Multiple Sequence Alignment

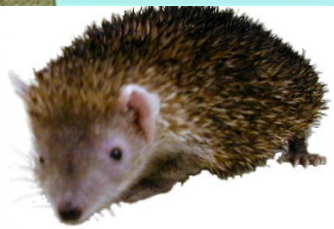






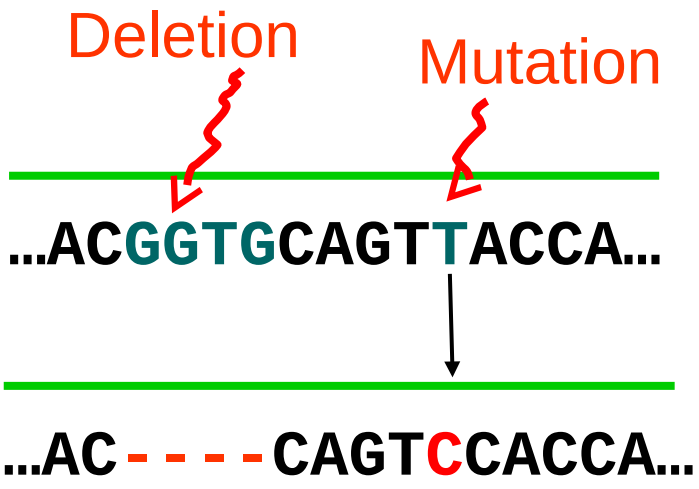






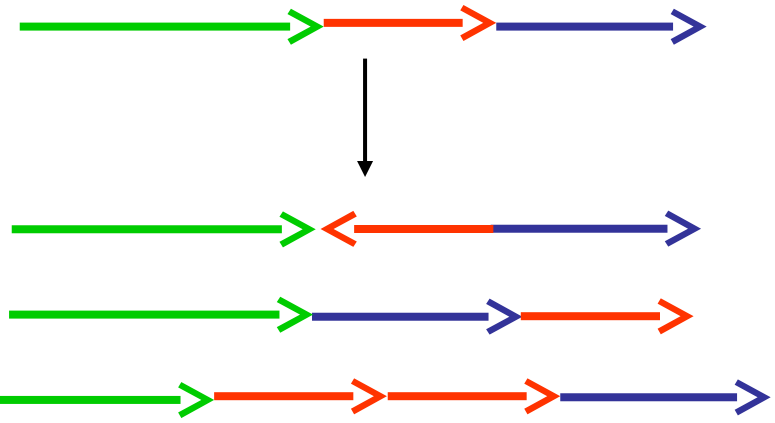



Evolution at the DNA level

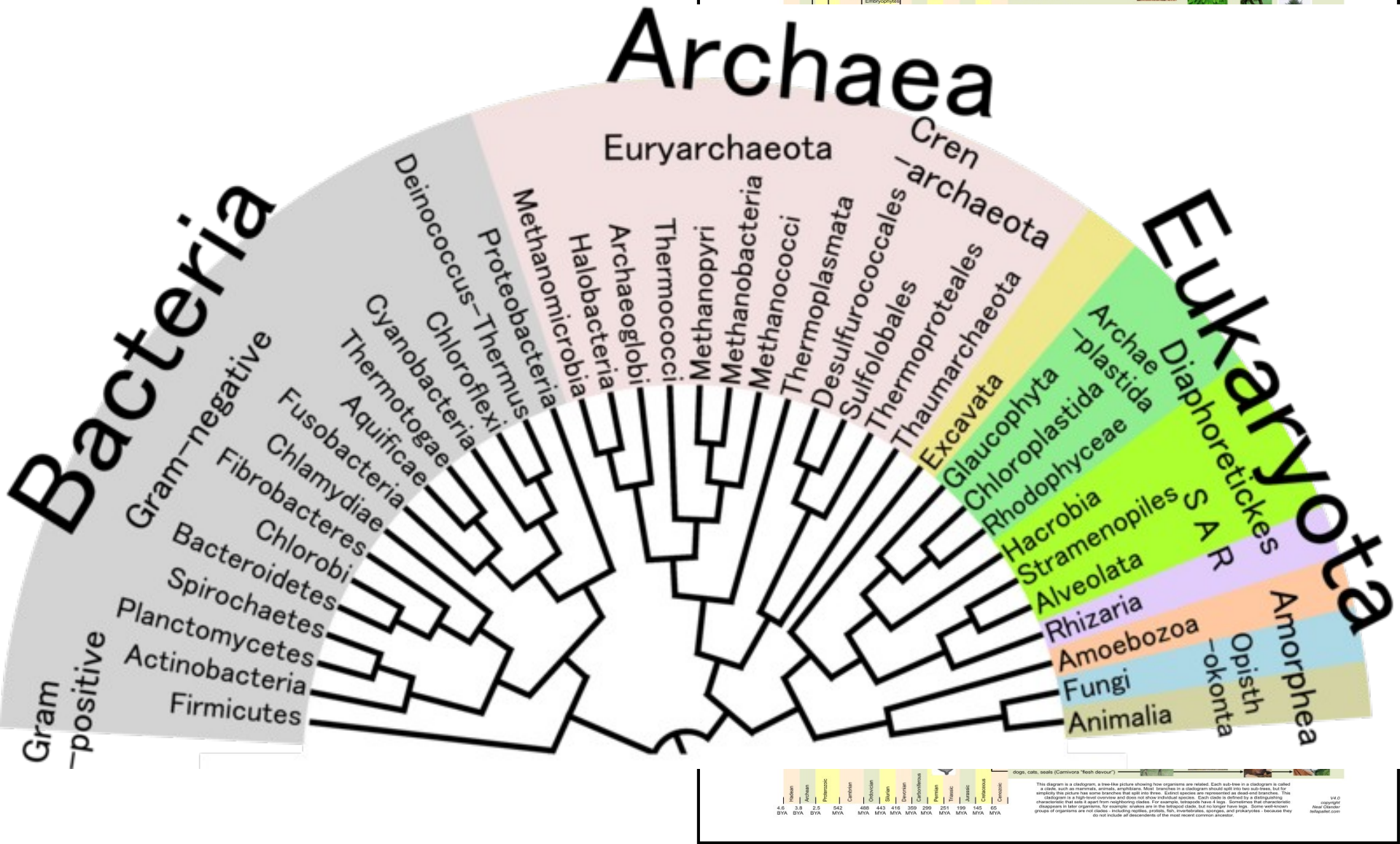


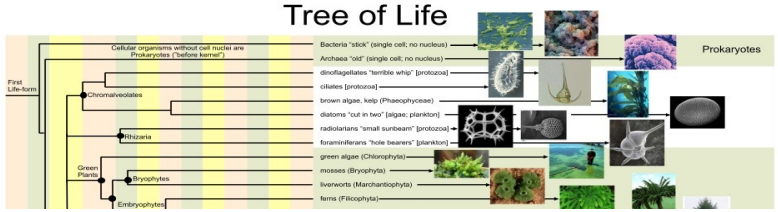
SEQUENCE EDITS

REARRANGEMENTS

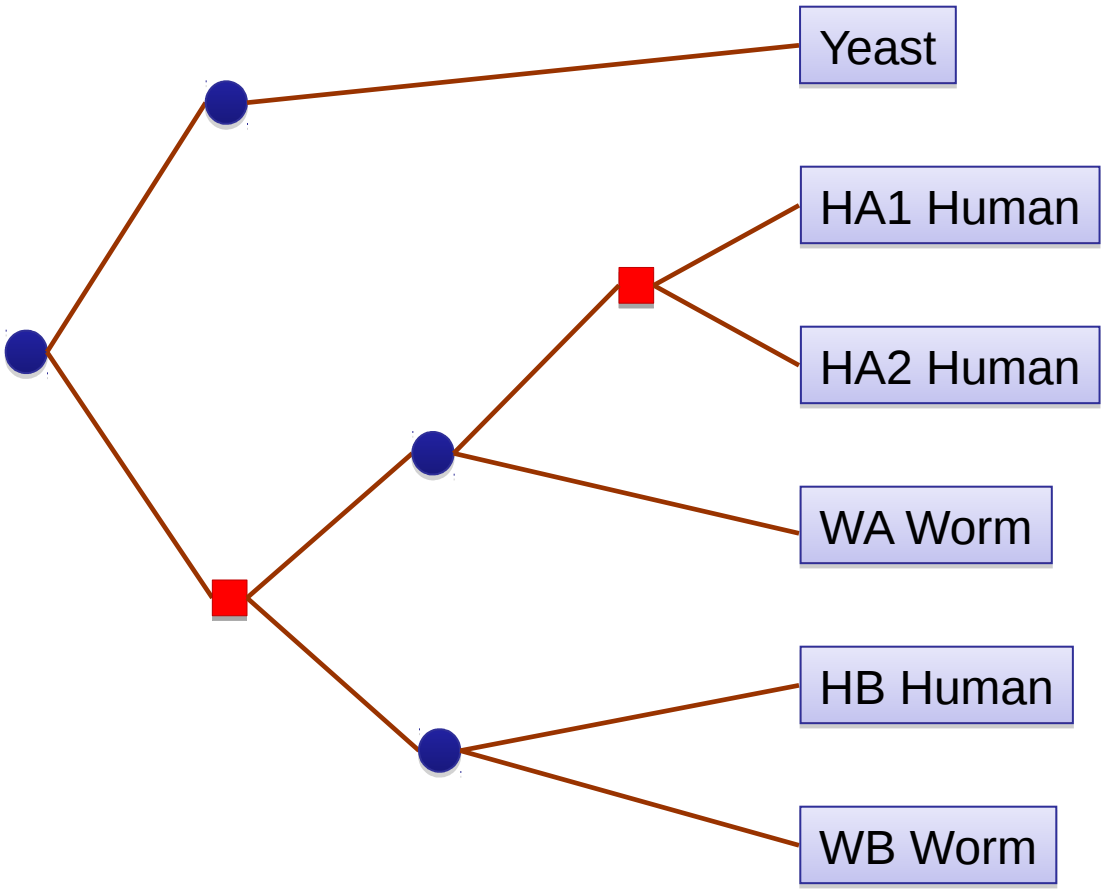


All Homologous Sequ





Orthology and Paralogy



Orthologs: *Derived by speciation*

Paralogs: *Everything else*

Orthology, Paralogy, Inparalogs, Outparalogs

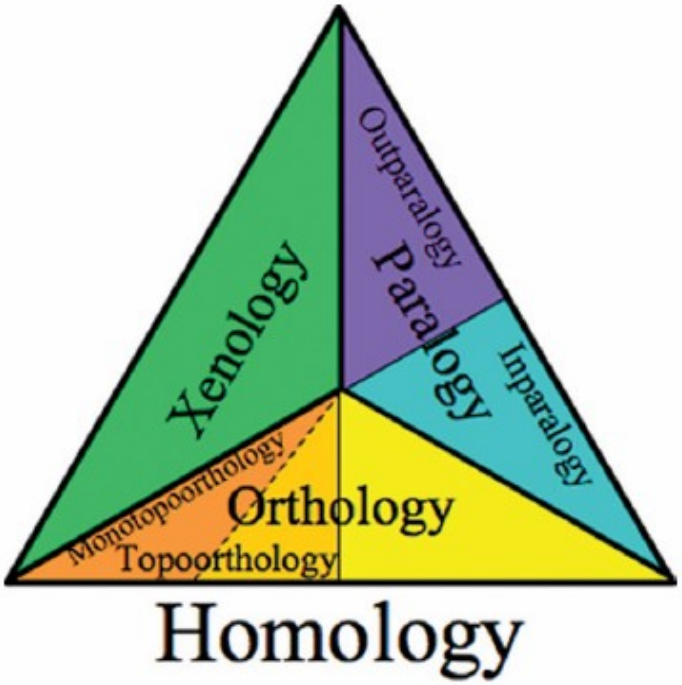


Figure 1. Refinements of homology.

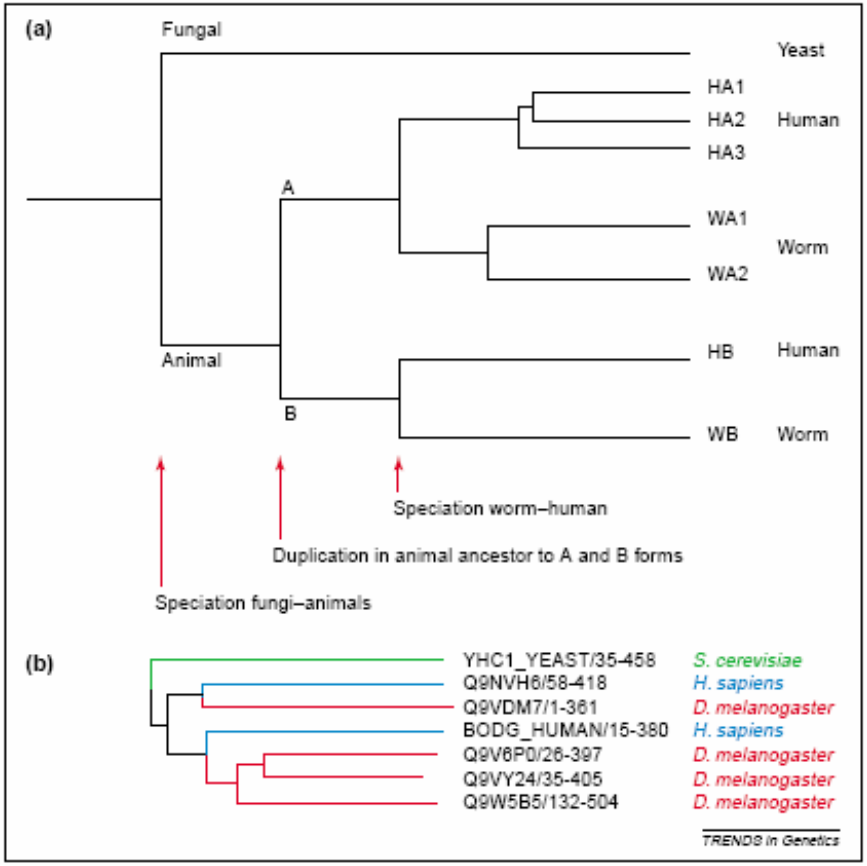
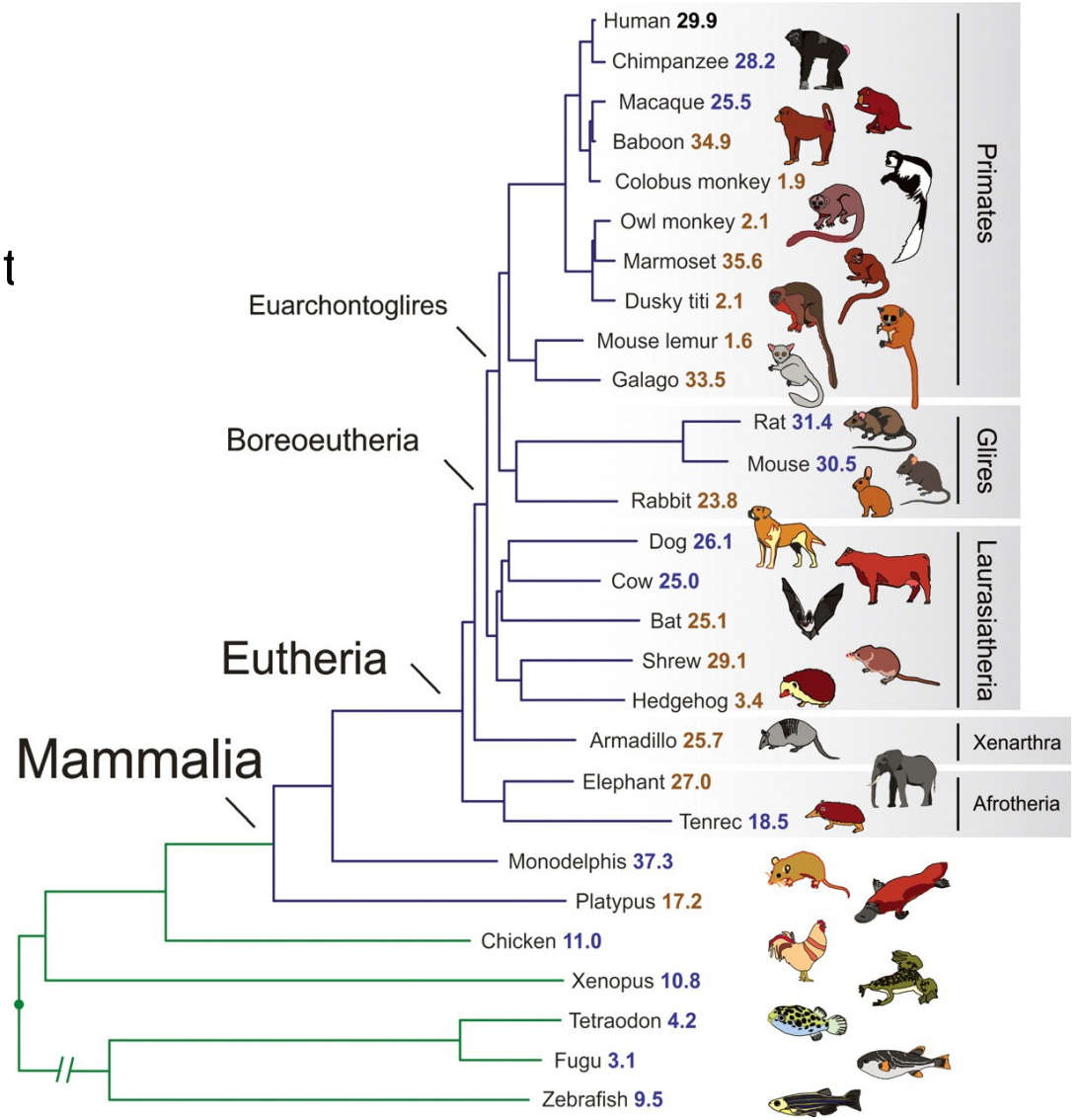


Fig. 1. The definition of inparalogs and outparalogs. (a) Consider an ancient gene inherited in the yeast, worm and human lineages. The gene was duplicated early in the animal lineage, before the human–worm split, into genes A and B. After the human–worm split, the A form was in turn duplicated independently in the human and worm lineages. In this scenario, the yeast gene is orthologous to all worm and human genes, which are all co-orthologous to the yeast gene. When comparing the human and worm genes, all genes in the HA* set are co-orthologous to all genes in the WA* set. The genes HA* are hence 'inparalogs' to each other when comparing human to worm. By contrast, the genes HB and HA* are 'outparalogs' when comparing human with worm..However, HB and HA*, and WB and WA* are inparalogs when comparing with yeast, because the animal–yeast split pre-dates the HA*–HB duplication. (b) Real-life example of inparalogs: γ-butyrobetaine hydroxylases. The points of speciation and duplication are easily identifiable. The alignment is a subset of Pfam:PF03322 and the tree was generated by neighbor-joining in Belvu. All nodes have a bootstrap support exceeding 95%.

Phylogenetic Trees

- Nodes: species
- Edges: time of independent evolution
- Edge length represents evolution time
  - AKA genetic distance
  - Not necessarily chronological time



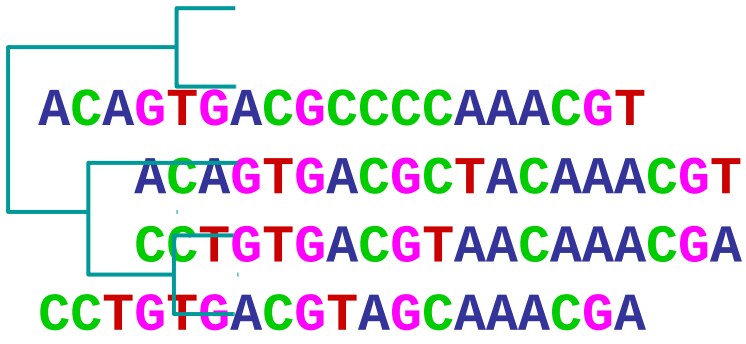
Inferring Phylogenetic Trees



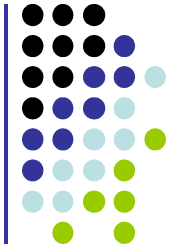

Trees can be inferred by several criteria:

- Morphology of the organisms
  - *Can lead to mistakes*
- Sequence comparison

Example:



Distance Between Two Sequences



Basic principle:

- Distance proportional to degree of independent sequence evolution

Given sequences x^i, x^j,

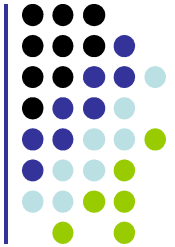
d_{ij} = distance between the two sequences

One possible definition:

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

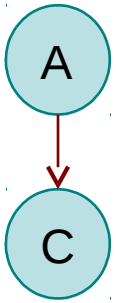
Better scores are derived by modeling evolution as a continuous change process

Molecular Evolution



Modeling sequence substitution:

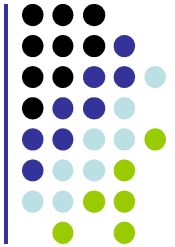
Consider what happens at a position for time Δt,



Δt ~= 0

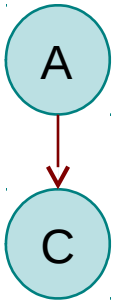
- P(t) = vector of probabilities of {A,C,G,T} at time t
- Given an alignment between two sequences, we can estimate P(t)
  - (Simplistic) Count non-match positions in the alignment
  - How do we estimate t from that information?

Molecular Evolution



Modeling sequence substitution:

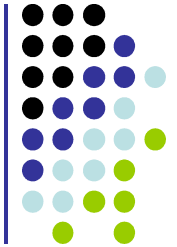
Consider what happens at a position for time Δt,



Δt ~= 0

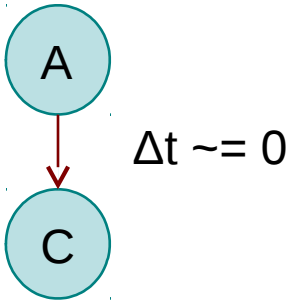
- P(t) = vector of probabilities of {A,C,G,T} at time t
- μ_{AC} = rate of transition from A to C per unit time
- $\mu_A = \mu_{AC} + \mu_{AG} + \mu_{AT}$ rate of transition out of A
- $p_A(t+\Delta t) = p_A(t) - p_A(t)\,\mu_A\,\Delta t + p_C(t)\,\mu_{CA}\,\Delta t + p_G(t)\,\mu_{GA}\,\Delta t + p_T(t)\,\mu_{TA}\,\Delta t$

Molecular Evolution



In matrix/vector notation, we get

$$P(t+\Delta t) = P(t) + Q\,P(t)\,\Delta t$$



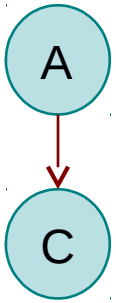
where Q is the substitution rate matrix

$$Q = \begin{pmatrix} -\mu_A & \mu_{GA} & \mu_{CA} & \mu_{TA} \\ \mu_{AG} & -\mu_G & \mu_{CG} & \mu_{TG} \\ \mu_{AC} & \mu_{GC} & -\mu_C & \mu_{TC} \\ \mu_{AT} & \mu_{GT} & \mu_{CT} & -\mu_T \end{pmatrix}$$

Molecular Evolution

- This is a differential equation:

$$P'(t) = Q\,P(t)$$

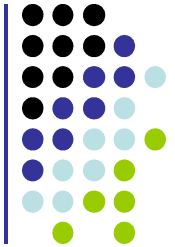


$\Delta t \sim= 0$

- Q => prob. distribution over {A,C,G,T} at each position, stationary (equilibrium) frequencies π_A, π_C, π_G, π_T

- Each Q is an evolutionary model
  - Some work better than others

Evolutionary Models



- Jukes-Cantor

$$Q = \begin{pmatrix} * & \frac{\mu}{4} & \frac{\mu}{4} & \frac{\mu}{4} \\ \frac{\mu}{4} & * & \frac{\mu}{4} & \frac{\mu}{4} \\ \frac{\mu}{4} & \frac{\mu}{4} & * & \frac{\mu}{4} \\ \frac{\mu}{4} & \frac{\mu}{4} & \frac{\mu}{4} & * \end{pmatrix}$$

- Kimura

$$Q = \begin{pmatrix} * & \kappa & 1 & 1 \\ \kappa & * & 1 & 1 \\ 1 & 1 & * & \kappa \\ 1 & 1 & \kappa & * \end{pmatrix}$$

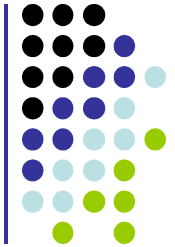
- Felsenstein

$$Q = \begin{pmatrix} * & \pi_T & \pi_T & \pi_T \\ \pi_C & * & \pi_C & \pi_C \\ \pi_A & \pi_A & * & \pi_A \\ \pi_G & \pi_G & \pi_G & * \end{pmatrix}$$

- HKY

$$Q = \begin{pmatrix} * & \kappa\pi_T & \pi_T & \pi_T \\ \kappa\pi_C & * & \pi_C & \pi_C \\ \pi_A & \pi_A & * & \kappa\pi_A \\ \pi_G & \pi_G & \kappa\pi_G & * \end{pmatrix}$$

Estimating Distances



- Solve the differential equation and compute expected evolutionary time given sequences

$$P'(t) = Q\,P(t)$$

Jukes-Cantor:

Let $P_{AA}(t) = P_{CC}(t) = P_{CC}(t) = P_{CC}(t) = r$

$P_{AC}(t) = \ldots = P_{TG}(t) = s$

Then,

$r'(t) = -\,¾\,r(t)\,\mu + ¾\,s(t)\,\mu$

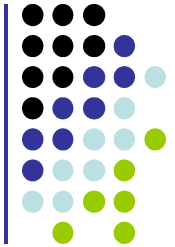
$s'(t) = -\,¼\,s(t)\,\mu + ¼\,r(t)\,\mu$

Which is satisfied by

$r(t) = ¼\,(1 + 3e^{-\mu t})$

$s(t) = ¼\,(1 - e^{-\mu t})$

Estimating Distances



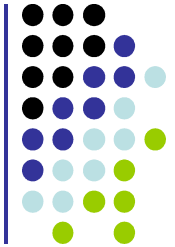
- Solve the differential equation and compute expected evolutionary time given sequences

$$P'(t) = Q\,P(t)$$

Jukes-Cantor:

$$P = \begin{pmatrix} \frac{1}{4} + \frac{3}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} \\ \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} + \frac{3}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} \\ \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} + \frac{3}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} \\ \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} + \frac{3}{4}e^{-t\mu} \end{pmatrix}$$

Estimating Distances



Let p = probability a base is different between two sequences,

Solve to find **t**

- Jukes-Cantor $r(t) = 1 - p = \frac{1}{4}(1 + 3e^{-\mu t})$

$p = \frac{3}{4} - \frac{3}{4} e^{-\mu t}$

$\frac{3}{4} - p = \frac{3}{4} e^{-\mu t}$

$1 - 4p/3 = e^{-\mu t}$

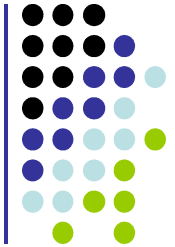
Therefore,

$\mu t = - \ln(1 - 4p/3)$

Letting $d = \frac{3}{4} \mu t$, denoting substitutions per site,

$$d = -\frac{3}{4}\ln(1 - \frac{4}{3}p)$$

Estimating Distances



d: Branch length in terms of substitutions per site

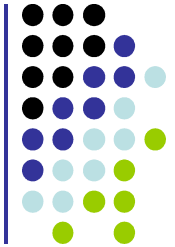
- Jukes-Cantor

$$d = -\frac{3}{4}\ln(1 - \frac{4}{3}p)$$

- Kimura

$$d = -\frac{1}{2}\ln(1 - 2P - Q) - \frac{1}{4}\ln(1 - 2Q)$$

A simple clustering method for building tree



UPGMA (unweighted pair group method using arithmetic averages)
Or the **Average Linkage Method**

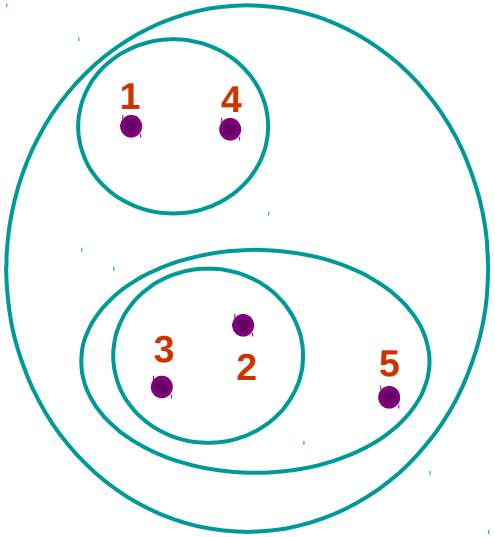
Given two disjoint clusters C_i, C_j of sequences,

$$d_{ij} = \frac{1}{|C_i| \times |C_j|} \Sigma_{\{p \in Ci,\ q \in Cj\}} d_{pq}$$

Claim that if $C_k = C_i \cup C_j$, then distance to another cluster C_l is:

$$d_{kl} = \frac{d_{il}\,|C_i| + d_{jl}\,|C_j|}{|C_i| + |C_j|}$$

Algorithm: Average Linkage



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

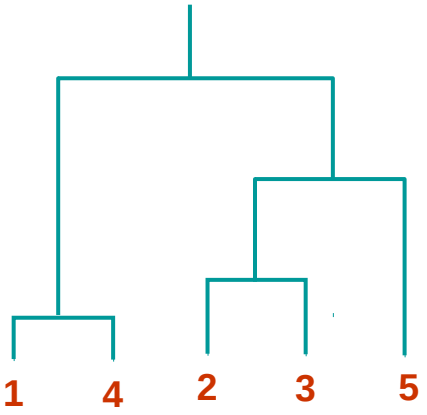
Let $C_k = C_i \cup C_j$

Define node connecting C_i, C_j, and place it at height $d_{ij}/2$

Delete C_i, C_j

Termination:

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



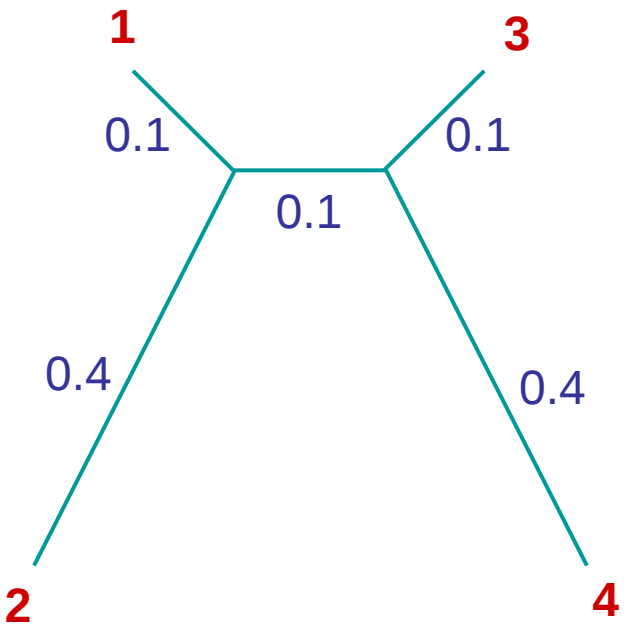
Neighbor-Joining

- 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

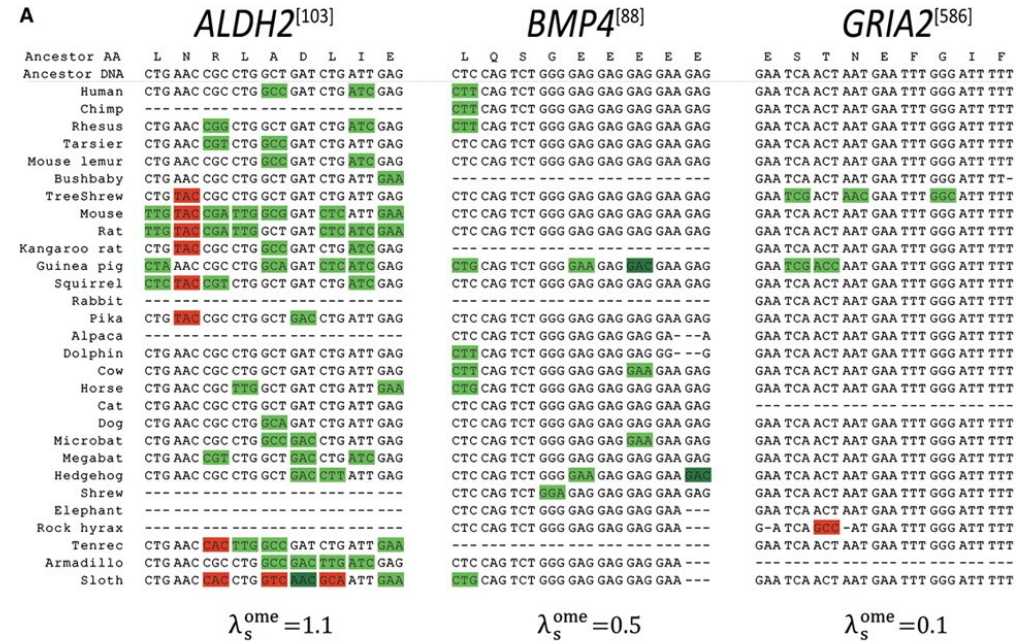
Define

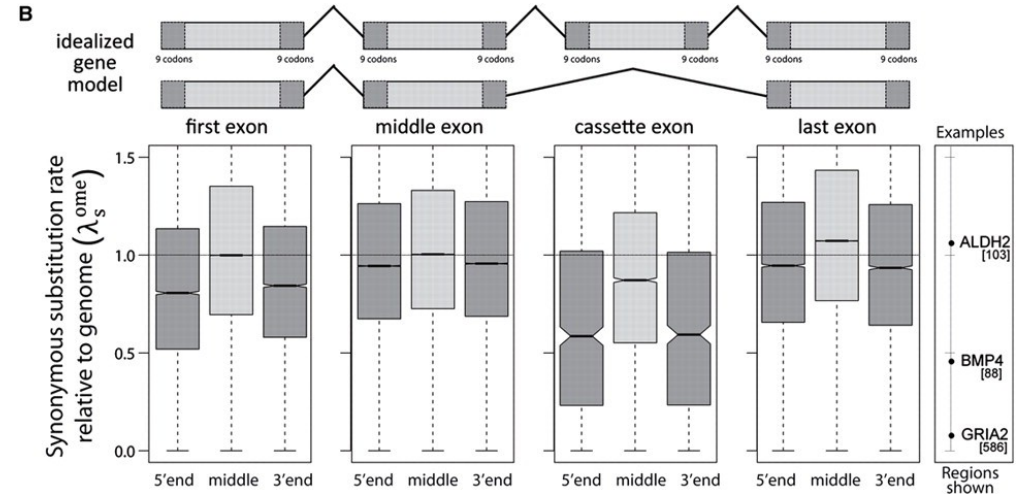
$$D_{ij} = (N - 2)\, d_{ij} - \sum_{k \neq i} d_{ik} - \sum_{k \neq j} d_{jk}$$

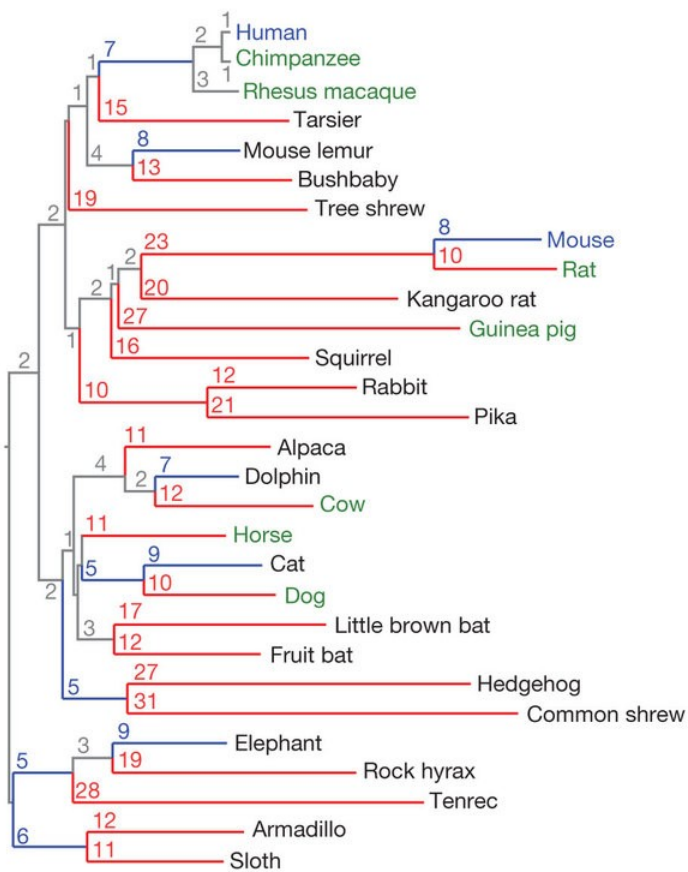


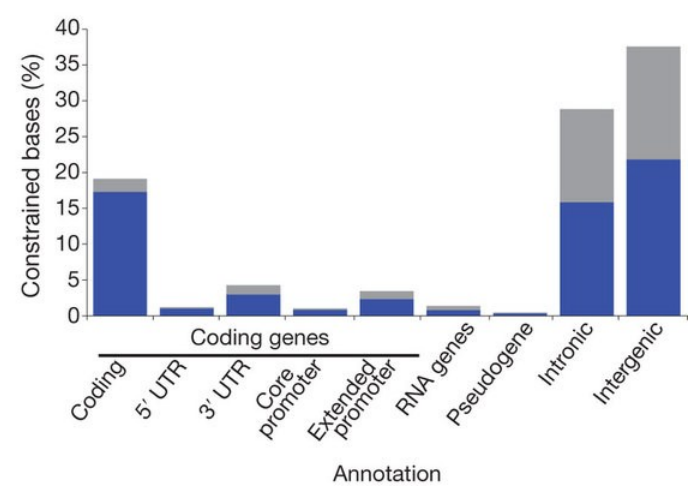
Claim: The above "magic trick" ensures that i, j are neighbors if D_{ij} is minimal

Mammalian alignments

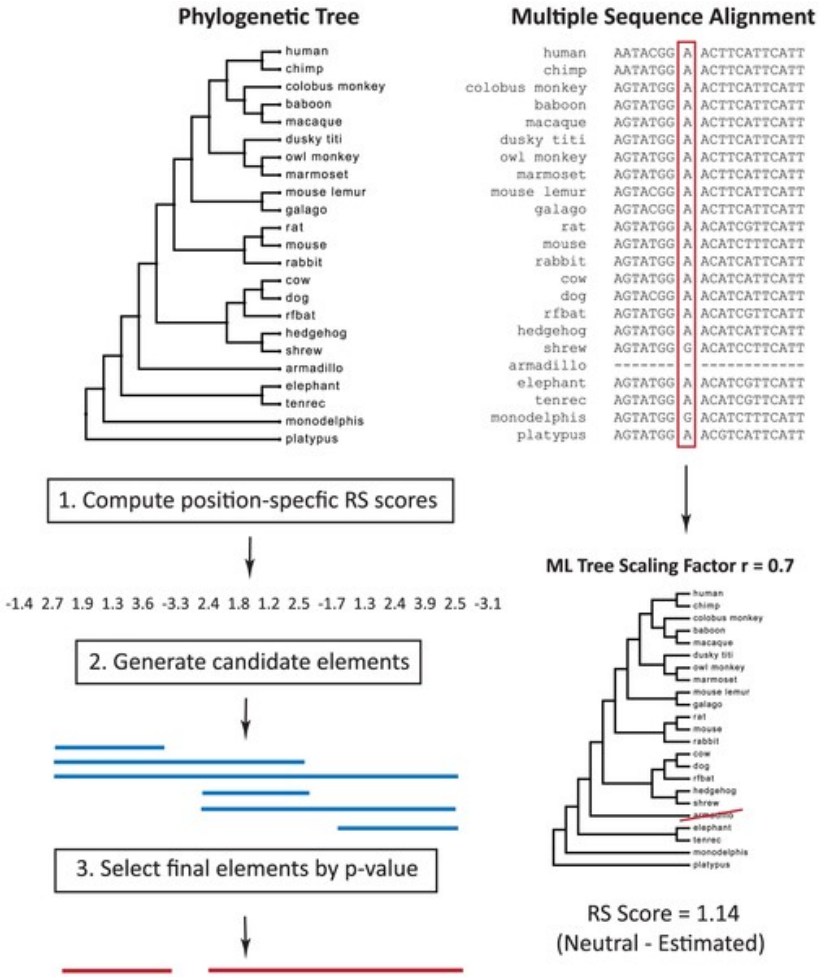


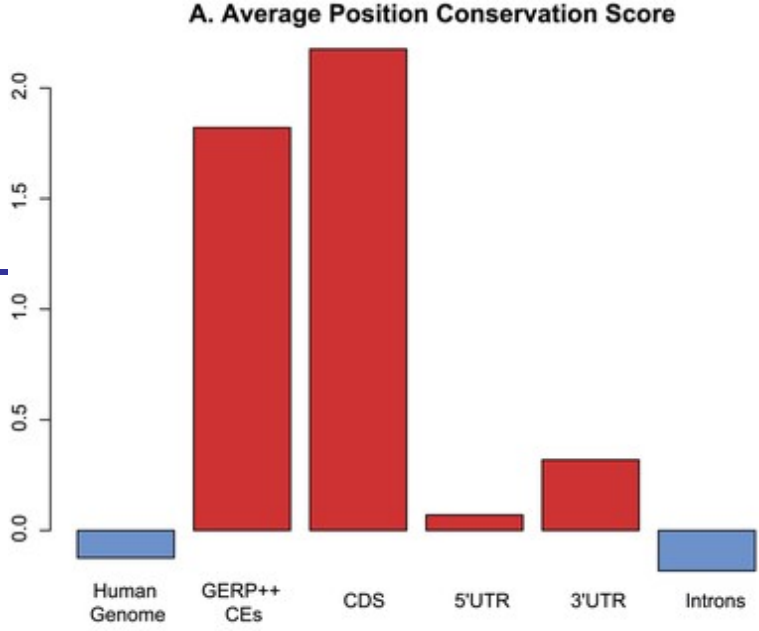


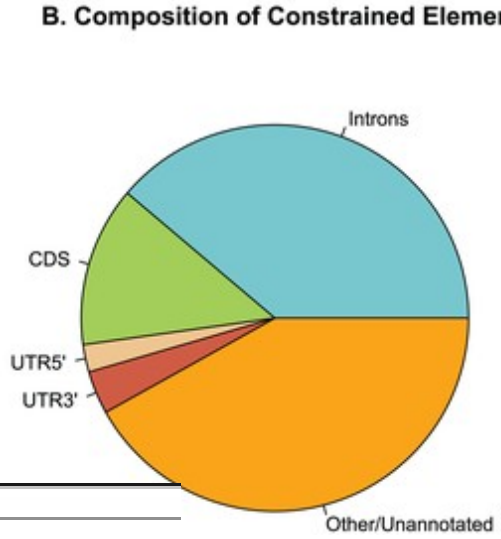




Genome Evolutionary Rate Profiling (GERP)



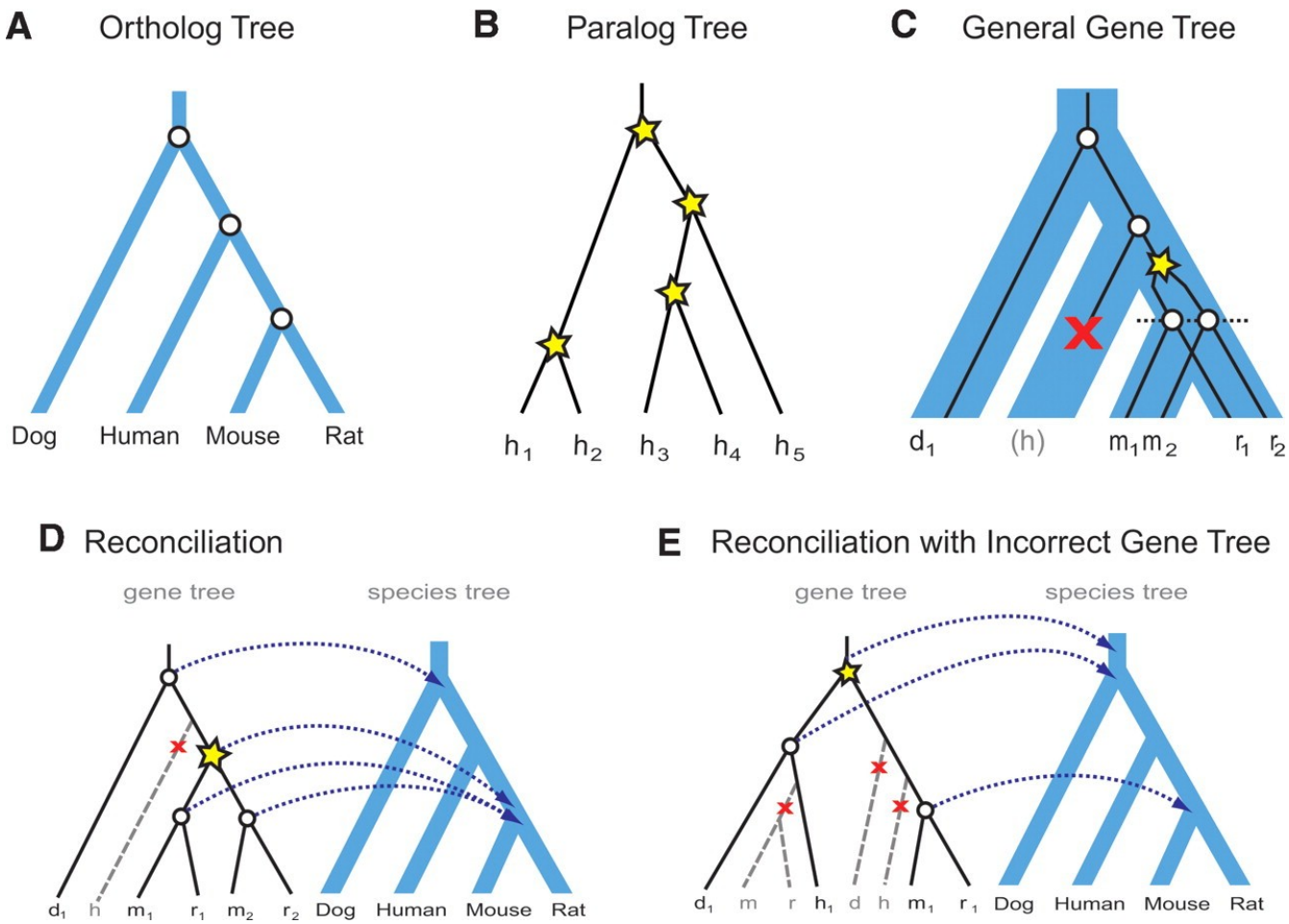




Annotation	% Coverage by CEs
Exons	84.6%
Introns	6.9%
UTR5'	23.7%
UTR3'	33.9%
ncRNA	10.1%

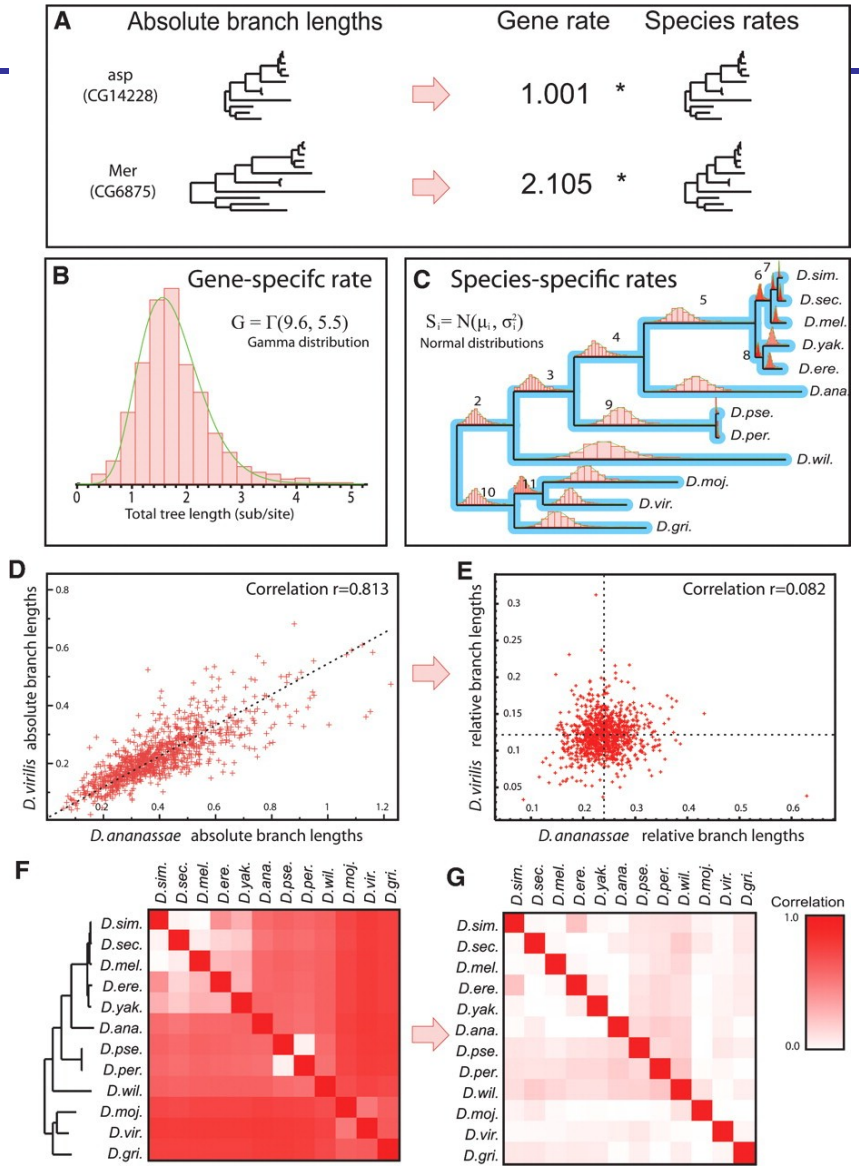
doi:10.1371/journal.pcbi.1001025.t001

Species Trees and Gene Trees



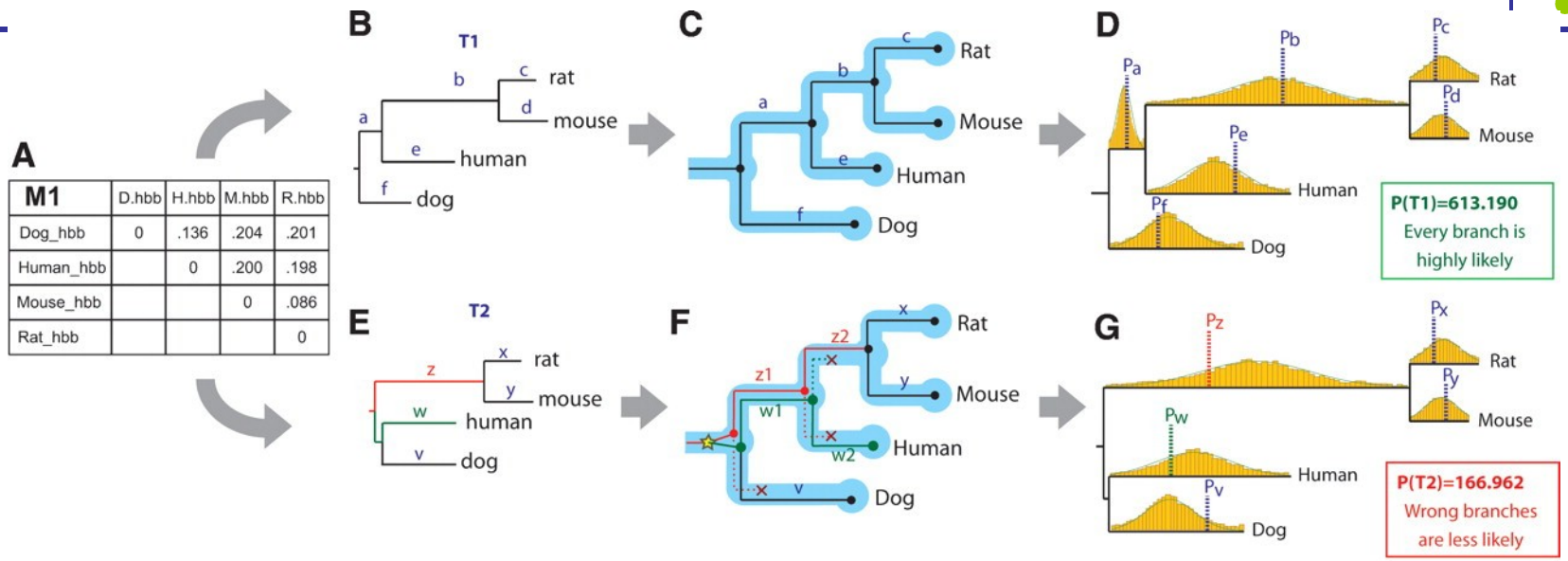
Rasmussen M D , Kellis M Genome Res. 2007;17:1932-1942

Evolutionary rates decoupled into gene-specific and species-specific components



Rasmussen M D , Kellis M Genome Res. 2007;17:1932-1942

Evaluating gene-tree likelihood using learned rate distributions.



M1	D.hbb	H.hbb	M.hbb	R.hbb
Dog_hbb	0	.136	.204	.201
Human_hbb		0	.200	.198
Mouse_hbb			0	.086
Rat_hbb				0

H

Method	Topology chosen	Ratio	Result	T1	T2
SPIDIR likelihood	T1	3.7	Correct	**613.190**	166.962
Max likelihood	T2	119.9	Wrong	-1689.5	**-1687.4**
Neighbor Joining	T2	8.0	Wrong	128	**828**
Parsimony	T2	1.2	Wrong	307	**300**

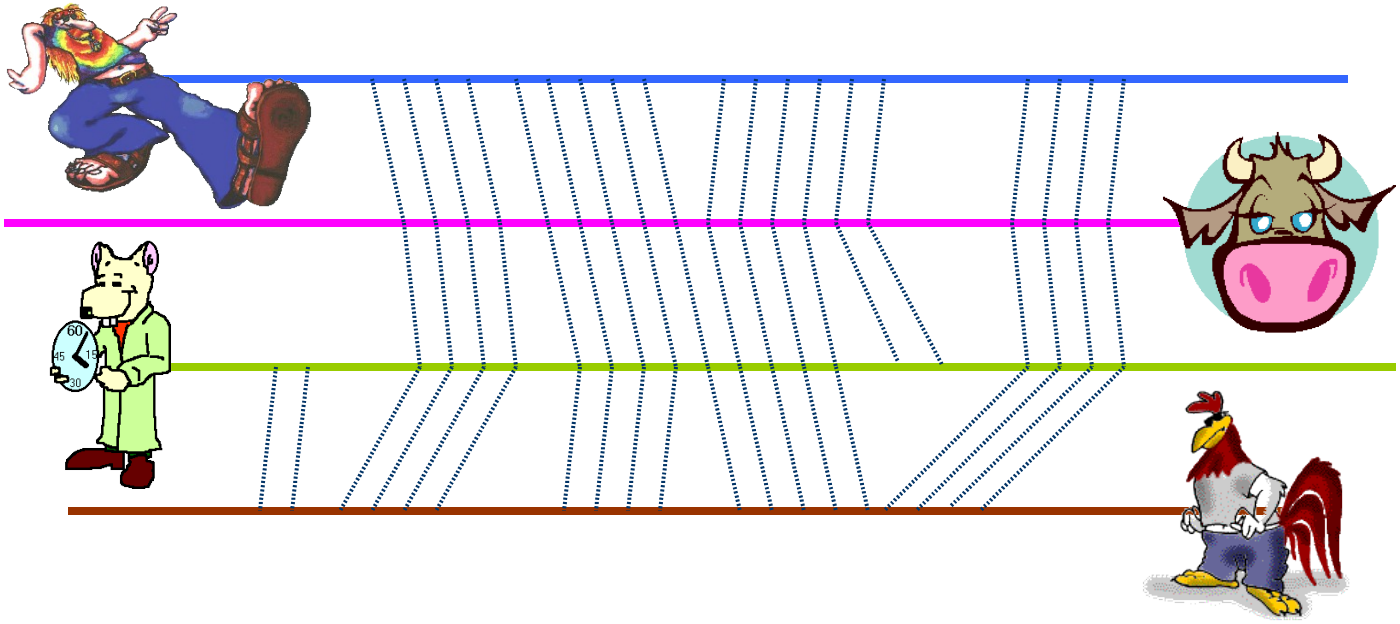
I

Species branch	Expected Length	Expected Stdev	Topology 1 Branch	Topology 1 Length	Topology 1 Likelihood	Topology 2 Branch	Topology 2 Length	Topology 2 Likelihood	Likelihood comparison
mouse	0.079	0.065	**d**	0.083	6.133	**y**	0.074	6.132	1.00
rat	0.084	0.068	**c**	0.060	5.510	**x**	0.053	5.299	1.04
human	0.172	0.083	**e**	0.241	3.403				1.04
hmr+human	0.279	0.099				**w**	0.215	3.261	
dog	0.107	0.054				**v**	0.179	3.025	1.22
dog+hmr	0.215	0.107	**a+f**	0.199	3.684				
mr	0.314	0.108	**b**	0.325	3.676				
hmr+mr	0.422	0.121				**z**	0.290	1.830	2.01
Family rate	alpha=1.31	b=4.95	rate=0.602		0.394	rate=0.675		0.285	1.38
Total					**613.190**			166.962	3.67

Rasmussen M D , Kellis M Genome Res. 2007;17:1932-1942

CSH PRESS

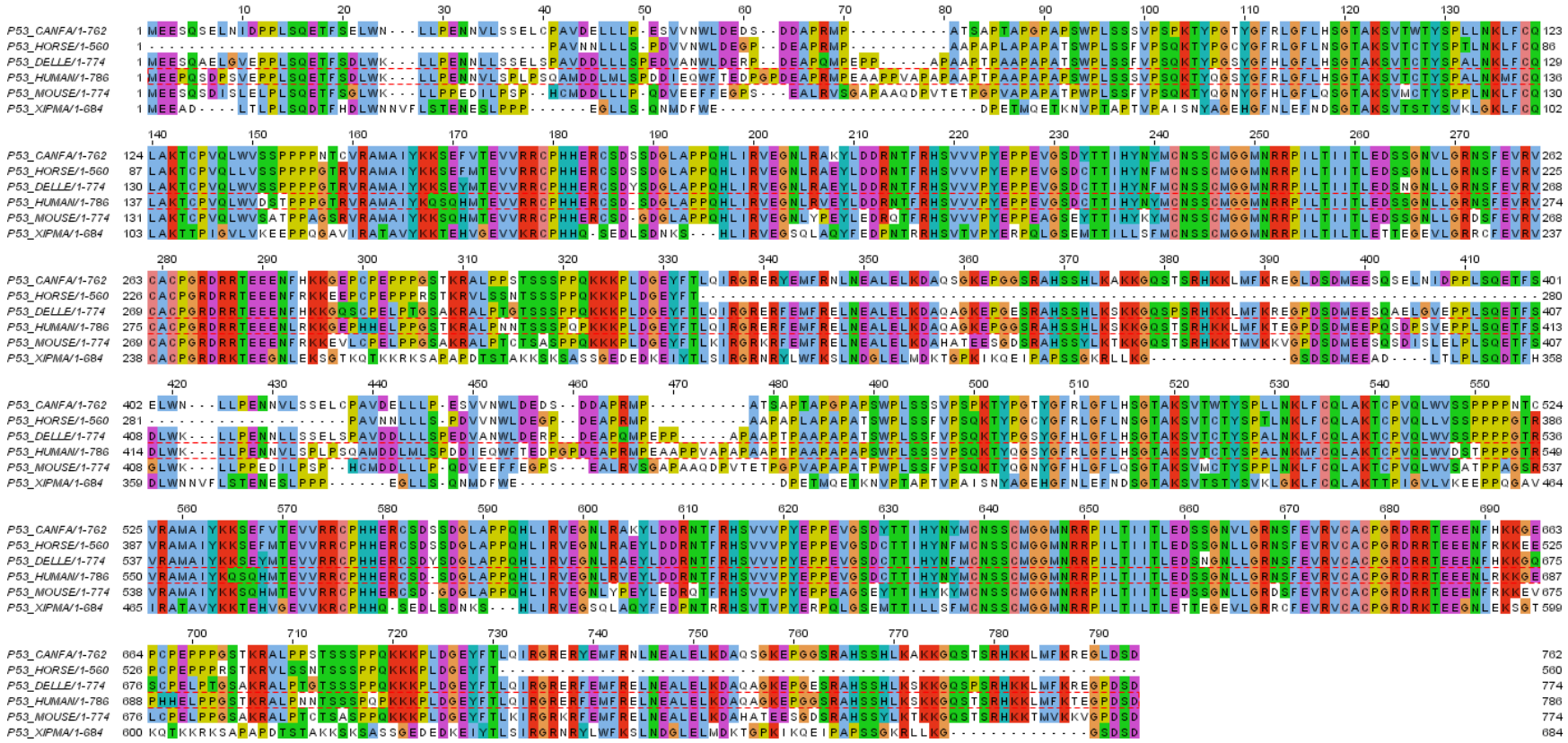
Multiple Sequence Alignments



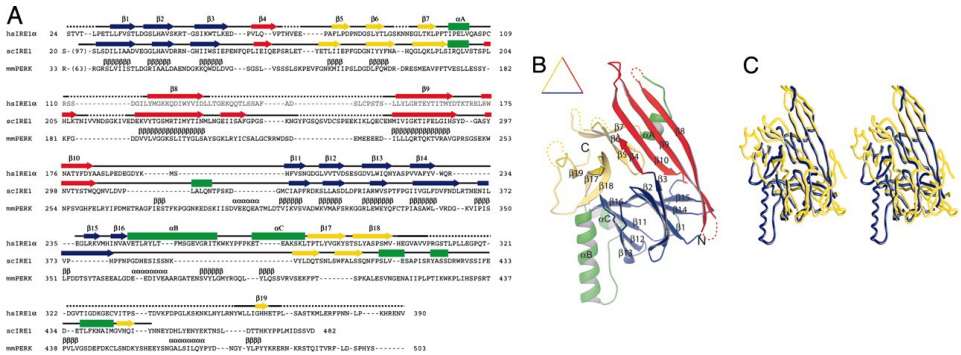
Definition

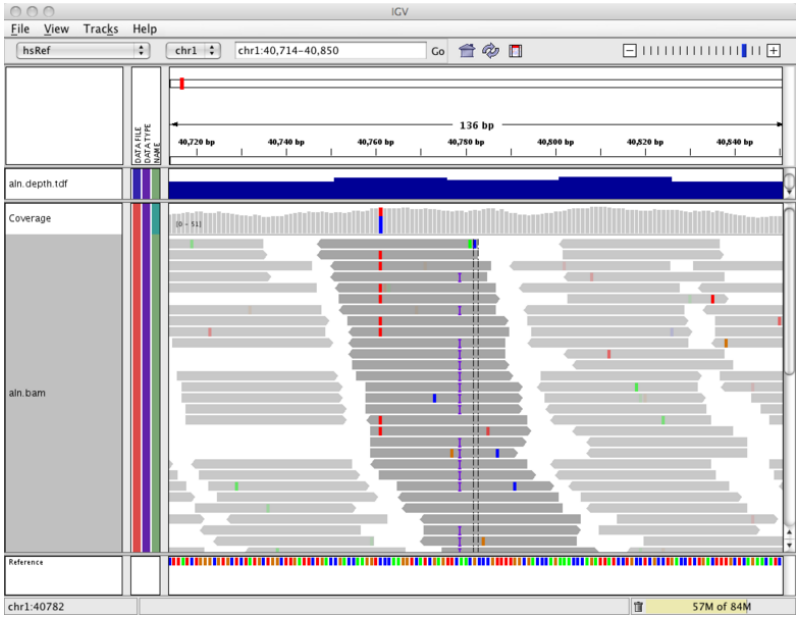
Given N sequences x^1, x^2,…, x^N:

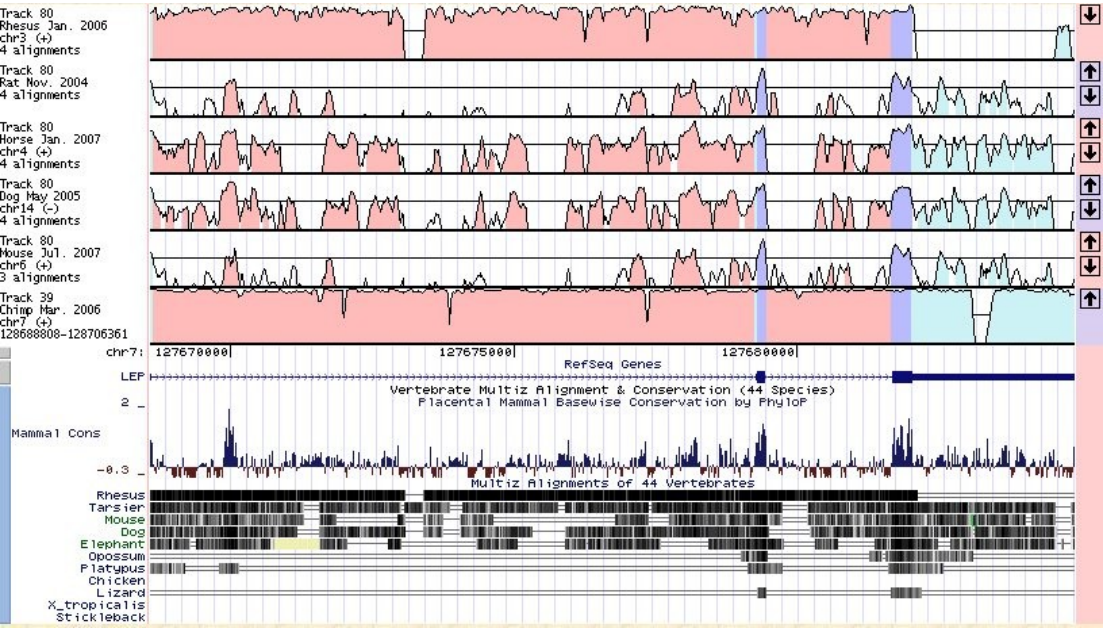
- Insert gaps (-) in each sequence x^i, such that
  - All sequences have the same length L
  - Score of the global map is maximum



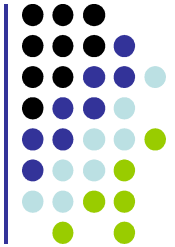
Applications







Scoring Function: Sum Of Pairs



Definition: Induced pairwise alignment

A pairwise alignment induced by the multiple alignment

Example:

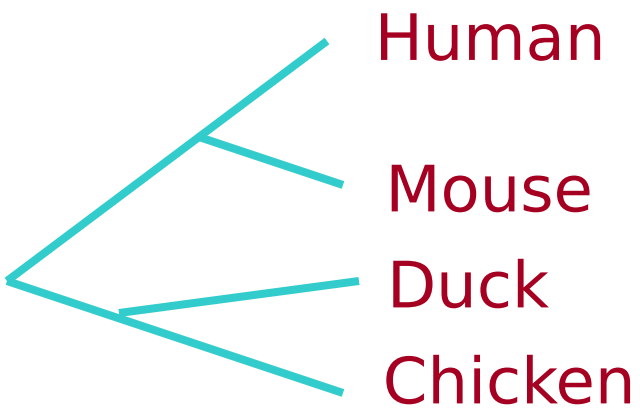
```
x:  AC-GCGG-C
y:  AC-GC-GAG
z:  GCCGC-GAG
```

Induces:

```
x: ACGCGG-C;   x: AC-GCGG-C;   y: AC-GCGAG
y: ACGC-GAC;   z: GCCGC-GAG;   z: GCCGCGAG
```

Sum Of Pairs (cont'd)

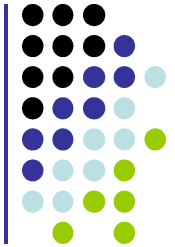
- Heuristic way to incorporate evolution tree:



- Weighted SOP:

$$S(m) = \Sigma_{k<l} w_{kl} s(m^k, m^l)$$

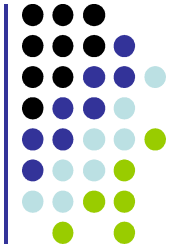
A Profile Representation



```
  -  A  G  G  C  T  A  T  C  A  C  C  T  G
  T  A  G  –  C  T  A  C  C  A  -  -  -  G
  C  A  G  –  C  T  A  C  C  A  -  -  -  G
  C  A  G  –  C  T  A  T  C  A  C  –  G  G
  C  A  G  –  C  T  A  T  C  G  C  –  G  G

A  0  1  0  0  0  0  1  0  0 .8  0  0  0  0
C .6  0  0  0  1  0  0 .4  1  0 .6 .2  0  0
G  0  0  1 .2  0  0  0  0  0 .2  0  0 .4  1
T .2  0  0  0  0  1  0 .6  0  0  0  0 .2  0
- .2  0  0 .8  0  0  0  0  0  0 .4 .8 .4  0
```

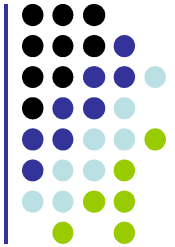
- Given a multiple alignment $M = m_1 \ldots m_n$
  - Replace each column m_i with profile entry p_i
    - Frequency of each letter in Σ
    - # gaps
    - Optional: # gap openings, extensions, closings
  - Can think of this as a “likelihood” of each letter in each position



Multiple Sequence Alignments

Algorithms

Multidimensional DP



Generalization of Needleman-Wunsh:

$$S(m) = \sum_i S(m_i)$$

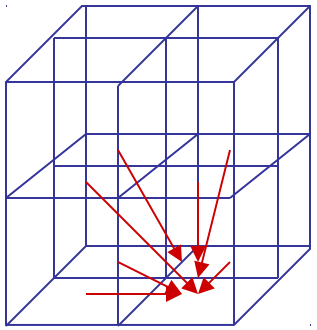
(sum of column scores)

$F(i_1,i_2,\ldots,i_N)$: Optimal alignment up to $(i_1, \ldots, i_N)$

$F(i_1,i_2,\ldots,i_N) = \max_{\text{(all neighbors of cube)}}(F(nbr)+S(nbr))$

Multidimensional DP

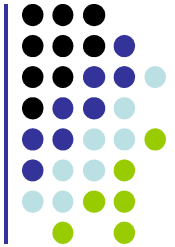
- Example: in 3D (three sequences):



- 7 neighbors/cell

$$
\begin{aligned}
F(i,j,k) = \max\{ \; & F(i-1, j-1, k-1) + S(x_i, x_j, x_k), \\
& F(i-1, j-1, k\;\;) + S(x_i, x_j, -), \\
& F(i-1, j\;\;\;, k-1) + S(x_i, -, x_k), \\
& F(i-1, j\;\;\;, k\;\;) + S(x_i, -, -), \\
& F(i\;\;, j-1, k-1) + S(-, x_j, x_k), \\
& F(i\;\;, j-1, k\;\;) + S(-, x_j, -), \\
& F(i\;\;, j\;\;\;, k-1) + S(-, -, x_k) \}
\end{aligned}
$$

Multidimensional DP



Running Time:

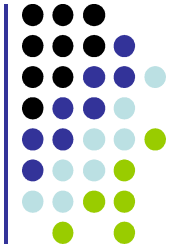
1. Size of matrix: L^N;

   Where L = length of each sequence
   N = number of sequences

2. Neighbors/cell: $2^N - 1$

Therefore…… $O(2^N L^N)$

Multidimensional DP



Running Time:

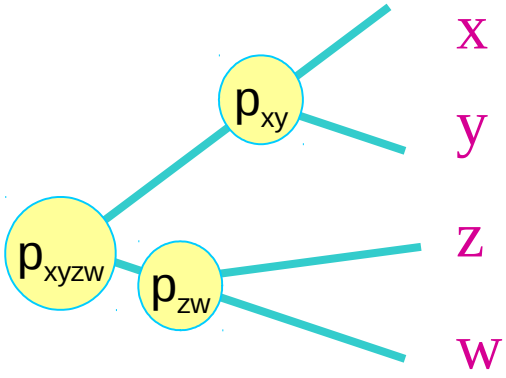
1. Size of matrix: L^N;

   Where L = length of each sequence
   N = number of sequences

2. Neighbors/cell: $2^N - 1$

Therefore…… $O(2^N L^N)$

Progressive Alignment

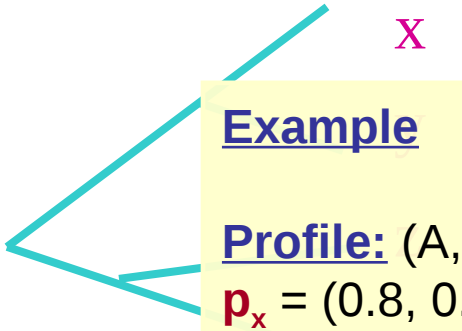


- When evolutionary tree is known:
  - Align closest first, in the order of the tree
  - In each step, align two sequences x, y, or profiles p_x, p_y, to generate a new alignment with associated profile p_{result}

Weighted version:

- Tree edges have weights, proportional to the divergence in that edge
- New profile is a weighted average of two old profiles

Progressive Alignment



Example

Profile: (A, C, G, T, -)
p_x = (0.8, 0.2, 0, 0, 0)
p_y = (0.6, 0, 0, 0, 0.4)

$s(p_x, p_y)$ = 0.8*0.6*s(A, A) + 0.2*0.6*s(C, A)
+ 0.8*0.4*s(A, -) + 0.2*0.4*s(C, -)

Result: p_{xy} = (0.7, 0.1, 0, 0, 0.2)

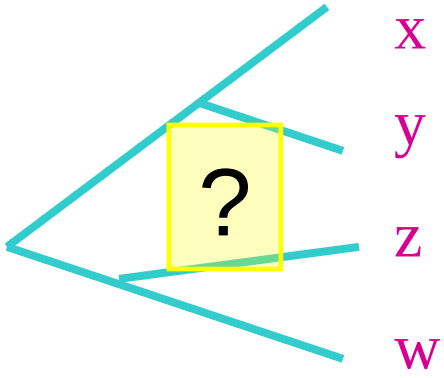
$s(p_x, -)$ = 0.8*1.0*s(A, -) + 0.2*1.0*s(C, -)

Result: p_{x-} = (0.4, 0.1, 0, 0, 0.5)

- When evolutionary tree is known:
  - Align closest first, in the order of the tree
  - In each step, align two sequence profiles to generate a new alignment with associated profile p_{result}

Weighted version:
- Tree edges have weights, proportional to evolutionary distance
- New profile is a weighted average of two old profiles

Progressive Alignment



- When evolutionary tree is unknown:
  - Perform all pairwise alignments
  - Define distance matrix D, where D(x, y) is a measure of evolutionary distance, based on pairwise alignment
  - Construct a tree *(UPGMA / Neighbor Joining / Other methods)*
  - Align on the tree

MUSCLE at a glance

1. Fast measurement of all
   - $D_{DRAFT}(x, y)$ defined in t
2. Build tree T_{DRAFT} based o
3. Progressive alignment o
4. Measure new Kimura-ba
5. Build tree T based on D
6. Progressive alignment o
7. Iterative refinement; for r
   - *Tree Partitioning:* Split
   - If new alignment M' has