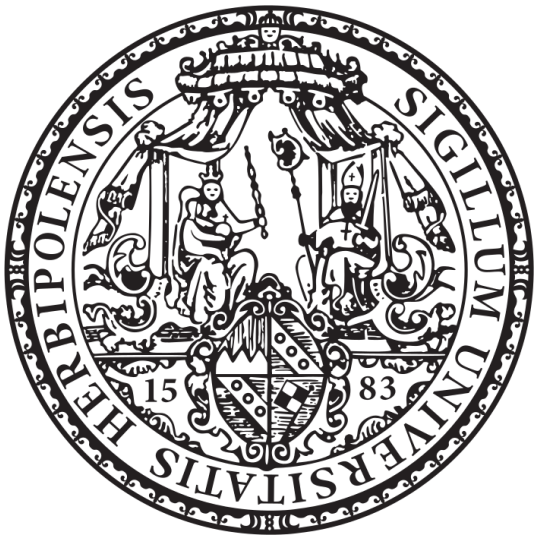
**Bachelorarbeit**

# Ein Algorithmus zur gemeinsamen Visualisierung von Arten- und Genbäumen

Moritz Niederer

| | |
|---|---|
| Abgabedatum: | 24. Februar 2021 |
| Betreuer: | Prof. Dr. Alexander Wolff |
| | Dr. Jonathan Klawitter |




Julius-Maximilians-Universität Würzburg
Lehrstuhl für Informatik I
Algorithmen und Komplexität

# Zusammenfassung

Eine wesentliche Aufgabe der Bioinformatik ist die Rekonstruktion der Verwandschaftsbeziehungen zwischen biologischen Taxa, wie zum Beispiel Arten oder Genen [BB03]. Diese Beziehungen werden üblicherweise mit sogenannten phylogenetischen Bäumen modelliert. Ein phylogenetische Baum ist ein gewurzelter Binärbaum, bei dem die Blätter mit einzelnen Taxa beschriftet sind und die inneren Knoten die Vorfahren dieser Taxa darstellen. Sind die Blätter mit verschiedenen Arten beschriftet, spricht man von einem Artenbaum. Betrachtet man die Gene von verschiedenen Arten, fallen oft Ähnlichkeiten auf, die auf eine gemeinsame Herkunft der Gene schließen lässt. Phylogenetische Bäume, die die Verwandschaftsbeziehungen von Genen modellieren, werden Genbäume genannt. Da die Gene einer Art die Gestalt und das Verhalten dieser Art bestimmen, stehen Arten und ihre Gene in unmittelbarem Zusammenhang. Vergleicht man jedoch einen Artenbaum, mit seinem Genbaum fallen oft Unterschiede zwischen den Bäumen auf, die unter anderem durch zufällige Genmutationen hervorgerufen werden. Um diese Unterschiede und den genauen Verlauf der Evolution zu erklären, werden die Genbaumblätter den Artenbaumblättern zugeordnet. Jedes Gen wird dabei der Art zugeordnet, aus der das Gen stammt. Der Baum über den Blättern wird dann durch Inferenzalgorithmen rekonstruiert und stellt ein komplexes Problem der Bioinformatik dar, was Reconciliation genannt wird. Die bei der Zuordnung entstehende Verschmelzung von Arten- und Genbaum wird Reconciliation Tree genannt. Diese Arbeit befasst sich mit der Visualisierung solcher Reconciliation Trees. In dieser Arbeit wird ein Zeichenstil definiert, bei dem die Artenbaumkanten durch die Flächen eines Rechtecks dargestellt werden, die Genbaumkanten rechtwinklig verlaufen und innere Genbaumknoten mittig über ihre Kinder platziert werden. Danach wird ein Algorithmus vorgestellt, der Reconciliation Trees im definierten Zeichenstil abbildet. Dabei versucht der Algorithmus eine Zeichnung mit möglichst wenigen Kreuzungen auszugeben. Anschließend stellen wir eine Verfahren vor, wie sich eine, von unserem Algorithmus entworfene Zeichnung, strecken lässt, sodass der Populationsverlauf der Arten durch die Breite der Artenbaumkanten repräsentiert wird. Zuletzt vergleichen wir unsere Zeichnungen mit den Zeichnungen von Douglas [Dou20] und bewerten unseren Algorithmus.

# Inhaltsverzeichnis

# 1 Einleitung

Die Phylogenetik ist eine Fachrichtung der Bioinformatik, die sich mit der Erforschung von Abstammungen beschäftigt [SS+03]. In diesem Fachgebiet verwendet man phylogenetische Bäume, um evolutionäre Prozesse zu modellieren und zu visualisieren. Im mathematischen Sinne ist ein phylogenetischer Baum ein gewurzelter, binärer Baum mit beschrifteten Blättern, der die Verwandschaftbeziehungen verschiedener biologischer Einheiten modelliert. Bei diesen Einheiten kann es sich zum Beispiel um verschiedene Arten oder Gene handeln. In diesem Zusammenhang werden solche Einheiten auch Taxa genannt. Jedes Blatt eines phylogenetischen Baumes repräsentiert ein Taxon. Sind die Blätter nach verschiedenen Arten beschriftet, spricht man von einem Artenbaum. Abbildung 1.1 zeigt den Artenbaum der Familie der Gebirgstaschenratten oder wissenschaftlich *Thomomys*, wobei jedes Blatt für eine bestimmte Art der Gebirgstaschenratte steht.

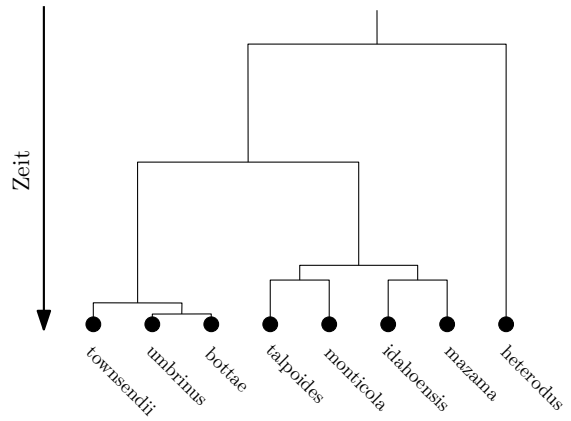



**Abb. 1.1:** Der Artenbaum der Gebirgstaschenratten. Jedes Blatt repräsentiert eine individuelle Art. Die Kanten visualisieren den evolutionären Verlauf der Arten.

Die inneren Knoten phylogenetischer Bäume stellen Verzweigungen der Taxa in ihrer evolutionären Geschichte dar. Die Kanten repräsentieren die Entwicklung hin zu den derzeitig lebenden Arten. Am Artenbaum der Taschenratte lässt sich zum Beispiel erkennen, dass die Art *Thomomys umbrinus* nahe verwandt mit der Art *Thomomys bottae* ist, da sie den selben direkten Vorfahren haben. Oft wird den Knoten auch ein Zeitpunkt zugeordnet, um den Verlauf der Evolution genau zu modellieren. Deshalb ist es für phylogenetische Bäume üblich den Verlauf der Zeit auf horizontaler Achse von links nach rechts oder auf vertikaler Achse von oben nach unten zu repräsentieren. Der Zeitwert jedes Knotens bestimmt dann dessen Position auf der Zeitaschse. Am Artenbaum

der Taschenratte lässt sich zum Beispiel ablesen, dass sich die Art *Thomomys heterodus* schon vor langer Zeit von den anderen Taschenratten abgespaltet hat. Möchte man auch den den Verlauf der Populationsgröße der einzelnen Arten in einem Artenbaum ablesen können, lässt sich dies durch die Kantenbreite ausdrücken. Dann entspricht die Breite einer Kante in einer gewissen Höhe der Populationsgröße der Art zu diesem Zeitpunkt. In der Phylogenetik werden nicht nur die Arten selbst, sondern auch die DNA der Arten untersucht. Denn die evolutionären Prozesse, welche die heutige Artenvielfalt geformt haben, spiegeln sich in den Genen der Arten wieder [DEMS14]. Das Erbgut einer Art verändert sich manchmal durch Zufall in einer Weise, dass ein Gen doppelt vorkommt. Diese doppelten Gene müssen die Eigenschaften der Arten vorerst nicht verändern, doch durch weitere Mutationen werden aus identischen Genen, unterschiedliche Gene, die dann auch jeweils unterschiedliche Eigenschaften der Gene mit sich bringen. Oder es kann vorkommen, dass eine Art ein Gen verliert, da es nicht von Nutzen war. Um die Entwicklung von Genen zu modellieren, werden Genbäume verwendet. Sie sind eine weitere Klasse der phylogenetischen Bäume.

Vergleicht man einen Genbaum und seinen zugehörigen Artenbaum kann es sein, dass diese kongruent zueinander sind, da die Gene sich in den Arten entwickelt haben. Durch Genduplikationen und Genverluste kann es aber auch vorkommen, dass die Bäume sehr unterschiedlich aussehen. Um die evolutionären Prozesse zu verstehen ist es wichtig Genbaum und Artenbaum gemeinsam zu betrachten und die eventuelle Inkongruenz zu erklären. Dies wird getan, indem man jedem Genbaumknoten eine Artenbaumkante zuordnet. Der Genbaum wird sozusagen in den Artenbaum platziert. Die entstehende Verschmelzung von Artenbaum und Genbaum wird *Reconcilation Tree* genannt. Abbildung 1.2 [HD09] zeigt einen Reconciliation Tree. Der Artenbaum ist so gezeichnet, dass die Breite der Kanten der Populationsgröße der Arten entspricht. Der Genbaum ist dann in den Artenbaum eingebettet.

Doch wie werden Arten- und Genbäume eigentlich erstellt? Uns stehen nur die Gene aktuell lebender Tiere zur Verfügung. Aus der Ähnlichkeit der Gene und anderen Faktoren, wie zum Beispiel die Populationsgröße der Arten oder der Wahrscheinlichkeit für eine Mutation, wird mit phylogenetischen Inferenzalgorithmen gefolgert, wie der Genbaum wohl aussieht. Ein Ansatz, welcher dieses Problem angeht, ist der *Bayesian multispecies coalescent Process*. Anhand von DNA-Sequenzen, dem zugehörigen Artenbaum, der Populationsgröße der Arten und weiteren Faktoren berechnet dieser stochastische Prozess die Genbäume und eine wahrscheinliche Einbettung des Genbaums in den Artenbaum [RY03].

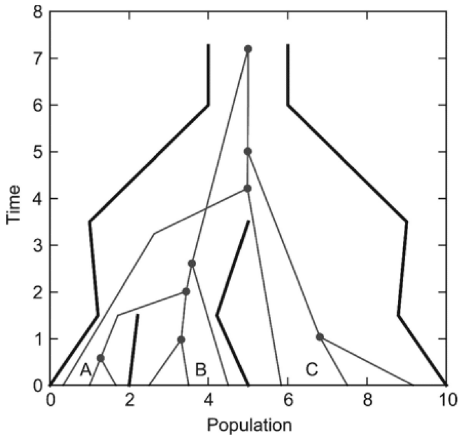



**Abb. 1.2:** Ein Reconciliation Tree. Die Breite der Artenbaumkanten entspricht der Population der jeweiligen Arten. Der Genbaum ist in den Artenbaum eingebettet. Diese Grafik ist aus einem Artikel von Heled und Drummond [HD09] entnommen.

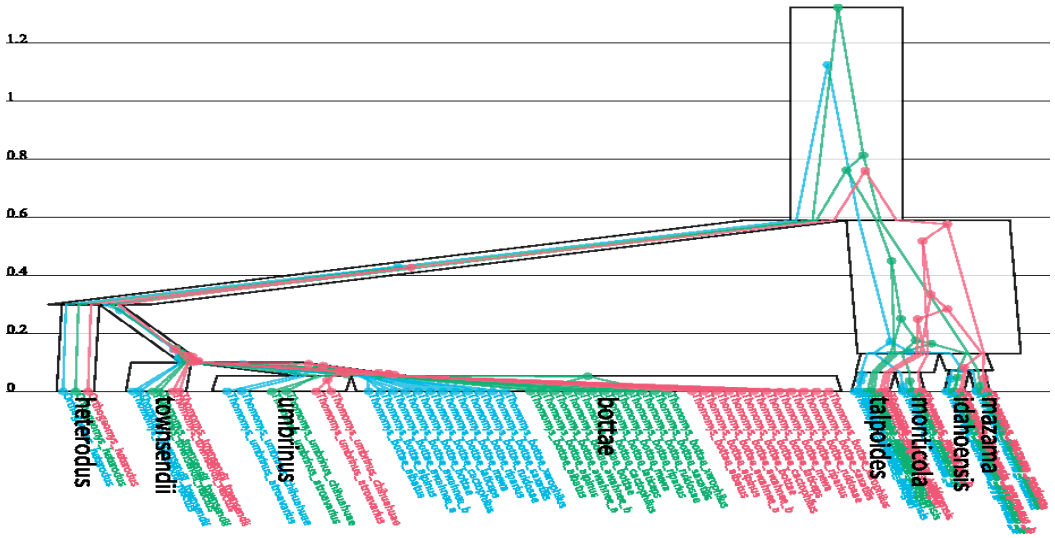



**Abb. 1.3:** Ein Reconciliation Tree erstellt mit Douglas' Algorithmus [Dou20]. Die Breite der Artenbaumkanten entspricht der Population der jeweiligen Art. Der Genbaum ist in den Artenbaum eingebettet.

## 1.1 Problemstellung

Über das Modellieren von Reconciliation Trees gibt es viel Literatur und bereits viele Methoden, die den Zusammenhang zwischen Arten- und Genbaum erarbeiten[RLGL14]. Doch nicht nur das Modellieren, sondern auch die Visualisierung von Reconciliation Trees ist eine wichtige Aufgabe der Phylogenetik [Dou20]. Speziell für den Multispecies

coalescent Process, eine von vielen Methoden der Reconciliation, gibt es leider nur wenige Methoden für die Visualisierung der Ergebnisse [Dou20]. Deshalb stellen wir in dieser Arbeit einen Algorithmus vor, der sich diesem Visualisierungsproblem annimmt. Genauer erstellt unser Algorithmus aus einem Artenbaum, einem Genbaum und einer Einbettung des Genbaums in den Artenbaum eine Zeichnung des modellierten Reconciliation Trees.

## 1.2 Verwandte Arbeiten

Der Artikel von Douglas [Dou20] beschäftigt sich mit einem sehr verwandten Problem. Im Artikel stellt er ein Programm zur Visualisierung des MSC-Models vor. Abbildung 1.3 zeigt einen Reconciliation Tree erstellt mit Douglas' Programm. Der Verlauf der Zeit wird durch die vertikale Achse repräsentiert und die Populationsgröße der Arten lassen sich an der Breite der Artenbaumkanten ablesen. Douglas ist jedoch selbst nicht zufrieden mit seinen Ergebnissen, was sich zum Beispiel an dem von ihm gewählten Namen für sein Programm - UglyTree (zu deutsch, hässlicher Baum) – zeigt. [Dou20]. Deshalb wollen wir den Visualisierungsprozess optimieren. Als Vorbild für unseren Zeichenstil nehmen wir eine Beispielgrafik von Heled und Drummond [HD09], siehe Abbildung 1.2. Vergleicht man diese beiden Abbildungen fallen auch zwei unterschiedliche Modelle für Populationsgröße auf. Bei Douglas hat jede Kante zwei Populationswerte. Den Populationswert bei Entstehung der Arten und einen Wert bei Aufspaltung der Arten. Diese beiden Werte sind durch das Modell nicht eingeschränkt und zwischen ihnen wird ein linearer Populationsverlauf angenommen. Dies wird *stückweise konstantes Populationsmodell* genannt. Bei der Abbildung von Heled und Drummond wird die Populationgröße jeder Art auch durch zwei Werte festgelegt. Diese sind jedoch nicht frei wählbar, sondern die Populationsgröße am Ende einer Kante, muss der Summe der Populationsgrößen der Kinderkanten am oberen Ende entsprechen. Abbildung 2.3 verdeutlicht diesen Zusammenhang. Dies wird *kontinuierliches, lineares Populationsmodell* genannt. Bei Douglas gibt es außerdem, im Gegensatz zu der Abbildung von Heled und Drummond, freie Flächen zwischen den Artenbaumkanten. In unserer Zeichnung sollen die Populationsgrößen dem kontinuierlichen, linearen Modell entsprechen. Außerdem sollen zwischen den Artenbaumkanten in unserer Zeichnung keine freien Flächen entstehen. Es gibt auch noch andere Zeichenstile zur Visualisierung von Reconciliation Trees. Zum Beispiel wird im Artikel von Chevenet, Doyon, Scornavacca, Jacox, Jousselin und Berry [CDS$^+$16] ein Programm vorgestellt, dass den Reconciliation Tree in sehr verschiedener Weise zeichnet. Dort wird nur der Genbaum gezeichnet und dann seitlich markiert, welche Gene zu welcher Art gehören, siehe Abbildung 1.4.

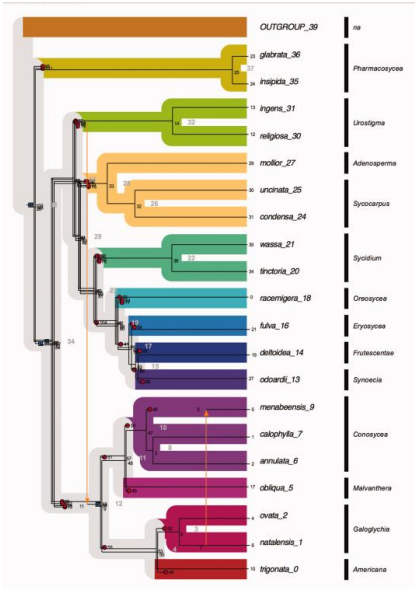



**Abb. 1.4:** Ein Reconciliation Tree erstellt mit dem SylvX-Editor.

# 2 Grundlagen

Es folgen wichtige Definitionen und Notation, die wir im Folgenden benötigen werden.

## 2.1 Graphen und Bäume.

Ein *Graph* ist ein abstraktes Konstrukt, welches Verbindungen zwischen Elementen beschreibt. Man spricht von gerichteten Graphen falls die Verbindungsrichtung von Bedeutung ist und von ungerichteten Graphen falls nicht. Abbildung 2.1 zeigt einen Graphen. Die Knoten werden als Punkte dargestellt und die Kanten werden durch Strecken zwischen den Punkten repräsentiert. Dies ist eine sehr übliche Darstellung von Graphen.

Formal ist ein *gerichteter Graph* ein Tupel $(V, E)$ aus zwei Mengen. Die nicht leere Knotenmenge $V$ beinhaltet die einzelnen Objekte für die der Graph die Verbindungen beschreibt. Die Kantenmenge $E \subseteq V \times V$ repräsentiert die Verbindungen der Objekte. Ist das Paar $(v, w)$ in der Kantenmenge $E$ enthalten, steht der Knoten $v$ mit dem Knoten $w$ in Verbindung. Die Gegenrichtung, dass $w$ mit $v$ verbunden ist, gilt aber nicht zwingend, sondern nur wenn das Paar $(w, v)$ in $E$ liegt. Wie auch in Abbildung 2.2, werden Kanten bei gerichteten Graphen oft durch Pfeile dargestellt . Die Kanten eines Knotens $v$ lassen sich in zwei Gruppen aufteilen. *Eingehende Kanten* des Knotens $v$ sind alle Kanten $(u, v)$, welche von einem beliebigen Knoten $w$ auf $v$ zeigen. *Ausgehenden Kanten* $(v, w)$ von $v$ nennt man alle Kanten, die von $v$ auf einen beliebigen Knoten $u$ zeigen. Der Knoten $v$ in Abbildung 2.2 hat zwei ausgehende Kanten und eine eingehende Kante.

Ein *ungerichteter Graph* ist auch ein Tupel $(V, E)$ mit Knotenmenge $V$ und Kantenmenge $E$. Die Kantenmenge bei ungerichteten Graphen ist jedoch keine Teilmenge des Kreuzproduktes $V \times V$, sondern ist Teilmenge aller zweielementigen Teilmengen von $V$. Also $E \subseteq \Big\{\{v, u\} \subseteq V\Big\}$. Da wir in der Kantenmenge $E$ Zweiermengen haben und keine Tupel, haben die Kanten keine Richtung. In Abbildung 2.1 sieht man einen ungerichteteten Graphen.

In dieser Arbeit sind speziell Bäume, eine Untergruppe der Graphen, von Bedeutung. Ein *Baum* ist ein zusammenhängender, ungerichteter Graph der keine Kreise enthält. Man spricht von einem *gewurzelten Baum* $B = (V, E)$, wenn $B$ ein gerichteter Baum ist und alle Kanten von einem Knoten wegzeigen. Dies ist die Wurzel des Baumes, welche demzufolge keine eingehenden Kanten hat. Ein Teilbaum ist ein Baum, dessen Wurzel ein Knoten eines anderen Baumes ist. Als *Blätter* werden alle Knoten ohne ausgehende Kanten bezeichnet und alle anderen Knoten nennt man *innere Knoten*. Mit $L(v)$ bezeichnen wir die Menge der Blätter des Teilbaumes mit Wurzel $v$. Innere Knoten haben genau eine eingehende Kante und mindestens eine ausgehende Kante. Der *Elternknoten* eines Knotens $v$ ist der einzige Knoten $u$, für den eine Kante $(u, v)$ existiert. Die *Kindknoten*

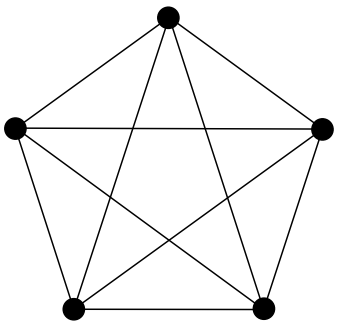


**Abb. 2.1:** Ein ungerichteter Graph.

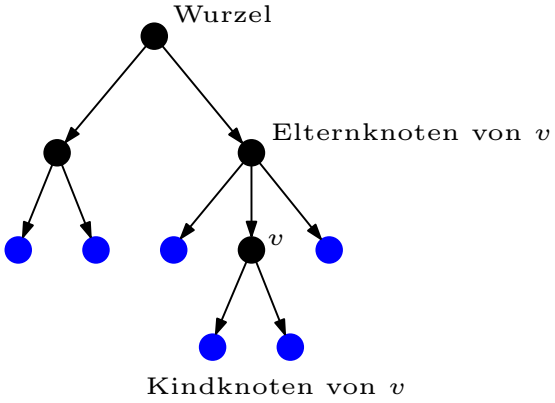



**Abb. 2.2:** Ein gewurzelzer Baum.

von $v$ sind die Knoten, für die $v$ der Elternknoten ist. Ein *Binärbaum* ist ein gerichteter Baum, bei dem die Wurzel und alle inneren Knoten genau zwei ausgehende Kanten haben. Abbildung 2.2 zeigt einen gewurzelten Baum mit blau gefärbten Blättern.

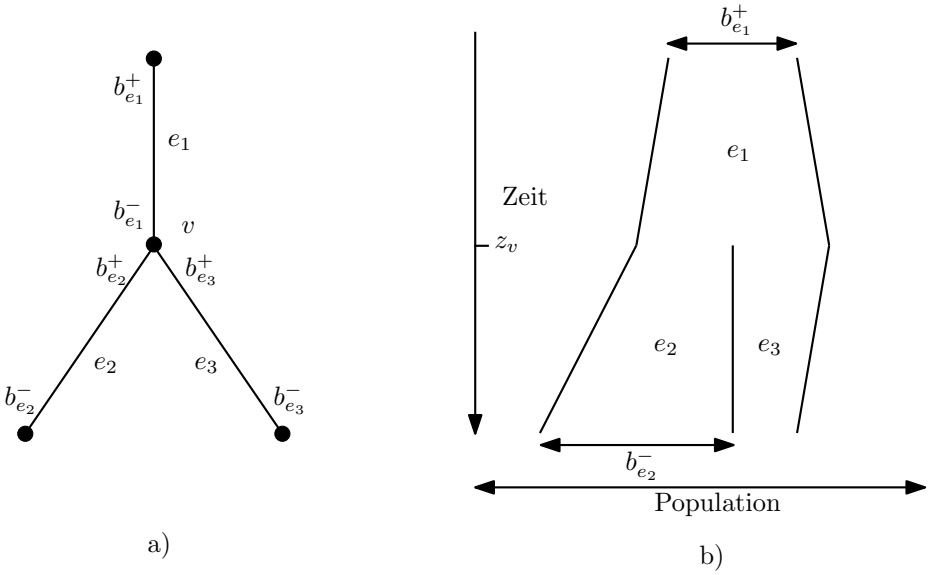



**Abb. 2.3:** Abbildung a) zeigt einen Ausschnitt des Artenbaums. Abbildung b) zeigt den selben Ausschnitt in einer Zeichnung, in der die Populationsgrößen der Arten durch die Breite der Kanten repräsentiert sind.

## 2.2 Phylogenetische Bäume und Reconciliation

Ein *gewurzelter phylogenetischer Baum* ist ein gewurzelter Binärbaum $B$, der die evolutionäre Geschichte verschiedener Taxa, wie zum Beispiel Arten oder Gene, darstellt. Die

Blätter eines solchen Baumes stehen für die Taxa und sind nach deren Namen beschriftet. Die Kanten repräsentieren die Entwicklung der Taxa. Jeder innere Knoten repräsentiert eine evolutionäre Verzweigung eines Taxons in zwei Taxa. Betrachten wir einen inneren Knoten $v$ mit eingehender Elternkante und ausgehenden Kindkanten, dann repräsentiert die Elternkante den direkten evolutionären Vorfahren der Kindkanten. Die Höhen der Knoten beschreiben den Verlauf der Zeit. Beginnend bei der Wurzel, dem Zeitpunkt an dem der Vorfahre aller betrachteten Einheiten sich entwickelt hat, bis zur heutigen Zeit. Deshalb hat jeder Knoten $v$ unserer Bäume einen Zeitwert $z_v$ zugewiesen.

Ein *Artenbaum* ist ein gewurzelter phylogenetischer Baum, der die evolutionäre Geschichte einer Menge von Arten modelliert [Nak13]. Da wir in unserer Zeichnung auch die Populationsgröße der einzelnen Arten mit einbeziehen wollen, hat jede Artenbaumkante $e = (u, v)$ zwei Werte, die ihre Population beschreiben. Am Anfang, also an $u$, gibt der Wert $b_e^+$ die Populationgröße an und $b_e^-$ repräsentiert die Population am Ende der Kante, also an $v$. Außerdem wollen wir das kontinuierlich, lineare Populationsmodell verwenden, bei dem die Population am Ende einer Kante, der Population der Summe der Kindkanten entsprechen. Wir fordern also für jede Kante $e_1$ und seine Kindkanten $e_2$ und $e_3$, dass $b_{e_1}^- = b_{e_2}^+ + b_{e_3}^+$ sein muss. Dies wird nochmal in Abbildung 2.3 veranschaulicht. Abbildung 2.3 a) zeigt einen Ausschnitt eines Artenbaums mit Kanten die mit Populationswerten beschriftet sind. In Abbildung 2.3 b) sehen wir den selben Ausschnitt des Artenbaums in einer Zeichnung des Artenbaums, bei der die Population der Arten durch die Kantenbreite repräsentiert wird. An der Breite lässt sich ablesen, dass wie gefordert $b_{e_1}^- = b_{e_2}^+ + b_{e_3}^+$ gilt.

Ein *Genbaum* ist ein gewurzelter phylogenetischer Baum, der die evolutionäre Geschichte verschiedener Gene modelliert [Nak13]. In dieser Arbeit verwenden wir den Ausdruck 'ein Genbaum und sein zugehöriger Artenbaum'. Damit ist der Artenbaum gemeint, der die Arten, in denen die Gene des Genbaums vorkommen, beschreibt.

Unter *Reconciliation* versteht man das Vergleichen von einem Artenbaum $A = (V_A, E_A)$ und einem Genbaum $G = (V_G, E_G)$ um die evolutionäre Geschichte der Taxa zu erarbeiten. Dabei werden die evolutionären Prozesse modelliert, indem man den Genbaum in den Artenbaum einbettet. Mathematisch gesehen ist diese Einbettung eine Funktion $f : V_G \to E_A$, die jedem Genbaumknoten eine Kante des Artenbaums, also eine Art, zuordnet.

## 2.3 Zeichenstil und Definition des Problems

Wir wollen den Reconciliation Tree so zeichnen, dass die Genbäume in den Artenbaum gezeichnet werden. An der vertikale Achse der Zeichnung soll der Verlauf der Zeit erkennbar sein. Deshalb ist die Höhe jedes Knotens $v$ eindeutig festgelegt durch seinen Zeitwert $z_v$. Der Artenbaum soll in einer Weise visualisiert werden, dass die Populationsgröße an der Breite der Kanten abgelesen werden kann. Dafür hat jede Artenbaumkante $e$ zwei Werte $b_e^+$ und $b_e^-$ die, wie in der Definition phylogenetischer Bäume beschrieben, die Population der Arten angeben. In Abbildung 2.3 ist dieses Prinzip veranschaulicht. Außerdem sollen die Artenbaumkanten kreuzungsfrei gezeichnet werden. Da die vertikale

Achse die Zeit repräsentiert, sollen die Genbaumkanten von einem Knoten zu seinen Kindern monoton fallend sein. Ein Beispiel für unseren angestrebten Zeichenstil sieht man in Abbildung 1.2. Wir definieren außerdem einen vereinfachten Zeichenstil, der eine einfache Visualisierung ermöglichen soll. Im einfachen Zeichenstil soll die Höhe der Knoten weiterhin durch die Zeitwerte festgelegt sein. Die Breite der Artenbaumkanten müssen jedoch nicht mehr den Populationsgrößen entsprechen. Stattdessen wird jede Artenbaumkante $e$ als Kontur eines Rechtecks gezeichnet, damit wird die Kante durch die Fläche des Rechtecks dargestellt. Wenn $n$ die Anzahl der in $e$ platzierten Genbaumblätter ist, dann zeichnen wir die Artenbaumkante $e$ mit einer Breite von $n+1$ Einheiten, sodass wir jedem Genbaumblatt einen ganzahligen x-Wert zuordnen können. Dann ist zwischen den Blättern und zu den Begrenzungen der Artenbaumkante immer eine Einheit frei. Die Breite der inneren Kanten entspricht der Summe der Breite ihrer Kindkanten. Daraus ergibt sich für den Artenbaum eine rechteckige Form. Die inneren Genbaumknoten werden immer mittig über ihre Kinder platziert. Damit bestimmt die Platzierung der Genbaumblätter die Position aller Genbaumknoten. Die Kanten der Genbäume sollen rechteckig gezeichnet werden. Dies bedeutet, dass eine Kante von einem Knoten zu seinem Kind erst in horizontaler Richtung gezeichnet wird. Bei der x-Koordinate des Kindes angekommen, macht die Kante dann einen rechtwinkligen Knick und verläuft in vertikaler Richtung bis zum Kindknoten. In Abbildung 2.4 sieht man einen Reconciliation Tree im vereinfachten Zeichenstil.

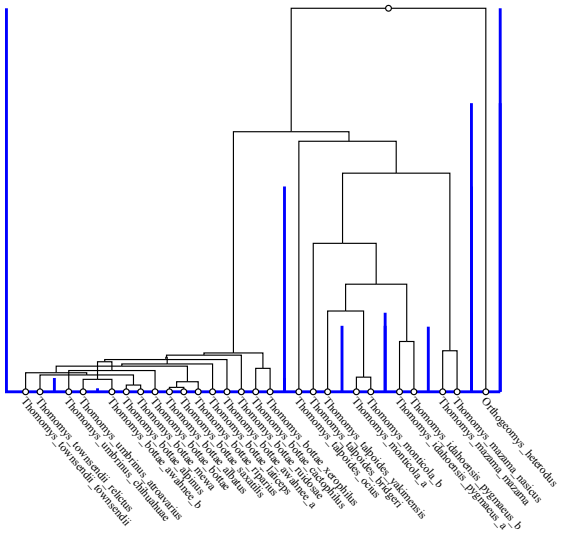



**Abb. 2.4:** Ein Reconciliation Tree im einfachen Zeichenstil.

Eine *geradlinige Zeichnung* eines Graphen $G$ mit Knotenmenge $V$ und Kantenmenge $E$ definieren wir als eine Abbildung $Z$ von der Knotenmenge auf Punkte in der Ebene.

$$Z\colon V \to \mathbb{R}^2 \text{ mit Streckenmenge } S_Z = \{\overline{Z(u)Z(v)} \text{ mit } uv \in E\} \tag{2.1}$$

Für jede Kante $uv \in E$ existiert eine Strecke in $S_Z$. Diese Strecke verbindet den Punkt $Z(u)$

mit dem Punkt $Z(v)$. Da die Zeichnung geradlinig ist, ist die Zeichnung durch die Bilder der Knoten eindeutig bestimmt.

Die Anzahl *Kreuzungen einer geradlinigen Zeichnung Z* definieren wir als die Anzahl der Paare von sich kreuzenden Kanten in der Zeichnung. Zwei Kanten $uv$ und $wx$ kreuzen sich, wenn die Strecke $\overline{Z(u)Z(v)}$ mit der Strecke $\overline{Z(w)Z(x)}$ einen Punkt gemeinsam hat und dieser gemeinsame Punkt nicht Ausgangspunkt beider Strecken ist.

Formal definieren wir das Problem wie folgt: Gegeben sei ein gewurzelter Artenbaum $T = \{V, E\}$ und eine beliebige Anzahl gewurzelter Genbäume $G$. Weiterhin sei eine Funktion $f$ gegeben, die jedem Genbaumknoten eine Art zuordnet. Aus diesen Eingaben soll eine Zeichnung im oben definierten vereinfachten Zeichenstil entstehen. Dabei wollen wir in die Anzahl der Kreuzungen in unserer Abbildung minimieren.

# 3 Eine Algorithmus zur Lösung des Problems

In diesem Kapitel stellen wir einen Algorithmus vor, der einen Reconciliation Tree im einfachen Zeichenstil zeichnet. Dazu bekommt der Algorithmus einen Artenbaum $A = (V_A, E_A)$, einen Genbaum $G = (V_G, E_G)$ und das Ergebnis der Reconciliation der beiden in Form einer Funktion $f : V_G \to E_A$ übergeben. Wir nehmen bei den übergebenen Bäumen an, dass alle Blätter den selben Zeitwert haben und deshalb auf einer Ebene liegen.

Die Anzahl der Kreuzungen in unserer Zeichnung hängt unter anderem von der Reihenfolge der Artenbaumblätter ab. Diese ist variabel, denn an jedem inneren Knoten $v$ des Artenbaums, lässt sich der Teilbaum des linken Kindes von $v$ mit dem Teilbaum des rechten Kindes von $v$ vertauschen, ohne dass Kreuzungen zwischen den Kanten entstehen. Falls zwei Genbaumblätter den selben Elternknoten haben, aber in unterschiedlichen Arten $b$ und $c$ liegen, entstehen tendenziell weniger Kreuzungen, wenn $b$ und $c$ in der Reihenfolge der Artenbaumblätter aufeinander folgen. In Abbildung 3.1 ist ein Beispiel für diesen Fall gezeigt. Hier ist zweimal der selbe Artenbaum mit unterschiedlicher Anordnung der Artenbaumkanten zu sehen. Mit der Ordnung $a, b, c, d$ spart man sich vier Kreuzungen im Vergleich zu Anordnung $b, a, d, c$. Um unser Problem zu vereinfachen, sehen wir die Anordnung der Artenbaumblätter als gegeben an.

Stattdessen widmen wir uns der Sortierung der Genbaumknoten innerhalb der Artenbaumkanten. Da die Höhe jedes Knotens $v$ bereits durch seinen Zeitwert $z_v$ festgelegt ist, können wir die Knoten lediglich auf der horizontalen Achse setzen. Mehr noch, da im einfachen Zeichenstil jeder innere Knoten mittig über seine Kinder platziert wird, können wir nur die Platzierung der Genbaumblätter bestimmen. Die Reihenfolge der Genbaumblätter innerhalb einer Art ist für die Minimierung der Kreuzungen daher von entscheidender Bedeutung. Zum Setzen der Knoten geht unser Algorithmus in zwei Schritten vor. Zuerst wird die Reihenfolge der Blattknoten berechnet. Erst danach setzen wir die Koordinaten der Knoten mithilfe der vorher berechneten Reihenfolge. Da wir in unserem einfachen Zeichenstil rechtwinklige Kanten haben, ist mit der Positionierung der Genbaumknoten auch der Verlauf der Kanten beschrieben. Möchte man an der Breite der Artenbaumkanten die Populationsgröße ablesen können, kann die Zeichnung nachträglich gestreckt werden. Dazu wird für jede Artenbaumkante eine Funktion erstellt, die die Streckung der Kanten selbst und die Streckung der in ihr liegenden Knoten beschreibt.

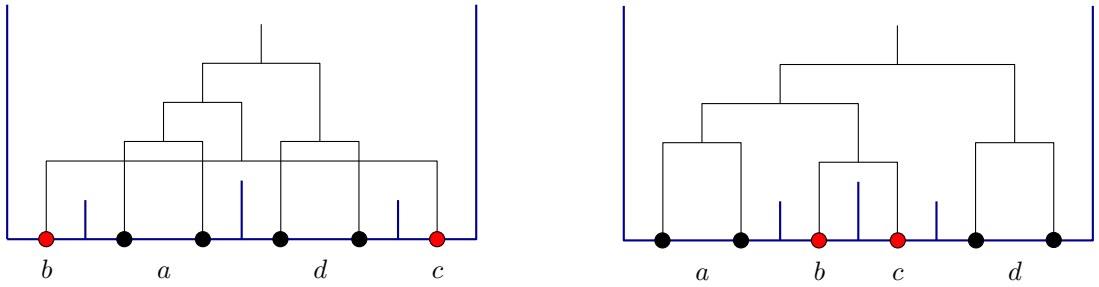



**Abb. 3.1:** Zweimal der selbe Reconciliation Tree. Durch die Umordnung der Artenbaumblätter spart man sich mehrere, sonst unvermeidbare, Kreuzungen.

## 3.1 Überblick zur Vorgehensweise

Die Hauptaufgabe des Algorithmus ist die Sortierung der Genbaumblätter innerhalb der Artenblätter. Dazu betrachten wir iterativ die inneren Knoten des Genbaums der Höhe nach. In jedem Schritt wird der am niedrigsten liegende, noch nicht behandelte, innere Knoten betrachtet und die Blätter unter ihm eingeordnet. Die Sortierung in den Artenbaumblättern wird also nicht auf einmal berechnet, sondern in jedem Schritt wird immer nur ein Teil der Knoten platziert. Um die Sortierung zu speichern haben wir für jedes Artenblatt eine Liste für die linke Seite der Kante und eine Liste für die rechte Seite der Kante. In diese Listen werden die Genbaumknoten im Laufe des Algorithmus eingefügt. Für jeden inneren Knoten $v$ vergleichen wir die Teilbäume seiner Kindknoten $l$ und $r$ miteinander. Falls diese Teilbäume in verschiedenen Arten liegen, fügen wir die Blätter der Teilbäume von $l$ und $r$ in die Listen der Artenbaumkanten ein, sodass die Blätter möglichst nahe beieinander liegen. Damit erreichen wir, dass die Kanten von $v$ zu seinen Kindern möglichst kurz ausfallen und wenige Knoten im Inneren des Teilbaums liegen. Da wir in jedem Sortierungsschritt den am tiefsten liegenden Knoten betrachten, stellen wir sicher, dass nachfolgende Knoten immer um bereits platzierte Knoten angeordnet werden und der aktuell betrachtete Teilbaum über und um bereits platzierte Knoten gezeichnet werden kann. Am Ende des Algorithmus werden die beiden Listen jedes Artenblatts zusammengefügt um die Sortierung der Genbaumblätter zu erhalten. In den folgenden Abschnitten werden alle Teile des Algorithmus im Detail beschrieben.

## 3.2 Genaue Beschreibung des Algorithmus

In diesem Abschnitt erklären wir jeden Arbeitsschritt zum Zeichnen des Reconciliation Trees im Detail.

### DrawTrees

Die in Abschnitt 3.1 beschriebene grobe Vorgehensweise wird in Algorithmus 1, namens drawTrees, umgesetzt. drawTrees bekommt einen Artenbaum, einen Genbaum und ei-

ne Funktion, die die Genbaumknoten ihrer Art zuordnet, übergeben und ruft Methoden auf, um einen Reconciliation Tree zu zeichnen. Dazu wird zuerst der Algorithmus `findNodeOrder` (siehe Algorithmus 2) aufgerufen. Hier wird die Reihenfolge der Blattknoten innerhalb der einzelnen Kanten des Artenbaums festgelegt. Dies ist der Hauptteil des Algorithmus, da durch die Anordnung der Blätter, die Anzahl der Kreuzungen entschieden wird. Anschließend berechnet Algorithmus 3, namens `calculateXRange`, wie breit jede Kante des Artenbaumes sein muss, sodass alle Knoten hineinpassen. Das Verfahren `placeNodes`, siehe Algorithmus 4, weist nun jedem Genbaumknoten seinen x-Wert zu. Mit der Platzierung der Knoten ist unsere Zeichnung eindeutig definiert und unser Algorithmus terminiert.

**Algorithmus 1:** drawTrees(Speciestree $G_S = (V_S, E_S)$, Genetree $G_G = (V_G, E_G)$, Mapping $f : V_G \to E_S$)

**Eingabe:** Artenbaum $G_A = (V_A, E_A)$, Genbaum $G_G = (V_G, E_G)$ Mapping $M$ : NodeG $\to$ EdgeS )
**Ausgabe:** Zeichnung $Z$ der Genbäume im einfachen Zeichenstil

```
1 findNodeOrder(G_A, G_G, M)
2 calculateXRange(G_A.root)
3 placeNodes()
```

### FindNodeOrder

Um die folgende Beschreibung zu erleichtern bezeichnen wir mit $L(v)$ die Blätter des Teilbaumes mit Wurzel $v$. Der Algorithmus `findNodeOrder` bestimmt die Reihenfolge der Genbaumblätter innerhalb der einzelnen Artenbaumkanten. Dazu hat jede Artenbaumkante $e$ jeweils zwei Listen, welche die Sortierung der Blätter speichern. Die Liste $l_e$ repräsentiert die linke Seite der Kante $e$ und die Liste $r_e$ steht für die rechte Seite von $e$. Hängt man $r_e$ an die Liste $l_e$ beschreibt dies die vollständige Sortierung der Blätter. Die Reihenfolge der Blätter in den Listen von Anfang bis Ende, beschreibt die Sortierung der Blätter von links nach rechts innerhalb der Artenbaumkante. Das heißt, der erste Knoten in Liste $l_e$ ist am weitesten links und der letzte Knoten in $r_e$ ist am weitesten rechts. Wenn wir einen neuen Knoten platzieren, wollen wir, dass der neue Knoten immer zwischen den bereits sortierten Knoten eingefügt wird. Deshalb fügen wir in die Liste $l_e$ immer hinten ein und in die Liste $r_e$ immer vorne ein.

Unser Algorithmus `findNodeOrder` geht iterativ vor, um die Genbaumblätter nacheinander zu platzieren. Um zu entscheiden welche Blätter in welche Listen eingefügt werden sollen, betrachten wir die Eltern und Vorfahren der Blätter an und vergleichen diese nacheinander. Genauer fügen wir zunächst alle Knoten, die Eltern von Blättern sind, in eine Prioritätsschlange, die immer den Knoten mit minimalen $y$-Wert extrahiert, ein. In jedem Schritt des Algorithmus entnehmen wir den am niedrigsten liegende Knoten $v$ (also mit niedrigstem $y$-Wert) aus der Prioritätsschlange und wollen nun die Blätter im Teilbaum unter $v$ in Listen einfügen. Dafür unterscheiden wir drei Fälle:

Der erste Fall trifft ein, falls alle Blätter $L(v)$ in der selben Art $a$ liegen. Wie in Abbildung 3.2 a) zu sehen, ist zu diesem Zeitpunkt unklar, ob die Blätter besser in die

linke Liste oder in die rechte Liste von $a$ eingefügt werden sollten. Also verschieben wir die Einordnung der Blätter auf ein späteren Zeitpunkt, indem wir den Elternknoten $w$ von $v$ in die Prioritätsschlange einfügen.

Für den zweiten und dritten Fall, nehmen wir nun an, dass die Blätter $L(v)$ in verschiedenen Arten liegen. Seien $v.l$ und $v.r$ die Kinder von $v$.

Falls die Blätter $L(v.l)$ in einer einzigen Spezies liegen und auch die Blätter $L(v.r)$ in einer einzigen Spezies liegen, wie in Abbildung 3.2 b) gezeigt, dann können wir die Blätter platzieren. Unser Ziel ist es, die Blätter der Teilbäume möglichst nah beieinander zu platzieren. Also fügen wir die Blätter in der Art $a$, welche weiter links liegt, in die rechte Liste $r_a$ ein und die Blätter in der Art $b$, welche weiter rechts liegt, fügen wir in die linke Liste $l_b$ ein.

Der letzte Fall tritt ein, wenn die Blätter eines der Teilbäume in mehr als einer Art liegen. Dies ist in Abbildung 3.2 c) veranschaulicht. Wie in der Abbildung gezeigt, sei $v.l$ die Wurzel des Teilbaums, dessen Blätter $L(v.l)$ in mehreren Arten liegen. Außerdem seien die Blätter $L(v.r)$ alle in der Art $c$. Da $v.l$ tiefer als sein Elternknoten $v$ liegt, wurde $v.l$ bereits vom Algorithmus behandelt. Da die Blätter $L(v.l)$ in verschiedenen Arten liegen, wurden sie bereits zu einem früheren Zeitpunkt platziert. Wir möchten, dass die Blätter $L(v.r)$ möglichst nah bei den Blättern $L(v.l)$ liegen.

Wenn mindestens ein Blatt aus $L(v.l)$ in einer Art $d$ liegt, wobei $d$ links von $c$ liegt, dann werden die Blätter $L(v.r)$ in die linke Liste $l_c$ eingefügt. Ansonsten werden sie in die rechte Liste $r_c$ eingefügt. Algorithmus 2 zeigt das Verfahren in Pseudocode. Beim Einfügen der Blätter eines Teilbaums muss außerdem darauf geachtet werden, dass der Teilbaum planar gezeichnet werden kann. Indem man beim Einfügen der Sortierung der Inorder-Traversierung folgt, ist dies immer garantiert.

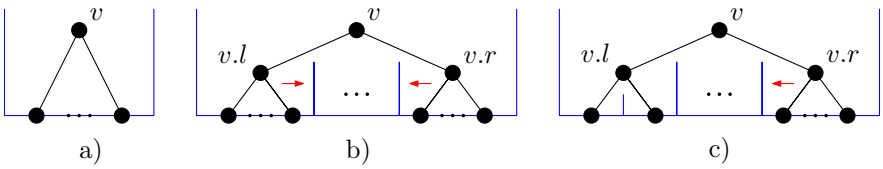



**Abb. 3.2:** Beim Extrahieren eines Knotens $v$ aus der Prioritätsschlange können die hier zu sehenden Fälle auftreten.

**Algorithmus 2:** findNodeOrder(Speciestree $G_S = (V_S, E_S)$, Genetree $G_G = (V_G, E_G)$, Mapping $f : V_G \to E_S$)

```
1  PriorityQueue Q = {v|v.l ∨ v.r is leaf }
2  while Q not empty do
3      v = Q.extractMinY
4      if L(v) are mapped to the same species then
5          if v.parent ∉ Q then
6              Q.insert(v.parent)
7      else if ∀u ∈ L(v.l) : f(u) = a ∧ ∀v ∈ L(v.r) : f(v) = b then
8          insert leaves as close as possible
9      else
10         insert only one subtree
11 foreach e ∈ E_S do
12     append right list to left list
```

### calculateXRange

Dieser Algorithmus bestimmt die Breite und Position jeder Kante des Artenbaums. Der Baum ist so aufgebaut, dass in jedem Knoten seine Kinder und die Kanten zu den Kindern gespeichert sind. Der Algorithmus `calculateXRange` setzt für jede Kante die Werte leftRange und rightRange, die für die $x$-Koordinaten der Seitenwände stehen. Um immer die richtige Koordinate zuzuweisen, wird eine statischen Variable $x$ verwendet, die für die Blattkanten immer die nächste freie $x$-Koordinate speichert. Falls eine Blattkante $e$ durch den Algorithmus positioniert wird, kann für die linke Wand von $e$ direkt der Wert von $x$ übernommen werden. Dann wird $x$ so erhöht, dass alle Genbaumknoten in $e$ genug Platz haben. Damit hat $x$ den Wert von der rechten Wand von $e$, was gleichzeitig die $x$-Koordinate für die linke Wand der folgenden Blattkante ist. Der Algorithmus geht rekursiv vor um alle Werte zu bestimmen. Für den übergebenen Knoten *root* wird überprüft, ob sein linkes Kind ein Blatt ist. Falls dies zutrifft, wird wie oben beschrieben mit Hilfe der Variable $x$ der Kante *root*.leftEdge eine Position zugewiesen. Andernfalls, also falls das linke Kind kein Blatt ist, ruft sich der Algorithmus selbst auf. Nach dem Aufruf sind die $x$-Koordinaten für die Kanten zu den Kinder von *root*.left bestimmt und können kopiert werden. Die selbe Prozedur wird anschließend für die rechte Kante *root*.rightEdge durchgeführt. Wenn CalculateValue mit der Wurzel des Artenbaums aufgerufen wird, wird so jeder Kante des Artenbaums ihre Position und Breite zugewiesen. Die Reihenfolge der Kanten wird dabei nicht verändert, sondern wird als gegeben angesehen. In Algorithmus 3 ist das Verfahren in Pseudocode zu sehen. Mit der Notation

*root.l* und *root.r* sind die Kinder des Knotens *root* gemeint.

**Algorithmus 3:** calculateXRange(NodeS *root*)

```
1  static x = 1
2  if root.l is leaf then
3      root.leftEdge.leftRange = x
4      x = x + 1+ number of leaves in root.leftedge
5      root.leftEdge.rightRange = x
6  else
7      calculateXRange(root.left)
8      copy ranges for left edge from the edges of left child
9  if root.r is leaf then
10     root.rightEdge.leftRange = x
11     x = x + 1+ number of leaves in root.rightEdge
12     root. rightEdge.rightRange = x
13 else
14     calculateXRange(root.r)
15     copy ranges for right edge from the edges of right child
```

## PlaceNodes

Nachdem die Reihenfolge der Genbaumblätter und auch die Position der Artenbaumkanten bestimmt sind, kann die Methode placeNodes, siehe Algorithmus 4, nun jedem Genbaumknoten einen konkreten $x$-Wert zuweisen. Dabei geht der Algorithmus iterativ vor, indem er die Knoten der Höhe nach platziert. In jedem Schritt wird der am niedrigsten liegende Knoten $v$ aus einer Schlange herausgenommen und platziert. Jede Genbaumkante $e$ speichert die Reihenfolge der in ihr liegenden Knoten in einer Liste $Q_e$. Außerdem hat jede Artenbaumkante die $x$-Koordinate der linken Wand der Kante in der Variable leftRange gespeichert. Falls $v$ ein Blatt ist, wird durch seine Position in der Liste $Q_e$ und dem Wert leftRange die $x$-Koordinate des Blattes ermittelt. Falls der Knoten $v$ kein Blatt ist, wird $v$ in die Mitte der Kinder platziert. Dann muss noch überprüft werden, ob bei der Platzierung genau in der Mitte keine überlappenden Kanten entstehen. Was in diesem Fall getan wird und wie es zu überlappenden Kanten kommen

kann, wird im nächsten Abschnitt beschrieben.

**Algorithmus 4:** placeNodes(Mapping $f$ : NodeG $\rightarrow$ EdgeS)

```
1  Queue Q = Queue with all genetree nodes sorted by height ascending
2  while Q not empty do
3      v = Q.Dequeue
4      if v is leaf then
5          s = f(v)
6          v.x= s.leftRange + Q_s.indexOf(v)+1
7      else
8          v.x = (v.left.x+v.right.x)/2
9          if edges overlap then
10             find another spot for v
```

## Das Vermeiden von überlappenden Kanten

Beim Platzieren der inneren Knoten kann es passieren, dass sich Kanten überschneiden und dadurch die Grafik schwer lesbar wird und man die Zusammenhänge nicht klar erkennen kann. Dies tritt auf, wenn zwei Knoten genau übereinander oder nebeneinander platziert werden. In Abbildung 3.3 a) sind die Knoten $u$ und $v$ direkt übereinander platziert. Dies führt zu einer nicht vollständig lesbaren Grafik führt. Indem man sich beim Platzieren eines Knotens die vertikale Strecke ausgehend vom eingefügten Knoten bis zur Höhe seines Elternknoten merkt, lassen sich nachfolgende Knoten so platzieren, dass Knoten nicht in diesem Bereich platziert werden. Falls die Platzierung eines Knotens in einen gesperrten Bereich fällt, wird er ein wenig nach links oder rechts geschoben. Auch nahe beieinander liegende Knoten auf der selben Höhe können zu Problemen führen. Ein Beispiel ist in Abbildung 3.3 b) gezeigt. Auf den ersten Blick sieht es so aus als wären die Knoten $u$ und $x$ Geschwister. Tatsächlich sind hier die Geschwisterpaare $(u, w)$ und $(v, x)$ abgebildet. Durch die überlappenden Kanten ensteht eine unleserliche Zeichnung. Deshalb muss bei Einfügen eines Knotens darauf geachtet werden, dass sich nicht bereits nahe liegende Knoten auf der selben Höhe befinden. Falls dies der Fall ist, muss der einzufügende Knoten ein kleines Stück weiter oben platziert werden. Da wir beim Einfügen der Knoten von unten beginnen, ist nach oben hin auf jeden Fall Platz. Durch das vertikale Verschieben eines Knoten $v$ entspricht die Höhe des Knotens leider nicht mehr exakt seinem Zeitwert $z_v$. Da die Änderungen nur sehr klein sind, fällt diese Verschiebung aber nicht ins Gewicht.

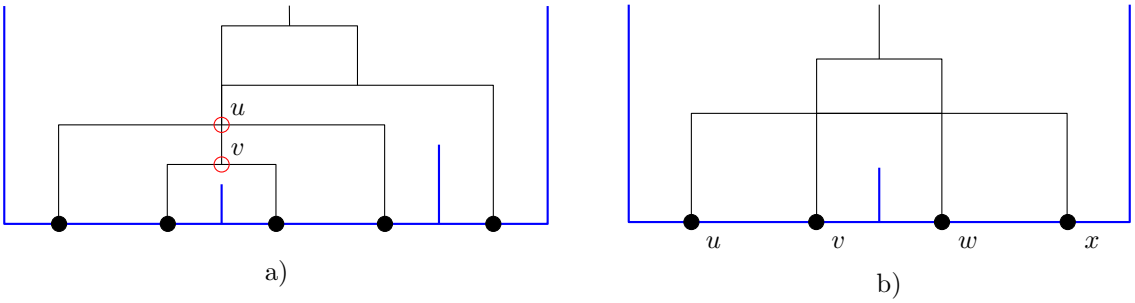



**Abb. 3.3:** Zwei Beispiele, wie durch überlappende Kanten eine schlecht leserliche Zeichnung entsteht.

### Zeichnen des Graphen in rechteckiger Form

Das Erstellen der Zeichnung im einfachen Zeichenstil ist nach Platzierung der Knoten keine Hürde mehr. Die Artenbaumkanten werden wie im einfachen Zeichenstil beschrieben als Fläche eines Rechtecks dargestellt. Die Höhe und Länge der Kanten ist definiert durch die Zeitwerte der Knoten. Die Breite der Kanten wurde durch calculateXRange, Zeile 12, definiert. Da unser Algorithmus eine Zeichnung mit rechtwinkligen Genbaumkanten erstellt, ist die Positionierung der Genbaumknoten durch den Verlauf der Kanten bestimmt. Indem wir jede Genbaumkante an ihrem Knick unterteilen und einen Hilfsknoten einfügen, werden die Genkanten zu geradlinigen Kanten. Damit haben wir eine geradlinige Zeichnung definiert, allein durch die Platzierung der Knoten.

## 3.3 Steckung der Zeichnung

Wenn die Breite der Artenbaumkanten der Population der Arten entsprechen soll, lässt sich die Zeichnung im einfachen Zeichenstil nach ihrer Erstellung noch strecken. Dazu werden die Genbaumkanten und die Seiten der Artenbaumkanten durch zusätzliche Knoten untergliedert. Auf der Höhe jedes Artenknotens müssen alle Kanten durch einen Hilfsknoten unterteilt werden. Außerdem müssen die Genbaumkanten, wie oben beschrieben, an ihrem Knickpunkt durch einen Knoten untergliedert sein. Dann muss für jeden Knoten sein neuer gestreckter $x$-Wert berechnet werden. Dazu wird jeder Artenbaumkante $e$ mit Populationswerten $b_e^-$ und $b_e^+$ und den y-Werten $y_e^+$ und $y_e^-$ eine Funktion zugeordnet, die zur Streckung der Kante selbst und der in ihr liegenden Knoten verwendet wird. Für die Berechnung der Kante benötigen wir außerdem die Breite $b_e$ der Kante $e$ in der ungestreckten Zeichnung und die Sreckfunktion der Kante $l$, die links von $e$ liegt. $r_l$ bezeichnet die $x$-Koordinate der rechte Wand von $l$ in der ungestreckten Zeichnung. Der Wert $\frac{y-y_e^+}{y_e^- - y_e^+} \in [0,1]$ drückt aus, in welcher Höhe sich ein Knoten in Bezug auf die Artenbaumkante befindet und bestimmt dadurch zu welchen Teilen der untere Populationswert der Kante berücksichtigt werden muss. Da dieser Wert mehrfach in unser Gleichung vorkommt, kürzen wir ihn mit $p$ ab. Unsere Gleichung für die Kante $e$ lautet:

$$f_e(x,y) = f_l\left(\frac{r_l}{b_l}, y\right) + \frac{p \cdot b_e^- + \left(1-p\right) \cdot b_e^+}{b} \cdot x$$

Der Teil $f_l\left(\frac{r_l}{b_l}, y\right)$ berechnet die Position der linken Seite der Artenbaumkante $e$. Möchte man die Funktion einer Kante weit rechts erstellen, kommt es durch diesen Teil der Funktion zu einer Reihe in sich verschachtelten Funktionen. Indem man sich bereits berechnete Funktionen speichert und die Funktionen dynamisch berechnet, spart man sich lange Berechnungen. Wie schon erwähnt bestimmt $p = \frac{y-y_e^+}{y_e^- - y_e^+}$ zu welchen Teilen die beiden Populationswerte der Kante $e$ mit einberechnet werden müssen. Wenn $p$ für einen Knoten $v$ berechnet wird, der am Anfang der Kante $e$ platziert ist, dann ist der $y$-Wert des Knotens gleich $y_e^+$ und damit $\frac{y-y_e^+}{y_e^- - y_e^+} = 0$. Also spielt für die Platzierung des Knotens der untere Populationswert $b_e^-$ keine Rolle und der obere Populationswert $b_e^+$ der Kante bestimmt allein die Platzierung des Knotens. Anders herum wenn $v$ am Ende der Kante $e$ liegt, dann ist $\frac{y-y_e^+}{y_e^- - y_e^+} = 1$ und allein der obere Populationswert $b_e^+$ bestimmt den Streckungfaktor des Knotens. Hat man für jede Kante ihre Funktion erstellt, können nun für jeden Knoten und die Seiten der Artenbaumkanten die neuen Werte berechnet werden und man erhält eine Zeichnung, in der die Populationsgröße der Arten der Breite der Kanten entspricht.

# 4 Bewertung des Algorithmus

Unser Algorithmus liefert keine Zeichnung mit einer minimalen Anzahl an Kreuzungen. Dies erkennt man im Abbildung 4.1 gezeigten Gegenbeispiel. Da die Blätter $u, v$ und $w$ in der selben Art liegen trifft unser Algorithmus keine Entscheidung darüber, in welcher Reihenfolge die Blätter gesetzt sind. Die Reihenfolge spielt aber eine Rolle bei der Anzahl der Kreuzungen in der Zeichnung. In Abbildung 4.1 a) schneidet die Kante zum Knoten $w$ zwei andere Kanten. In Abbildung 4.1 b) schneidet die selbe Kante nur eine Kante. Diese zusätzliche Kreuzung kann auch vermieden werden, indem man den Elternknoten von $t$ weiter rechts platziert. Die Platzierung der inneren Knoten in der Mitte über ihren Kindern ist also nicht immer optimal. Die Reihenfolge der Blätter der Artenbaumkanten verändert unser Algorithmus nicht. Wie in Abbildung 3.1 veranschaulicht, ist diese Reihenfolge aber entscheidend für die Anzahl der Kreuzungen zwischen den Genbaumkanten. Vor Anwendung unseres Algorithmus sollte man also eine geeignete Reihenfolge der Artenbaumblätter wählen.

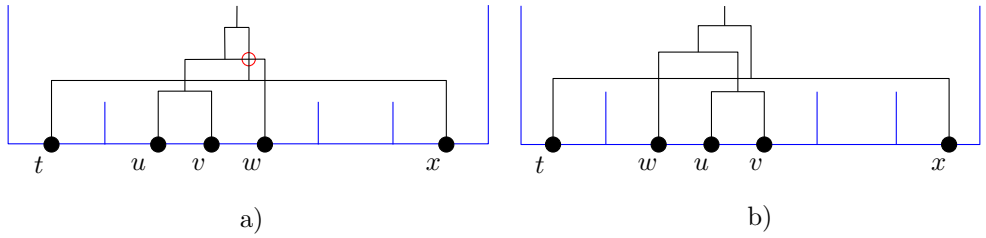



**Abb. 4.1:** Wenn die Genblätter eines Teilbaums in der selben Art liegen, dann ändert unser Algorithmus die Reihenfolge der Blätter innerhalb dieses Teilbaumes nicht. Deshalb könnte unser Algorithmus sowohl Zeichnung a) als auch Zeichnung b) ausgeben. Beide Zeichnungen visualisieren den selben Reconciliation Tree. Zeichnung a) hat jedoch eine Kreuzung mehr als Zeichnung b)

Leider konnten wir keine Güte für unseren Algorithmus beweisen. Trotzdem hat sich die Lesbarkeit von unserer Zeichnung im Vergleich zu den Zeichnungen von Douglas' unseres Erachtens deutlich verbessert. In Abbildung 4.2 ist eine von Douglas' Programm erstellte Zeichnung zu sehen. Durch die meist schmalen Artenbaumkanten entstehen eine Reihe von sich überlappenden Kanten. Die Blätter in Douglas' Zeichnung sind nicht gleichmäßig auf den gegebenen Platz in der Artenbaumkante verteilt. Dadurch fallen viele Knoten dicht aufeinander, wo es gar nicht notwendig wäre. In Abbildung 4.3 ist der selbe Reconciliation Tree, wie in Abbildung 4.2 abgebildet. Abbildung 4.3 ist mit einer Implementierung des in dieser Arbeit vorgestellten Algorithmus gezeichnet. In unserer Zeichnung entstehen keine überlappenden Kanten und die Genbaumblätter sind gleichmäßig auf den gegebenen Platz verteilt. In Douglas' Abbildung entspricht allerdings, im

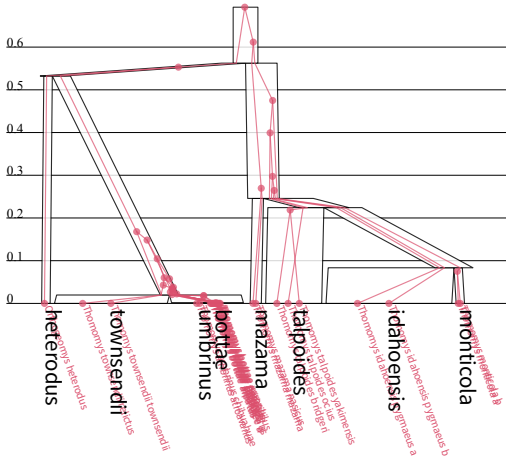



**Abb. 4.2:** Ein Reconciliation Tree der Gebirgstaschenratten erstellt mit dem Programm von Douglas.

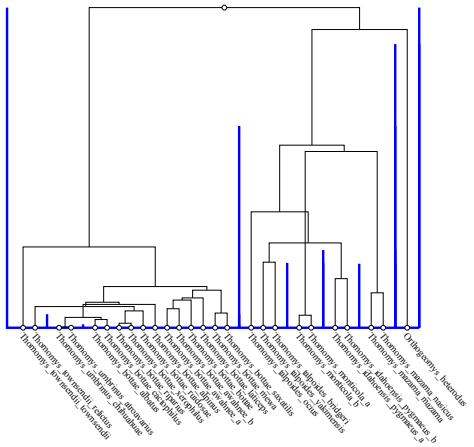


**Abb. 4.3:** Ein Reconciliation Tree der Gebirgstaschenratten erstellt mit dem in dieser Arbeit vorgestellten Algorithmus.

Gegensatz zu unserer Abbildung, die Breite der Artenbaumkanten der Population der Arten. Da dadurch die Lesbarkeit der Zeichnung sehr abnimmt, sollte man die Populationsgröße der Arten vielleicht doch nicht in die Zeichnung mit einbeziehen.

# 5 Zusammenfassung und offene Probleme

In dieser Arbeit haben wir das Problem der Zeichnung von Reconciliation Trees betrachtet. Wir haben gezeigt, dass die Reihenfolge der Artenbaumblätter und die Reihenfolge der Genbaumblätter entscheidend für die Anzahl der Kreuzungen in der Zeichnung sind. In der Arbeit wurde ein vereinfachter Zeichenstil definiert, bei dem die Artenbaumkanten durch die Flächen eines Rechtecks dargestellt werden, die Genbaumkanten rechtwinklig verlaufen und innere Genbaumknoten mittig über ihre Kinder platziert werden. Dann haben wir einen Algorithmus vorgestellt, der eine Zeichnung eines Reconciliation Trees im einfachen Zeichenstil erstellt. Die Hauptaufgabe des Algorithmus besteht darin, die Genbaumblätter zu ordnen, sodass möglichst wenige Kreuzungen entstehen. Dabei wird die Reihenfolge der Artenbaumblätter nicht durch den Algorithmus geändert. Anschließend wird ein Verfahren zur Streckung unserer Zeichnung vorgestellt, der die Breite der Artenbaumblätter anpasst, sodass die Breite dem Verlauf der Population der Arten entspricht. Es ist allerdings unklar, ob diese Streckung zur Lesbarkeit der Zeichnung beiträgt.

Zur Visualisieren von Reconciliation Trees gibt es noch viele ungeklärte Probleme. Ein wichtiges Problem ist das Finden einer optimalen Reihenfolge der Artenbaumblätter, sodass die Kreuzungen in dem Genbaum minimiert werden. Oder das Entwerfen eines Algorithmus, der die Genbaumknoten optimal platziert oder eine garantierte Güte hat, im Bezug auf die Anzahl der entstehenden Kreuzungen. In unserem Zeichenstil werden innere Knoten mittig über seine Kinder platziert. Diese Platzierung ist allerdings nicht optimal im Bezug auf die Anzahl der Kreuzungen. Künftige Arbeiten könnten sich mit der Platzierung der inneren Knoten beschäftigen. Auch der Verlauf der Genbaumkanten ist in dieser Arbeit vereinfacht, dadurch dass unser Zeichenstil rechtwinklige Kanten fordert. Wenn man nur fordert, dass die Genkanten monoton gezeichnet werden, gibt es viele Möglichkeiten die Kanten zu zeichnen. Dadurch lassen sich eventuell Kreuzungen vermeiden.

# Literaturverzeichnis


[BB03] Hans Joachim Böckenhauer und Dirk Bongartz: Phylogenetische Bäume. In: *Algorithmische Grundlagen der Bioinformatik*, Seiten 246–276. Springer, 2003.

[CDS+16] François Chevenet, Jean Philippe Doyon, Celine Scornavacca, Edwin Jacox, Emmanuelle Jousselin und Vincent Berry: SylvX: a viewer for phylogenetic tree reconciliations. *Bioinformatics*, 32(4):608–610, 2016, https://doi.org/10.1093/bioinformatics/btv625.

[DEMS14] Riccardo Dondi, Nadia El-Mabrouk und Krister M Swenson: Gene tree correction for reconciliation and species tree inference: complexity and algorithms. *Journal of Discrete Algorithms*, 25:51–65, 2014.

[Dou20] Jordan Douglas: UglyTrees: a browser-based multispecies coalescent tree visualizer. *Bioinformatics*, Juli 2020, 10.1093/bioinformatics/btaa679, ISSN 1367-4803. btaa679.

[HD09] Joseph Heled und Alexei J. Drummond: Bayesian Inference of Species Trees from Multilocus Data. *Molecular Biology and Evolution*, 27(3):570–580, November 2009, 10.1093/molbev/msp274, ISSN 0737-4038.

[Nak13] Luay Nakhleh: Evolutionary Trees. In: Stanley Maloy und Kelly Hughes (Herausgeber): *Brenner's Encyclopedia of Genetics*, Seiten 549–550. Academic Press, San Diego, 2. Auflage, 2013, 10.1016/B978-0-12-374984-0.00504-0.

[RLGL14] L Yu Rusin, EV Lyubetskaya, K Yu Gorbunov und VA Lyubetsky: Reconciliation of gene and species trees. *BioMed research international*, 2014, 2014, https://doi.org/10.1155/2014/642089.

[RY03] Bruce Rannala und Ziheng Yang: Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. *Genetics*, 164(4):1645–1656, 2003. `https://www.genetics.org/content/164/4/1645`.

[SS+03] Charles Semple, Mike Steel *et al.*: *Phylogenetics*, Band 24. Oxford University Press on Demand, 2003, ISBN 9780198509424, 10.1080/10635150490888895.

# Erklärung

Hiermit versichere ich die vorliegende Abschlussarbeit selbstständig verfasst zu haben, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben, und die Arbeit bisher oder gleichzeitig keiner anderen Prüfungsbehörde unter Erlangung eines akademischen Grades vorgelegt zu haben.

Würzburg, den 24. Februar 2021

. . . . . .
Moritz Niederer