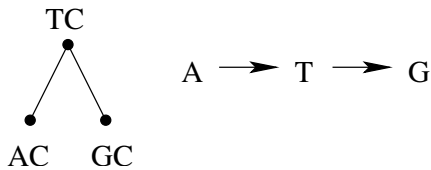


9.2 Maximum Likelihood Bäume

1. FELSENSTEIN-Verfahren

- Problem der Clusteringverfahren:
 - Distanz aus Alignment
 - ↔ keine Berücksichtigung multipler Ersetzungen



- ↔ keine wirklichen Distanzen
- Lösung:
 - direkte Beschreibung der Evolution als stochastisches Modell
 - ↔ wie bei PAM-Markovkette
 - Problem: Zeit t bei Markovkette fest
 - ↔ Matrix für jeden Zeitpunkt gesucht
- allgemeine Form:

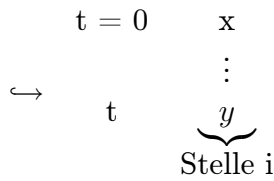
$$P(t) = \begin{pmatrix} P_{AA}(t) & \dots & P_{AT}(t) \\ \vdots & & \\ P_{TA}(t) & \dots & P_{TT}(t) \end{pmatrix}$$

- mit Markoveigenschaft
- P(t) müssen stochastisch sein

$$\forall_t : \sum_y P_{xy}(t) = 1$$

$$\forall_t : 0 \leq P_{xy}(t) \leq 1$$

- $P_{xy}(t)$... Wahrscheinlichkeit, dass Nukleotid y an einer Stelle i nach t Zeiteinheiten zu haben ist, falls x an Stelle i zum Zeitpunkt 0 ist



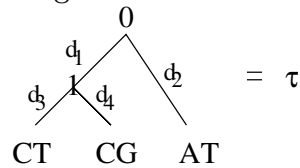
↔ kontinuierliche Zeitpunkte ⇒ Markovprozesse

• Zusammenhang statistisches Modell und Baum

– Beispiel:

Sequenzen: a = CT, b = CG, c = AT

möglicher Baum:



↔ Likelihood eines Baumes:

$$Pr[Seq(a, b, c)|\tau] = L(\tau)$$

• Berechnung der Wahrscheinlichkeit $Pr[Seq(a, b, c)|\tau]$

– Betrachtung aller Variationen für die Knoten 0,1

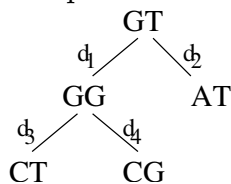
– Bedingung:

alle Taxa und interne Knoten haben gleiche Länge

– Begründung:

evolutionäres Modell erlaubt nur Substitutionen

– Beispiel:



– Annahme:

Stellen (Positionen) unabhängig

$$L(\begin{array}{c} \text{GT} \\ \swarrow \quad \searrow \\ \text{GG} \quad \text{AT} \\ \swarrow \quad \searrow \\ \text{CT} \quad \text{CG} \end{array}) = L(\begin{array}{c} \text{G} \\ \swarrow \quad \searrow \\ \text{G} \quad \text{A} \\ \swarrow \quad \searrow \\ \text{c} \quad \text{c} \end{array}) \cdot L(\begin{array}{c} \text{T} \\ \swarrow \quad \searrow \\ \text{G} \quad \text{T} \\ \swarrow \quad \searrow \\ \text{G} \quad \text{G} \end{array})$$

$$\hookrightarrow L(\begin{matrix} & & G & & \\ & & / \quad \backslash & & \\ & G & & A & \\ & / \quad \backslash & & & \\ c & & c & & \end{matrix}) = \prod_{GA} P_{GA}(d_2) \cdot P_{GG}(d_1) \cdot P_{GC}(d_3) \cdot P_{GC}(d_4)$$

– komplette Berechnung:

alle Variationen für 0,1

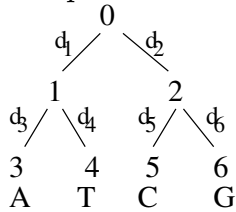
$$\hookrightarrow L(\begin{matrix} & & GT & & \\ & & / \quad \backslash & & \\ & GG & & AT & \\ & / \quad \backslash & & & \\ c & & c & & \end{matrix}) = \sum_{S_0 \in \{A,C,G,T\}} \sum_{S_1 \in \{A,C,G,T\}} \prod_{S_0} P_{S_0T}(d_2) P_{S_0S_1}(d_1) P_{S_1T}(d_3) P_{S_1G}(d_4)$$

• Komplexität:

- n Taxa → n-1 innere Knoten
- ↪ Anzahl Summanden: $4^{n-1} \rightarrow O(4^{n-1})$
- exponentiell

• Verbesserung:

- Umordnen der Terme → Struktur: $[() ()]$
- Beispiel:



$$\begin{aligned} & \sum_{S_0} \sum_{S_1} \sum_{S_2} \prod_{S_0} \cdot P_{S_0S_1}(d_1) \cdot P_{S_0S_2}(d_2) \cdot P_{S_1A}(d_3) \cdot P_{S_1T}(d_4) \\ & \quad \cdot P_{S_2C}(d_5) \cdot P_{S_2G}(d_6) \\ = & \sum_{S_0} \sum_{S_1} \prod_{S_0} \cdot P_{S_0S_1}(d_1) \cdot P_{S_1A}(d_3) \cdot P_{S_1T}(d_4) \cdot \sum_{S_2} P_{S_0S_2}(d_2) \\ & \quad \cdot P_{S_2C}(d_5) \cdot P_{S_2G}(d_6) \\ = & \sum_{S_0} [\prod_{S_0} \cdot (\sum_{S_2} P_{S_0S_2}(d_2) \cdot P_{S_2C}(d_5) \cdot P_{S_2G}(d_6)) \cdot (\sum_{S_1} P_{S_0S_1}(d_1) \\ & \quad \cdot P_{S_1A}(d_3) \cdot P_{S_1T}(d_4))] \end{aligned}$$

– Lösung: Dynamisches Programmieren

2. Dynamisches Programmieren

$L_{k,S} \dots$ Likelihood für Teilbaum unter Knoten k, falls mit S belegt

- Initialisierung:

– für jedes Blatt k

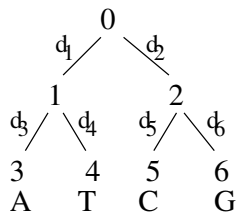
$$L_{k,s} = \begin{cases} 1 & \text{falls von } k = S \\ 0 & \text{sonst} \end{cases}$$

- Rekursionsgleichung:

– k ... interner Knoten

$$L_{k,S} = \left(\sum_{S'} P_{SS'}(d_i) \cdot L_{k',S'} \right) \left(\sum_{S''} P_{SS''}(d_j) \cdot L_{k'',S''} \right)$$

- Beispiel:



– Initialisierung:

$$L_{3,A} = 1, L_{3,C} = 0, L_{3,G} = 0, L_{3,T} = 0$$

$$L_{4,T} = 1, L_{4,C} = 0, L_{4,G} = 0, L_{4,A} = 0$$

$$L_{5,C} = 1, L_{5,A} = 0, L_{5,G} = 0, L_{5,T} = 0$$

$$L_{6,G} = 1, L_{6,C} = 0, L_{6,A} = 0, L_{6,T} = 0$$

–

$$\begin{aligned} L_{1A} &= \left(\sum_{S'} P_{AS'}(d_3) \cdot L_{3,S'} \right) \left(\sum_{S''} P_{AS''}(d_4) \cdot L_{4,S''} \right) \\ &= [P_{AA}(d_3) \cdot 1][P_{AT}(d_4) \cdot 1] \end{aligned}$$

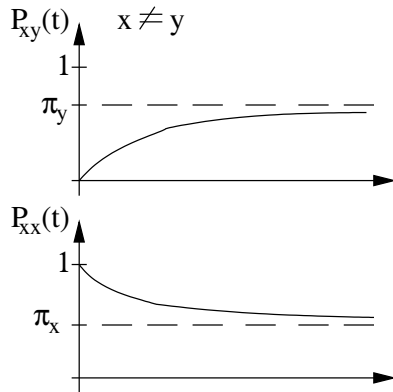
– allgemein:

$$L_{1,S_1} = P_{S_1A}(d_3) \cdot P_{S_1T}(d_4)$$

$$L_{2,S_2} = P_{S_2C}(d_5) \cdot P_{S_2G}(d_6)$$

$$L_{0,S_0} = \left(\sum_{S_1} P_{S_0S_1}(d_1) \cdot L_{1,S_1} \right) \left(\sum_{S_2} P_{S_0S_2}(d_2) \cdot L_{2,S_2} \right)$$

- Bestimmung von $P(t)$



– Eigenschaften:

Markoveigenschaft $\rightarrow P(t + s) = P(t) \cdot P(s)$

Grenzwert: $\lim_{t \rightarrow \infty} P(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \mathbf{I}$

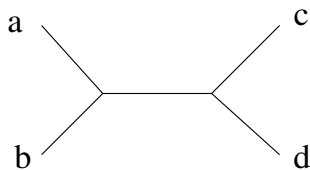
I ... Einheitsmatrix

9.3 Quartet Puzzling

- Kombination der Quartets

Satz: Sei t ein Baum, Q die Menge dieser Quartet trees, die t impliziert. Dann lä"st sich t eindeutig aus Q rekonstruieren!

Def.: Sei $t =$



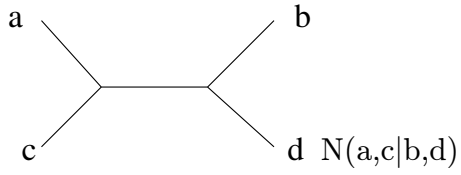
ein Quartet. Dann definiere dies $N(a,b | c,d)$

- Beispiel: Baum und abgeleitete Quartets

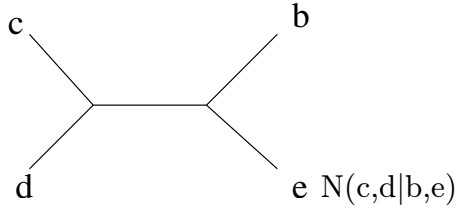
5 taxa... $\{a,b,c,d,e\}$

$\binom{5}{4}$ Teilmengen:

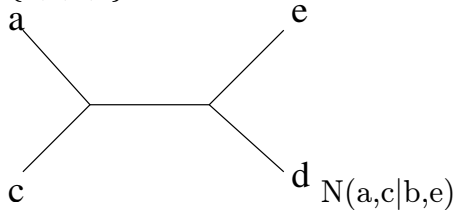
1. {a,b,c,d}:



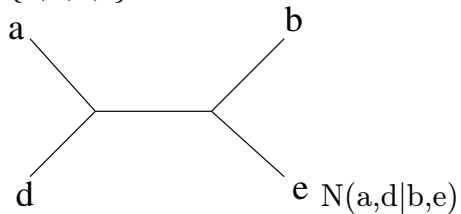
2. {b,c,d,e}:



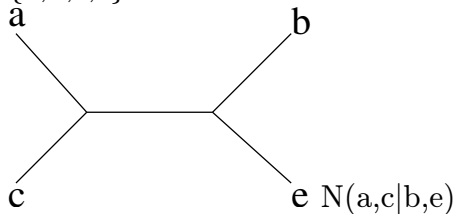
3. {a,c,d,e}:



4. {a,b,d,e}:



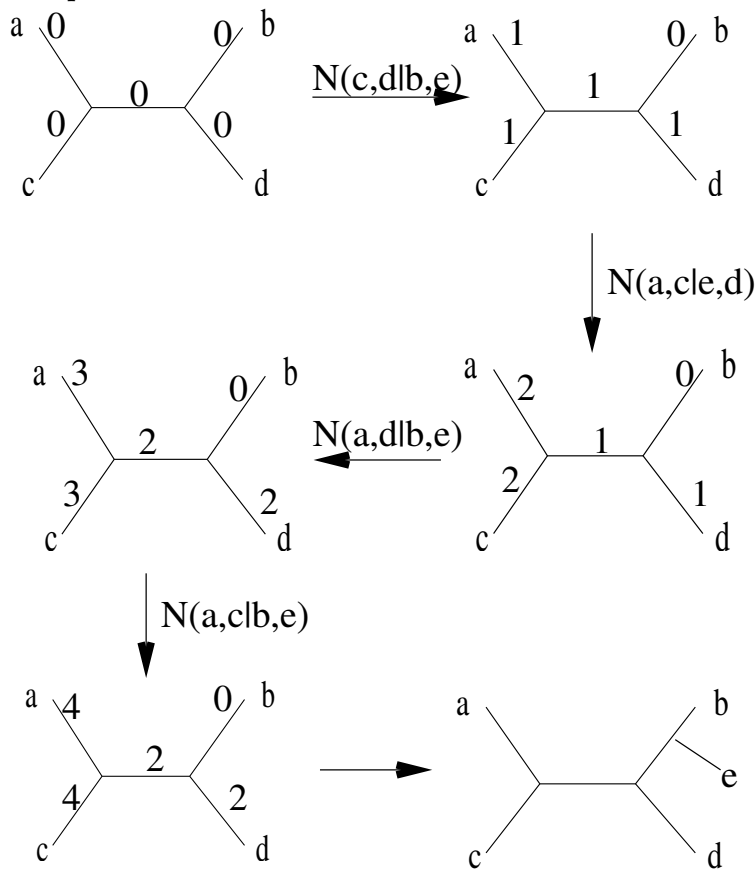
5. {a,b,c,e}:



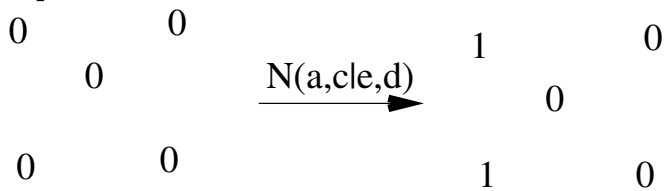
- Quartet Puzzling Algo:
 - Beginne mit 1 Quartet:
 - (zufällige Auswahl)
 - Füge der Reihe nach neue Taxa hinzu.

- * Setze alle Kantenmarkierung auf 0.
- * Betrachte alle Kanten an die neues taxon x hinzugefügt werden kann.
- * Betrachte alle $N(\dots)$ -Relationen, die x enthalten.
- * Addiere 1 zu Kantenbeschreibung, falls $N(,)$ verletzt.
- * Wähle minimale Kante.

– Beispiel:



– Beispiel 2:



– Bemerkung:

1. Satz gilt nur für Quartetts aus B^{aume} \Rightarrow Quartett Puzzling ohne Kante mit 0
2. Quartetts über Maximum Likelihood