Algorithmische Bioinformatik

Character-basierte Phylogenie

Ulf Leser

Wissensmanagement in der Bioinformatik

Ziel dieser Vorlesung

- Grundprinzip der Maximum Parsimony Idee erkennen und bewerten können
- Verfahren dazu kennenlernen
- Unterschiede zu distanzbasierten Verfahren verstehen

Inhalt dieser Vorlesung

- Characterbasierte Verfahren
- Perfect Phylogeny
- Maximum Parsimony
  - Fitch / Sankoff‘s Algorithmus
    - Fitch, W. M. (1971). "Toward defining the course of evolution: minimum change for a specified tree topology." *Systematic Zoology* **20**: 406-416.
    - Sankoff, D. D. (1975). "Minimal mutation trees of sequences." *SIAM Journal on Applied Mathematics* **28**(35-42)
  - Heuristiken

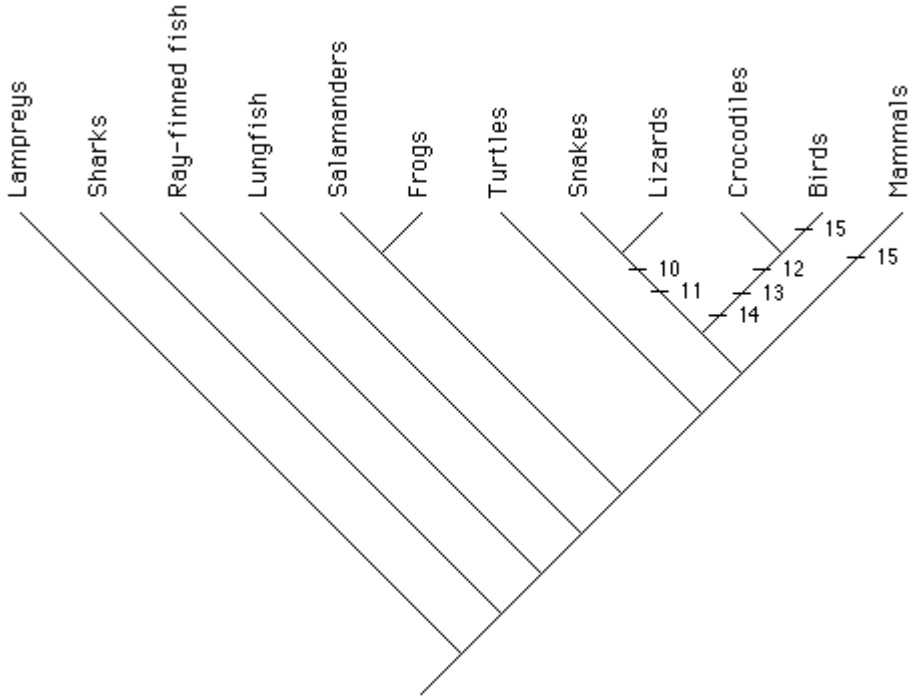
Distanz versus Zeichen

- Distanzbasierte Algorithmen abstrahieren von einzelnen Zeichen und basieren nur auf dem Abstand von Taxa
- Character-basierte Verfahren betrachten die Entwicklung jedes einzelnen „Characters“ (oder „Site“)
  - Nuklein- oder Aminosäure
  - Morphologische Eigenschaften
  - Vorhandensein / Abwesenheit bestimmter Gene/Funktionen
  - ...
- Zustände der Character müssen in einem Abstammungs-verhältnis stehen
- Wahl der Character beeinflusst das Ergebnis erheblich
  - Eine „korrekte“ Wahl gibt es nicht – erheblicher Freiraum

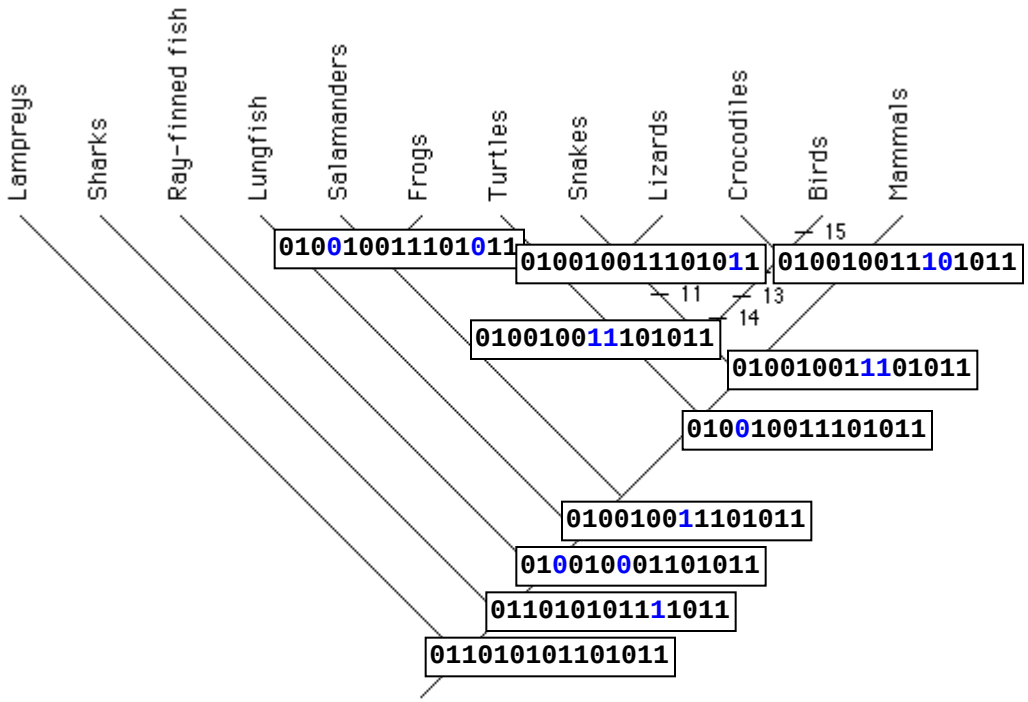
Character	Lampreys	Sharks	Teleosts	Lungfish	Frogs	Salamanders	Turtles	Lizards	Snakes	Crocodiles	Birds	Mammals
1 Internal skeleton	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
2 Jaws	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
3 Ossified skeleton	no	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
4 Internal nostrils	no	no	no	yes	yes	yes	yes	yes	yes	yes	yes	yes
5 Atrial septum	no	no	no	yes	yes	yes	yes	yes	yes	yes	yes	yes
6 Four limbs	no	no	no	no	yes	yes	yes	yes	yes	yes	yes	yes
7 Teeth pedicillate	no	no	no	no	yes	yes	no	no	no	no	no	no
8 Amniotic egg	no	no	no	no	no	no	yes	yes	yes	yes	yes	yes
9 Temporal fenestrae	none	none	none	none	none	none	none	two	two	two	two	one
10 Hemipenes	no	no	no	no	no	no	no	yes	yes	no	no	no
11 Suspensorium streptosylous	no	no	no	no	no	no	no	yes	yes	no	no	no
12 Antorbital fenestrae	no	no	no	no	no	no	no	no	no	yes	yes	no
13 Lateral fenestrae ossified	no	no	no	no	no	no	no	no	no	yes	yes	no
14 Gizzard	no	no	no	no	no	no	no	no	no	yes	yes	no
15 Homeothermy	no	no	no	no	no	no	no	no	no	no	yes	yes
16 Body covering	scale-less	dermal denticles	dermal scales	dermal scales	smooth epidermis	smooth epidermis	epidermal scales	epidermal scales	epidermal scales	epidermal scales	feathers	hair

Quelle: Morrison, Phylogenetic-Tree Building, 1996

Phylogenetischer Baum



Bester phylogenetischer Baum



Der Baum mit den wenigsten Änderungen

Maximum Parsimony

- Generelles Prinzip
  - Findet man in vielen Problemen / Disziplinen
  - „Occam's razor“ (William of Ockham, UK, 14 Jhdt.)
    - „*One should not increase, beyond what is necessary, the number of entities required to explain anything*“
  - KISS principle – „Keep it as simple as possible“
- In der Phylogenie: Aktuelle Zustände der Character mit so wenig evolutionären Ereignissen wir möglich erklären
- Wahl eines Baumes „objektivieren“
  - Denn es gibt sehr viele mögliche Bäume

MP und Multiple Sequence Alignment

- Für (DNA, Protein) Sequenzen: Base / AA = Character
- Welche Basen soll man miteinander vergleichen?
  - Multiple Sequence Alignment berechnen
  - Spalten mit vielen Indels werden i.d.R. ignoriert
  - Oft: Zusätzliches Löschen von Bereichen großer Variabilität
    - Lieber weniger Informationen als unsichere
- Aber – MSA benötigt oft einen evolutionären Baum
  - Möglichkeit: Iterieren

```
taxon                 ....|....10...|....20...|....30...|....40...|....50
 Fu  Nosema.40928     QFGLFSPEEIRASSVALIR--YPETLENG--VPKESGLVCAGHFGHIELVK
 Fu  Aspergillus.     QFGLFSPEEIKRMSVVHVE--YPETMDEQRQRPRTKGLECPGHFGHIELAT
 Ap  Plasmodium.3     ELGVLDPEIIKKISVCEIV--NVDIYKDG--FPREGGLYCPGHFGHIELAK
 An  Cricetulus.2     QFGVLSPDELKRMSVTEGGIKYPETTE--GGRPKLGGLECPGHFGHIELAK
 An  Homo.7434727     QFGVLSPDELKRMSVTEGGIKYPETTE--GGRPKLGGLECPGHFGHIELAK
 An  Drosophila.9     QFGILSPDEIRRMSVTEGGVQFAETME--GGRPKLGGLECPGHFGHIDLAK
 An  Celegans.133     QFGILGPEEIKRMSVAH--VEFPEVYE--NGKPKLGGLDCPGHFGHLELAK
 Fu  Spombe.54881     QFGILSPEEIRSMSVAK--IEFPETMDESGQRPRVGGLDCPGHFGHIELAK
 Pl  Athaliana.40     QFGILSPDEIRQMSVIH----VEHSETTEKGKPKVGGLECPGHFGYLELAK
 My  Ddiscoideum.     ----------------------------------------ECPGHFGHIELAK
 Rh  Porphyra.316     ----------------------------------------ECPGHFGFIELAK
 Kt  Tbrucei.1021     QFEIFKERQIKSYAVCLVEHAKSYANA----ADQSGEAECPGHFGYIELAE
 Kt  Leishmania.7     QFEVFKEAQIKAYAKCIIEHAKSYEHG----QPVRGGIECPGHFGYVELAE
```

Welche Character benutzen?

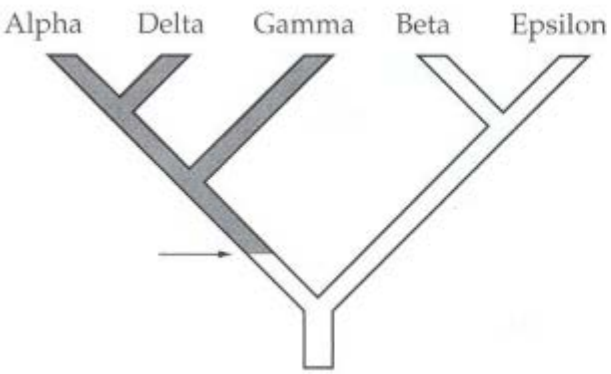
- Beschränkung auf die „informative sites“
- Homogene Spalten im MSA können entfernt werden

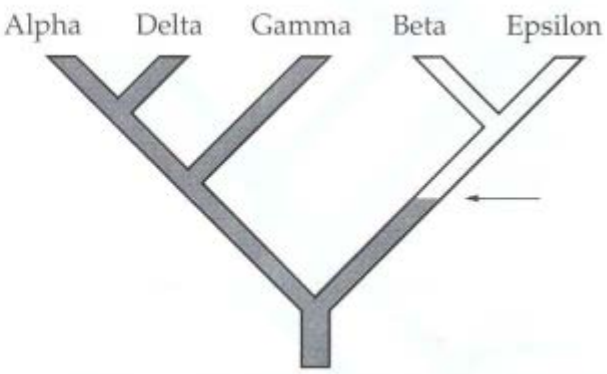
	Position								
Sequenz	1	2	3	4	5	6	7	8	9
A	A	A	G	A	G	T	G	C	A
B	A	G	C	C	G	T	G	C	G
C	A	G	A	T	A	T	C	C	A
D	A	G	A	G	A	T	C	C	G

Beispiel

- Wir nehmen $|\Sigma|=2$ an
- Fünf Spezies, 6 Character
- Betrachten wir den ersten Character
- Wie könnte der Baum aussehen?

Species	Characters 1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0



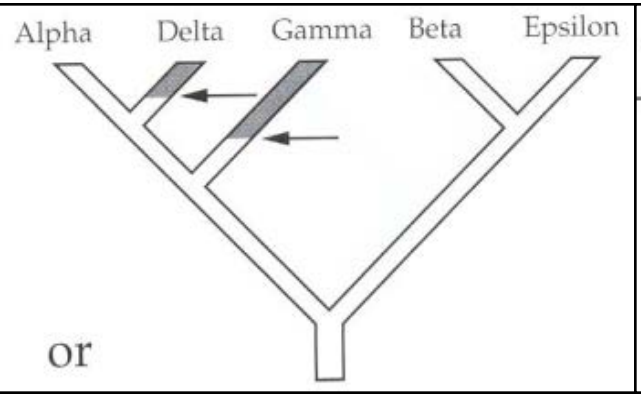


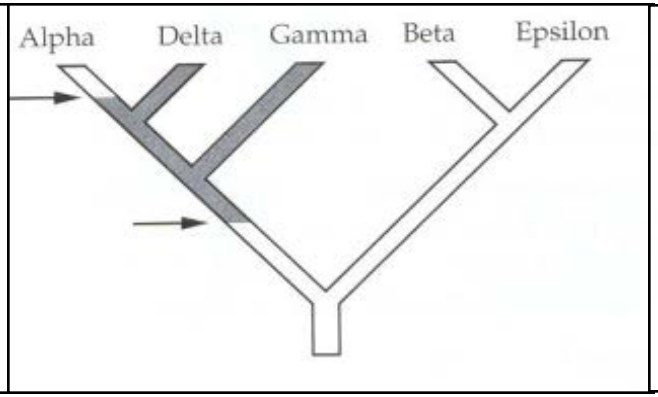
Mount: „Bioinformatics“

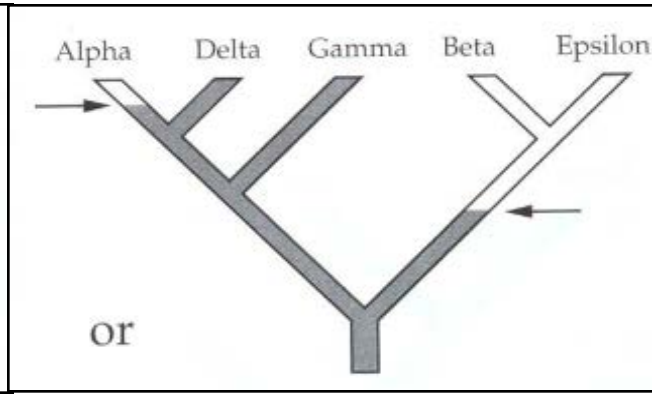
Small Parsimony

- Nehmen wir nun an, dass die Baumtopologie feststeht
  - Optimal für Char. 1
- Betrachten wir Character 2
- Welche Lösungen gibt es?

Species	Characters 1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0





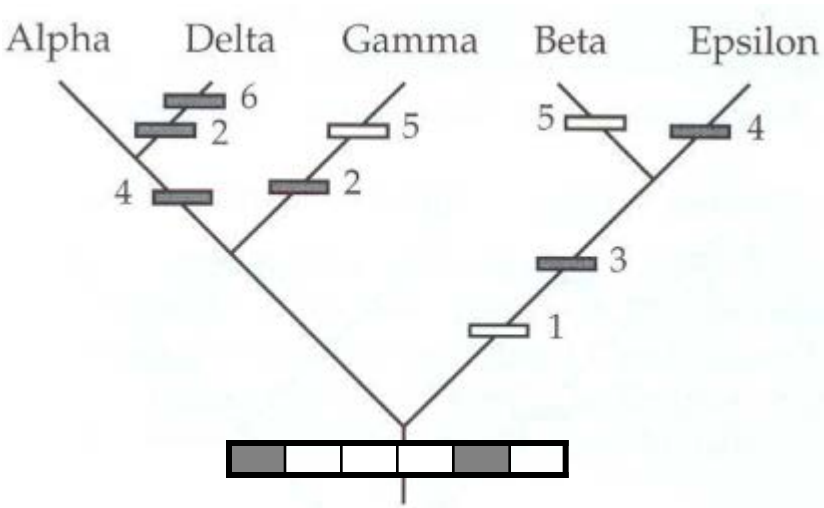


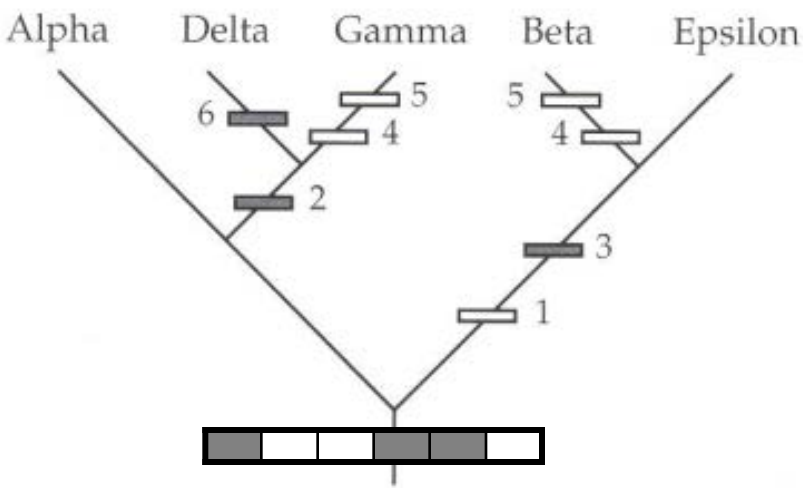
Large Parsimony

- Wenn die Topologie feststeht und man nur einen Character betrachtet, ist alles einfach
  - Small Parsimony Problem
  - Fitch/Sankoff Algorithmus schafft das ist O(n*z)
- Aber: Character haben verschiedene optimale Topologien
- Welche Topologie ermöglicht die insgesamt kleinste Menge an Zustandsänderungen?
  - Large Parsimony Problem

Beispiel

Species	Characters 1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0





- Topologie 1
- Benötigt 9 Änderungen

- Topologie 2
- Benötigt 8 Änderungen

Inhalt dieser Vorlesung

- Characterbasierte Verfahren
- Perfect Phylogeny
- Maximum Parsimony
  - Fitch / Sankoff‘s Algorithmus
  - Heuristiken

Eingeschränkteres Problem

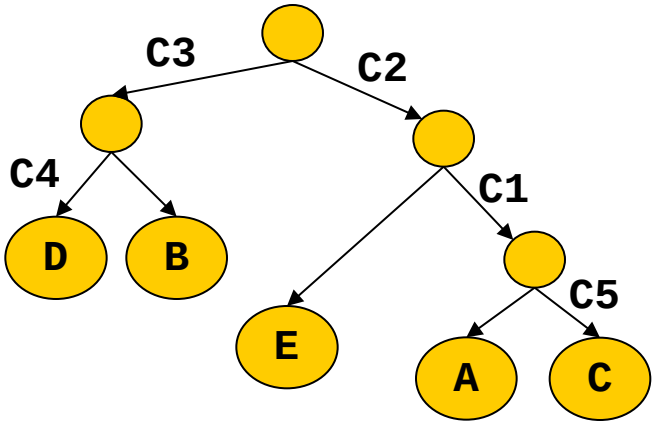
- Perfect Phylogeny
  - Modell: Alle Character sind binär (vorhanden oder nicht)
  - In der Wurzel sind alle Character „aus“
  - Jede Eigenschaft darf im Baum nur einmal erzeugt werden
- Für Sequenzen unrealistisch, für komplexere (morphologische) Eigenschaften nicht
- Überspringen wir leider
  - Es gilt: Nicht zu jeder Matrix Spezies/Character gibt es einen konsistenten Baum
  - Man kann in O(m*n) sowohl die Existenz des Baums testen als ihn auch konstruieren

Perfect Phylogeny

- Definition
  Sei D eine binäre Matrix aus n Zeilen (Arten) und m Spalten (Character) mit D[i,j]=1 gdw Art i Eigenschaft j hat. T heißt perfekter phylogenetischer Baum für D gdw
  - *T hat n Blätter, beschriftet mit den Zeilen von D*
  - *Jeder Character, der in mindestens einer Art vorhanden ist, steht an genau einer Kante von T*
  - *Für jede Art i gilt, dass die Beschriftungen der Kanten auf dem Pfad von der Wurzel zu i genau die Character sind, die i besitzt*
- Bemerkungen
  - Nicht an jeder Kante von T muss ein Character stehen, aber jeder Character muss an genau einer Kante stehen
  - Character werden nur auf einer Kante „angeschaltet“ und nie mehr „abgeschaltet“

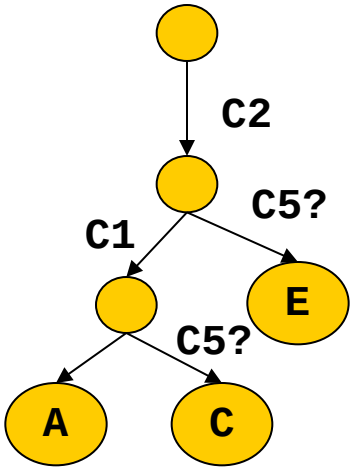
Beispiel

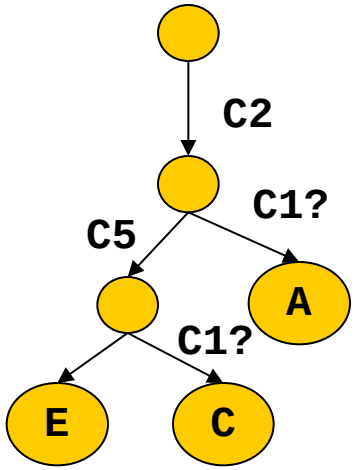
	C1	C2	C3	C4	C5
A	1	1	0	0	0
B	0	0	1	0	0
C	1	1	0	0	1
D	0	0	1	1	0
E	0	1	0	0	0



Klappt das immer?

	C1	C2	C3	C4	C5
A	1	1	0	0	0
C	1	1	0	0	1
E	0	1	0	0	1

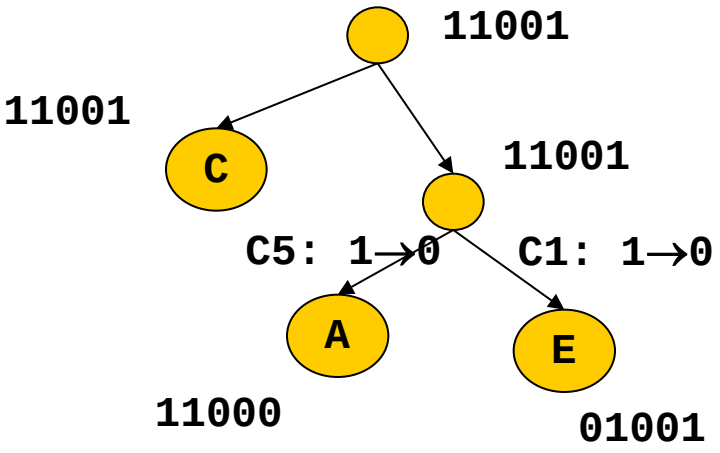




Einschränkungen aufheben

- Zu jeder binären Matrix gibt es einen phylogenetischen Baum, wenn...
  - ein Character an mehreren Stellen wechseln darf („convergent evolution“)
  - Character an der Wurzel nicht abwesend sein müssen und später auch verschwinden dürfen

	C1	C2	C3	C4	C5
A	1	1	0	0	0
C	1	1	0	0	1
E	0	1	0	0	1



Existenz perfekter phylogenetischer Bäume

- Vorarbeiten
  - Sei D eine binäre n*m Matrix und i eine Spalte (=Character)
  - Wir erzeugen eine sortierte Matrix D‘ aus D wie folgt
    - Interpretiere jede Spalte als binäre Zahl mit der sig. Stelle in Zeile 1
    - Sortiere die Spalten in D absteigend nach diesen Zahlen
  - Mit Z(i) bezeichnen wir die Menge aller Arten (Zeilen), die die Eigenschaft i besitzen (also in i eine 1 haben)
- Unverändertes Problem, da Character beliebig angeordnet werden können

	C1	C2	C3	C4	C5
A	1	1	0	0	0
B	0	0	1	0	0
C	1	1	0	0	1
D	0	0	1	1	0
E	0	1	0	0	0
	20	21	11	2	4

	C2	C1	C3	C5	C4
A	1	1	0	0	0
B	0	0	1	0	0
C	1	1	0	1	0
D	0	0	1	0	1
E	1	0	0	0	0
	21	20	11	4	2

Theorem und Beweis

- Theorem
  Sei D eine binäre Matrix. Die sortierte Matrix D‘ hat einen perfekten phylogenetischen Baum T gdw. für jedes Paar i, j von Spalten gilt:

$$Z(i) \cap Z(j) = \emptyset \text{ oder } Z(i) \subseteq Z(j) \text{ oder } Z(j) \subseteq Z(i)$$

- Beweis
  - Richtung „$\Rightarrow$“: Sei T ein perfekter phylogenetischer Baum für D‘, i,j zwei Spalten, und e_i bzw. e_j die Kanten in T, die mit i bzw. j beschriftet sind
    - Damit: Die Menge der Blätter unter e_i bzw. e_j ist exakt Z(i) bzw. Z(j)
    - Folgende Fälle können in T auftreten
      - e_i liegt auf dem Pfad von der Wurzel zu e_j. Dann gilt $Z(j) \subseteq Z(i)$
      - e_j liegt auf dem Pfad von der Wurzel zu e_i. Dann gilt $Z(i) \subseteq Z(j)$
      - e_j und e_i liegen in unterschiedlichen Pfaden. Dann gilt $Z(i) \cap Z(j) = \emptyset$

Gegenrichtung

- Richtung „⇐“: Sei D‘ wie verlangt. Wir konstruieren den perfekten phylogenetischen Baum T zu D‘ (und zeigen dabei auch die Eindeutigkeit des Baumes)
  - Seien p und q zwei Zeilen und k die rechteste Spalte mit D‘[q,k]=D‘[p,k]=1.

	…	i	…	k	k+1	…
…	…	?	?	?	?	…
p	…	?	?	1	?	…
…	…	?	?	?	?	…
q	…	?	?	1	?	…
…		…	…	…	…	…

Nicht beide 1

Gegenrichtung

- Richtung „$\Leftarrow$“
  - Wir zeigen erst: Für jede Spalte i links von k gilt D‘[p,i]=D‘[q,i]
    - Sei i eine Spalte links von k mit D‘[p,i]=1. Gemäß Voraussetzung gibt es für Z(k) und Z(i) drei Möglichkeiten:
      - $Z(k) \cap Z(i) = \emptyset$; kann nicht sein, weil $p \in Z(k) \cap Z(i)$
      - $Z(k) \subseteq Z(i)$; dann muss D‘[q,i]=1 gelten, weil D‘[q,k]=1
      - $Z(i) \subseteq Z(k)$; dann muss Z(i)=Z(k) sein, sonst kann i nicht links von k liegen
    - Weil p und q vertauschbar sind, folgt, dass D‘[p,i] und D‘[q,i] für alle i links von k entweder beide 0 oder beide 1 sind

	…	i	…	k	k+1	…
…	…	?	?	?	?	…
p	…	1	0	1	?	…
…	…	?	?	?	?	…
q	…	1	0	1	?	…
…		…	…	…	…	…

Gegenrichtung 2

- Wir haben bisher
  - $\forall i \leq k$: D‘[p,i]=D‘[q,i]
  - $\forall j > k$: Entweder D‘[p,j]=D‘[q,j]=0 oder D‘[p,j]$\neq$D‘[q,j]
- Sei q_S die Zeile von q in D‘ interpretiert als String plus „$“
  - Für alle Paare p, q gilt, dass q_S und p_S identisch sind bis zu einer Position (k) und danach niemals an derselben Stelle eine 1 haben
  - Wegen „$“ kann kein String einer Zeile Präfix eines anderen sein
- Wir bauen Keyword Tree T für alle Zeilenstrings von D‘
- T ist der perfekte phylogenetische Baum für D‘ (nach Löschen aller „$“)
  - Für alle Paare p,q ist der Pfad in T bis zu einem Knoten nach k identisch, danach können nur noch verschiedene Character im p bzw. q Ast erscheinen (keine gemeinsame 1 nach k)
  - Keine Eigenschaft vor k kann irgendwo sonst im Baum als Kantenbeschriftung auftreten

Komplexität

- Komplexität des Existenztests eines perfekten phylogenetischen Baums zu D?
  - Wir haben $O(m^2)$ Spaltenvergleiche
  - Und müssen jeweils O(n) Zeilen vergleichen (Enthaltenssein)
  - Also $O(nm^2)$
- Konstruktion des perfekten phylogenetischen Baum, wenn er existiert
  - Konstruktion des Keyword-Trees ist O(mn)
- Es gibt auch O(mn) Algorithmen zur gleichzeitigen Prüfung der Matrix und Konstruktion des Baums
  - Siehe Gusfield

Abschwächungen der Voraussetzungen

- Generalized perfect phylogeny
  - Jeder Character darf z Zustände haben
  - Jeder Zustand darf nur einmal im Baum angenommen werden (nach maximal z-1 vorherigen Wechseln)
  - Problem ist NP-vollständig (falls z beliebig)

Inhalt dieser Vorlesung

- Characterbasierte Verfahren
- Perfect Phylogeny
- Maximum Parsimony
  - Fitch / Sankoff‘s Algorithmus
  - Heuristiken

Phylogenetische Bäume

- Definition
  Gegeben eine Matrix D mit n Arten und m Character. Jeder Character kann Werte aus einer Menge Z mit |Z|=z annehmen. Ein Baum T heißt phylogenetischer Baum für D wenn gilt:
  - *T ist ein binärer gewurzelter Baum mit n Blättern, beschriftet mit Zeilen von D*
  - *Jeder innere Knoten k (inklusive der Wurzel) ist beschriftet mit einem Label, das aus einem Zustand pro Character besteht*
    - $label(k) \in Z^m$
- Bemerkungen
  - Erweiterung auf unterschiedliche Zustandsmengen pro Character wäre einfach

Suchraum

- Zahl binärer Baumtopologien (für n Taxa)

$$\frac{(2n-3)!}{2^{n-2} * (n-2)!}$$

- Jeder dieser Bäume hat n-1 innere Knoten
- Jeder innere Knoten jedes Baums kann z^m verschiedene Label haben

Maximum Parsimony

- Definition
  Sei T ein phylogenetischer Baum zu D mit Kantenmenge E. Der parsimony score S(T) von T ist definiert als:

$$S(T) = \sum_{(u,v)\in E} |\{ j \mid v_j \neq u_j \}|$$

Keine InDel, Hamming-Distanz

  Ein phylogenetischer Baum T für eine Matrix D heißt maximal parsimony, wenn S(T) der kleinstmögliche Score aller phylogenetischen Bäume von D ist.

- Bemerkungen
  - u_j ist der Zustand des Characters j in Knoten u

Wie finden wir den?

- Wir zerlegen das Problem
  - „Small parsimony“: Feste Baumtopologie
  - „Large parsimony“: Beliebige Topologie

Small Parsimony

- Definition
  Geg. eine feste Baumtopologie T und eine Taxa/Character Matrix D. Das Small Parsimony Problem (SPP) sucht nach den Labels der inneren Knoten so, dass S(T) minimal ist.

- Beobachtung: Alle Character sind unabhängig voneinander
- Damit kann man das SPP wie folgt lösen
  - Berechne die besten Label pro Character
  - Setze Knotenlabel aus einzelnen Characterlabeln zusammen
- Das n*m Problem wird auf m n*1 Probleme reduziert
  - m Probleme mit einer Matrix mit genau einem Character C

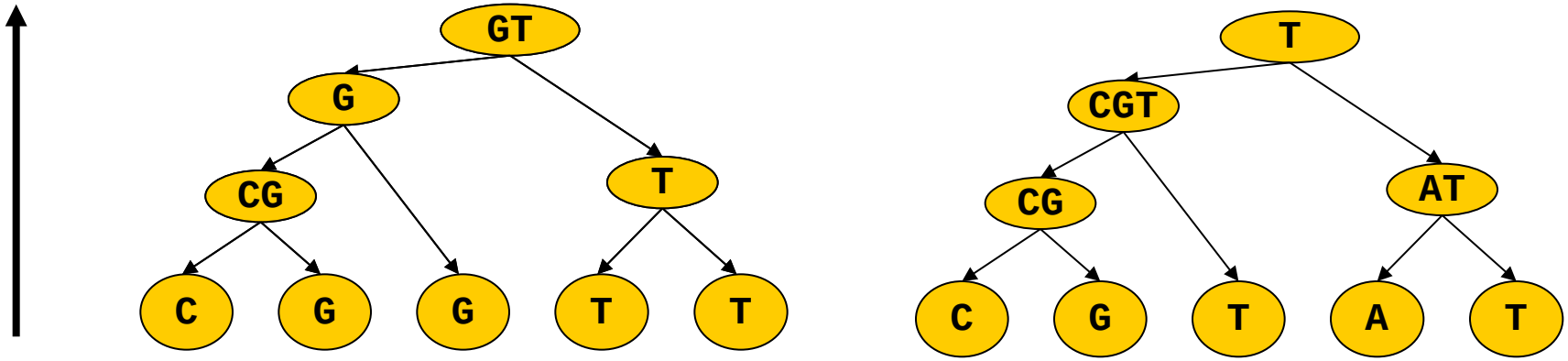
Fitch‘s Algorithmus

- Fitch, W. M. (1971). Toward defining the course of evolution: minimum change for a specified tree topology. *Systematic Zoology* 20 (4), 406-416
- Zwei Phasen
  - Wir berechnen bottom-up mögliche beste Label P pro Knoten
  - Dann legen wir top-down die Label pro Knoten fest
- Phase 1: Berechne P(k) für alle Knoten k von T
  - Wenn k ein Blatt ist, dann $P(k)=k_c$
  - Sonst habe k Kinder u und v. Dann

$$P(k) = \begin{cases} P(u) \cap P(v), & falls\, P(u) \cap P(v) \neq \emptyset \\ P(u) \cup P(v) & sonst \end{cases}$$

- Intuition: Wenn es eine Gemeinsamkeit gibt, dann nutze sie aus und propagiere nur diese nach oben; sonst nimm alle Möglichkeiten, kosten eh alle gleich viel

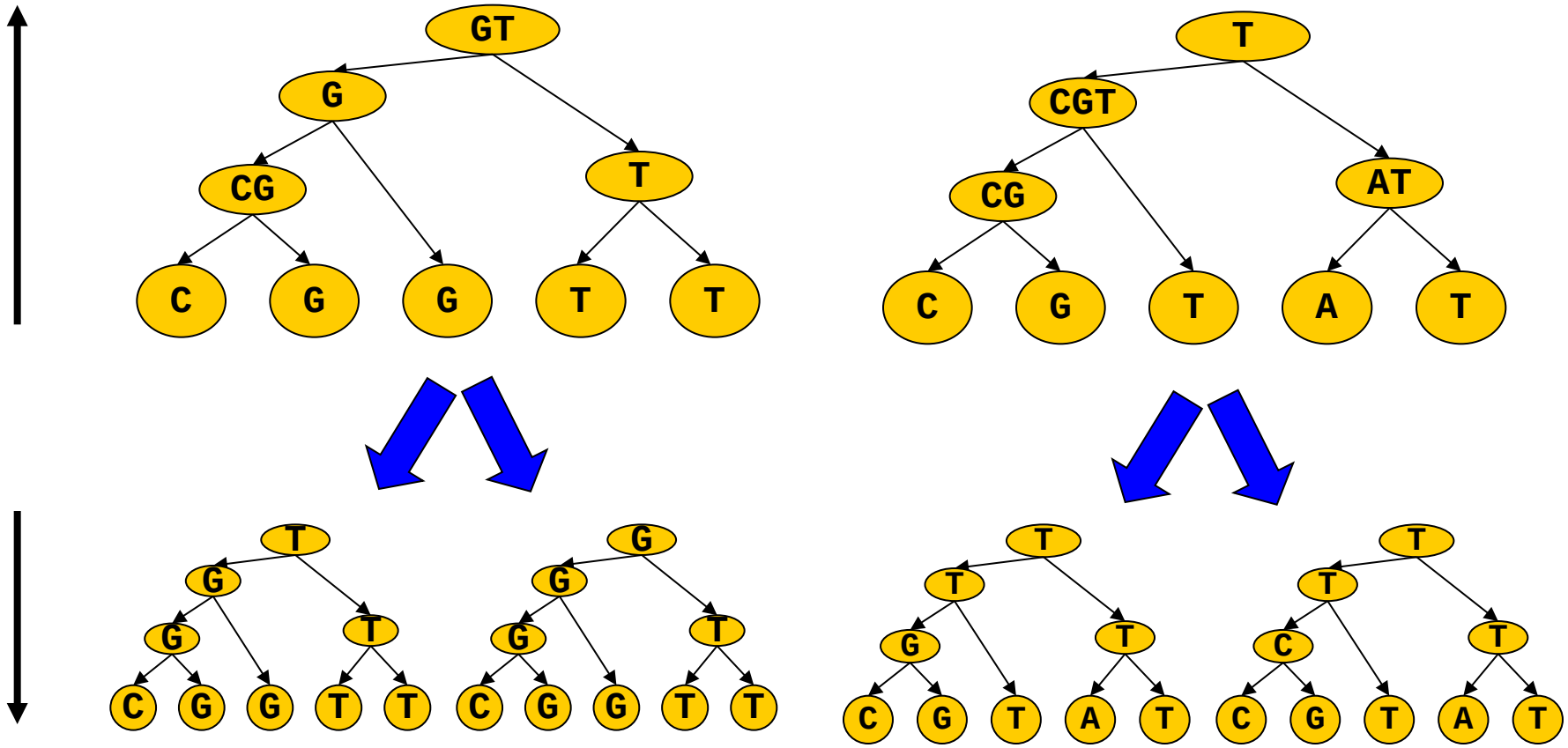
Beispiel



Fitch, Phase 2

- Phase 2
  - Wähle label(root) beliebig aus P(root)
  - Traversiere alle inneren Knoten k
    - Wenn label(parent(k))$\in$P(k), dann setze label(k)= label(parent(k))
    - Sonst wähle label(k) beliebig aus P(k)
- Theorem
  Jedes von *Fitch's Algorithmus berechnete Labeling von T hat den kleinstmöglichen Parsimony Score.*
- Beweis: Literatur

Beispiel

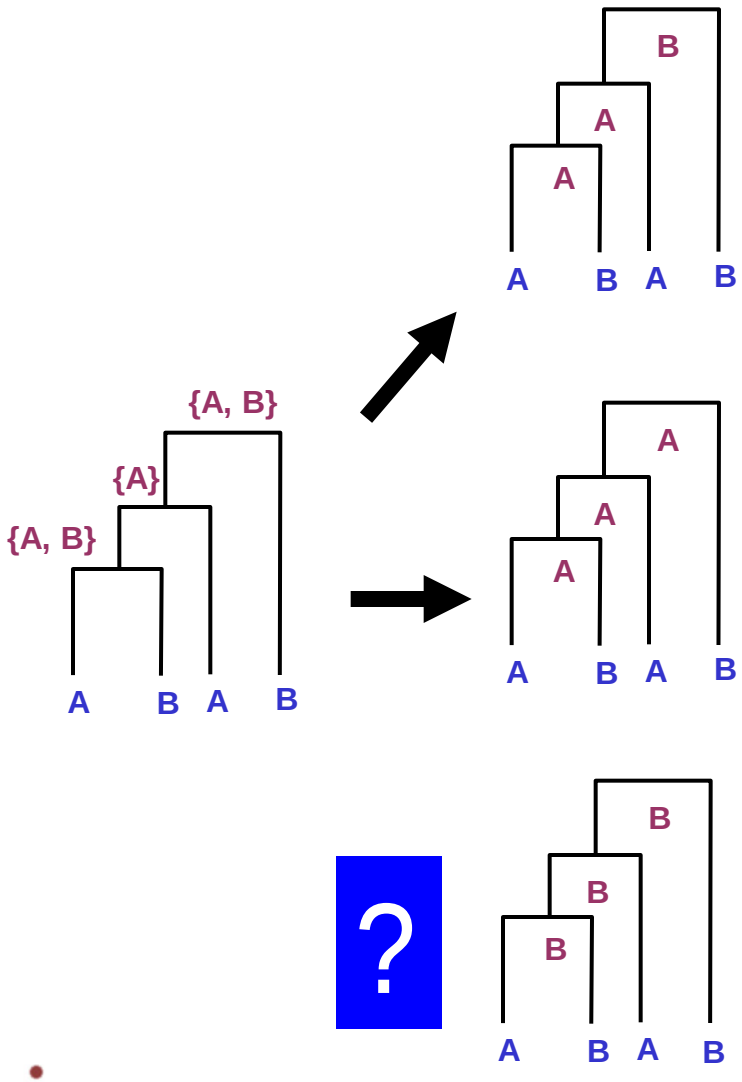


Scores: **2** **2** **3** **3**

Komplexität von Fitch‘s Algorithmus

- Beachte: Jedes P kann maximal z Elemente haben
- Phase 1: Für jeden inneren Knoten müssen wir O(z) Vergleiche machen, um P auszurechnen – also O(n*z)
- Phase 2: Traversierung aller innerer Knoten ist O(n); dazu ein O(log(z)) Test auf Enthaltensein des Vaterlabels im Label jedes Kindes
- Zusammen: O(n*z + n*log(z)) = O(n*z)

Aber Vorsicht



- Fitch's Algorithmus findet nicht alle optimalen Bäume
  - Algorithmus ist greedy
  - Erkennt nicht, dass einen Wechsel in Kauf zu nehmen sich später auszahlen kann
- Verbesserung (und Verallgemeinerung)
  - Weighted Parsimony
  - Sankoff's Algorithmus

Weighted Parsimony

- Bisher nehmen wir an, dass alle Änderungen von Zuständen eines Characters gleich viel kosten (nämlich 1)
  - Schlecht: Siehe PAM, BLOSSUM, etc.
- Weighted Parsimony: Beachte Substitutionsmatrix M
- Formal
  - Geg. Topologie T, Substitutionsmatrix M und Taxa/Charactermatrix D
  - Finde eine Beschriftung der inneren Knoten von T so, dass der gewichtete Parsimony Score $S^w(T)$ minimiert wird

$$S^w(T) = \sum_{(u,v)\in E(T)} \sum_{j\in Z} M(v_j, u_j)$$

Sankoff‘s Algorithmus

- Sankoff, Rousseau. "Locating the vertices of a Steiner tree in an arbitrary metric space." *Mathematical Programming* 9.1 (1975): 240-246.
- Wieder zwei Phasen
  - Berechne für jeden Knoten k und jeden Zustand z die minimalen Kosten $S_z(k)$ des Baumes unter k, wenn k mit z beschriftet wird
  - Zweite Phase legt dann die Label fest
- Phase 1
  - Für alle Blätter, setze

$$S_z(k) = \begin{cases} 0, wenn\ label(k) = z \\ \infty\ sonst \end{cases}$$

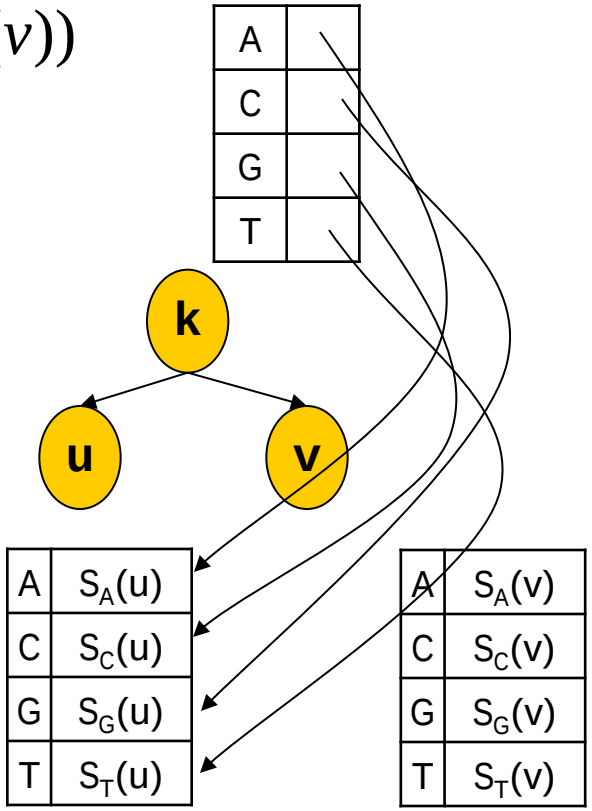
  - Traversiere den Baum bottom-up und berechne für jeden Knoten k mit Kindern u und v (z läuft über alle Zustände):

$$S_z(k) = \min_i (M(z,i) + S_i(u)) + \min_i (M(z,i) + S_i(v))$$

Erläuterung

$$S_z(k) = \min_i(M(z,i) + S_i(u)) + \min_j(M(z,j) + S_j(v))$$

- Wir berechnen die minimalen Kosten für den Teilbaum ab k für alle möglichen Label für k
  - Wenn wir k mit „A“ beschriften würden
  - Kann u A,C,G, oder beschriftet werden
    - Mit A: k mit A kostet $M[A,A]+S_A(u)$
      - M: Kante; $S_A(u)$: Unterbaum ab u
    - Mit C: k mit A kostet $M[A,C]+S_C(u)$
    - ...



  - Wir brauchen der Minimum der z Möglichkeiten für u und für v
  - Kindern u,v sind unabhängig

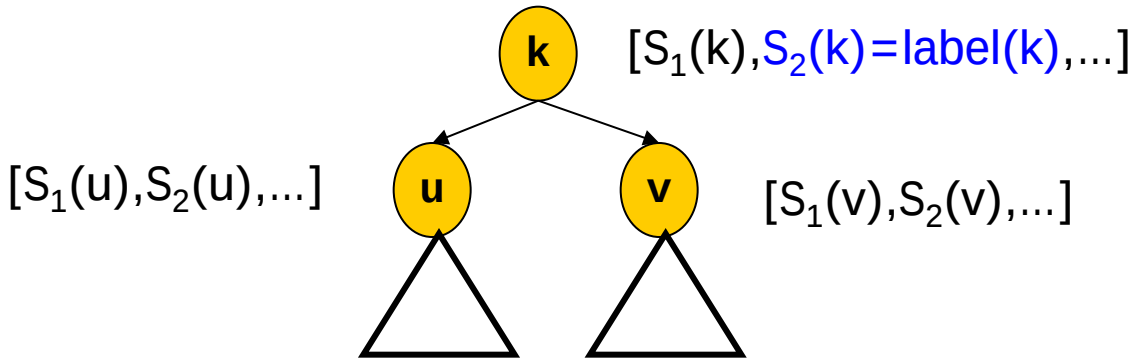
Sankoff‘s Algorithmus

- Phase 2 (Top-down)
  - Wurzel:

$$label(root) = \min_i (S_i(root))$$

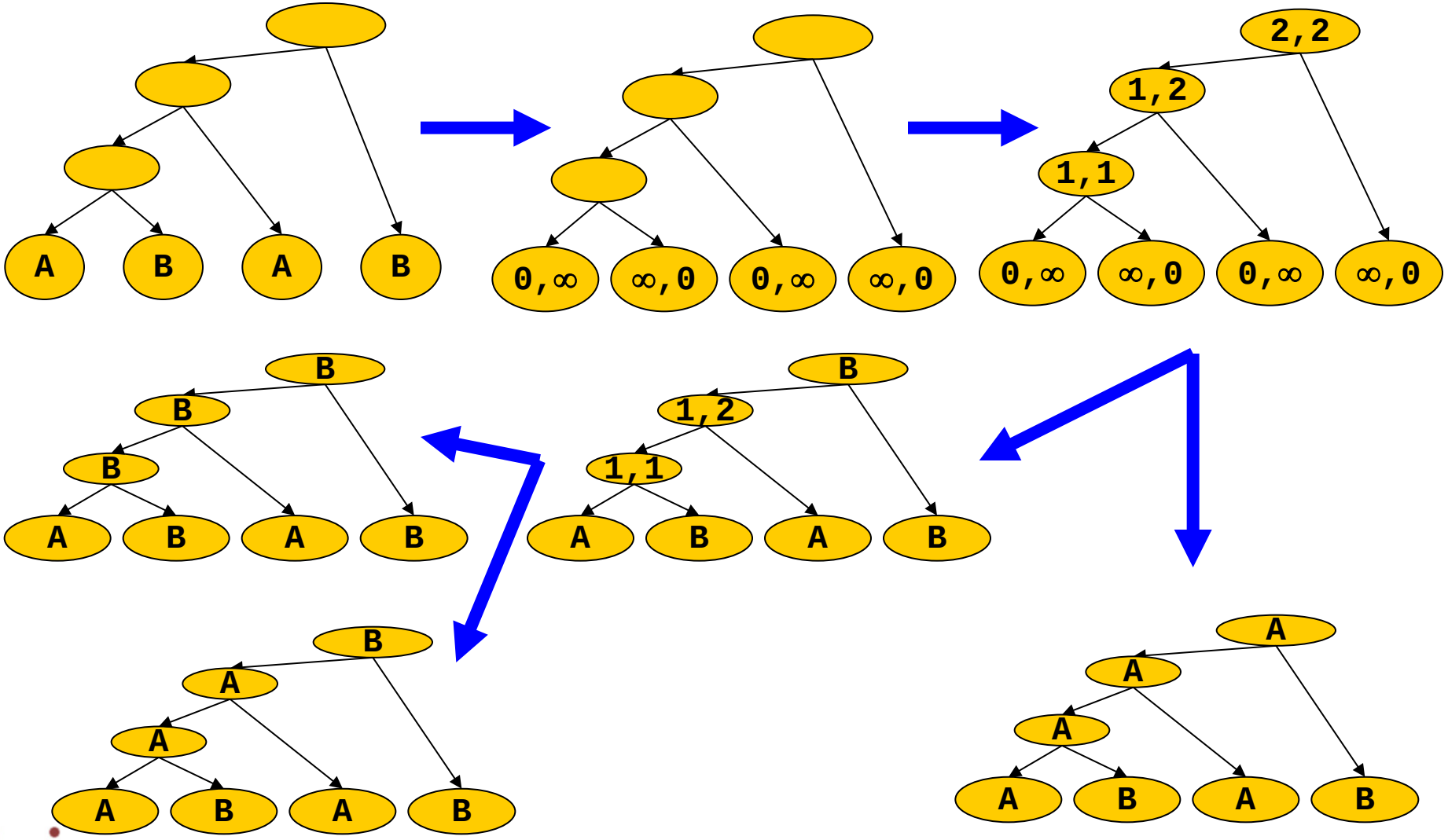
  - Knoten u mit Vater k:

$$label(u) = \min_i (M(label(k), i) + S_i(u))$$



Beispiel – Was bei Fitch schief ging

	A	B
A	0	1
B	1	0

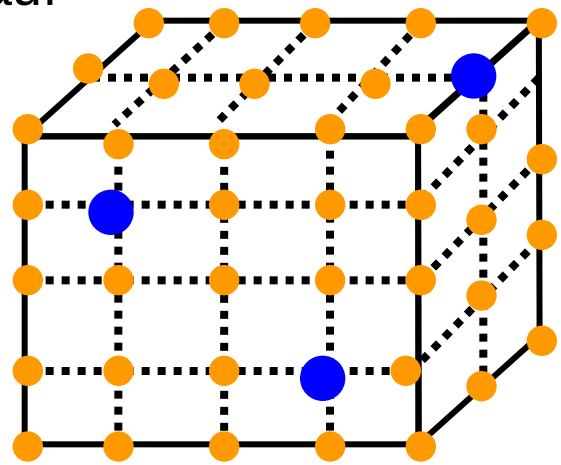


Large parsimony

- Small parsimony kann man also effizient lösen
- Definition
  Geg. eine Taxa/Character Matrix D. Das Large Parsimony Problem (LPP) sucht nach der Baumtopologie T und den Labels der inneren Knoten so, dass der (gewichtete) Parsimony Score von T minimal ist.
- Es gilt
  - LPP ist NP-schwierig
  - Im Prinzip müssen wir alle möglichen Topologien ausprobieren
  - Warum ist das so (Sketch)?

MP und Steiner Bäume

- Reduktion auf Steiner-Baum Problem auf m-dim. Würfel
  - ... also kann man die Lösung leicht bis auf Faktor 2 approximieren
- Gegeben m Character mit jeweils z Zustände
  - Das spannt einen m-dimensionalen Raum auf
  - Bilde den Graphen G mit Knoten für alle Gitterpunkte und Kanten zwischen Knoten entlang aller Achsen
  - Jedes Taxa ist ein Knoten im Graph
  - Der Abstand zweier Taxa ist die Zahl ihrer Koordinaten mit ungleichen Werten
  - Das MP Problem ist jetzt äquivalent zu: Finde den Steiner Baum für alle Taxa-Punkte in G



Branch & Bound

- Heuristik zur Lösung: Branch & Bound
- Beobachtung
  - Der Parsimony Score eines Baumes wird durch Hinzufügen eines neuen Blattes niemals kleiner

Branch & Bound Algorithmus

- Gegeben eine Matrix D
- Rekursive Tiefensuche durch alle möglichen Topologien
  - Berechne optimales S(T) für jeden (wachsenden) Baum
    - Beginne mit allen Topologien für die ersten 3 Arten
    - Zähle alle Möglichkeiten auf, die 4. Art hinzuzufügen
    - Bei k bisherigen Arten im Baum, gibt es 2k-1 Möglichkeiten
    - Halte eine davon fest und steige weiter ab (5. Art, 6. Art ...)
- An jedem Blatt des Suchbaums haben wir eine komplette Topologie K für D mit optimalem Score S=S(K)
  - Der ist natürlich i.A. nicht optimal für D
- Traversiere den Rest des Baums
  - Immer, wenn ein (Teil-)Baum einen Score größer S hat, vergiss diesen Ast des Suchraums (Pruning)
  - Passe S ständig an das aktuelle Optimum an

Eigenschaften

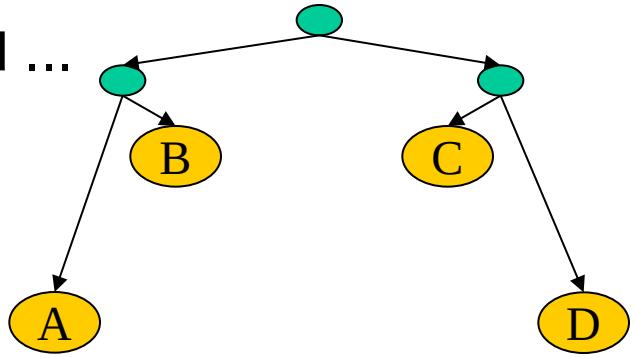
- Vergleichbar dem A* Algorithmus
- Worst-Case ist unverändert, aber AC deutlich besser
- Idee zum schnellen Finden einer guten oberen Schranke
  - Topologie durch Neighbor Joining bestimmen
  - Bestes Labeling mit Fitch/Sankoff berechnen
  - Dessen Score als erste obere Schranke benutzten
- Viele weitere Heuristiken möglich/nötig
  - Welche Bäume soll man zuerst aufzählen?
  - Vielleicht immer die besten 2 Kinder weiterverfolgen?
  - ...

Andere Möglichkeiten

- Iterative Verbesserung
  - Beginne mit irgendeiner Topologie und berechne optimalen Score
  - Verändere diese „lokal“, nach Möglichkeit zum Guten
  - So lange, bis es nicht mehr besser wird
  - Wiederhole das mit vielen zufälligen Startbäumen
- Greedy
  - Zähle alle Bäume mit wachsender Größe auf
  - Berechne für jede Topologie jeweils den besten Score
  - Wähle jeweils den besten Baum, und erweitere nur diesen

„Felsenstein Zone“

- Eine Methode ist "statistisch konsistent", wenn die Wsk, bei geg. Daten den richtigen Baum auszurechnen, mit wachsender Länge der Eingabe gegen 1 geht
- MP ist nicht statistisch konsistent
  - Bsp: Je länger die Kanten zu A und D sind …
    - Desto größer die Unterschiede zwischen A,B und D,C
    - Desto größer die Wahrscheinlichkeit, dass A und D durch Mehrfachmutationen zufällig ähnlich werden



  - „Long branch attraction“ – A und D werden zu Nachbarn
- Ursache
  - MP normiert nicht über die Sequenzlänge
  - MP ignoriert (wie alle Methoden bisher) Mehrfachmutationen

Vergleich

- Man liest häufiger, dass alle Phylogeniemethoden recht gut funktionieren
  - Gilt nur bei einfachen Evolutionsmodellen
  - Güte hängt von den Eigenschaften der Daten ab
- Distanzbasierte Methoden
  - Am ungenauesten, dafür schnell
  - Brauchen numerische Abstandsmasse
- Maximum Parsimony: Besser, teuer, braucht MSA
- Maximum Likelihood: Noch besser, noch teurer
- Also: Mehrere Methoden vergleichen
  - Gruppen, die überall gleich sind, gelten als sehr robust
  - „Consensus tree“