Johann Wolfgang Goethe-Universität
Frankfurt am Main

Fachbereich Biologie und Informatik (15)

SS05

Seminar:
"Aktuelle Themen der Bioinformatik

Phylogenie:

Splits in the Neighborhood of a Tree

A Classification of Consensus Methods for Phylogenetics

Susanne Franssen

Inhaltsverzeichnis:

1. Allgemeine Einleitung

Diese Ausarbeitung zum Thema „Phylogenie in der Bioinformatik“ behandelt zwei Veröffentlichungen „The Splits in the Neighborhood of a Tree“ und „A Classification of Consensus Methods for Phylogenetics“. Beide Veröffentlichungen sind von Prof. David Bryant verfasst, der zurzeit am McGill Centre für Bioinformatik der Universität von Montreal, Kanada tätig ist.

Der Raum aller phylogenetischen Bäume ist sehr komplex und wächst überexponentiell in der Anzahl der Blätter. In „The Splits in the Neighborhood of a Tree“ wird eine Möglichkeit aufgezeigt die Suche im Raum aller Bäume leichter handhabbar zu machen. Phylogenetische Bäume werden durch die Menge ihrer Splits dargestellt. Die Splitnachbarschaft eines festen Baumes wird zum Ausgangspunkt der Suche.
Zunächst werden vier verschiede Baummetriken vorgestellt, mit denen der Abstand zwischen zwei Bäumen auf einem identischen Taxaset ausgedrückt werden kann. Die Splits, die in der Nachbarschaft eines festen Baumes vorkommen, werden charakterisiert. Es wird untersucht, in welchem Maß die Größe einer festen Splitnachbarschaft mit der Anzahl der Blätter anwächst. Bei den gesamten Betrachtungen werden kombinatorische, jedoch keine biologischen Aspekte behandelt.

In „A Classification of Consensus Methods for Phylogenetics“ werden verschiedene Consensusmethoden vorgestellt. Diese Methoden erhalten jeweils eine Sammlung phylogenetischer Bäume auf einem festen Taxaset als Eingabe und geben einen “repräsentativen“ Baum zurück. Des Weiteren werden Gemeinsamkeiten und Unterschiede zwischen den Methoden aufgezeigt, was letztlich zu einer Klassifizierung der Methoden führt.

2. The splits in the Neighborhood of a Tree

2.1. Einleitung

Phylogenese ist die Stammesentwicklung der Lebewesen im Verlauf der Erdgeschichte. Einer der zentralen Aspekte in der Phylogenie ist daher die Rekonstruktion phylogenetischer Bäume für ermittelte Daten. Dies sind z.B. morphologische, anatomische und vor allem Sequenzdaten verschiedener, ’verwandter’ Organismen. Zur Rekonstruktion phylogenetischer Bäume gibt es zwei hauptsächliche Gruppen von Methoden: die distanzbasierten Methoden wie UPGMA und Neighbor-Joining und Merkmalbasierte Methoden wie Maximum Parsimony und Maximum Likelihood.
Um den phylogenetischen Baum zu finden, der eine für einen Baum gegebene Funktion maximiert, oder unter Bayesianischem Ansatz, der einen Baum mit der größten posterior Wahrscheinlichkeit besitzt, muss der Raum aller möglichen phylogenetischen Bäume auf einem festen Taxaset durchsucht werden. Hier ergibt sich das Problem, das dieser Raum aller Bäume sehr komplex ist und überexponentiell mit der Anzahl der Blätter im Baum anwächst. Aus diesem Grund durchsuchen Suchalgorithmen die Nachbarschaften im Raum der Bäume. Dies hat die Motivation, dass Bäume mit großer Likelihood oder auch Bäume, die ähnliche Werte auf einer gegeben Funktion erzielen, dazu neigen in Anhäufungen vorzukommen. Es stellt sich allerdings die Frage, wie die Nachbarschaften von Bäumen zu definieren sind! In diesem Zusammenhang werden später vier verschiedene Metriken vorgestellt, die verschieden Splitnachbarschaften definieren.

Es gibt mehrere Möglichkeiten den Raum aller Bäume handhabbarer zu machen. Die Grundidee ist erst einmal die Dekomposition der Bäume in Mengen von Splits. Splits, das sind Bipartitionen, die durch das Entfernen einer Kante im Baum erzeugt werden. Hierbei ist es wichtig, dass ein Baum durch die Menge aller seiner Splits eindeutig bestimmt ist (gemeint ist die Topologie des Baumes).
Diese Vereinfachung bringt wichtige algorithmische Folgen mit sich. So wurde zum Beispiel gezeigt, dass durch das Erzwingen einer „Baum-Suche“ auf diejenigen Bäume, die eine bestimmte Menge von Splits enthalten, NP-harte Optimierungsprobleme in polynomieller Laufzeit gelöst werden können.
Wenn die Splits in der Nachbarschaft eines Baumes verstanden werden, kann diese Information dazu genutzt werden effizientere Suchalgorithmen zu entwerfen.
Ziel ist es die Splits in der Nachbarschaft eines Baumes anhand der vier am häufigsten genutzten Metriken zu charakterisieren. Dies wird in einem Fall zu einer exakten Formel für die Anzahl der Splits in der Nachbarschaft eines Baumes führen, in anderen Fällen zu asymptotischen Schranken in Abhängigkeit der Blätterzahl des Baumes.

2.2. Terminologie

Im Folgenden beschäftigen wir uns, wenn nicht anders erwähnt, mit ungewurzelten, phylogenetischen, binären X-Bäumen. X ist dabei die Menge aller Blätter des Baumes. Für einen Baum allgemein gilt, dass er keine Knoten von Grad zwei (zwei inzidente Kanten) haben kann. Ein binärer (oder auch fully resolved) Baum hat für jeden inneren Knoten genau drei inzidente Kanten.

Sei T ein binärer X-Baum, so wird die Menge der inneren Knoten von T mit V°(T) und die der inneren Kanten mit E°(T) bezeichnet.
Ein Split A|B von X ist eine Partition in zwei nichtleere Teilmengen A und B. Die Splits, die einen X-Baum T beschreiben, erhält man jeweils durch das Entfernen einer Kante e des Baumes. Der Split, der durch das Entfernen der Kante e erhalten wird, ist mit dieser Kante assoziiert.
Die Menge aller Splits des X-Baumes T werden mit Σ (T) bezeichnet.

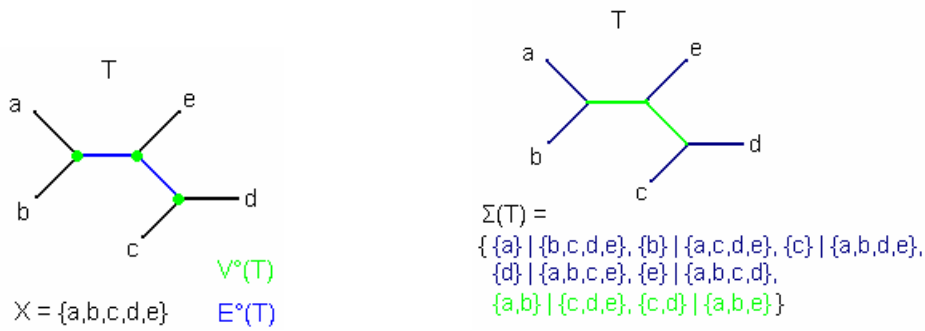

Fig.1: T ist ein binärer phylogenetischer X-Baum auf der Menge X.
Σ (T) bezeichnet die Menge aller Splits von T.

Eine Menge S von Splits über der Menge X ist kompatibel, wenn sie in Σ(T) eines beliebigen Baumes T enthalten ist.
Für jede Menge S von kompatiblen Splits gilt:
für je zwei Splits A|B und C|D aus S ist eine der Schnittmengen A∩C, A∩D, B∩C, B∩D leer.

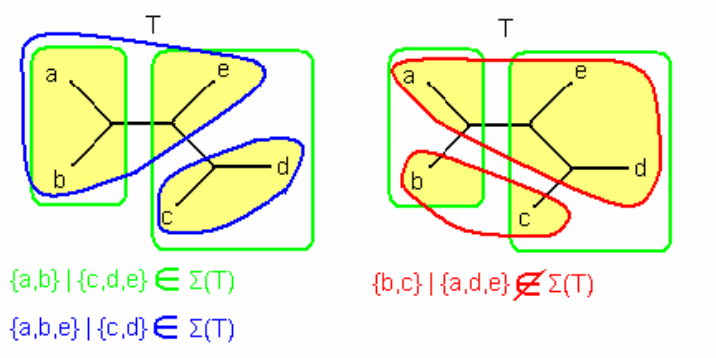

Fig. 2: T ist ein binärer phylogenetischer X-Baum. Die gelben Flächen kennzeichnen jeweils die nichtleeren Schnittmengen „A∩C, A∩D, B∩C, B∩D“.

Die *r-Nachbarschaft* eines binären phylogenetischen X-Baumes T ist folgendermaßen definiert: Sei zuerst einmal UB(X) die Menge aller binären phylogenetischen X-Bäume und d eine auf UB(X) definierte Metrik.
Die r-Nachbarschaft von T in Abhängigkeit von d ist die Menge aller binären X-Bäume, die in der Metrik d einen Abstand kleiner gleich r zu T besitzen. Die r-Nachbarschaft von T wird mit $N_d(T,r)$ bezeichnet.

$$N_d(T,r) = \{T' \in UB(X) : d(T,T') \leq r\}$$

Die *Split Nachbarschaft* von T ist die Menge aller Splits, aus der jeder Split in mindestens einem Baum aus der r-Nachbarschaft von T enthalten ist. Die Split Nachbarschaft von T in der Metrik d und dem Baumabstand r wird mit $S_d(T,r)$ bezeichnet.

$$S_d(T,r) = \{A \mid B : \exists T' \in N_d(T,r), A \mid B \in \Sigma(T')\}$$
$$= \bigcup_{T' \in N_d(T,r)} \Sigma(T')$$

2.3. Baum Metriken

2.3.1. Die Robinson-Foulds Metrik (partition metric)

Die Robinson-Foulds Metrik ist eine der einfachsten Baummetriken. Der Abstand zwischen zwei X-Bäumen T_1 und T_2, $d_{RF}(T_1,T_2)$, ist der Betrag der symmetrischen Distanz der Splits beider Bäume.

$$d_{RF}(T_1,T_2) = ½ |\Sigma(T_1) \Delta \Sigma(T_2)| = ½ |\Sigma(T_1) - \Sigma(T_2)| + ½ |\Sigma(T_2) - \Sigma(T_1)|$$

Die Robinson-Foulds Distanz zwischen zwei phylogenetischen X-Bäumen kann in Zeit O(|X|) berechnet werden.
Die Robinson-Foulds Metrik kann auf Bäume mit gewichteten Kanten erweitert werden. Jede Kante des betrachteten Baumes besitzt ein positives Gewicht bzw. eine Länge. Um die Distanz zwischen zwei gewichteten Bäumen T_1 und T_2 zu berechnen, wird jedem Split A|B aus $\Sigma(T_i)$, i = 1,2, das Kantengewicht seiner assoziierten Kante zugeordnet. Das Gewicht der Kante, durch deren Entfernen der Split A|B entsteht, wird mit $w_i(A|B)$ bezeichnet. Alle Splits von X, die weder in T_1 noch in T_2 vorkommen erhalten ein Kantengewicht von 0.

Die gewichtete Robinson-Foulds Distanz d_ω ist nun wie folgt definiert:

$$d_\omega(T_1, T_2) = \sum_{A|B \in \Sigma(T_1) \cup \Sigma(T_2)} |w_1(A|B) - w_2(A|B)|$$

2.3.2. Die Nearest Neighbor Interchange Metrik

Die Nearest Neighbor Interchange Metrik basiert auf der Nearest Neighbor Interchange (NNI) Operation. Die NNI kann an einer beliebigen inneren Kante e eines X-Baumes T stattfinden. Durch Entfernen der Kante e und der zu e inzidenten Knoten u und v, erhält man vier Teilbäume über den Mengen A_1, A_2, A_3 und A_4. Je zwei dieser Teilbäume befinden sich in T in der gleichen Komponente von (T – e). Durch eine NNI Operation werden zwei dieser Teilbäume, die sich jeweils in verschiedenen Komponenten von (T – e) befinden ausgetauscht.

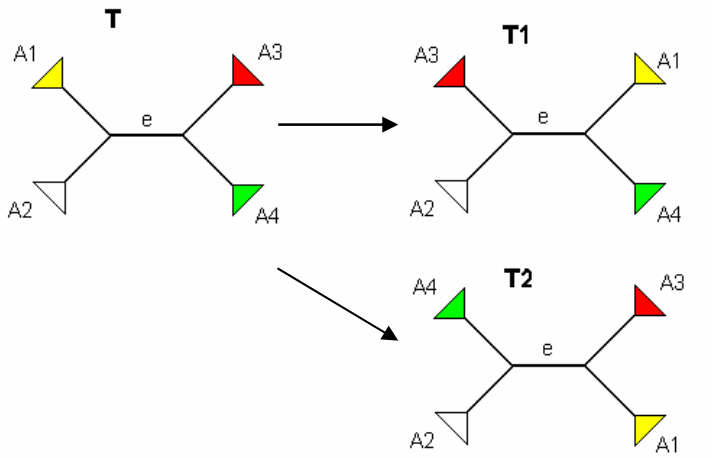

Fig. 3: Baum T kann durch eine NNI Operation an Kante e in zwei verschiedene Bäume, T_1 oder T_2, umgewandelt werden. Die Dreiecke repräsentieren Teilbäume über den Mengen A_i.

Für jeden binären X-Baum T mit n Blättern gilt, dass es genau 2(n-3) viele X-Bäume mit der Robinson-Foulds Distanz gleich Eins zu T gibt. Dies erklärt sich folgendermaßen:
(n-3) ist die Anzahl der inneren Kanten in einem binären X-Baum mit n Blättern. Weiterhin gibt es 2 Möglichkeiten eine NNI Operation an einer festen Kante e durchzuführen (Fig. 3). Durch eine NNI Operation an T wird genau ein Split aus Σ(T) durch einen anderen ersetzt. Hierbei handelt es sich jeweils um den Split, der mit der Kante e assoziiert ist. Folglich sind 2(n-3) NNI Operationen möglich, die jeweils einen anderen X-Baum T_i erzeugen, der sich genau in einem Split von T unterscheidet. Dies entspricht einem Robinson-Foulds Abstand von eins.

Jeder binäre X-Baum T_1 kann durch eine Folge von NNI Operationen in jeden beliebigen binären X-Baum T_2 überführt werden. Die Nearest Neighbor Interchange Distanz $d_{NNI}(T_1,T_2)$ zwischen zwei X-Bäumen T_1 und T_2 ist die kleinstmögliche Anzahl von NNI Operationen, die benötigt wird um T_1 in T_2 zu überführen. Die Bestimmung von $d_{NNI}(T_1,T_2)$ ist NP-hart (Das Gupta et al.).

Es gilt: $d_{NNI}(T_1, T_2) \geq d_{RF}(T_1, T_2)$.
Es gibt keinen Fall, für den $d_{NNI}(T_1, T_2)$ kleiner als $d_{RF}(T_1, T_2)$ ist, da es durch eine NNI Operation nur möglich ist genau einen Split aus $\Sigma(T_i)$ zu verändern. Umgekehrt gibt es jedoch den Fall, dass $d_{NNI}(T_1, T_2) > d_{RF}(T_1, T_2)$. Hierbei ist es nötig auf dem Weg von T_1 nach T_2 (über eine Folge von X-Bäumen T'_i, wobei T'_{i+1} durch eine NNI Operation von T'_i aus

erreicht wird) eine NNI Operation von T'_i nach T'_{i+1} durchzuführen, bei der ein Split $A \mid B \notin \Sigma(T_2)$ in einen Split $A' \mid B' \notin \Sigma(T_2)$ umgewandelt wird (Fig. 4).

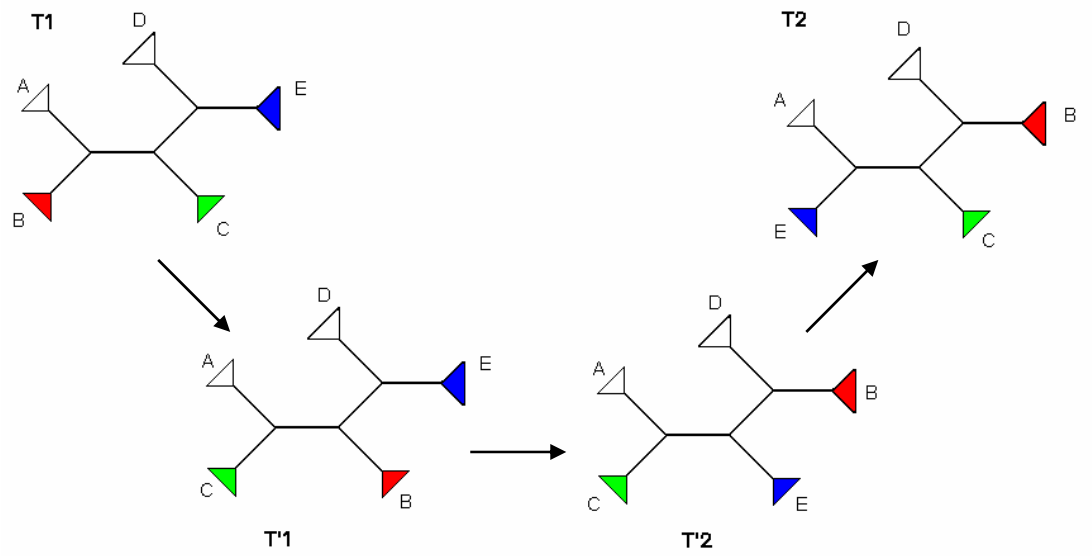

Fig. 4: In diesem Beispiel ist $d_{RF}(T_1,T_2) = 2$ und $d_{NNI}(T_1,T_2) = 3$. In der ersten hier dargestellten NNI Operation wird der Split $A \cup B \mid C \cup D \cup E \notin \Sigma(T_2)$ durch den Split $A \cup C \mid B \cup D \cup E \notin \Sigma(T_2)$ ersetzt.

2.3.3. Die Subtree Prune and Regraft Metrik

Die Operation in der Subtree Prune and Regraft (SPR) Metrik besteht aus drei Schritten: Zuerst wird eine Kante {u,v} aus dem gegebenen X-Baum T ausgewählt und entfernt. Dabei entstehen zwei verbundene Teilbäume T_u und T_v (enthalten jeweils den Knoten u bzw. v). Im nächsten Schritt wird eine Kante im Teilbaum T_v ausgewählt. Auf ihr wird ein neuer Knoten w eingefügt. Im letzten Schritt werden u und w durch eine neue Kante verbunden und alle Knoten im Baum mit nur zwei inzidenten Kanten unterdrückt.

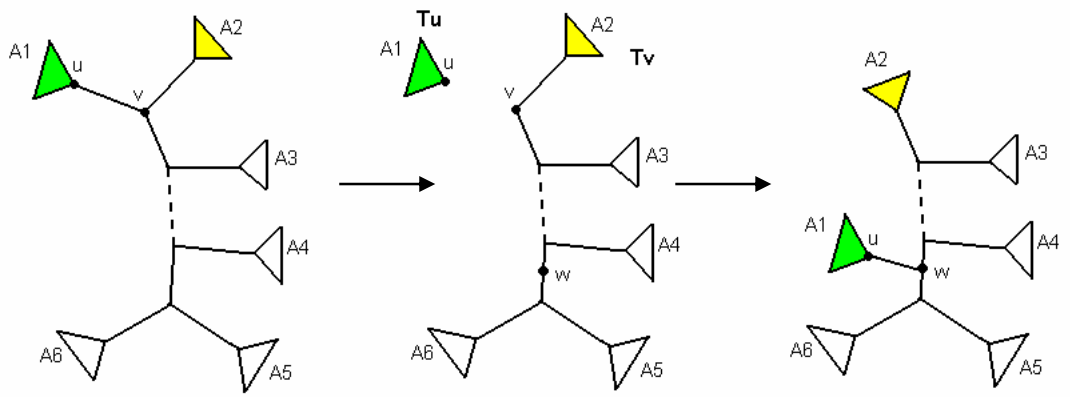

Fig. 5: Darstellung der Schritte in einer SPR Operation: Die Kante {u,v} wird entfernt. Es entstehen zwei Teilbäume T_u und T_v. Eine Kante in T_v wird zweigeteilt, indem sie einen neuen Knoten erhält. Der neue Knoten wird mit u verbunden. Alle Kanten mit einem Grad von zwei werden unterdrückt.

Die Subtree Prune and Regraft Distanz zwischen zwei Bäumen T_1 und T_2 $d_{SPR}(T_1,T_2)$ entspricht der minimalen Anzahl an SPR Operationen, die benötigt werden um T_1 in T_2 zu

überführen. Die Berechnungskomplexität von $d_{SPR}(T_1,T_2)$ ist noch nicht im Detail bestimmt. Es wird jedoch vermutet, dass sie ebenfalls NP-hart ist.
Jede NNI Operation ist auch eine SPR Operation das heißt, dass zwei Bäume T_1 und T_2 mit $d_{NNI}(T_1,T_2) = 1$ auch eine SPR Distanz von eins haben. Zusätzlich sind durch eine SPR Operation auf T_1 auch Bäume zu erreichen, für die man mehrere NNI Operationen benötigen würde. Es folgt: $d_{SPR}(T_1,T_2) \leq d_{NNI}(T_1,T_2)$.

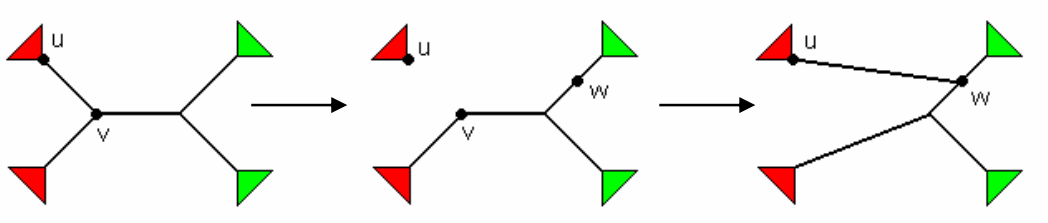

Fig. 6: Darstellung der Auswirkung einer NNI Operation durch eine SPR Operation.

2.3.3. Die Tree Bisection Reconnection Metrik

Die Tree Bisection Reconnection (TBR) Operation ist eine Erweiterung der SPR Operation. Der Unterschied zwischen den beiden Operationen besteht darin, dass in jedem Teilbaum, T_v und T_u, der durch das Entfernen der Kante {u,v} entsteht, ein weiterer Knoten auf je einer beliebigen Kante eingefügt wird. Die beiden neuen Knoten werden durch eine neue Kante verbunden und alle Knoten vom Grad zwei werden unterdrückt. Besteht einer der Teilbäume T_i aus nur einem Knoten, so wird dieser im Reconnection Schritt durch eine neue Kante mit dem neu erzeugten Knoten in dem anderen Teilbaum verbunden.

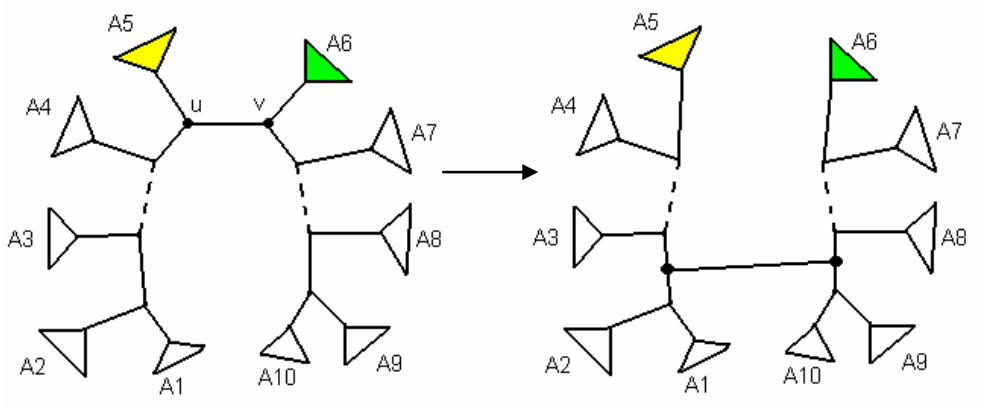

Fig. 7: Die Kante {u,v} wird entfernt, wodurch zwei Teilbäume entstehen. Jeder Teilbaum erhält einen neuen Knoten auf einer beliebigen Kante. Die Teilbäume werden über eine neue Kante zwischen den beiden neuen Knoten verbunden.

Die Tree Bisection Reconnection Distanz zwischen zwei Bäumen T_1 und T_2 $d_{TBR}(T_1,T_2)$ ist die minimale Anzahl an TBR Operationen, die benötigt wird um T_1 in T_2 zu überführen. Die Berechnung von $d_{TBR}(T_1,T_2)$ ist ein NP-hartes Problem (B.L. Allen und M. Steel, Subtree transfer operations and their induced metrics on evolutionary trees).
Da jede SPR Operation auch durch eine TBR Operation dargestellt werden kann folgt: $d_{TBR}(T_1,T_2) \leq d_{SPR}(T_1,T_2)$.

2.4. Splits in den Nachbarschaften der vorgestellten Metriken

2.4.1. Splits in der Robinson-Foulds Nachbarschaft

Ziel ist es nun die Anzahl der Splits in der Robinson-Foulds Nachbarschaft in Abhängigkeit der Anzahl der Blätter eines Baumes zu bestimmen. In diesem Fall werden wir eine asymptotische Schranke für die Anzahl der Splits in Abhängigkeit der Blattzahl erhalten. Hierzu ist es jedoch zuerst nötig bestimmte Eigenschaften der Splits zu zeigen.

Sei T ein binärer X-Baum und A|B ein Split von X. Für diesen Split gilt einer der beiden folgenden Fälle: $A|B \in \Sigma(T)$ oder $A|B \notin \Sigma(T)$. Wenn $A|B \in \Sigma(T)$ gilt, ist A|B paarweise kompatibel mit jedem Split aus Σ(T). Gilt jedoch $A|B \notin \Sigma(T)$, so ist A|B paarweise inkompatibel mit *einigen* Splits aus Σ(T). |Σ(T)| ist die maximale Anzahl an kompatiblen Splits von T. Die paarweise mit A|B inkompatiblen Splits aus Σ(T) werden im Folgenden die mit A|B im Konflikt stehenden Splits aus Σ(T) oder als „conflicting splits" bezeichnet. Entsprechend dazu heißen die Kanten, die zu einem „conflicting split" assoziiert sind „conflicting edges" oder im Konflikt mit A|B stehende Kanten.
In diesem Zusammenhang ist zu bemerken, dass generell nur innere Kanten im Konflikt zu einem Split A|B stehen können. Die restlichen Kanten verbinden jeweils nur ein einzelnes Element a der Menge X mit allen übrigen Elementen der Menge X (X \ {a}) und sind daher in jeder Menge Σ(T'), mit T' ein binärer X-Baum, enthalten.

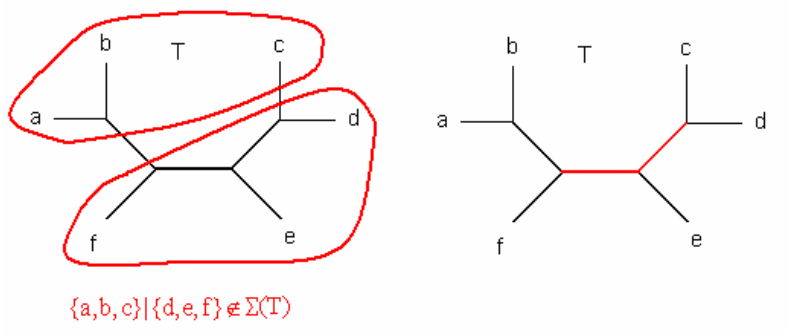

Fig. 8: Beispiel zur Darstellung der mit dem Split $\{a,b,c\}|\{d,e,f\}$ im Konflikt stehen den Kanten aus T. Die „conflicting edges" sind rot gekennzeichnet.

Lemma 2.1.:
Sei T ein binärer X-Baum und A|B ein Split von X, dann gilt: Die mit A|B im Konflikt stehenden Kanten von T bilden einen verbundenen Subgraph.

Beweis:
Gibt es nur eine einzige Kante, die mit A|B im Konflikt steht, so ist der entsprechende aus einer Kante bestehende Subgraph zwangsläufig zusammenhängend. Gibt es jedoch mehr als eine Kante die mit A|B im Konflikt steht, wird folgendes Vorgehen gewählt:
e_1 und e_k sind zwei mit A|B im Konflikt stehende Kanten. e_1, e_2, e_3, …, e_k sind die Kanten auf dem Weg von e_1 nach e_k im Baum T. Es soll nun gezeigt werden, dass die Kanten e_2, e_3, …, e_{k-1} auch im Konflikt mit A|B stehen.

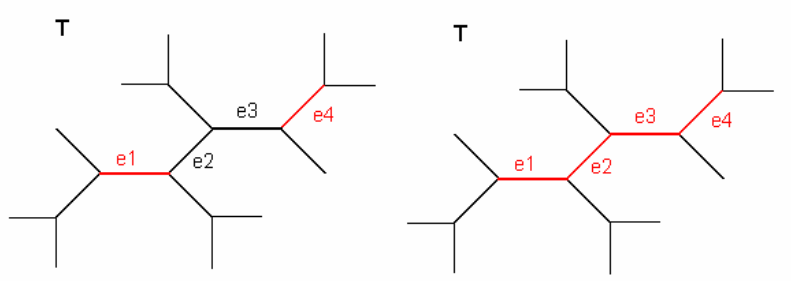

Fig. 9: Verdeutlichung der Vorangehensweise: Beispiel: Es ist bekannt, dass A|B im Konflikt mit e_1 und e_4 steht. Es soll nun gezeigt werden, dass A|B auch im Konflikt mit e_2 und e_3 steht.

Für alle i = 1, 2, …, k ist $X_i|Y_i$ der mit e_i assoziierte Split, wobei gilt: $X_1 \subset X_2 \subset \ldots \subset X_k$. Da $X_i|Y_i$ Bipartitionen von X sind gilt umgekehrt: $Y_k \subset Y_{k-1} \subset \ldots \subset Y_1$.
Zu Beginn wurde davon ausgegangen, dass e_1 mit A|B im Konflikt steht. Aus diesem Grund gilt: Es gibt ein $a \in X_1 \cap A$ und ein $b \in X_1 \cap B$. Da e_k ebenfalls im Konflikt mit A|B steht, gilt analog: Es gibt ein $a' \in Y_k \cap A$ und ein $b' \in Y_k \cap B$.
Aufgrund der Teilmengenbedingung für X_i bzw. Y_i $(\forall i \in \{1, 2, \ldots k\})$ gilt für alle $i \in \{1, 2, \ldots k\}$: Es gibt ein $a \in X_i \cap A$, ein $b \in X_i \cap B$, ein $a' \in Y_i \cap A$ und ein $b' \in Y_i \cap B$. Damit stehen die Splits $X_i|Y_i$ für alle $i \in \{1, 2, \ldots k\}$ im Konflikt mit dem Split A|B. □

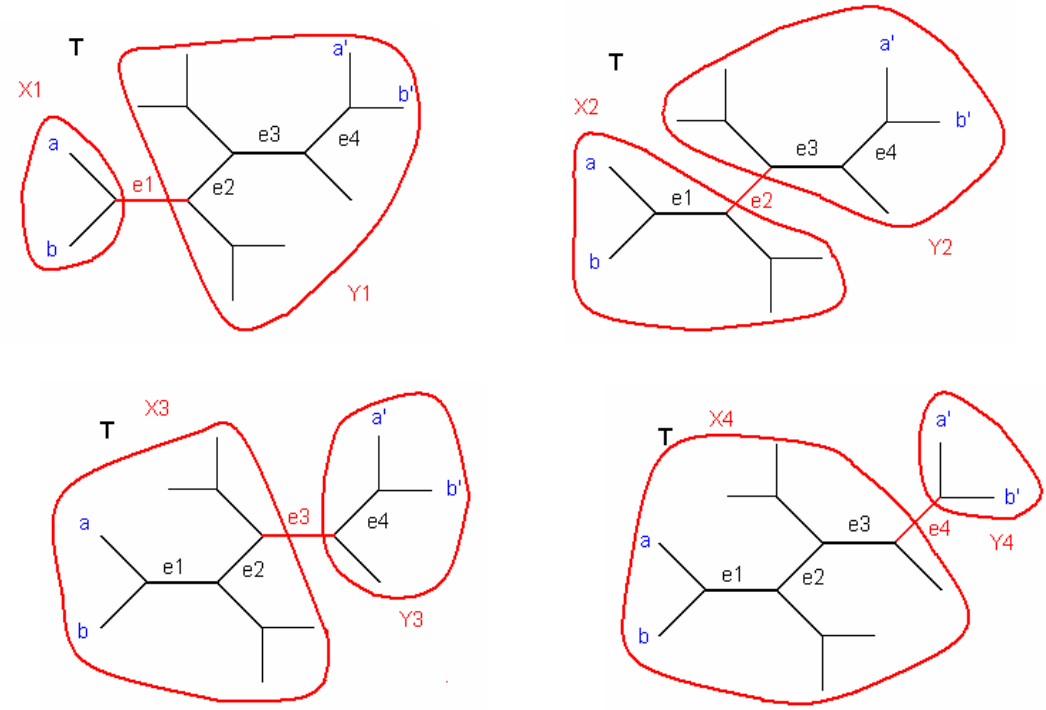

Fig. 10. Die Beweisidee ist hier an einem Beispiel verdeutlicht.
Es ist bekannt, dass der Split A|B mit den Kanten e_1 und e_4 im Konflikt steht. Es gibt daher ein $a \in X_1 \cap A$, ein $b \in X_1 \cap B$, ein $a' \in Y_4 \cap A$ und ein $b' \in Y_4 \cap B$ mit $a, a' \in A$ und $b, b' \in B$.
Aufgrund der Teilmengenbedingung für X_i bzw. Y_i folgt $\forall i \in \{1,2,3,4\}$:
Es gibt ein $a \in X_i \cap A$, ein $b \in X_i \cap B$, ein $a' \in Y_i \cap A$ und ein $b' \in Y_i \cap B$.

Es gibt eine konstruktive Charakterisierung der Menge aller Splits, die jeweils mit genau allen Kanten eines gegebenen zusammenhängenden Subgraphen des X-Baumes T aus den Kanten der Menge E' mit $E' \subseteq E°(T)$ im Konflikt stehen.

Sei V' die Menge aller Knoten, die zu mindestens einem der Knoten in E' inzident sind. $\Pi(E') = A_1|A_2| \ldots |A_k$ sei die Partition der Menge X, die durch Komponenten von (T-V') erzeugt wird. Zwei Blöcke A_i und $A_{i'}$ dieser Partition sind adjazent, wenn sie sich in der gleichen Zusammenhangskomponente von (T-E') befinden.

Um alle Splits, die mit dem geforderten Subgraphen aus den Kanten in E' in Konflikt stehen, zu erhalten muss folgende Anweisung ausgeführt werden: Bilde alle möglichen Splitkombinationen aus den Blöcken A_i mit der Einschränkung, dass adjzente Blöcke nicht in der gleichen Menge (A bzw. B) des Splits A|B vorkommen dürfen.

Das Vorgehen ist im folgenden Beispiel verdeutlicht:

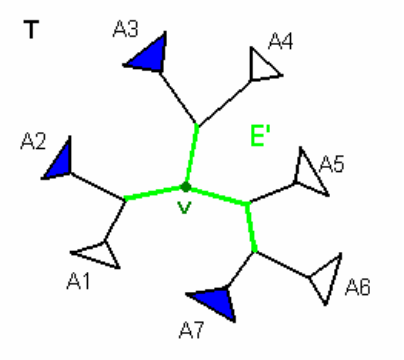

Fig. 11: Gegeben ist ein X-Baum T mit einer Menge von Kanten E' (grün), die einen zusammenhängenden Subgraphen von T bilden. Zusätzlich ist ein Split dargestellt (blaue und weiße Teilmenge von X), die genau mit allen Kanten aus E' im Konflikt stehen. v ist ein im Konflikt stehender Knoten (2.4.2.).

Für Beispiel aus Fig. 11 gilt:
$\Pi(E') = A_1|A_2|A_3|A_4|A_5|A_6|A_7$, adjazente Blöcke: {A1, A2} , {A3, A4} , {A6, A7} ,
Die folgenden Splits stehen im Konflikt mit jeweils allen Kanten aus E':

$$\{A1 \cup A3 \cup A6\}|\{A2 \cup A4 \cup A7 \cup A5\}$$
$$\{A1 \cup A3 \cup A7\}|\{A2 \cup A4 \cup A6 \cup A5\}$$
$$\{A1 \cup A4 \cup A6\}|\{A2 \cup A3 \cup A7 \cup A5\}$$
$$\{A1 \cup A4 \cup A7\}|\{A2 \cup A3 \cup A6 \cup A5\}$$
$$\{A2 \cup A3 \cup A6\}|\{A1 \cup A4 \cup A7 \cup A5\}$$
$$\{A2 \cup A3 \cup A7\}|\{A1 \cup A4 \cup A6 \cup A5\}$$
$$\{A2 \cup A4 \cup A6\}|\{A1 \cup A3 \cup A7 \cup A5\}$$
$$\{A2 \cup A4 \cup A7\}|\{A1 \cup A3 \cup A6 \cup A5\}$$

Allgemein gilt für die Anzahl der Splits, die jeweils mit den Kanten E' eines zusammenhängenden Subgraphen im Konflikt stehen folgende Formel:
2.2.

$$2^{a+b-1}$$

Hierbei ist a gleich der Anzahl aller Paare von adjazenten Blöcken $\{A_i, A_{i'}\}$ und b gleich der Anzahl aller Blöcke A_i, die zu keinem anderen Block A_j adjazent sind.

Die Formel erklärt sich folgendermaßen:
2^a ist die Anzahl der verschiedenen Kombinationsmöglichkeiten zur Belegung eines Feldes mit a vielen Positionen, wenn man davon ausgeht, dass jede Position jeweils nur zwei verschiede Werte annehmen kann. Diese Werte entsprechen jeweils einem der beiden Blöcke (A_i oder $A_{i'}$) aus einem Paar von adjazenten Blöcken $\{A_i, A_{i'}\}$.
Wenn man davon ausgeht, dass alle Elemente eines Feldes Elemente der Menge A des Splits A|B sein sollen (B = X/A), so müssen in einem Feld mindestens a viele Elemente enthalten sein (alle adjazenten Blöcke müssen getrennt sein).
Bis jetzt wurde angenommen, dass alle der Blöcke A_i, die zu keinem anderen Block A_j adjazent sind, in der Menge B enthalten sind. Dies ist zwar möglich, jedoch nicht gefordert. Jeder dieser Blöcke kann (und muss) jeweils einmal in jeder Menge A und im anderen Fall in der Menge B enthalten sein. Es ergibt sich somit ein weiterer Faktor 2^b.
Bisher wurde nun allerdings außer Acht gelassen, dass zwei Splits A|B und B|A nicht verschieden sind. Das heißt jeder Split wurde doppelt gezählt. Die Formel muss um den zusätzlichen Faktor (1/2) erweitert werden. Es folgt: $2^{a+b}/2$.
Mit diesem Vorwissen können nun die Splits in der Robinson-Foulds Nachbarschaft charakterisiert werden.

Theorem 2.3.:
Sei T ein binärer phylogenetischer X-Baum. Ein Split A|B ist genau dann in $S_{RF}(T,r)$, wenn er mit höchstens r Kanten im Konflikt steht.

Beweis:
$\Rightarrow$ Angenommen $A|B \in \Sigma(T')$ und $d_{RF}(T,T') \leq r$. In diesem Fall gibt es höchstens r viele Splits in $\Sigma(T) - \Sigma(T')$. Da A|B mit allen Splits aus $\Sigma(T')$ kompatibel ist folgt:
A|B ist kompatibel mit allen Splits aus $\Sigma(T)$ mit Ausnahme von höchstens r vielen.

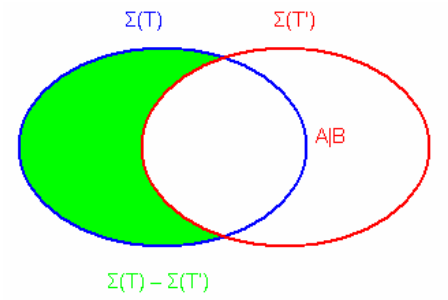

Fig. 12: Split A|B ist mit den Splits aus $\Sigma(T)$ mit Ausnahme der Splits aus $\Sigma(T) - \Sigma(T')$ kompatibel. $|\Sigma(T) - \Sigma(T')| \leq r$
Der Fall $A|B \in \Sigma(T) \cap \Sigma(T')$ ist hier nicht dargestellt.

$\Leftarrow$ Angenommen A|B steht höchstens im Konflikt mit r vielen Kanten aus T. S ist die zugehörige Menge von Splits aus $\Sigma(T)$, die mit A|B im Konflikt stehen ($S \subseteq \Sigma(T)$).
Es folgt: $(\Sigma(T) - S) \cup \{A|B\}$ ist kompatibel. Es gibt einen binären X-Baum T', der $(\Sigma(T) - S) \cup \{A|B\}$ enthält (Falls $|(\Sigma(T) - S) \cup \{A|B\}| < |\Sigma(T)|$ müssen noch weitere Splits von X hinzugefügt werden, die mit den bereits vorhandenen kompatibel sind, um einen binären Baum zu erhalten.). T' enthält $|\Sigma(T)| - |S|$ mit $|S| \leq r$ viele Splits, die auch in $\Sigma(T)$ enthalten sind und genügt daher $d_{RF}(T, T') \leq r$.

□

Sei T° der Subgraph bestehend aus den inneren Knoten und Kanten des X-Baumes T. Ein zusammenhängender Subgraph von T° mit k Knoten (und k – 1 Kanten) wird als k-subtree von T° bezeichnet.
Die Anzahl dieser k-subtrees ($\forall k$) für ein T° wird als *Whitney number* von T° bezeichnet. Es gibt keine allgemeingültige geschlossene Formel für die Whitney Zahl eines Baumes T. Die größten Whitney Zahlen erhalten allerdings diejenigen Bäume mit Knoten von hohem Grad. Durch die Restriktion dieser Betrachtung auf binäre Bäume kann eine obere Schranke für die Whitney Zahl eines Baumes gefunden werden, die für ein beschränktes k linear in der Anzahl der Blätter wächst.
Die Catalan Zahlen sind eine Folge von Zahlen. Sie beschreiben die Anzahl geordneter (Unterscheidung linker und rechter Sohn), binärer Bäume mit n Knoten. Die geschlossene Formel für die Catalan Zahlen lautet folgendermaßen: $C_n = \frac{1}{n+1} * \frac{(2n)!}{2n!}$
Die ersten sechs Catalan Zahlen: 1, 2, 5, 14, 42, 132, …

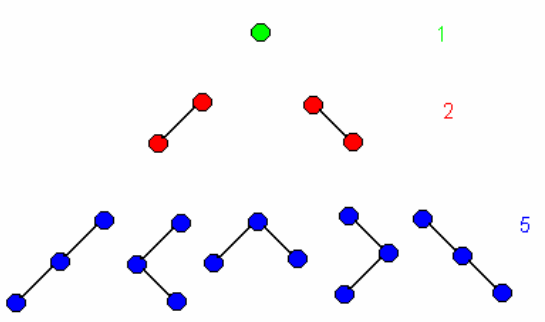

Fig. 13: Die n-te Catalan Zahl ist die Anzahl aller geordneten Binären Bäume mit n Knoten.

Lemma 2.4.:
Sei T ein binärer phylogenetischer X-Baum. Die Anzahl der k-subtrees von T° ist $O(n\ C_k)$ mit $n = |X|$ und C_k ist die k-te Catalan Zahl.

Beweis:
Aus T° wird ein beliebiges Blatt ausgewählt. Alle Kanten aus T° erhalten nun eine Orientierung, die von diesem Blatt weggerichtet ist. Für jeden der n-2 inneren Knoten gilt: Die Anzahl der k-subtrees mit Wurzel v ist durch C_k beschränkt. Es ergibt sich also (n-2) * C_k als obere Schranke für die Anzahl an k-subtrees. □

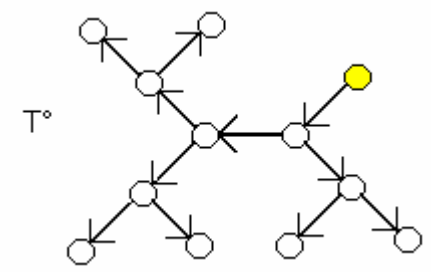

Fig. 14: Verdeutlichung der Beweisidee an einem Beispiel: Alle Kanten in T° erhalten eine Orientierung, die von dem beliebig gewählten, gelb markierten Blatt weggerichtet ist. T° enthält n-2 Knoten. Jeweils von jedem dieser Knoten v ausgehend ist die Anzahl der möglichen k-subtrees mit Wurzel v durch C_k beschränkt. Auf diese Weise werden alle k-subtrees in T° betrachtet.

Mit Lemma 2.1., 2.2., Theorem 2.3. und Lemma 2.4. ist es möglich die Anzahl der Splits in $S_{RF}(T,r)$ asymptotisch in Abhängigkeit von n mit n = |X| für ein festes r zu bestimmen:

Ein Split ist nur in $S_{RF}(T,r)$ enthalten, wenn er mit höchstens r vielen Kanten aus $\Sigma(T)$ im Konflikt steht (Theorem 2.3.).
Daraus ergeben sich folgende Fragen:

a) Wie viele k-subtrees mit k = 1, 2, …, r gibt es? (Whitney Zahl eines Baumes)
b) Wie viele Splits gibt es, die mit genau allen Kanten eines gegebenen k-subtrees im Konflikt stehen?

Zu a) Die Anzahl der k-subtrees von T° für ein festes k ist durch $O(n\ C_k)$ beschränkt. Die Whitney Zahl eines Baumes ist durch $O((\sum_{1\le k\le r} C_k) * n)$ beschränkt.

Zu b) Die Anzahl an Splits, die mit genau allen Kanten eines gegebenen k-subtrees im Konflikt stehen, ist nicht abhängig von der Blattzahl des X-Baumes T, sondern nur von den Variablen a und b und damit von k, der Größe des k-subtrees (2.2.).

Korollar 2.5.:
Die Anzahl der Splits in $S_{RF}(T,r)$ ist linear in n für ein festes r.

2.4.2. Splits in der Nearest Neighbor Interchange Nachbarschaft

Wie bereits erwähnt wurde gilt: $d_{RF}(T_1,T_2) \le d_{NNI}(T_1,T_2)$. Daraus folgt $S_{NNI}(T,r) \subseteq S_{RF}(T,r)$. Die Anzahl der Splits in der Nearest Neighbor Interchange Nachbarschaft wächst folglich auch linear in der Anzahl der Blätter für ein festes r. Die Splits in der NNI Nachbarschaft haben eine elegante graphische Charakterisierung:
Sei v ein innerer Knoten eines binären phylogenetischen X-Baumes und A|B sei ein Split von X. v ist ein mit A|B im Konflikt stehender Knoten, wenn alle zu v inzidenten Kanten mit A|B im Konflikt stehen.

Theorem 2.6.:
Sei T ein binärer phylogenetischer X-Baum, A|B ein Split von X und E', V' die Kanten und Knoten von T, die mit A|B im Konflikt stehen. A|B ist in $S_{NNI}(T,r)$ genau dann wenn $|E'|+|V'| \le r$ gilt.

Beweis:
$\Rightarrow$ Sei $A|B \in \Sigma(T')$ und $d_{NNI}(T,T') = s \le r$. Es gibt eine Folge von X-Bäumen T_0, T_1, T_2, …, T_s mit T' = T_0 und T = T_s, so dass für alle i = 0, 1, …, s-1 gilt: T_{i+1} unterscheidet sich von T_i durch eine NNI Operation. Weiterhin seien E_i' und V_i' die Kanten bzw. Knoten von T_i. Behauptung: $|E_i'|+|V_i'| \le i$, für alle i = 0, 1, …, s. Dies impliziert $|E'|+|V'| = |E_s'|+|V_s'| \le s \le r$. Die Behauptung gilt für i = 0 (Induktionsanker). $|E_0'|+|V_0'| \le 0$ gilt, da T_0 identisch mit T' ist und $A|B \in \Sigma(T')$ gilt. Angenommen Die Behauptung gilt für alle $i \le j$ und T_{j+1} wird aus T_j durch einen NNI um die Kante {u,v} erhalten.

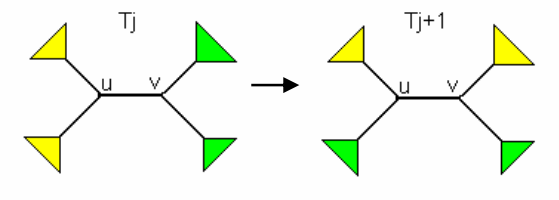

Fig. 15: NNI Operation an Kante {u,v}.

Es können zwei verschiedene Hauptfälle auftreten:

a) Kante {u,v} steht noch nicht im Konflikt mit A|B.
b) Kante {u,v} steht bereits im Konflikt mit A|B.

Zu a) Für diesen Fall gibt es zwei Möglichkeiten für den Baum T_j: Es stehe genau eine der vier zu {u,v} adjazenten Kanten im Konflikt mit A|B, oder es stehen genau zwei der vier zu {u,v} adjazenten Kanten im Konflikt mit A|B und seien o.B.d.A. diese beiden Kanten adjazent zu u. Die Möglichkeit, dass es im Baum T_j gleichzeitig Kanten gibt, die nur zu u, und Kanten, die nur zu v inzident sind, gibt es nicht, da in diesem Fall die Kante {u,v} bereits im Baum T_j im Konflikt mit A|B stehen müsste (Lemma 2.1.).

Fig. 16: Dies sind die zwei Möglichkeiten, wie der Baum T_j im Fall a) aussehen kann.
Rote Kanten stehen im Konflikt mit A|B.

Durch eine NNI Operation an der Kante {u,v} wird die Anzahl der zu A|B im Konflikt stehenden Kanten um genau eins erhöht. Die Anzahl der „conflicting vertices“ bleibt unverändert. Die X-Bäume T_{j+1}, die aus T_j durch eine NNI Operation entstehen sind in der folgenden Grafik dargestellt.

Fig.17: Darstellung der X-Bäume T_{j+1}, die in Fall a) aus T_j durch eine NNI Operation entstehen können.
Rote Kanten stehen im Konflikt mit A|B.

Zu b) Im Folgenden sind alle möglichen X-Bäume T_j dargestellt, die in diesem Fall kombinatorisch gesehen eintreten können:

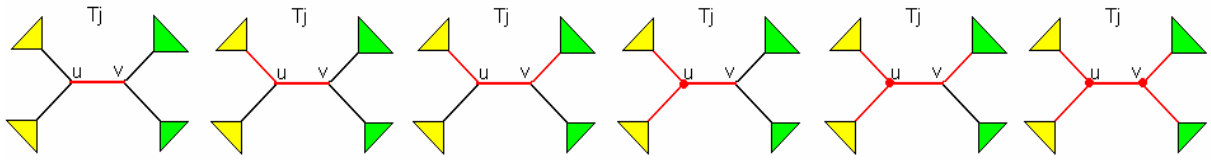

Fig. 18: Darstellung aller möglichen X-Bäume T_j, die in Fall b) kombinatorisch gesehen eintreten können.
Die rot markierten Kanten und Knoten stehen im Konflikt mit A|B.

Durch eine NNI Operation an der Kante {u,v} entstehen o.B.d.A. folgende Bäume T_{j+1}:

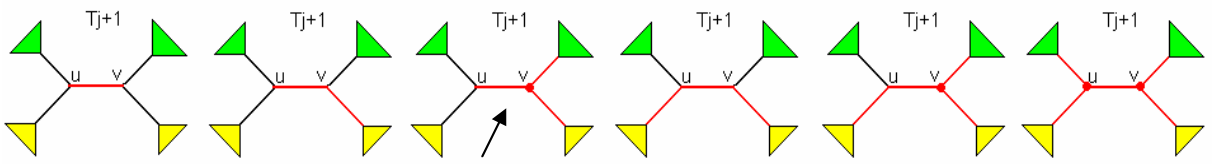

Fig. 19: Darstellung der X-Bäume T_{j+1}, die in Fall b) aus T_j durch eine NNI Operation entstehen können. Die rot markierten Kanten und Knoten stehen im Konflikt mit A|B. Der dritte dargestellte Baum ist der einzige, bei dem die Anzahl der mit A|B im Konflikt stehenden Kanten zunimmt (um genau eins). Es ist allerdings möglich, dass hier durchgeführte NNI Operationen in der Baumfolge von T_0 nach T_s nie auftreten.

Wie in den Abbildungen gezeigt ist gibt es auch in Fall b) keine mögliche NNI Operation, die die Anzahl der zu A|B im Konflikt stehenden Kanten und Knoten zusammen um mehr als eins erhöht. Die Anzahl der im Konflikt stehenden Kanten bleibt unverändert. Die Anzahl der im Konflikt stehenden Knoten kann um höchstens eins erhöht werden.

Da es in keinem Fall möglich ist die Anzahl der zu A|B im Konflikt stehenden Kanten und Knoten zusammen um mehr als eins zu erhöhen, folgt: $|E'_{j+1}|+|V'_{j+1}| \leq |E'_j|+|V'_j| + 1 \leq j+1$ für j = 0 (siehe Induktionsanker). Dass diese Aussage auch für alle j = 1, 2, …, s-1 gilt, folgt durch Induktion.

⇐ Angenommen A|B steht im Konflikt mit den Kanten aus E', den Knoten aus V' und es gilt $|E'|+|V'| \leq r$. Es wird nun eine Kante {u,v} aus E' ausgewählt, wobei u zu keiner weiteren Kante aus E' inzident ist. Hier gibt es zwei verschiedene Fälle:
Steht v im Konflikt mit A|B, verschwindet durch die NNI Operation dieser „conflicting vertex" v. Ist v jedoch kein conflicting vertex, so ist es möglich eine NNI Operation an {u,v} durchzuführen, bei der eine mit A|B im Konflikt stehende Kante verschwindet. Durch $|E'|+|V'|$ -faches Wiederholen dieser Vorgehensweise entsteht einen Baum T' mit $d_{NNI}(T,T')$, der A|B enthält. □

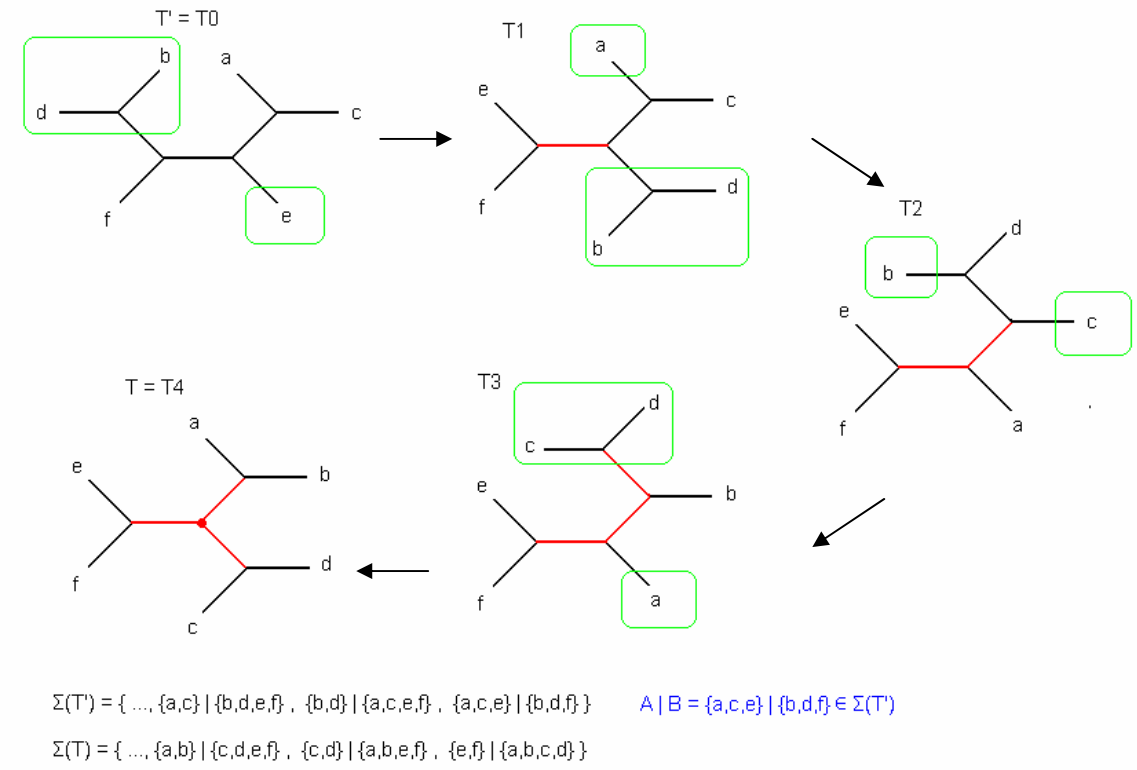

Fig. 20: Beispiel zur graphischen Charakterisierung der Splits in der NNI Nachbarschaft:

In dieser Abbildung ist die Folge von Bäumen von T'(=T_0) nach T(=T_4) dargestellt. Jeder Baum T_j unterscheidet sich von T_{j+1} durch eine NNI Operation und umgekehrt. Der Split A|B ist Element aus Σ(T'). Alle roten Kanten bzw. Knoten stehen im Konflikt mit A|B. Die grün eingerahmten Telbäume werden jeweils im nächsten Schritt durch eine NNI Operation vertauscht.

2.4.3. Splits in der Subtree Prune and Regraft und in der Tree Bisection Reconnection Nachbarschaft

Wie bereits gezeigt wurde, ist jede NNI Operation auch eine SPR Operation und jede SPR Operation ist auch eine TBR Operation. Es gilt:

$$d_{NNI}(T_1, T_2) \geq d_{SPR}(T_1, T_2) \geq d_{TBR}(T_1, T_2)$$

Dabei gilt in einigen Fällen strenge Ungleichheit.
Für alle X-Bäume T_1 und T_2 mit |X| = 4 gilt $d_{NNI}(T_1, T_2) = d_{SPR}(T_1, T_2) = d_{TBR}(T_1, T_2) = 1$, da es für einen vierblättrigen X-Baum nur drei verschiede Topologien gibt, die jeweils alle durch nur eine NNI, SPR oder TBR Operation ineinander zu überführen sind.
Die Split Nachbarschaften stehen daher in folgendem Verhältnis:

$$S_{NNI}(T,r) \subseteq S_{SPR}(T,r) \subseteq S_{TBR}(T,r)$$

Ziel ist es nun zu zeigen, dass die Split Nachbarschaften von SPR und TBR identisch und beträchtlich größer sind als die NNI Nachbarschaft. Die Gleichheit der SPR und der TBR Nachbarschaft soll im Folgenden über die Parsimony Länge eines Charakters erklärt werden.

Ein *binärer Charakter* für die Menge X ist eine Funktion $\chi : X \rightarrow \{0,1\}$. Eine *Erweiterung* von χ auf einen phylogenetischen X-Baum ist eine Funktion χ': V(T) → {0,1}, so dass die Restriktion von χ' auf X gleich χ ist (V(T) ist die Menge aller Knoten im X-Baum T, X ist die Menge seiner Blätter.).
Die *Länge* von χ' wird mit $l'_T(\chi')$ bezeichnet und enthält die Anzahl von Kanten {u,v}, für die gilt: $\chi'(u) \neq \chi'(v)$. Die *Parsimony Länge* von χ' auf T wird mit $l_T(\chi)$ bezeichnet und ist das Minimum von $l'_T(\chi')$ über alle χ' von χ.

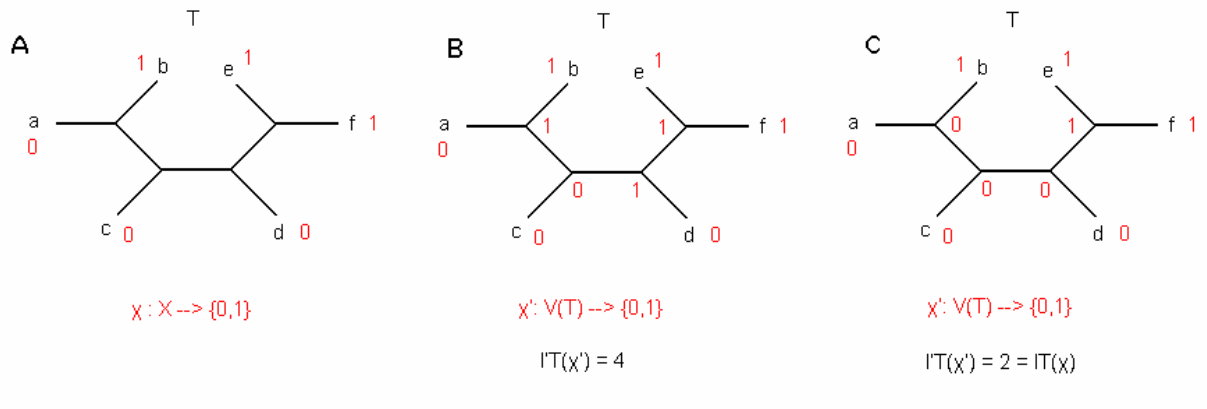

Fig. 21: Beispiel: T ist ein X-Baum. In Bild A ist eine Funktion $\chi : X \rightarrow \{0,1\}$ dargestellt. In Bildern B und C ist diese Funktion χ auf die gesamte Knotenmenge von T erweitert → Funktion χ'. χ' in Bild C stellt zusätzlich eine Erweiterung von χ minimaler Länge dar. Daher gilt für C: $l'_T(\chi') = l_T(\chi)$.

Um die Gleichheit der SPR und der TBR Nachbarschaft zeigen zu können, wird zuvor noch eine Eigenschaft benötigt.

Lemma 2.7.:
T‘ unterscheidet sich von T durch eine TBR Operation. Für jeden Charakter χ gilt: $l_T‘(\chi) \leq l_T(\chi)+1$.

Beweis:
Sei $\chi‘$ eine Erweiterung von χ auf den Baum T von minimaler Länge, so dass gilt: $l‘_T(\chi‘) = l_T(\chi)$. Es ist nun zu zeigen, dass die Länge $l_T(\chi)$ durch eine TBR Operation um höchstens eins erhöht wird. Eine TBR Operation wird in zwei Schritten ausgeführt.
Im ersten Schritt wird eine Kante {u,v} aus T entfernt und alle Knoten vom Grad zwei werden unterdrückt. Angenommen A|B der Split ist, der mit Kante {u,v} assoziiert ist. Dann entstehen durch diesen Schritt zwei Teilbäume T_A und T_B mit der jeweiligen Blattmenge A bzw. B. Seien χ'_A und χ'_B die Restriktionen von χ' auf $V(T_A)$ bzw. auf $V(T_B)$. Dann gilt

$$l'_{T_A}(\chi'_A) + l'_{T_B}(\chi'_B) \leq l'_T(\chi')$$

da das Entfernen einer Kante und das Unterdrücken von Knoten die Länge nicht erhöhen kann.
Im zweiten Schritt der TBR Operation werden die Teilbäume T_A und T_B wieder verbunden.
Fall a) Einer der beiden Teilbäume besteht nur aus einem Knoten. In diesem Fall kann die Länge von χ' durch das Verbinden der beiden Teilbäume um höchstens eins verlängert werden.
Fall b) Beide der beiden Teilbäume bestehen aus mehr als einem Knoten. Auf der Kante $\{u_A, v_A\}$ und der Kante $\{u_B, v_B\}$ wird jeweils ein neuer Knoten x_A bzw. x_B eingefügt. x_A und x_B werden mit einer neuen Kante verbunden. Der Baum T’ entsteht. Wird $\chi'(x_A) = \chi'(u_A)$ und $\chi'(x_B) = \chi'(u_B)$ gesetzt, so hat die Erweiterung von χ auf V(T’) höchstens die Länge $l_T(\chi) + 1$.

□

Für das Theorem 2.8. wird folgende Definition einer Funktion χ benötigt.
Für jeden Split A|B von X soll gelten (mit $x \in V(T)$):

$$\chi_{A|B}(x) = \begin{cases} 1 & \text{falls } x \in A \mid B \\ 0 & \text{sonst} \end{cases}$$

Theorem 2.8.:
Sei T ein binärer phylogenetischer X-Baum and A|B sei ein Split von X. Die folgenden drei Aussagen sind äquivalent:

1) $A \mid B \in S_{SPR}(T,r)$
2) $A \mid B \in S_{TBR}(T,r)$
3) $l_T(\chi_{A|B}) \leq r + 1$

Ringschluss:

1) $\Rightarrow$ 2)
Es wurde bereits gezeigt, dass $S_{SPR}(T,r) \subseteq S_{TBR}(T,r)$. Es gilt daher:
$A \mid B \in S_{SPR}(T,r) \Rightarrow A \mid B \in S_{TBR}(T,r)$

2) $\Rightarrow$ 3)

$A|B \in S_{TBR}(T,r)$ ist gegeben. Sei $A|B \in \Sigma(T')$ und $d_{TBR}(T,T') = s \leq r$. Es gibt eine Folge von X-Bäumen mit: $T' = T_0, T_1, \ldots T_s = T$, wobei T_{i+1} aus T_i durch eine TBR Operation erhalten wird. Da $A|B \in \Sigma(T')$ gilt: $l_T'(\chi A|B) = 1$.
Mit Lemma 5.1. gilt für alle i = 1, 2, …, s : $l_{Ti}(\chi A|B) \leq l_{Ti-1}(\chi A|B) + 1$
$\Rightarrow l_{Ts}(\chi A|B) = l_T(\chi A|B) \leq s+1 \leq r+1$

3) $\Rightarrow$ 1)
$l_T(\chi_{A|B}) \leq r + 1$ ist gegeben und sei $l_T(\chi_{A|B}) \leq s+1 \leq r + 1$.
Wenn s = 0 folgt 1), da $l_T(\chi A|B) = 1$, nur wenn $A|B \in \Sigma(T)$.
Für s > 0: Sei χ' eine Erweiterung von $\chi_{A|B}$ mit minimaler Länge: es gibt drei Knoten u,v,w mit $\{u,v\} \in E(T)$, v liegt auf dem Weg von u nach w und $\chi'(v) \neq \chi'(u) = \chi'(w)$
Es wird eine SPR Operation durchgeführt: Entfernen der Kante {u,v}, Einfügen eines neuen Knotens x an einer zu w adjazenten Kante, Hinzufügen der Kante {u,x}, Setzen von $\chi'(x) = \chi'(u)$. χ' des neuen Baumes hat nach diesem Schritt die Länge s.
Nach s Durchläufen dieser Art wird T' mit $A|B \in \Sigma(T')$ und $d_{SPR}(T, T') = s$ erhalten. □

Charleston und Steel geben in ihrer Veröffentlichung (M. Charleston and M. Steel, Five surprising properties of parsimonously colored trees) eine genaue Formel für die Anzahl binärer Funktionen mit der Parsimony Länge k in einem binären phylogenetischen X-Baum an. Mit dieser Formel zusammen mit Theorem 2.8. ist es möglich die Anzahl der Splits in $S_{SPR}(T,r)$ und $S_{TBR}(T,r)$ zu bestimmen.

Korollar 2.9.:
Sei T ein binärer phylogenetischer X-Baum und sei n = |X|. Es gilt:

$$|S_{SPR}(T,r)| = |S_{TBR}(T,r)| = \sum_{k=1}^{r+1} \left(\binom{n-k}{k} + \binom{n-k-1}{k} \right) 2^{k-1}$$

2.5. Zusammenfassung

Über die Größe der Splitnachbarschaften ist folgendes festzuhalten: Die Anzahl der Splits in $S_{RF}(T,r)$ und die Anzahl der Splits in $S_{NNI}(T,r)$ wachsen linear in n mit n = |X| für ein festes r. Weiterhin gilt: $S_{NNI}(T,r) \subseteq S_{RF}(T,r)$. Die Split Nachbarschaften von SPR und TBR sind identisch und wachsen bei beschränktem r exponentiell in der Anzahl der Blätter des X-Baumes. Die SPR und die TBR Nachbarschaft sind beträchtlich größer als die NNI Nachbarschaft. Für diese drei Nachbarschaften gilt folgende Beziehung: $S_{NNI}(T,r) \subseteq S_{SPR}(T,r) \subseteq S_{TBR}(T,r)$.

3. A Classification of Consensus Methods for Phylogenetics

3.1. Einleitung

Consensus Baum Methoden, sind Methoden, die aus einer Sammlung von phylogenetischen Bäumen auf dem gleichen Taxaset einen einzelnen „repräsentativen“ Baum, der Consensus Baum, erstellen. Hierbei stellt sich natürlich die Frage, wie am sinnvollsten Informationen von miteinander konkurrierenden Bäumen in einem Baum vereinigt werden können. Von

biologischer Seite gibt es hierfür keine eindeutige Lösung. Die meisten Consensus Methoden gehen so vor, dass sie gemeinsame Substrukturen aus den Eingabebäumen identifizieren und diese im Ausgabebaum wiedergeben. Konfliktreiche Regionen werden auf diese Weise ausgeschlossen.
Es gibt eine beachtliche Diskussion über den Nutzen und den Missbrauch von Consensus Methoden: Hierbei ist es wichtig, die Art der Interpretation von Consensus Bäumen zu beachten. Consensus Methoden werden nicht nur als Hilfsmittel zur Repräsentation sondern auch als Werkzeug, um phylogenetische Schlussfolgerungen zu ziehen, genutzt. Dieser Ansatz ist problematisch, da sich die meisten Methoden hauptsächlich auf kombinatorische Eigenschaften stützen. Schlussfolgerungen aus phylogenetischen Betrachtungen spielen bei der Arbeitsweise der Consensus Methoden höchstens eine geringe Rolle, wobei sich auch die Frage stellt in wieweit es überhaupt möglich ist diese Kriterien gewinnbringend einzusetzen. Dennoch ist es möglich Consensus Methoden als Werkzeug für phylogenetische Schlussfolgerungen zu nutzen. Dies sollte jedoch nur im Zusammenhang mit einer bestimmten Zielsetzung, einem Modell oder Paradigma geschehen. Im einfachsten Fall könnte eine Zielsetzung so aussehen, dass konfliktreiche Baumregionen oder übereinstimmende Regionen in den Eingabebäumen ermittelt werden sollen.
Allgemein lässt sich sagen, dass Standard Consensus Methoden sehr gut dafür geeignet sind Gemeinsamkeiten und Differenzen zwischen Eingabebäumen zu bestimmen. Für verschiedene Zielsetzungen sind jeweils auch unterschiedliche Consensus Methoden besser bzw. schlechter geeignet.

3.2. Terminologie

In diesem Teil werden ungewurzelte und gewurzelte phylogenetische X-Bäume betrachtet.
Gewurzelte phylogenetische X-Bäume:

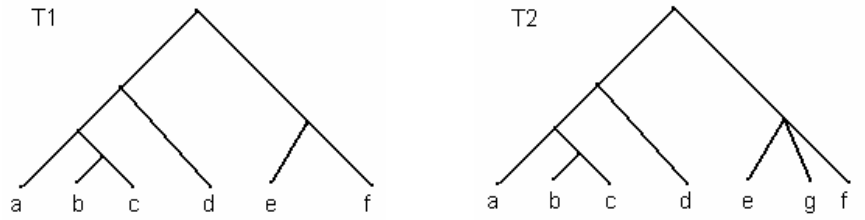

Fig. 22: Die Darstellung zeigt zwei gewurzelte phylogenetische Bäume T_1 mit dem Taxaset X = {a,b,c,d,e,f} und T_2 mit dem Taxaset X = {a,b,c,d,e,f,g}. Nur T_1 ist ein binärer Baum.

Eine *Gruppe* ist eine Teilmenge der Menge aller Taxa. Eine Gruppe ist monophyletisch, wenn alle Nachkommen des jüngsten gemeinsamen Vorfahrens aller Gruppenmitglieder Teil dieser Gruppe sind. *Monophyletische Gruppen* werden auch als *Cluster* eines Baumes bezeichnet. Cluster sind die gewurzelten Äquivalente zu Splits.
Ein *gewurzeltes Tripple ab|c* mit $a,b,c \in X$ bezeichnet eine Gruppierung von a und b relativ zu c. In Figur 22 wären z.B. bc|a und cd|e gewurzelte Tripple. r(T) ist die Menge aller Tripple im Baum T.
Eine Sammlung von Gruppen C ist kompatibel, wenn es einen gewurzelten Baum T gibt, für den jede Gruppe ein Cluster von T bildet. Für jedes Cluster A und B in C gilt: $A \subseteq B$ oder $B \subseteq A$ oder $A \cap B = \emptyset$.
Eine *Restriktion von T auf X* bedeutet, dass jedes Cluster A aus T durch die Schnittmenge $A \cap X$ ersetzt wird. Eine solche Restriktion wird mit $T|_X$ bezeichnet.

Ein Baum T *verfeinert* einen Baum T‘, wenn jedes Cluster / jeder Split aus T‘ auch in T enthalten ist.
Ein gewurzelter Baum ist binär, wenn jeder innere Knoten genau zwei Kinder hat. Für ungewurzelte wie auch für gewurzelte binäre Bäume gilt, dass diese nicht mehr verfeinert werden können.

3.3. Vorstellung verschiedener Consensus Methoden

3.3.1. Übersicht von Consensus Methoden

Consensus Methoden basierend auf Splits & Clustern:
Strict Consensus Tree
Majority Rule Tree
Loose Consensus Tree
Greedy Consensus Tree
Nelson Page & Asymmetric Median Consensus Tree

Cluster Schnittmengen Methoden
Adams Consensus Tree
Cluster Height Methods

Consensus Methoden basierend auf Teilbäumen
Local Consensus Tree
Prune & Regraft Tree
Q* & R* Consensus Tree

Consensus Methoden basierend auf Recoding
Matrix Repräsentation mit Parsimony
Average Consensus Tree
Buneman Consensus Tree

Im Folgenden werden alle kursiv gedruckten Methoden beschrieben. Bei diesen Methoden ist es wichtig, dass alle Eingabebäume das gleiche Taxaset besitzen. Der Ausgabebaum ist ebenfalls auf dem gleichen Taxaset, wie die Eingabebäume.

3.3.2. Der Strict Consensus Tree

Gegeben eine Sammlung von Eingabebäumen enthält der Strict Consensus Tree genau alle Splits bzw. Cluster, die in jedem Eingabebaum vorkommen.

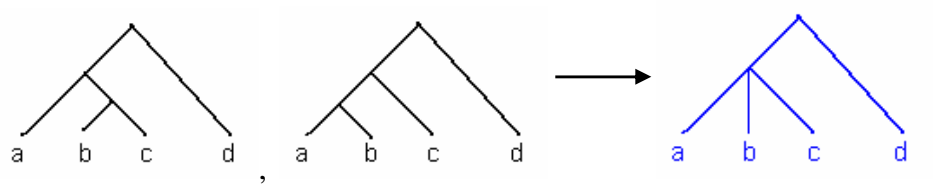

Fig. 23: Beispiel: Eingabebäume $\mathcal{T}$ = { ((a,(b,c)),d) , (((a,b),c),d) },
Strict Consensus Tree (blau dargestellt): ((a,b,c),d),
Die beiden Cluster {a,b,c} und {a,b,c,d} sind die einzigen, die in beiden Eingabebäumen vorkommen.

3.3.3. Der Majority Rule Tree

Gegeben eine Sammlung von Eingabebäumen enthält der Majority Rule Tree genau die Splits bzw. Cluster, die in mehr als der Hälfte der Eingabebäume vorkommen. Durch diese Eigenschaft verfeinert der Majority Rule Tree den Strict Consensus Tree.

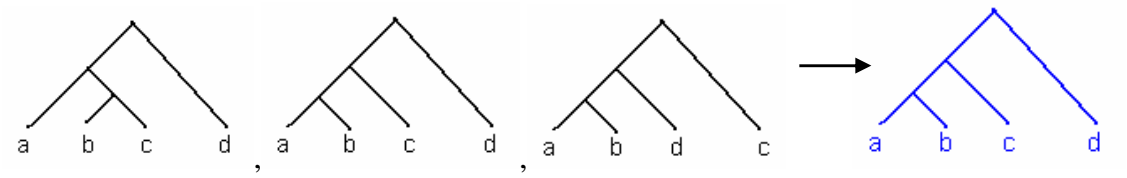

Fig. 24: Beispiel: Eingabebäume $\mathcal{T}$ = { ((a,(b,c)),d) , (((a,b),c),d) , (((a,b),d),c) },
Majority Rule Tree (blau dargestellt): (((a,b),c),d)
Die Cluster {a,b}, {a,b,c} und {a,b,c,d} kommen jeweils in mindestens zwei Eingabebäumen vor.

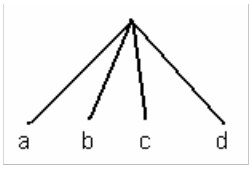

Fig. 25: Darstellung des Strict Consenus Tree für die Eingabebäume aus Fig.24.

Barthélemy und McMorris (J.P. Barthélemy and F.R. McMorris and R.C. Powers, Dictatorial consensus functions on n-trees) haben gezeigt, dass der Majority Rule Tree auch ein Median Tree ist. Sei $d(T_1,T_2)$ (entspricht $d_{RF}(T_1,T_2)$) die symmetrische Differenz Distanz zwischen zwei Bäumen T_1 und T_2, dann minimiert der Majority Rule Tree für $\mathcal{T} = \{T_1, \ldots T_k\}$ die folgende Funktion:

$$d(T,\mathcal{T}) = \sum_{i=1}^{k} d(T,T_i)$$

Der Majority Tree ist damit ein Median von $\mathcal{T}$ unter Verwendung der symmetrischen Distanz Metrik.

3.3.4. Der Loose Consensus Tree

Der Loose Consensus Tree enthält gegeben eine Sammlung von Eingabebäumen genau alle Splits bzw. Cluster, die mit jedem Baum aus T kompatibel sind. Mit dieser Eigenschaft verfeinert der Loose Consensus Tree den Strict Consensus Tree.

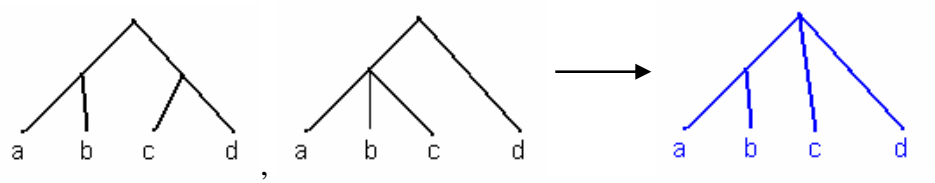

Fig. 26: Beispiel: Eingabebäume $\mathcal{T}$ = { ((a,b),(c,d)) , ((a,b,c),d) },
Loose Consensus Tree (blau dargestellt): ((a,b),c,d),

Die Cluster des Loose Consensus Tree sind mit allen Eingabebäumen kompatibel.

Sei T ein binärer ungewurzelter phylogenetischer X-Baum und A|B ein Split, der mit Σ(T) kompatibel ist, dann ist $A \mid B \in \Sigma(T)$. T kann nicht mehr weiter verfeinert werden. Es folgt: Ist $\mathcal{T}$ eine Sammlung von binären phylogenetischen ungewurzelten X-Bäumen, so ist der Loose Consensus Tree mit dem Strict Consensus Tree identisch. Das gleiche Vorgehen lässt sich auf den gewurzelten Fall anwenden.
Queiroz (A. de Queiroz, For consensus (sometimes)) hat herausgefunden, dass es Loose Consensus Trees gibt, die Splits bzw. Cluster enthalten, die in keinen der Eingabebäume vorkommen.

3.3.5. Der Greedy Consensus Tree

Der Greedy Consensus Tree ist eine Verfeinerung des Majority Rule Tree. Zusätzlich zu den Splits bzw. Clustern, die im Majority Rule Tree enthalten sind, wird hier das Einfügen weiterer Splits bzw. Cluster erlaubt. Diese Strategie wird auch bei PHYLIP und PAUP angewendet. Die Auswahl der Splits bzw. Cluster erfolgt mit Hilfe einer „greedy strategy" (gierige Strategie). Es wird eine Liste aller Splits/Cluster, die in den Eingabebäumen vorkommen, in der Reihenfolge ihrer Häufigkeiten (häufigste zu Beginn) erstellt. Als Nächstes erfolgt ein schrittweiser Aufbau einer kompatiblen Menge von Splits/Clustern. Dabei wird die Liste durchlaufen. Jeder Split / jedes Cluster wird in S aufgenommen, wenn er / es mit allen in S bereits enthaltenen Splits / Clustern kompatibel ist. Bei dieser Strategie ergibt sich allerdings das Problem der willkürlichen Ungleichbehandlung von Splits/Clustern, die mit gleicher Häufigkeit vorkommen. Der Zufall entscheidet, welcher Split / welches Cluster von zwei oder mehr Splits / Clustern der gleichen Häufigkeit um eine Position weiter vorne in der Liste steht. Sind diese Splits / Cluster nicht miteinander kompatibel, so wird durch diese Wahl entschieden wer im Greedy Consensus Tree enthalten ist und wer nicht.

Theorem 2.6.:
Die Greedy Selection Methode erzeugt einen Consensus Baum, der den Majority Rule Tree und den Loose Consensus Tree verfeinert.

Beweis (für ungewurzelte Bäume)**:**
Majority Rule Tree: Alle Splits, die in mehr als der Hälfte der Eingabebäume enthalten sind müssen kompatibel sein, denn: Seien jeweils A|B und C|D Splits, die in mehr als der Hälfte aller Eingabebäume enthalten sind. In diesem Fall muss es einen Baum $T_i \in \mathcal{T}$ geben, der A|B und C|D enthält. A|B und C|D sind somit kompatibel.
Die Splits, die in mehr als der Hälfte der Eingabebäume enthalten sind (Splits des Majority Rule Trees), stehen in der ersten „Hälfte" der Liste. Sie sind kompatibel und werden deshalb in der greedy strategy alle in S aufgenommen.
Loose Consensus Tree: Jeder Split A|B der mit allen Eingagebäumen kompatibel ist muss in S aufgenommen (es kann in keinem Baum einen Split geben, der mit A|B inkompatibel ist).

□

3.3.6. Der Adams Consensus Tree

Der Adams Consensus Tree gehört zu den Cluster Schnittmengen Methoden und ist die erste Consensus Methode, die für Bäume entwickelt wurde (1972). Cluster Schnittmengen Methoden sind nur für gewurzelte Bäume definiert.

Folgende Definitionen sind zur Anwendung des Algorithmus für den Adams Consensus Tree notwendig: $\Pi_1, \Pi_2, \ldots \Pi_k$ sind k viele Partitionen auf der Menge X. Für das Produkt Π dieser k vielen Partitionen gilt, dass $a \in X$ und $b \in X$ mit $a \neq b$ nur in dem gleichen Block von Π enthalten sind, wenn sie in allen Partitionen Π_i (für alle i) im gleichen Block vorkommen. Beispiel: Π_1 = ab|cde und Π_2 = ac|bde , es folgt Π = a|b|c|de. Maximale Cluster eines Baumes T sind die größten Cluster in T, die nicht alle Elemente aus X enthalten. Die Maximale Cluster Partition von T ist die Partition, die die maximalen Cluster von T enthält. Die Maximale Cluster Partition wird mit $\Pi(\mathcal{T})$ bezeichnet.

Der Adams Consensus Tree entsteht durch rekursives Aufstellen von Partitionen Π von $\mathcal{T}$ und Restriktionen der Bäume in $\mathcal{T}$:

Procedure AdamsTree($T_1, \ldots T_k$)
If T_1 enthält nur ein Blatt
→ return T_1
else erzeuge $\Pi(\mathcal{T})$, das Produkt von $\Pi(T_1), \ldots \Pi(T_k)$
For jeder Block B von $\Pi(\mathcal{T})$ do
AdamsTree ($T_1|_B, \ldots T_k|_B$)
Verbinde die Wurzeln dieser Bäume mit einem neuen Knoten v
return diesen Baum
end

Beispiel:

$\mathcal{T}$ = {T_1, T_2} mit T_1: (((a,b),c),d) und T_2: (((c,b),a),d)
$\Pi(T1)$ = abc|d
$\Pi(T2)$ = cba|d
→ $\Pi(\mathcal{T})$ = abc|d

$T_1|_A$: ((a,b),c)
$T_2|_A$: ((c,b),a)
$\Pi(T_1|_A)$ = ab|c
$\Pi(T_2|_A)$ = cb|a
→ $\Pi(\mathcal{T}|_A)$ = a|b|c

Adams Consensus Tree von $\mathcal{T}$: ((a,b,c),d)

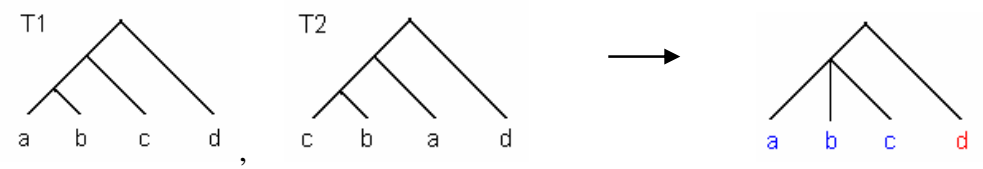

Fig. 27: Darstellung der Eingabebäume und des Adams Consensus Tree des oben genannten Beispiels.

3.4. Klassifikation der vorgestellten Consensus Methoden

Eine kleine Auswahl von Consensus Methoden ist vorgestellt worden. Nun stellt sich die Frage welche Art von Information in welchem Ausgabebaum der verschiedenen Methoden

enthalten ist und wie diese interpretiert werden kann. In diesem Zusammenhang ist es sinnvoll eine Klassifikation von Consensus Methoden zu erstellen.
Ein Kriterium zur Klassifikation von Consensus Methoden stellt die Art und Menge des zusätzlichen Informationsgehalts im Vergleich zum Strict Consensus Tree dar.

In der Figur 27 sind weitere Merkmale aufgezeigt, mit denen eine Klassifikation von Consensus Methoden erfolgen kann.
Co-Pareto Eigenschaft von Splits / Clustern: Jeder Split / jedes Cluster des Consensus Baumes kommt in mindestens einem Eingabebaum vor.
Co-Pareto Eigenschaft von gewurzelten Trippeln: Jedes gewurzelte Tripple des Consensus Baumes kommt in mindestens einem Eingabebaum vor.
Pareto Eigenschaft von gewurzelten Trippeln: Jedes gewurzelte Tripple, das in allen Eingabebäumen enthalten ist, kommt auch im Consensus Baum vor.

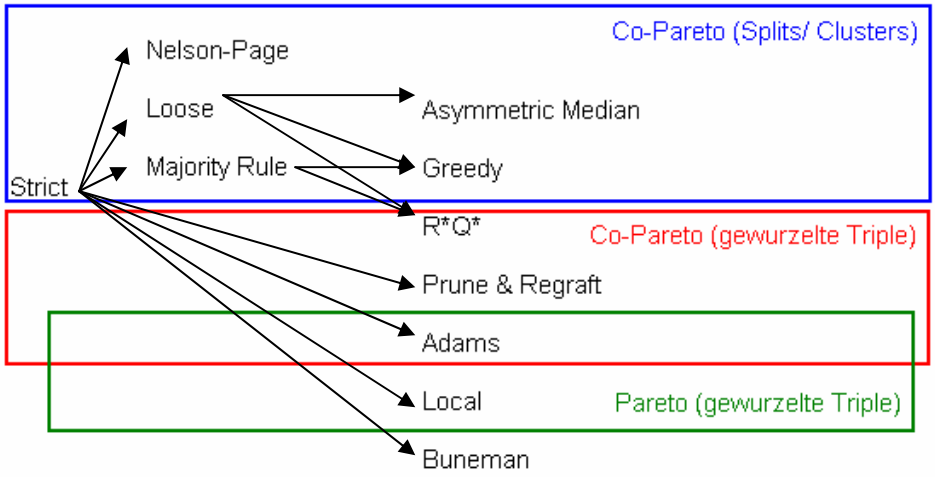

Fig. 28: Klassifikation von Consensus Methoden. Ein Pfeil von Methode A zu Methode B impliziert, dass jeder Split / jedes Cluster aus einem Consensus Baum der Methode A von $\mathcal{T}$ auch in einem Consensus Baum der Methode B von $\mathcal{T}$ enthalten ist.

3.5. Subtrees & Supertrees

Bisher haben alle Eingabebäume und Ausgabebäume folgende Bedingungen erfüllt:

1) Alle Eingabebäume besitzen das gleiche Taxaset.
2) Die Blätter des Consensus Baumes bilden das gleiche Taxaset wie das jedes Eingabebaumes.

In der Anwendung kann es jedoch möglich sein, dass einer der beiden Fälle nicht erfüllt ist.

3.5.1. Subtrees

Zur Erstellung von Subtrees muss die Bedingung 2) nicht unbedingt erfüllt sein. In den Ausgabebäumen können folglich bestimmte Taxa weggelassen werden. In manchen Fällen macht es Sinn Taxa im Ausgabebaum wegzulassen. Ein Beispiel hierfür ist, wenn ein einzelnes, weit entferntes Taxon in sonst identischen Eingabebäumen in verschiedenen Positionen erscheint. Die meist genutzte Subtree Methode ist der Agreement Subtree. Der Agreement Subtree T für $\mathcal{T} = \{T_1, \ldots T_k\}$ auf einer Teilmenge X lautet: $T = T_i|_X$ für alle $i = 1, \ldots k$.

3.5.1. Supertrees

Supertrees können aus Eingabebäumen erstellt werden, die verschiedene Taxasets beinhalten. Es gibt einige wichtige Situationen, in denen Supertrees Anwendung finden: Sie werden dafür genutzt die Ergebnisse aus verschiedenen Datenmengen, die jeweils Informationen von unterschiedlichen Taxasets enthalten, und sie sind ein Teil von divide & conquer Strategien, die zum Aufbau von großen Phylogenien genutzt werden.
Einige Consensus Methoden wurden bereits auf das Supertree Problem angepasst (Strict, Adams, …). Wilkinson & Thorley (M. Wilkinson and J. Thorley, reduced consensus supertrees) haben eine Methode entwickelt, die über Teilmengen aus der Gesamtmenge von Taxa einen Supertree erzeugt, eine Subtree-Supertree Methode.

3.6. Zusammenfassung

Es wurden einige Consensus Methoden mit verschieden Eigenschaften besprochen. Die aufgezeigten Vorteile bzw. Nachteile machen deutlich, dass es nötig ist verschiedene Consensus Methoden für unterschiedliche Zielsetzungen anzuwenden.

4. Quellenangabe

Splits in the Neighborhood of a Tree,
David Bryant, DIMACS Series in Discrete Mathematics and Theoretical Computer Science

A Classifikation of Consensus Methods for Phylogenetics,
David Bryant, Annals of Combinatorics, 2003