Bioinformatik

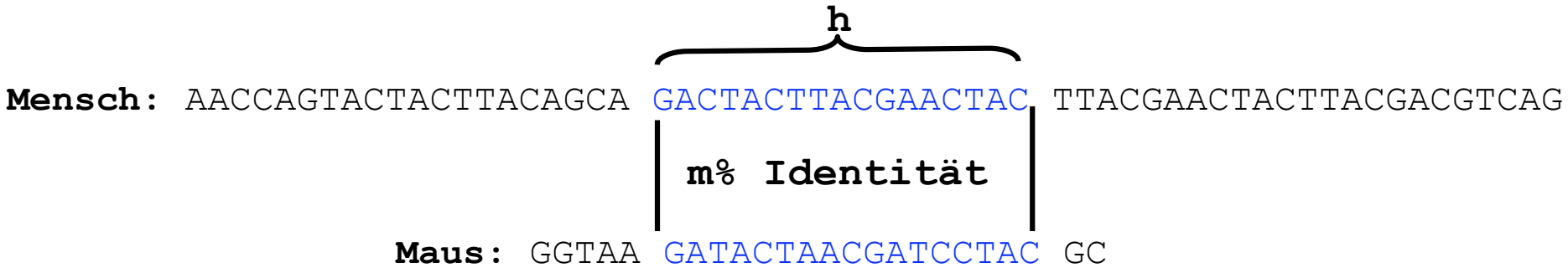
Phylogenetische Algorithmen

Ulf Leser / Silke Trißl

Wissensmanagement in der Bioinformatik

BLAT Szenario

- Vergleich einer Maus-cDNA Q mit einer humanen cDNAs
- Hintergrundwissen über Menschen und Mäuse
  - Wenn es eine homologe Teilsequenz gibt, wird diese im Schnitt zu m% identisch sein und eine durchschnittliche Länge von h haben
  - Frameshifts (Insertions/Deletions) sehr unwahrscheinlich
- Frage
  - Wie lang müssen Seeds aus Q sein, damit wir mit sehr hoher Wahrscheinlichkeit mindestens einen Treffer in einer homologen Region finden, wenn es diese gibt?



BLAT – Statistik

- Ziel: Minimales k und Anzahl von Hits abschätzen
  - m: Erwartete Anzahl Matches in zwei Sequenzen
    - Z.B.: 99% für Maus-Mensch cDNAs, 90% für Proteine
  - h: Durchschnittliche Länge homologer Regionen
  - g: Größe der Datenbank (in Basen)
  - q: Länge der Querysequenz
  - a: Größe des Alphabets (DNA oder Protein)
- Berechnung
  - Wahrscheinlichkeit, dass ein beliebiges k-mer aus der Suchsequenz mit seinem Gegenstück in der homologen Datenbanksequenz perfekt matched
    - $p_1 = m^k$
  - Wahrscheinlichkeit, dass mindestens ein nicht-überlappendes k-mer aus T mit dem entsprechenden k-mer in Q perfekt matched (wenn Q,T homolog)
    - $p = 1-(1-p_1)^z = 1-(1-m^k)^z$

Trefferwahrscheinlichkeiten

Table 3. Sensitivity and Specificity of Single Perfect Nucleotide K-mer Matches as a Search Criterion

	7	8	9	10	11	12	13	14
A. 81%	0.974	0.915	0.833	0.726	0.607	0.486	0.373	0.314
83%	0.988	0.953	0.897	0.815	0.711	0.595	0.478	0.415
85%	0.996	0.978	0.945	0.888	0.808	0.707	0.594	0.532
87%	0.999	0.992	0.975	0.942	0.888	0.811	0.714	0.659
89%	1.000	0.998	0.991	0.976	0.946	0.897	0.824	0.782
91%	1.000	1.000	0.998	0.993	0.981	0.956	0.912	0.886
93%	1.000	1.000	1.000	0.999	0.995	0.987	0.968	0.957
95%	1.000	1.000	1.000	1.000	0.999	0.998	0.994	0.991
97%	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.999

- q=100, m und k variabel (a und g hier nicht notwendig)
- Werte sind Wahrscheinlichkeit, dass mindestens ein perfekter Match in der Region vorkommt
- Ein Match reicht – Extensionsphase wird die ganze Region finden
- Beispiel
  - Bei erwarteter Sequenzähnlichkeit von m=97% findet man in einer homologen Region T von 100 Basen praktisch immer einen perfekten Match der Länge 13 für einen Substring der Länge q

Falsch-positive Treffer

- Aber: Wie viele k-mere matchen zufällig?
  - Abhängig von g, q und a
  - $F = (q-k+1) * (g/k) * (1/a)^k$
    - $(1/a)^k$: Alle Zeichen eines k-mers matchen per Zufall
    - (g/k): Anzahl nicht-überlappender k-mere in DB
    - (q-k+1): Anzahl (überlappender) k-mere in Query

B. K	7	8	9	10	11	12	13	14
F	1.3e+07	2.9e+06	635783	143051	32512	7451	1719	399

$a=4,\ g=3*10^9,\ q=500$

- Falsch-positive werden in Extensionsphase ausgesiebt
- Trade-Off
  - Hohe k-Werte: Wenig falsch-positive, aber eventuell fehlende echte Hits
  - Niedrige k-Werte: Viele falsch-positive, aber weniger falsch-negative

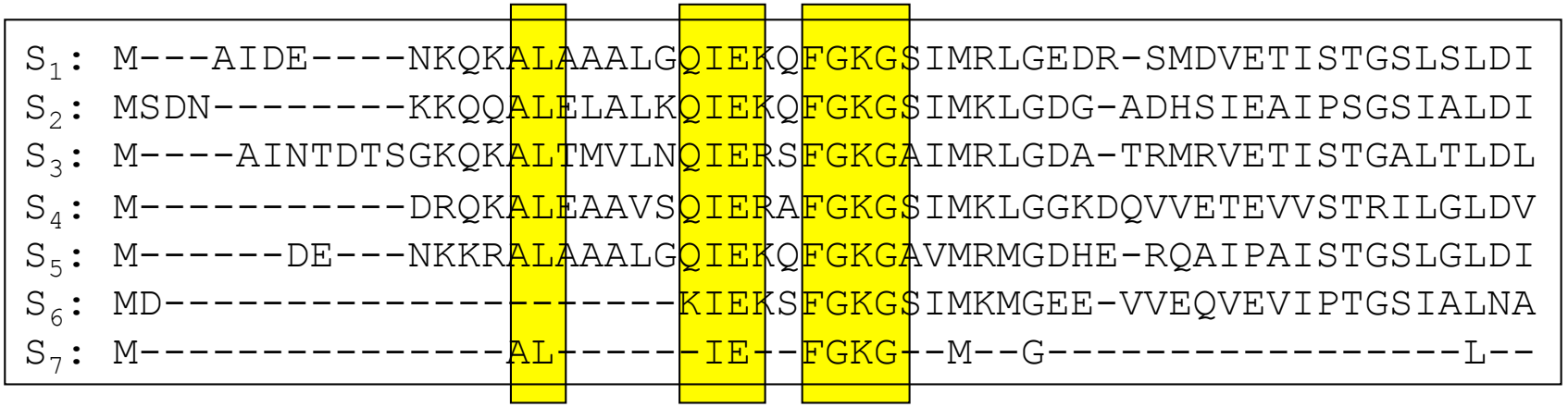
Weitere Möglichkeiten

- Suche mit mehreren (perfekten) Hits
  - Idee von BLAST-2
  - Wir sparen uns die Formeln
  - Drastische Verringerung der erwarteten Anzahl falsch-positiver Hits

Table 7. Sensitivity and Specificity of Multiple (2 and 3) Perfect Nucleotide K-mer Matches as a Search Criterion

	2,8	2,9	2,10	2,11	2,12	3,8	3,9	3,10	3,11	3,12
A. 81%	0.681	0.508	0.348	0.220	0.129	0.389	0.221	0.112	0.051	0.021
83%	0.790	0.638	0.475	0.326	0.208	0.529	0.339	0.193	0.099	0.045
85%	0.879	0.762	0.615	0.460	0.318	0.676	0.487	0.313	0.180	0.093
87%	0.942	0.866	0.752	0.611	0.461	0.809	0.649	0.470	0.305	0.177
89%	0.978	0.940	0.868	0.761	0.625	0.910	0.801	0.648	0.476	0.314
91%	0.994	0.980	0.947	0.884	0.787	0.969	0.914	0.815	0.673	0.505
93%	0.999	0.996	0.986	0.962	0.912	0.993	0.976	0.933	0.851	0.722
95%	1.000	1.000	0.998	0.993	0.979	0.999	0.997	0.987	0.961	0.902
97%	1.000	1.000	1.000	1.000	0.999	1.000	1.000	0.999	0.997	0.987
B. N,K	2,8	2,9	2,10	2,11	2,12	3,8	3,9	3,10	3,11	3,12
F	524	27	1.4	0.1	0.0	0.1	0.0	0.0	0.0	0.0

MSA – Beispiel



- Uns interessieren „möglichst gute“ MSAs
  - Intuition
    - Möglichst wenig Spalten – wenig Leerzeichen
    - Möglichst homogene Spalten – hohe Übereinstimmung
  - Exakte Definition später
- MSAs erfassen das Gemeinsame verschiedener Sequenzen

MSA Überblick

- Weg über m-dimensionale Substitutionsmatrizen nicht gangbar
- Verschiedene alternative Vorschläge für Zielfunktionen existieren
  - Maximiere die Summe aller paarweisen Alignments
  - Maximiere die Summe der Alignments jeder Sequenz zu einer Consensussequenz (Center-Star)
  - Maximiere die Summe der Alignments folgend dem phylogenetischen Baum der Sequenzen

Definitionen

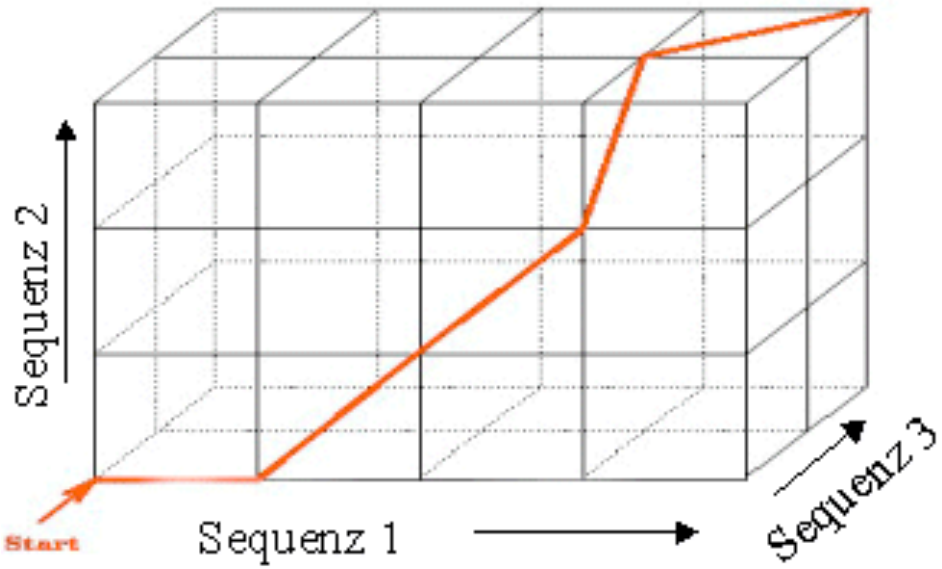
- Definition
  - *Gegeben ein MSA M für Sequenzen $S_1,...,S_k$. Das durch M induziertes Alignment für zwei Sequenzen S_i und S_j ist das folgende:*
    - *Entferne aus M alle Zeilen außer i und j*
    - *Entferne alle Spalten, die in i und j ein Leerzeichen enthalten*
  - *Gegeben ein MSA M für Sequenzen $S_1,...,S_k$. Der Sum-Of-Pairs Score für M (SP-Score) ist die Summe aller Alignmentscores der durch M induzierten paarweisen Alignments*
  - *Das SP-Alignment Problem für Sequenzen $S_1,...,S_k$ sucht das MSA M mit minimalem SP-Score*
- Bemerkung
  - Vergleich aller Sequenzen mit allen anderen Sequenzen – aber entsprechend dem vorgegebenen MSA

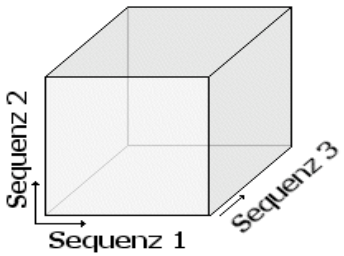
Dynamische Programmierung in k Dimensionen

- k = 2
  - 2-dimensionale Matrix

		0	1	2	3	4	5	6	7
			w	r	i	t	e	r	s
0		0	1	2	3	4	5	6	7
1	v	1	1	2	3	4	5	6	7
2	i	2	2	2	2	3	4	5	7
3	n	3	3	3	3	3	4	5	6
4	t	4	4	4	4	3	4	5	6
5	n	5	5	5	5	4	4	5	6
6	e	6	6	6	6	5	4	5	6
7	r	7	7	6	7	6	5	4	5

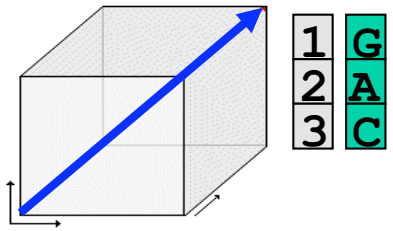
- k = 3
  - 3-dimensionale Matrix

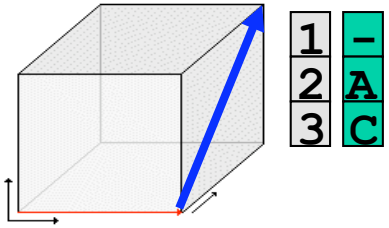


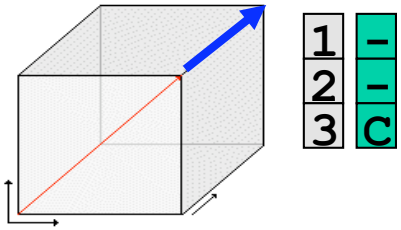


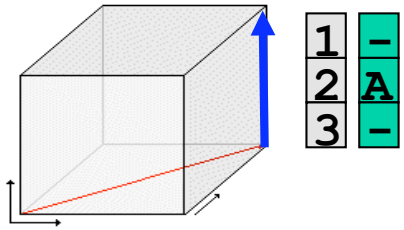
Dyn Prog. für SP-MSA

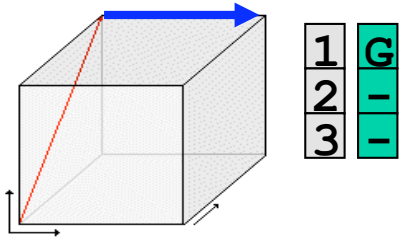
- d(i-1,j-1,k-1)
- d(i,j-1,k-1)
- d(i,j,k-1)
- d(i,j-1,k)
- d(i,j,k)
- d(i-1,j-1,k)
- d(i-1,j,k-1)

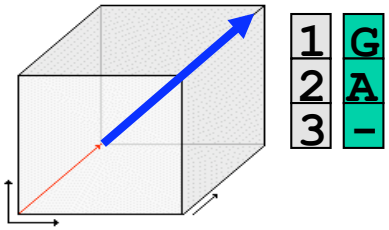


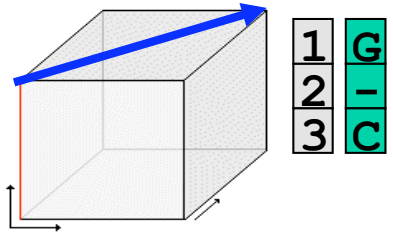












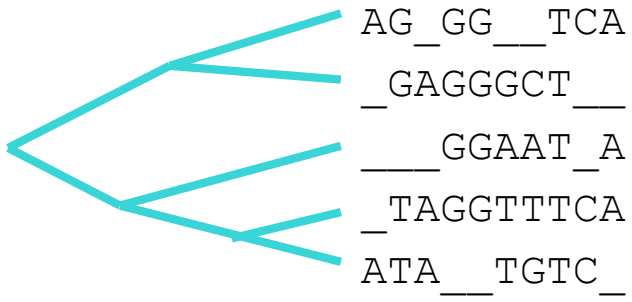
Formal

- Wir nehmen ein einfaches Kostenmodell (I/D/R=1, M=0)
- Theorem
  - *Gegeben Sequenzen S_1, S_2, S_3.*
    - *Sei d(i,j,k) der Score des SP-optimalen Alignments der Strings $S_1[1..i]$, $S_2[1..j]$, $S_3[1..k]$*
    - *Sei $c_{ij} = 0$, wenn $S_1(i) = S_2(j)$, sonst 1*
    - *Sei $c_{ik} = 0$, wenn $S_1(i) = S_3(k)$, sonst 1*
    - *Sei $c_{jk} = 0$, wenn $S_2(j) = S_3(k)$, sonst 1*
  - *Dann berechnet sich d(i,j,k) als*

$$
d(i,j,k) = \min \begin{cases}
d(i-1, j-1, k-1) + c_{ij} + c_{ik} + c_{jk} \\
d(i-1, j-1, k) + c_{ij} + 2 \\
d(i-1, j, k-1) + c_{ik} + 2 \\
d(i, j-1, k-1) + c_{jk} + 2 \\
d(i-1, j, k) + 2 \\
d(i, j-1, k) + 2 \\
d(i, j, k-1) + 2
\end{cases}
$$

MSA praktisch unlösbar?

- SP-Score für mehr als eine Handvoll Sequenzen nennenswerter Länge nicht berechenbar
- Außerdem: SP berechnet nicht die Menge an notwendiger Evolution



- Viele andere Zielfunktionen
  - Center-Star, MSA entlang des phylogenetischen Baums
- Viele Heuristiken (Branch&Bound, iterative, greedy, …)

Progressives Alignment

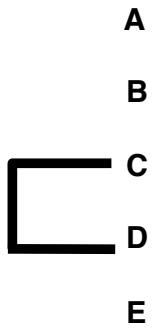
- Grundproblem des SP-MSA
  - Ständige Betrachtung aller Sequenzen
  - Das sind zu viele Möglichkeiten = Dimensionen
- Grundidee der progressiven Verfahren
  - Berechne zunächst MSA für viele kleine Teilmengen von Sequenzen
  - Baue das Gesamt-MSA aus den Teil-MSA
- Das wirft Fragen auf
  - Wie wähle ich die Teilmengen?
  - Wie „aligniert" man zwei MSA?
    - Wir können MSA (Profil) und Sequenz, aber zwei MSA?
  - In welcher Reihenfolge verschmilzt man die Teil-MSA?

CLUSTAL W: Grundaufbau

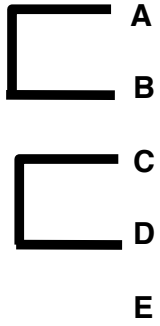
- Gegeben k Sequenzen
- Drei Schritte
  - Berechne alle paarweisen Alignmentscores
  - Konstruiere „Guide Tree" durch hierarchisches Clustering
  - Berechne und verschmelze Teil-MSA gemäß dem Guide Tree
- Idee dahinter
  - Aligniere erst sehr ähnliche Sequenzen – Signale werden verstärkt
  - Werden z.B. zwei sehr verschiedene Cluster von Sequenzen betrachtet, berechnet CLUSTAL automatisch erst zwei (homogene) MSA und verschmilzt diese am Ende
    - Hohe Chance, dass konservierte Blöcke erhalten bleiben
  - Außenseiter kommen erst spät dazu und können die Blockstruktur nicht mehr stören
  - Orientierung an der „tatsächlichen" Entstehungsgeschichte, dem phylogenetischen Baum

Konstruktion des Guide Trees

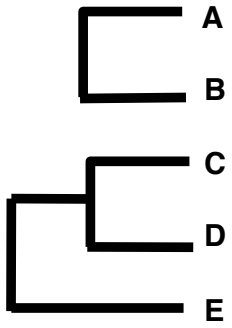
	A	B	C	D	E
A		17	59	59	77
B			37	61	53
C				13	41
D					21

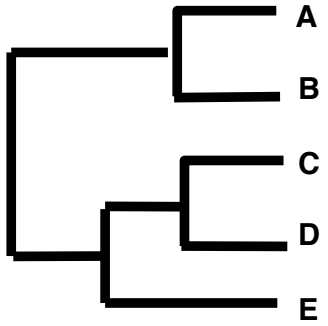


	A	B	E	CD
A		17	77	59
B			53	49
E				31



	E	CD	AB
E		31	65
CD			54





Schritt 3: Progressive MSA Generierung

- Berechnung paarweiser Alignments in der Reihenfolge des Guide Trees
- Alignment eines MSA M_1 mit einem MSA M_2
  - Dynamische Programmierung mit linearem Gapscore
  - Wert eines Mismatches/Matches ist der Durchschnittsscore aller Paare mit einem Zeichen aus M_1 und dem anderen aus M_2
  - Gaps werden mit dem schlechtesten Score der verwendeten Substitutionsmatrix bestraft
  - Bei k Sequenzen sind das maximal k/2*k/2 = $O(k^2)$ Scores
- Beispiel

```
A   ...P...
B   ...G...
C   ...P...

D   ...A...
E   ...A...
F   ...Y...
```

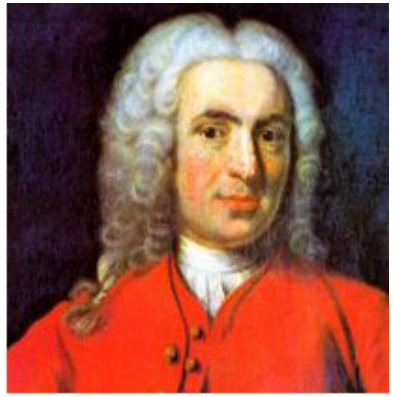
- Score dieser Spalte
  - `(2*s(P,A)+s(P,Y)+`
    `2*s(G,A)+s(G,Y)+`
    `2*s(P,A)+s(P,Y) )/9`

Inhalt dieser Vorlesung

- Stammbäume
- Ultrametriken
- Phylogenetische Algorithmen
  - Distanzbasiert: UPGMA
  - Neighbor Joining
  - Maximum Parsimony

Stammbäume versus Klassifikation

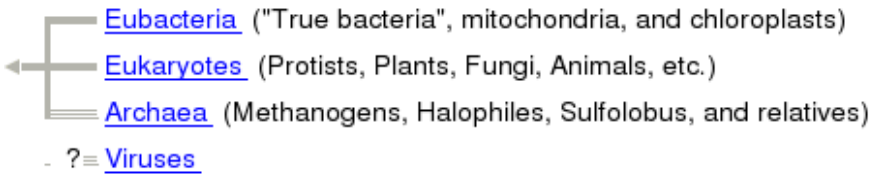
- Carl Linnaeus, ca. 1740: *Systema Naturae*
- Annahme: Arten verändern sich nicht
  - Prä-Darwin
- Einteilung der Lebewesen in
  - Kingdoms - classes – orders –families – genera-species
  - SKOFGA (Stamm, Klasse, Ordnung, Familie, Gattung, Art)
- Innere Knoten einer Klassifikation sind abstrakt
  - Es gab nie das Tier „Säugetier" oder „Schädelknochen"
  - Vielleicht gab es ein Urwesen aller Säugetiere
    - Für eine Klassifikation ist das aber nicht notwendig



'Bridegroom portrait' of Linnaeus,
Clarisſ: LINNÆI, M.D.
METHODUS plantarum SEXUALIS
in SISTEMATE NATURÆ
deſcripta

Linnaean sexual system
Illustration by Georg Dionysius Ehret of Linnaeus' sexual system (1736)

Klassifikation



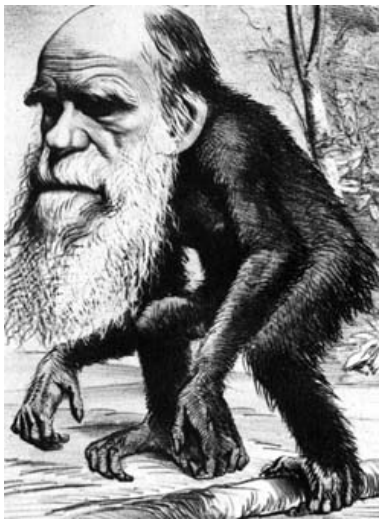
- Eukaryoten
- Tiere
- … (diverse Zwischenstufen)
- Chraniata (Schädelknochen)
- Vertebraten (Wirbeltier)
- … (viele Zwischenstufen)
- Mammals (Säugetiere)
- Eutheria (Placenta)
- Primaten (Affen)
- Catarrhini
- Hominidae (Mensch, Schimpanse, Orang-Utan, Gorilla)
- Homo (erectus, sapiens ...)
- Homo Sapiens

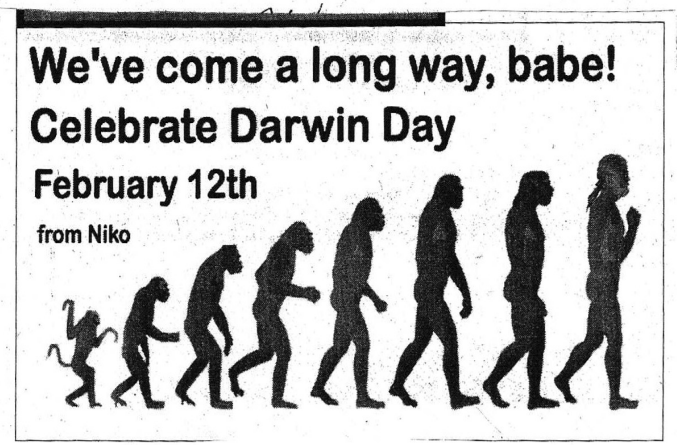
Popular Groups on the Tree of Life

Eubacteria
Eukaryotes
- **Animals**
  - Echinoderms (sea urchins, starfish, sea cucumbers, etc)
  - **Vertebrates (fish etc.)**
    - **Terrestrial Vertebrates**
      - Frogs
      - Salamanders
      - Turtles
      - **Dinosaurs**
        - Modern Birds
      - Mammals
    - Teleost fish
  - Cnidaria (jellyfish, anemones, corals, etc.)
  - Annelida (segmented worms)
  - Cephalopoda (octopods, squids, etc.)
  - **Arthropoda**
    - **Insects**
      - Dragonflies and Damselflies
      - Lice
      - True Bugs
      - Beetles
      - Wasps, Bees, and Ants
      - Flies
      - Butterflies and Moths
      - Crickets, Katydids, and Grasshoppers
    - **Arachnids**
      - Spiders
      - Mites
      - Scorpions
- Fungi
- **Green Plants**
  - Ferns
  - Flowering Plants

Stammbäume

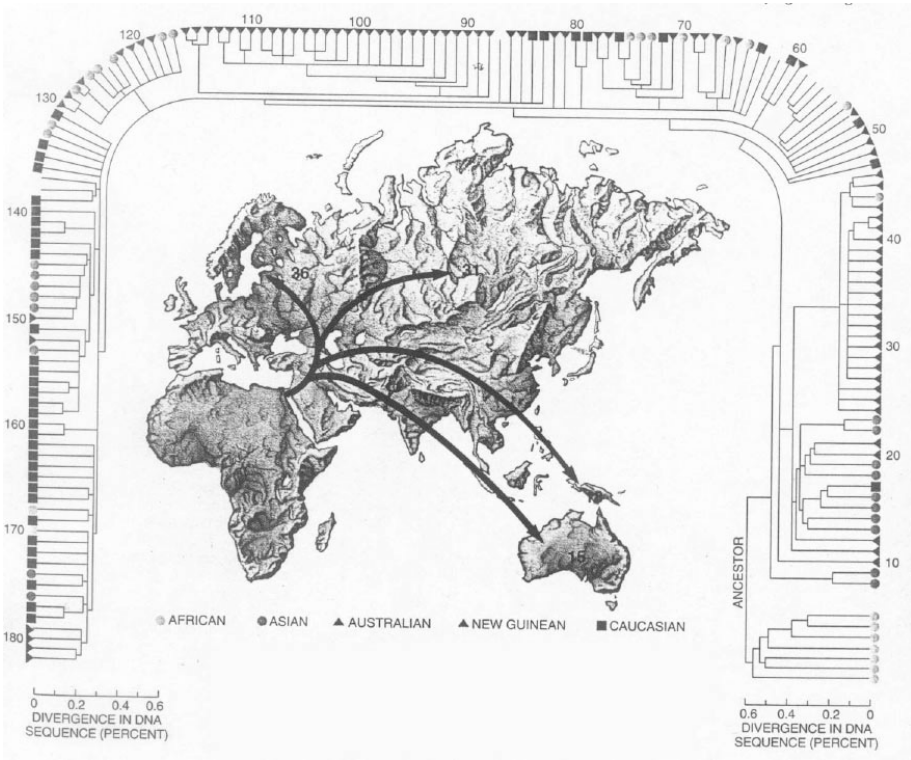
- Charles Darwin: „The origin of species“ (1859)
  - Arten sind nicht unveränderlich, sondern unterliegen im Laufe der Zeit einem Wandel
  - „Survival of the fittest“
  - Damals war noch unklar, was sich eigentlich wandelt
- Stammbäume (Abstammungsbäume)
  - Ergeben sich aus der Annahme der Evolution
  - Jeder Knoten in einem Stammbaum hat einmal als Art existiert
  - Knoten im Baum (also Spezies) heißen Taxa
- Was definiert eine Spezies?





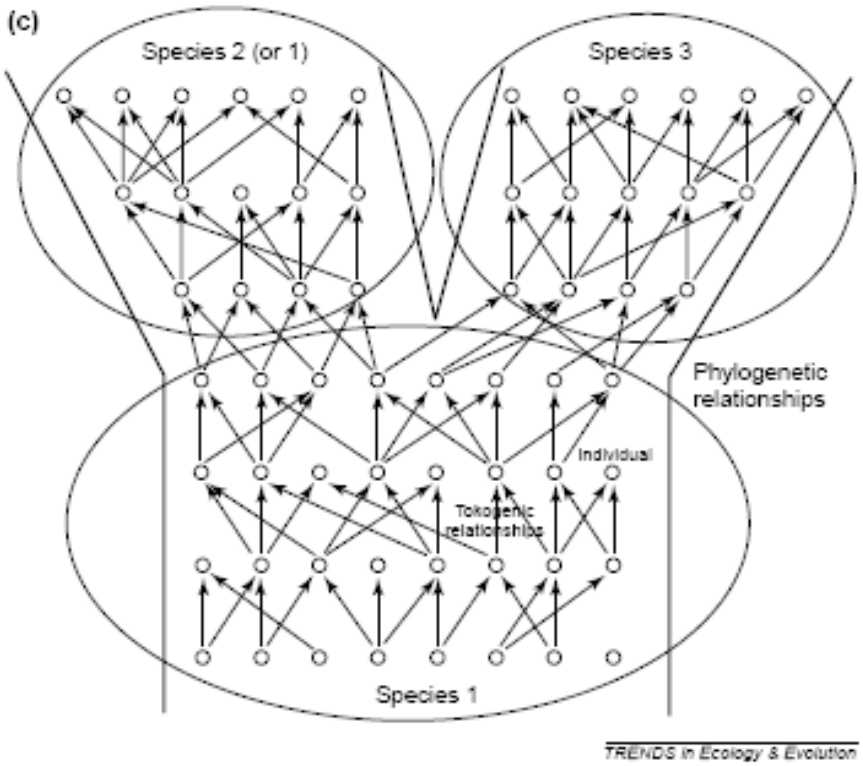
Moderne Stammbaumberechnung

- *Molecular phylogeny*
- Mendel + Darwin: Das Erbgut unterliegt dem Wandel
- Berechnung von Stammbäumen aus molekularen Daten
  - Zuckerkandl und Pauling, 1965
- Berechnung aufgrund von DNA oder Proteinsequenzen
- Annahme: Evolution verläuft in kleinen Schritten
- Wenn sich Sequenzen ähnlich sind, sind die Spezies evolutionär eng verwandt
  - Denn zufällige Ähnlichkeit ist zu unwahrscheinlich



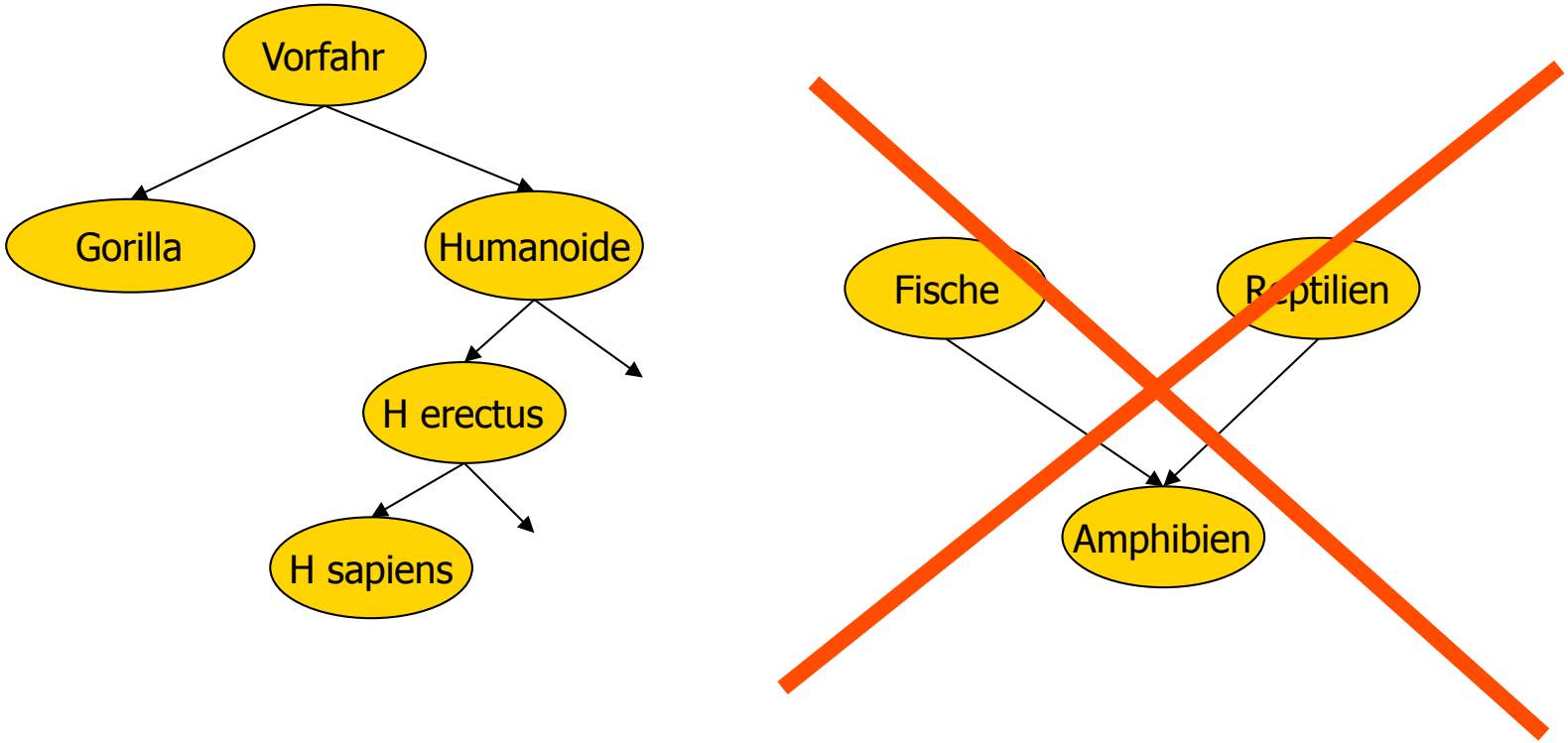
Arten von Stammbäumen

- Individuelle Abstammung
  - Stammbäume, Ahnentafeln
  - Natürlich kein Baum: Zwei Eltern
  - Rekombination
- Speziesstammbäume
  - Ein Baum, wenn Spezies nicht verschmelzen können
  - Sprachen verschmelzen
- Gene Trees
  - Geschichte eines Sequenzabschnitts
    - Nicht leicht zu definieren
  - Baumförmig, wenn Gene nicht verschmelzen
  - Aber: 2 Allele jedes Gens vorhanden (Besser: Haplotype Tree)



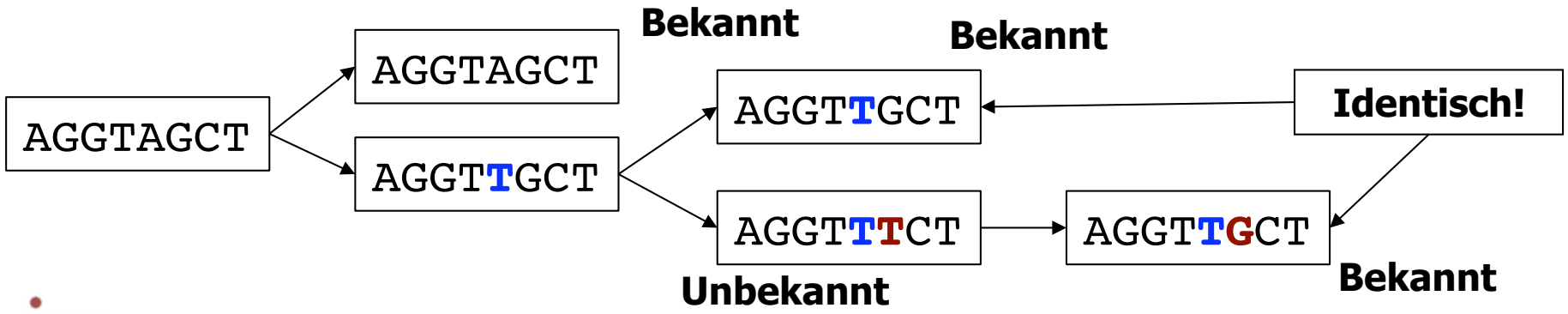
Artenbildung

- Arten entstehen durch Veränderungen aus einer anderen Art



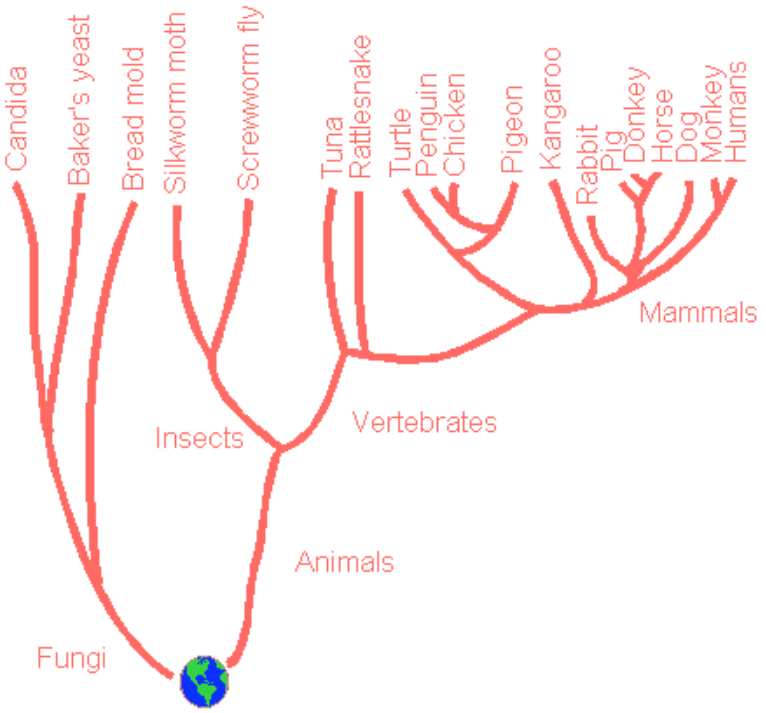
Daten

- Man kennt nur die DNA existierender Arten
  - (Einschränkung: Sequenzen von Neantertalern, …)
- Zwei mögliche Ziele
  - Rekonstruktion des wahrscheinlichsten Stammbaums der Arten
  - Rekonstruktion der wahrscheinlichsten Ur-DNA und aller Zwischenstufen
- Den tatsächlichen Stammbaum kann man nicht berechnen
  - Man kennt die ausgestorbenen Arten nicht
  - Man kann ausgestorbene Mutationen nicht erkennen
  - Man kann Doppelmutationen nicht erkennen



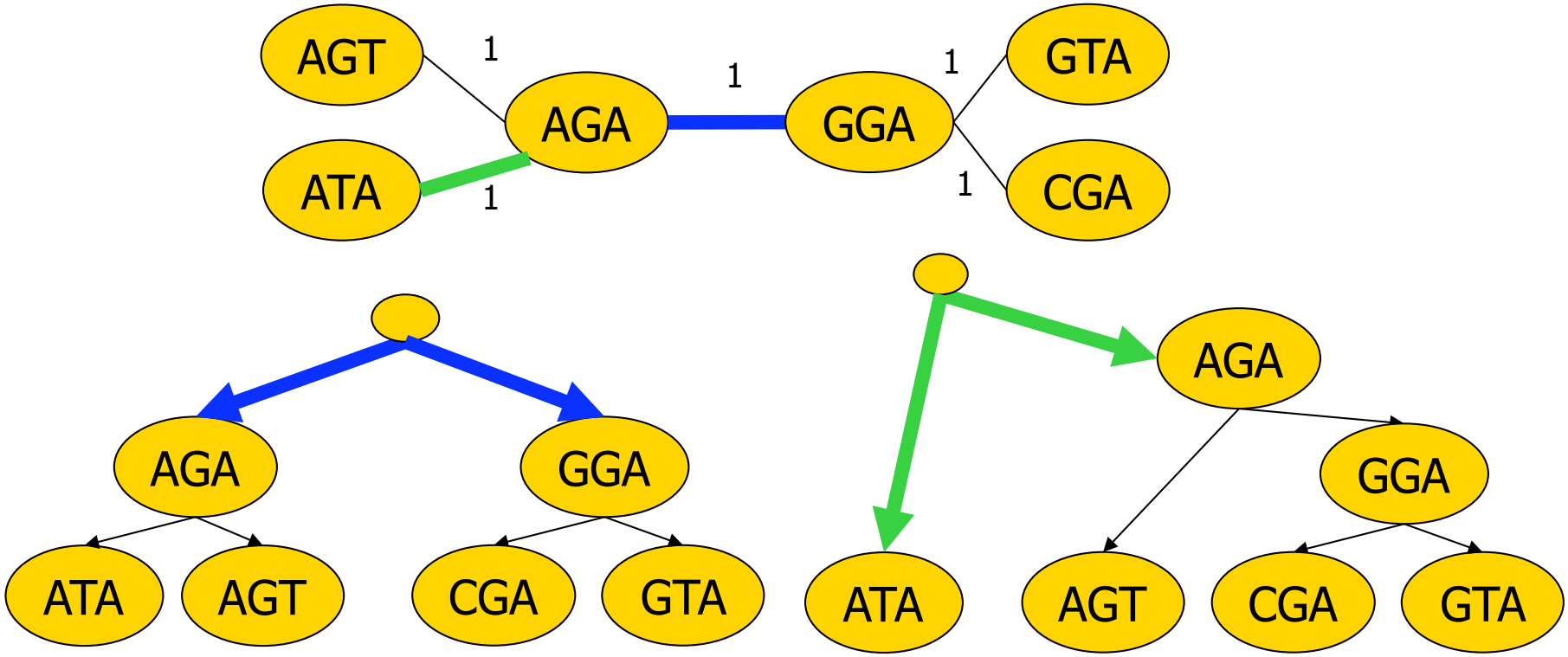
Bäume

- Knoten = Arten
- Blätter = Lebende Arten
- Kanten
  - Länge kann (aber muss nicht) mit zeitlicher Entfernung korrelieren
  - Scaled trees
- Jeder Knoten hat exakt einen Vater
- Eine Wurzel
  - Nicht immer (gleich)
- Binäre Bäume
  - Unproblematisch (gleich)
  - Reihenfolge der Kinder ist egal
- Viele Visualisierungsvarianten



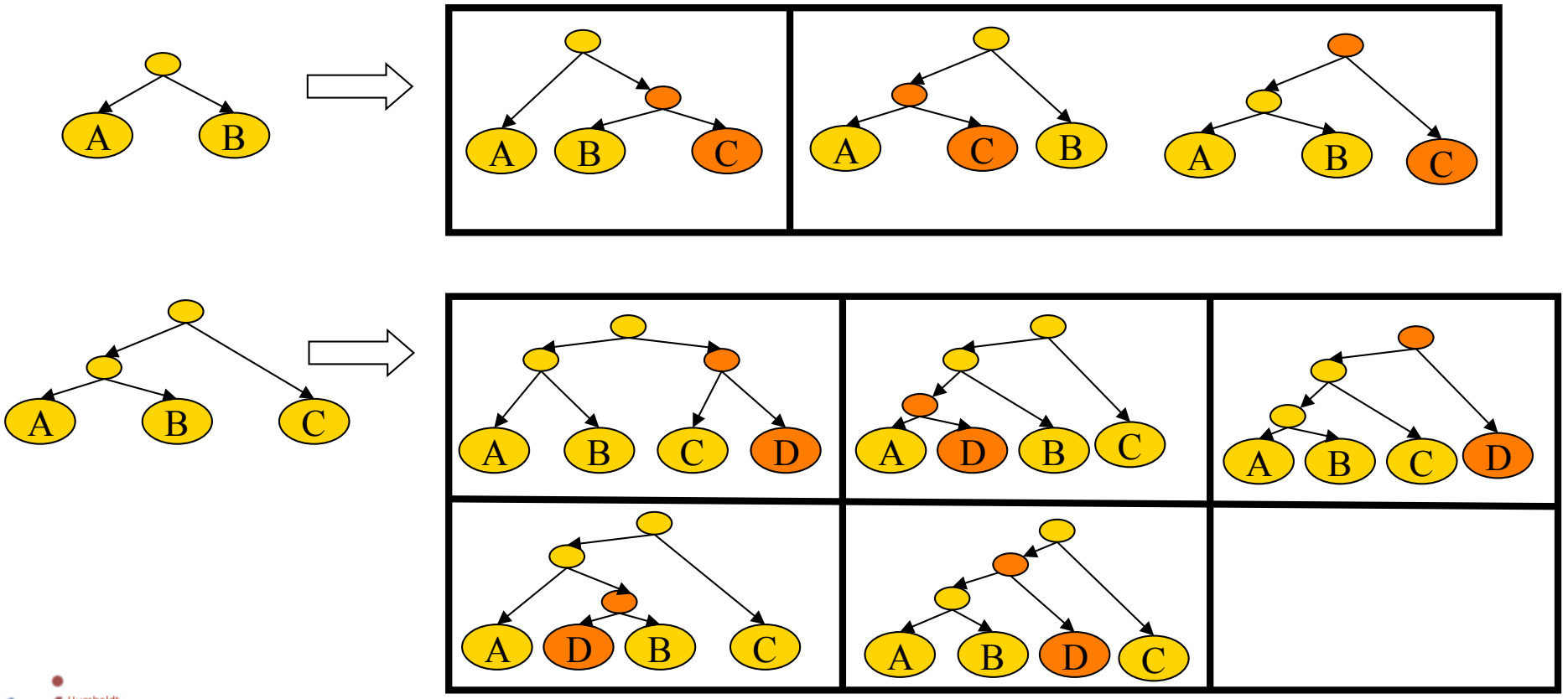
Bäume ohne Wurzeln

- Kanten symbolisieren Veränderungen
- Wir können nur Blätter vergleichen
- Damit ist idR keine Richtung der Entwicklung berechenbar



Wie schwierig wird das?

Wie viele binäre, ungeordnete Bäume für n Spezies gibt es?



Ergebnis

- Sei t(n) die Zahl binärer Bäume mit n Blättern

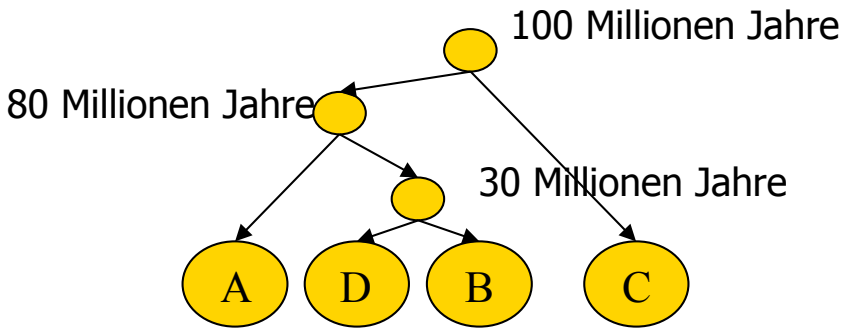
$$t(n) = t(2) * t(3) * t(4) * \ldots * t(n-1) =$$
$$= 1 * 3 * 5 * \ldots * (2(n-1)-1) =$$
$$= \frac{(2n-3)!}{2 * 4 * 6 * \ldots * (2n-4)} =$$
$$= \frac{(2n-3)!}{2\left(\frac{2}{2}\right) * 2\left(\frac{4}{2}\right) * 2\left(\frac{6}{2}\right) * \ldots * 2\left(\frac{2n-4}{2}\right)} =$$
$$= \frac{(2n-3)!}{2 * (1) * 2 * (2) * 2 * (3) * \ldots * 2(n-2)} =$$
$$= \frac{(2n-3)!}{2^{n-2} * (n-2)!}$$

1	1
2	1
3	3
4	15
5	105
6	945
7	10.395
8	135.135
9	2.027.025
10	34.459.425
11	654.729.075
12	13.749.310.575
13	316.234.143.225
14	7.905.853.580.625
15	213.458.046.676.875
16	6.190.283.353.629.370
17	191.898.783.962.511.000
18	6.332.659.870.762.850.000
19	221.643.095.476.700.000.000
20	**8.200.794.532.637.890.000.000**

Inhalt dieser Vorlesung

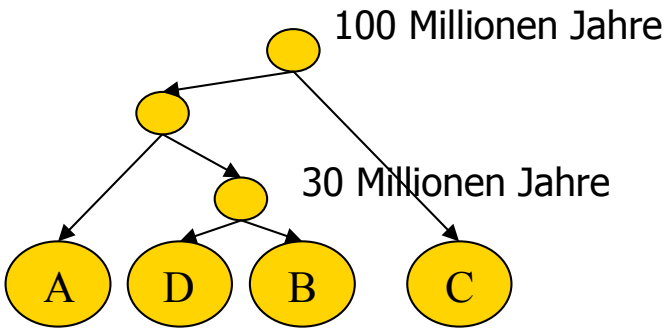
- Distanzbasierte Phylogenie
- Ultrametriken und UPGMA
- Additive Bäume und Neighbor Joining

Voraussetzungen



- An innere Knoten eines Stammbaum kann man den Speziationzeitpunkt schreiben („branch points")
- Wenn die Molecular Clock Theory gilt
  - Ist die Menge an Veränderungen auf einer Kante proportional zu der verstrichenen Zeit
  - Damit proportional zur Länge der Kante
  - Damit ist der Editabstand zweier Knoten (=Evolution dazwischen) proportional zur Summe der Editabstände beider Knoten zum kleinsten gemeinsamen Vater
  - Damit ist der Editabstand zweier Knoten proportional zur Summe der Kantenlängen zum jüngsten gemeinsamen Vorfahren

Ultrametriken



- Wenn man den Baum und die Zeitpunkte weiß, dann gilt
  - Alle Zahlen auf einem Pfad von der Wurzel zu einem beliebigen Blatt nehmen strikt ab
  - Der Zeitpunkt der Aufspaltung ist ein Abstandsmaß für zwei Arten
    - Für Blätter X, Y sei d(X,Y) das Label des kleinsten gemeinsamen Vorfahren
    - Im Beispiel: d(A,B)=80, d(B,C)=100, d(A,D)=80
  - Das ist eine Metrik
    - d(X,X)=0, d(X,Y)>0, d(X,Y)=d(Y,X), und d(X,Y)≤d(X,Z)+d(Z,Y)
  - Es ist sogar eine Ultrametrik (gleich)

Alles ganz einfach?

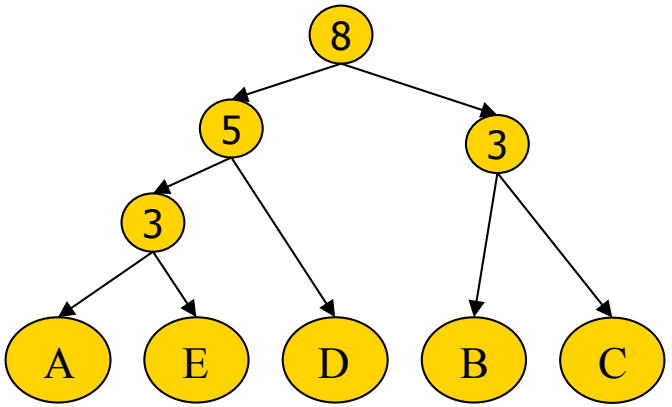
- Wir zeigen gleich
  - Schreibt man diese Abstände in eine Ähnlichkeitsmatrix, hat diese bestimme Eigenschaften
  - Jede Matrix mit diesen Eigenschaften entspricht genau einem Baum und umgekehrt
  - Den Baum kann man aus der Matrix effizient berechnen
- Also ist das Phylogenie-Problem überhaupt kein Problem?
- Doch
  - Molecular Clock Theory stimmt nicht
  - Editabstand korreliert nicht mit der vergangenen Zeit
    - Selektionsdruck
  - Damit kennen wir die Speziationszeitpunkte nicht
  - Man kann das alles nur approximieren
- Wenn die Molecular Clock Theory aber gelten würde …

Ultrametrische Bäume

- Definition
  Sei T ein Baum und D eine symmetrische Matrix mit n Zeilen und n Spalten. T heißt ultrametrischer Baum für D wenn gilt:
  - *T hat n Blätter, beschriftet mit den Zeilen von D*
  - *Jeder innere Knoten von T hat zwei Kinder und ist mit einem Wert aus D beschriftet*
  - *Auf jedem Pfad von der Wurzel zu einem Blatt in T sind die Zahlen strikt abnehmend*
  - *Für alle Blätter i,j mit $i \neq j$ gilt: der kleinste gemeinsame Vorfahr von i und j ist mit D(i,j) beschriftet*
- Bemerkung
  - Jeder Stammbaum ist ultrametrisch für die Abstandsmatrix mit den Aufsplittzeitpunkten als Abstandsmaß

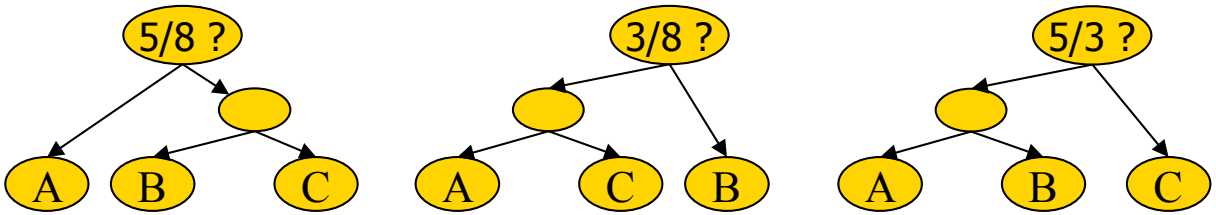
Beispiel

	A	B	C	D	E
A		8	8	5	3
B			3	8	8
C				8	8
D					5
E					



Das geht nicht immer

	A	B	C
A		8	5
B			3
C			



Überlegungen

- Das kann auch nicht immer gehen
  - Matrix hat $(n^2-n)/2$ relevante Zellen
  - Baum hat nur n-1 innere Knoten
  - Eine Matrix, zu der man einen ultrametrischen Baum konstruieren kann, muss also Duplikate enthalten
- *Definition*
  Eine symmetrische Matrix D mit n Spalten und Zeilen ist ultrametrisch, wenn für beliebige Zeilen i, j, k gilt, dass das Maximum von D(i,j), D(j,k) und D(i,k) genau zweimal vorkommt
- Bemerkung
  - Also entweder
    - D(i,j)=D(j,k) und D(i,j)>D(i,k)
    - D(i,j)=D(i,k) und D(i,j)>D(j,k)
    - D(j,k)=D(i,k) und D(j,k)>D(i,j)

Von der Matrix zum Baum und zurück

- Theorem
  Eine symmetrische Matrix D hat einen ultrametrischen Baum gdw. D selber ultrametrisch ist.

  Sei D eine ultrametrische Matrix. Dann gibt es genau einen ultrametrischen Baum T für D.

- Beweis
  - …

Distanzbasierte Algorithmen

- Konstruktion des ultrametrischen Baumes basiert rein auf Distanzmassen
  - Einen ultrametrischen Baum gibt es nicht für alle Matrizen
  - Es gibt weniger empfindliche Verfahren (gleich)
- Die Geschichte einzelner „Sites" wird nicht berücksichtigt
- Solche Algorithmen zur Berechnung von Stammbäumen nennt man distanzbasiert
- Alternative: Merkmalsbasierte Verfahren
  - Beachten jedes einzelne Merkmal (Sequenzposition)
  - Z.B. Perfect Phylogeny; Maximum Parsimony
  - Später

UPGMA - Hierarchisches Clustering

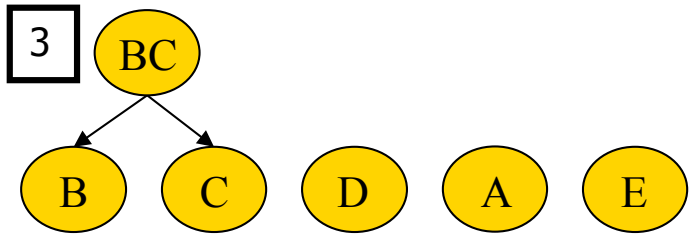
- UPGMA
  - „Unweighted pair group method with arithmetic mean“
  - Anderer Name: Hierarchisches Clustering
- Sehr einfaches und allgemeines Verfahren, kann bei allen möglichen Problemen angewandt werden
- Wenn eine Matrix ultrametrisch ist, dann findet UPGMA den dazugehörenden ultrametrischen Baum
  - UPGMA nimmt die Molecular Clock an – alle Pfade von einem Blatt zur Wurzel haben am Ende die selbe Länge
- Achtung: UPGMA konstruiert immer einen Baum
  - Auch wenn die Matrix nicht ultrametrisch ist

UPGMA Verfahren

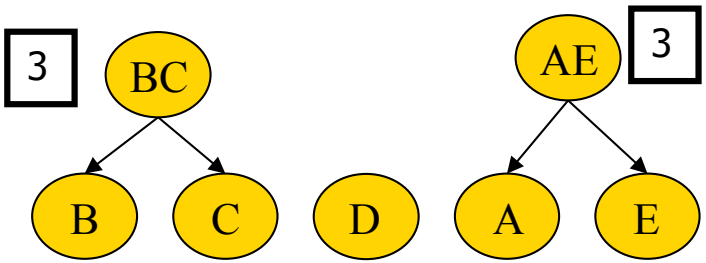
- Gegeben: Distanzmatrix D
- Erzeuge ein „Baumgerüst" mit n Blättern
- Wähle den kleinsten D(i,j) Wert der Matrix und verbinde die Knoten i und j durch einen neuen Knoten (ij) mit Beschriftung D(i,j) und Kanten zu i und zu j
  - Anfangs sind i und j Blätter, später können es auch innere Knoten sein
- Lösche Zeilen und Spalten i und j aus D
- Füge in D eine Zeile und eine Spalte (ij) hinzu mit D(ij,k) = (D(i,k)+D(j,k))/2
- Wiederhole, bis D leer ist

Beispiel

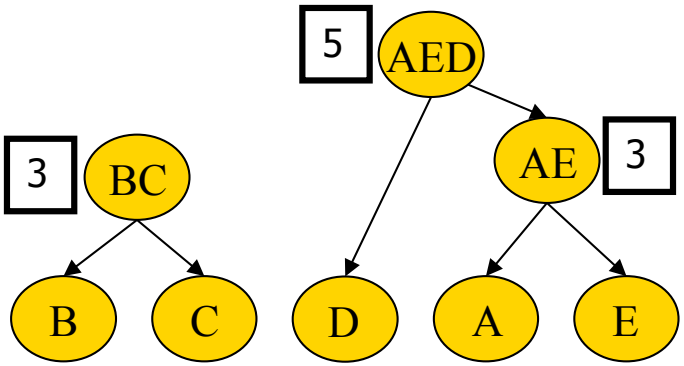
	B	C	D	E
A	8	8	5	3
B		3	8	8
C			8	8
D				5



	BC	D	E
A	8	5	3
BC		8	8
D			5

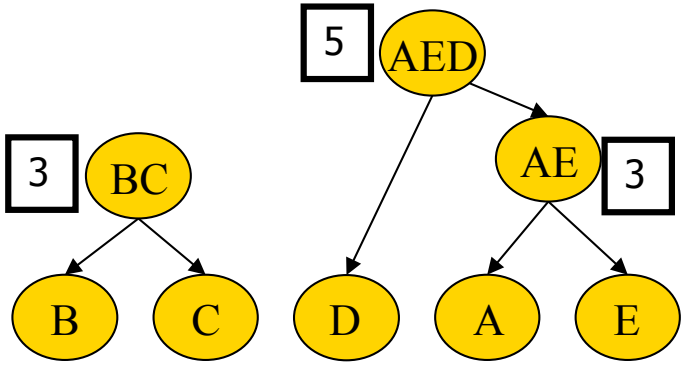


	BC	D
AE	8	5
BC		8

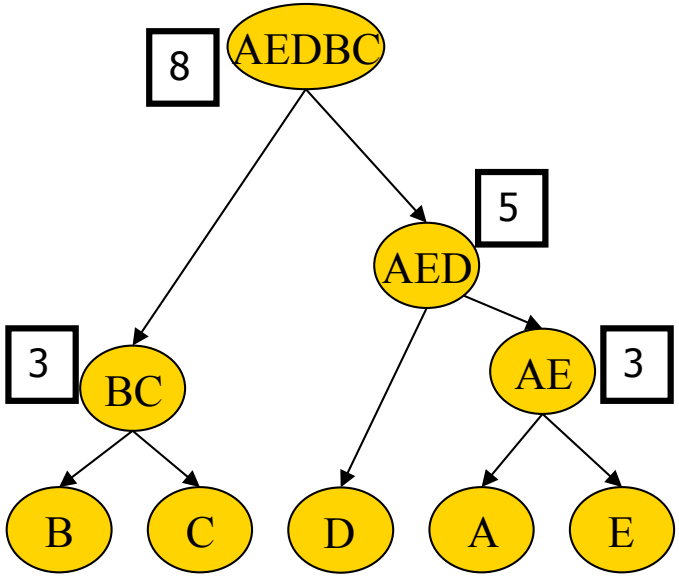


Beispiel

	AE	BC	D
AE		8	5
BC			8



	BC
AED	8

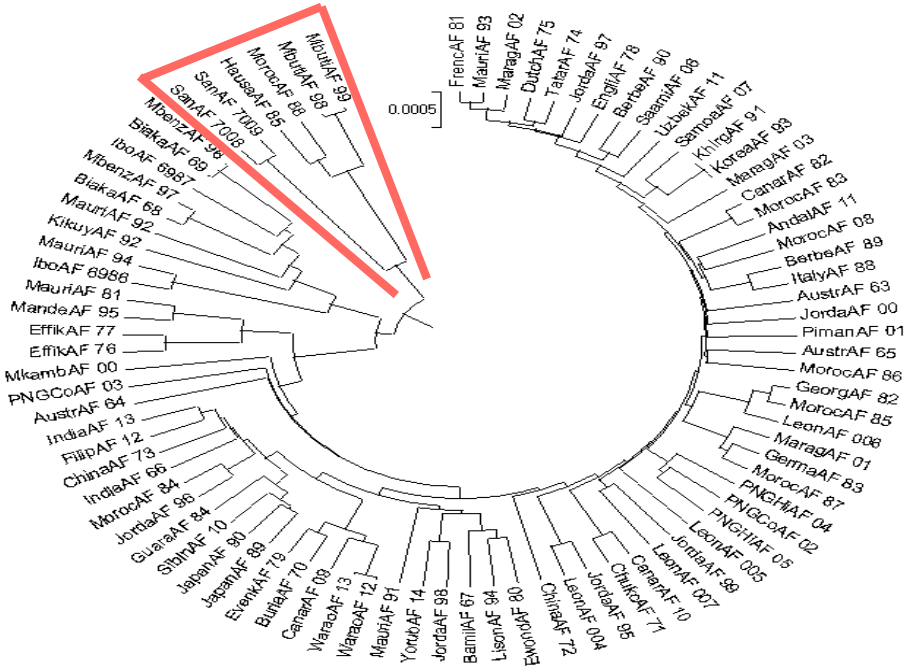


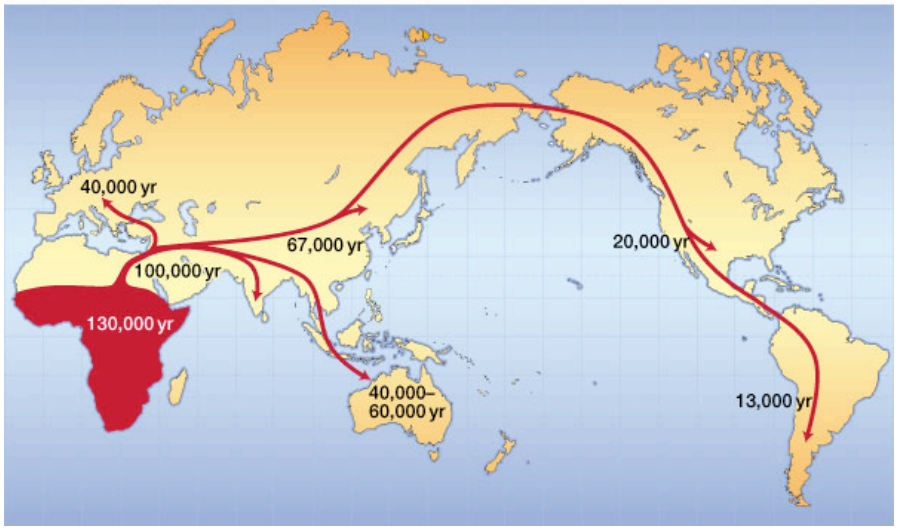
Kontrolle

	B	C	D	E
A	8	8	5	3
B		3	8	8
C			8	8
D				5

Anwendungsbeispiel

- Sequenzierung der mitochondrialer DNA (16 KB) von 86 geographisch verteilt lebenden Personen
- Ergebnis: Mitochondriale DNA scheint nach einer molekularen Uhr abzulaufen; Divergenz ist ca. 1,7E-8 pro Base und Jahr





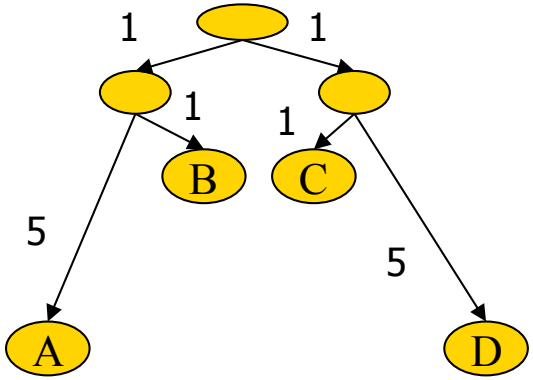
Quelle:
Ingman, M., Kaessmann, H., Pääbo, S. & Gyllensten, U. (2000)
Nature 408: 708-713

Quelle:
http://www.genpat.uu.se/mtDB/sequences.html
Methode: UPGMA

Wo UPGMA irrt

Der echte Baum

	B	C	D
A	6	8	12
B		4	8
C			6
D			

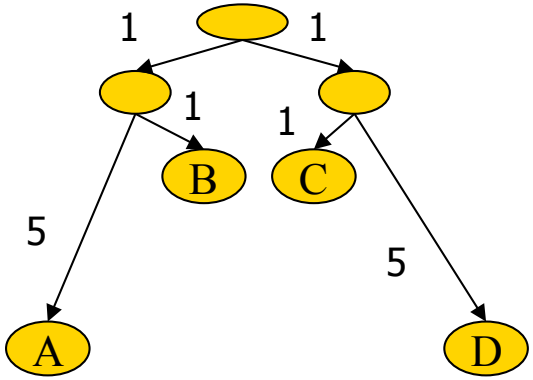


Was erzeugt UPGMA?

Wo UPGMA irrt

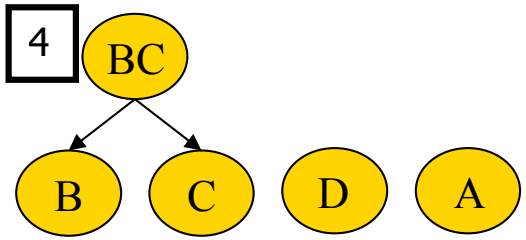
Der echte Baum

	B	C	D
A	6	8	12
B		4	8
C			6
D			



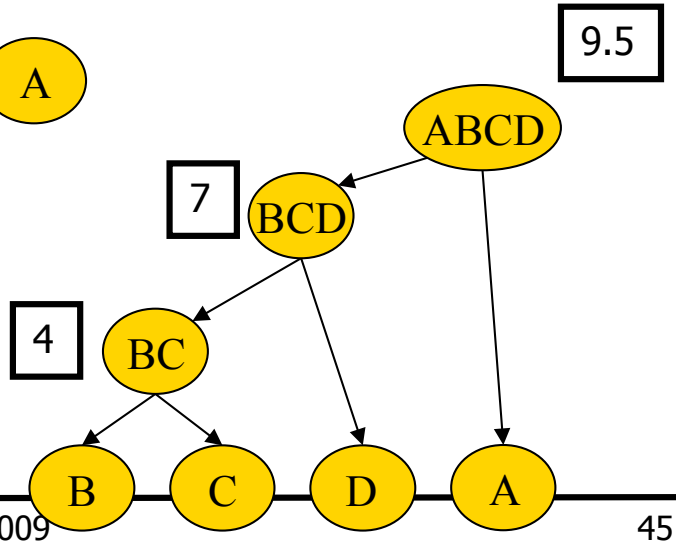
Was erzeugt UPGMA?

	B	C	D
A	6	8	12
B		4	8
C			6



	A	BC	D
A		7	12
BC			7

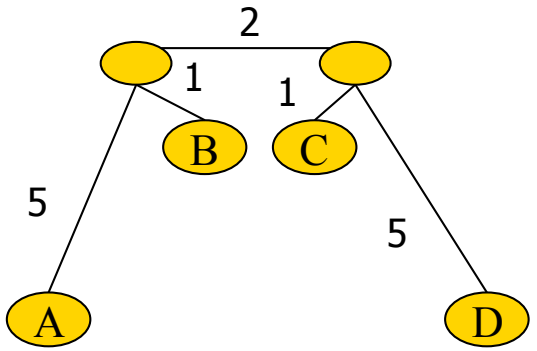
	A
BCD	9.5



Additive Bäume

- Ultrametriken berechnen im Kern die Label der inneren Knoten
  - Und dadurch kann man Kanten nur auf bestimmte Weisen aufteilen
- Besser: Additive Bäume
  - Bäume mit Labeln auf den Kanten
- Eingabe ist wieder eine Ähnlichkeitsmatrix
  - Die Werte in der Matrix sind nicht direkt die Kantenlabeln, sondern Längen von Pfaden im Baum
    - Abstände enthalten sich gegenseitig
- Da es nur Abstände gibt, hat der Baum keine Wurzel
  - Gesucht ist eine Anordnung so, dass die Pfade im Baum die „richtigen" Längen haben

	B	C	D
A	6	8	12
B		4	8
C			6
D			

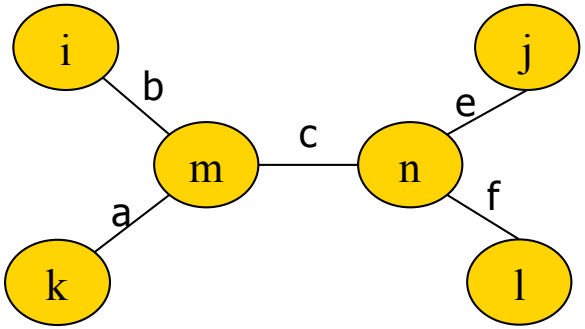


Additive Bäume

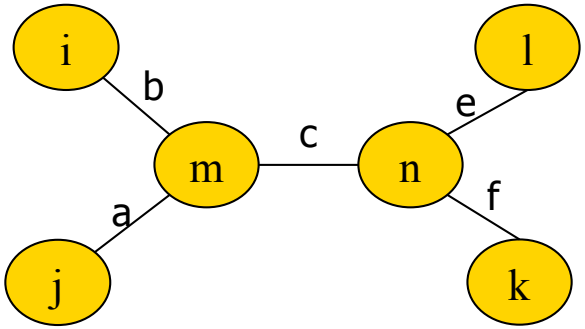
- Definition
  Sei D eine symmetrische Matrix mit n Spalten und Zeilen (und nur positiven Werten; D(i,i)=0). Ein Baum T mit Kantenlabeln heißt additiver Baum für D, wenn gilt
  - *T hat n Blätter, beschriftet mit den Zeilen von D*
  - *Für jedes Paar i,j ist D(i,j) gleich der Summe der Kantenlabel auf dem (eindeutigen) Pfad von i nach j*
- Bemerkung
  - Jede ultrametrische Matrix induziert einen additiven Baum aber nicht umgekehrt

4-Punkt Bedingung

- Theorem
  Eine Matrix D hat einen additiven Baum gdw. für alle Zeilen i, j, k, l die 4-Punkt Bedingung gilt:
  D(i,k)+D(j,l) ≤ max(D(i,j)+D(k,l) , D(i,l)+D(k,j))
- Beweis: Literatur



(a+b) + (e+f) ≤ max((b+c+e) + (a+c+f),
(b+c+f) + (a+c+e))



(b+c+f) + (a+c+e) ≤ max((a+b) + (e+f),
(b+c+e) + a+c+f))

Neighbor-Joining

- Matrizen und Algorithmen
  - Ultrametrische Matrizen – UPGMA
  - Additive Matrizen – Neighbor Joining
- Hierarchisches Clusterverfahren (wie UPGMA)
  - Erzeugt einen binären Baum ohne Wurzel
  - Grundaufbau wie UPGMA
    - Beginne mit so vielen Clustern wie Blättern
    - Wähle nach bestimmtem Kriterium zwei Cluster
    - Verschmelze die zwei Cluster und verbinde Knoten im Baum
    - Iteriere, bis nur noch ein Cluster vorhanden ist
- Unterschiede
  - UPGMA wählt Cluster zur Verschmelzung nur nach Nähe zueinander
  - Neighbor Joining wählt Cluster nach der Nähe zueinander und dem Abstand zu anderen Clustern

Weitere Algorithmen

- Maximum parsimony
  - Character-based
  - Jedes Zeichen einer Zeichenkette / jedes Merkmal wird verwendet, um innere Knoten zu beschriften
  - Small parsimony
    - Vorgegebener Baum
    - Berechnet die besten inneren Sequenzen
    - Fitch's Algorithmus
  - Large parsimony
    - Berechne Baum und innere Sequenzen
    - Branch & Bound

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
- Maximum Parsimony: Besser, aber sehr teuer
- Maximum Likelihood: Noch besser, noch viel teurer
- Also: Mehrere Methoden vergleichen
  - Gruppen, die überall gleich sind, gelten als sehr robust
  - „Consensus tree“