel tag tree hier einbinden

2.2.2 Multifurcation

One property of the tree is its ridge of multifurcation. A complete kinship tree would be binary, but since we only work with a synthesis tree of individual trees, this tree is multifurcated. For a first overview we collected for every node its number of children (degree -1), and plottet this in an histogram. In Figure 2.4 you can see how much nodes we have for every degree.
TODO: Subfigure 2.4b beschränkt sich hierbei nur auf die inner nodes während Subfigure 2.4a alle Knoten abbildet.
TODO: Welchen der beiden Plots wollen wir nehmen, einer genügt eigentlich: As you



(a) all nodes **(b)** inner nodes (shiftet to 0)

Figure 2.4: plots of children of nodes

can see, we are very far from a binary tree. Therefore, the impact of multifurcations on our research question has been investigated:

- stärke der Multifurcation in verschiedenen Subbäumen
- zusammenhang zur tiefe im Baum
- zusammenhang zu höhe im baum (max, min, mean)
- einfluss auf die vorhersage des Castor und Fitch Algorithmus (Simulation) TODO: ...

"Nested models were compared using likelihood ratio tests, models using different predictors were compared according to their deviance and AIC."

2.3 Simulation

Anmerkung: Motivate the goal of the simulation (Bernhard)

- build random binary trees, tag these (parameters: parasites vs free-living, beta-distribution)

- run fitch-parsimony, wagner-parsimony, our parsimony like algorithm
- build not binary tree (poisson distribution?)
- run new algorithms
- compare trees (distances)

i) build random binary trees

ii) tag tree

iii) multifurcate tree

iv) run maximum parsimony algorithms
    - Fitch
    - Sankoff (Castor package)
    - my algorithm

v) Evaluation

2.3.1 random binary tree

Anmerkung: Again, motivate first, why this is required and why you choose this solution (Bernhard)

To get a random binary tree, I used the Phylo package from biopython. They offer a randomized function which returns a BaseTree [1]:

```
from Bio import Phylo
Phylo.BaseTree.Tree.randomized(number_leaf nodes)
```

From the BaseTree class:

Anmerkung: trivial, does not give real info (Emanuel)

[1] https://github.com/biopython/biopython/blob/master/Bio/Phylo/BaseTree.py

```
def randomized(cls, taxa, branch_length=1.0,
        branch_stdev=None):
"""Create a randomized bifurcating tree given a list
        of taxa.
  :param taxa: Either an integer specifying the number
        of taxa to create (automatically named taxon#),
        or an iterable of taxon names, as strings.
  :returns: a tree of the same type as this class.
"""
```

TODO: Zitat von BaseTree und buildTree.py

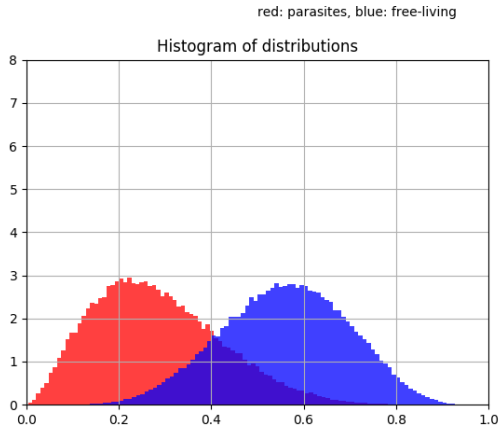
2.3.2 tag tree

Anmerkung: instead of 'tag tee' simulating states and transitions between them (Emanuel)

At this point we want one fully tagged tree, and one less tagged tree which looks like our real data.
Let's say the first specie (the root node) was free-living (start with a parasite without a host makes no sence). For every transition from a node to his child, we take a random number from the father distribution. We decided that from the biological perspective a beta distribution reflects our transition probabilities best (see Figure 2.1 TODO: ref einfügen).
Anmerkung: why is that the case and why is that from a biological perspective? (Bernhard)

Anmerkung: I'd rather say that to ensure that the parameter of the binomial distribution is restricted to the [0,1] interval, we model it... (Bernhard)
For example when our father node was free-living, then we take from the free-living beta distribution. Is the number under the threshold for beeing parasite, we get a change and tag the current node as parasite. Otherwise we tag it as free-living.
With this procedure we traverse through the tree from the root to every leaf node. A part of this code you see here:

Figure 2.5: 60% Free-living - 40% Parasites




```
from numpy import random
if father_tag == 0:
  # freeliving_distribution:
  new_random = random.beta(a=A_FL, b=B_FL)
else:
  # parasite_distribution:
  new_random = random.beta(a=A_P, b=B_P)
tag = 0          # $\rightarrow$ FL
if new_random < percentage_parasites:
  tag = 1        # $\rightarrow$ P
```

TODO: Bessere Beschriftung, Plot neu erstellen! U.a. mit threshold

We save each tag with the associated node ID in a nodelist.

Anmerkung: simulationg loss of information (1) states (2) topology/multifurcation (Emanuel)

The real tree has much less information, we have only information from some current species (leaf nodes) and TODO: and probably negligible internal nodes .
To simulate our real tree we save for every node an empty placeholde except for some leaf nodes. There we save the states again. The amount of these unknown information is one parameter, which we got from our real tree. Or which we can change to TODO: ...

TODO: Was hiervon gehört ind Methoden, was schon in Implementierung oder ganz woanders hin?

2.3.3 multifurcate tree

Anmerkung: simulating loss of information (Emanuel)

Another parameter is the nature and strength of the multifunction of the tree, since we do not have a binary tree in the real case. After several measurements and analyzes, which we explain in TODO: section/chapter x , Anmerkung: fit, justification (Emanuel) we decided to use a $\frac{1}{x}$ distribution, where x is the depth of a node. This means, how deaper we are, how less information we have.

We traverse through the tree and pick a random number between 0 and 1. If random number is smaller as our limit ($\frac{1}{x}$), than we forget the node and hang every child to the father node of the current node.

Anmerkung: poisson process → fit that distribution, include depth as a predictor, see if significant (Emar

---

```
  from numpy import random
  from utilities import Helpers

  def get_non_binary_tree(subtree, nodelist):
    i = 0
    while i != len(subtree.clades):
      if subtree.clades[i].is_terminal():
# is leaf node?
        i += 1
      else:
        element = Helpers.find_element_in_nodelist(subtree.clades[i].name
        limit = get_limit(element[1])
        new_random = random.uniform()
# choose if we want to delete ourselve
        if new_random < limit:
# or new_random < 0.9:
          subtree.clades += subtree.clades[i].clades
# add children
```

```
            del subtree.clades[i]
# delete internal node
        else:
# if we don't deleted ourselve go on with children
            get_non_binary_tree(subtree.clades[i], nodelist)
# otherwise the children are in the current clade array
            i += 1
    return

    def get_limit(depth):
      limit = 1 - 1 / ((depth + 3) / 4)
      if limit < 0.1:
        limit = 0.1
      return limit
```

Wir lassen das Limit nicht beliebig klein werden, sondern beschränken es auf 0.1.

2.3.4 maximum parsimony algorithms

Fitch maximum parsimony

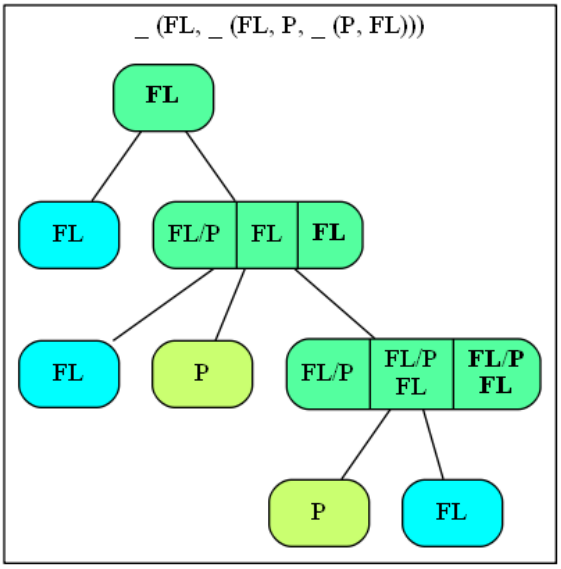
Described from [COO98] + others ... - implemented for multifurcating trees

Fitch algorithm for binary trees:

Der Baum hat die folgende Struktur: Alle inneren Knoten sind leer. In den Blattknoten befindet sich entweder das Tag FL oder P, oder deren Vereinigung, wenn es sich um einen unknown node handelt.

Der Fitch Algorithmus ist aufgeteilt in drei Schritte, in welchen man jeweils durch den Baum traversiert. Schritt 1 beginnt von de Blättern aus, da sich dort zu Beginn die einzige Information befindet. Für jeden Knoten gilt, wenn seine Kinder schon Information enthalten,

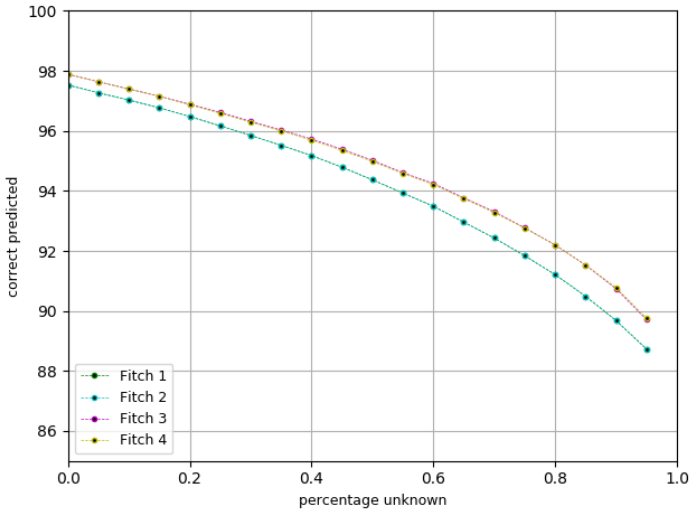
Figure 2.6: bla



dann bilde die Schnittmenge der states und schreibe diese als Information in den aktuellen Knoten. Ist die Schnittmenge leer, dann schreibe die Vereinigung aller möglichen states in den Knoten. Für alle Kinder, die noch keine Information haben, führe diesen Schritt erst für diese aus. Schritt zwei geht von den Kindern der Wurzel bis zu den Vätern der Blätter. Jeder Knoten bekommt einen zweiten Tag, der sich aus der Vereinigung des states des Vaterknoten und der Geschwisterknoten zusammensetzt. Ist diese leer, bekommt der Knoten wieder die Vereinigung aller states, also $\{FL, P\}$ als Tag.

Hier gibt es einige Möglichkeiten, wie dieser Schritt genau aussieht. 1. Version: Es wird nur der erste Tag vom Vaterknoten genutzt. Außerdem wird von den Geschwisterknoten zuerst der Schnitt gebildet, und danach vom Ergebnis nochmal mit dem Vaterknoten zusammen. (Immer wenn der Schnitt leer ist, ist das Ergebnis die Vereinigung aller states, also $\{FL, P\}$. Auch im folgenden...) 2. Version: Es wird nur der erste Tag vom Vaterknoten genutzt. Er wird zusammen mit den Geschwisterstates genommen und direkt ein Schnitt aller Mengen gebildet. 3. Version: Es werden alle vorherigen states vom Vater genutzt und von diesen ein Schnitt gebildet. Das selbe gilt für die Geschwisterstates. Und dann wird ein dritter Schnitt zwischend en Ergebnissen gebildet. 4. Version: Es werden alle states genutzt und direkt in einem Schnitt zusammengenommen.

Figure 2.7: bla



Der Finale Schritt traversiert nochmal über den Baum und Bildet aus den zwei states pro Knoten einen finalen Tag, indem wieder der Schnitt der beiden states das Ergebnis ist.

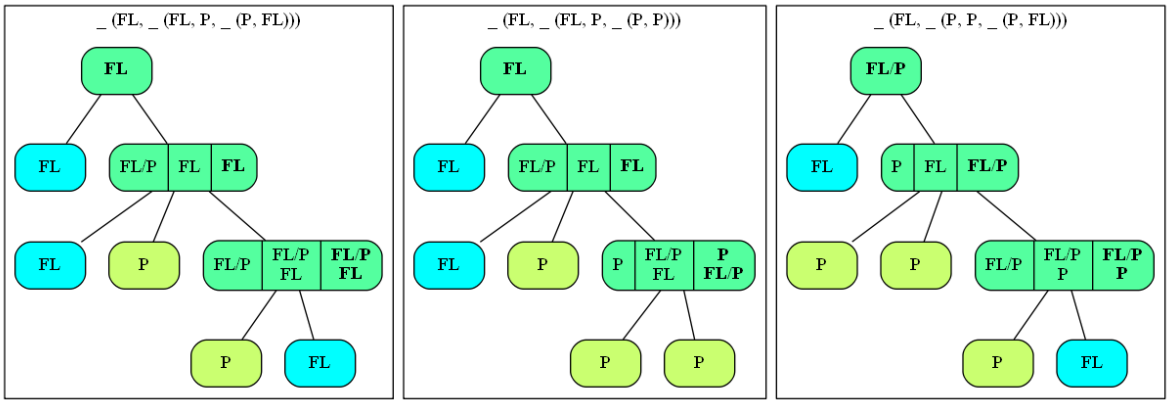
Ich habe diese Versionen mit 100 Bäumen mit 10000 Blattknoten und der Verteilung 60% FL zu 40% P simuliert. Bei 90 % unbekannten Knoten lag Version 1 zu 89.67 %, Version 2 zu 89.67 %, Version 3 zu 90.72 % und Version 4 zu 90.74 % richtig.

How to extend Fitch for multifunction?:

Sankoff

Maximum parsimony algorithm from Sankoff implemented in the R package castor.

Figure 2.8: bla



my Algorithm

2.4 real data analysis

- Import tree
- Import interactions
- run castor algorithm / and others?
- interprete results (leave one out)

2.5 Implementation

3 Results

Ein großer Punkt in diesem Kapitel ist das Ergebnis der Untersuchung der Datenlage. Wie ist der Stand? Welchen Einfluss hat das auf unser eigentliches Ergebnis? Was kann man dagegen tun?
Ansonsten geht es in diesem Kapitel hauptsächlich um die tatsächliche Rekonstruktion der states. Dass heißt zum einen der der inneren Knoten und dadurch der Untersuchung von Origins und Losses und zum anderen der Vorhersage von unbekannten states von Leaf nodes. Und natürlich dem Ergebnis der Simulation.

3.1 Metadata analysis

3.1.1 Taxa

Die Untersuchung der Taxa der Eukaryoten hat das folgende ergeben: Es gibt 3 Kingdoms: Chloroplastida, Metazoa, Fungi.
TODO: Hier fehlt nichts. Ref einfügen.
Phyla (53):
Acanthocephala, Amoebozoa, Apicomplexa, Arthropoda, Ascomycota, Bacillariophyta, Basidiomycota, Brachiopoda, Bryozoa, Chaetognatha, Chlorophyta, Chordata, Chromerida, Chytridiomycota, Ciliophora, Cnidaria, Colponemidia, Ctenophora, Cycliophora, Echinodermata, Entoprocta, Entorrhizomycota, Euglenida, Foraminifera, Gastrotricha, Glomeromycota, Gnathostomulida, Haplosporida, Haptophyta, Hemichordata, Kinorhyncha, Loricifera, Microsporidia, Mollusca, Myzostomida, Nematoda, Nematomorpha, Nemertea, Onychophora, Orthonectida, Phaeophyceae, Picozoa, Placozoa, Platyhelminthes, Porifera, Priapulida, Rhodophyta, Rhombozoa, Rotifera, Streptophyta, Tardigrada, Xanthophyceae

Wobei von Streptophyta -> Anthocerotophyta, Marchantiophyta, Bryophyta, Tracheophyta als Phylum im Phylum gefunden und nicht einbezogen wurden und Magnoliophyta als Phylum in Tracheophyta ebenfalls nicht.
Außerdem haben wir 195 Klassen und 924 Ordnungen feststellen können.

3.2 Multifurcation

3.2.1 Poisson regression

```
glm(formula = multifurc ~ 1, family = "poisson", data = inner.taxa)
```

The intercept is $2.821 > 0 \Rightarrow$ there is a multifunction. (Intercept: Stärke der Multifurcation)
Comparing the different kingdoms, we find that multifunctionality is greater in Fungi than in Chloroplastida than in Metazoa:

$$4.0999(FungiIntercept) > -0.9132(ChloroplastidaIntercept) > -1.4320(MetazoaIntercept)$$

Wir haben außerdem drei komplexitätsstärken von Modellen verglichen bezüglich der höhe und tiefe des Baums mit dem folgenden Deviance Table:

Model	Residuals Dev	Deviance Pr
multifurc ~ kingdom	7774454	
multifurc ~ phylum	7435700	338754
multifurc ~ class	7337241	98459
multifurc ~ order	7076068	261172
multifurc ~ phylum + depth	7431609	
multifurc ~ phylum + max.height	7375889	55721
multifurc ~ phylum + min.height	7233486	142403
multifurc ~ phylum + mean.height	7128318	105167
multifurc phylum * depth	7335396	
multifurc phylum * max.height	7311241	24155

multifurc	phylum * min.height	7177002	134238
multifurc	phylum * mean.height	7020258	156745
multifurc	phylum	7435700	
multifurc	phylum + mean.height	7128318	307381
multifurc	phylum * mean.height	7020258	108061
multifurc ~ class + depth		7334754	
multifurc ~ class + max.height		7275856	58898
multifurc ~ class + min.height		7144686	131170
multifurc ~ class + mean.height		7055313	89374
multifurc	class * depth	7250759	
multifurc	class * max.height	7187504	63256
multifurc	class * min.height	7094933	92570
multifurc	class * mean.height	6965794	129139
multifurc	class	7337241	
multifurc	class + mean.height	7055313	281928
multifurc	class * mean.height	6965794	89518
multifurc ~ order + depth		7027578	
multifurc ~ order + max.height		7005424	22154
multifurc ~ order + min.height		6890703	114721
multifurc ~ order + mean.height		6815271	75432

* Residuals: Fehler - wieviele Werte sind nicht gut modelliert. (umso kleiner umso besser - grün)
* Deviance - Abweichung

```
  class * depth / min/max/mean.height: Warning message: glm.fit: fitted r
  order * depth / min/max/mean.height: Error: cannot allocate vector of s
```

TODO: Nur ein Ausschnitt hier und dann... You can find the whole Deviance Table in the Appendix...

Interpretation: Die Multifurkation ist sehr ungleich verteilt. Daher ist die vorhersage umso genauer umso kleinere Subtrees wir betrachten. ...

```
Model 1: multifurc ~ kingdom
Model 2: multifurc ~ phylum
Model 3: multifurc ~ class
Model 4: multifurc ~ order
  Resid. Df Resid. Dev  Df Deviance  Pr(>Chi)
1    167222    7774454
2    167173    7435700  49   338754 < 2.2e-16 ***
3    167039    7337241 134    98459 < 2.2e-16 ***
4    166352    7076068 687   261172 < 2.2e-16 ***
```

Anmerkung: different parameters for different taxa? (Emanuel)

3.3 Results of castor

3.3.1 Biological view

TODO: Castor replaces originaltags with finaltags. There are 82 originaltags != finaltag.

We picked a few phyla to evaluate the results from the biological point of view.
Table 3.2 shows some known phyla: Chordata, Nematoda, Platyhelminthes and Apicomplexa. As we know TODO: ref einfügen the Chordata are full of free-living species and there are only a few parasites. The algorithm reflects this. We started with 99.83% free-living species and predicted 99.94% species as parasites TODO: (inklusive all known nodes) . Only 0.06% were predicted as parasites. These few parasites are mostly breeding parasites (brood parasitism) and some individual errors from the GloBI database.

Phylum	# nodes	original states FL	original states P	final states 0 (FL)	final states 0.4	final states 0.5	final states 0.67	final states 0.75	final states 1 (P)
Chordata	91785	10451 99.83%	18 0.49%	91734 99.94%	0	0	0	0	51 0.06%
Nematoda	30127	21 0.63%	3289 99.37%	791 2.63%	0	1017 3.38%	0	0	28319 94%
Platyhelminthes	22683	7 0.1%	7086 99.9%	949 4.18%	0	151 0.67%	0	0	21583 95.15%
Apicomplexa	1863	1 0.39%	255 99.61%	1 0.05%	0	0	0	0	1862 99.95%
Arthropoda	1198981	18912 62.93%	11141 37.07%	1099509 91.7%	1313 0.11%	22478 1.87%	4176 0.35%	1665 0.14%	70223 5.86%

Table 3.2: Phylum (leaf nodes)

Same observation but with less free-livings is the Apicomplexa Phylum. Here we have only a few free-livings TODO: ref einfügen . And as we see, we had good start data and predicted 00.95% as parasites.

For the Platyhelminthes the literature says that there are mostly all Platyhelminthes parasites TODO: ref einfügen . But at the end we predicted 4.18% as free-living. The class Seriata is the reason for the most of free-livings in this phylum. These are partly free-living flatworms, so the prediction looks right.

TODO: Wiki:

*The Seriata are an order of turbellarian flatworms.[1][2]

They are found in both freshwater and marine environments, and also include a number of species found in damp terrestrial conditions. Most are free-living, but the group includes the genus Bdelloura, which lives comensally on the gills of horseshoe crabs. Seriatans are distinguished from other related groups by the presence of a folded pharynx and of a number of diverticula arising from the intestine. The intestine itself may be either simple or branched.[3]

With the Nematoda it looks very different. In the Nematoda its much worse. Here speaks the literature of TODO: x percent free-living, but we found only 2.63% of them. The problem at this point, however, is obvious: The parasites have been much more studied and thus we start with only 0.63% free-living species. Against such a shifted data situation, the algorithm is powerless. And yet the percentage has increased.

Kingdom	# nodes	original states		final states							
		FL	P	0 (FL)	0.25	0.33	0.4	0.5	0.67	0.75	1 (P)
none	84456	45	529	15035	243	25910	0	8764	6183	0	28140
Fungi	324105	577	2983	39088	0	0	0	5858	0	0	274803
Chloroplastida	460457	3519	77	454211	0	0	0	4688	0	0	1558
Metazoa	1670956	30758	22373	1485749	0	0	1313	29002	5102	1957	147833

Table 3.3: Kingdom (inkl internal nodes)

Phylum	# nodes	original states		final states					
		FL	P	0 (FL)	0.4	0.5	0.67	0.75	1 (P)
Chordata	122546	10451	18	122473	0	0	0	0	73
Nematoda	33564	21	3289	846	0	1133	0	0	31585
Platyhelminthes	27142	7	7086	1010	0	175	0	0	25957
Apicomplexa	2102	1	255	1	0	0	0	0	2101
Arthropoda	1319460	18912	11141	1207204	1313	25499	4852	1957	78635

Table 3.4: Phylum (inkl internal nodes)

Anmerkung: Das sind schonmal vier große Kontraste, wenn dann noch Zeit bleibt, die schwirigen... Arthropoden, Fungi, Pflanzen... (Emanuel)

Kingdom	# nodes	original states		final states							
		FL	P	0 (FL)	0.25	0.33	0.4	0.5	0.67	0.75	1 (P)
none	75446	45	529	13426	220	24082	0	7792	5302	0	24493
Fungi	31457	577	2983	38520	0	0	0	5723	0	0	266463
Chloroplastida	416478	3519	77	410795	0	0	0	4182	0	0	1501
Metazoa	1491012	30758	22373	1328135	0	0	930	25535	4423	1665	130324

Table 3.5: Kingdom (leaf nodes)

3.3.2 Origins and Losses

Sara B. Weinstein and Armand M. Kuris have been searching for origins of parasitism in Animalia [13]. They identified 223 parasitic origins: 223 in Metazoa $\supset$ 143 in Arthropoda $\supset$ 87 in Insecta.

This has led us to count the origins and losses of parasitism in our investigation as well.

We count only one origin / loss in a parent node with different children's nodes.

Here we have encountered a problem: The Castor algorithm gives us probabilities for states. That means there are also nodes with state like 0.3 or 0.5. So how do you count? Our solution was, to round these values. We have to say that we round 0.5 to 0.

Domain / Kingdom / Phylum / Class	# internal nodes	# leaf nodes	Rootnode state	without and with rounding # origins (FL -> P)	# losses (P -> FL)
Eukaryota	241974	2293463	1.0	415	363
			P	462	369
Metazoa	179944	1491012	0.5	294	123
				321	129
Fungi	9534	314571	0.5	80	222
				97	222
Chloroplastida	43486	412434	0.0	40	2
			FL	42	2
Arthropoda	120479	1198981	0.0	260	102
			FL	281	108
Apicomplexa	239	1863	1.0	0	1
			P	0	1
Nematoda	3437	30127	1.0	0	11
			P	2	11
Chordata	30761	91785	0.0	12	1
			FL	12	1
Platyhelminthes	4459	22683	1.0	0	5
			P	0	5
Insecta	91256	989572	0.0	234	77
			FL	245	77

Table 3.6: Origins and losses

In Table 3.6 we can see, that we found some more origins than Weinstein and on top of that some losses.

Lets have a look at the same phyla as in the section before: Chordata, Nematoda, Platyhelminthes and Apicomplexa.
Chordata are full of free-living species and so we see only a few origins of parasitism. The root and mostly all species are predicted as free-living.
In Apicomplexa and the Platyhelminthes are looking fine too. Our algorithm gives us only one loss of parasitism in Apicomplexa and five in the Platyhelminthes. They are both from the root over mostly all species predicted as parasites.
Nematoda is again full of problems. The rootnode is predicted as a parasite and so we have more losses of parasitism for the less information of free-living species in this phylum. The rest is parasitic

```
# possible tags: 0, 0.333, 0.4, 0.5, 0.667, 0.75, 1
# rounded to:    0  0      0    0    1      1     1
if node_state != father_state:
  if father_state == 0:
      origins += 1          # FL -> P
      new_found = True
  else:
      losses += 1           # P -> FL
      new_found = True
```

TODO: without rounding change else: to elif $father_state == 1$

3.3.3 Cross evaluation - leave 100 out

We ran the castor algorithm (TODO: 10000) 10 times with leaving 100 randomized free-living or parasitic species out of the input data to see how stable our result is.

run	distance			changed	lost
	all nodes	leaf nodes	internal nodes	tag	tag
0	4.0	4.0	0.0	4	48
1	4.0	4.0	0.0	2	45
2	85.3	79.3	6.0	3	51
3	1.0	1.0	0.0	1	47
4	1248.0	1230.5	17.5	0	54
5	169.0	168.0	1.0	3	52
6	1.0	1.0	0.0	1	48
7	1.0	1.0	0.0	1	53
8	1.0	1.0	0.0	1	54
9	1.0	1.0	0.0	1	52

Table 3.7: Example runs.
green: min value, red: max value

		min	max	mean	variance
distance	all	1.0	1248.0	151.53	151528.92
	leaf nodes	1.0	1230.5	149.08	147380.74
	internal nodes	0.0	17.5	2.45	31.47
changed tag		0.0	4.0	1.7	1.57
lost tag	all tags	45.0	54.0	50.40	10.04
	FL tags	16.0	32.0	23.10	18.77
	P tags	16.0	36.0	27.30	41.34

Table 3.8: Statistics to Cross validation

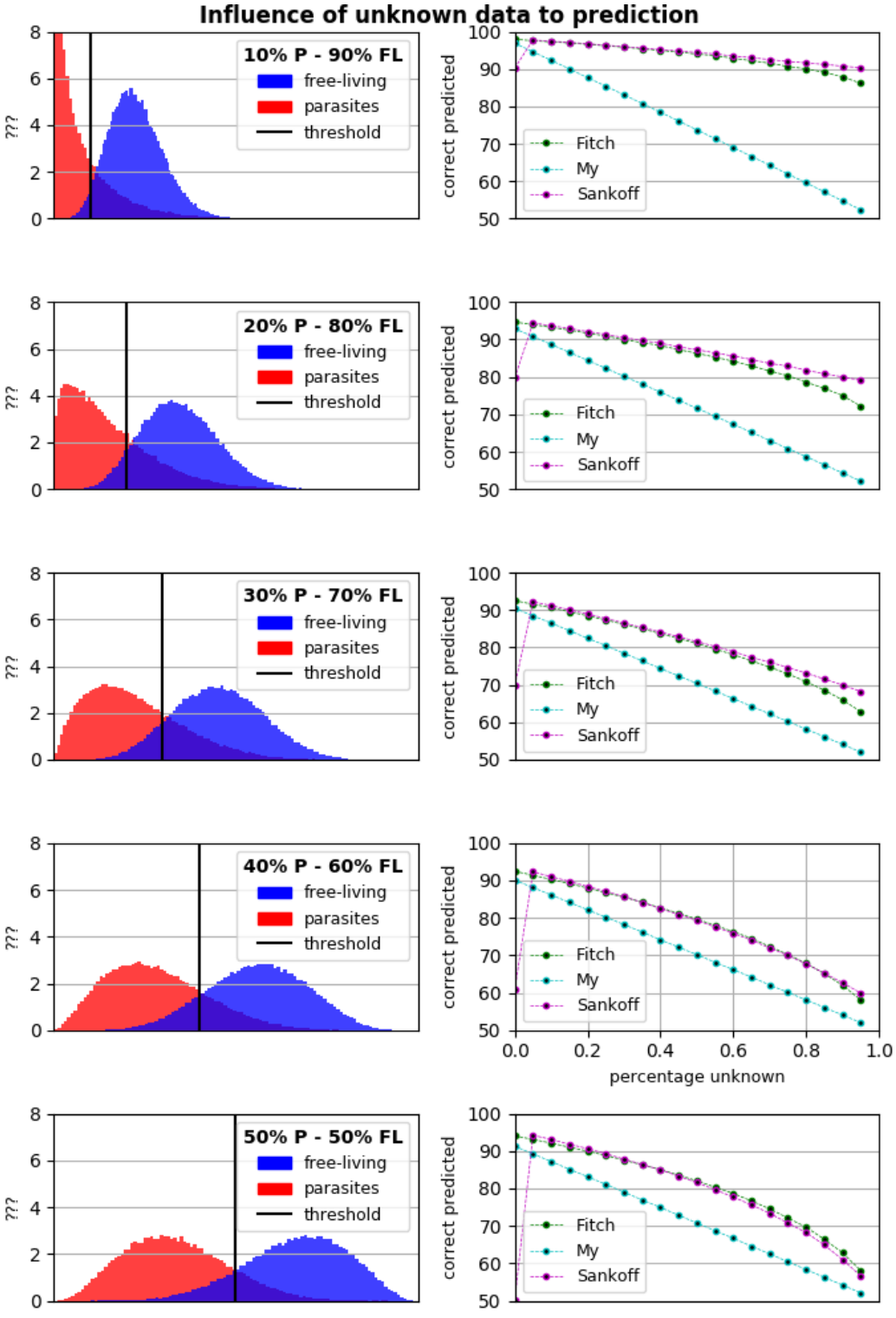
What could happen by removing a parasite or free-living of the list?

- It could be a specie, which don't exist in the tree leaf nodes. -> no effect
- It could be a specie, which exists in both lists. -> If it was a parasite, it is now free-living, because we prefer parasites. Otherwise we have no effect again. (1053 are possible)
- Normal case: We loose information, because its a specie in our tree and we change it to a leave node with no information.

3.4 Results of simulation

3.4.1 Influence of different parameters

Figure 3.1: Influence of unknown data to prediction



4 Discussion

Wie gut ist der randomisiert erstellte Baum?
Wie gut kommt unsere Simulation an die echte Datenlage heran.
Fehlerqoute der Daten an sich?
Wie gut ist unsere Datenlage? 3 mio knoten, 1.8 named species (leaf nodes), 200.000 leaf nodes mit Information.
Simulation von subtrees
Welche Teile des Baumes sind gut, an welchen muss noch viel geforscht werden.
Wieviele Origins haben wir gefunden, was bedeutet diese Zahl?

Parameter der Simulation:

- Wie ist die Verteilung der vergessenen internen Knoten? Zum Wurzelknoten hin mehr vergessen?
- Wie sehen die übergangswahrscheinlichkeiten aus von P→FL und andersherum?
- Verteilung Parasiten zu Freilebend zu keine Information

Selecting of the 'right' / best Distribution

Bibliography

[1] Fitch, W. M. Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. *Systematic Biology* **1971**, *20*, 406–416.

[2] Swofford, D. L.; Maddison, W. P. Reconstructing ancestral character states under Wagner parsimony. *Mathematical Biosciences* **1987**, *87*, 199 – 229.

[3] Hinchliff, C. E. et al. Synthesis of phylogeny and taxonomy into a comprehensive tree of life. *Proceedings of the National Academy of Sciences* **2015**, *112*, 12764–12769.

[4] ITIS, Integrated Taxonomic Information System. `https://www.itis.gov`.

[5] Bethesda (MD): National Library of Medicine (US), N. C. f. B. I. National Center for Biotechnology Information (NCBI). 1988; `https://www.ncbi.nlm.nih.gov/`.

[6] Horton, T. et al. World Register of Marine Species (WoRMS). http://www.marinespecies.org, 2018; `http://www.marinespecies.org`, Accessed: 2018-02-27.

[7] GBIF, Global Biodiversity Information Facility. `https://www.GBIF.org`.

[8] Diego Vázquez, R. N., Jeremy Goldberg Interaction Web Database (IWDB). 2003; `https://www.nceas.ucsb.edu/interactionweb/`.

[9] Webs on the Web (WOW): 3D visualization of ecological networks on the WWW for collaborative research and education. 2004; pp 5295 – 5295 – 9.

[10] Myers, P.; Espinosa, R.; Parr, C. S.; Jones, T.; Hammond, G. S.; Dewey, T. A. The Animal Diversity Web (online). 2018; `https://animaldiversity.org`.

[11] Cohen, J. E. c. Ecologists' Co-Operative Web Bank. Version 1.1. Machine-readable database of food webs. *New York: The Rockefeller University* **2010**,

[12] Poelen, J. H.; Simons, J. D.; Mungall, C. J. Global biotic interactions: An open infrastructure to share and analyze species-interaction datasets. *Ecological Informatics* **2014**, *24*, 148 – 159.

[13] Weinstein, S. B.; Kuris, A. M. Independent origins of parasitism in Animalia. *Biology Letters* **2016**, *12*.

5 Appendices

5.1 Distribution of data in the taxa

Kingdom (3)	Number of Nodes	Phylum (25)	Number of Nodes	max max height
Metazoa	1 465 207	Arthropoda	1 170 539	54
		Chordata	106 650	74
		Mollusca	80 022	22
		Platyhelminthes	27 141	9
		Nematoda	24 564	23
		Cnidaria	14 878	36
		Porifera	11 737	26
		Echinodermata	10 654	14
		Bryozoa	8 631	11
		Rotifera	3 093	7
		Nemertea	1 793	7
		Tardigrada	1 654	7
		Acanthocephala	1 596	6
		Brachiopoda	1 055	9
		Nematomorpha	633	7
		Chaetognatha	360	7
		Hemichordata	196	5
		Cycliophora	11	5
Fungi	254 871	Ascomycota	157 045	19
		Basidiomycota	92 931	18
		Microsporidia	1 949	6
		Glomeromycota	1 490	6
		Chytridiomycota	1 456	6
Chloroplastida	121 239	Streptophyta	120 731	49
		Chlorophyta	508	6

5.2 Taxa table

Taxa	Number of Nodes	Internal Nodes	Leaf Nodes	Terminal Leaf Nodes
domain	1	1		
kingdom	3	3		
subkingdom	2	2		
infrakingdom	5	5		
superphylum	2	2		
phylum	57	54	3	
subphylum	22	20	2	
infraphylum	2	2		
superclass	8	8		
class	209	175	34	
subclass	120	106	14	
infraclass	14	11	3	
superorder	40	37	3	
order	1014	772	242	
suborder	285	222	63	
infraorder	95	78	17	
parvorder	14	12	2	
superfamily	829	626	203	
family	8449	6143	2306	
subfamily	3090	2250	840	
supertribe	1	0	1	
tribe	1764	1285	479	
subtribe	435	359	77	
genus	164656	105452	59204	
subgenus	1266	869	397	
section	3	2	1	
subsection	1	1	0	
species group	365	288	77	
species subgroup	123	100	23	
species	2247251	7423	2239828	2228993
subspecies	22437	75	22362	22239
variety	1755	7	1748	1726
varietas	5970	19	5951	5909
forma	1181		1181	1181
no rank	954	719	235	7
no rank - terminal	37452	41	37452	37452
(no entry)	40099	40099		

extended leaf nodes (real leaf nodes)