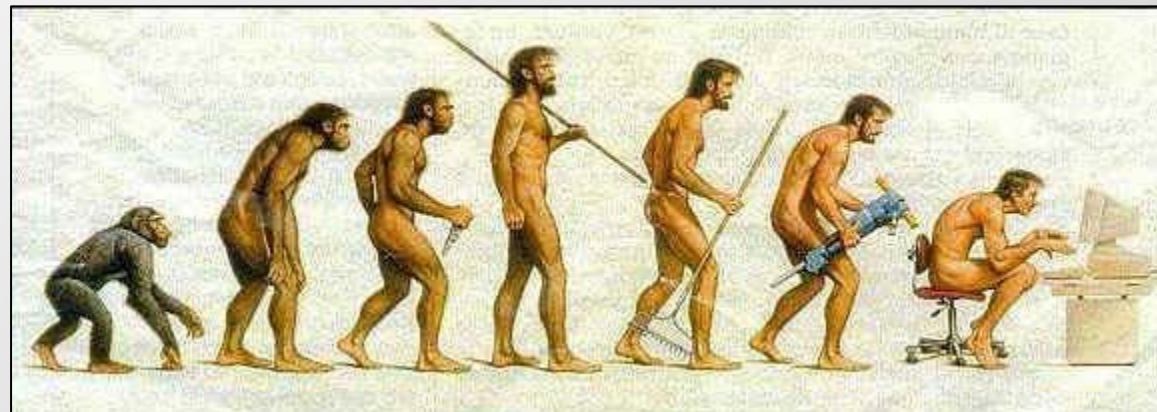
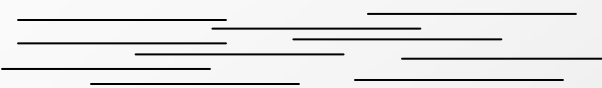


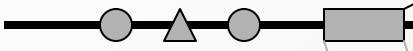
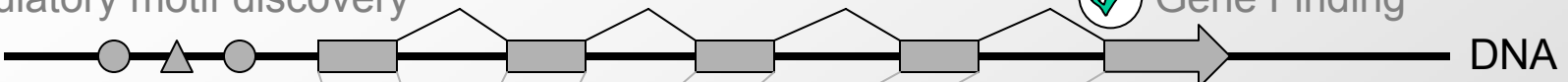
Phylogenetics and Multiple alignments

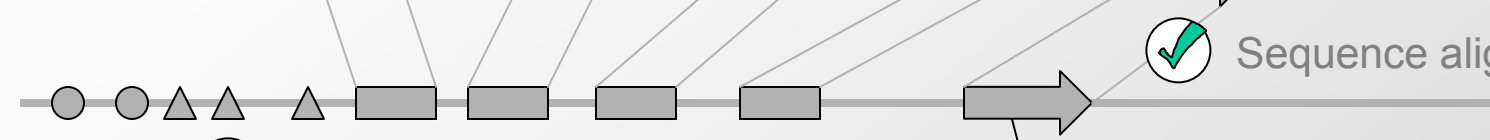


Somewhere, something went wrong...

Challenges in Computational Biology

④ Genome Assembly 

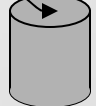
✓ Regulatory motif discovery  Gene Finding  DNA

✓ Sequence alignment 

✓ Comparative Genomics

⑦ Evolutionary Theory

```
TCATGCTAT
TCGTGATAA
TGAGGATAT
TTATCATAT
TTATGATTT
```

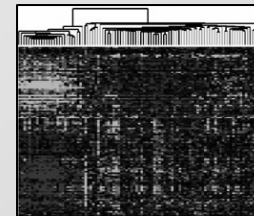
✓ Database lookup 

✓ RNA folding

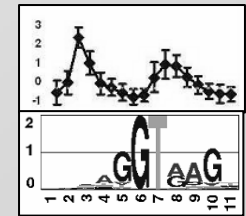


⑨ Gene expression analysis

RNA transcript 

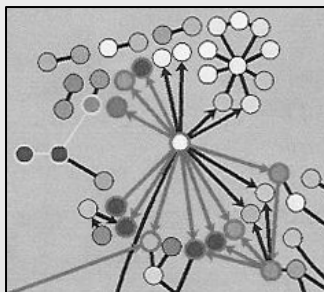


⑩ Cluster discovery



✓ Gibbs sampling

⑫ Protein network analysis



⑬ Regulatory network inference

⑭ Emerging network properties

Goals for today

- **Modeling sequence evolution**
 - Probabilistic modeling of divergence
 - Jukes-Cantor, Kimura 2-parameter model
 - Probabilistic interpretation of sequence alignment
- **Phylogenetics**
 - Tree building from distance matrices
 - UPGMA / Neighbor Joining / Maximum Likelihood
 - Tree building from sequence alignments
 - Parsimony methods, set-based vs. dynamic programming
- **Multiple sequence alignment**
 - Scoring schemes for multiple alignment
 - Sum of pairs, consensus score, parsimony
 - Algorithms for multiple alignment
 - Multi-dimensional DP
 - Progressive alignment
 - Iterative refinement

Open questions (?)

Image removed
due to
copyright restrictions.

- Panda
 - Bear or raccoon?
- Out of Africa
 - mitochondrial evolution story?
- Human evolution
 - Did we ever meet Neanderthal?
- Primate evolution
 - Are we chimp-like or gorilla-like?
- Vertebrate evolution
 - How did complex body plans arise?
- Recent evolution
 - What genes are under selection?

Image removed
due to
copyright restrictions.

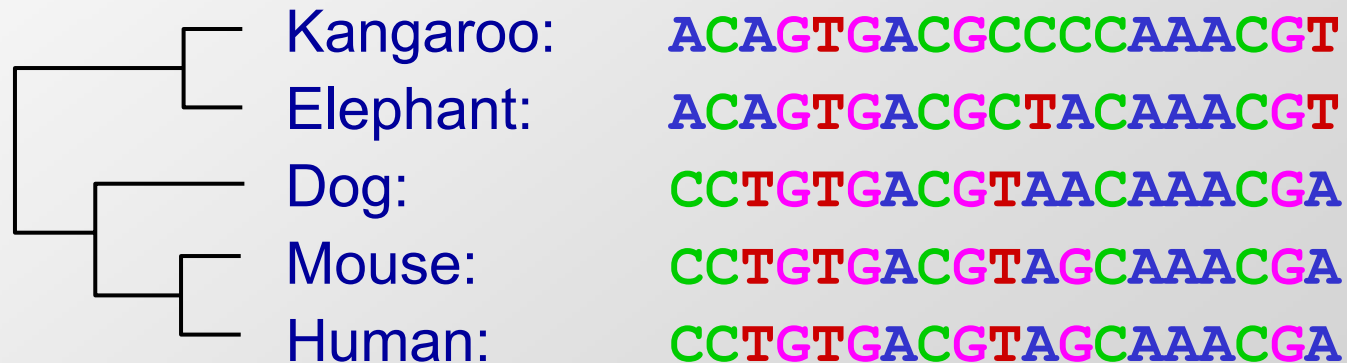
Image removed
due to
copyright restrictions.

Inferring Phylogenies

Trees can be inferred by several criteria:

- Morphology of the organisms
- Sequence comparison

Example:



Traits – as many as we have letters in DNA

YAL042W
candida586
cdub17784
cgla72177
cgui48535
clus15345
ctro67868
klac20931

-MKRSTLLSLDAFAKTEEDVRVRTRAGGLITLSCILITLFLLVNEWGQFNSVVTRPQLVV
MSSRPKLLSFDAFAKTVEDARIKTTSGGIITLICILITLVLIRNEYVDYTTIITRPELVV
MSSRPKLLSFDAFAKTVEDARIKTTSGGIITLICILITLVLIRNEYVDYTTIITRPELVV
-MKKSTLLSFDAFAKTEEDVRIRTRSGGFITLGCLVVTLMLLLSEWRDFNSVVTRPELVI
-MPQPKLLSFDAFAKTVEDARVRTPAGGIITLICVIVVLYLIRNEYLEYTSIINRPELVV
MSSRPRLLSLDAFAKTVEDARVKTASGGVITLVCVLIVLFLIRNEYSDYMLVVVRPELVV
MSSRPKLLSFDAFAKTVEDARIKTASGGIITLICVLITLILIRNEYIDYTTIITRPELVV
-MKKSPLLSIDAFGKTEEDVRVRTRTGGLITVSCIIITMLLLVSEWKQFSTIVTRPDLVV
:. **:*:*:* **.*:*: *:*:*: *:: .: *: .*: :: :: **:*:

YAL042W
candida586
cdub17784
cgla72177
cgui48535
clus15345
ctro67868
klac20931

DRDRHAKLELNMDVTFPSPMPCDLVNLDIMDDSGEMQLDILDAGFTMSRLNSEG-----R
DRDINKQLDINLDISFINLPCDLISIDLLDVTGDLNLNIDSGLKKIRLLKKNKQGDVIVN
DRDINKQLDINLDISFINLPCDLISIDLLDVTGDLNLNIDSGLKKIRLLKKNKQGDVIVN
DRDRSLRLDLNLDITFPSPMPCCELLTLDIMDDSGEVQLDIMNAGFEKTRLSKEG-----K
DRDINKKLEINLDISFPDIPCDVLTMDILDVSGDLQVDLLLSGFEEKFRLLKDG-----L
NRDVNRQLDINLDITFPDVP CGVMSLDILDMTGDLHLDIVESGFEMFRVPLG-----E
DRDINKQLDINLDISFINLPCDLISVDLLDVTGDDQLDIIDSGLKKVRLKKNKQGDVIIN
DRDRHLKLDLNLDTFPSPMPCNVNLNLDILDDSGEFQINLLDSGFTKIRISPEG-----K
:* * :*:*:*:*:* .:* * :*:*:*:* *:* :*:* :*:* :*:

YAL042W
candida586
cdub17784
cgla72177
cgui48535
clus15345
ctro67868
klac20931

PVGDATELHVGGNGDGTAPV--NND--PNY-CGPCYGAQDQSQN-ENLAQEEKVCQCQC
EIEDDEPAFNNDIELSDLAKGLPEGS DENAY-CGSCY GALPQDK-----KQFCCNDC
EIEDDEPAFNNDIELTDLAKGLPEGS DENAY-CGSCY GALPQDK-----KQFCCNDC
VLGTA-DMKIGEA AKKDKEA--QLAKLGANY-CGNCY GARDQ GKNNDDTPRDQWVCCQTC
EIRDESPVMSSAGELEERAR----GRAPDGL-CGSCY GALPQDEN-----LDYCCNDC
EISDDLPLLSGAKKFEDVCGPLTEDEISRGVPCGPCYGAVDQTD-----NKRCNDC
EIEDDKPALNSDVSLKELAKGLPEGS DQ NAY-CGPCY GALPQDK-----KQFCCNDC
ELSKE-KFQVGDKS--SKQS--FNE--EGY-CGPCY GALDQSKN-DELPQDQKVCCQTC
: . ** **** * . **:*:

YAL042W
candida586
cdub17784
cgla72177
cgui48535
clus15345
ctro67868
klac20931

DAVRSAYLEAGWAFFDGNIEQCEREGYVSKINEHLN--EGCRIKGS AQINRIQGNLHFA
NTVRRAYA EKHW SFYDGENIEQCEKEGYVGRRLRERINNNEGCRIGTKINRVSGTMDFA
NTVRRAYA EKHW SFYDGENIEQCEKEGYVARLRERINNNEGCRIGTKINRVSGTMDFA
DDVRQAYFEKNWAFFDGDIEQCEREGYVQKIADQLQ--EGCRVSGSAQLNRIDGNLHFA
ETVRLAYA QKAWGFFDGENIEQCEREGYVARLNEKINNFEGRIGTKINRISGNLHFA
EAVRMAYAVQEWGFFDGSNIEQCEREGYVEKMSRINNNEGCRIGKSAKINRISGNLHFA
NTVRRAYA EKQWQFFDGENIEQCEKEGYVKRLRERINNNEGCRIGSTKINRVSGTMDFA
DDVRAAYGQKGAFFDKGKVEQCEREGYVESINARIH--EGCRVQGRAQLNRIOGTIHF
: ** * * * * .:*:*:*:*:* : *:* :*:*:*:* .:*:*:*:*:* .

From physiological traits to DNA characters

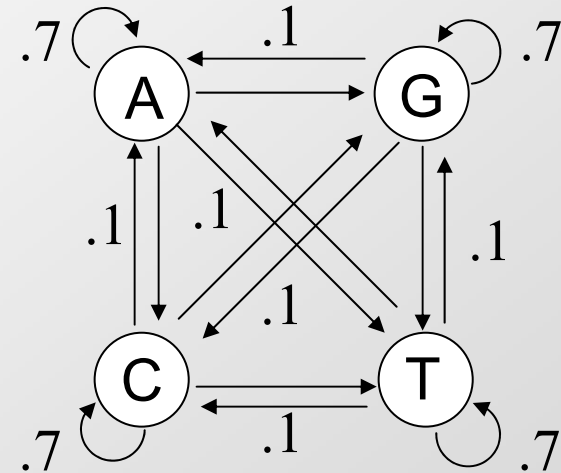
- **Traditional phylogenetics**
 - Building species trees
 - Small number of traits
 - Hoofs, nails, teeth, horns
 - Well-behaved traits, each arose once
 - Parsimony principle, Occam's razor
- **Modern phylogenomics**
 - Building gene trees and species trees
 - Very large number of traits
 - Every DNA base and every protein residue
 - Frequently ill-behaved traits
 - Back-mutations are frequent (convergent evolution)
 - Small number of letters, arise many times independently

Modeling evolution

Inferring evolutionary distance

'Evolving' a nucleotide under random model

- At time step 0, start with letter A
- At time step 1:
 - Remain A with probability 0.7
 - Change to C,G,T with prob. 0.1 each
- At time step 2:
 - In state A with probability 0.52
 - Remain A with probability $0.7 * 0.7$
 - Go back to A from C,G,T with $0.1*0.1$ each
 - In states C,G,T with prob. 0.16 each



| | t=1 | t=2 | t=3 | t=4 | t=5 |
|---|-----|-----|------|-------|--------|
| A | 1 | 0.7 | 0.52 | 0.412 | 0.3472 |
| C | 0 | 0.1 | 0.16 | 0.196 | 0.2176 |
| G | 0 | 0.1 | 0.16 | 0.196 | 0.2176 |
| T | 0 | 0.1 | 0.16 | 0.196 | 0.2176 |

Modeling Nucleotide Evolution

During infinitesimal time Δt , there is not enough time for two substitutions to happen on the same nucleotide

So we can estimate $P(x | y, \Delta t)$, for $x, y \in \{A, C, G, T\}$

Then let

$$S(\Delta t) = \begin{pmatrix} P(A|A, \Delta t) & \dots & P(A|T, \Delta t) \\ \dots & & \dots \\ P(T|A, \Delta t) & \dots & P(T|T, \Delta t) \end{pmatrix}$$

Modeling Nucleotide Evolution

Reasonable assumption: multiplicative
(implying a stationary Markov process)

$$S(t+t') = S(t)S(t')$$

That is, $P(x | y, t+t') = \sum_z P(x | z, t) P(z | y, t')$

Jukes-Cantor: constant rate of evolution

$$\text{For short time } \varepsilon, S(\varepsilon) = \begin{pmatrix} 1 - 3\alpha\varepsilon & \alpha\varepsilon & \alpha\varepsilon & \alpha\varepsilon \\ \alpha\varepsilon & 1 - 3\alpha\varepsilon & \alpha\varepsilon & \alpha\varepsilon \\ \alpha\varepsilon & \alpha\varepsilon & 1 - 3\alpha\varepsilon & \alpha\varepsilon \\ \alpha\varepsilon & \alpha\varepsilon & \alpha\varepsilon & 1 - 3\alpha\varepsilon \end{pmatrix}$$

Modeling Nucleotide Evolution

Jukes-Cantor:

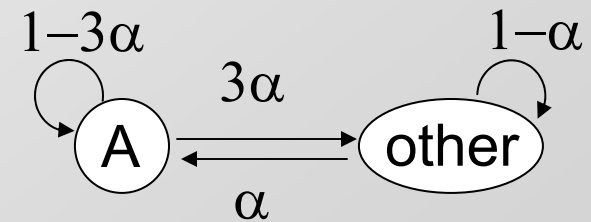
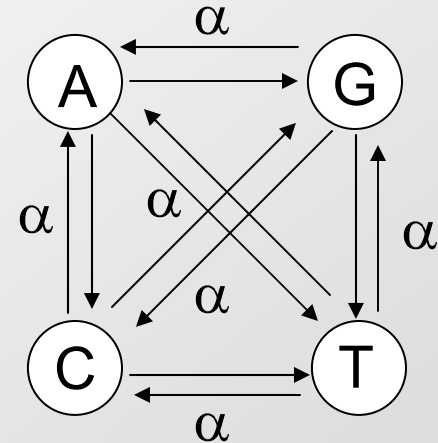
For longer times,

$$S(t) = \begin{pmatrix} r(t) & s(t) & s(t) & s(t) \\ s(t) & r(t) & s(t) & s(t) \\ s(t) & s(t) & r(t) & s(t) \\ s(t) & s(t) & s(t) & r(t) \end{pmatrix}$$

Where we can derive:

$$r(t) = \frac{1}{4} (1 + 3 e^{-4\alpha t})$$

$$s(t) = \frac{1}{4} (1 - e^{-4\alpha t})$$



Modeling Nucleotide Evolution

Kimura:

Transitions: A/G, C/T

Transversions: A/T, A/C, G/T, C/G

Transitions (rate α) are much more likely than transversions (rate β)

$$S(t) = \begin{array}{c} \text{A} \\ \text{G} \\ \text{C} \\ \text{T} \end{array} \begin{pmatrix} \text{A} & \text{G} & \text{C} & \text{T} \\ r(t) & s(t) & u(t) & u(t) \\ s(t) & r(t) & u(t) & u(t) \\ u(t) & u(t) & r(t) & s(t) \\ u(t) & u(t) & s(t) & r(t) \end{pmatrix}$$

Where

$$s(t) = \frac{1}{4} (1 - e^{-4\beta t})$$

$$u(t) = \frac{1}{4} (1 + e^{-4\beta t} - e^{-2(\alpha+\beta)t})$$

$$r(t) = 1 - 2s(t) - u(t)$$

Distance between two sequences

Given (well-aligned portion of) sequences x^i, x^j ,

Define

d_{ij} = distance between the two sequences

One possible definition:

d_{ij} = fraction f of sites u where $x^i[u] \neq x^j[u]$

Better model (Jukes-Cantor):

$$d_{ij} = -\frac{3}{4} \log(1 - 4f / 3)$$

$$r(t) = \frac{1}{4} (1 + 3 e^{-4\alpha t})$$

$$s(t) = \frac{1}{4} (1 - e^{-4\alpha t})$$

Observed $F = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]$

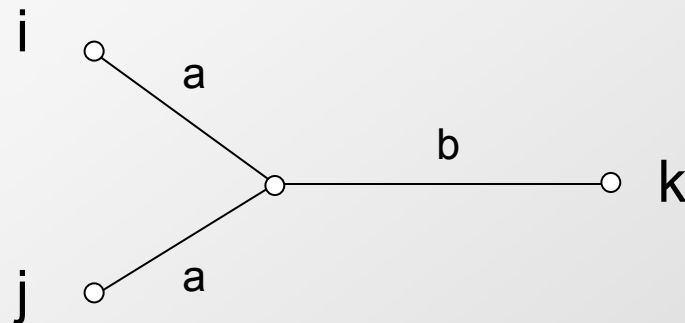
Actual $D = [0.11, 0.23, 0.38, 0.57, 0.82, 1.21, 2.03]$

From distances to trees

Ultrametric, additive, and general
distance matrices

1. Ultrametric distances

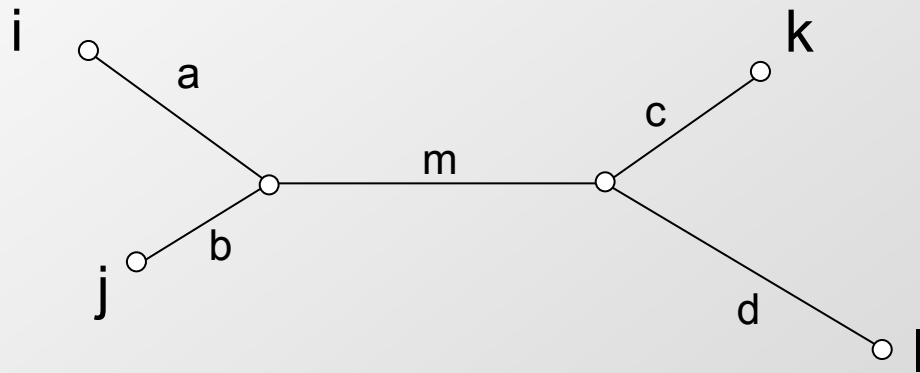
- For all points i, j, k
 - two distances are equal and third is smaller
- $$d(i,j) \leq d(i,k) = d(j,k)$$
- $$a+a \leq a+b = a+b$$



- Result:
 - All paths from labels are equidistant to the root
 - Rooted tree with uniform rates of evolution

2. Additive distances

- All distances satisfy the four-point condition
 - For all i, j, k, l :
 - $d(i, j) + d(k, l) \leq d(i, k) + d(j, l) = d(i, l) + d(j, k)$
 - $(a+b) + (c+d) \leq (a+m+c) + (b+m+d) = (a+m+d) + (b+m+c)$



- Result:
 - All pairwise distances obtained by traversing a tree

3. General distances

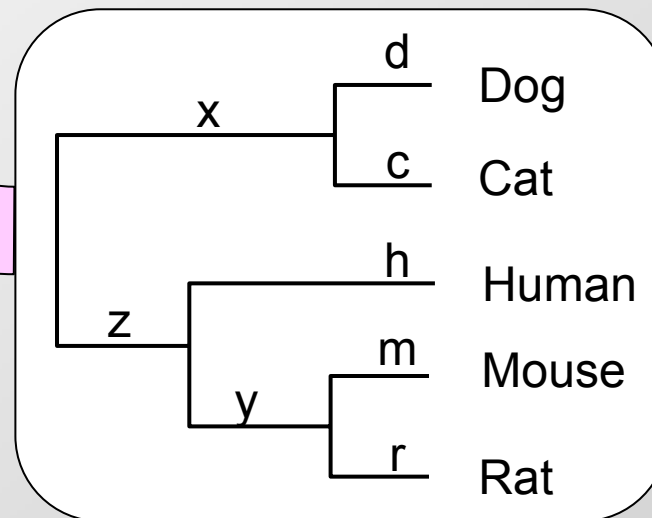
- In practice, a distance matrix is neither ultrametric nor additive
 - Noise
 - Measured distances are not exact
 - Evolutionary model is not exact
 - Fluctuations
 - Regions used to measure distances not representative of the species tree
 - Gene replacement (gene conversion), lateral transfer
 - Varying rates of mutation can lead to discrepancies
- In the general case, tree-building algorithms generate an approximation to the distance matrix
 - Such a tree can be obtained by
 - Enumeration and scoring of all trees (too expensive)
 - Neighbor-Joining (typically gives a good tree)
 - UPGMA (typically gives a poor tree)

Distance matrix \Leftrightarrow Phylogenetic tree

| | Hum | Mou | Rat | Dog | Cat |
|-------|---------|-----------|-----------|-----|-----|
| Human | 0 | 4 | 5 | 7 | 6 |
| Mouse | h.y.m | 0 | 3 | 8 | 5 |
| Rat | h.y.r | m.r | 0 | 9 | 7 |
| Dog | h.z.x.d | m.y.z.x.d | r.y.z.x.d | 0 | 2 |
| Cat | h.z.x.c | m.y.z.x.c | r.y.z.x.c | d.c | 0 |

Tree implies
a distance matrix

M_{ij}



Map distances D_{ij}
to a tree

$$\min \sum_{ij} (D_{ij} - M_{ij})^2$$

Goal:

Minimize discrepancy between **observed distances** and **tree-based distances**

Tree-building algorithms

Mapping a distance matrix to a tree

Algorithm: UPGMA

Initialization:

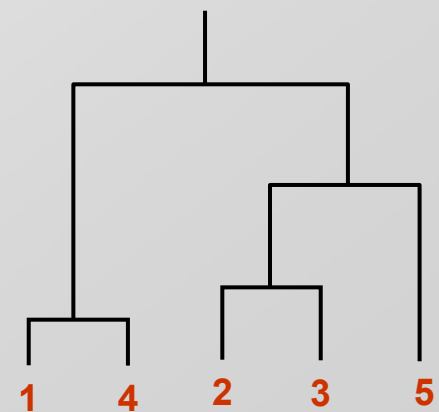
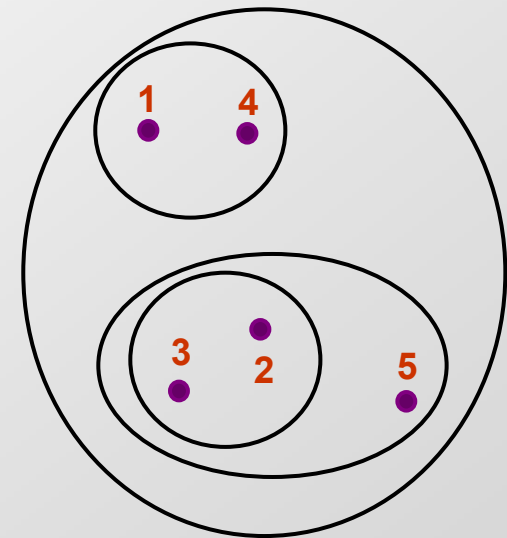
Assign each x_i into its own cluster C_i
Define one leaf per sequence, height 0

Iteration:

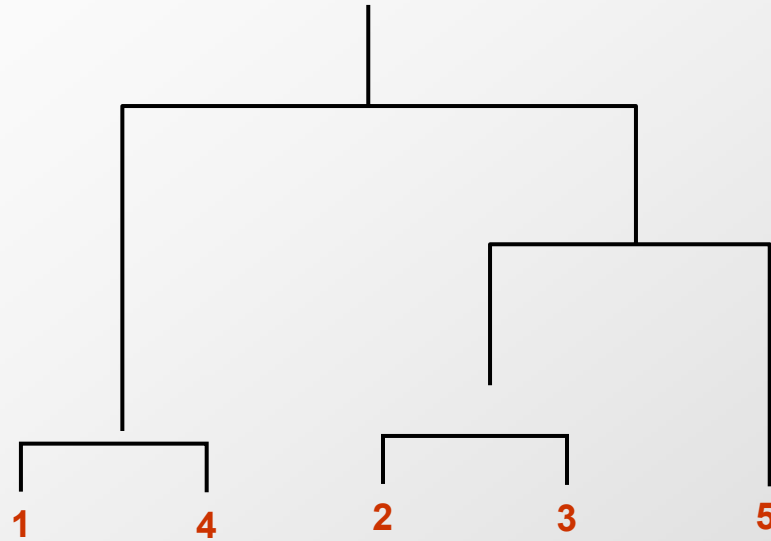
Find two clusters C_i, C_j s.t. d_{ij} is min
Let $C_k = C_i \cup C_j$
Define node connecting C_i, C_j ,
& place it at height $d_{ij}/2$
Delete C_i, C_j

Termination:

When two clusters i, j remain,
place root at height $d_{ij}/2$



Ultrametric Distances & UPGMA



UPGMA is guaranteed to build the correct tree if distance is ultrametric

Proof:

1. The tree topology is unique, given that the tree is binary
2. UPGMA constructs a tree obeying the pairwise distances

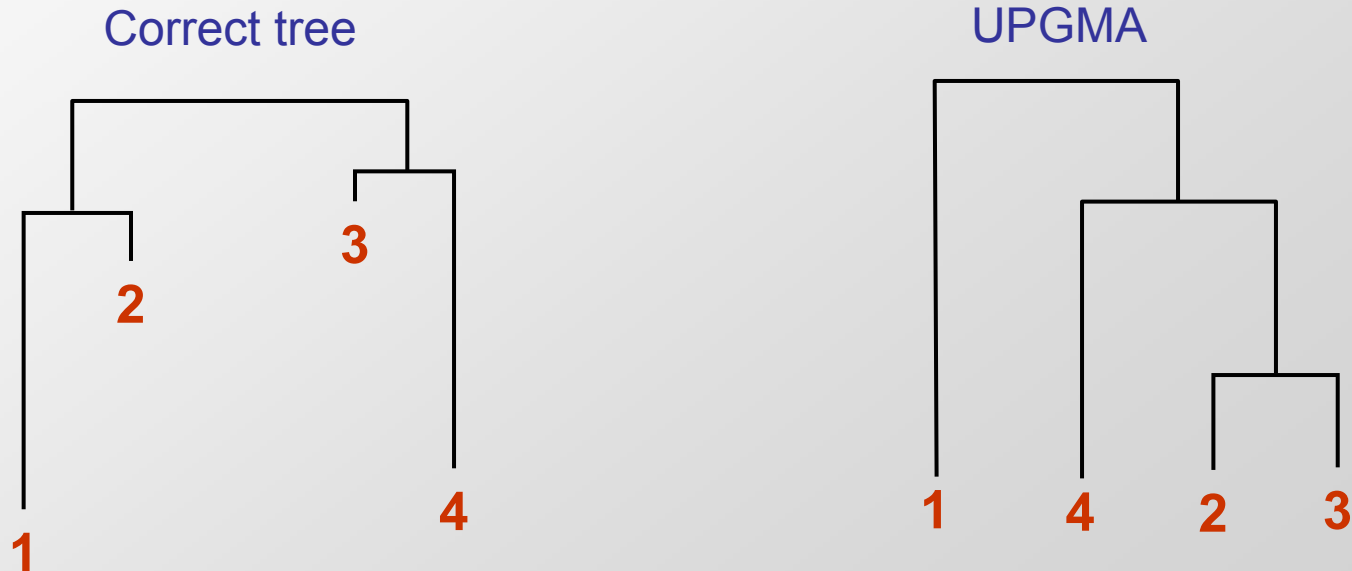
Weakness of UPGMA

Molecular clock assumption:

implies time is constant for all species

However, certain species (e.g., mouse, rat) evolve much faster

Example where UPGMA messes up:



Neighbor-Joining

- Guaranteed to produce the correct tree if distance is additive
- May produce a good tree even when distance is not additive

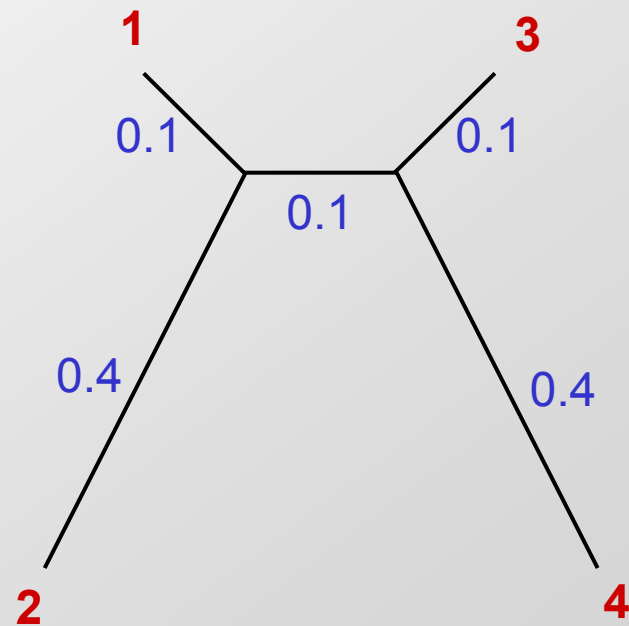
Step 1: Finding neighboring leaves

Define

$$D_{ij} = d_{ij} - (r_i + r_j)$$

Where

$$r_i = \frac{1}{|L| - 2} \sum_k d_{ik}$$



Claim: The above “magic trick” ensures that D_{ij} is minimal **iff** i, j are neighbors

Proof: Beyond the scope of this lecture (Durbin book, p. 189)

Algorithm: Neighbor-joining

Initialization:

Define T to be the set of leaf nodes, one per sequence

Let $L = T$

Iteration:

Pick i, j s.t. D_{ij} is minimal

Define a new node k , and set $d_{km} = \frac{1}{2} (d_{im} + d_{jm} - d_{ij})$ for all $m \in L$

Add k to T , with edges of lengths $d_{ik} = \frac{1}{2} (d_{ij} + r_i - r_j)$

Remove i, j from L ;

Add k to L

Termination:

When L consists of two nodes, i, j , and the edge between them of length d_{ij}

From alignments to trees

Parsimony

- One of the most popular methods

Idea:

Find the tree that explains the observed sequences with a minimal number of substitutions

Two computational sub-problems:

1. Find the parsimony cost of a given tree (easy)
2. Search through all tree topologies (hard)

Parsimony Scoring

Given a tree, and an alignment column

Label internal nodes to minimize the number of required substitutions

Initialization:

Set cost $C = 0$; $k = 2N - 1$

Iteration:

If k is a leaf, set $R_k = \{ x^k[u] \}$

If k is not a leaf,

Let i, j be the daughter nodes;

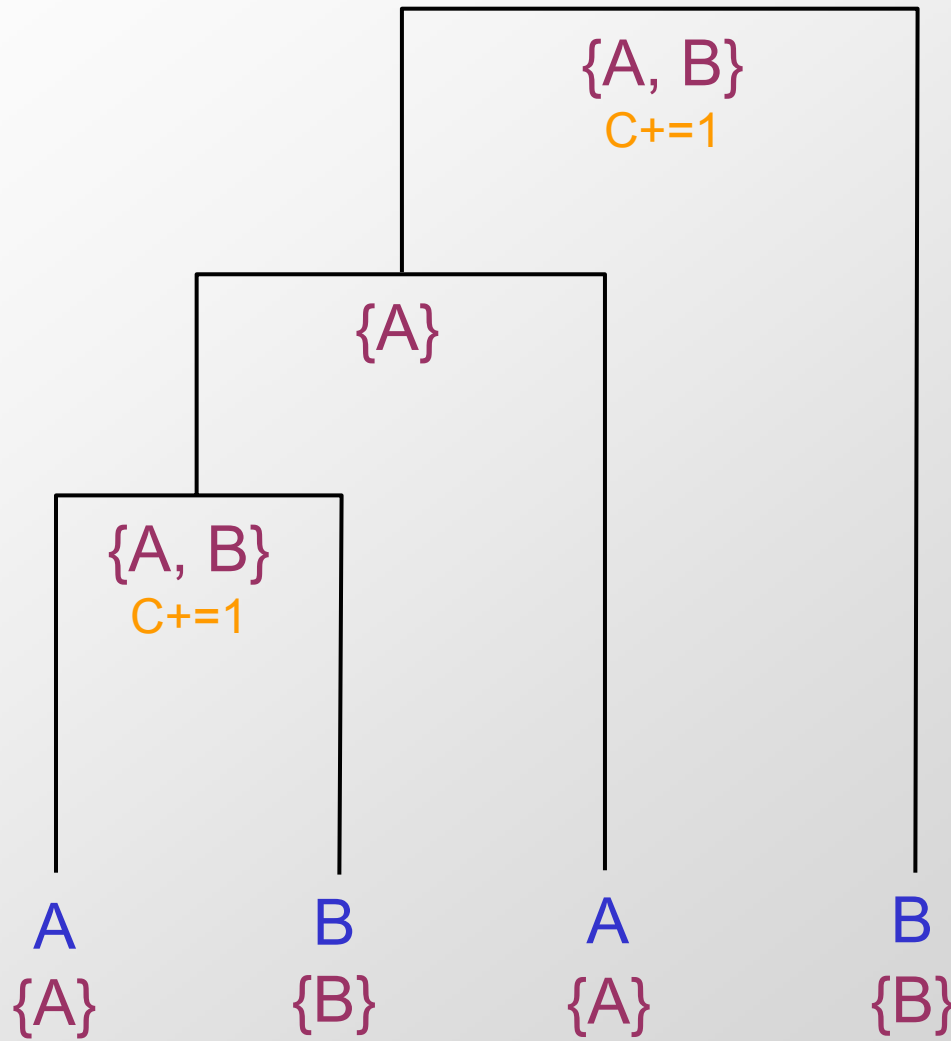
Set $R_k = R_i \cap R_j$ if intersection is nonempty

Set $R_k = R_i \cup R_j$, and $C += 1$, if intersection is empty

Termination:

Minimal cost of tree for column u , = C

Example



Traceback to find ancestral nucleotides

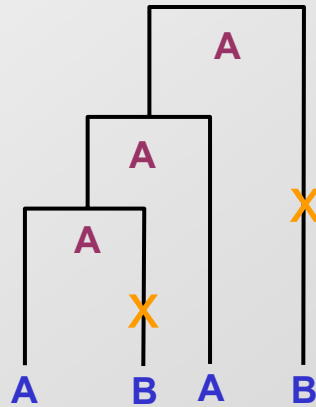
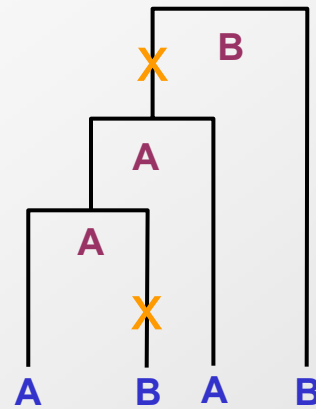
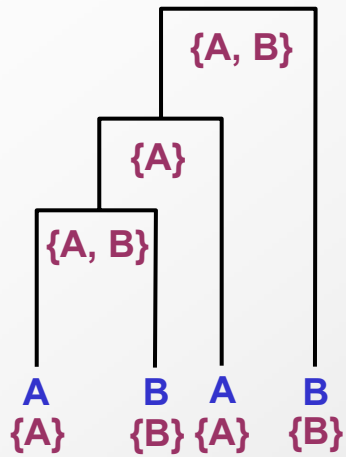
Traceback:

1. Choose an arbitrary nucleotide from R_{2N-1} for the root
2. Having chosen nucleotide r for parent k ,
If $r \in R_i$ choose r for daughter i
Else, choose arbitrary nucleotide from R_i

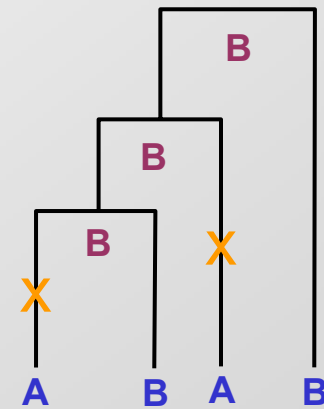
Easy to see that this traceback produces some assignment of cost C

Example

Accessible to traceback



Still optimal, but not found by traceback

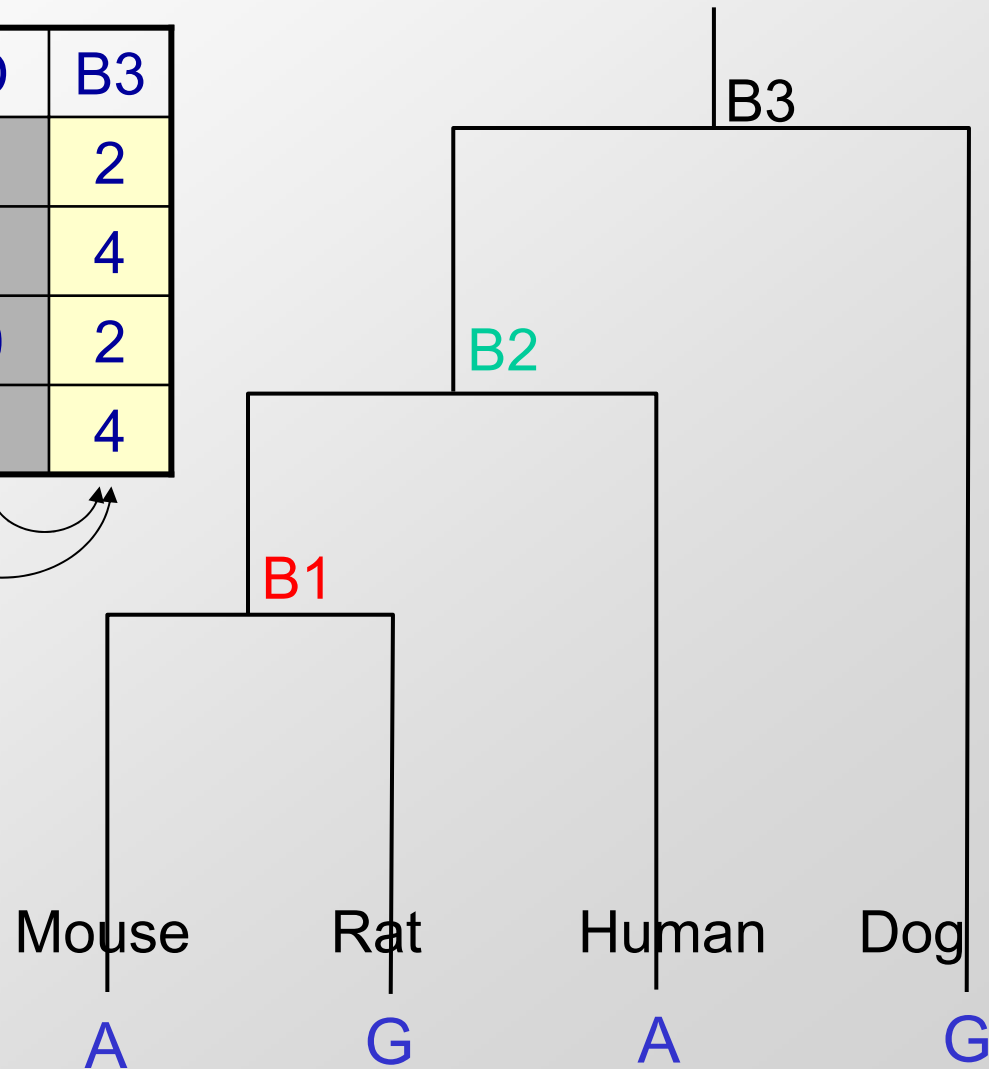


Parsimony with dynamic programming

| | M | R | B1 | H | B2 | D | B3 |
|---|---|---|----|---|----|---|----|
| A | 0 | 1 | 1 | 0 | 1 | 1 | 2 |
| C | 1 | 1 | 2 | 1 | 3 | 1 | 4 |
| G | 1 | 0 | 1 | 1 | 2 | 0 | 2 |
| T | 1 | 1 | 2 | 1 | 3 | 1 | 4 |



Update cost by walking up the tree.



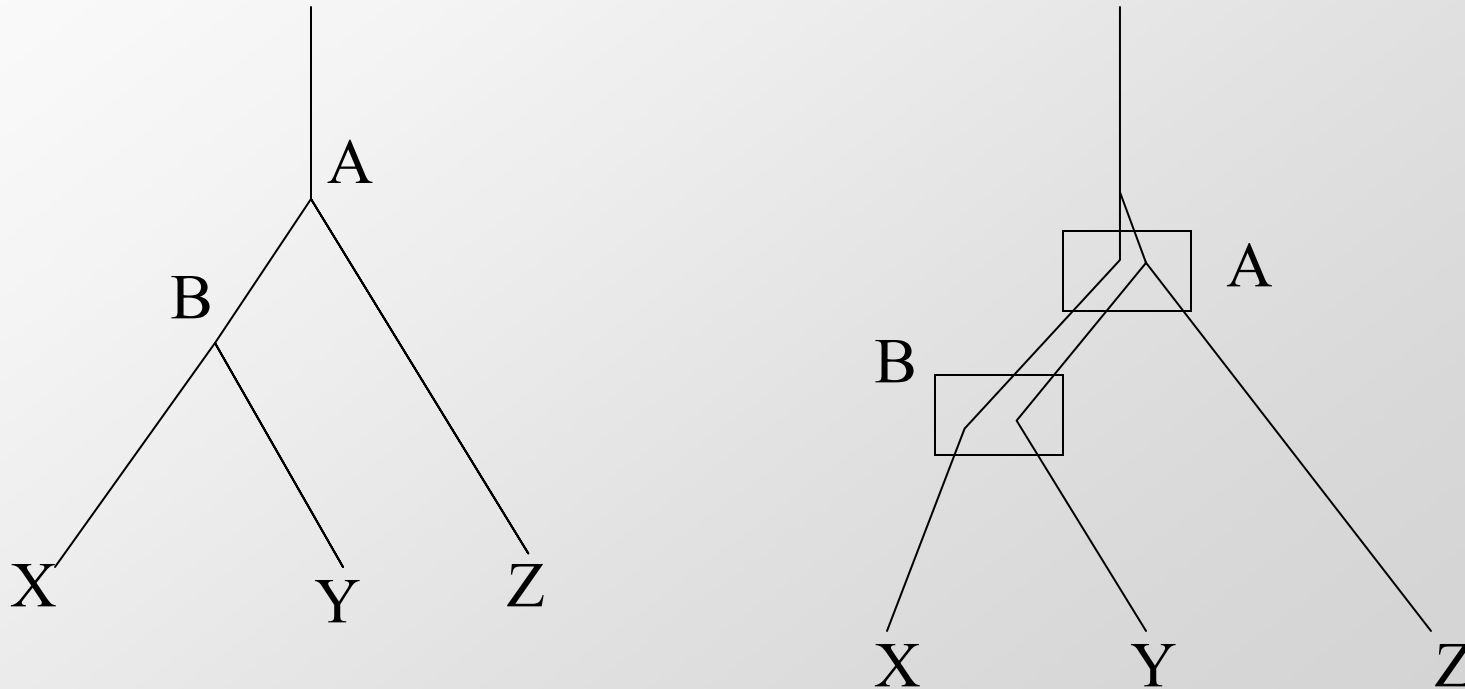
Bootstrapping to get the best trees

Main outline of algorithm

1. Select random columns from a multiple alignment – one column can then appear several times
2. Build a phylogenetic tree based on the random sample from (1)
3. Repeat (1), (2) many (say, 1000) times
4. Output the tree that is constructed most frequently

Gene vs. Species evolution

- Genes can start diverging before species separate
 - Genetic polymorphism within population could exist
 - After divergence, forms evolve differently in each species
 - Gene divergence could predate species divergence
 - Gene tree topology could be misleading



- Solution: Use multiple genes to infer a species tree

Summary

- **Modeling sequence evolution**
 - Probabilistic modeling of divergence
 - Jukes-Cantor, Kimura 2-parameter model
 - Probabilistic interpretation of sequence alignment
- **Phylogenetics**
 - Tree building from distance matrices
 - UPGMA / Neighbor Joining / Maximum Likelihood
 - Tree building from sequence alignments
 - Parsimony methods, set-based vs. dynamic programming
- **Multiple sequence alignment**
 - Scoring schemes for multiple alignment
 - Sum of pairs, consensus score, parsimony
 - Algorithms for multiple alignment
 - Multi-dimensional DP
 - Progressive alignment
 - Iterative refinement