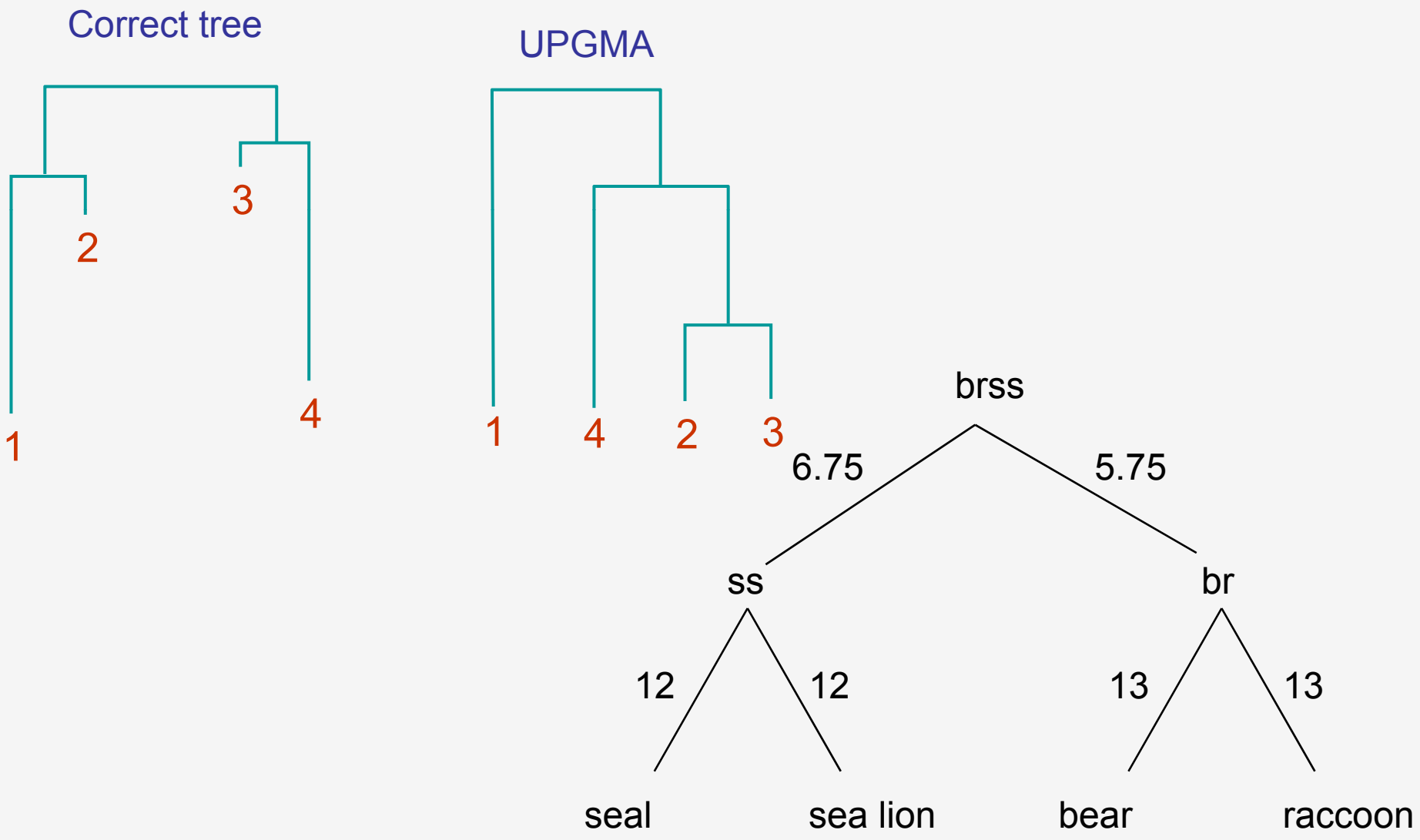
UPGMA's Weakness

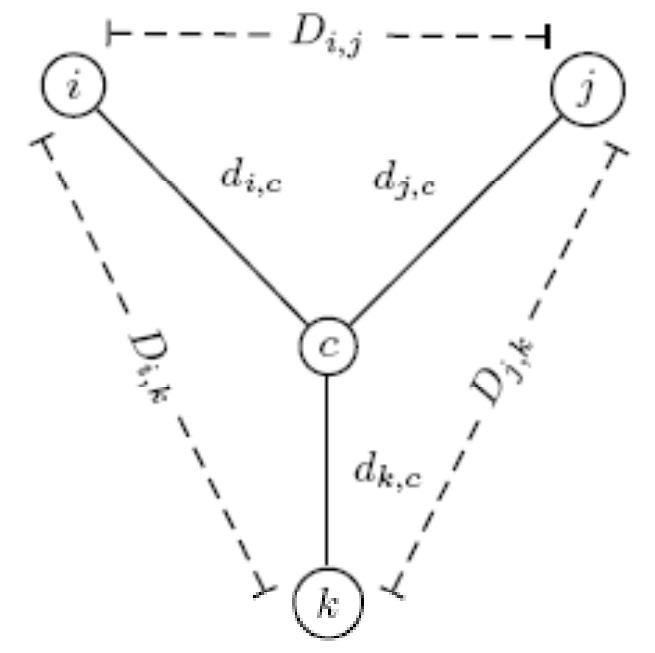
- The algorithm produces an **ultrametric** tree : the distance from the root to any leaf is the same
  - UPGMA assumes a constant molecular clock: all species represented by the leaves in the tree are assumed to accumulate mutations (and thus evolve) at the same rate. This is a major pitfall of UPGMA.

UPGMA's Weakness: Example



Clustering algorithm 2 – NJ (neighbor joining)

- Tree reconstruction for any 3x3 matrix is straightforward
- We have 3 leaves *i*, *j*, *k* and a center vertex *c*



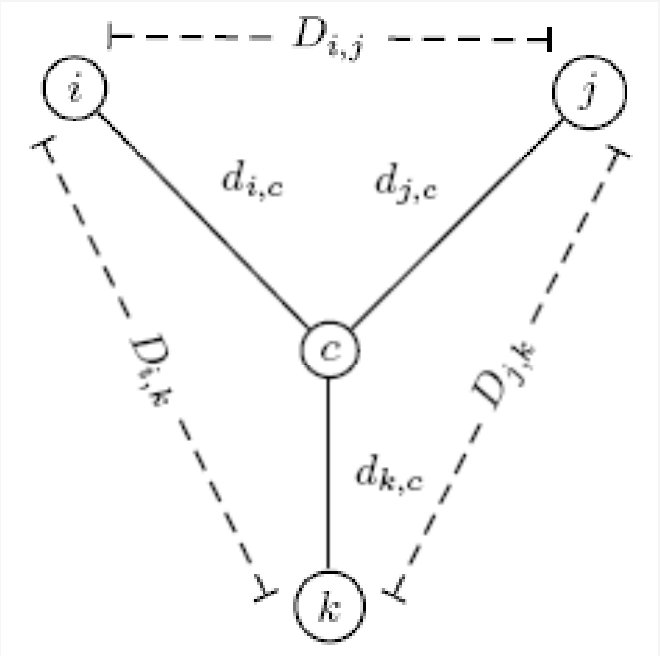
Observe D's, infer d's

$$d_{ic} + d_{jc} = D_{ij}$$

$$d_{ic} + d_{kc} = D_{ik}$$

$$d_{jc} + d_{kc} = D_{jk}$$

NJ –Cont'd



$$2d_{ic} + D_{jk} = D_{ij} + D_{ik}$$

$$d_{ic} = (D_{ij} + D_{ik} - D_{jk})/2$$

Similarly,

$$d_{jc} = (D_{ij} + D_{jk} - D_{ik})/2$$

$$d_{kc} = (D_{ki} + D_{kj} - D_{ij})/2$$

Trees with > 3 Leaves – NJ Cont'd

- An unrooted tree with *n* leaves has *2n-3* branches
- This means fitting a given tree to a distance matrix *D* requires solving a system of "n choose 2" equations with *2n-3* variables
- This is not always easy to solve for *large n*

NJ algorithm

- For each tip compute $u_i = \sum_{j:\, j \neq i}^{n} D_{ij} \,/\, (n-2)$
- Choose i and j for which, $D_{ij} - u_i - u_j$ is smallest
- Join nodes i and j to X. Compute branch length from i to X and j to X

$$v_{i \to X} = (D_{ij} + u_i - u_j)/2$$

$$v_{j \to X} = (D_{ij} + u_j - u_i)/2$$

- Compute the distance between X an remaining nodes

$$v_{X \to k} = (D_{ik} + D_{jk} - D_{ij})/2$$

NJ algorithm – Cont'd

- New node X is treated as a new tip and old nodes I, j are deleted

- If more than two nodes remain go back to step-1, else connect the two nodes (*l,m*) by $D_{l,m}$

Maximum likelihood approach

Tree that maximizes the likelihood of the observed data is optimal.

$$L = P(data \mid tree) = P(D \mid T) = \prod_{i=1}^{m} P(D^i \mid T)$$

Core Assumptions

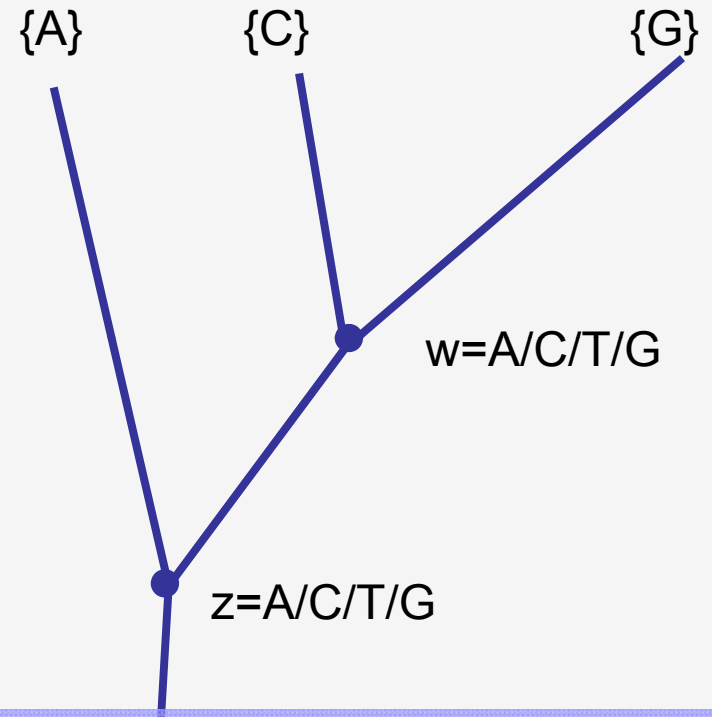
1. Evolution in different sites is independent
2. Evolution in different lineages is independent

Assumptions (the fine print):

1. A uniform evolutionary process operated across the entire tree.
2. The process of evolution is a homogeneous Markov process.

ML – Felsenstein's pruning method

$$P(D \mid T) = \prod_{i=1}^{m} P(D^i \mid T)$$



$$P(D^i \mid T) = \sum_{z=A/C/T/G} \sum_{w=A/C/T/G} P(z)P(A \mid z,t3) \; P(w \mid z,t4)P(C \mid w,t1)P(G \mid w,t2)$$

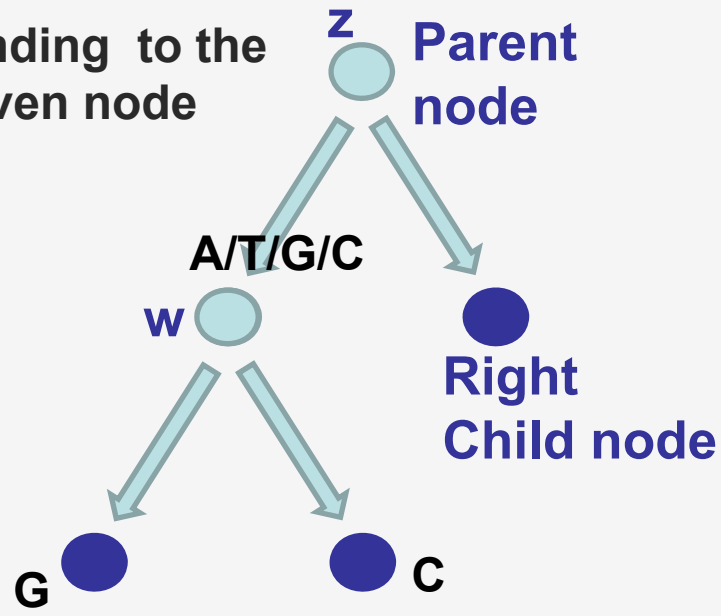
$$= \sum_{z=A/C/T/G} P(z)P(A \mid z,t3) \sum_{w=A/C/T/G} P(w \mid z,t4)P(C \mid w,t1)P(G \mid w,t2)$$

ML – Cont'd

Remember the sankoff algo?

- We maintained a Score for each state corresponding to the best parsimony score for everything above a given node
- Scores were summed for each branch

$$S_p(i) = S_p'(i)_{left} + S_p'(i)_{right}$$



- Similarly, we will define

$$L_p(i) = L_{c1|p1}'(i)_{left} . L_{c2|p2}'(i)_{right}$$

Total Likelihood Conditional on each parent node

$$L'_p(i)_{left} = \sum_{j=A/C/T/G} P(j|i, t_{parent\_i \to left\_child\_j}) L_{left\_child}(j)$$

$$L'_p(i)_{right} = \sum_{j=A/C/T/G} P(j|i, t_{parent\_i \to right\_child\_j}) L_{right\_child}(j)$$

ML – Intialization and final score

Total Likelihood Conditional on each parent node

$$L'_p(i)_{left} = \sum_{j=A/C/T/G} P(j|i, t_{parent\_i \rightarrow left\_child\_j}) L_{left\_child}(j)$$

$$L'_p(i)_{right} = \sum_{j=A/C/T/G} P(j|i, t_{parent\_i \rightarrow right\_child\_j}) L_{right\_child}(j)$$

We need the values at leaves, for each state:

This is obvious, if we have state A at a given leaf, L(A)=1 and the rest, L(C)/L(T)/L(G)=0 and so on.

Finally,
We need to add up all state values multiplied by their prior probability to be at the root – the priors are equivalent to the background probability

$$L(tree\_for\_a\_given\_site) = \sum_{j=A/C/T/G} \pi_j L_{root}(j)$$

More “realistic” approaches

- Allowing for rates to differ among sites
- Must not assume that we know the relative rate at individual sites
- Must allow some correlation between rates of evolution at adjacent sites

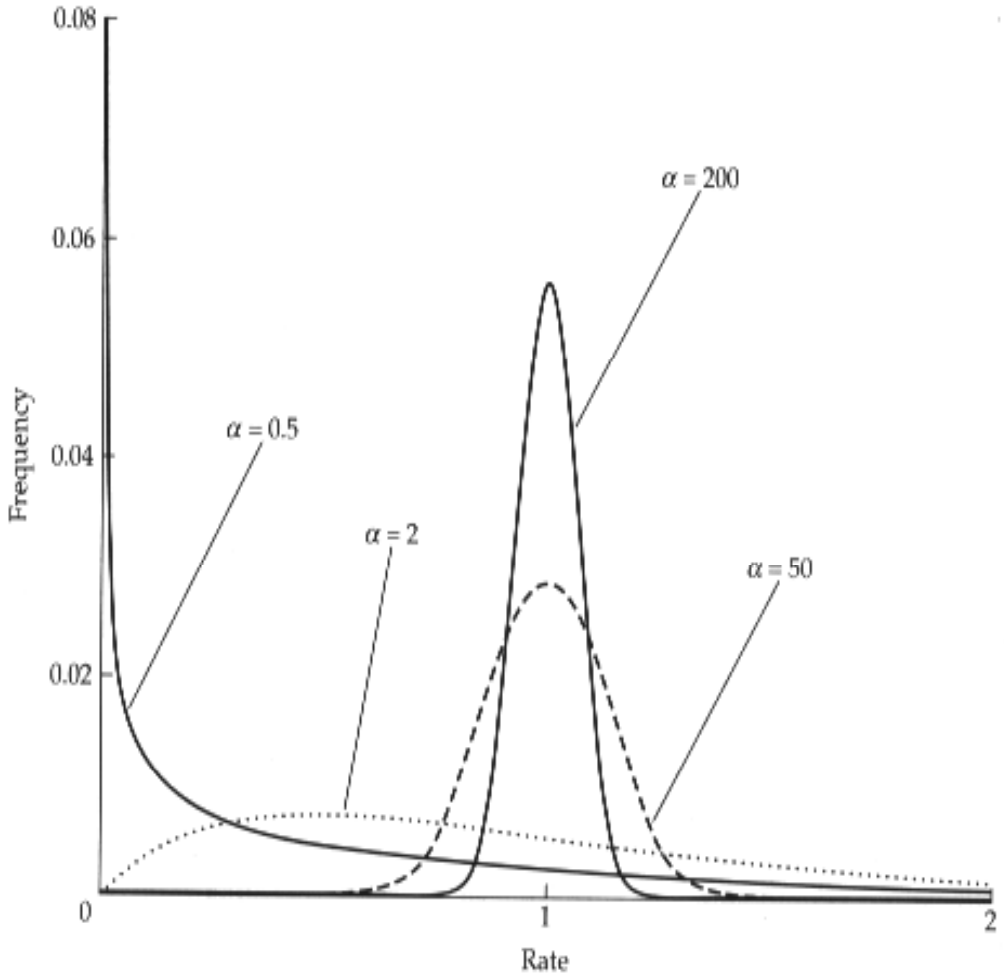
Rate heterogeneity and gamma distr

- Mutation rates vary considerably

  Commonly used: gamma distribution of rates across sequence sites.

The shape of the gamma distribution is controlled by a parameter α, and the distribution's mean and variance are 1 and 1/α, respectively. Large values of α (particularly α>1) give a bell curve-shaped distribution, suggesting little or no rate heterogeneity

$$f(r) = \frac{\alpha^{\alpha} r^{\alpha-1} e^{-\alpha r}}{\Gamma(\alpha)}$$



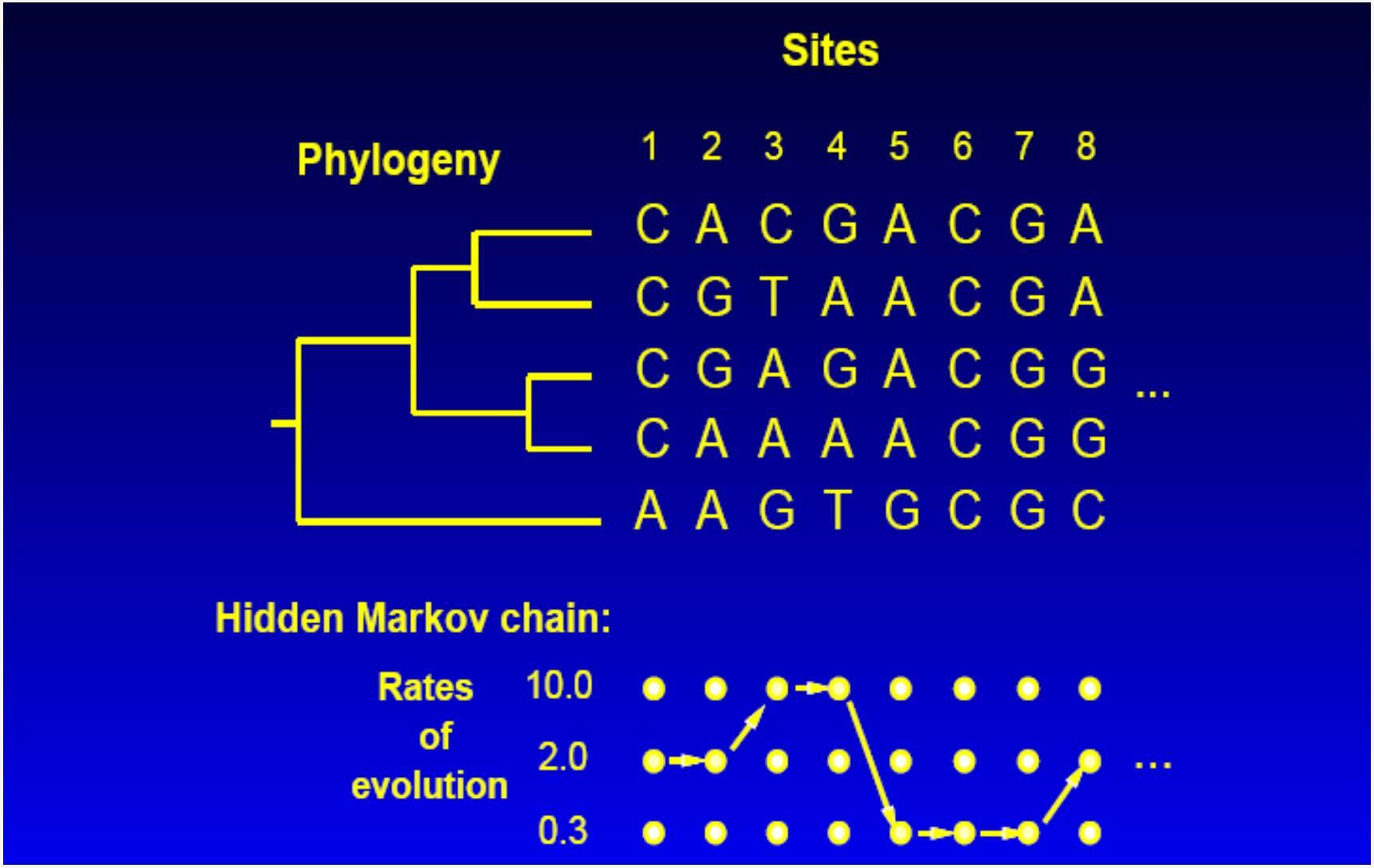
HMM for site-specific variation rates using ML

These are the most widely used models allowing rate variation to be correlated along the sequence.

We assume:

- There are a finite number of rates, m. Rate i is r_i.
- There are probabilities p_i of a site having rate i.
- A process not visible to us ("hidden") assigns rates to sites. It is a Markov process working along the sequence. For example it might have transition probability Prob $(j|i)$ of changing to rate j in the next site, given that it is at rate i in this site.
- The probability of our seeing some data are to be obtained by summing over all possible combinations of rates, weighting approproately by their probabilities of occurrence.

Site-specific rates and HMM



Likelihood calculation using HMM & rates

lets say Ci represent the category that a given site i belongs to,

$$L = \mathrm{P}(D \mid T) = \sum_{C1} \sum_{C2} .. \sum_{Cn} \{ P(C1, C2..Cn) . \prod_{1}^{n} P(Di \mid T, Ci)$$

$$where\ P(C1, C2..Cn) = P_{C1} . P_{C1.C2} . P_{C1.C3} ... P_{Cn-1.Cn} - \text{Markov Chain Probabilities}$$

$$L = \sum_{C1} \left\{ \sum_{C2} .... \sum_{Cn} P(C2..Cn \mid C1) \times \prod_{1}^{n} P(Di \mid T, Ci) \right\}$$

We can build a recursive relation to claculate L

$$L_{ck}^{k} = P(Dk \mid T, Ck) \sum_{Ck+1} P_{Ck.Ck+1} . L_{Ck+1}^{k+1}$$

Bootstrapping: Confidence in trees

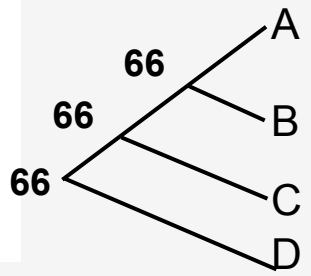
1. Select random columns from a multiple alignment – one column can then appear several times

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 or calculate a probability for each sub-tree topology

	1234		2214		3344
A	aggt	A	ggat	A	ggtt
B	aggt	B	ggat	B	aatt
C	tggc	C	ggtc	C	aatc
D	tcac	D	cctc	D	aacc



Phylogenetic software

Software packages

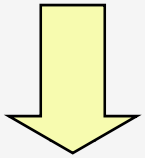
- Freely available
  - Phylip (widely used)
  - BioNJ
  - PhyML
  - Tree Puzzle
  - MrBayes
- Commercial
  - PAUP (widely used)
  - MEGA

Phylogenetic servers

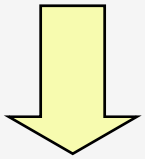
- http://www.phylogeny.fr/
- http://bioweb.pasteur.fr/seqanal/phylogeny/intro-uk.html
- http://atgc.lirmm.fr/phyml/
- http://phylobench.vital-it.ch/raxml-bb/
- http://www.fbsc.ncifcrf.gov/app/htdocs/appdb/drawpage.php?appname=PAUP
- http://power.nhri.org.tw/power/home.htm

Multiple sequence alignment using phylogenetic methods – Clustal-W

Pairwise alignment: calculation of distance matrix



Rooted NJ tree (guide tree) and calculation of sequence weights



Progressive alignment following the guide tree

```
human          ---MEEPQSDPSVEP-PLSQETFS 20
monkey         ---MEEPQSDPSIEP-PLSQETFS 20
mouse          MTAMEESQSDISLEL-PLSQETFS 23
rat            ---MEDSQSDMSIEL-PLSQETFS 20
xenopus        ---ME-PSSETGMDP-PLSQETES 19
chicken        ---MA-EEMEPLLEPTEVFMDLW- 19
                   *    . :  ::   :  :
```

Step 1-Calculation of Distance Matrix using pairwise alignment

Use the Distance Matrix to create a Guide Tree to determine the “order” of the sequences.

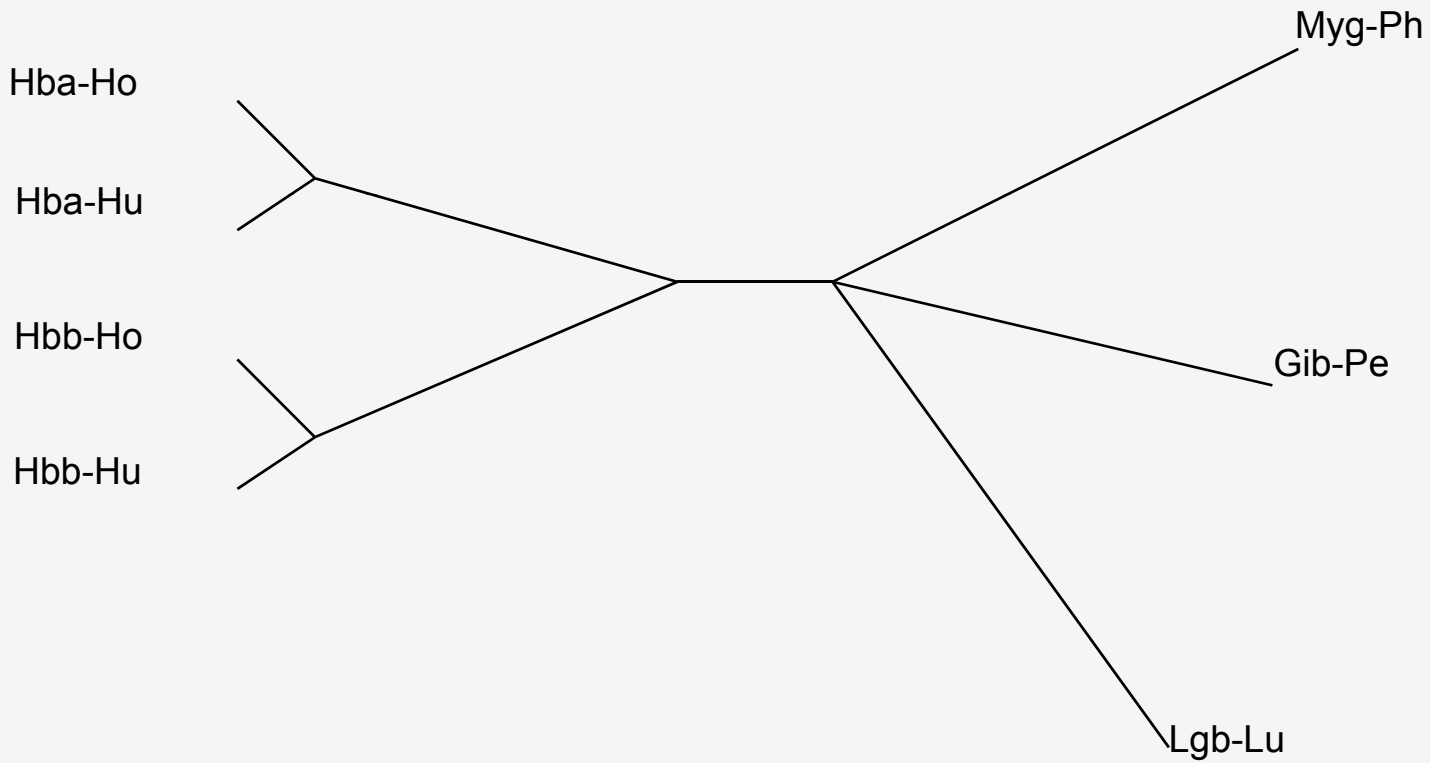
Hbb-Hu	**1**	**-**						
Hbb-Ho	**2**	**.17**	**-**					
Hba-Hu	**3**	**.59**	**.60**	**-**				
Hba-Ho	**4**	**.59**	**.59**	**.13**	**-**			
Myg-Ph	**5**	**.77**	**.77**	**.75**	**.75**	**-**		
Gib-Pe	**6**	**.81**	**.82**	**.73**	**.74**	**.80**	**-**	
Lgb-Lu	**7**	**.87**	**.86**	**.86**	**.88**	**.93**	**.90**	**-**
		1	**2**	**3**	**4**	**5**	**6**	**7**

D = 1 – (I)

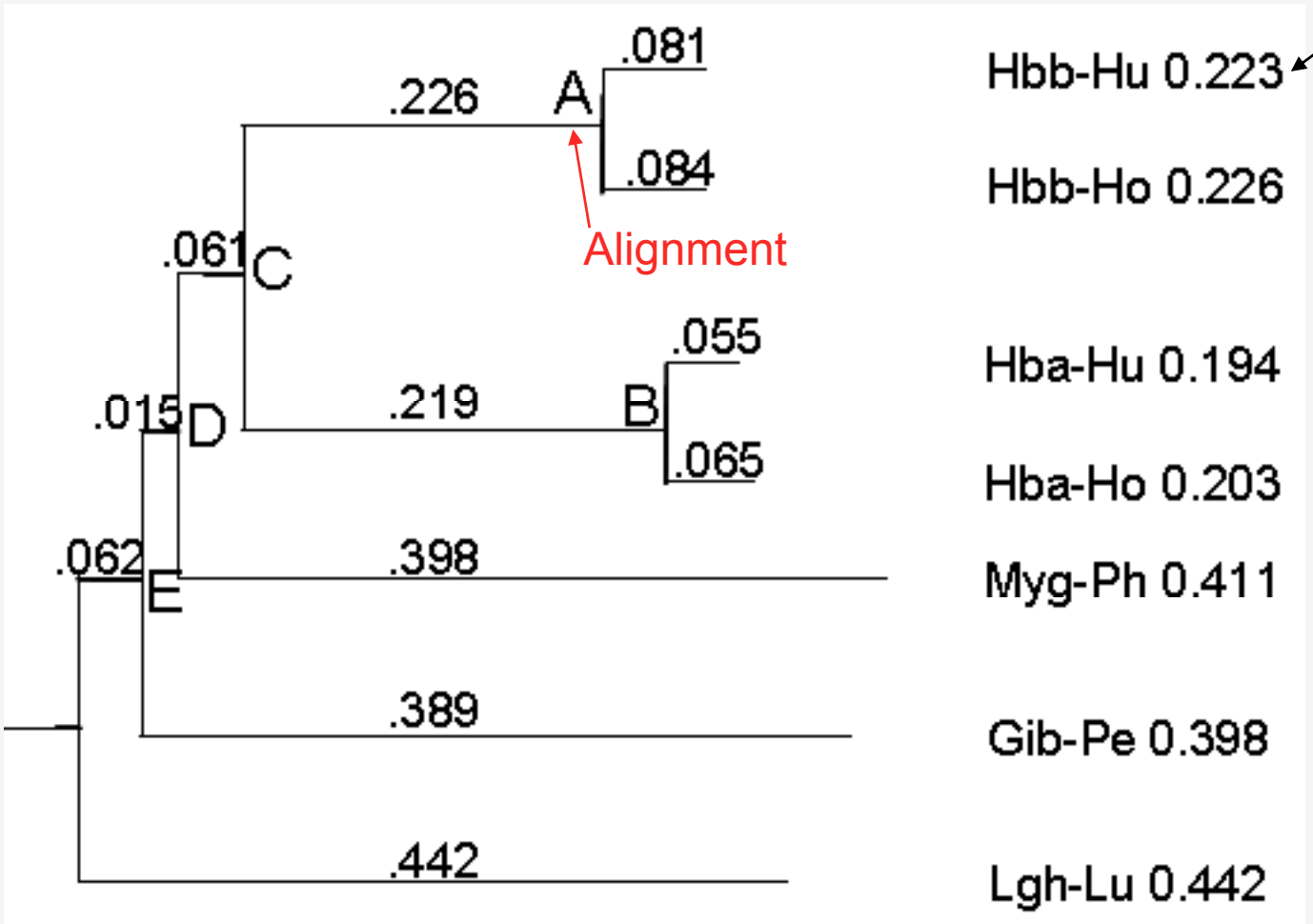
D = Difference score

$$I = \frac{\text{\# of identical aa's in pairwise global alignment}}{\text{total number of aa's in shortest sequence}}$$

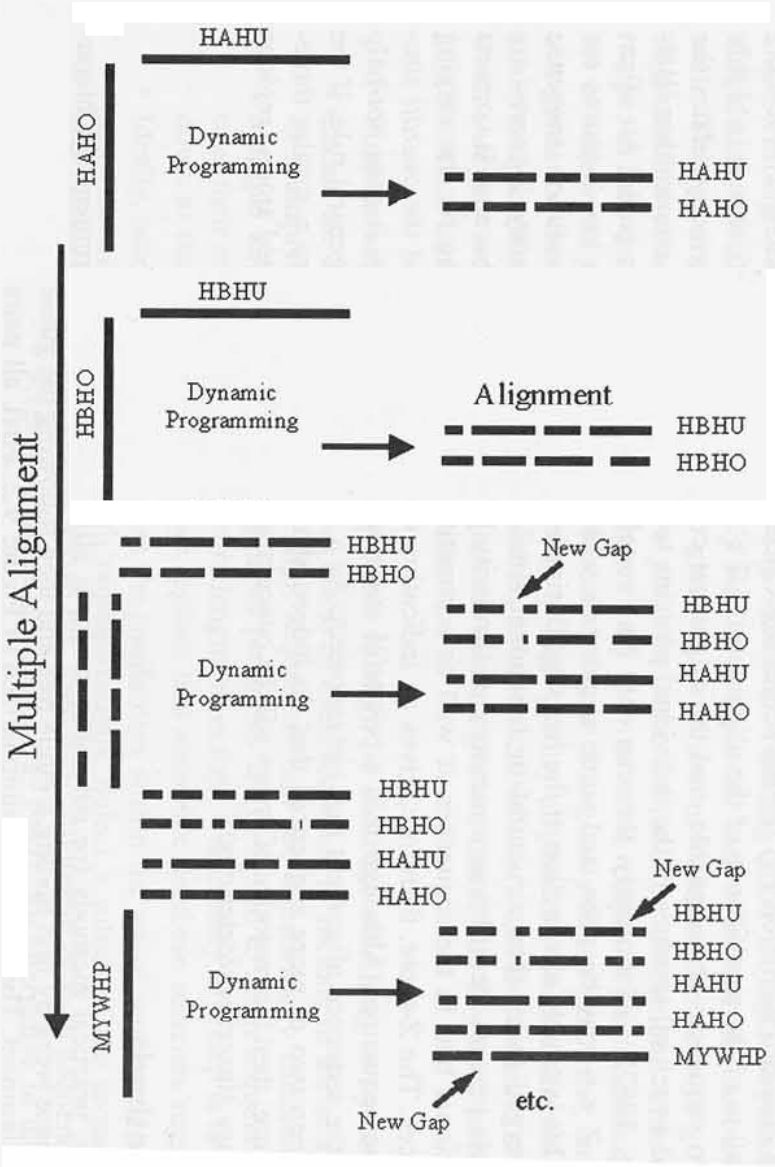
Step 2-Create unrooted tree using neighbor joining



Step 2-Create Rooted Tree and calculate weights

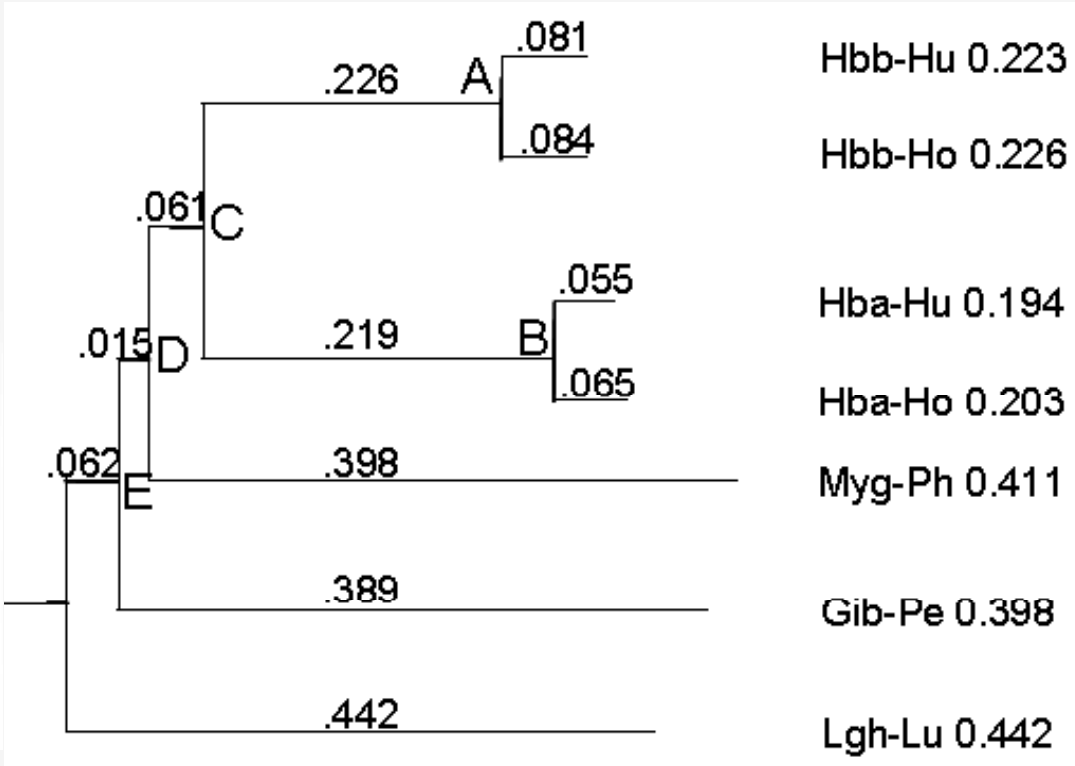


Step 3-Progressive alignment

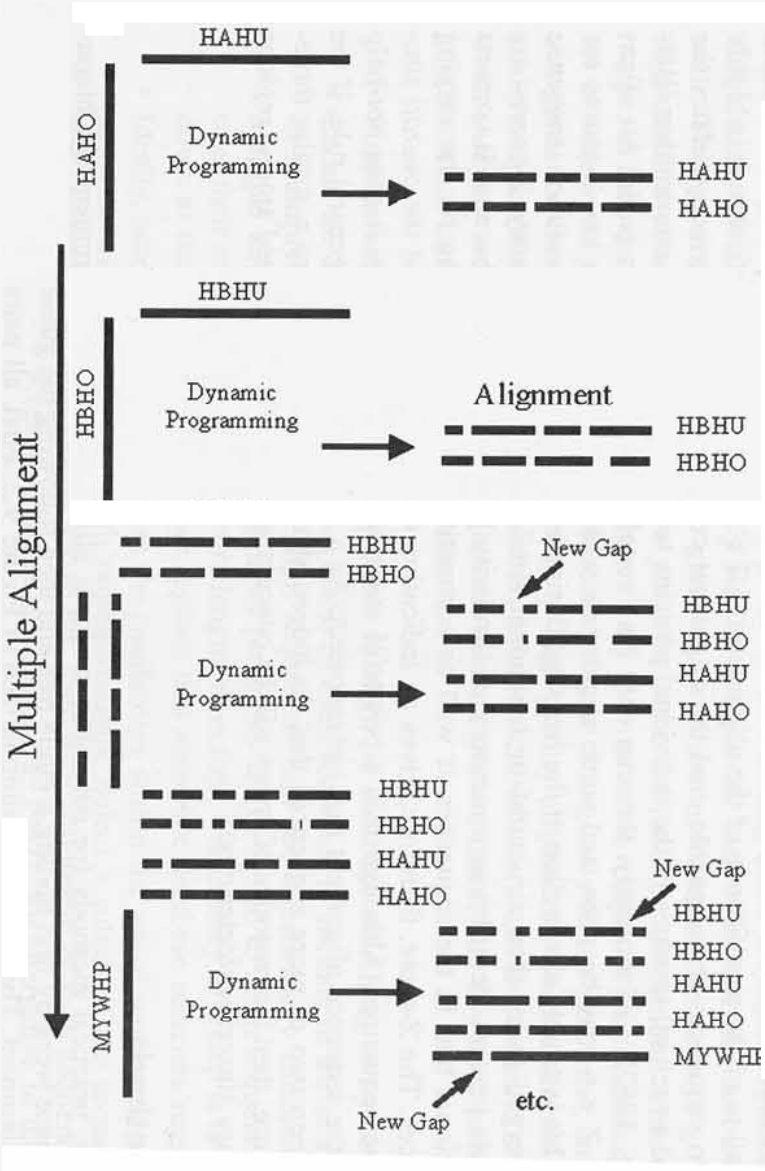


Order of alignment:

1 Hba-Hu vs Hba-Ho
2 Hbb-Hu vs Hbb-Ho
3 A vs B
4 Myg-Ph vs C
5 Gib-Pe vs D
6 Lgh-Lu vs E



Step 3-Progressive alignment



Scoring during progressive alignment

```
Set of 4:      1 eeksavtal
               2 eekaavlal
               3 adktnvkaa
               4 adktnvkaa

Set of 2:      5 gewqlvlhv
               6 aektkirsa
```

$$\text{Score} = M(t,v)*W_1*W_5 + M(t,i)*W_1*W_6 + M(l,v)*W_2*W_5 + M(l,i)*W_2*W_6 + M(k,v)*W_3*W_5 + M(k,i)*W_3*W_6 + M(k,v)*W_4*W_5 + M(k,i)*W_4*W_6$$

divided by 8

Recommended MSA Programs

- MUSCLE (fast and accurate)
- MAVID (genome-scale alignment)
- SAM (hidden markov, powerful and wide range of options)