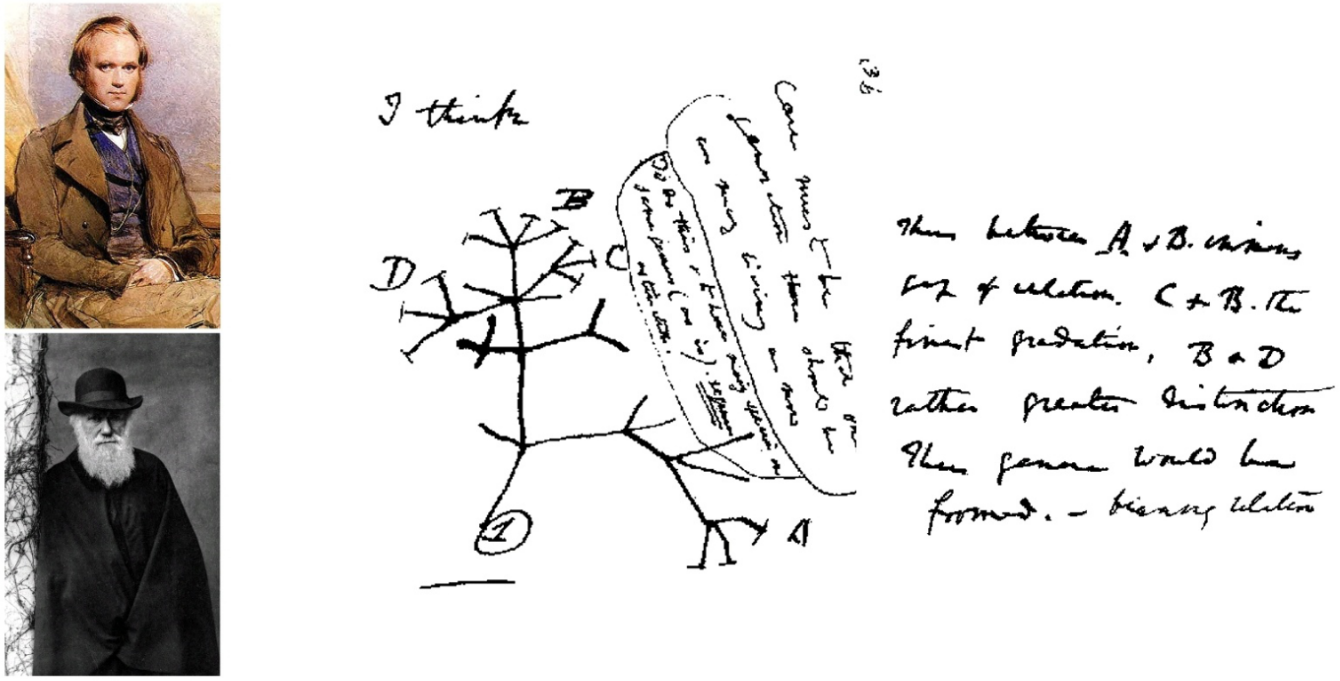
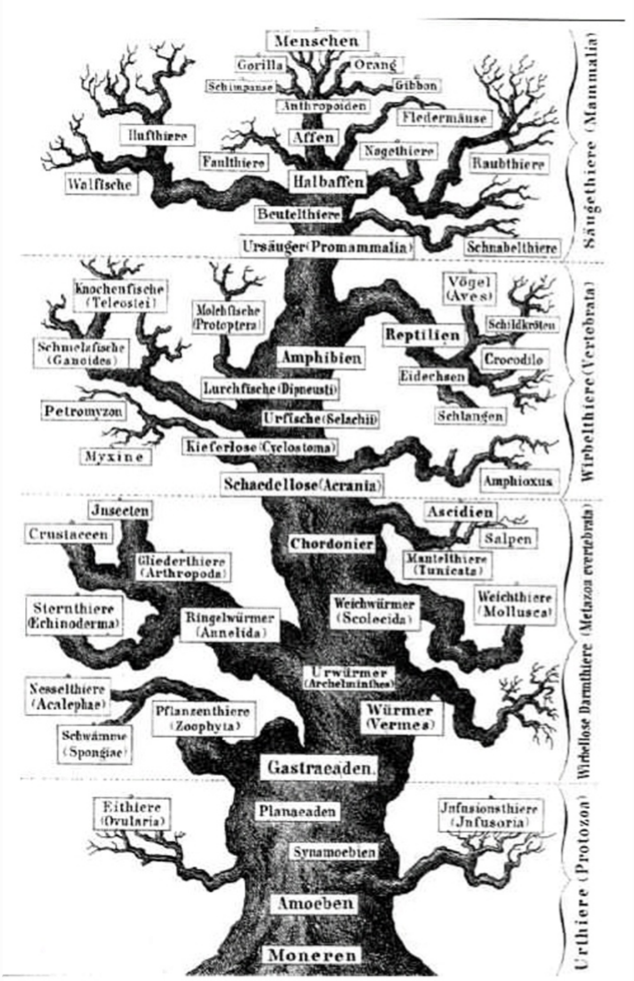
The Idea of Phylogenetic Trees

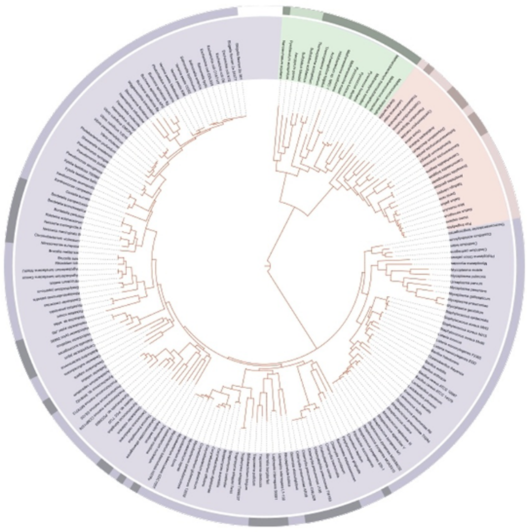

I think
1
A
B
C
D
Case must be that one generation then should be as many living as now. To do this & to have many species in same genus (as is) requires extinction.
Thus between A & B immens gap of relation. C & B the finest gradation, B & D rather greater distinction. Thus genera would be formed. — bearing relation

“I think” by Charles Darwin (1837) - One of the first evolutionary trees.



Ernst Haeckel, 1879

Relationship between species with sequenced genomes.



center = last universal ancestor of all life on earth.

three domains of life:

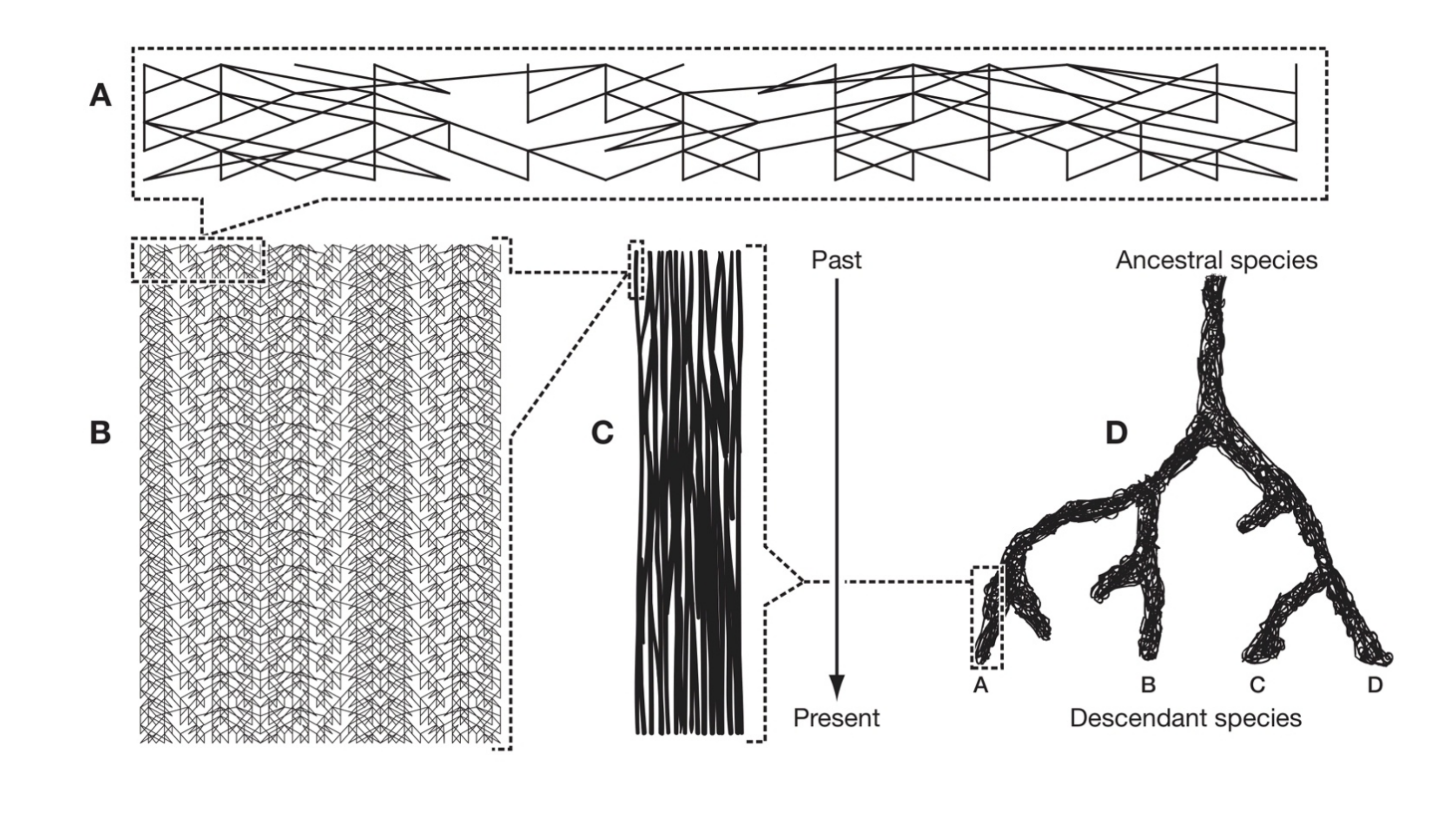
eukaryota (animals, plants and fungi);

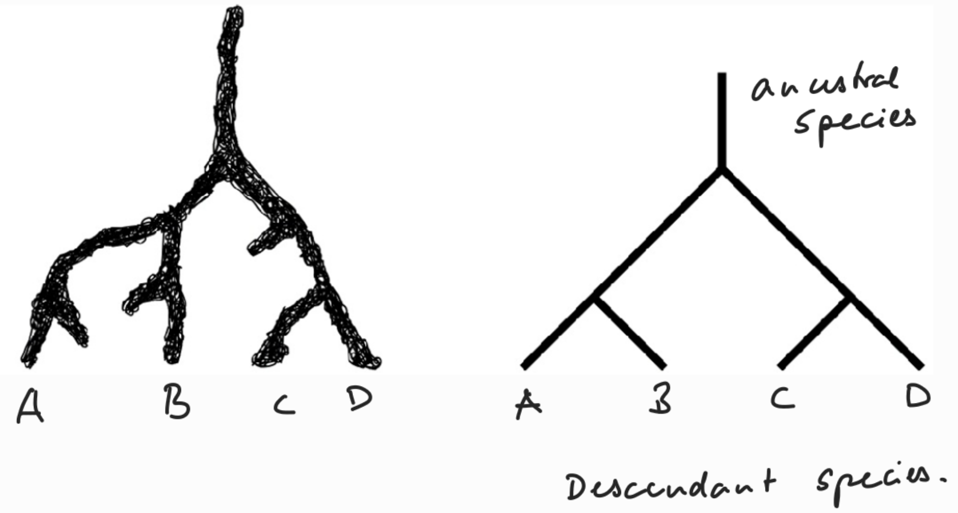
bacteria;

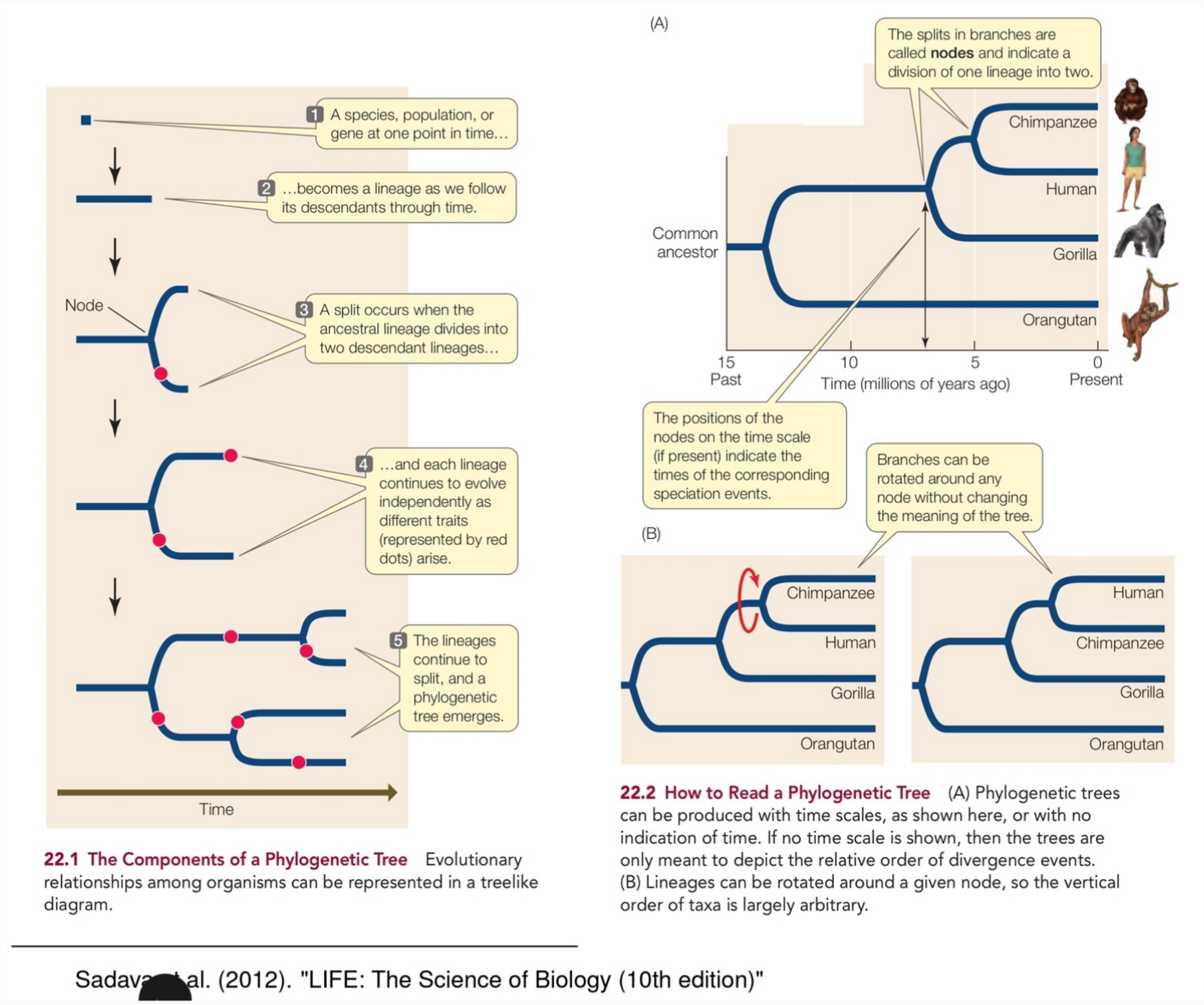
archaea.

Ciccarelli, FD (2006). "Toward automatic reconstruction of a highly resolved tree of life.". Science; Letunic, I (2007). "Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation.". Bioinformatics

A general view







1 A species, population, or gene at one point in time…

2 …becomes a lineage as we follow its descendants through time.

3 A split occurs when the ancestral lineage divides into two descendant lineages…

4 …and each lineage continues to evolve independently as different traits (represented by red dots) arise.

5 The lineages continue to split, and a phylogenetic tree emerges.

22.1 The Components of a Phylogenetic Tree Evolutionary relationships among organisms can be represented in a treelike diagram.

The splits in branches are called **nodes** and indicate a division of one lineage into two.

The positions of the nodes on the time scale (if present) indicate the times of the corresponding speciation events.

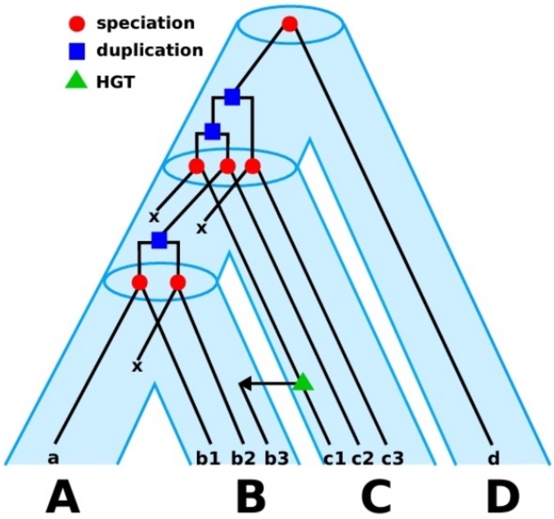
Branches can be rotated around any node without changing the meaning of the tree.

22.2 How to Read a Phylogenetic Tree (A) Phylogenetic trees can be produced with time scales, as shown here, or with no indication of time. If no time scale is shown, then the trees are only meant to depict the relative order of divergence events. (B) Lineages can be rotated around a given node, so the vertical order of taxa is largely arbitrary.

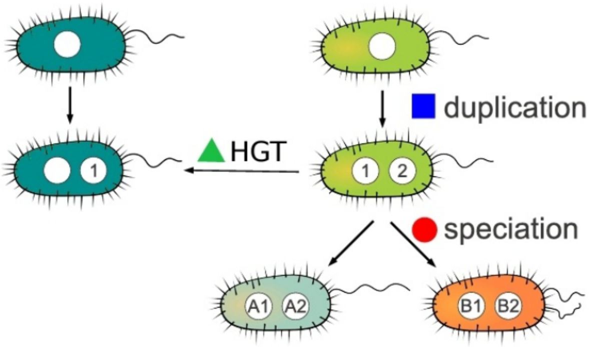
Sadav[illegible]al. (2012). "LIFE: The Science of Biology (10th edition)"

Applications: ↗ beamer-slides.

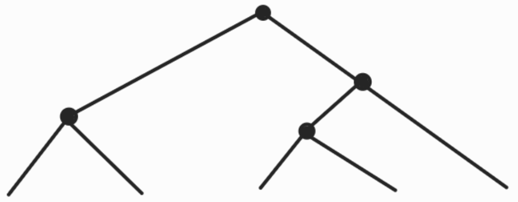
- species are characterized by its genome: a “bag of genes”
- “Genes” evolve along a *rooted* tree with unique coloring $t : V^0 \to M = \{●, ■, ▲\}$
- "×" = gene loss



- ■ Gene duplication : an offspring has two copies of a single gene of its ancestor
- ● Speciation : two offspring species inherit the entire genome of their common ancestor
- ▲ HGT : transfer of genes between organisms in a manner other than traditional reproduction and across different species

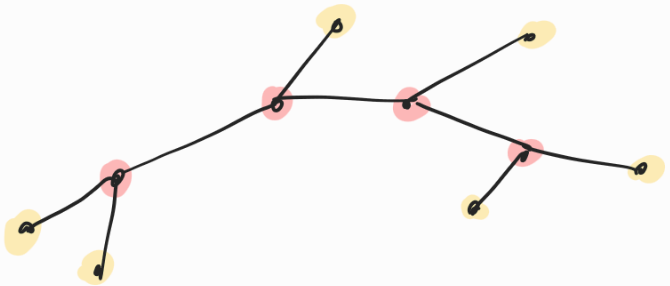


- All proposed phylogenetic trees are just HYPOTHESIS!
- only leaves of trees are known & this knowledge must be used to infer the underlying trees
- trees are not only about species evolution, but also of genes or other taxa as languages.



TAXA:	human	ape	whale	cow	horse
	ACGT	AGTT	TGCA	TCCA	AGAT
	IRISH	WELSH	SWEDISH	DANISH	GERMAN

$T = (V, E)$ is **tree** if connected & acyclic

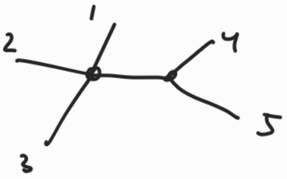


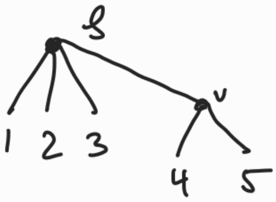
● **leaf**

● **internal vertices (inner)**

$L(T)$ = leafset of T

T **rooted** if one vertex $\rho \in V$ is called root

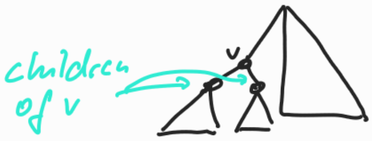
unrooted 

rooted 

IF not stated differently T is **phylogenetic**, ie.

unrooted T every inner vertex has at least degree 3

rooted T —— " —— —— " —— 2 children

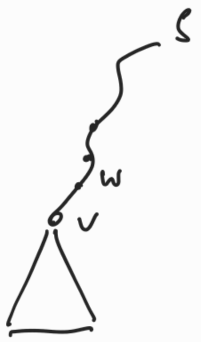


T fully resolved (= **binary**) if

$\forall\, v \in V \setminus L(T)$: degree v is 3 (T unrooted)

v has exactly 2 children (T rooted)

in rooted trees we have partial order $\le_T$ along vertices in T:



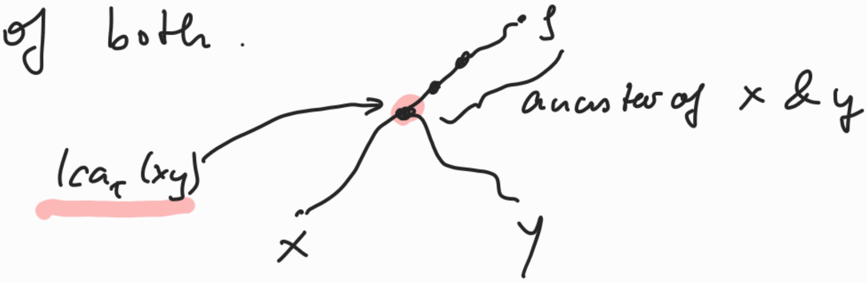
$v \le_T w$ if w lies on unique path from v to ρ

($v = w$ possible)

in this case,
v descendant of w
w ancestor of v

write $v <_T w$ if $v \le_T w$ & $v \neq w$.

last common ancestor $lca_T(xy)$ of $x, y \in V$ is $\le_T$-minimal vertex that is ancestor of both.



Depending on the application, phylogenetic trees may:

- be rooted or unrooted
- have weighted or unweighted edges / vertices
- labeled vertices / edges
- have bounded degree
  (maximum nr of children of each internal node)
- ...

- Inference of the gene or species tree T is a classical problem of molecular phylogenetics.

  In practice it can only be solved approximately.
- Only leaves of tree corresponding to extant (currently "observable") taxa is available.
- **Reconstructed trees do only provide a hypothesis about history!**

Lemma

There are $(2n-3)!!$ *rooted trees* $(2n-5)!!$ *unrooted trees with n leaves labeled from* $1, \dots, n$.

$(m)!! := \prod_{k=0}^{\lceil \frac{m}{2} \rceil -1}(m-2k) = m(m-2)(m-4)\cdots$.

	n	3	4	5	6	10	20
Exmpl:	unrooted	1	3	15	105	2'027'025	$2.22{\cdot}10^{20}$
	rooted	3	15	105	945	34'459'425	$8.20{\cdot}10^{21}$

Enumeration / exhaustive search is no option!

Aim: Assemble a tree representing a hypothesis about the evolutionary history of a set of genes, species or other taxa.

Methods:

- Distance Based e.g.:
  - Ultrametric Tree Reconstruction (UPGMA)
  - Additive Tree Reconstruction (Neighbor-Joining)
- Character Based e.g.:
  - Parsimony Methods (Fitch- and Sankoff Algorithm)
  - Maximum Likelihood (not part here)
- Consensus Methods e.g.:
  - Supertree from subtrees (BUILD)

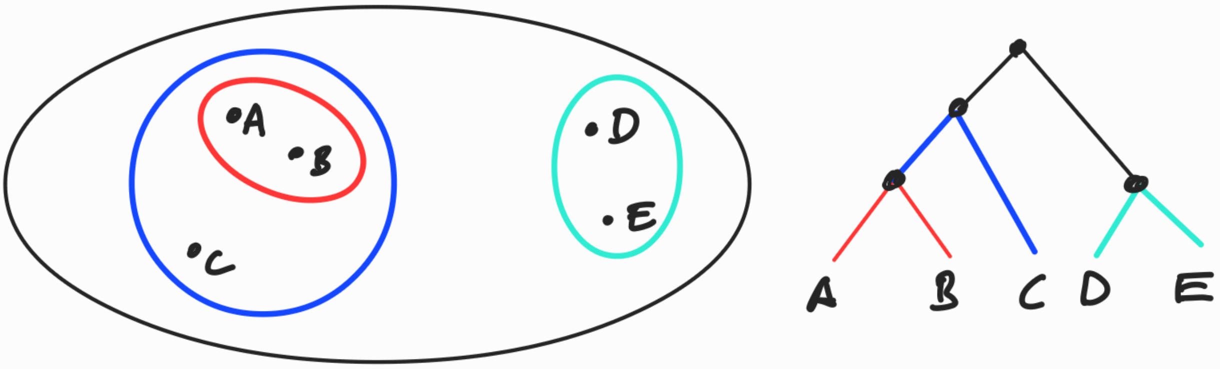
DISTANCE-BASED METHOD

UPGMA (unweighted pair group method with arithmetic mean)

(≡ bottom-up hierarchichal clustering method)

ALGO: IN: symmetric Distance matrix (or similarities) $D: X \times X \rightarrow \mathbb{R}$, $X = \{x_1 \cdots x_n\}$

now, in each step merge two "closest" cluster starting with $C_1=\{x_1\}, \ldots, C_n=\{x_n\}$ as singleton clusters.



After merging 2 clusters C_i & C_j into new cluster C_{new} distance as

$$D(C_{new}, C) = \frac{1}{|C_{new}||C|} \sum_{\substack{x \in C_{new} \\ y \in C}} D(x,y) \quad \forall C \neq C_{new}.$$

≙ mean distance between objects $x \in C_{new}$ & $y \in C$

REPEAT until one cluster remains

$C_1=\{a\},\ C_2=\{b\},\ C_3=\{c\},\ C_4=\{d\}$

D

	a	b	c	d
a	0	8	5	3
b		0	8	8
c			0	5
d				0

closest

⇒ merge C_1 & C_4 into $C_{new}=\{a,d\}$

⇒ new distances:

$$D(C_{new},C_2)=\frac{1}{2\cdot 1}\left(D(a,b)+D(db)\right)$$
$$=\frac{1}{2}(8+8)=8$$
$$D(C_{new},C_3)=\frac{1}{2}\left(D(a,c)+D(dc)\right)$$
$$=\frac{1}{2}(5+5)=5$$

↓ update

	C_{new}	C_2	C_3
	$\{a,d\}$	b	c
$\{a,d\}$	0	8	5
b		0	8
c			0

⇒ merge C_{new} & C_3 into $C_{new}'=\{a,c,d\}$

⇒ new distances

$$D(C_{new}',C_2)=\frac{1}{3\cdot 1}\left(D(a,b)+D(cb)+D(db)\right)$$
$$=\frac{1}{3}(8+8+8)=8$$

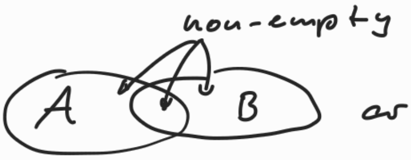
↓ update

	C_{new}'	C_2
	$\{\{a,d\},c\}\}$	b
$\{$	0	8
b		0

finally merge C_{new}' & C_2.

In this iterative process we obtained the set of clusters:

$$\mathcal{C}=\{\{a\},\{b\},\{c\},\{d\},\{ad\},\{acd\},\{abcd\}\}.$$

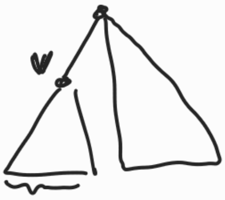
DEF 2 sets A, B s.t. 

are said to <u>overlap</u>.

Hence, A, B do <u>not overlap</u> if $A\cap B\in\{A,B,\emptyset\}$.

A set $\mathcal{C}$ of clusters is a <u>hierarchy</u> if no two elements of $\mathcal{C}$ overlap.

Given a rooted tree T: with leaf set X



let $\mathcal{L}(v) = \{x \in X : x \leq_T v\}$

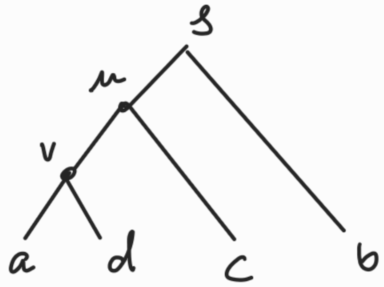
& put $\mathcal{C}(T) = \{\mathcal{L}(v) \mid v \in V(T)\}$

$\Rightarrow$ $\mathcal{C}(T)$ is a hierarchy [Exercise]

<u>Thm:</u> [without proof] Let $\mathcal{C}$ be a collection of non-empty subsets of X. Then, there is a phylogenetic rooted tree on X st. $\mathcal{C}(T) = \mathcal{C}$ $\Leftrightarrow$ $\mathcal{C}$ is hierarchy on X

By construction, UPGMA gives us a hierarchy $\mathcal{C}$ & thus a tree!

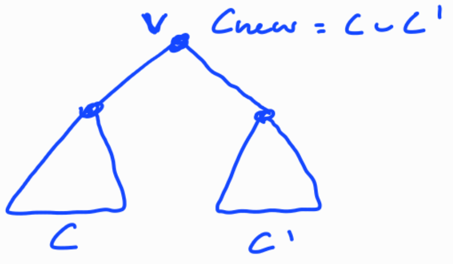
$\mathcal{C} = \{\{a\}, \{b\}, \{c\}, \{d\}, \{ad\}, \{acd\}, \{abcd\}\}.$

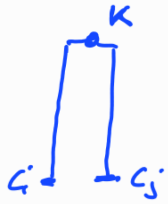


Can keep track of branch length δ

merging C & C' into C_{new}:

≙



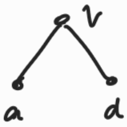


≙ divergence time of k unit on each branch.

$\Rightarrow D(C_i, C_j) = 2K$

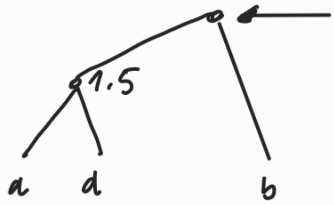
"timing" on $v = \dfrac{D(C,C')}{2}$

In Exampl: merged first {a} & {d}



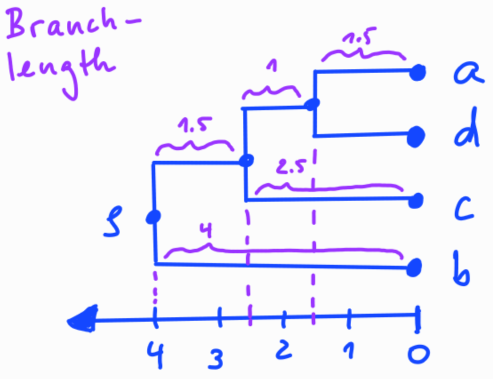
$\dfrac{D(a,d)}{2} = 1.5$

& updated distances & merged $C_{new} = \{ad\}$ & $\{b\}$ into C_{new}



$\dfrac{D(C_{new}, \{b\})}{2} = \dfrac{5}{2} = 2.5.$

& so on ....



Perfectly represented by tree

	a	b	c	d
a	0	8	5	3
b		0	8	8
c			0	5
d				0

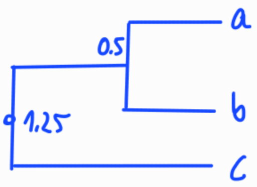
IF "branch length" not needed => we a have a heuristic to build tree.

Q: IF "branch length" needed .... does it always work?

A: NO!

$$\begin{array}{c|ccc} & a & b & c \\ \hline a & 0 & 1 & 2 \\ b & & 0 & 3 \\ c & & & 0 \end{array} = \begin{array}{c|cc} & \{a,b\} & c \\ \hline \{a,b\} & 0 & 2{,}5 \\ c & & 0 \end{array}$$

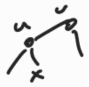
=> tree



$\delta(ac) = 2 \cdot 1{,}25 = 2{,}5 \neq D(a,c) = 2$

[in this case, at least a heuristic to get some tree]

DEF: rooted tree T with branch-length δ st all leaves have same distance to root ρ

&  $\Rightarrow \delta(u) \le \delta(v)$

is called **ultrametric tree**

DEF: Distance $D: X \times X \to \mathbb{R}_{\geq 0}$ is **ultrametric** if

1) $D(x,y) = 0 \iff x = y$
2) $D(xy) = D(yx)$
3) instead of usual Δ-inequ.:
$D(x,y) \le \max\{D(x,z), D(y,z)\} \quad \forall x,y,z \in X$

Let $D: X \times X \to \mathbb{R}_{\geq 0}$ be a map that satisfies D1/D2.

Lemma: [3 point condition] Then, D is ultrametric $\iff$ the two largest dist. among $D(x,y), D(x,z), D(yz)$ are equal $\forall x,y,z \in X$

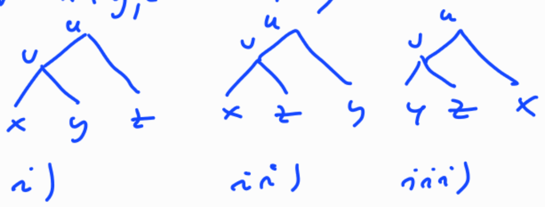
Proof: Exercise

Thm $\exists$ ultrametric tree T with branchlength δ that represents $D: X \times X \to \mathbb{R}_{\geq 0}$ $\iff$ D is ultrametric.

proof:

„$\Rightarrow$" (T, δ) ultrametric tree

$\Rightarrow$ $\forall\, x, y, z$ $T|_{x,y,z}$:



case i) $lca(xy) = v <_T lca(xz) = lca(yz) = u$

Since (T, δ) is ultrametric & it represents D we have: $D(xy) = \delta(v) \leq \delta(u) = D(xz) = D(y,z)$

$\Rightarrow$ 2 largest Dist are equal.

$\xRightarrow{\text{3 point cond}}$ D ultrametric. [analog case ii/iii]

„$\Leftarrow$"

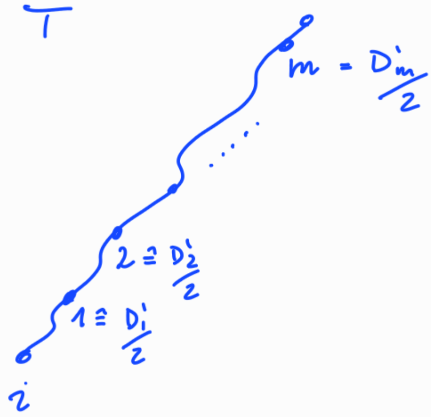
$X = \{1 .. n\}$

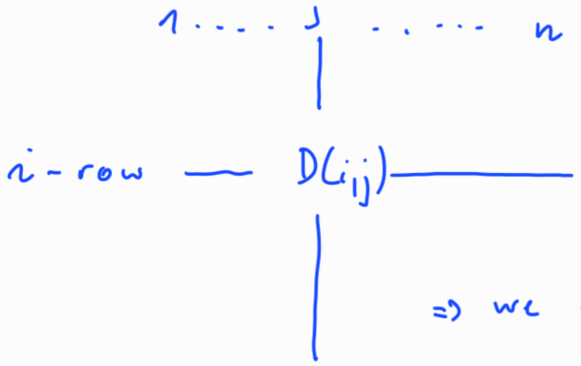
$D \begin{pmatrix} 1\,2\,2\,4\,1\,4\,4\,1 \end{pmatrix} i$ $D_1' = 1 < D_2' = 2 < D_3' = 4$ Exmpl.

Take i-th row of D ($\hat{=}$ i-th leaf in T)

Assume there are $m \geq 1$ dif. values $D_1' < D_2' < .. < D_m'$ in ith row of D

$\Rightarrow$ add path to T





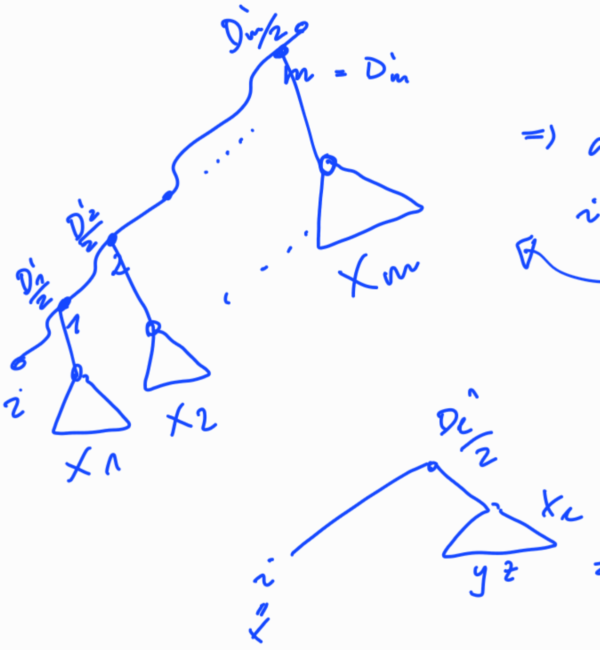
$D_{ij} \overset{?}{=} D'_\ell$

⇒ we can partition X into $X_1 \dots X_m$

with $X_\ell = \{ j \mid D(i,j) = \ell \}$

with $\ell \in \{ D'_1 \dots D'_m \}$

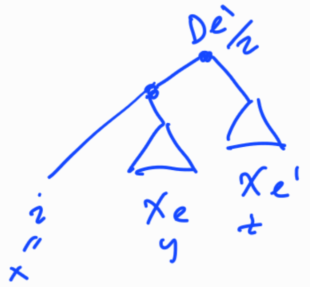
⇒



⇒ all $D(i,j)$, $1 \le j \le n$ are represented in T constructed so-far.

⇒ $D(xy) = D(xz)$ since $x, y \in X_\ell$

Rest for $D(yz)$ still to be computed. ⇒ correctly represented



⇒ $D'_\ell > D_\ell$

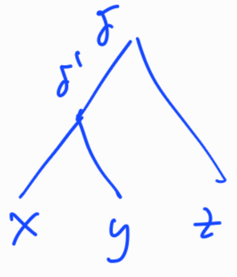
& $D(xy) < D(xz)$

$\overset{3p.cond}{\Longrightarrow}$ $D(yz) = D(xz) = \frac{D'_\ell}{2}$

now recurse on each class X_ℓ ⇒ correctly represented

by ind. on steps ⇒ we get tree „$T + \delta$" of D is represented by it.

we never obtain:



s.t. $\delta' > \delta$

⇒ get ultrametric tree T that represents D

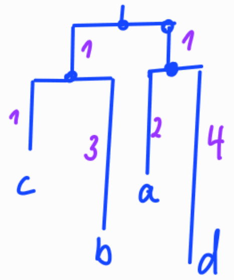
Drawbacks

constant-molecular-clock assumption:

"speed of evolution" ≙ mutation rate is constant, & thus, the same along all branches, i.e. path dist. from every leaf to root is the same.

Different rates (reflected as branch lengths):

possible true history:

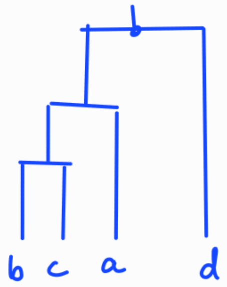


branchlength ≙ rates

D	a	b	c	d
a	0	7	5	6
b		0	4	9
c			0	7
d				0

UPGMA:

→



Neighbor - Joining (NJ)

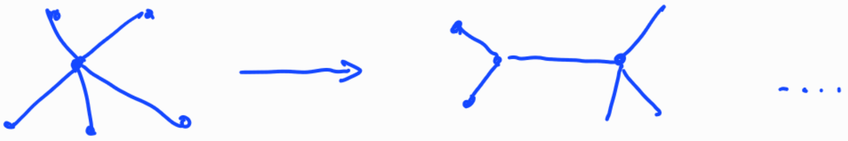
- no "const. mol. clock" assumption
- Based on concept of minimum evolution, i.e., resulting tree will have min total branch length.
- quite fundamental approach!

IDEA: start with "startree"

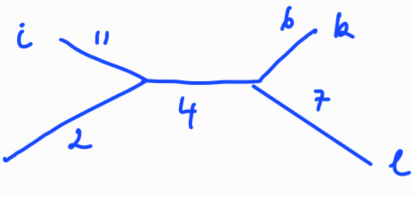
& stepwise separate vertices that are "quite" close to each other & "quite" far away from rest

until $\underbrace{\text{fully resolved}}$ unrooted tree has been built.

≙ binary, ie each inner vertex has degree 3



!



i, j neigbors but $D_{ij} = 13 > D_{jk} = 12$

but i, j are together "farer" away from rest.

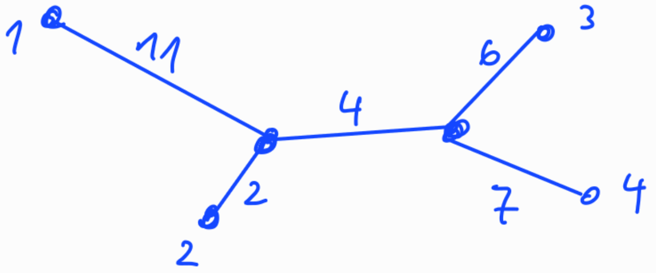
DEF: given n×n dist. matrix D. Then D^* denotes <u>neighbor-joining matrix</u> defined by

$$D^*_{i,j} = (n-2)\cdot D_{ij} - \text{TotalDist}_D(i) - \text{TotalDist}_D(j)$$

"degree of freedom" n taxa, 2 taxa in D_{ij}

where TotalDist(x) = sum of distances from x to all other

$$\begin{array}{c|cccc}
D & 1 & 2 & 3 & 4 \\
\hline
1 & 0 & 13 & 21 & 22 \\
2 & & 0 & 12 & 13 \\
3 & & & 0 & 13 \\
4 & & & & 0
\end{array}$$



D_{23} min but not neighbors in tree!

Sum branch-length between i,j = D(i,j)

TotalDist (1) = 56
(2) = 38
(3) = 46
(4) = 48

$$\begin{array}{c|cccc}
D^* & 1 & 2 & 3 & 4 \\
\hline
1 & 0 & -68 & -60 & -60 \\
2 & & 0 & -60 & -60 \\
3 & & & 0 & -68 \\
4 & & & & 0
\end{array}$$

$$D^*_{12} = (4-2)\cdot 13 - 56 - 38 = -68$$

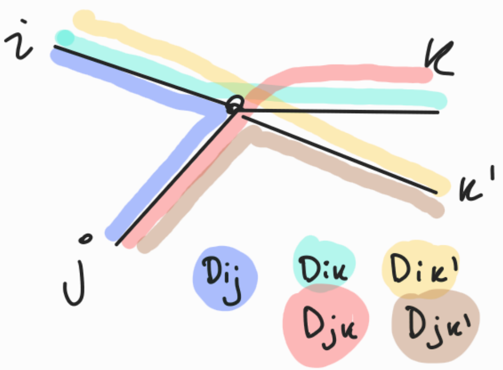
<u>"Intuition"</u> $D^*(i,j) \mathrel{\hat{=}}$ "common net divergence"

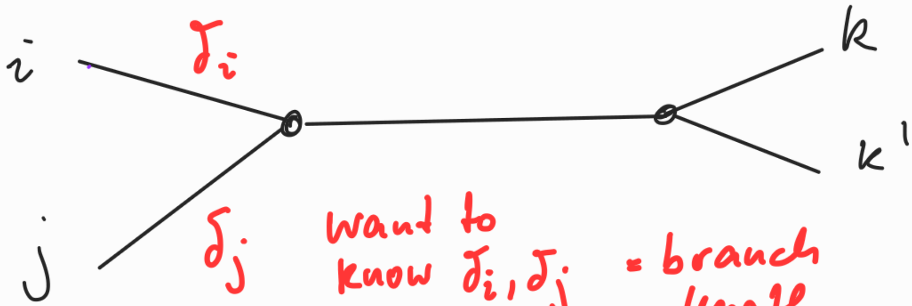
→ TAKE lowest one.

DEF: $\Delta_{ij} = (|\text{TotalDist}_D(i) - \text{TotalDist}_D(j)|) \cdot \frac{1}{n-2}$

Take i,j with min D^*_{ij} & $\boxed{\text{adjust } D}$ by "joining" i & j column/row to new m-th column/row → $D_{km} = D_{mk} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}$

"Intuition" (i,j) "joined as neighbors" in tree



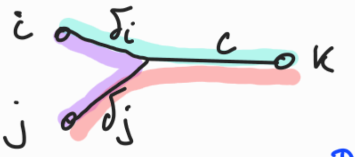


want to know δ_i, δ_j = branch length.

$$\Delta_{ij} = (D_{ij} + D_{ik} + D_{ik'} - (D_{ij} + D_{jk} + D_{jk'}))\ \frac{1}{4-2} \qquad // n=4$$

$$= (\ D_{ik} - D_{jk} + D_{ik'} - D_{jk'}) \cdot \frac{1}{2} = \delta_i - \delta_j$$

$$D_{ik} - D_{jk} = \delta_i + c - (\delta_j + c) = \delta_i - \delta_j$$



[Analog: $D_{ik'} - D_{jk'} = \delta_i - \delta_j$]

$D_{ij} = \delta_i + \delta_j + 2\delta_i - 2\delta_j$

$$\frac{1}{2}(D_{ij} + \Delta_{ij}) = \frac{1}{2}\left(\underbrace{(\delta_i + \delta_j)}_{= D_{ij}} + (\delta_i - \delta_j)\right) = \delta_i$$

$$\frac{1}{2}(D_{ij} - \Delta_{ij}) = \frac{1}{2}(\underbrace{\delta_i + \delta_j}_{= D_{ij}} - \delta_i + \delta_j) = \delta_j$$

ALGO:

[runtime $O(n^3)$]

```
NeighborJoining (D)
    IF D = 1x1 matrix stop
    ELSE
        1  construct $D^*$ from D
        2  Take i, j st $D^*_{ij} \overset{!}{\to}$ min
        3  Compute $\Delta_{i,j}$
        4  compute $\delta_i$ & $\delta_j$
        5  "Refine" tree   // starting from star tree
        6  D ← adjusted D [join i&j]
        7  NeighborJoining (D)
```

Exmpl:

first assume we dont know anything about tree => start with star tree

Neighb.-J.(D):

D	1	2	3	4
1	0	13	21	22
2		0	12	13
3			0	13
4				0

step 1: →

D^*	1	2	3	4
1	0	-68	-60	-60
2		0	-60	-60
3			0	-68
4				0

step 2 may choose (1,2) or (3,4)
decide here for (1,2)

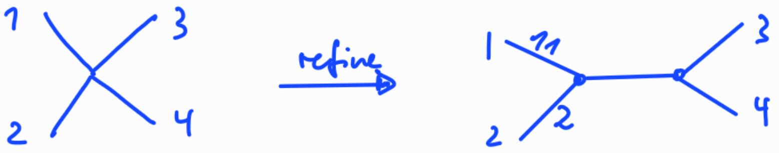
step 3

$$\Delta(i,j) = (\text{TotalDistD}(i) - \text{TotalDist}(j)) \cdot \frac{1}{2}$$
$$= (56 - 38) \cdot \frac{1}{2} = 9$$

step 4

$$\delta_i = \frac{1}{2}(D_{ij} + \Delta(i,j)) = \frac{1}{2}(13+9) = 11$$
$$\delta_j = \frac{1}{2}(D_{ij} - \Delta(i,j)) = \frac{1}{2}(13-9) = 2$$

step 5



step 6

D adjusted:

$$\boxed{D_{km} = D_{mk} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}}$$

D	(12)	3	4
(12)	0	10	11
3		0	13
4			0

$$\rightarrow D_{(12),3} = \frac{D_{13} + D_{23} - D_{12}}{2} = \frac{21 + 12 - 13}{2} = 10$$

step 7 recurse on D.

Def: Distance $D : X \times X \to \mathbb{R}_{\geq 0}$ is **additive** if

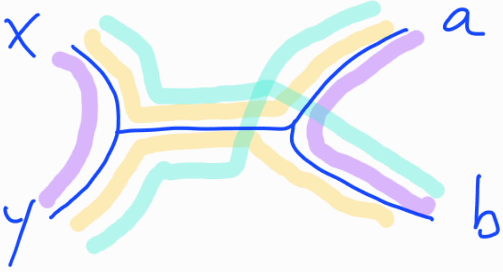
1) $D(x,y) = 0 \iff x = y$
2) $D(xy) = D(yx)$
3) instead of usual Δ-inequ.: $\forall\, x,y,a,b \in X$:

$$D(xy) + D(ab) \leq \max\{D(xa) + D(yb),\ D(xb) + D(ya)\}$$

[largest of must be equal]

$\equiv$ 4point condition

if T looks like:

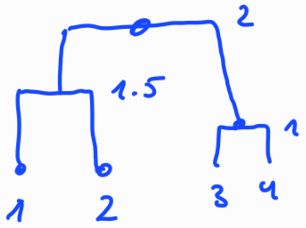


Ex: D metric $\overset{?}{\Rightarrow}$ additiv
$\Leftarrow$

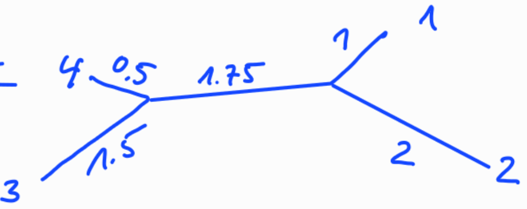
other example:

D	1	2	3	4
1	0	3	4	3
2	3	0	4	5
3	4	4	0	2
4	3	5	2	0

UPGMA



NJ



No ultrametric - why?

$|\{D(1,3), D(1,4), D(3,4)\}| = 3$

not additive:

$$D(1,2) + D(34) = 3+2 = 5$$
$$D(1,3) + D(24) = 4+5 = 9 \quad \Big\} \; 9 \not\leq \max(5,7).$$
$$D(1,4) + D(2,3) = 3+4 = 7$$

Thm [without proof] $\exists$ "additive" tree T with branchlength δ that represents $D: X \times X \to \mathbb{R}_{\geq 0}$

[that is, $D(i,j) = \sum$ weights δ along path connecting i,j in T]

$\Longleftrightarrow$ D additive metric

Drawback: trees may have negative branch-length.

Summary - Distance based methods

- Distance base method work well on ultrametric or additive distances.
- in any other case quite useful as heuristics.

- when taking sequ. alignments ACGT
  ACGC ...

  → get distances (but we loose inform in dist. matrix)

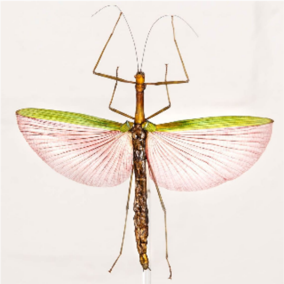
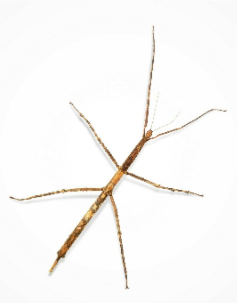
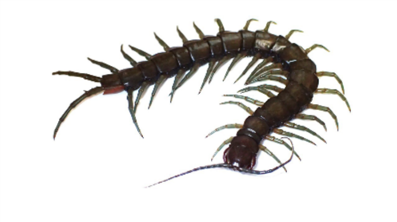
  ⇒ we cannot say anything about the ancestral state !

CHARACTER-BASED METHODS

Before "DNA-age", half a century ago, researchers constructed tree from morphological characters.

stick insects

giant centipide →

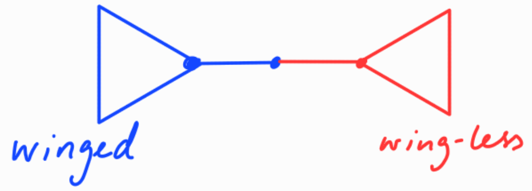
	wings	# legs
	yes	6
	NO	6
	NO	42

Aim: Reconstruct phylogeny from characters

Input: n × m matrix (n taxa, m characters)

output: tree in which taxa with similar character-values occur near each other.

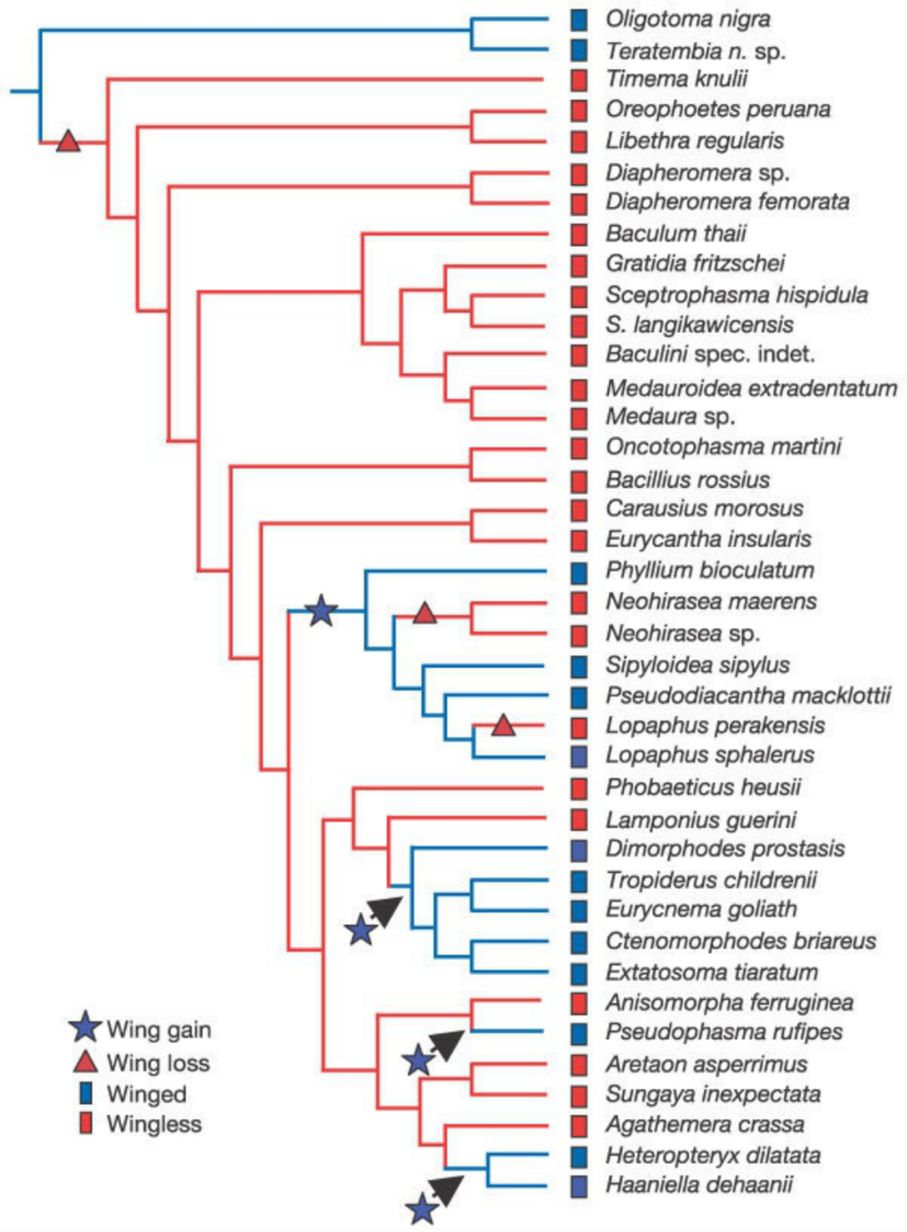
stick-insects



DOLLO's principle of irreversibility (1893):

Evolution doesn't reinvent the same organ (e.g. insect wings)
[evolution is efficient]

Stick-Insects Phylogeny:



☆/△ ≙ 7 times where wings were gained or lost in stick insects alone!

What happened?

Evolution did not reinvent wings from scratch

"genes switched on/off

⇒ wings yes/no"

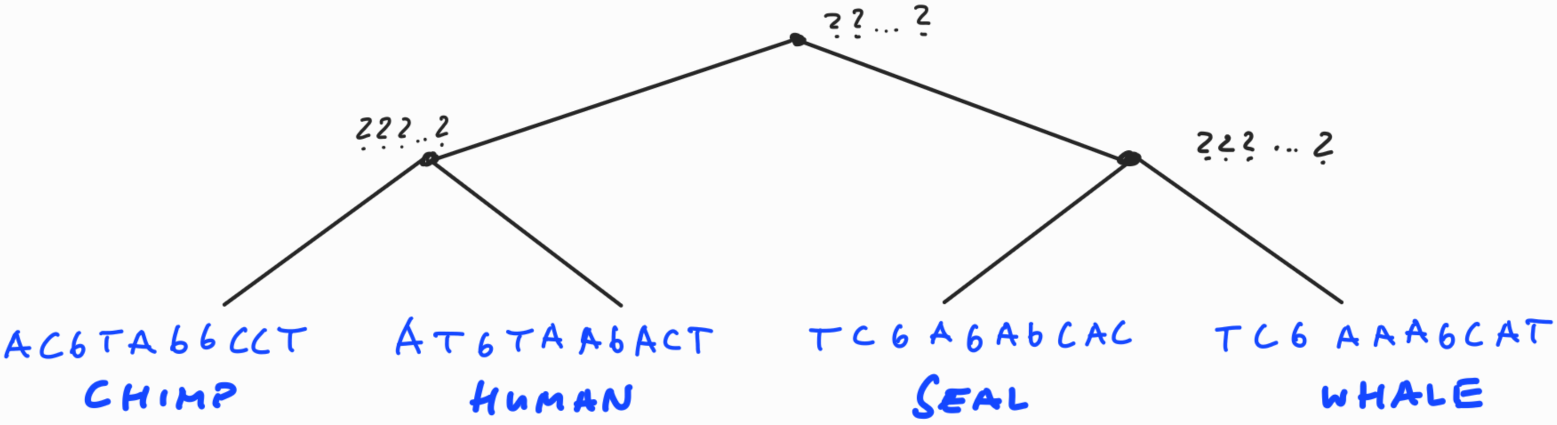
We can use genetic data as characters instead!

Species	Alignment
CHIMP	ACGTAGGCCT
HUMAN	ATGTAAGACT
SEAL	TCGAGAGCAC
WHALE	TCGAAAGCAT

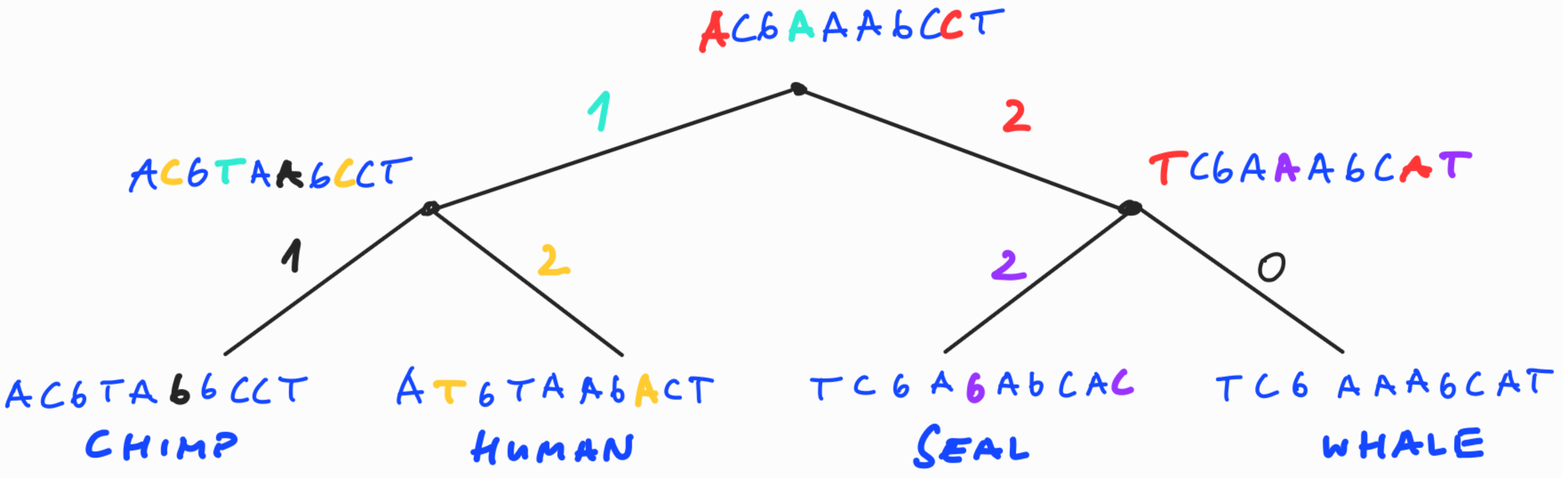
n taxa

m characters

Given tree & reconstruct most-likely ancestral sequences.



parsimony-score = sum of Hamming dist. along edges.



parsimony-score = 8

Now we have info about ancestral states!

Ockham's razor "simplest explanation is usually best one"
(1287-1347)

SMALL PARSIMONY problem:

Given rooted tree T, each leaf labelled by string of length m
Find labeling (= string of length m) for all internal nodes that minimize parsim. score.

IF "position" of string are independent THEN

Given rooted tree T, each leaf labelled by single symbol
Find labeling (= single symbol) for all internal nodes that minimize parsim. score.

Fitch-Algo (Walter M Fitch 1971)

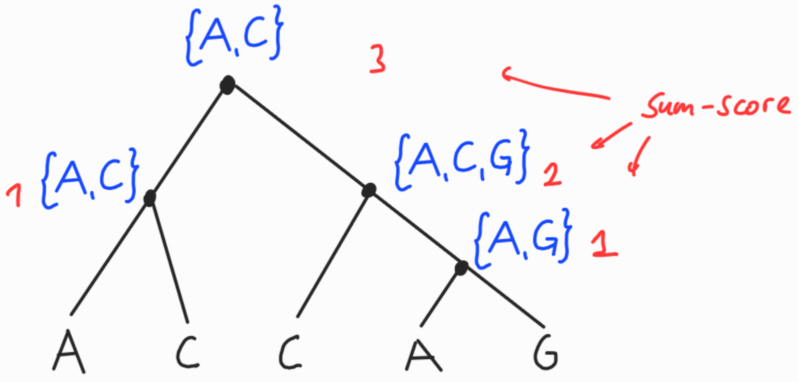
Given binary tree T with leaf labels

let X_v set of charactrset assigned to v

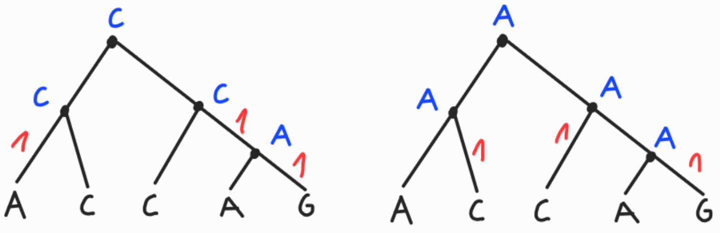
bottom up: assign to each internal vertex v with children u, w the state set

$$X_v := \begin{cases} X_u \cup X_w & , \text{if } X_u \cap X_w = \emptyset \\ X_u \cap X_w & , \text{else.} \end{cases}$$

until all vertices have been visited



possible solutions:



TOP DOWN:

root ρ take any character $l_\rho \in X_\rho$

Then for every internal node v

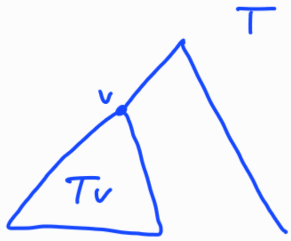
if $l_w \in X_v$ put $l_v = l_w$

else l_v any of X_v

SANKOFF-ALG (David Sankoff 1971)

Dynamic Prog!

T_v :



T_v = tree rooted at T.

$s_a(v)$ = min parsim. score of T_v over all labelings of T assuming v is labeled with symbol a

$$\boxed{\Rightarrow \text{ min pars. score of } T = s_a(\text{root}) \text{ over all symbols } a}$$

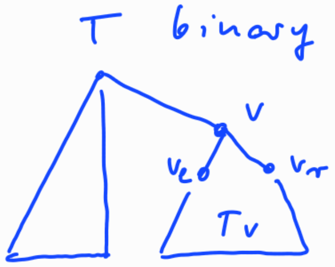
To implement: Def for symbols a,b :

$$\delta_{a,b} = \begin{cases} 0, & a = b \\ 1, & \text{else} \end{cases}$$

// also known as $\mathbb{1}_{ab}$ indicator fct

recurrence relation:

[can be generalized to non-binary]

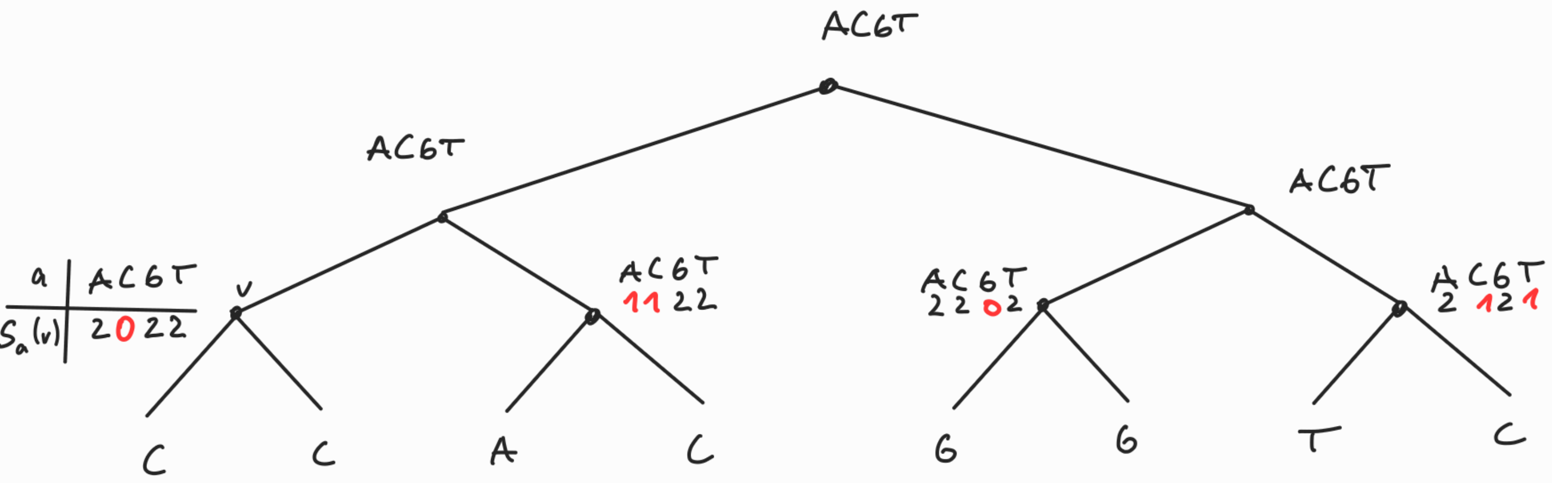


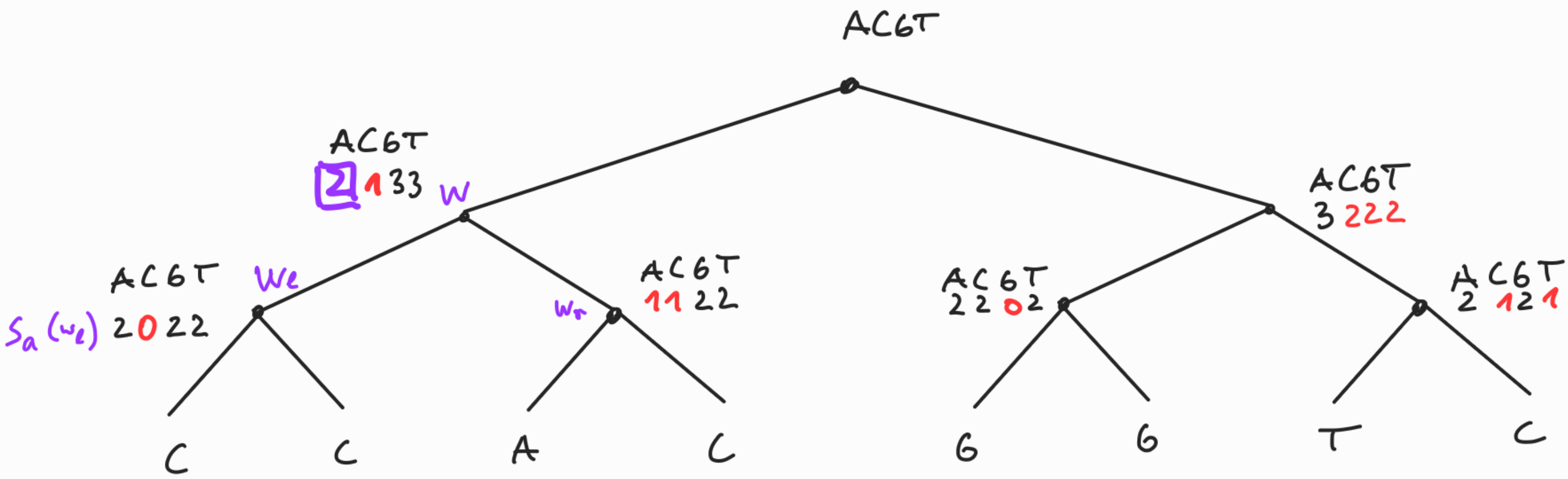
left/right child of v is v_ℓ / v_r

$$s_a(v) = \min_{\text{all symbols } y} \{ s_y(v_\ell) + \delta_{ay} \} + \min_{\text{all symbols } y} \{ s_y(v_r) + \delta_{ay} \}$$

where ∀ leaves ℓ: $s_a(\ell) = \begin{cases} 0, & \text{symbol of } \ell \text{ is } a \\ \infty, & \text{else} \end{cases}$

[detailed correctness as exercise]





<u>W = A</u>

$$S_A(w) = \min\{S_A(w_\ell)+\delta_{AA},\ S_C(w_\ell)+\delta_{AC},\ S_G(w_\ell)+\delta_{AG},\ S_T(w_\ell)+\delta_{TA}\}$$

$$+ \min(S_A(w_r)+\delta_{AA},\ S_C(w_r)+\delta_{AC},\ S_G(w_r)+\delta_{AG},\ S_T(w_r)+\delta_{TA}\}$$

$$= \min(2+0,\ \underline{0+1},\ 2+1,\ 2+1)$$

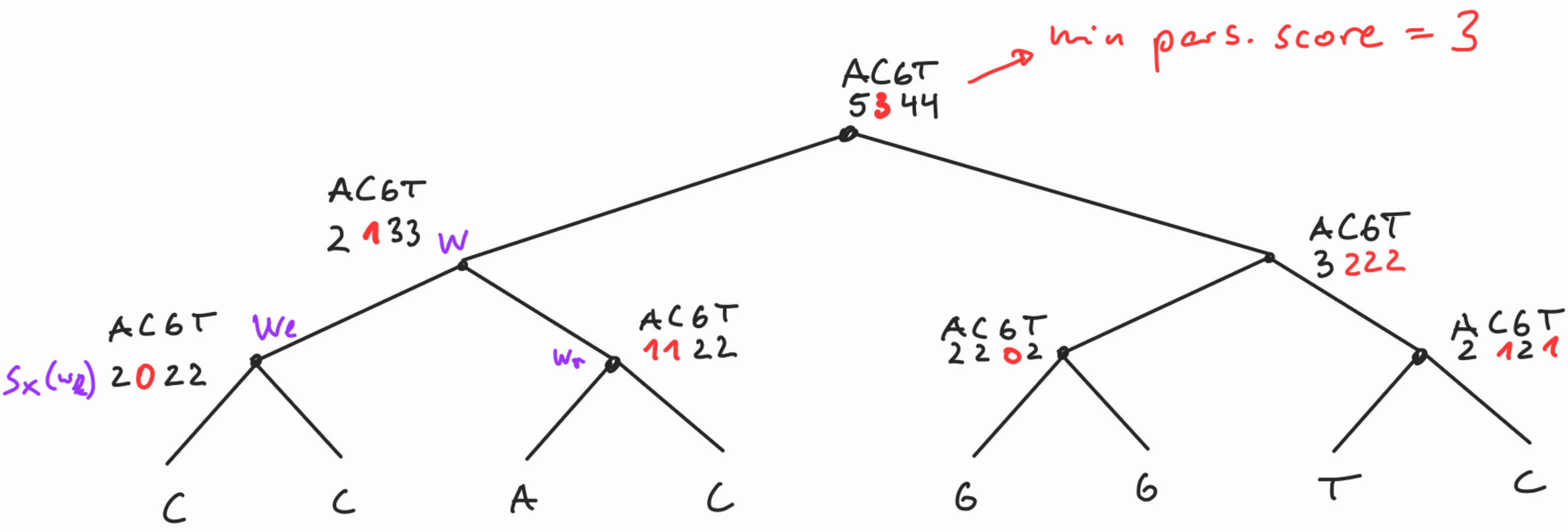
$$+ \min(\underline{1+0},\ 1+1,\ 2+1,\ 2+1)$$

$$= 1 + 1 = \boxed{2}$$

analog: $S_w(C) = 1$

$S_w(G) = 3$

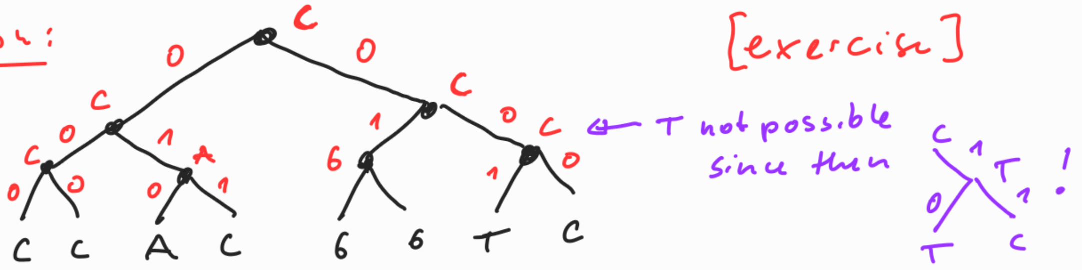
$S_w(T) = 3$



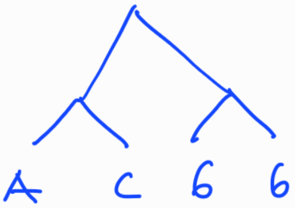
To reconstruct ancestral state: BACKTRACKING

possible solution:

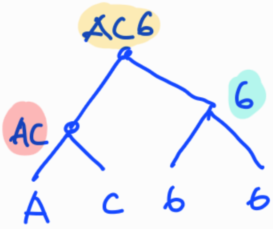
[exercise]



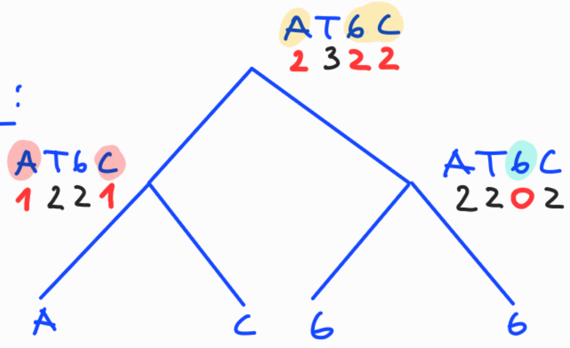
Fitch vs Sankoff Alg (both $O(nm)$ runtime)



Fitch:



Sankoff:



=> essentially "identical" in nature.

LARGE PARSIMONY problem:

Find rooted tree T, for given strings of length m with labeling (= string of length m) for all nodes that minimize parsim. score.

NP-hard!

=> heuristics needed!

[not part here]

CONSENSUS METHODS

IDEA Given a collection of trees $T_1 .. T_k$ Find common "supertree" that summarizes the information provided by $T_1 \dots T_k$ in a "best" way.

Why? ▶ Different datasets or tree-finding methods

⇒ Different trees.

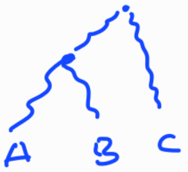
⇒ combine trees to get more reliable answer.

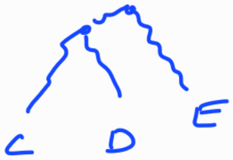
▶ comput. expensive methods can yield highly accurate trees on small (overlapping) data sets

⇒ Find 1 tree to represent entire data set.

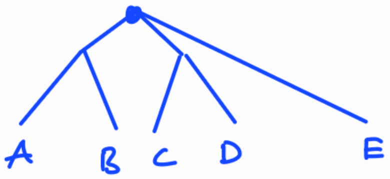
Exmpl: say we have only "partial" information about "similarities" between taxa A, B, C, D, E in the form: A & B are closer related than A compared to C & B compared to C

& C & D — " — than C — " — E
D — " — E

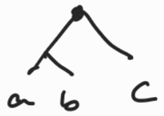
≘ 

& 

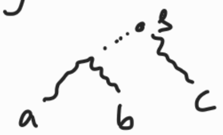
Q: Is there a common tree that reflects both relationships?

A: yes: 

DEF: <u>(ROOTED) TRIPLE</u> ab|c = binary rooted tree



ab|c <u>displayed</u> by rooted tree if

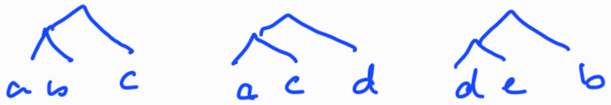


ab-path does not intersect with ρc-path

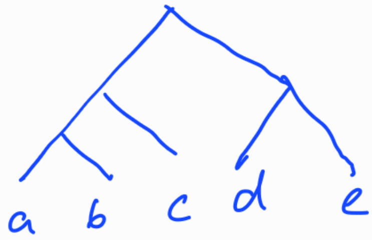
$(\Leftrightarrow lca(ab) <_T lca(ac) = lca(bc))$

Given set R of triples compatible if exists tree that <u>displays</u> all triples in R.

Exmpl: $ab|c$, $ac|d$, $de|b$



=> common supertree

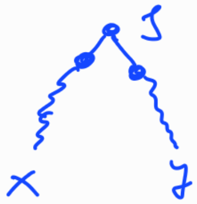


Exmpl: For $ab|c$ & $cb|a$ no tree!

OBSERVATION:

Given set R of triples & $xy|z \in R$

IF exists tree for R => x & y cannot be descendants of two different children of root.

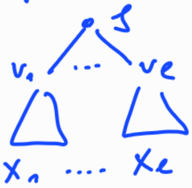


not possible!

=> Central idea: determine for potential tree the set of leaves that are descendents of each child of root.
[then recurse on children]

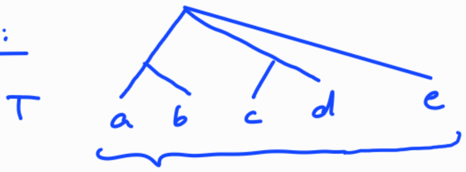
⇒ Find partition $X_1 \dots X_\ell$ of X

st



($|X_i| = 1$ identify v_i with $x \in X_i$)

Example:



$X = X_1 \dot\cup X_2 \dot\cup X_3$ with $X_1 = \{a,b\}$
$X_2 = \{c,d\}$
$X_3 = \{e\}$.

! $\forall$ triples $xy|z$ such a partition must satisfy:
xy are in same set X_i $(\nearrow \circledast)$

DEF Set of triples R, L set of leaves.

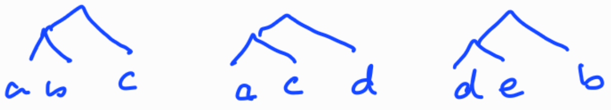
$$R_{|L} := \{ xy|z \in R : x,y,z \in L \}$$

DEF: Compatibility-graph $G[R,L]$

R = set of triples, L = set of leaves.

Then $G[R,L]$ has leaf set L &
$\{xy\}$ is an edge $\Leftrightarrow$ $\exists\ xy|z \in R_{|L}$

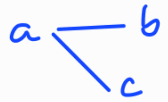
$R = \{ ab|c,\ ac|d,\ de|b \}$



$L = \{a, b, c\}$

$R = \{ab|c,\ ac|b\}$

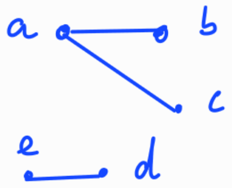
$G[R,L]$:



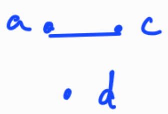
connected!

$G[R,L]$:

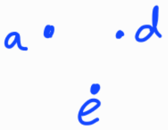
$L = \{a,b,\dots e\}$



$G[R,\{acd\}]$



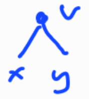
$G[R,\{a,d,e\}]$



disconnected!

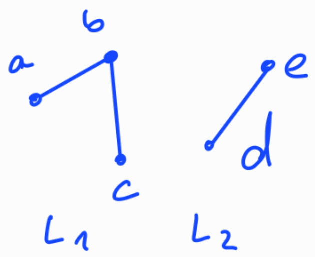
ALGO (from Aho, Sagiv, Szymanski & Ullmann 1981)

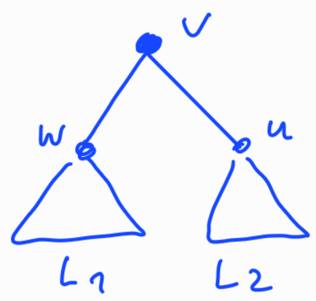
```
Build ( R, v, T, L )    // input: set of triples R,
                           vertex v & tree T
    IF (|L| = 1)
        output: rooted tree $\underset{x}{\bullet}$   ($x \in L$)
    IF ( |L| = 2 )
        output:  [tree]   ($x, y \in L$)
    IF ( |L| ≥ 3 )
        construct $G[R,L]$
        Let $L_1 \dots L_k$ vertex set of conn. components of $G[R,L]$
        IF ( k = 1 ) stop & output "R not compatible"
        FOR ( $i = 1 \dots k$ )
            call   BUILD( R, $v_i$, $T_i$, $L_i$ )
            IF ( BUILD( R, $v_i$, $T_i$, $L_i$ ) outputs tree $T_i$ )
                attach $T_i$ to v via edge $\{v_i, v\}$
```



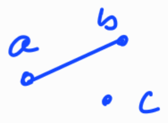
Exmpl: $R = \{ ab|c,\ ac|d,\ de|b \}$

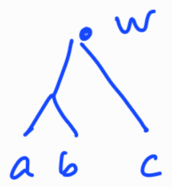
call of Build: $G[R, L]$, $L = \{a, b, \dots e\}$

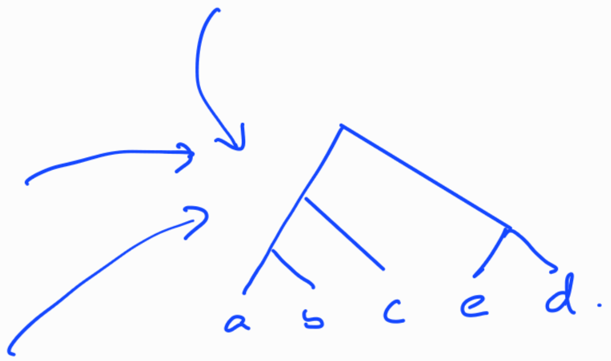




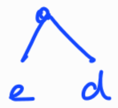
Build($R|_{L_1}$ ..)







Build($R_{|L_2}$, ..)



Theorem Build runs in $O(|L|\,|R|)$ time.

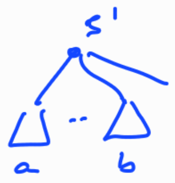
& is correct

proof [SKETCH]

R comp. $\Rightarrow$ R' comp. $\forall R' \subseteq R$.

$\Rightarrow \exists T'$ for R'

if $G[R', L']$ conn. $\Rightarrow \exists T'$



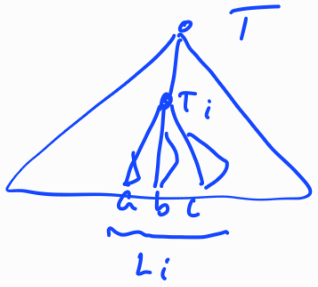
& $ab|c \in R'$

but $ab|c$ not displ. by T' ↯

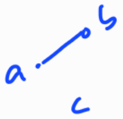
$\Rightarrow G[R', L']$ disconn. in each step.

remains to show T displays each triple in R.

let $ab|c \in R$ & T_i "min. subtree" in T that contain a, b, c

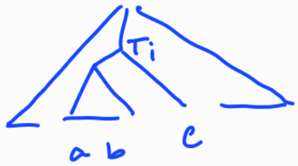


$\sigma(R, L_i) \Rightarrow$



$\Rightarrow$ ab in same comp.

$\longrightarrow$



[runtime: Exercise]

□