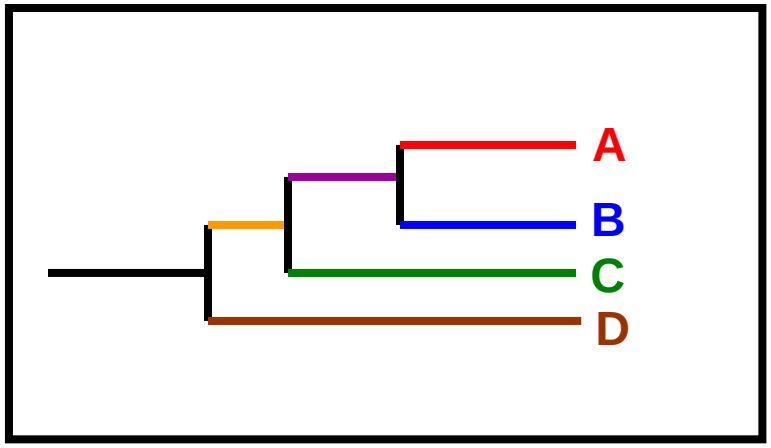
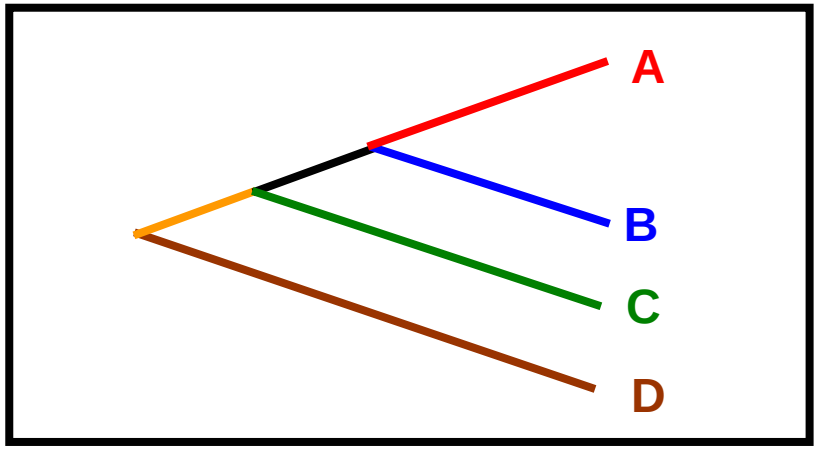
7.91 – Lecture #4 Michael Yaffe

Database Searching & Molecular Phylogenetics





(((A,B)C)D)

Outline

- FASTA, Blast searching, Smith-Waterman
- Psi-Blast
- Review of Genomic DNA structure
- Substitution patterns and mutation rates
- Synonymous and non-Synonymous substitutions
- Jukes-Cantor Model
- Kimura's Two-Parameter Model
- Molecular Clocks
- Phylogenetic Trees – rooted and unrooted
- Distance Matrix Methods
- Neighbor-Joining Method and Related Neighbor Methods
- Maximum Likelihood

Outline (cont)

- Parsimony
  - Branch and Bound
  - Heuristic Seaching
- Consensus Trees
- Software (PHYLIP, PAUP)
- The Tree of Life

Reading: Mount, p. 237-280, 283-286, 291-308

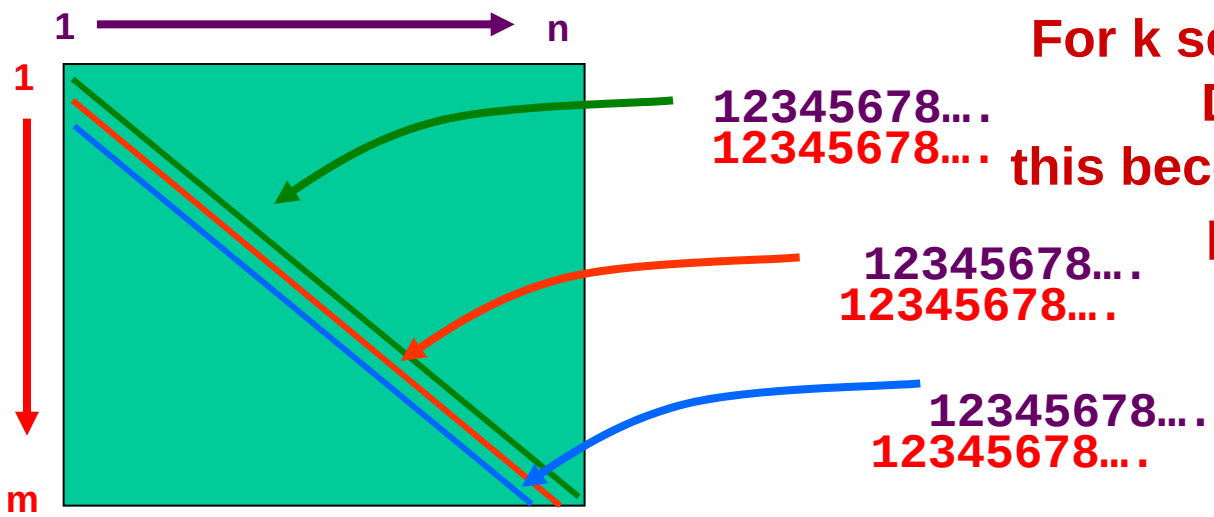
Database Searching

Problem is simple:
I want to find homologues to my protein in the database
How do I do it?

Do the obvious – compare my protein against every other protein in the database and look for local alignments by dynamic programming

Uh Oh!

For k sequences in the Database this becomes an O(mnk) problem!



….essentially an O(mn) problem

Database Searching

Still, this can be done - ~ 50x slower than Blast/FASTA, Smith-Waterman algorithm…
SSEARCH (ftp.virginia.edu/pub/fasta) – do it locally!

But in the old days, needed a faster method…
2 approaches – Blast, FASTA – both heuristic (i.e. tried and true) – almost always finds related Proteins but cannot guarantee optimal solution

FASTA: Basic Idea
1- Search for matching sequence patterns or words
Called k-tuples, which are exact matches of "k" characters between the two sequences
i.e. RW = 2-tuple
Seq 1: AHFYRWNKLCV Seq 2: DRWNLFCVATYWE

Database Searching

FASTA: Basic Idea

2- Repeat for all possible k-tuples

i.e. CV = 2-tuple

Seq 1: AHFYRWNKLCV **Seq 2: DRWNLFCVATYWE**

3- Make a Hash Table (Hashing) that has the position of each k-tuple in each sequence

i.e.

2-tuple	pos. in Seq1	pos in Seq 2	Offset (pos1-pos2)
RW	5	2	3
CV	10	7	3
AH	1	----	----

Database Searching

Seq 1: AHFYRWNKLCV Seq 2: DRWNLFCVATYWE

3- Make a Hash Table (Hashing) that has the position of each k-tuple in each sequence

i.e.

2-tuple	pos. in Seq1	pos in Seq 2	Offset (pos1-pos2)
RW	5	2	3
CV	10	7	3
AH	1	----	----

4- Look for words (k-tuples) with same offset

These are in-phase and reveal a region of alignment between the two sequences.

5- Build a local alignment based on these, extend it outwards

Seq 1: AHFYRWNKLCV
Seq 2: DRWNLFCVATYWE

Database Searching

With hashing, number of comparisons is proportional To the average sequence length (i.e. an O(n) problem), Not an O(mn) problem as in dynamic programming.

Proteins – ktup = 1-2,
Nucleotides, ktup=4-6

One big problem – low complexity regions.

Seq 1: AHFYPPPPPPPPFSER
Seq 2: DVATPPPPPPPPPPPNLFK

Database Searching

BLAST

Same basic idea as FASTA, but faster and more sensitive!

How?

BLAST searches for common words or k-tuples, but limits the search for k-tuples that are most significant, by using the log-odds values in the Blosum62 amino acid substitution matrix

i.e. look for WHK and might accept WHR but not HFK as a possible match (note 8000 possibilities)

Repeat for all 3-tuples in the query

Search the database for a match to the top 50 3-tuples that match the first query position in the sequence, the second query position, etc.

Use any match to seed an ungapped alignment (old BLAST)

Database Searching

Word length is fixed: **3-tuple for proteins**
11-tuple for nucleotides

By default, filters out low complexity regions.

Determine if the alignment is statistically significant. calculates the probability of observing a score greater than or equal to your alignment based on extreme value distribution.
Calculates an E-value = expectation value:

This is the probability of finding an unrelated sequence that shows this good an alignment just by chance.

Remember if p=.0001 and my database has 500,000 sequences, I will have an E=50! ***(normal starting E=10)***

NCBI

protein–protein BLAST

Nucleotide | Protein | Translations | Retrieve results for an RID

Search

Set subsequence From: To:

Choose database nr

Do CD-Search ☑

Now BLAST! or Reset query Reset all

Options for advanced blasting

Limit by entrez query or select from (none)

Composition-based statistics ☑

Choose filter ☑ Low complexity ☐ Mask for lookup table only ☐ Mask lower case

Expect 10

Word Size 3

Matrix PAM70 Gap Costs Existence: 7 Extension: 2

PSSM

Psi-BLAST

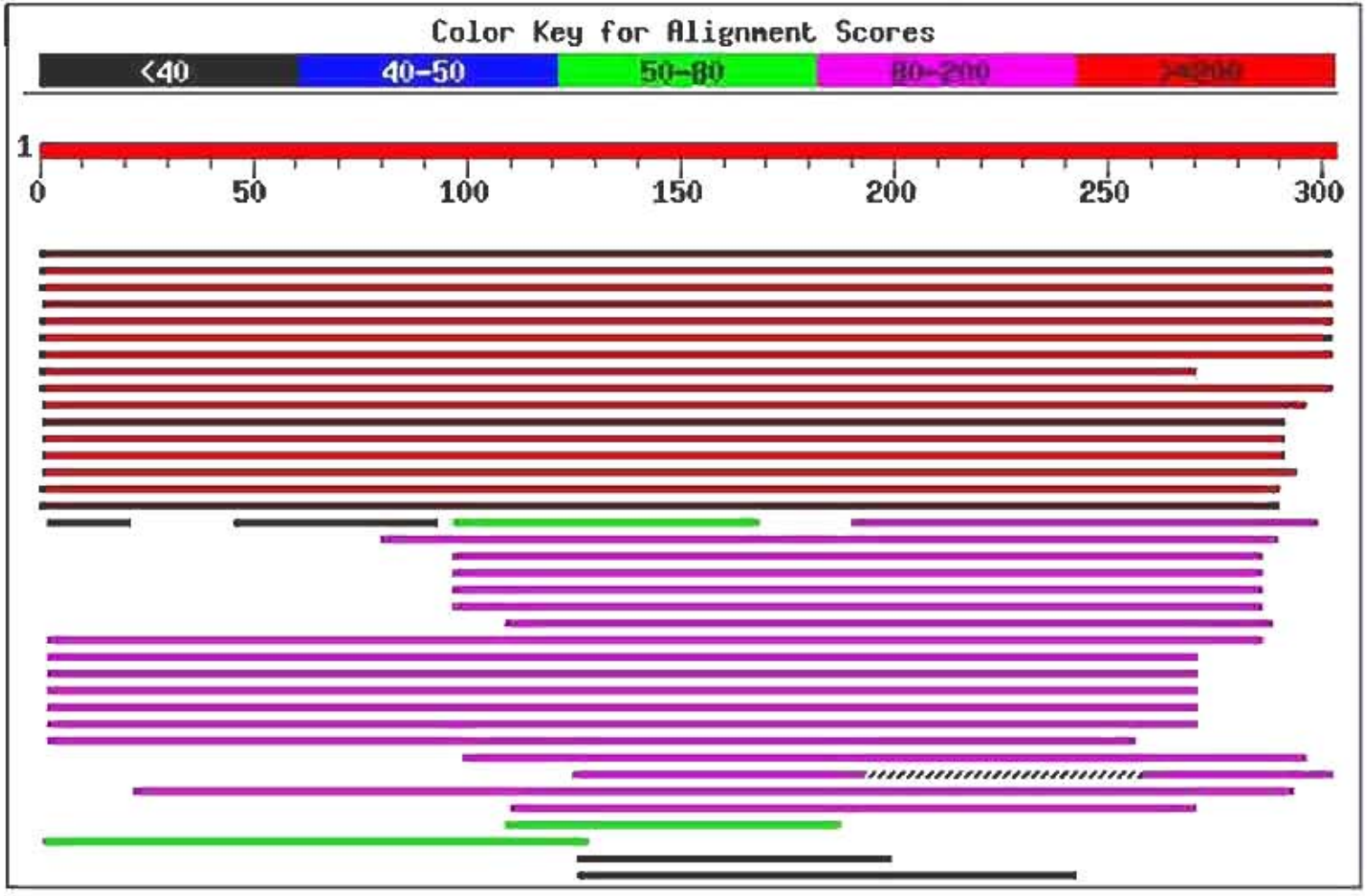
Position-specific iterative BLAST

Combines BLAST searching with PSSMs!

1- Start with regular BLAST search – look at the results

Distribution of 42 Blast Hits on the Query Sequence

Mouse-over to show defline and scores. Click to show alignments



Psi-BLAST

Position-specific iterative BLAST

Combines BLAST searching with PSSMs!

1- Start with regular BLAST search – look at the results

2- Pick the ones you believe are really homologous

Sequences with E-value BETTER than threshold

Sequences producing significant alignments:		Score (bits)	E Value
NEW ☑	gi|14785405|ref|XP_047240.1| (XM_047240) polo-like kinase (Droso...	623	e-178
NEW ☑	gi|4826916|ref|NP_005021.1| (NM_005030) polo-like kinase (Drosop...	622	e-177
NEW ☑	gi|393017|gb|AAA56634.1| (U01038) pLK [Homo sapiens]	621	e-177
NEW ☑	gi|403458|gb|AAA36659.1| (L19559) protein kinase [Homo sapiens]	619	e-176
NEW ☑	gi|6755104|ref|NP_035251.1| (NM_011121) polo-like kinase homolog...	600	e-171
NEW ☑	gi|1083470|pir||A47545 protein kinase (EC 2.7.1.37) Plk - mouse ...	597	e-170
NEW ☑	gi|12230396|sp|O62673|PLK1_RAT Serine/threonine-protein kinase P...	597	e-170
NEW ☑	gi|13507375|gb|AAK28550.1|AF339021_1 (AF339021) polo-like protei...	544	e-154
NEW ☑	gi|1537064|gb|AAC60017.1| (U58205) Plx1 [Xenopus laevis]	503	e-142
NEW ☑	gi|11463874|dbj|BAB18588.1| (AB043897) polo-like kinase [Hemicen...	362	3e-99
NEW ☑	gi|465783|sp|P34331|YKZ4_CAEEL Hypothetical 41.8 kDa protein C14...	293	2e-78
NEW ☑	gi|17554392|ref|NP_498770.1| (NM_066369) Protein kinase [Caenorh...	293	2e-78
NEW ☑	gi|3063645|gb|AAC14129.1| (AF057165) putative serine/threonine p...	293	2e-78
NEW ☑	gi|17510519|ref|NP_491036.1| (NM_058635) Y71F9B.7.p [Caenorhabdi...	283	1e-75
NEW ☑	gi|17737679|ref|NP_524179.1| (NM_079455) polo [Drosophila melano...	254	6e-67
NEW ☑	gi|14286167|sp|P52304|POLO_DROME PROTEIN KINASE POLO >gi|7293666...	254	9e-67
NEW ☑	gi|3366792|gb|AAC28624.1| (AF053092) polo-like kinase isoform [R...	152	3e-36
NEW ☑	gi|17541716|ref|NP_501196.1| (NM_068795) protein kinase [Caenorh...	150	2e-35
NEW ☑	gi|5730055|ref|NP_006613.1| (NM_006622) serum-inducible kinase [...	141	1e-32
NEW ☑	gi|14730424|ref|XP_041712.1| (XM_041712) serum-inducible kinase ...	140	1e-32
NEW ☑	gi|1711416|sp|P53351|SNK_MOUSE Serine/threonine-protein kinase S...	139	3e-32
NEW ☑	gi|13929172|ref|NP_114009.1| (NM_031821) serum-inducible kinase ...	139	5e-32
NEW ☑	gi|16902371|gb|AAL30177.1|AF357842_1 (AF357842) polo-like kinase...	138	6e-32
NEW ☑	gi|16902367|gb|AAL30175.1|AF357840_1 (AF357840) polo-like kinase...	135	4e-31
NEW ☑	gi|833810|gb|AAC52191.1| (U21392) putative serine/threonine kina...	135	7e-31
NEW ☑	gi|13878440|sp|Q60806|CNK_MOUSE CYTOKINE-INDUCIBLE SERINE/THREON...	134	7e-31
NEW ☑	gi|4758016|ref|NP_004064.1| (NM_004073) cytokine-inducible kinas...	131	8e-30

Psi-BLAST

Position-specific iterative BLAST

Combines BLAST searching with PSSMs!

1- Start with regular BLAST search – look at the results

2- Pick the ones you believe are really homologous

3- Now align these sequences to the query sequence and make up a PSSM that tells how much to weigh each amino acid in each position in the alignment

4- Use this PSSM to do another BLAST search

5- Add any new sequences that come up to the old ones if you believe they are really homologous

6- Repeat the alignment to make a new and improved PSSM that tells how much to weigh each amino acid in each position in the alignment

gi|13507375|gb|AAK28550.1|AF339021_1	(AF339021) polo-like protei...	461	e-129
gi|11463874|dbj|BAB18588.1|	(AB043897) polo-like kinase [Hemicen...	413	e-114
gi|17510519|ref|NP_491036.1|	(NM_058635) Y71F9B.7.p [Caenorhabdi...	392	e-108
gi|3063645|gb|AAC14129.1|	(AF057165) putative serine/threonine p...	391	e-108
gi|17554392|ref|NP_498770.1|	(NM_066369) Protein kinase [Caenorh...	390	e-108
gi|465783|sp|P34331|YKZ4_CAEEL	Hypothetical 41.8 kDa protein C14...	387	e-107
gi|17737679|ref|NP_524179.1|	(NM_079455) polo [Drosophila melano...	371	e-102
gi|14286167|sp|P52304|POLO_DROME	PROTEIN KINASE POLO >gi|7293666...	370	e-101
gi|16902367|gb|AAL30175.1|AF357840_1	(AF357840) polo-like kinase...	324	5e-88
gi|833810|gb|AAC52191.1|	(U21392) putative serine/threonine kina...	322	2e-87
gi|13878440|sp|O60806|CNK_MOUSE	CYTOKINE-INDUCIBLE SERINE/THREON...	322	3e-87
gi|4758016|ref|NP_004064.1|	(NM_004073) cytokine-inducible kinas...	314	5e-85
gi|15530236|gb|AAH13899.1|AAH13899	(BC013899) Unknown (protein f...	312	2e-84
NEW gi|13878438|sp|O9R011|CNK_RAT	Cytokine-inducible serine/threonin...	311	7e-84
gi|5730055|ref|NP_006613.1|	(NM_006622) serum-inducible kinase [...	304	6e-82
gi|14730424|ref|XP_041712.1|	(XM_041712) serum-inducible kinase ...	304	7e-82
gi|13878441|sp|O9H4B4|CNK_HUMAN	CYTOKINE-INDUCIBLE SERINE/THREON...	300	8e-81
gi|13929172|ref|NP_114009.1|	(NM_031821) serum-inducible kinase ...	298	4e-80
gi|1711416|sp|P53351|SNK_MOUSE	Serine/threonine-protein kinase S...	298	4e-80
gi|16902371|gb|AAL30177.1|AF357842_1	(AF357842) polo-like kinase...	292	4e-78
gi|13448668|gb|AAK27155.1|AF348425_1	(AF348425) FGF-inducible ki...	287	7e-77
gi|2644989|emb|CAA74301.1|	(Y13968) polo-like protein kinase [Tr...	264	6e-70
gi|17541716|ref|NP_501196.1|	(NM_068795) protein kinase [Caenorh...	258	4e-68
gi|1709661|sp|P50528|PLO1_SCHPO	Serine/threonine-protein kinase ...	230	1e-59
gi|6323643|ref|NP_013714.1|	(NC_001145) CDC5 is dispensable for ...	193	2e-48
gi|3366792|gb|AAC28624.1|	(AF053092) polo-like kinase isoform [R...	159	4e-38
gi|16902369|gb|AAL30176.1|AF357841_1	(AF357841) polo-like kinase...	156	2e-37
gi|13448666|gb|AAK27154.1|AF348424_1	(AF348424) serum-inducible ...	146	2e-34
gi|18591448|ref|XP_059051.2|	(XM_059051) similar to cytokine-ind...	145	4e-34
gi|18569167|ref|XP_095399.1|	(XM_095399) hypothetical protein XP...	96	5e-19
NEW gi|4099301|gb|AAD00575.1|	(U85755) serum-inducible kinase [Homo ...	48	9e-05

Run PSI-Blast iteration 3

Psi-BLAST

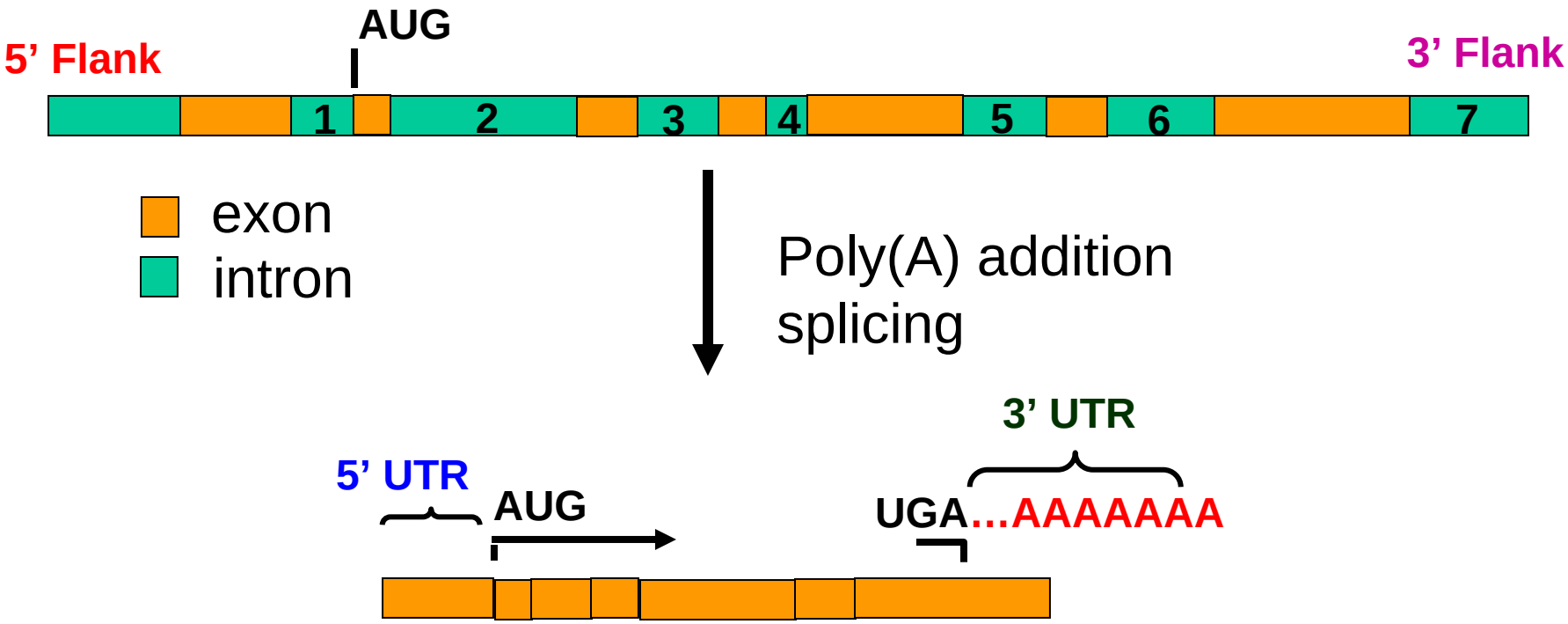
7-- Use this PSSM to do another BLAST search

8– Keep iterating until no new sequences are found

Very good for finding weakly related sequences

…on to Molecular Phylogenetics

Gene Structure



Mutation Rates

Mutations:
- deleterious
- neutral
- advantageous ←***substantial minority***

Consider 2 sequences

K= # of substitutions since they shared a common ancestor

T= divergence time

R = mutation rate = K/(2T)

KEY PREMISE OF PHYLOGENETICS:

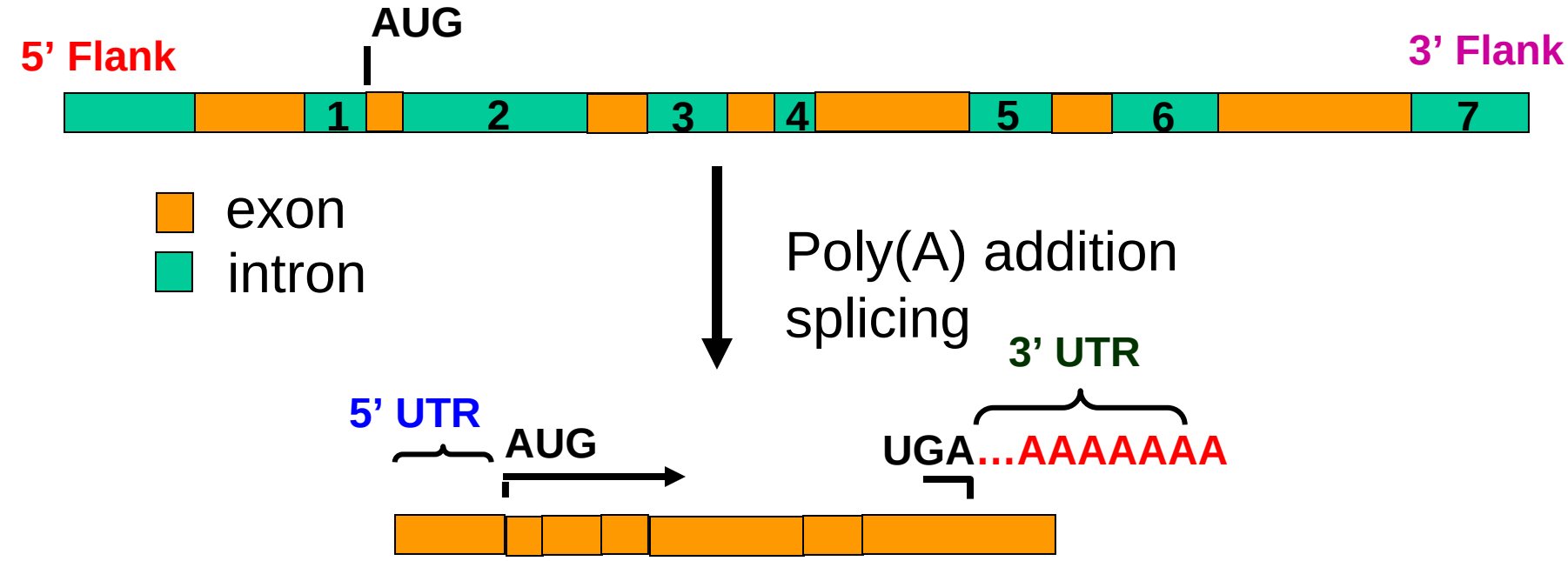
If R=constant for all species, then K will provide insight into evolutionary relatedness for which no other physical evidence is available.

Mutation Rates

Mutations: refined by process of natural selection…

Often, but not always at the protein level…

$\rightarrow$ Functional constraint



Human, mouse, rabbit and cow beta-globin

	Length, bp	# Pairwise Δ's (mean)		Substitution rate (subs/site x 10^9 years)	
Noncoding, overall	913	268		3.33	
Coding, overall	441	69		1.58	functionally constrained
5' Flank	300	96	rapid changes	3.39	
5' UTR	50	9		1.86	
Intron 1	131	42		3.48	
3' UTR	132	33	…generally true	3.00	
3' Flank	300	76		3.04	

Common ancestor 100 million years ago

Synonymous vs Nonsynonymous Substitutions

18 out of 20 amino acids have more than one codon

GGG, GGC, GGU, GGA → Glycine

Δ's here – same aa →synonymous substitution

GGG
GGC
GGU
GGA

nonsynonymous substitution, GCG →Alanine

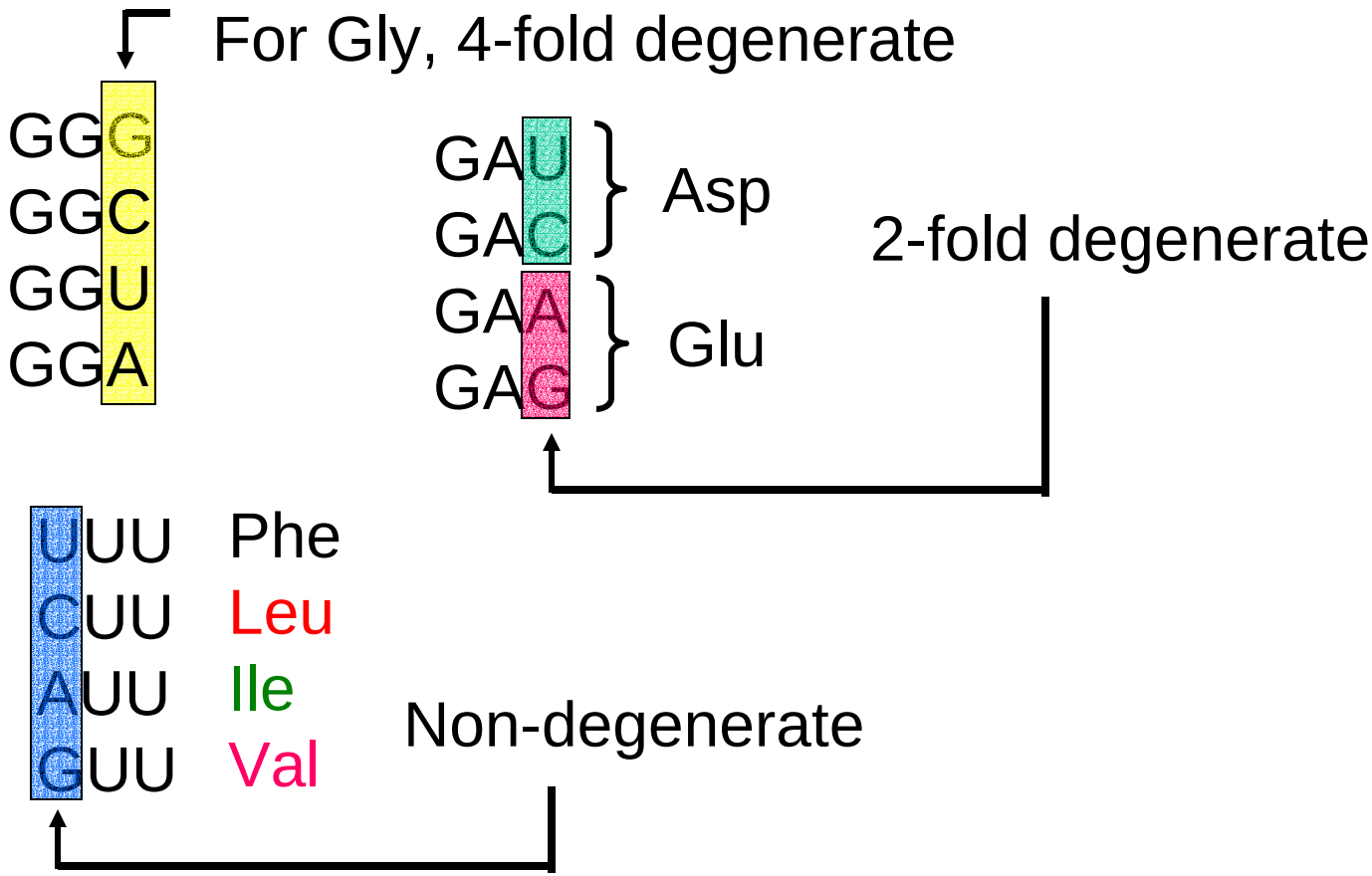
Human and rabbit beta-globin genes:
47 substitutions in coding sequence

- 27 synonymous substitutions
- 20 nonsynonymous substitutions

but 3x as many opportunities!

Synonymous vs Nonsynonymous Substitutions

Not all positions in a codon equally likely to give non-synonymous substitutions



Synonymous vs Nonsynonymous Substitutions

If natural selection operates at protein level, expect nucleotide substitutions appear most rapidly at 4-fold sites and least rapidly at non-degenerate sites

What does data show?

Human vs rabbit beta-globin genes (coding region)

Region	# sites (bp)	#changes	Sub. Rate **Subs/site.10^9 years**
Non-deg.	**302**	**17**	**.56**
2-fold deg.	**60**	**10**	**1.67**
4-fold deg.	**85**	**20**	**2.35**

Mutation versus Substitutions

Mutation: changes in nucleotide sequences due to errors in DNA replication or repair

Substitution: mutations that pass through the filter of natural selection

Synonymous substitution rates, **K**s, reflect actual mutation rate

Non-synonymous substitution rates, **K**a, do NOT reflect actual mutation rate, as subject to natural selection

New alleles (versions of a gene) typically begin at low frequencies $q = 1/(2N)$ where N=# of diploid reproducing organisms

Why are there persistant high levels of variation in populations? Why not $q \rightarrow 0$, $q \rightarrow 1$?

Most mutations are selectively neutral!

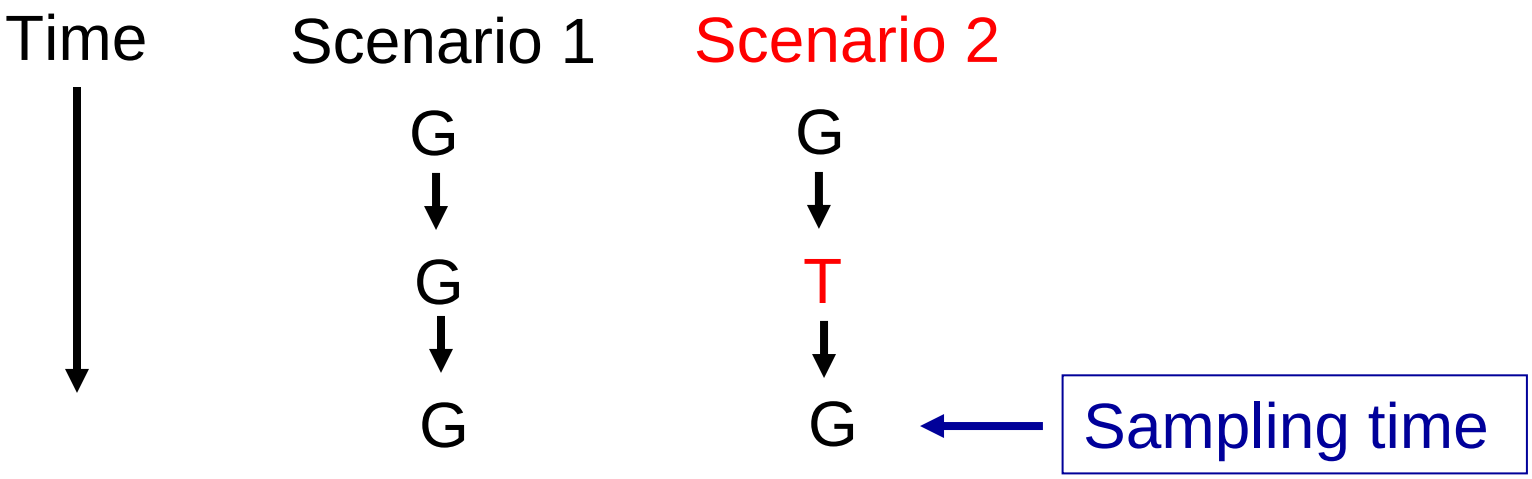
Estimating Substitution Numbers

Infrequent substitutions between 2 sequences:

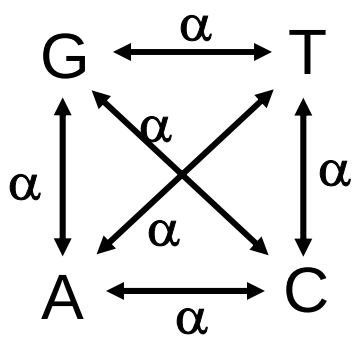
Count ‘em…gives **K**

More frequent substitutions – counting will significantly UNDERestimate the number of true substitutions since they shared a common ancestor

Why?



Jukes-Cantor Model



Assume each nucleotide equally likely
to change into any other nt,
with rate of change=α.
Overall rate of substitution = 3α
…so if G at t=0, at t=1, $P_{G(1)}=1-3\alpha$

and $P_{G(2)}=(1-3\alpha)P_{G(1)} +\alpha\,[1- P_{G(1)}]$

Expanding this gives $P_{G(t)}=1/4 + (3/4)e^{-4\alpha t}$

Can show that this gives $K = -3/4 \ln[1-(4/3)(p)]$

K = true number of substitutions that have occurred,
P = fraction of nt that differ by a simple count.

Captures general behaviour…

Compare J-C with real data…assumption of global uniformity in α was unrealistic…still provides a useful framework…

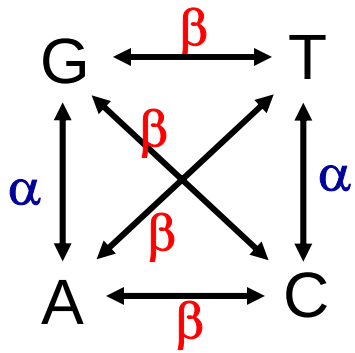
Nucleotides: two categories: purines: A, G
pyrimidines: C, T, U

Exchange nucleotides within or between classes happens at different rates!

Transitions: purine$\rightarrow$purine, pyrmidine $\rightarrow$pyrimidine
three times as commmon as
Transversions: purine$\rightarrow$pyrimidine or pyrimidine$\rightarrow$purine

Led to Kimura's Two Parameter Model

Kimura's Two Parameter Model



Transitions occur at rate α

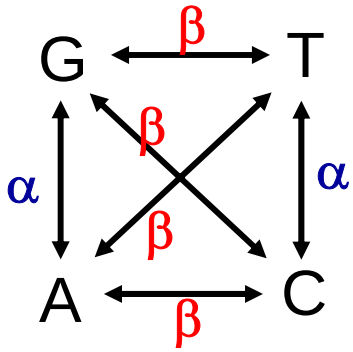
Transversions occur at rate β

Now $P_{GG(1)} = 1 - \alpha - 2\beta$

$P_{GG(2)}$: 4 possibilities:

Time	Scenario 1	Scenario 2	Scenario 3	Scenario 4
↓	G	G	G	G
	↓	↓	↓	↓
	G	A	C	T
	↓	↓	↓	↓
	G	G	G	G
	No Δ	1 Transition	2 Transversions	

Kimura's Two Parameter Model



Transitions occur at rate α
Transversions occur at rate β

$$P_{GG(1)} = 1 - \alpha - 2\beta$$

$P_{GG(2)}$: 4 possibilities:

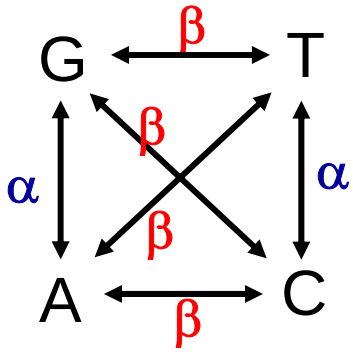
Time	Scenario 1	Scenario 2	Scenario 3	Scenario 4
↓	G	G	G	G
	↓	↓	↓	↓
	G	A	C	T
	↓	↓	↓	↓
	G	G	G	G
	No Δ	1 Transition	2 Transversions	

$$P_{GG(2)} = (1 - \alpha - 2\beta)\, P_{GG(1)} + \alpha P_{GA(1)} + \beta P_{GC(1)} + \beta P_{GT(1)}$$

expanding…

$$P_{GG(t)} = \tfrac{1}{4} + (1/4)e^{-4\beta t} + (1/2)e^{-2(\alpha+\beta)t}$$

Kimura's Two Parameter Model



Transitions occur at rate α
Transversions occur at rate β

$$P_{GG(2)} = (1 - \alpha - 2\beta)\, P_{GG(1)} + \alpha P_{GA(1)} + \beta P_{GG(1)} + \beta P_{GG(1)}$$

expanding…

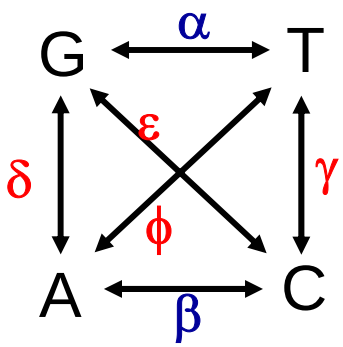
$$P_{GG(t)} = 1/4 + (1/4)e^{-4\beta t} + (1/2)e^{-2(\alpha+\beta)t}$$

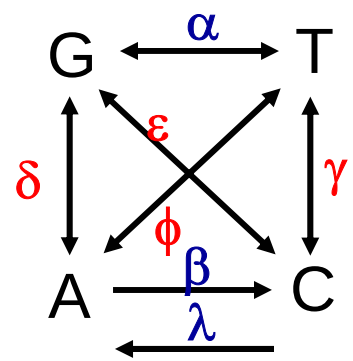
Manipulating equation gives estimate of true number of substitutions if only two sequences are available,

$$K = 1/2 \ln[1/(1-2P-Q)] + 1/4 \ln[1/(1-2Q)]$$

Where K = true number of substitutions
P = fraction of nts undergoing transitions by simple count
Q = fraction of nts undergoing tranversions by simple count

More complex Parameter Models Possible





Could even make $A \rightarrow C \neq C \rightarrow A$

Problem is sampling error – not enough data to get Good parameters within a single gene family, usually

Why not combine different genes?

Find strikingly different rates of evolution between different Genes – up to and greater than 200-fold.

RATE DEPENDS ON FUNCTION!

Histones – each aa interacts with DNA – slowest rate of substitution known

HLA gene locus – involved in immune system recognition of foreign antigens – needs to adapt rapidly - one of the Highest substitution rates known

However, rates of molecular evolution for loci with similar functional constraints often very uniform over long periods of evolutionary time.

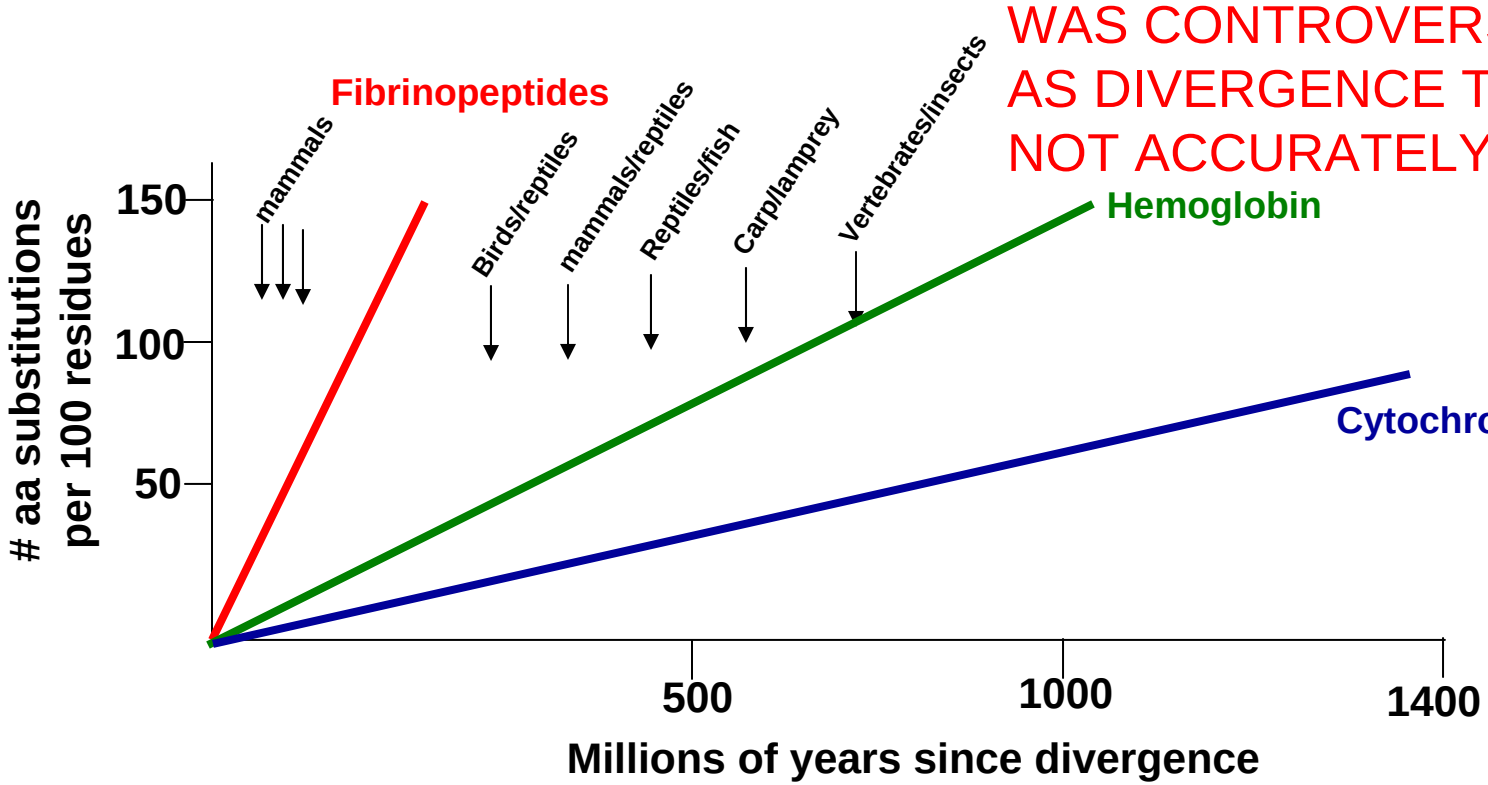
Molecular Clocks

1960s: Emile Zuckerkandl and Linus Pauling

Postulate: Substitution rates so constant within homologous proteins over long periods of evolutionary time that accumulation of amino acid changes reflects the steady ticking of a molecular clock.

Clock may run at different rates for different proteins

WAS CONTROVERSIAL AS DIVERGENCE TIMES NOT ACCURATELY KNOWN



Relative Rate Test of Molecular Clock Hypothesis

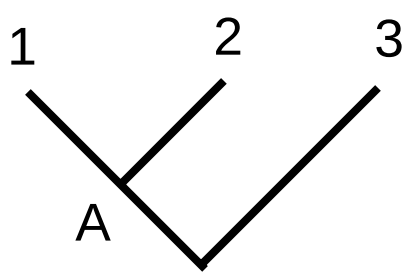
1973: Sarich and Wilson

Consider relative rate of substitution in lineage for species 1 and 2

Need to designate a less related species 3 as an outgroup

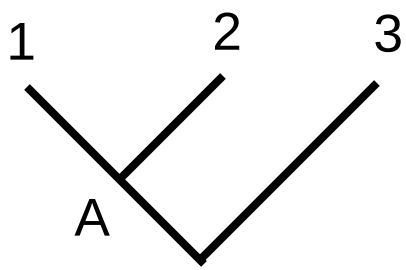
i.e. 1=humans, 2=gorillas, 3= baboons

Phylogenetic tree (more soon!)



1 and 2 diverged from a common ancestor, A

Number of substitutions between any two species = sum of number of substitutions along branches of the tree that connect them



1 and 2 diverged from a common ancestor, A

Number of substitutions between Any two species = sum of number of Substitutions along branches of the tree That connect them

d_{13}, d_{23}, d_{12} – can measure directly

$d_{13} = d_{A1} + d_{A3}$
$d_{23} = d_{A2} + d_{A3}$
$d_{12} = d_{A1} + d_{A2}$

Algebra:
$d_{A1} = (d_{12} + d_{13} - d_{23})/2$
$d_{A2} = (d_{12} + d_{23} - d_{13})/2$
Theorum 1
Molecular clock predicts $d_{A1} = d_{A2}$

Find, for the most part, this is true, but not always, depending on species…

So bottom line when comparing two species need to prove Theorum 1 before using the molecular clock!

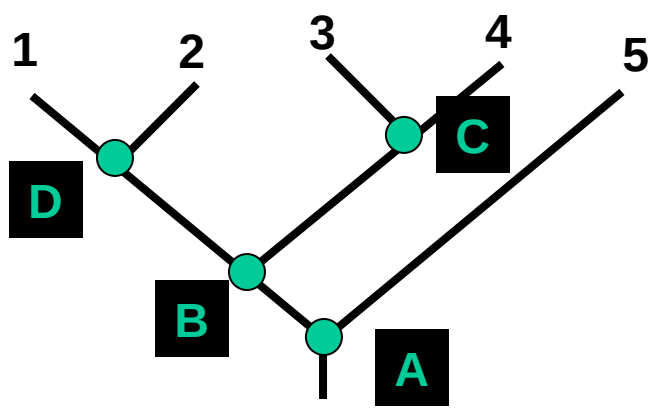
Distance-Based Phylogenetics

Phylogenetic Trees – also called dendrograms

- Made by arranging nodes and branches.
- Graphical representation of evolutionary relatedness of 3 or more sequences

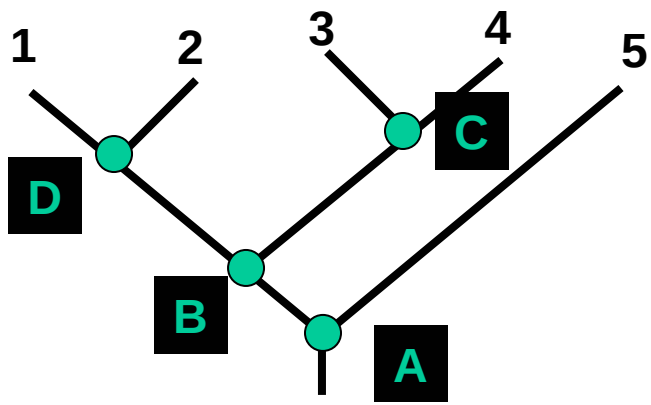
Nodes – distinct taxonomical unit

- Terminal nodes: gene or organism for which data has been collected
- Internal node – inferred common ancestor that gave rise to 2 lineages



For the mathematicians:
Tree - special graph with
n nodes, n-1 links, no circuits

Distance-Based Phylogenetics



Newick notation

(((1,2), (3,4)),5)

Scaled trees

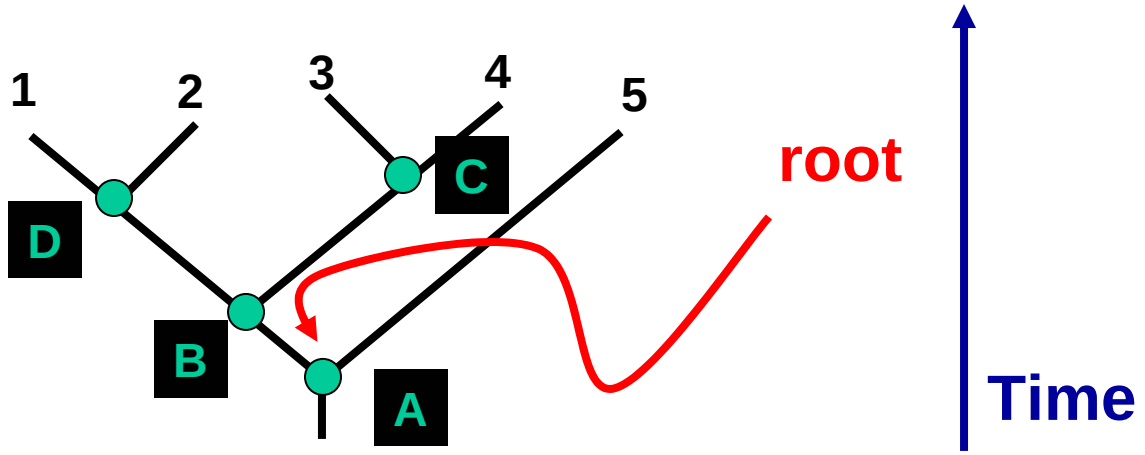
Branch length is α difference between pairs of neighboring nodes.

Ideally, scaled trees should be additive

Unscaled trees

Only convey relative kinship information without representing number of changes that separate sequences

Distance-Based Phylogenetics

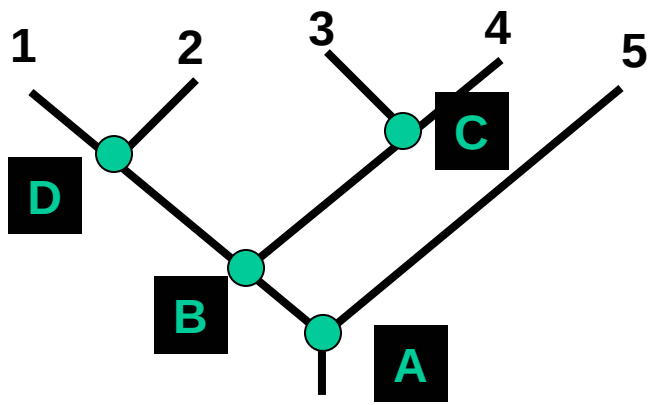


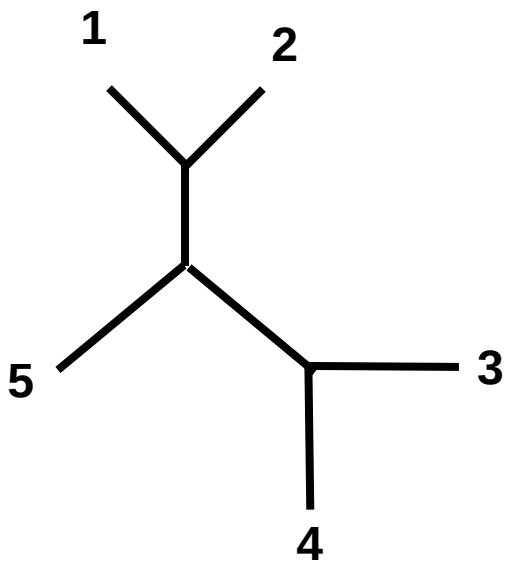
Rooted trees

Make an inference about common ancestor and direction of evolution. A single node is designated as common ancestor with unique path from it through evolutionary time to any other node.

Root is assigned through use of an outgroup – something that unambiguously separated earlier than species being considered.

Distance-Based Phylogenetics





Unrooted trees

Only specifies relationship between nodes. Says nothing about the direction of evolution

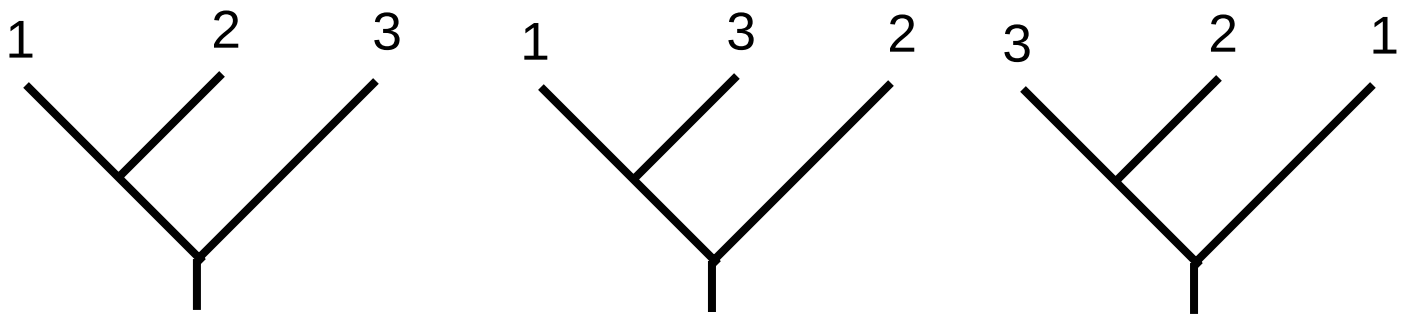
Why not always use rooted trees?

Distance-Based Phylogenetics

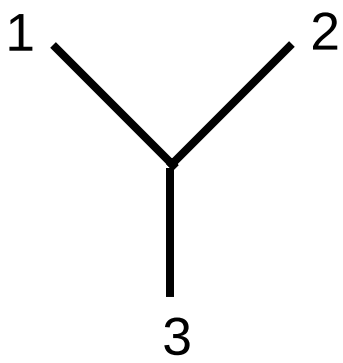
Why not always use rooted trees?

1 – Need a clear outgroup
2 – Computational difficulty

Consider 3 sequences 1, 2, and 3:
3 possible rooted trees



Only 1 possible unrooted tree



Distance-Based Phylogenetics

Why not always use rooted trees?

Number of sequences	Number of rooted trees	Number of unrooted trees
2	1	1
3	3	1
4	15	3
5	105	15
10	34,459,425	2,027, 025
15	213,458,046,767,875	7,905,853,580,625

$N_R=(2n-3)!/2^{n-2}(n-2)!$
$N_U=(2n-5)!/2^{n-3}(n-3)!$

Shortcuts…

UPGMA

Unweighted pair-group method with arithmetic mean

- Oldest distance method, statistically based
- Requires data be condensed to a measure of genetic distance
  **Build a distance matrix between taxa (I.e. sequences) **

Consider 4 sequences A, B, C, D

Species	A	B	C
B	d_{AB}		
C	d_{AC}	d_{BC}	
D	d_{AD}	d_{BD}	d_{CD}

d_{AB} is distance Between A and B

Step 1: Cluster the two closest sequences into composite group, i.e. if d_{AB} is smallest, make new group (AB).

UPGMA

Consider 4 sequences A, B, C, D

Species	A	B	C
B	d_{AB}		
C	d_{AC}	d_{BC}	
D	d_{AD}	d_{BD}	d_{CD}

d_{AB} is distance Between A and B

Step 1: Cluster the two closest sequences into composite group, i.e. if d_{AB} is smallest, make new group (AB).

Step 2: Create a new distance matrix between (AB) and C and D.

$d_{(AB)C} = 1/2\ (d_{AC} + d_{BC})$; $d_{(AB)D} = 1/2\ (d_{AD} + d_{BD})$

UPGMA

Consider 4 sequences A, B, C, D

Species	A	B	C
B	d_{AB}		
C	d_{AC}	d_{BC}	
D	d_{AD}	d_{BD}	d_{CD}

d_{AB} is distance Between A and B

Step 1: Cluster the two closest sequences into composite group, i.e. if d_{AB} is smallest, make new group (AB).

Step 2: Create a new distance matrix between (AB) and C and D.

$d_{(AB)C} = 1/2\ (d_{AC} + d_{BC})$; $d_{(AB)D} = 1/2\ (d_{AD} + d_{BD})$

Step 3: Using new matrix, cluster the two closest sequences into composite group. Repeat above until all species have been grouped.

UPGMA

Species	A	B	C
B	d_{AB}		
C	d_{AC}	d_{BC}	
D	d_{AD}	d_{BD}	d_{CD}

d_{AB} is distance Between A and B

Step 1: Cluster the two closest sequences into composite group, i.e. if d_{AB} is smallest, make new group (AB).

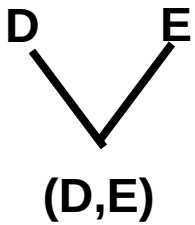
Step 2: Create a new distance matrix between (AB) and C and D.
$d_{(AB)C} = 1/2\ (d_{AC} + d_{BC})$; $d_{(AB)D} = 1/2\ (d_{AD} + d_{BD})$

Step 3: Using new matrix, cluster the two closest sequences into composite group. Repeat above until all species have been grouped.

Step 4: For scaled branch lengths, put node halfway between grouped species.

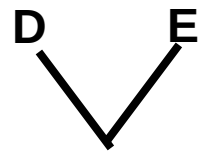
UPGMA - example

Species	A	B	C	D
B	9			
C	8	11		
D	12	15	10	
E	15	18	13	5



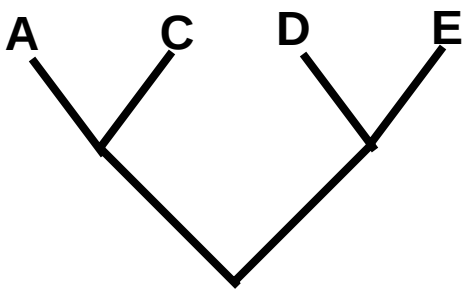
UPGMA - example

Species	A	B	C	D
B	9			
C	8	11		
D	12	15	10	
E	15	18	13	5



(D,E)

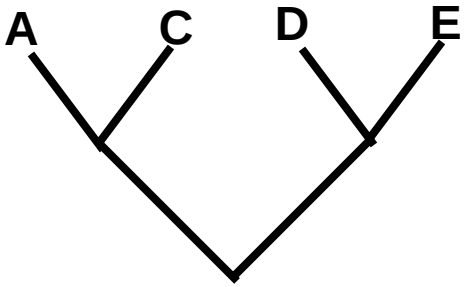
Species	A	B	C
B	9		
C	8	11	
DE	13.5	16.5	11.5



(A,C) (D,E)

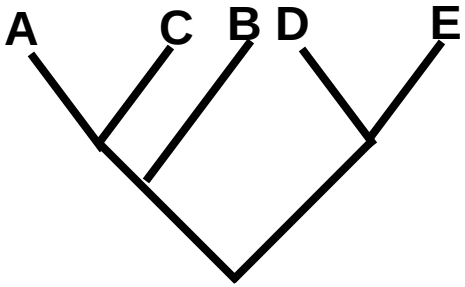
UPGMA - example

Species	A	B	C
B	9		
C	8	11	
DE	13.5	16.5	11.5



(A,C) (D,E)

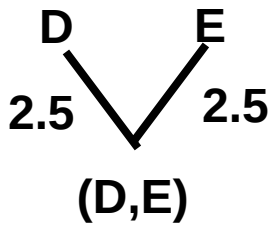
Species	B	AC
AC	10	
DE	16.5	12.5



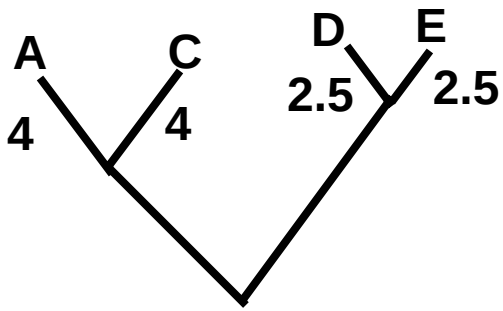
(((A,C)B)(D,E))

UPGMA – adding distances

Species	A	B	C	D
B	9			
C	8	11		
D	12	15	10	
E	15	18	13	5

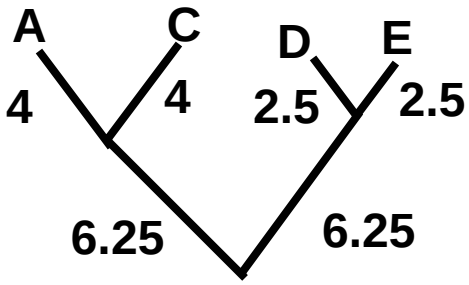


Species	A	B	C
B	9		
C	8	11	
DE	13.5	16.5	11.5

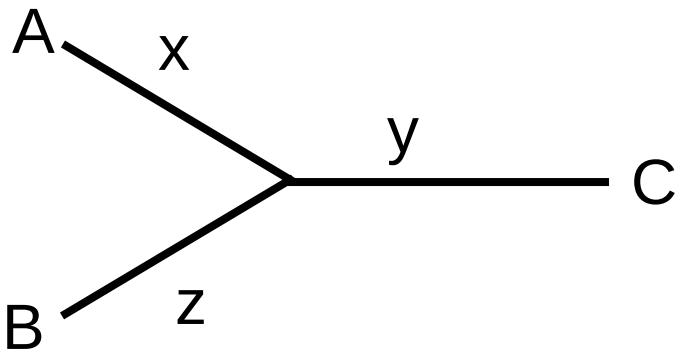


UPGMA – adding distances

Species	B	AC
AC	10	
DE	16.5	12.5



Branch lengths for scaled unrooted tree = Fitch-Margoliash Algorithm for 3 sequences

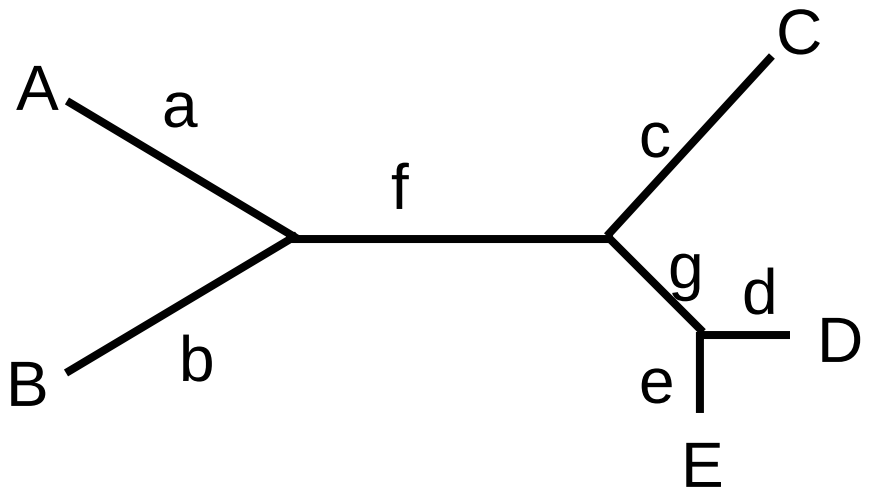


$d_{AC} = x+y$
$d_{AB} = x+z$
$d_{BC} = y+z$

$$x=(d_{AB} + d_{AC} - d_{BC})/2$$
$$y=(d_{AC} + d_{BC} - d_{AB})/2$$
$$z=(d_{AB} + d_{BC} - d_{AC})/2$$

Note: F-M assumes additivity of branch lengths but DOES NOT Assume equal rates of evolution along branches.
(Can specify this though : Kitsch-Margoliash)

Fitch-Margoliash Algorithm for >3 sequences



Steps:

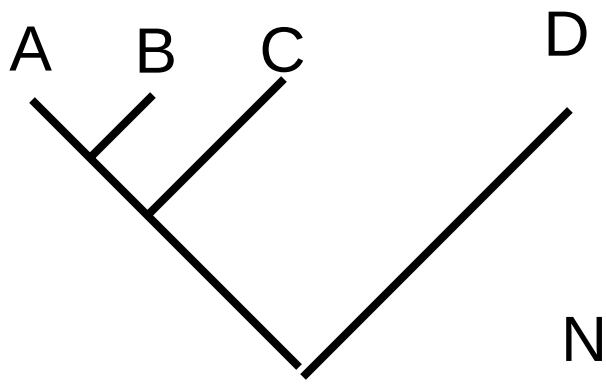
1- Find closest 2 sequences (D,E)
2- Treat rest as composite and take average of D to (ABC), E to (ABC)
3- Use these to calculate d, e
4- Make new composite DE
5- Make new distance table
6- Find next most closely related pair and repeat from step 2

Now repeat starting with another pair as the closest starting pair
In the end, calculate all predicted distances for all trees, and choose
What best fits data

Transformed Distance Method

UPGMA assumes constant rate
Of evolution across all lineages

Can allow different rates of evolution across different lineages if you normalize using an external reference that diverged early…i.e. use an outgroup



Define $\bar{d}_D$ =average distance
Between outgroup and all ingroups

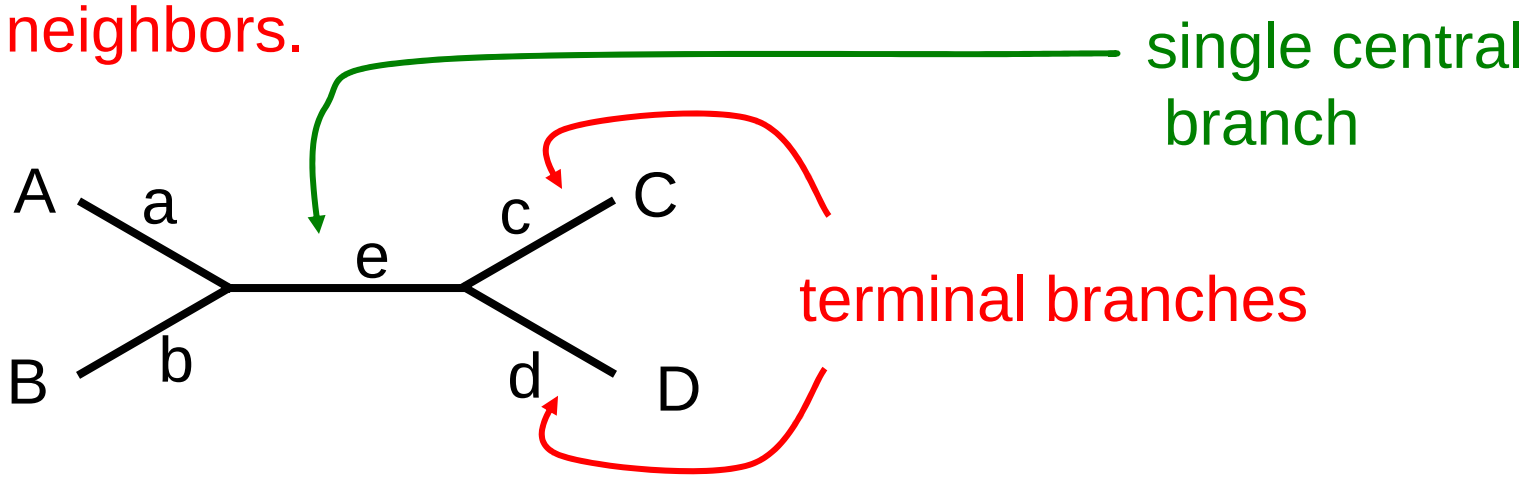
$$d'_{ij} = (d_{ij} - d_{iD} - d_{jD})/2 + \bar{d}_D$$

Now use d'_{ij} to do the clustering
..basically just comes from the insight that ingroups evolved separately from each other <u>ONLY AFTER</u> they diverged from outgroup

Neighbor's Relation Method

Variant of UPGMA that pairs species in a way that creates a tree with minimal overall branch lengths.

Pairs of sequences separated by only 1 node are said to be neighbors.



For this tree topology

$d_{AC} + d_{BD} = d_{AD} + d_{BC} = a + b + c + d + 2e = d_{AB} + d_{CD} + 2e$

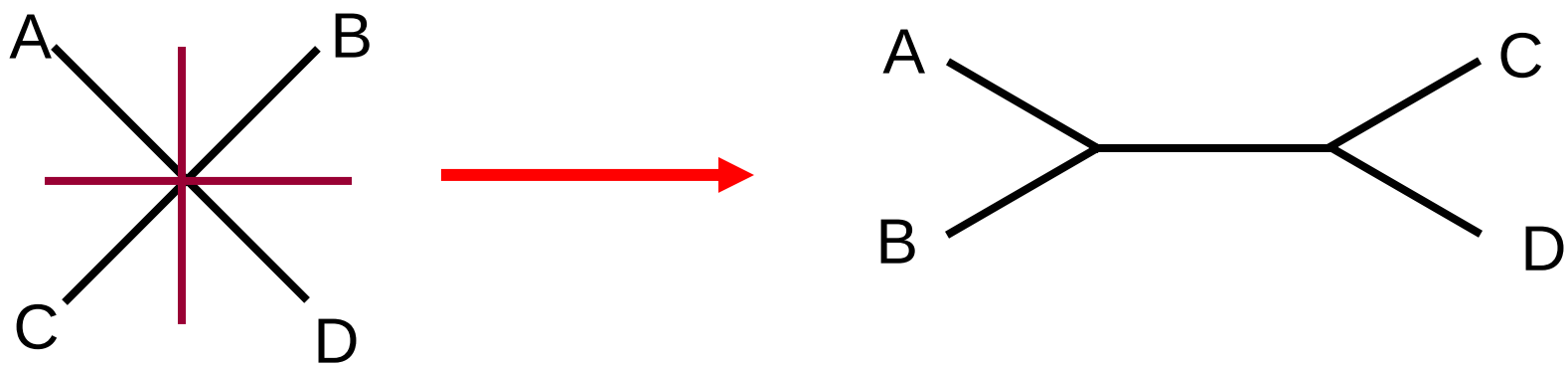
For neighbor relations, four-point condition will be true:

$d_{AB} + d_{CD} < d_{AC} + d_{BD}$ …and… $d_{AB} + d_{CD} < d_{AD} + d_{BC}$

So just have to consider all pairwise arrangements and determine which one satisfies the four-point condition.

Neighbor-Joining Methods

Start with star-like tree. Find neighbors sequentially to minimize total length of all branches



Studier & Kepler 1988:

$Q_{12}=(N-2)d_{12} - \Sigma d_{1i} - \Sigma d_{2i}$

Where any 2 sequences can be 1 and 2

Try all possible sequence combinations. Whichever combination of pairs gives the smallest Q_{12} is the final tree!

Maximum Likelihood

- A purely statistical method.
- Probablilities for every nucleotide substitution in a set of aligned sequences is considered.
- Calculation of probabilities is complex since ancestor is unknown
- Test all possible trees and calculate the aggregate probablility.
- Tree with single highest aggregate probablilty is the most likely to reflect the true phylogenetic tree.

VERY COMPUTATIONALLY INTENSE

Parsimony

Parsimony: a derogatory term from the 1930s and 1940s
To describe someone who was especially careful with
Spending money.

Biologically: Attach preference to an evolutionary pathway
That minimizes the number of mutational events since
(1) Mutations are rare events, and
(2) The more unlikely events a model postulates, the less l
likely the model is to be true.

Parsimony: a character-based method, NOT a distance-based
method.

Parsimony

For parsimony analysis, positions in a sequence alignment fall into one of two categories: informative and uninformative.

Position

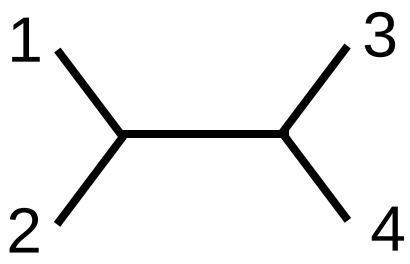
Sequence	1	2	3	4	5	6
1	G	G	G	G	G	G
2	G	G	G	A	G	T
3	G	G	A	T	A	G
4	G	A	T	C	A	T

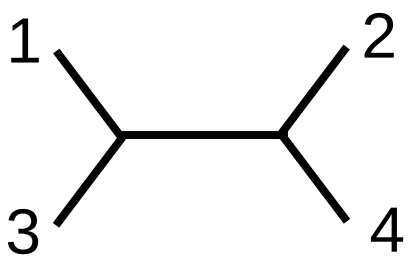
Only 3 possible unrooted trees you can make…

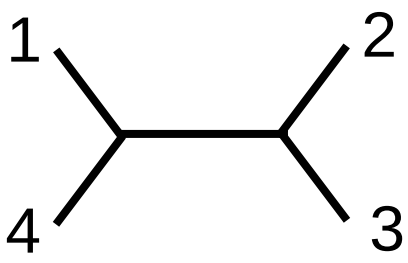
Parsimony

Position

Sequence	1	2	3	4	5	6
1	G	G	G	G	G	G
2	G	G	G	A	G	T
3	G	G	A	T	A	G
4	G	A	T	C	A	T





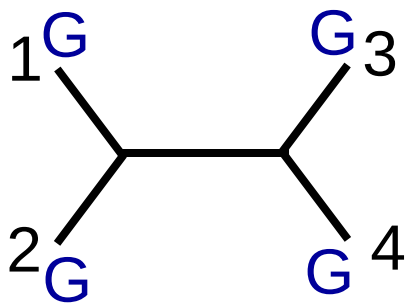


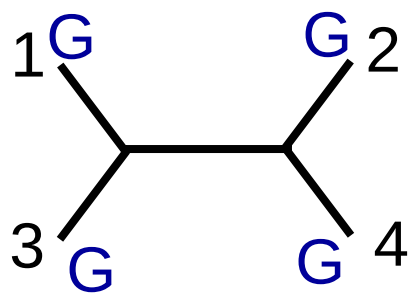
Which tree is the right one?

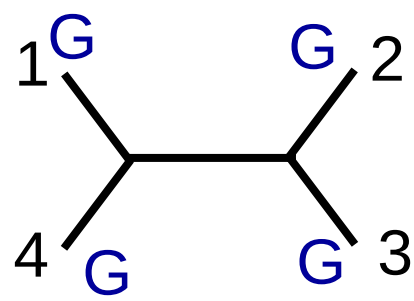
Parsimony

Position

Sequence	1	2	3	4	5	6
1	G	G	G	G	G	G
2	G	G	G	A	G	T
3	G	G	A	T	A	G
4	G	A	T	C	A	T





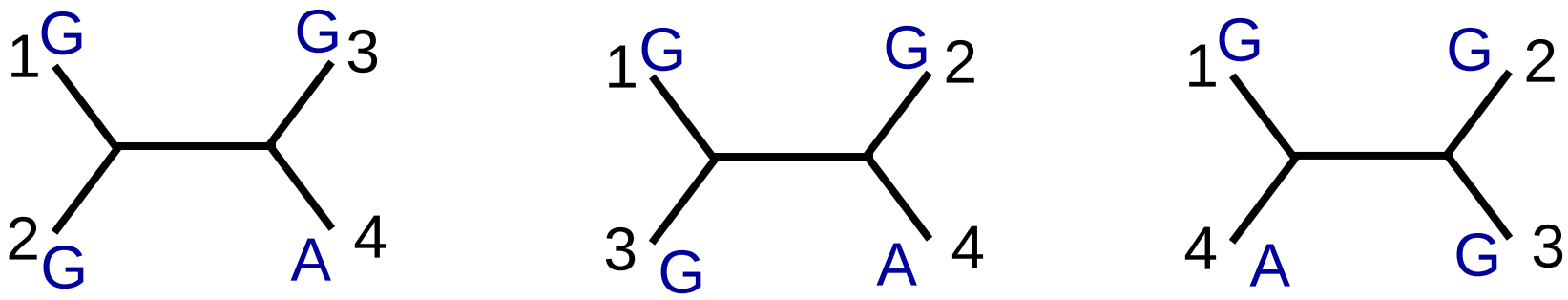


Invariant positions – contain NO INFORMATION → uninformative

Parsimony

Position

Sequence	1	2	3	4	5	6
1	G	G	G	G	G	G
2	G	G	G	A	G	T
3	G	G	A	T	A	G
4	G	A	T	C	A	T

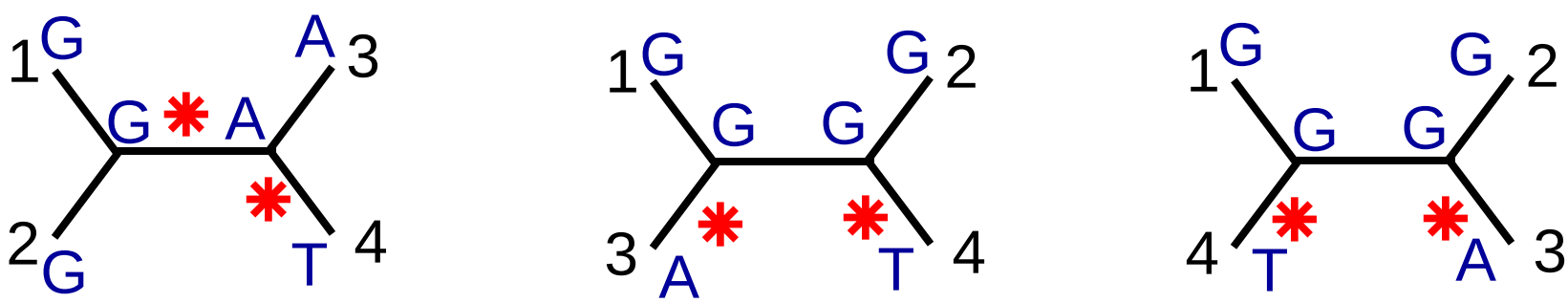


Equally uninformative – need one mutation in each tree

Parsimony

Position

Sequence	1	2	3	4	5	6
1	G	G	G	G	G	G
2	G	G	G	A	G	T
3	G	G	A	T	A	G
4	G	A	T	C	A	T

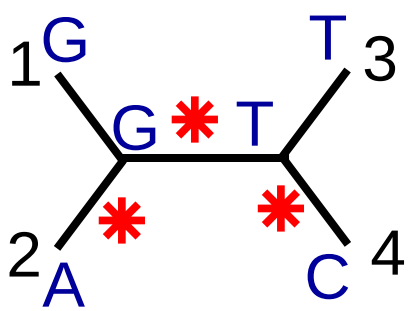


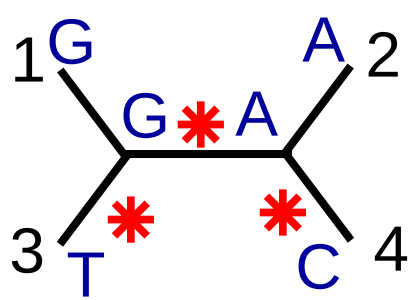
Also uninformative – need two mutations in each tree

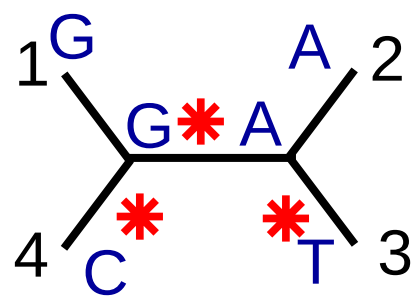
Parsimony

Position

Sequence	1	2	3	4	5	6
1	G	G	G	G	G	G
2	G	G	G	A	G	T
3	G	G	A	T	A	G
4	G	A	T	C	A	T





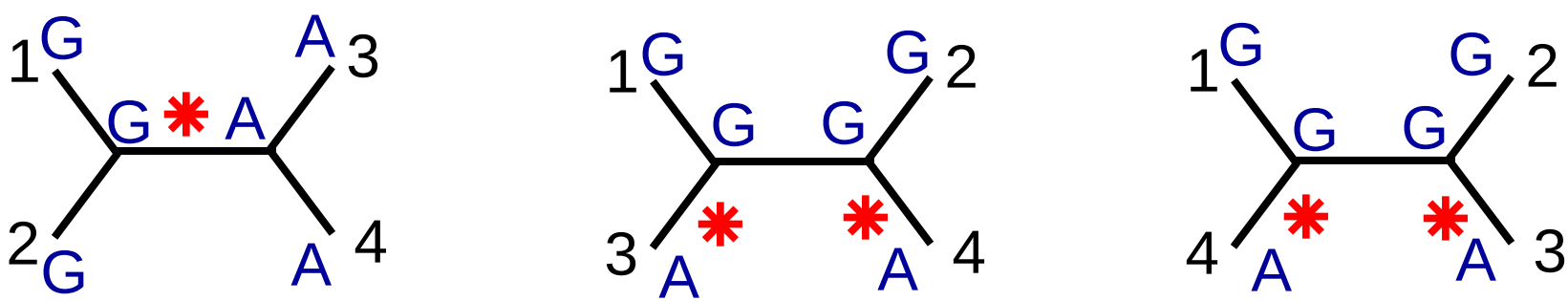


Also uninformative – need three mutations in each tree

Parsimony

Position

Sequence	1	2	3	4	5	6
1	G	G	G	G	G	G
2	G	G	G	A	G	T
3	G	G	A	T	A	G
4	G	A	T	C	A	T

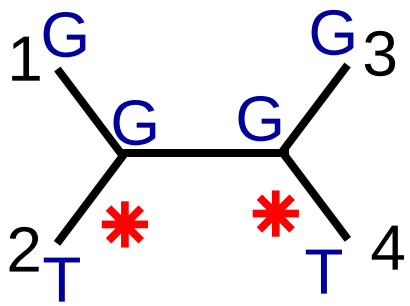


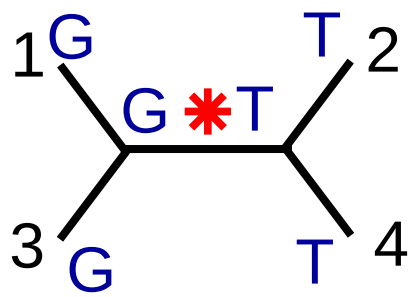
Informative! – need only one mutation in one tree but two In the other trees!

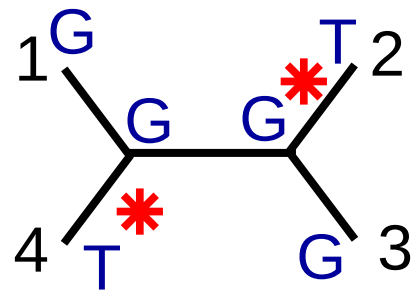
Parsimony

Position

Sequence	1	2	3	4	5	6
1	G	G	G	G	G	G
2	G	G	G	A	G	T
3	G	G	A	T	A	G
4	G	A	T	C	A	T







Informative! – need only one mutation in one tree but two In the other trees!

Parsimony

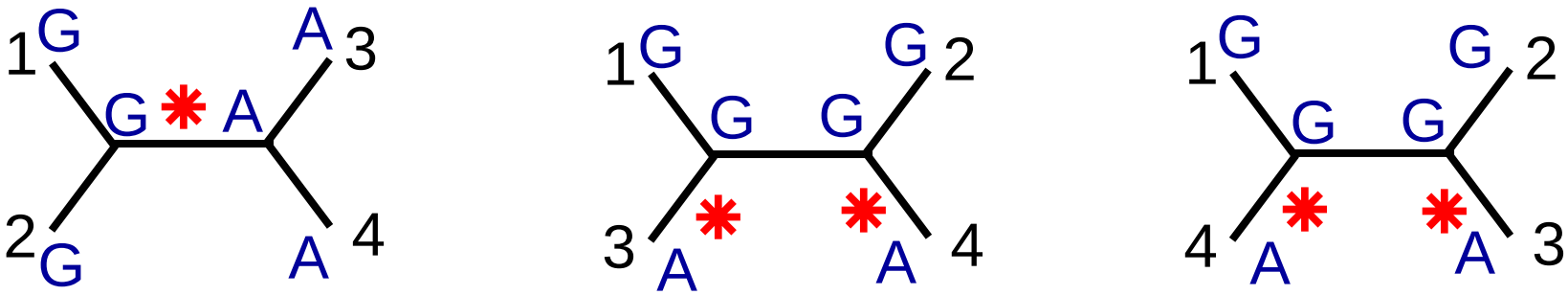
Position

Sequence	1	2	3	4	5	6
1	G	G	G	G	G	G
2	G	G	G	A	G	T
3	G	G	A	T	A	G
4	G	A	T	C	A	T

So to be Informative, need at least 2 different nucleotides And each has to be present at least twice.

Every tree is considered for every site, maintaining a running score of the number of mutations required. The tree with the smallest number of invoked mutations is the most parsimonious

Parsimony



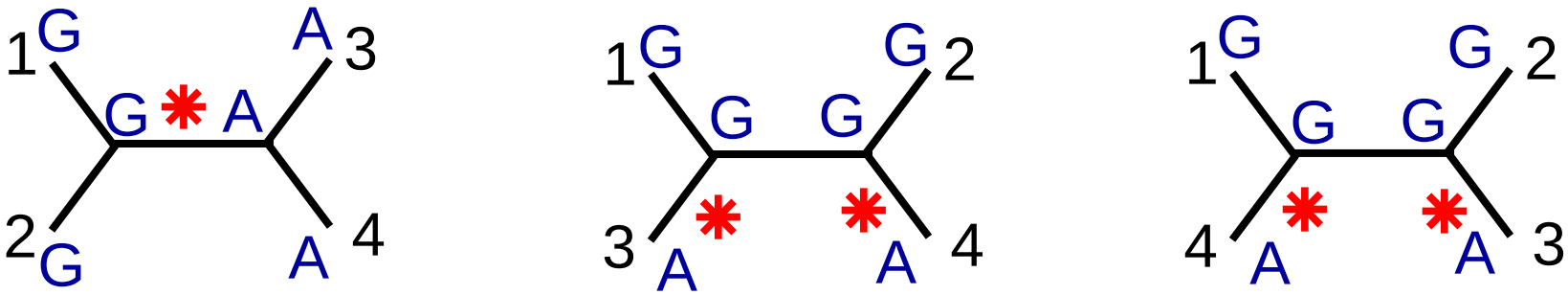
Mathematically, most likely candidate nts at an Internal node are:
{descendent node 1} Ω {descendant node 2}

IF this is null set, then most likely candidate nts are:
{descendent node 1} U {descendant node 2}

Σ ***U*** = minimum number of substitutions required to account for nts at terminal nodes since they last shared common ancestor

Total number of substitutions, informative + uninformative = tree length

Parsimony



If you use parsimony but weigh the mutations by some kind of scoring system that accounts for the likelihood of each mutation →weighted parsimony

By-product of parsimony is inference of nt identity in the ancestral sequence

Parsimony

10 sequences: > 2 million possible trees…
Need a better way…

Branch and bound (Hardy and Penny, 1982)

Step 1: Determine an upper bound to the length of the most parsimonious tree = L - either chosen randomly, or else using a computationally fast way like UPGMA

Step 2: Grow trees incrementally by adding branches to a smaller tree that describes just some of the sequences.

Step 3: If at any point, the number of required substitutions is > L, abandon that tree.

Step 4: As soon as you get a tree with fewer substiitutions than L, use that tree as the new upper bound to make remainder of the search even more efficient.

Works for <= 20 sequences

Parsimony

For > 20 sequences

Heuristic Searches

Assumption: Alternative trees are not all independent of each other. Most parsimonious trees have similar topologies.

Step 1: Construct an initial tree as a good guess: UPGMA, and use it as a starting point.

Step 2: Branch-swap subtrees and graft them onto the starting tree, keeping overall topology. See how many are shorter than the starting tree. The prune and re-graft, and see if it keeps getting better.

Step 3: Repeat until a round of branch swapping fails to generate any better trees

Parsimony

Often get tens or hundreds of equally parsimonious trees

Build a consensus tree – any internal node supported by At least half the trees becomes a simple bifurcation.

Phylogenetic Software

PHYLIP: Phylogenetics Inference Package
free at http://evolution.genetics.washington.edu

Includes many programs including various distance methods, maximum likelihood, parsimony, with many of the options we've discussed.

PAUP: Phylogenetic Analysis Using Parsimony
http://www.lms.si.edu/PAUP - NOT FREE
Now includes maximum likelihood and distance methods as well

Tree of Life

Carl Woese and colleagues, 1970s

Used 16S rRNA – all organisms possess.

Found 3 major evolutionary groups:
Bacteria
Eucarya
Archea (including thermophilic bacteria

Human Origins:
mt DNA sequences – huan populations differ by ~ 0.33% (very small). Greatest differences NOT between current populations on different continents, but between human populations residing in Africa – “out of Africa” theory

Mitochondrial “eve” and Y-chromosome “adam”