



# Evolutionäre Bioinformatik

## 2: Ableiten von Bäumen

**Benedikt Brors**

Abt. Theoret. Bioinformatik

Deutsches Krebsforschungszentrum

Heidelberg



# Ableitung von Bäumen



- Methoden, die auf Sequenzen beruhen:
  - Maximum parsimony
  - Maximum likelihood
- Methoden, die auf Distanzen basieren:
  - UPGMA (hierarchical clustering)
  - Neighbor Joining
  - Fitch-Margoliash (least squares)



# Maximum parsimony

- Prinzip in der Philosophie: Hypothesen, die weniger Hilfsannahmen machen, sind plausibler
- Wilhelm von Ockham (14. Jhdt.): *pluralitas non est ponendam sine necessitate* (Occam's razor)
- Anwendung in der Bioinformatik: Prinzip der maximalen **Sparsamkeit** (parsimony)
- Der Baum, der die wenigsten Änderungen von Buchstaben verlangt, ist der beste

# Begriffe

- **Zeichen** (character) sind die möglichen Ausprägungen eines Merkmals in einem Taxon
- **Zeichenzustände** (character states) sind die tatsächlich an einer gegebenen Stelle vorhandenen Zeichen
- Die **Zeichenzustandsmatrix**  $X$  gibt den Zeichenzustand  $x_{ij}$  an, der der Position  $j$  im Taxon  $i$  zugewiesen ist.
- Zeichen können nicht nur Nukleotide sein, sondern auch Vorhandensein oder Abwesenheit von Eigenschaften, Allel, das an einem bestimmten Locus vorliegt etc.
- Wir beschränken uns auf biologische Sequenzen.  
Zeichen  $\Rightarrow$  *unordered multistate characters*

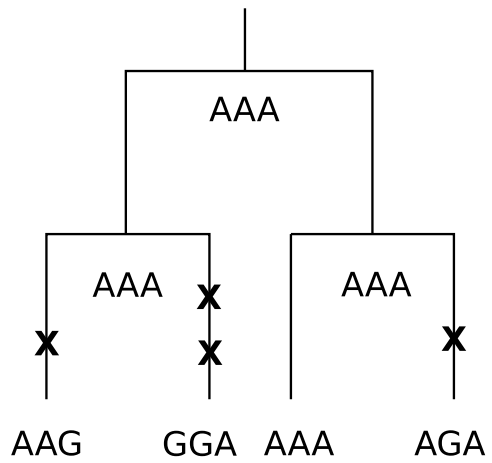
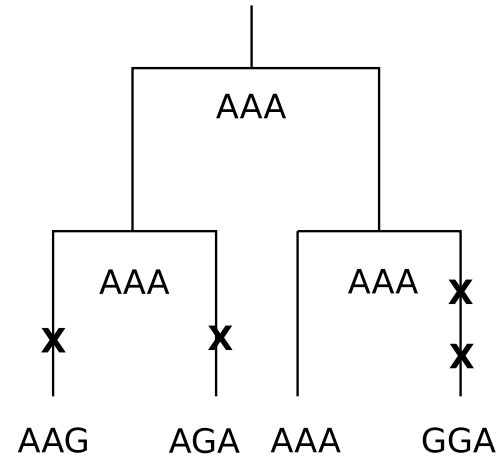
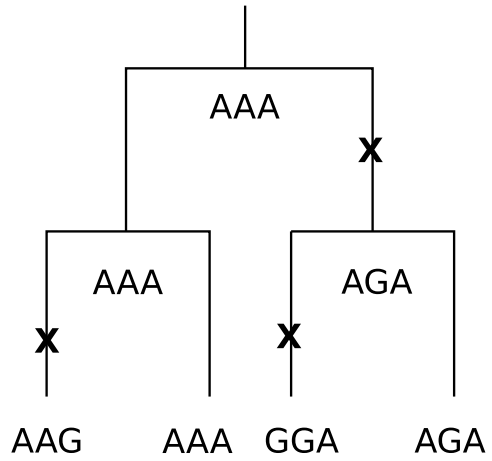
# Baumlänge

- Die Baumlänge  $TL$  wird dann definiert als:

$$TL = \sum_{k=1}^B \sum_{j=1}^N w_j \delta(x_{k'j}, x_{k''j})$$

- Die Summe ist über alle Äste  $B$  und alle Positionen  $N$ ;  $k'$  und  $k''$  sind die beiden Knoten, die zu einem Ast  $k$  gehören.
- Das maximum-parsimony Verfahren sucht einen Baum mit minimaler Baumlänge  $TL$  und versucht, den internen Knoten optimale Zeichenzustände zuzuweisen

# Beispiele



- |   |     |
|---|-----|
| 1 | AAG |
| 2 | AAA |
| 3 | GGA |
| 4 | AGA |



# Fitch-Algorithmus

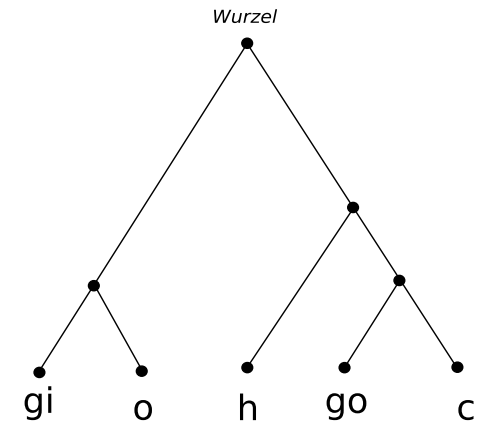
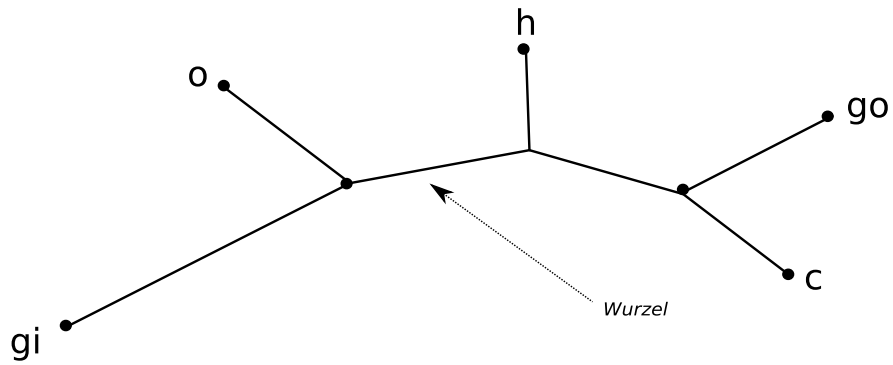
- Fitch (1971) hat einen Algorithmus vorgeschlagen, mit dem man die Länge eines gegebenen Baums effizient berechnen kann
- gilt für ungeordnete Multizustands-Zeichen (z.B. nt, AA etc.)
- Setzt volle Reversibilität voraus, jeder Zustand kann frei in jeden anderen Zustand wechseln
- Unabhängigkeit der Positionen wird vorausgesetzt
- Bestimmung der Länge kann mit einem Durchgang durch den Baum erzielt werden (*postorder traversal*)

# Algorithmus

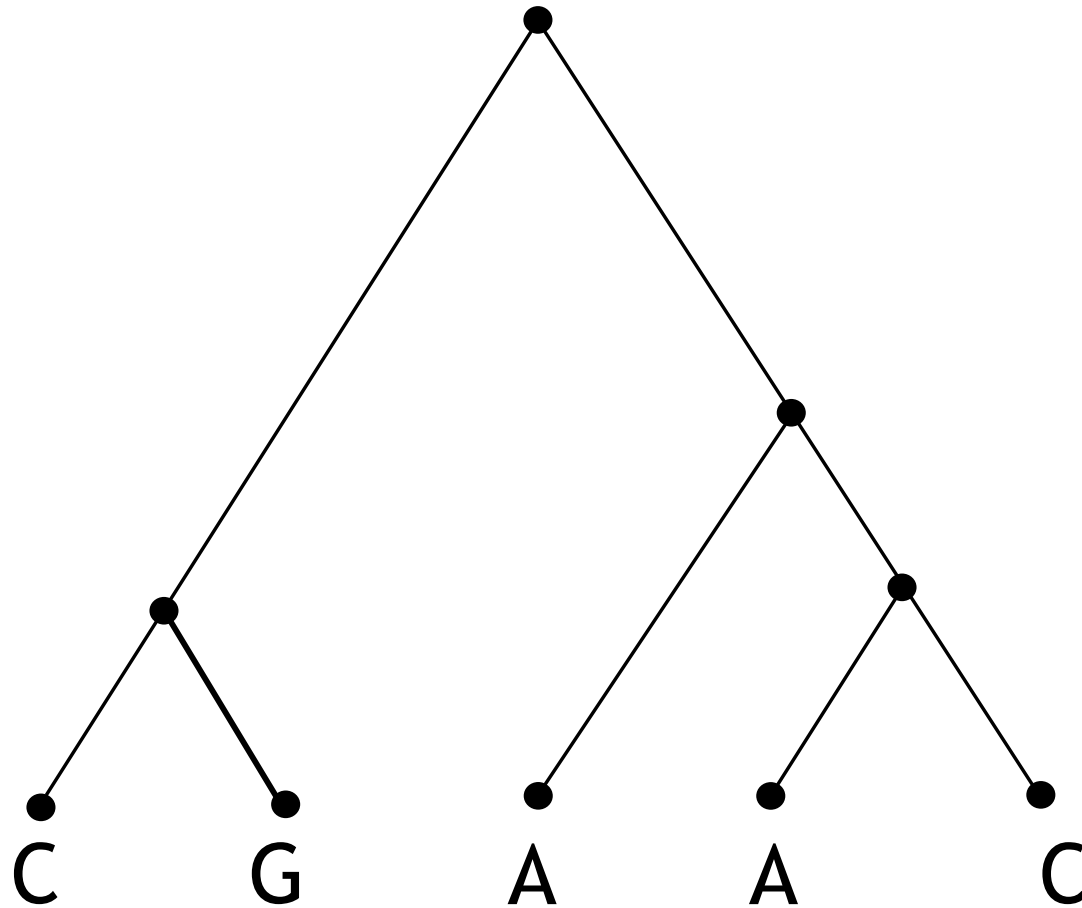
- Weise die Wurzel einer beliebigen Kante zu (Baumlänge ist unabhängig von der Position der Wurzel)
- Jedem terminalen Knoten ist ein Zustandsset  $S_i$  zugeordnet
- bottom-up pass (*postorder traversal*):
  - für jeden internen Knoten  $u$ : weise Zeichenzustände zu, so daß:

$$u = \begin{cases} \mathcal{V} \cup \mathcal{W} & \text{if } \mathcal{V} \cap \mathcal{W} = \emptyset \\ \mathcal{V} \cap \mathcal{W} & \text{else} \end{cases}$$

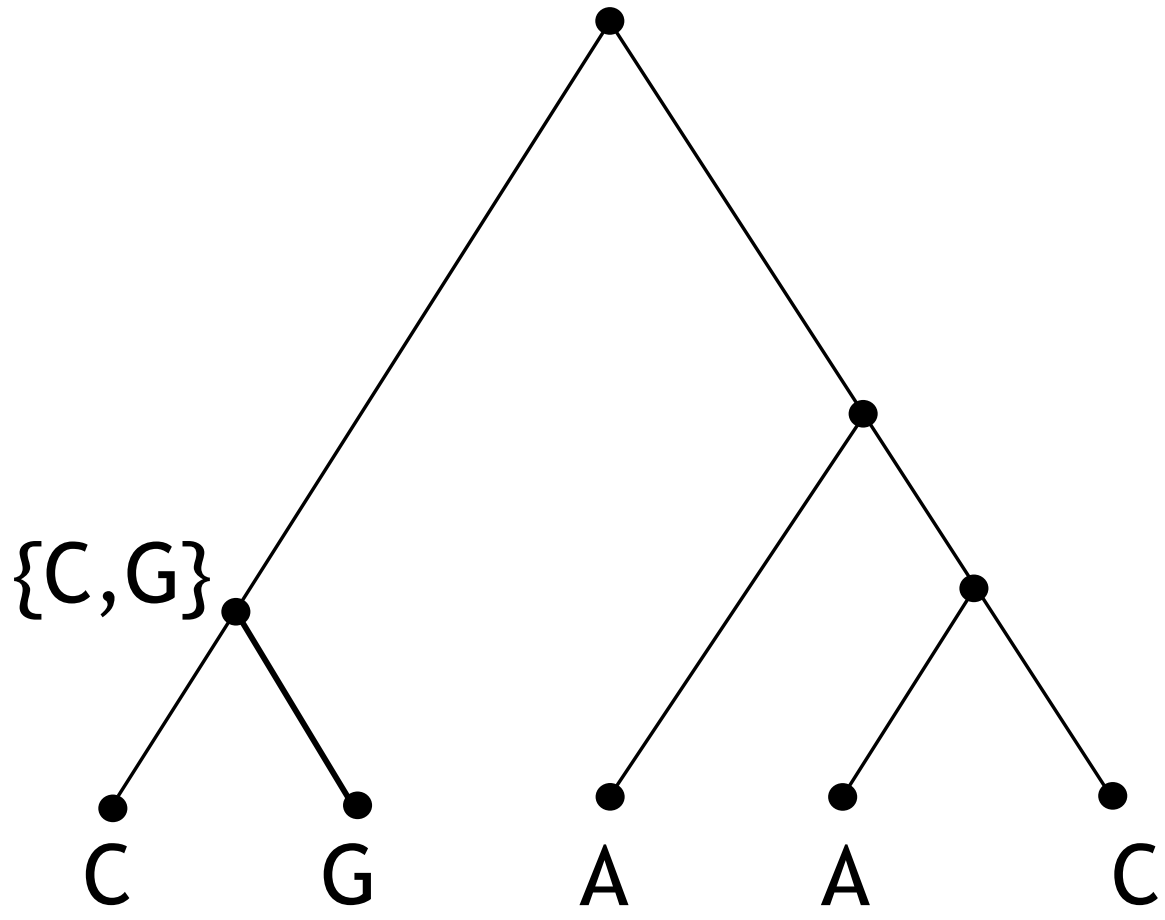
# (Forts.)



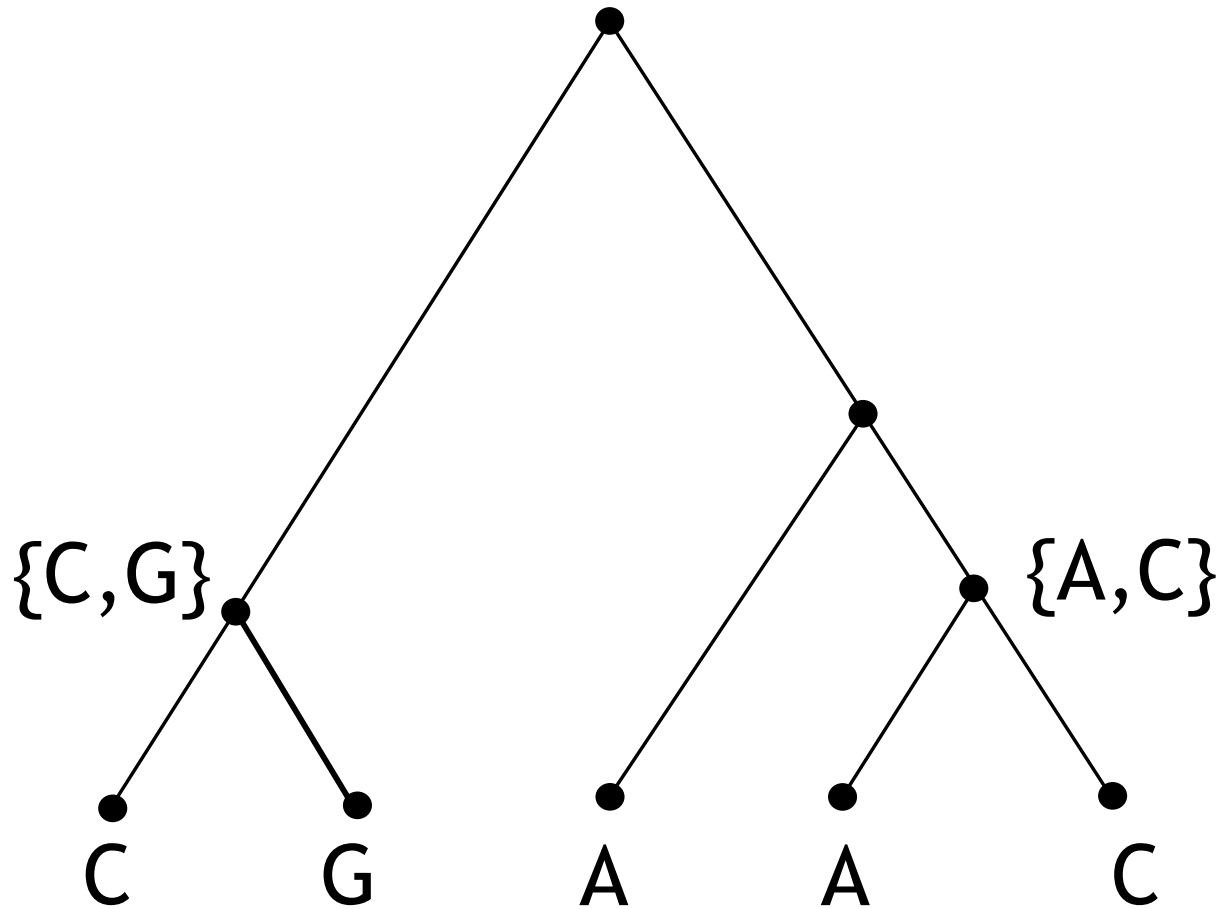
# Fitch Algorithmus



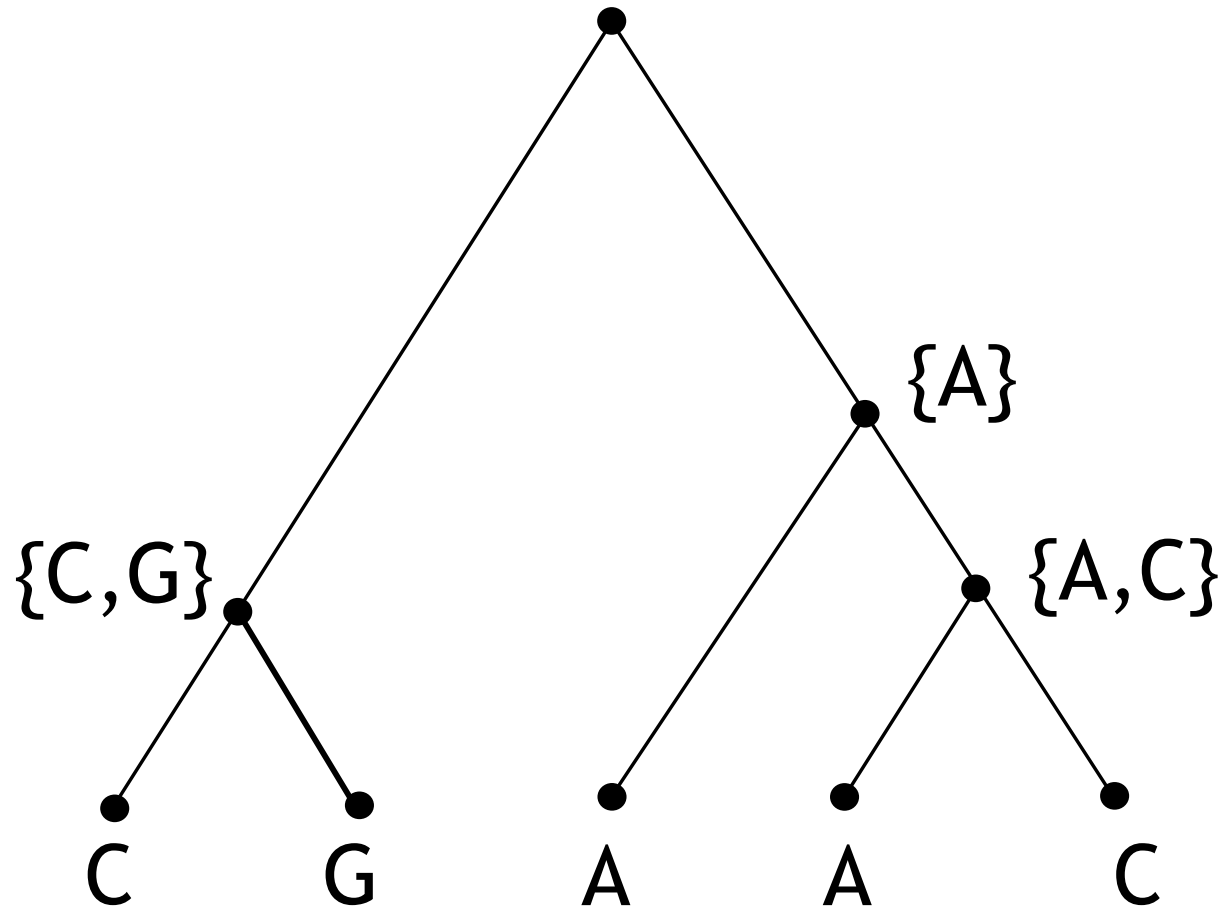
# Fitch Algorithmus



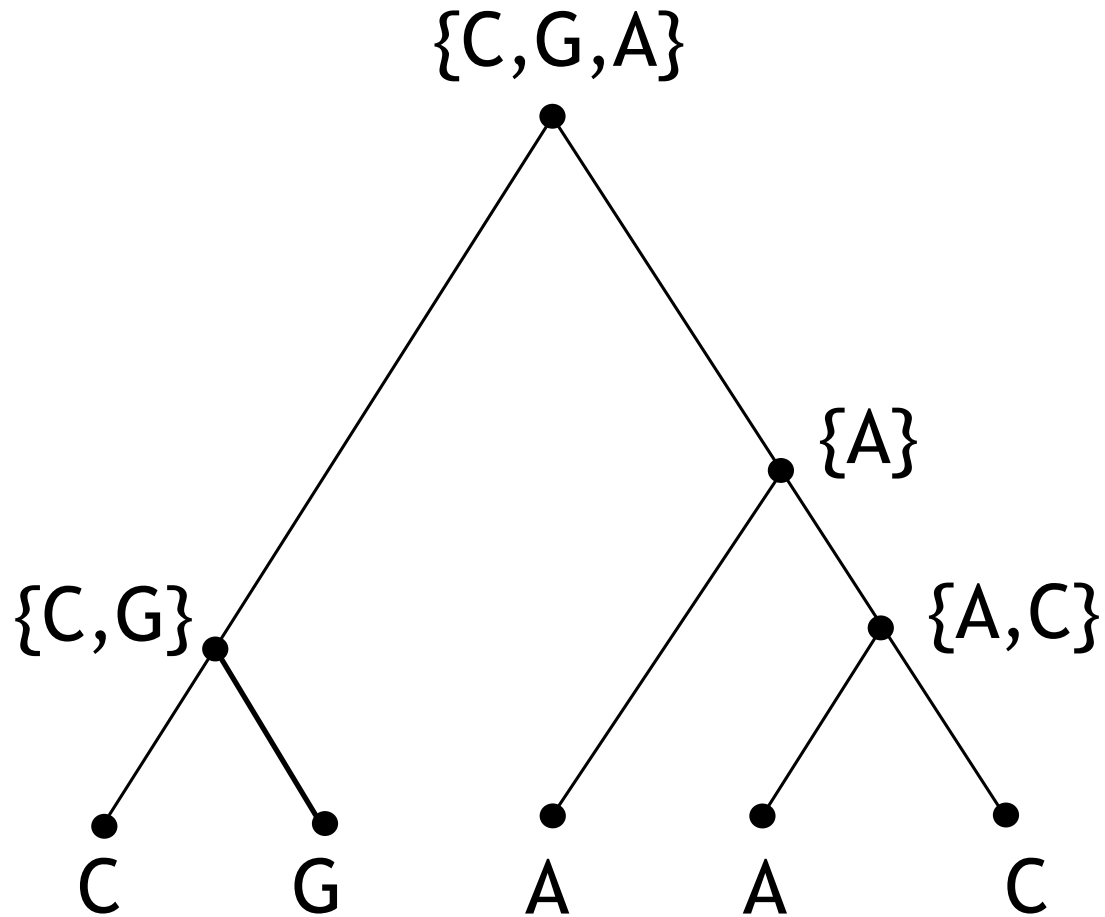
# Fitch Algorithmus



# Fitch Algorithmus



# Fitch Algorithmus

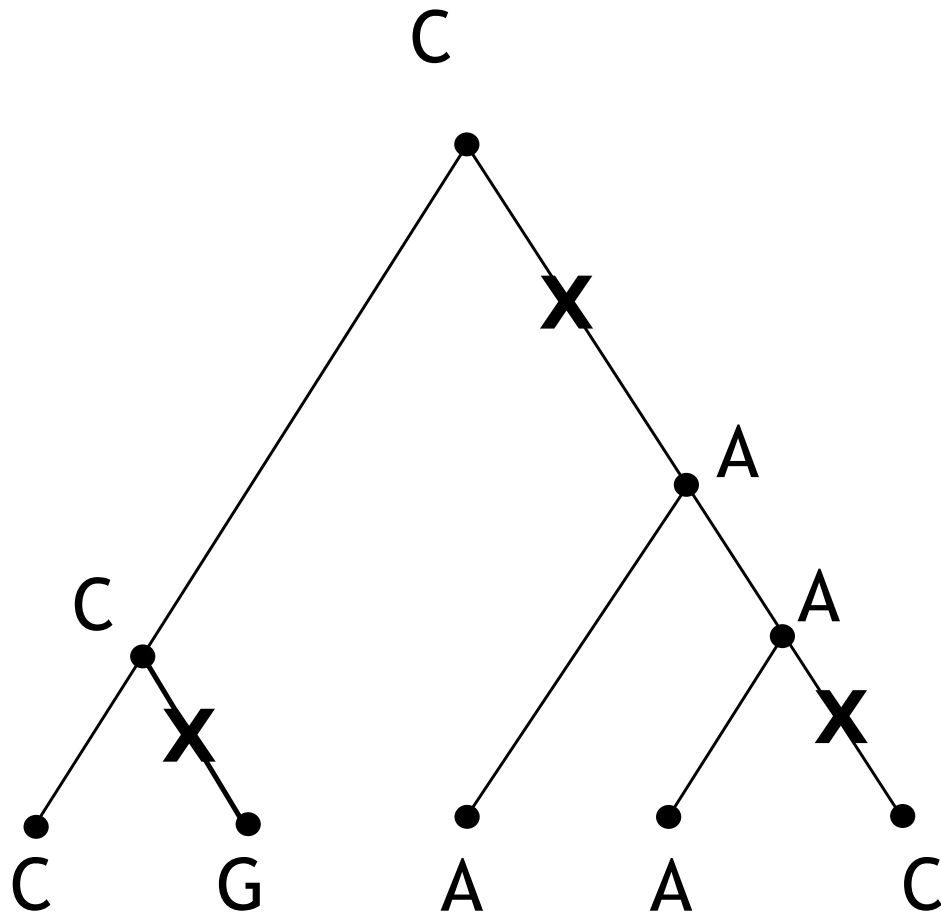


# interne Knoten

- Zum Zuweisen der Zeichenzustände zu internen Knoten ist ein zweiter Durchlauf durch den Baum erforderlich
- Wurzel  $\rightarrow$  Blätter: *preorder traversal*
  - Weise der Wurzel eins der Zeichen aus der Menge  $\mathcal{U}_{\text{Wurzel}}$  zu
  - für jeden Kindknoten  $v$ :

$$v_j = \begin{cases} x & \text{if } x \in \mathcal{U} \\ \text{any state} & \text{else} \end{cases}$$

# Bsp.



# Bester Baum

- Der Fitch Algorithmus erlaubt, die Baumlänge effizient zu berechnen ( $\mathcal{O}(n)$ ), aber er findet nicht den besten Baum
- Strategien:
  - $n$  klein ( $< 10$ ): Erschöpfende Suche über alle Topologien
  - $n$  mittelgroß ( $\lesssim 20$ ): **branch-and-bound**:
    - Berechne obere Schranke für  $TL$  (z.B. NJ-Algor.)
    - Beginne mit 3 Taxa, füge nächstes Taxon an jeder Kante ein
    - Wenn die obere Schranke erreicht wird, breche den Ast jeweils ab
    - Schließt viele Möglichkeiten aus, verkleinert den Suchraum erheblich



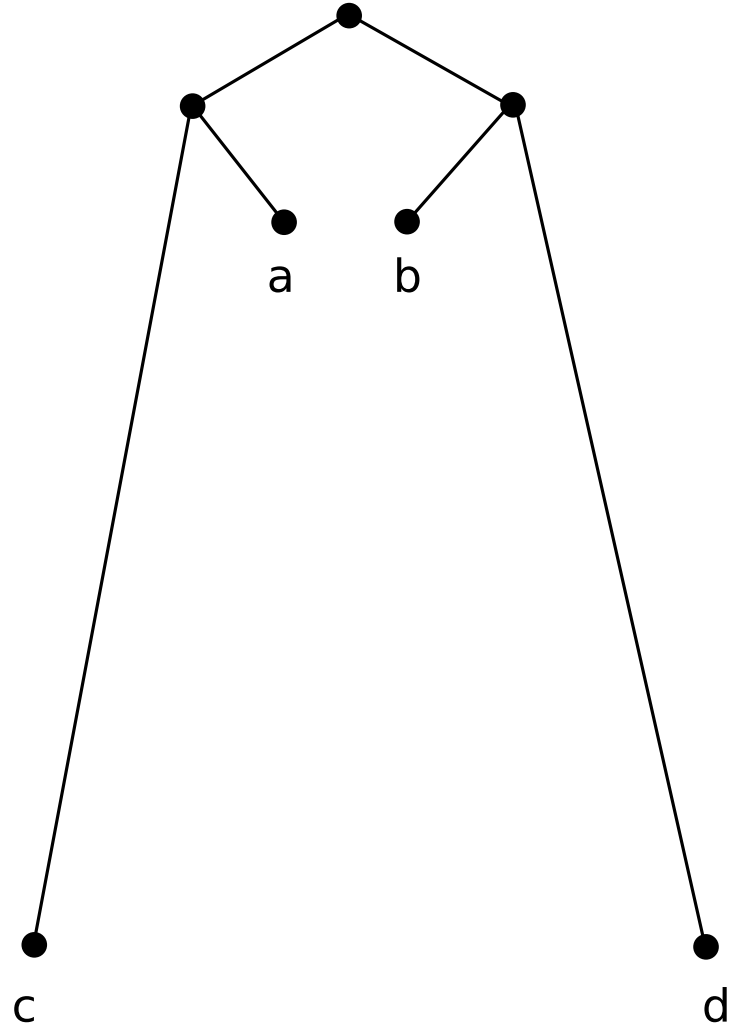
# Heuristiken

- Wenn  $n$  groß ist ( $> 20$ ), muß man heuristische Verfahren einsetzen
  - z.B. wird der Baum nach und nach aufgebaut, in jedem Schritt wählt man den Weg, der  $TL$  minimiert: **greedy search, hill climbing**
  - Am Ende wird der Baum durch Verstzen von Ästen (**branch swapping**) weiter optimiert.
  - Bei unterschiedlicher Reihenfolge der selben Sequenzen ergeben sich u.U. verschiedene Bäume

# Probleme von MP

- eignet sich nur für nahe verwandte Sequenzen (nicht beobachtbare multiple Substitutionen)
- **long branch attraction:**  
MP funktioniert schlecht, wenn die Bäume nicht dicht genug abgedeckt sind
  - Nur wenige Substitutionen zwischen nahe verwandten Sequenzen
  - nur wenige Übereinstimmungen zwischen entfernt verwandten Sequenzen
  - Nur Sequenzen, die zwischen den nahe verwandten Sequenzen verschieden, aber innerhalb eines Astes gleich sind, führen zur korrekten Topologie. Diese sind selten
  - Maximum-Likelihood-Verfahren finden die korrekte Topologie

# Beispiel





# Robustheit von Bäumen



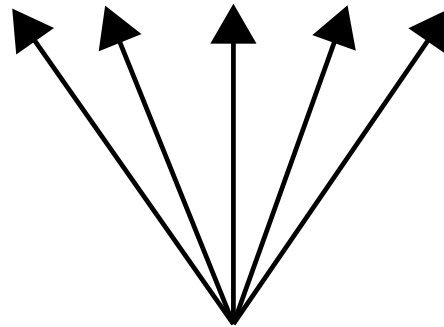
- Robustheit von Bäumen wird über **bootstrap** bestimmt
- Man zieht (mit Zurücklegen) Spalten aus dem Alignment
- Aus jedem Pseudo-Alignment wird ein Baum berechnet
- Man bestimmt, wie oft ein bestimmter Ast in den Bootstrap-Proben vorkommt
- Gegenteil: **jackknife** (streiche zufällig Spalten)



# Bootstrap



C	C	G	C	...
A	C	A	T	...
C	T	G	C	...
T	C	C	G	...



ziehe Spalten  
mit Zurücklegen

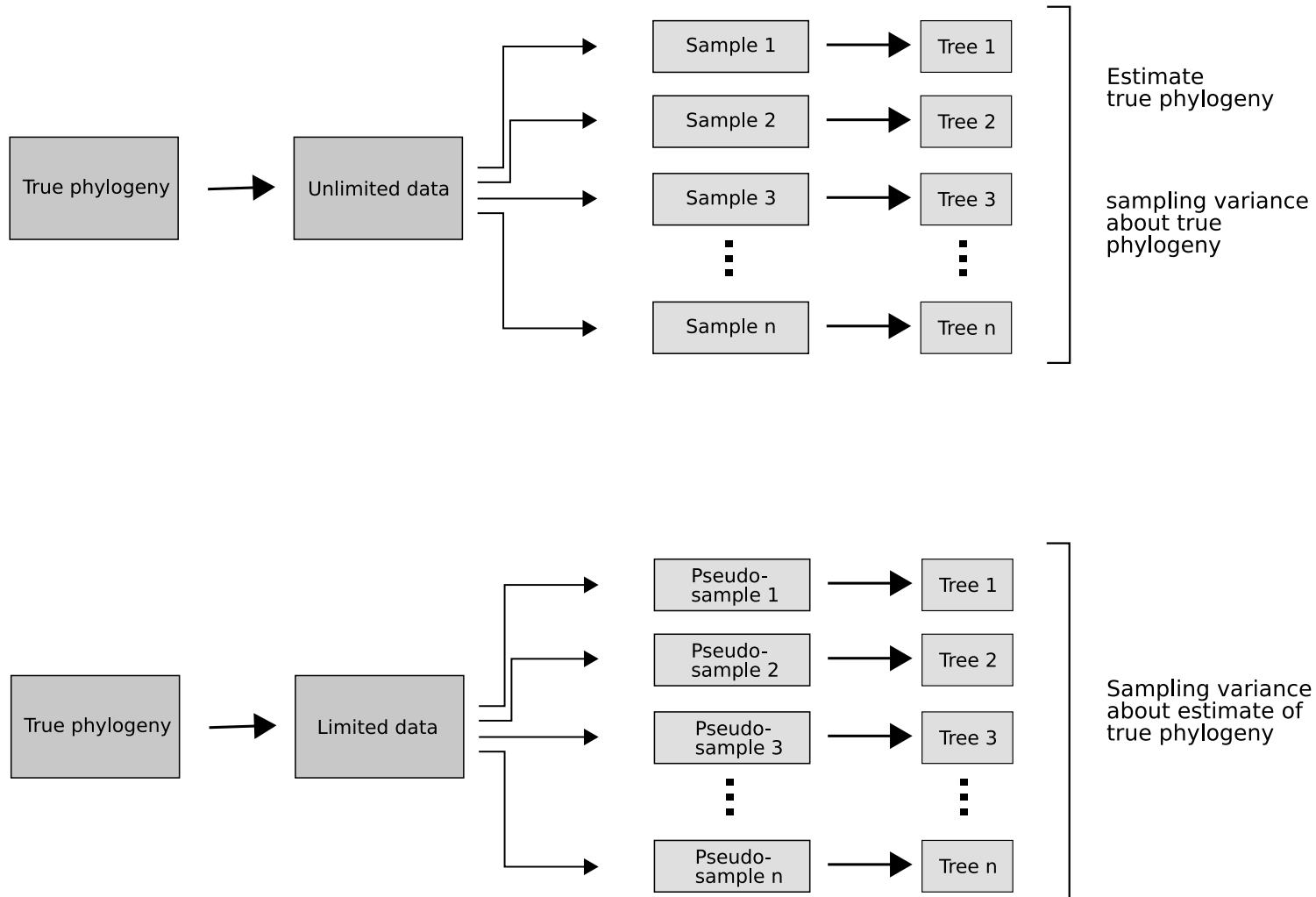


# Stichprobenvarianz

- Der Anteil an Topologien mit dem untersuchten Ast sei  $P$
- $P$  ist binomial verteilt
- Dann ist die Stichprobenvarianz (Fehler) von  $P$ :

$$\sigma^2 = \frac{P(1 - P)}{n}$$

- $P$  kann als Wahrscheinlichkeit interpretiert werden, in neuen, unabhängigen Datensätzen den internen Ast wiederzufinden



# Bias



- Die Bootstrap-Schätzungen weisen einen **Bias** auf
- Die Größe des Bias hängt ab von:
  - Anzahl der Taxa
  - Anzahl der Zeichen (d.h. Länge des Alignments)
  - Position des internen Astes innerhalb des Baums





---

# Maximum-Likelihood-Methode



# ML Methode

- Phylogenetische Analyse: Rekonstruktion des *wahrscheinlichsten* Baums gegeben ein Alignment  $\mathcal{D}$
- Baum: besteht aus Topologie  $\mathcal{T}$  und Astlängen
- “Wahrscheinlichkeitsfunktion” = Likelihood für Baum
- Suche Baum, der die Likelihood maximiert

# Modellannahmen

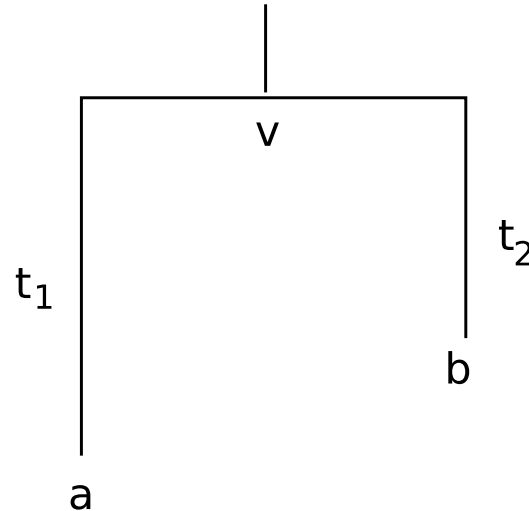
- konstante Substitutionsrate
- Gleichwertigkeit und Unabhängigkeit der einzelnen Positionen in einer Sequenz bzw. einem Alignment
- Zeitreversibilität  $\Rightarrow$  ungewurzelter Baum hat selbe Likelihood wie alle davon abgeleiteten gewurzelten Bäume



# Likelihood

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_1 \cdot \mathcal{L}_2 \cdots \mathcal{L}_N \\ \log(\mathcal{L}) &= \log(\mathcal{L}_1) + \log(\mathcal{L}_2) + \cdots + \log(\mathcal{L}_N)\end{aligned}$$

# einfacher Baum



$$P(a, b|T, t_1, t_2) = \sum_{v \in \mathcal{A}} \pi_v P(a|v, t_1) P(b|v, t_2)$$

Summiert über alle Positionen:

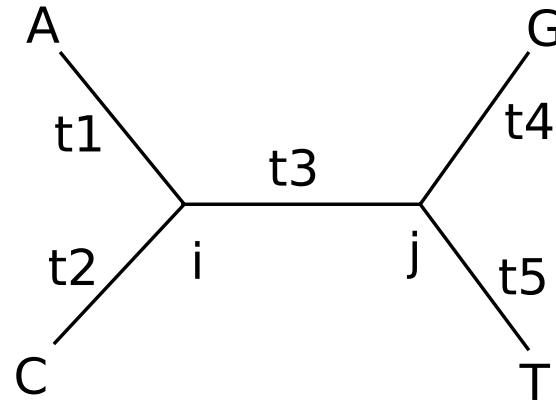
$$P(a, b|T, t_1, t_2) = \prod_{n=1}^N P(a_n, b_n|T, t_1, t_2)$$

# Berechnung der bed. Wahrsch.

- Die bedingten Wahrscheinlichkeiten können über probabilistische Evolutionsmodelle berechnet werden
- Im Jukes-Cantor-Modell sind z.B.

$$P(i|i) = \frac{1}{4} + \frac{3}{4} \exp(-4\alpha t)$$
$$P(i|j) = \frac{1}{4} - \frac{1}{4} \exp(-4\alpha t) \quad \forall i \neq j$$

# komplexer Baum



gegeben ein Alignment  $\mathcal{D}$ :

$$\Pr(\mathcal{D}_{Site} | \mathcal{T}, t_1, \dots, t_5) = \sum_{i \in \mathcal{A}} \pi_i P_{iA}(t_1) P_{iC}(t_2) \sum_{j \in \mathcal{A}} P_{ij}(t_3) P_{jG}(t_4) P_{jT}(t_5)$$





Und über alle Positionen:

$$\begin{aligned}\mathcal{L}(t_1, \dots, t_5, \mathcal{T}) &= Pr(\mathcal{D} | \mathcal{T}, t_1, \dots, t_5) \\ &= \prod_{sites} Pr(\mathcal{D}_{Site} | \mathcal{T}, t_1, \dots, t_5)\end{aligned}$$



# Anwendung von ML



- $\log(\mathcal{L})$  wird maximiert
- Die Likelihood für eine Position kann effizient durch Rekursion berechnet werden (postorder traversal)
- ist sehr rechenintensiv
- Heuristiken, um besten Baum zu finden
- Astlängen müssen zusätzlich optimiert werden (z.B. Newton-Verfahren)

