SISG 2014

SISG Module 12: Molecular Phylogenetics

19th Summer Institute in Statistical Genetics

UNIVERSITY of WASHINGTON

(This page left intentionally blank.)

Module 12: Molecular Phylogenetics

http://evolution.gs.washington.edu/sisg/2014/

MTH Thanks to Paul Lewis, Tracy Heath, Joe Felsenstein, Peter Beerli, Derrick Zwickl, and Joe Bielawski for slides

Monday July 14: Day I

8:30AM to 10:00AM	Introduction (Mark Holder) Parsimony methods for phylogeny reconstruction (Mark Holder) Distance–based methods for phylogeny reconstruction (Mark Holder)
10:30AM to noon	Topology Searching (Mark Holder) Parsimony and distances demo in PAUP* (Mark Holder)
1:30PM to 3:00PM	Nucleotide Substitution Models and Transition Probabilities (Jeff Thorne) Likelihood – (Joe Felsenstein)
3:30PM to 5:00PM	PHYLIP lab: likelihood – (Joe Felsenstein) PAUP* lab (Mark Holder)

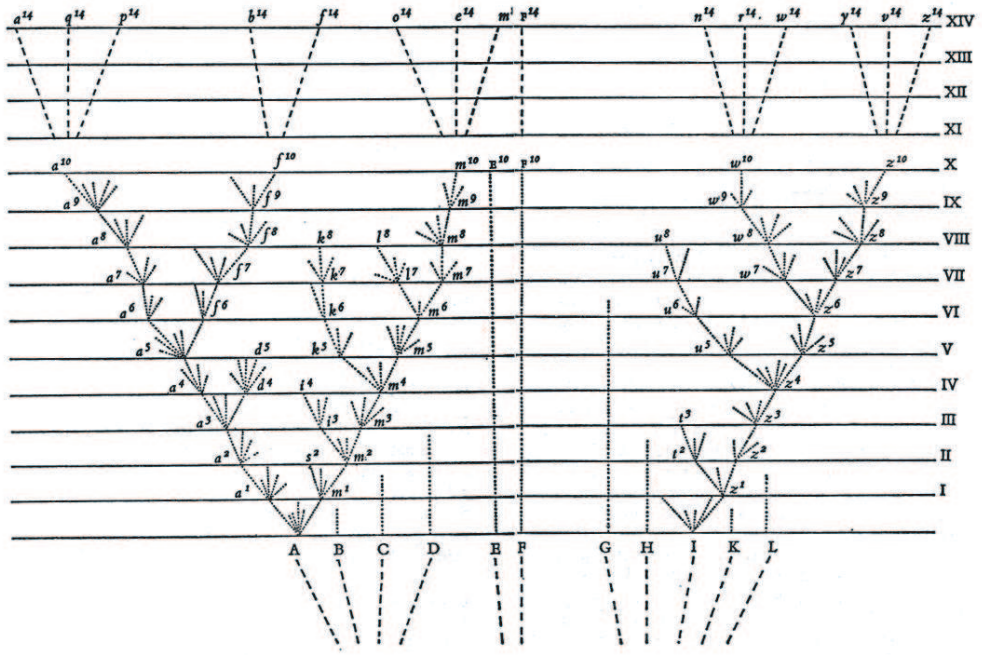
Tuesday July 15: Day II

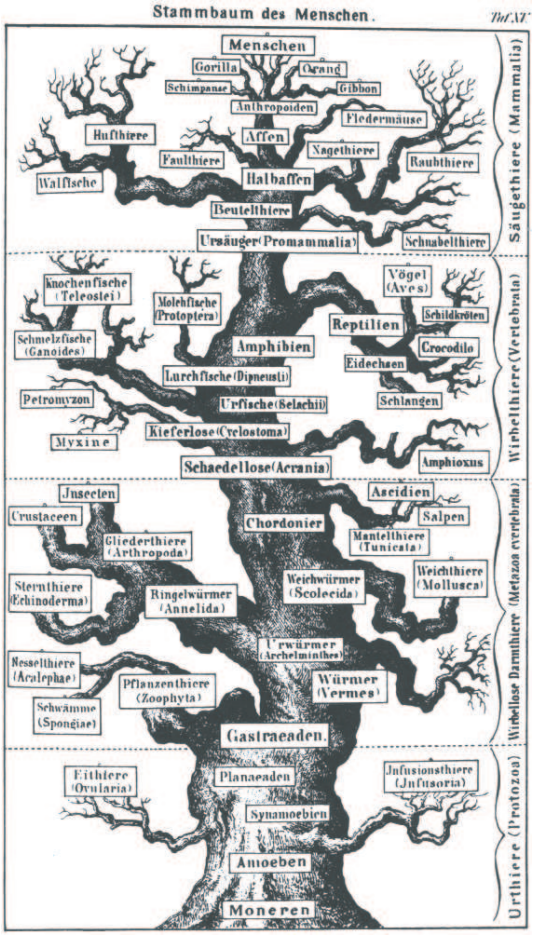
8:30AM to 10:00AM	Bootstraps and Testing Trees (Joseph Felsenstein) Bootstrapping in Phylip (Joe Felsenstein)
10:30AM to noon	More Realistic Evolutionary Models (Jeff Thorne)
1:30PM to 3:00PM	Bayesian Inference and Bayesian Phylogenetics (Jeff Thorne)
3:30PM to 5:00PM	MrBayes Computer Lab – (Mark Holder)
5:00PM to 6:00PM	Tutorial (questions and answers session)

Wednesday July 16: Day III

8:30AM to 10:00AM	Divergence Time Estimation – (Jeff Thorne) BEAST demo (Mark Holder)
10:30AM to noon	The Coalescent – (Joe Felsenstein) The Comparative Method – (Joe Felsenstein) Future Directions – (Joe Felsenstein)

Darwin's 1859 "On the Origin of Species" had one figure:





Human family tree

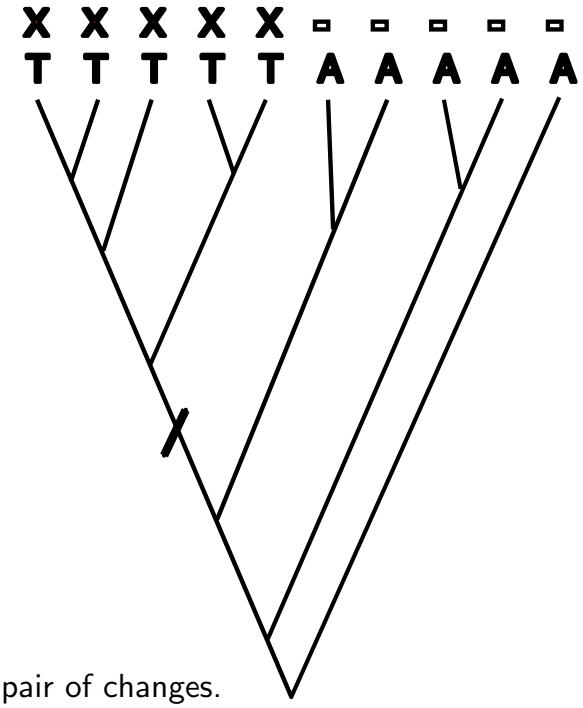
from Haeckel, 1874

Fig. 20, p. 171, in Gould, S. J. 1977.
Ontogeny and phylogeny.
Harvard University Press, Cambridge, MA

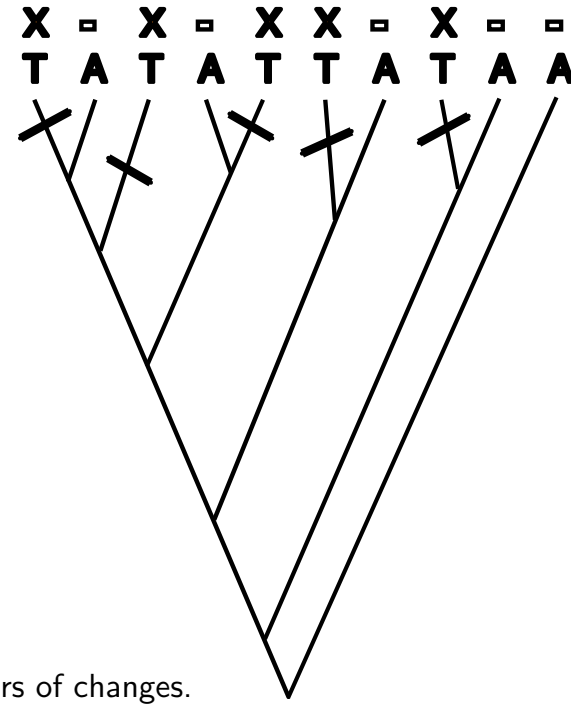
Are desert green algae adapted to high light intensities?

Species	Habitat	Photoprotection
1	terrestrial	xanthophyll
2	terrestrial	xanthophyll
3	terrestrial	xanthophyll
4	terrestrial	xanthophyll
5	terrestrial	xanthophyll
6	aquatic	none
7	aquatic	none
8	aquatic	none
9	aquatic	none
10	aquatic	none

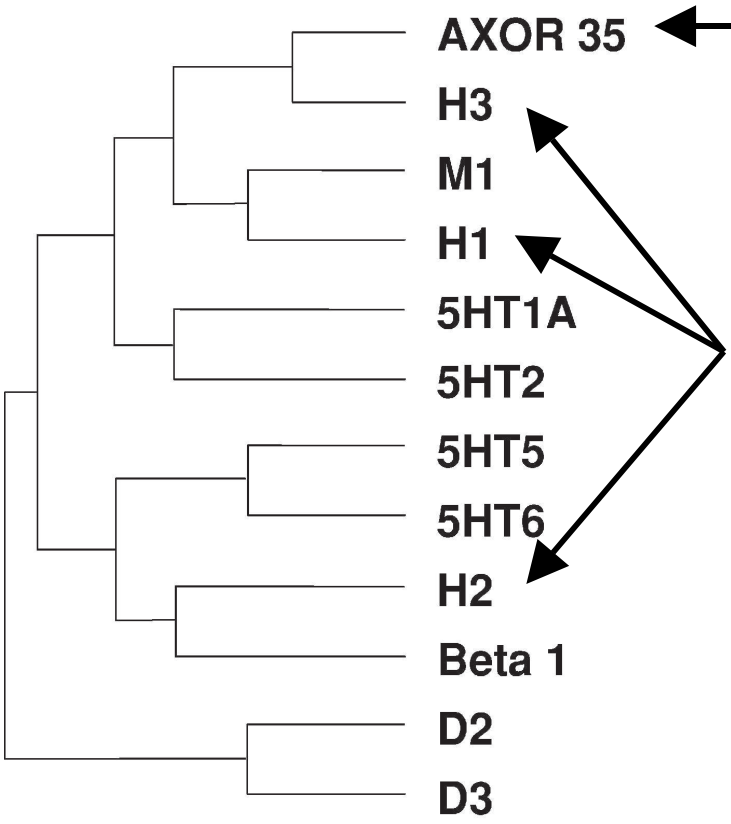
Phylogeny reveals the events that generate the pattern



pair of changes.
Coincidence?



5 pairs of changes.
Much more convincing



GPCR with unknown ligan

Natural ligand known
to be histamine

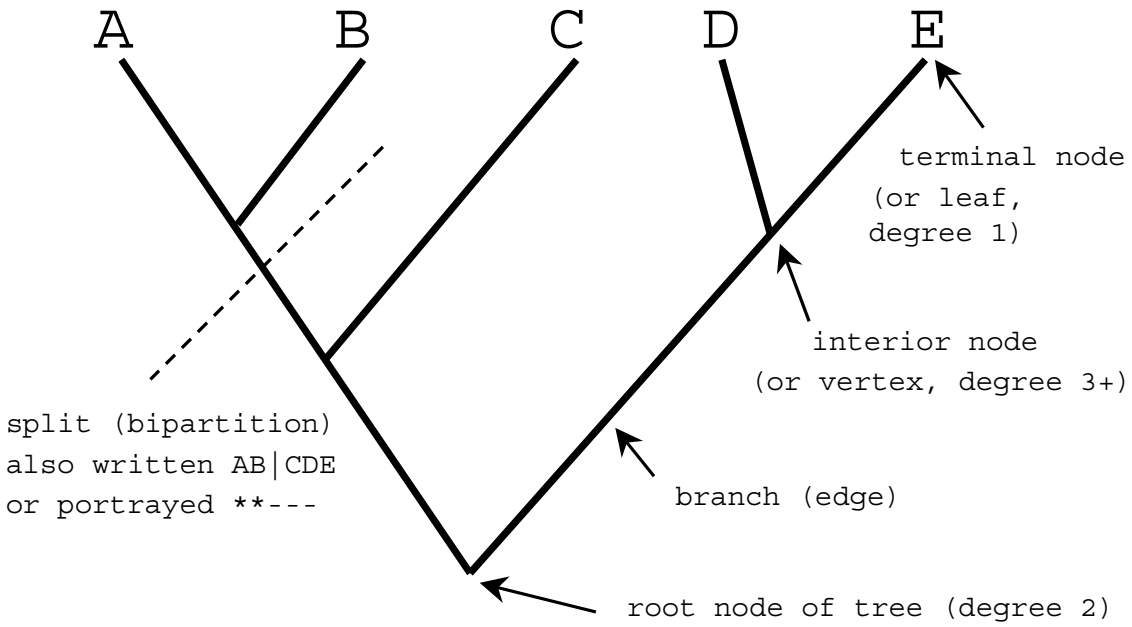
Which ligand for AXOR35
would you test first?

Wise, A., Jupe, S. C., and Rees, S. 2004. The identification of ligands at orphan G-protein coupled receptors. *Annu. Rev. Pharmacol. Toxicol.* 44:43-66.

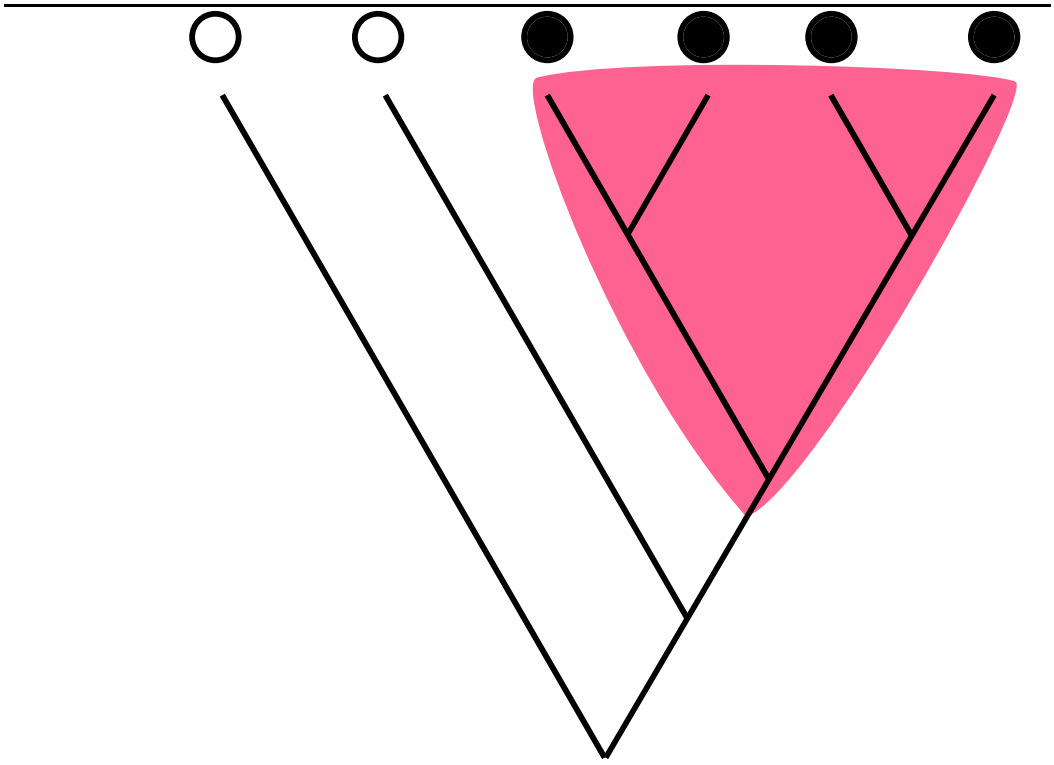
Many evolutionary questions require a phylogeny

- Estimating the number of times a trait evolved
- Determining whether a trait tends to be lost more often than gained, or vice versa
- Estimating divergence times
- Distinguishing homology from analogy
- Inferring parts of a gene under strong positive selection

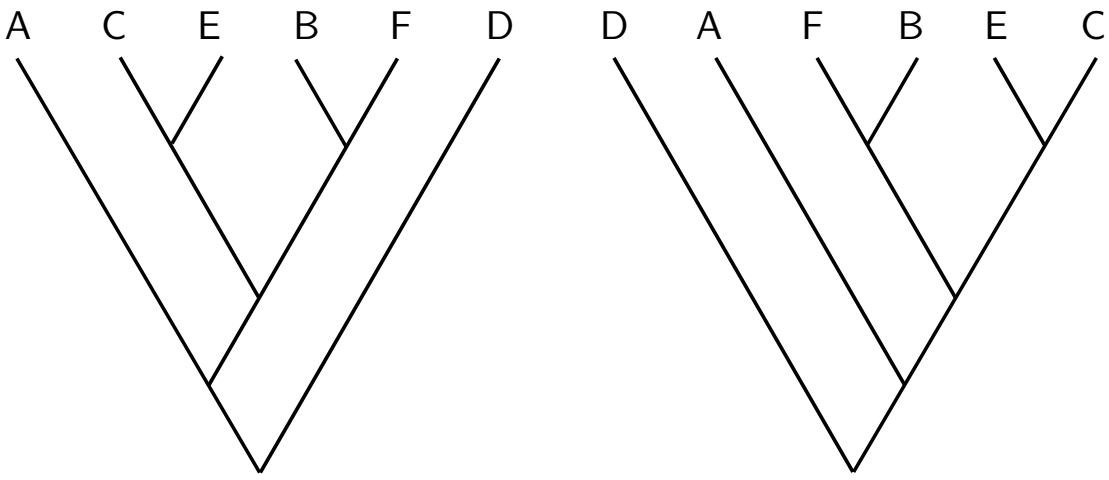
Tree terminology



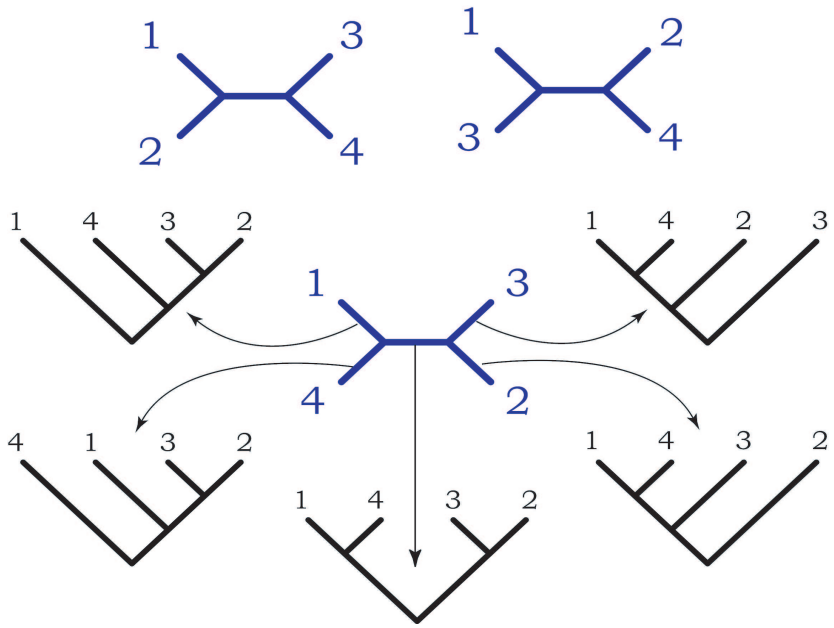
Monophyletic groups ("clades"): the basis of phylogenetic classification



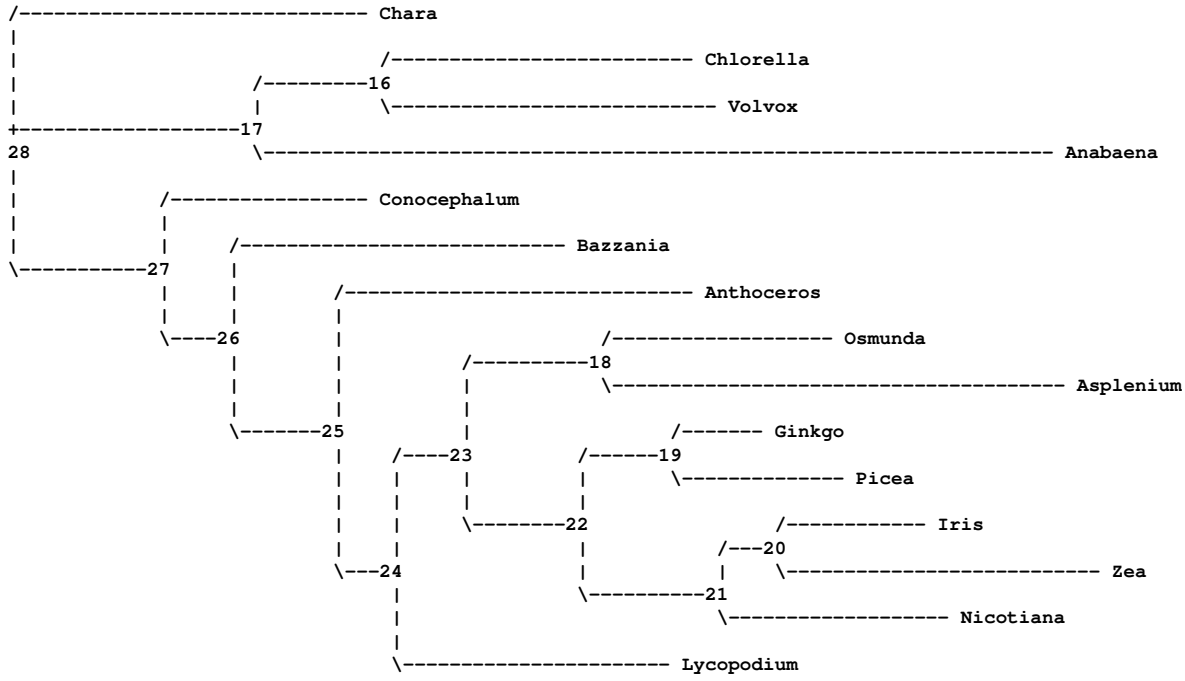
Branch rotation does not matter



Rooted vs unrooted trees



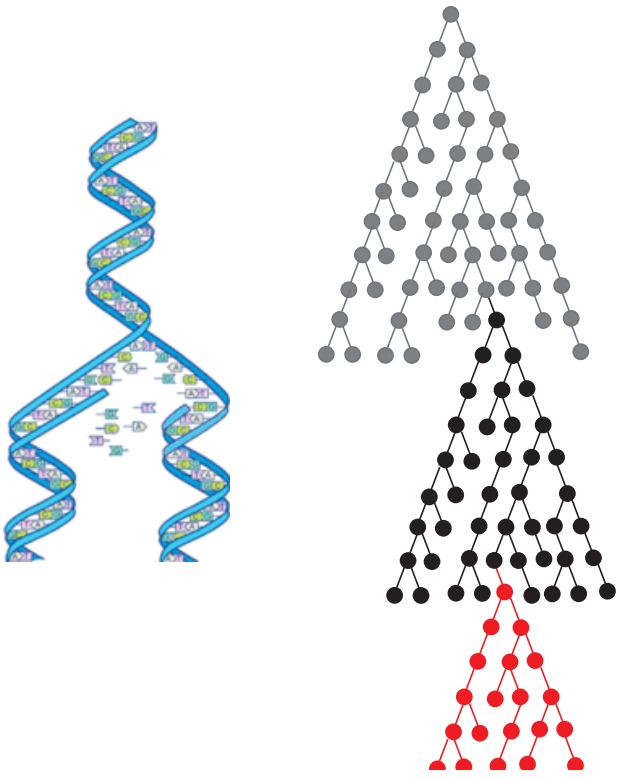
Warning: software often displays unrooted trees like this:



We use trees to represent genealogical relationships in several contexts.

Domain	Sampling	tree	The cause of splitting
Population Genetics	> 1 indiv/sp. Few species	Gene tree	> 1 descendants of a single gene copy
Phylogenetics	Few indiv/sp. Many species	Phylogeny	speciation
Molecular Evolution	> 1 locus/sp. > 1 species	Gene tree. Gene family tree	speciation or duplication

Phylogenies are an inevitable result of molecular genetics



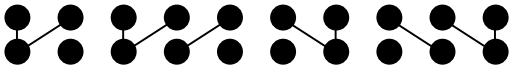
Genealogies within a population

Present

Past

Genealogies within a population

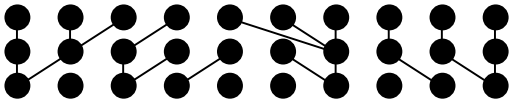
Present



Past

Genealogies within a population

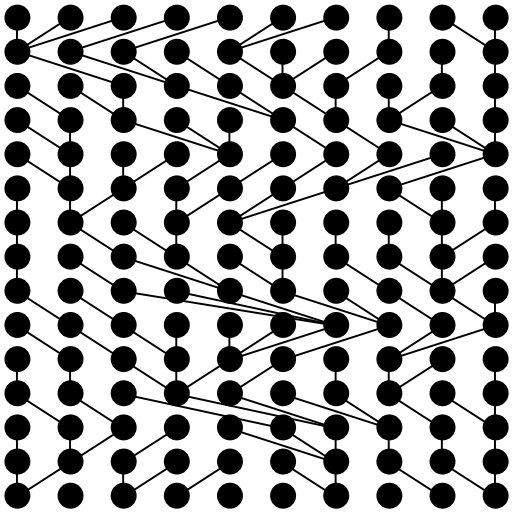
Present



Past

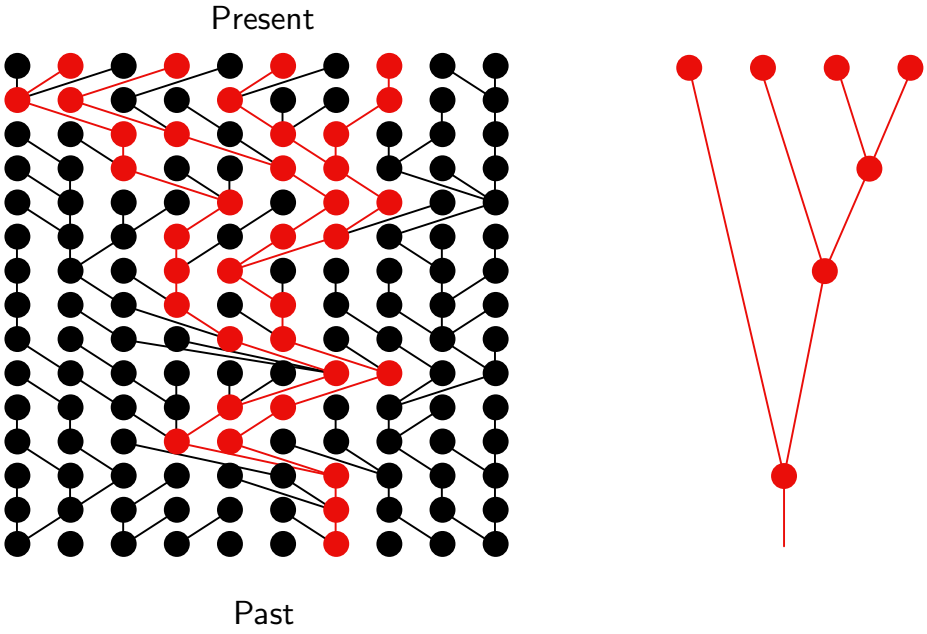
Genealogies within a population

Present



Past

Genealogies within a population



Biparental inheritance would make the picture messier, but the genealogy of the gene copies would still form a tree (if there is no recombination).

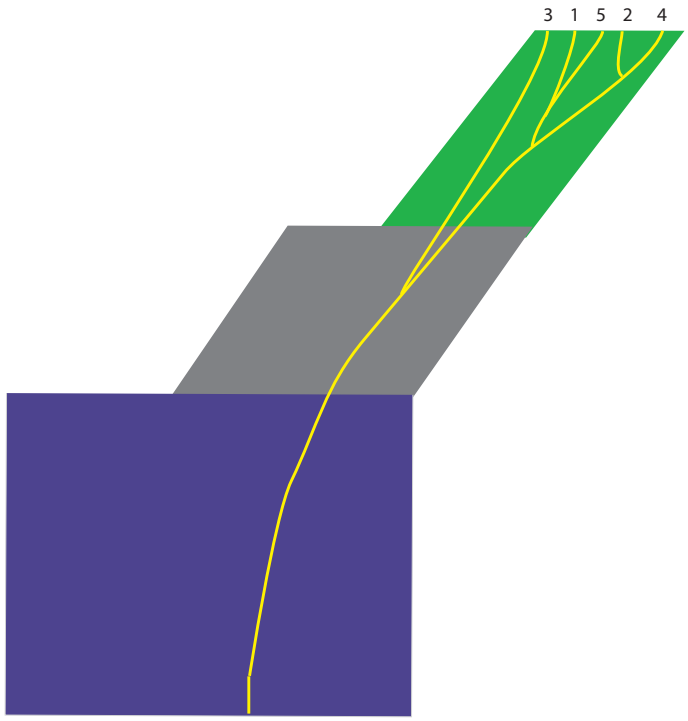
terminology: genealogical trees within population or species trees

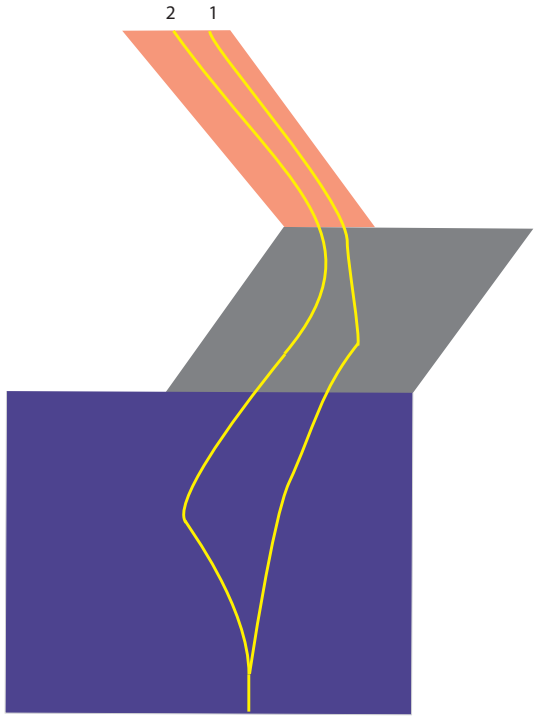
It is tempting to refer to the tips of these gene trees as alleles or haplotypes.

- allele – an alternative form a gene.
- haplotype – a linked set of alleles

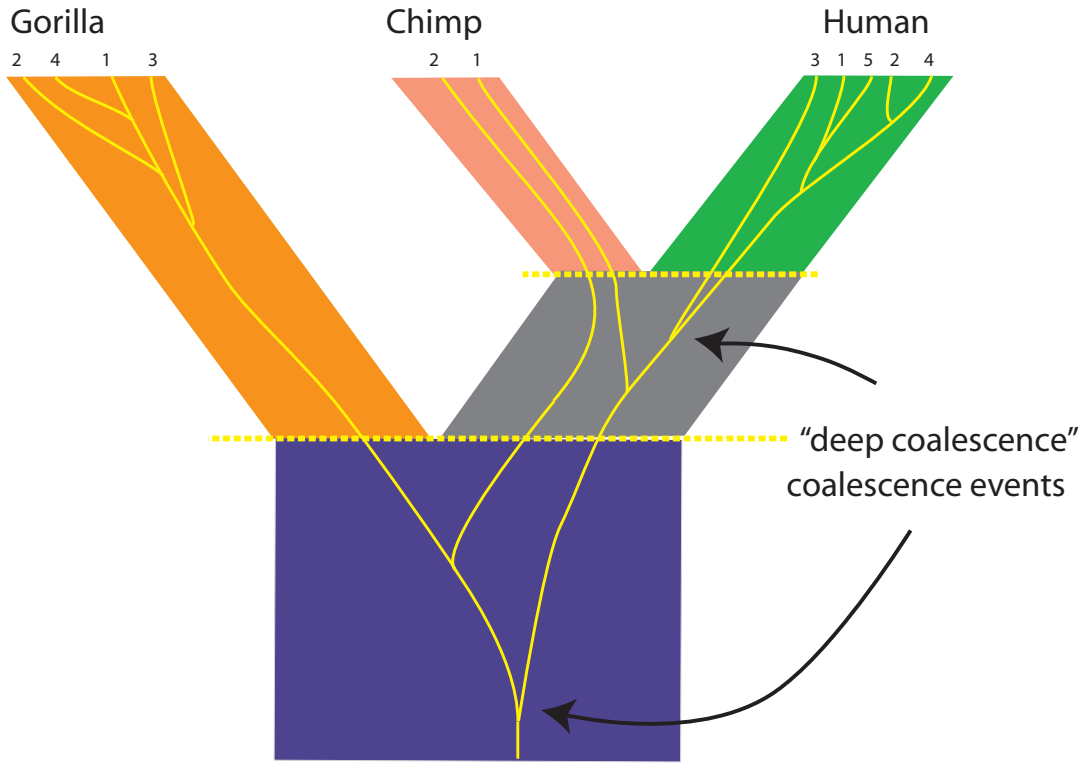
But both of these terms require a differences in sequence.

The gene trees that we draw depict genealogical relationships – regardless of whether or not nucleotide differences distinguish the "gene copies" at the tips of the tree.


3 1 5 2 4


2 1

A "gene tree" within a species tree



terminology: genealogical trees within population or species trees

- coalescence – merging of the genealogy of multiple gene copies into their common ancestor. "Merging" only makes sense when viewed *backwards in time*.

- "deep coalescence" or "incomplete lineage sorting" refer to the *failure* of gene copies to coalesce within the duration of the species – the lineages coalesce in an ancestral species

Inferring a species tree while accounting for the coalescent

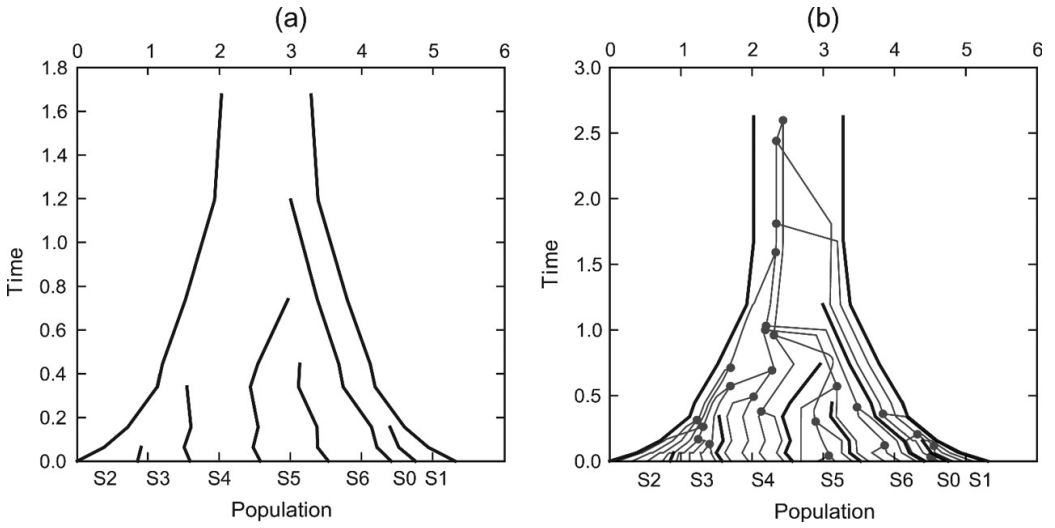
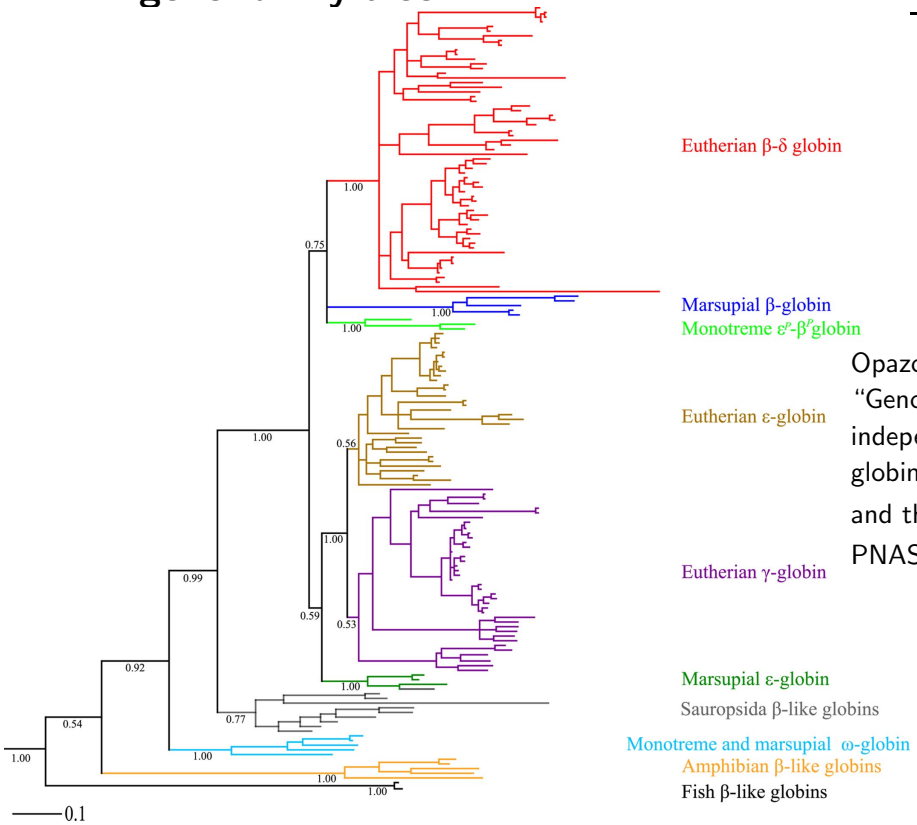
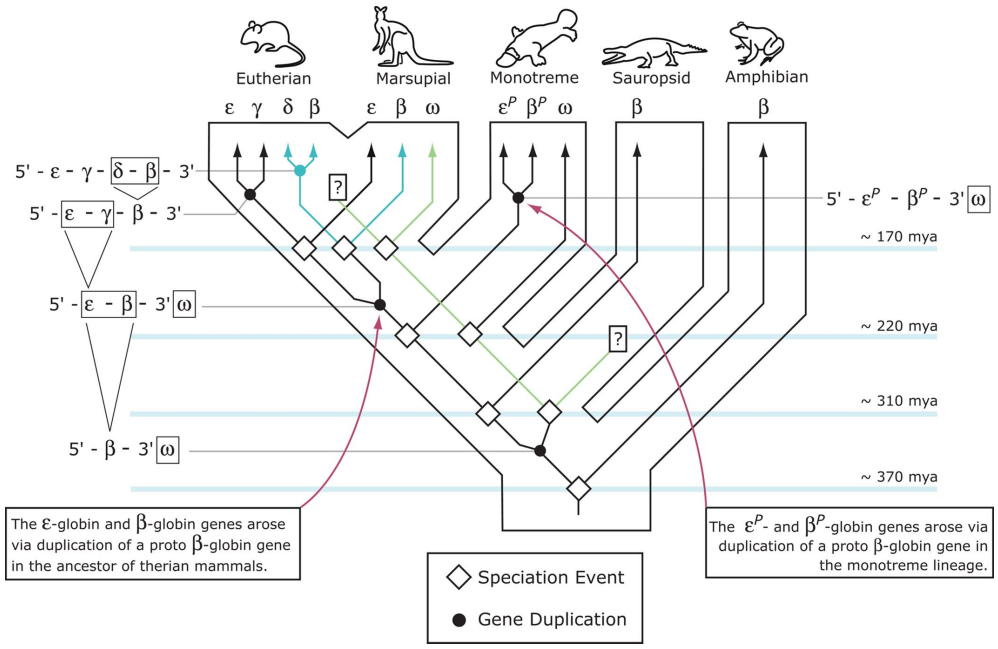


Figure 2 from Heled and Drummond (2010)

A "gene family tree"



Opazo, Hoffmann and Storz "Genomic evidence for independent origins of β-like globin genes in monotremes and therian mammals" PNAS **105(5)** 2008



Opazo, Hoffmann and Storz "Genomic evidence for independent origins of β-like globin genes in monotremes and therian mammals" PNAS **105(5)** 2008

terminology: trees of gene families

- duplication – the creation of a new copy of a gene within the same genome.
- homologous – descended from a common ancestor.
- paralogous – homologous, but resulting from a gene duplication in the common ancestor.
- orthologous – homologous, and resulting from a speciation event at the common ancestor.

Joint estimation of gene duplication, loss, and species trees using PHYLDOG

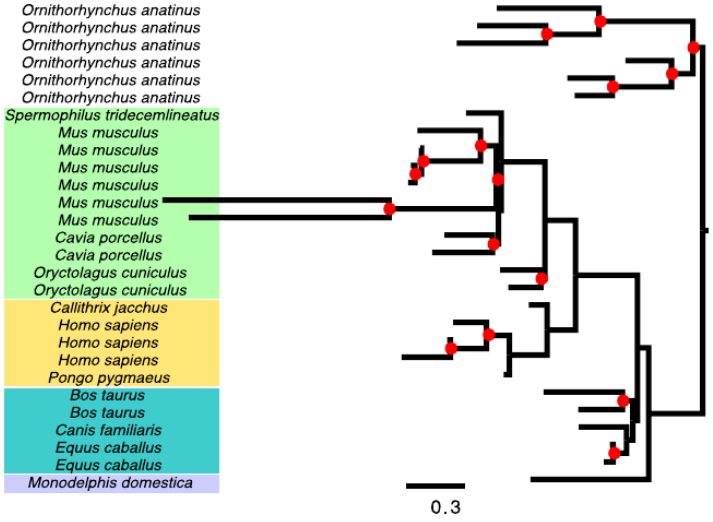


Figure 2A from Boussau et al. (2013)

Multiple contexts for tree estimation (again):

	The cause of splitting	**Important caveats**
"Gene tree" or "a coalescent"	DNA replication	recombination is usually ignored
Species tree Phylogeny	speciation	recombination, hybridization, lateral gene transfer, and deep coalescence cause conflict in the data we use to estimate phylogenies
Gene family tree	speciation or duplication	recombination (eg. domain swapping) is not tree-like

Joint estimation of gene duplication, loss, and coalescence with DLCoalRecon

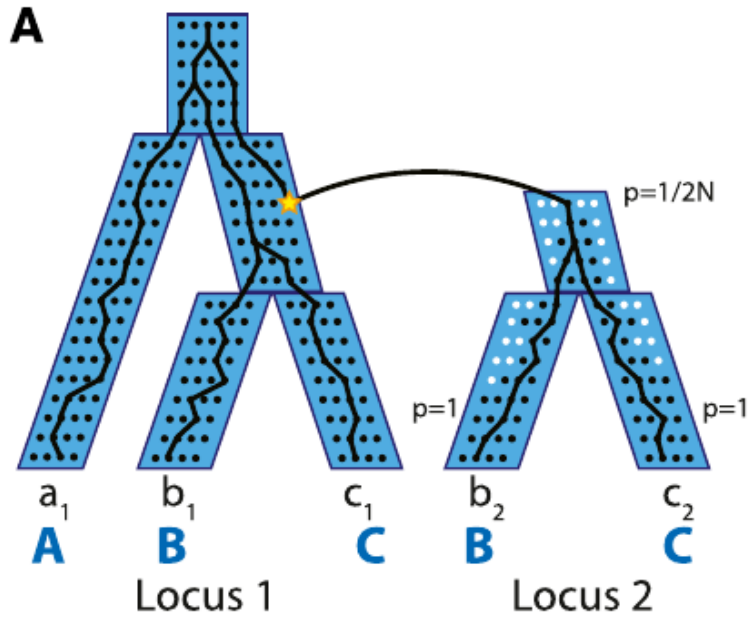


Figure 2A from Rasmussen and Kellis (2012)

Future: improved integration of DL models and coalescence

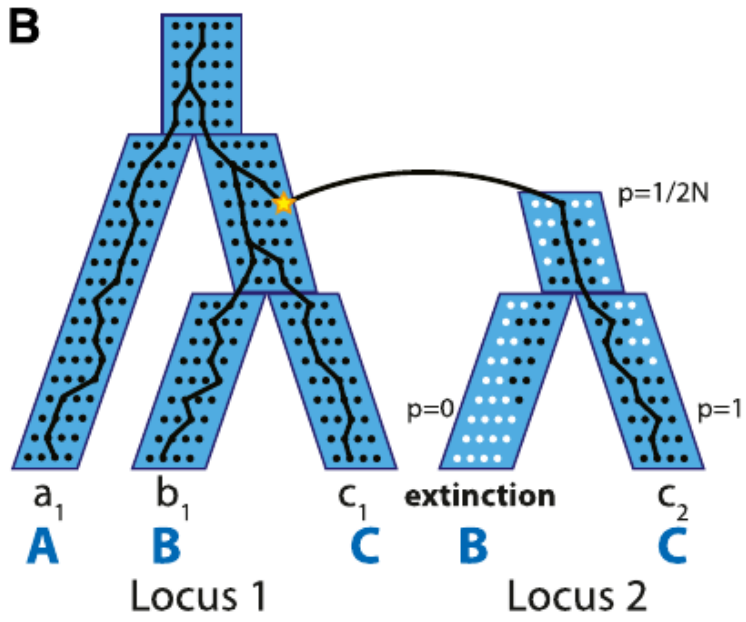


Figure 2B from Rasmussen and Kellis (2012)

Lateral Gene Transfer

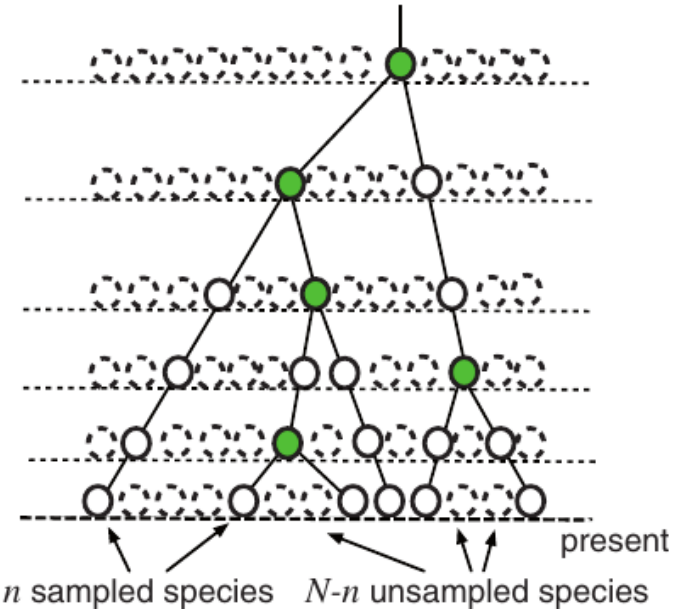


Figure 2c from Szöllősi et al. (2013)

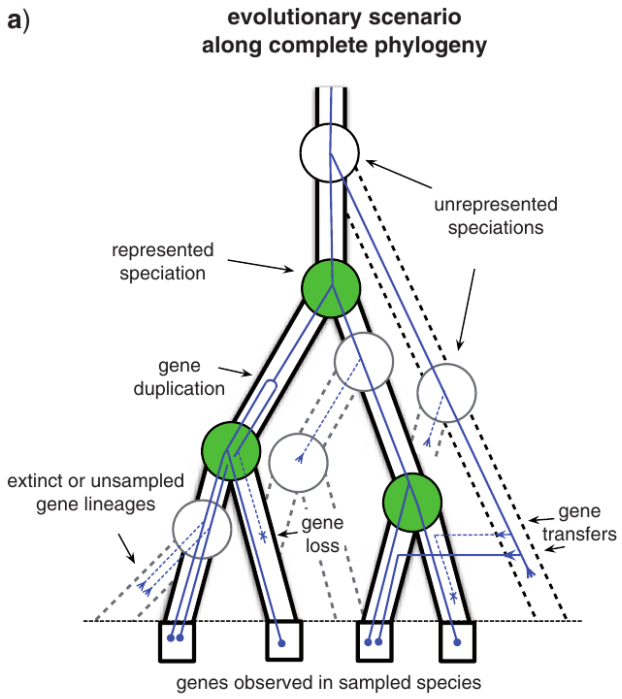


Figure 3 from Szöllősi et al. (2013)

They used 423 single-copy genes in ≥ 34 of 36 cyanobacteria

They estimate:

- 2.56 losses/family
- 2.15 transfers/family
- $\approx 28\%$ of transfers between non-overlapping branches

The main subject of this module: estimating a tree from sequence data

Tree construction:

- strictly algorithmic approaches - use a "recipe" to construct a tree
- optimality based approaches - choose a way to "score" a trees and then search for the tree that has the best score.

Expressing support for aspects of the tree:

- bootstrapping,
- testing competing trees against each other,
- posterior probabilities (in Bayesian approaches).

Optimality criteria

A rule for ranking trees (according to the data).
Each criterion produces a score.

Examples:

- Parsimony (Maximum Parsimony, MP)
- Maximum Likelihood (ML)
- Minimum Evolution (ME)
- Least Squares (LS)

Why doesn't simple clustering work?

Step 1: use sequences to estimate pairwise distances between taxa.

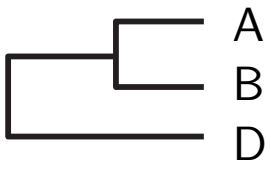
	A	B	C	D
A	-	0.2	0.5	0.4
B		-	0.46	0.4
C			-	0.7
D				-

Why doesn't simple clustering work?

	A	B	C	D
A	-	**0.2**	0.5	0.4
B		-	0.46	0.4
C			-	0.7
D				-

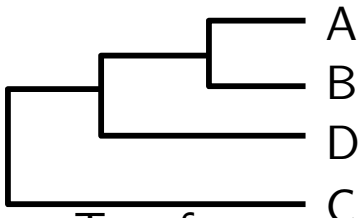
Why doesn't simple clustering work?

	A	B	C	D
A	-	0.2	0.5	**0.4**
B		-	0.46	**0.4**
C			-	0.7
D				-



Why doesn't simple clustering work?

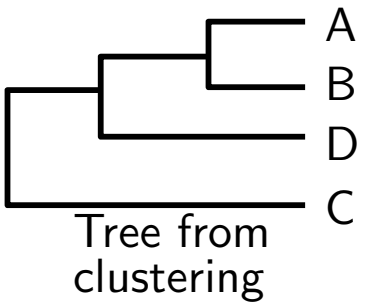
	A	B	C	D
A	-	0.2	0.5	**0.4**
B		-	0.46	**0.4**
C		-	-	**0.7**
D				0



Tree from clustering

Why doesn't simple clustering work?

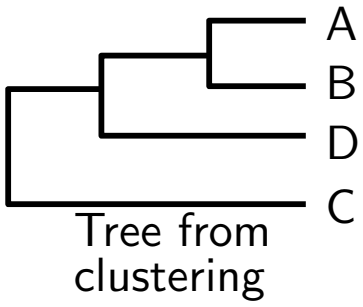
	A	B	C	D
A	0	0.2	**0.5**	0.4
B	0.2	0.2	**0.46**	0.4
C	**0.5**	**0.46**	0	**0.7**
D	0.4	0.4	**0.7**	0



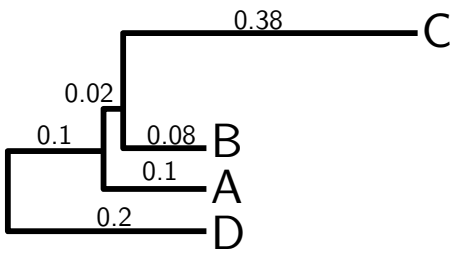
Tree from clustering

Why doesn't simple clustering work?

	A	B	C	D
A	0	0.2	0.5	0.4
B	0.2	0.	0.46	0.4
C	0.5	0.46	0	0.7
D	0.4	0.4	0.7	0



Tree from clustering



Tree with perfect fit

Why aren't the easy, obvious methods for generating trees good enough?

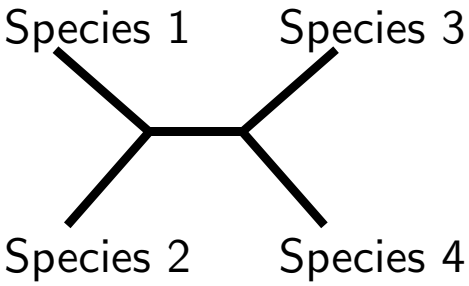
1. Simple clustering methods are sensitive to differences in the rate of sequence evolution (and this rate can be quite variable).

2. The "multiple hits" problem. When some sites in your data matrix are affected by more than 1 mutation, then the phylogenetic signal can be obscured. More on this later...

	1	2	3	4	5	6	7	8	9	.	.	.
Species 1	C	G	A	C	C	**A**	G	G	T	.	.	.
Species 2	C	G	A	C	C	**A**	G	G	T	.	.	.
Species 3	C	G	G	T	C	**C**	G	G	T	.	.	.
Species 4	C	G	G	C	C	**T**	G	G	T	.	.	.

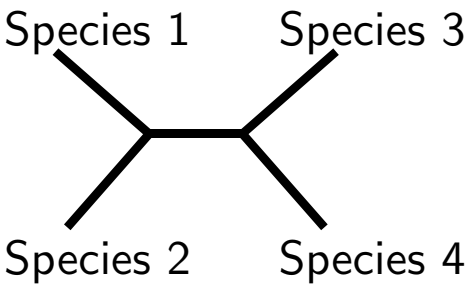
	1	2	3	4	5	6	7	8	9	.	.	.
Species 1	C	G	A	C	C	**A**	G	G	T	.	.	.
Species 2	C	G	A	C	C	**A**	G	G	T	.	.	.
Species 3	C	G	G	T	C	**C**	G	G	T	.	.	.
Species 4	C	G	G	C	C	**T**	G	G	T	.	.	.

One of the 3 possible trees:

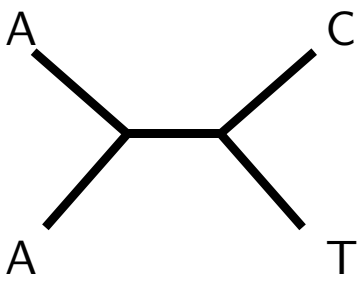


	1	2	3	4	5	6	7	8	9	.	.	.
Species 1	C	G	A	C	C	**A**	G	G	T	.	.	.
Species 2	C	G	A	C	C	**A**	G	G	T	.	.	.
Species 3	C	G	G	T	C	**C**	G	G	T	.	.	.
Species 4	C	G	G	C	C	**T**	G	G	T	.	.	.

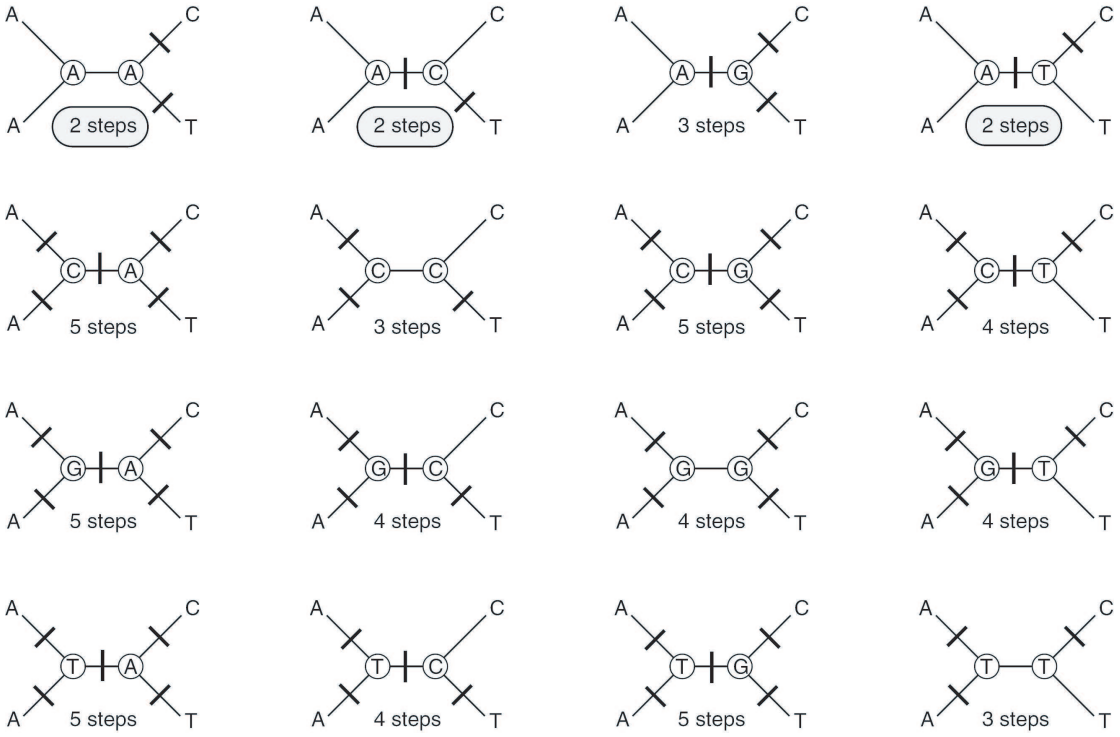
One of the 3 possible trees:



Same tree with states at character 6 instead of species names



Unordered Parsimony



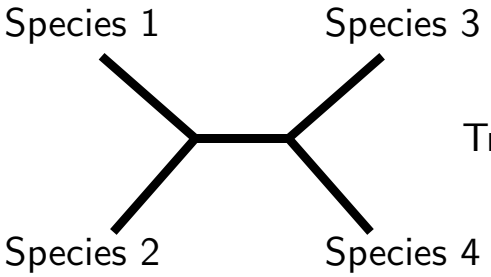
Things to note about the last slide

- 2 steps was the minimum score attainable.
- Multiple ancestral character state reconstructions gave a score of 2.
- Enumeration of all possible ancestral character states is **not** the most efficient algorithm.

Each character (site) is assumed to be independent

To calculate the parsimony score for a tree we simply sum the scores for every site.

	1	2	3	4	5	6	7	8	9
Species 1	C	G	**A**	C	C	A	G	G	T
Species 2	C	G	**A**	C	C	A	G	G	T
Species 3	C	G	**G**	T	C	C	G	G	T
Species 4	C	G	**G**	C	C	T	G	G	T
Score	0	0	**1**	1	0	2	0	0	0

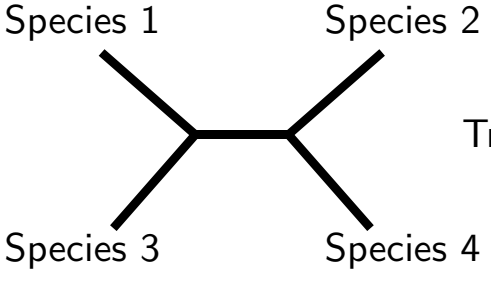


Tree 1 has a score of **4**

Considering a different tree

We can repeat the scoring for each tree.

	1	2	3	4	5	6	7	8	9
Species 1	C	G	**A**	C	C	A	G	G	T
Species 2	C	G	**A**	C	C	A	G	G	T
Species 3	C	G	**G**	T	C	C	G	G	T
Species 4	C	G	**G**	C	C	T	G	G	T
Score	0	0	**2**	1	0	2	0	0	0

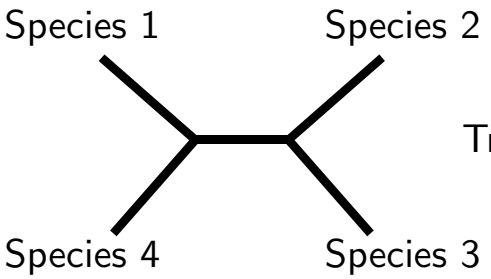


Tree 2 has a score of **5**

One more tree

Tree 3 has the same score as tree 2

	1	2	3	4	5	6	7	8	9
Species 1	C	G	**A**	C	C	A	G	G	T
Species 2	C	G	**A**	C	C	A	G	G	T
Species 3	C	G	**G**	T	C	C	G	G	T
Species 4	C	G	**G**	C	C	T	G	G	T
Score	0	0	**2**	1	0	2	0	0	0



Tree 3 has a score of **5**

Parsimony criterion prefers tree 1

Tree 1 required the *fewest* number of state changes (DNA substitutions) to explain the data.

Some parsimony advocates equate the preference for the fewest number of changes to the general scientific principle of preferring the simplest explanation (Ockham's Razor), but this connection has not been made in a rigorous manner.

Parsimony terms

- *homoplasy* multiple acquisitions of the same character state
  - parallelism, reversal, convergence
  - recognized by a tree requiring more than the minimum number of steps
  - minimum number of steps is the number of observed states minus 1

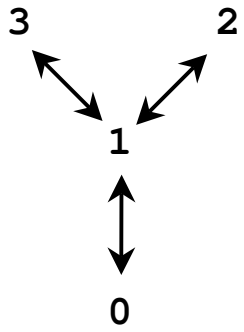
The parsimony criterion is equivalent to minimizing homoplasy.

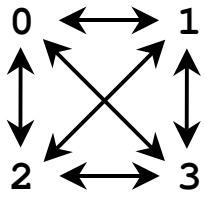
Homoplasy is one form of the multiple hits problem. In pop-gen terms, it is a violation of the infinite-alleles model.

In the example matrix at the beginning of these slides, only character 3 is parsimony informative.

	1	2	3	4	5	6	7	8	9
Species 1	C	G	**A**	C	C	A	G	G	T
Species 2	C	G	**A**	C	C	A	G	G	T
Species 3	C	G	**G**	T	C	C	G	G	T
Species 4	C	G	**G**	C	C	T	G	G	T
Max score	0	0	**2**	1	0	2	0	0	0
Min score	0	0	**1**	1	0	2	0	0	0

Assumptions about the evolutionary process can be incorporated using different step costs





Fitch Parsimony
"unordered"

Stepmatrices

Fitch Parsimony Stepmatrix

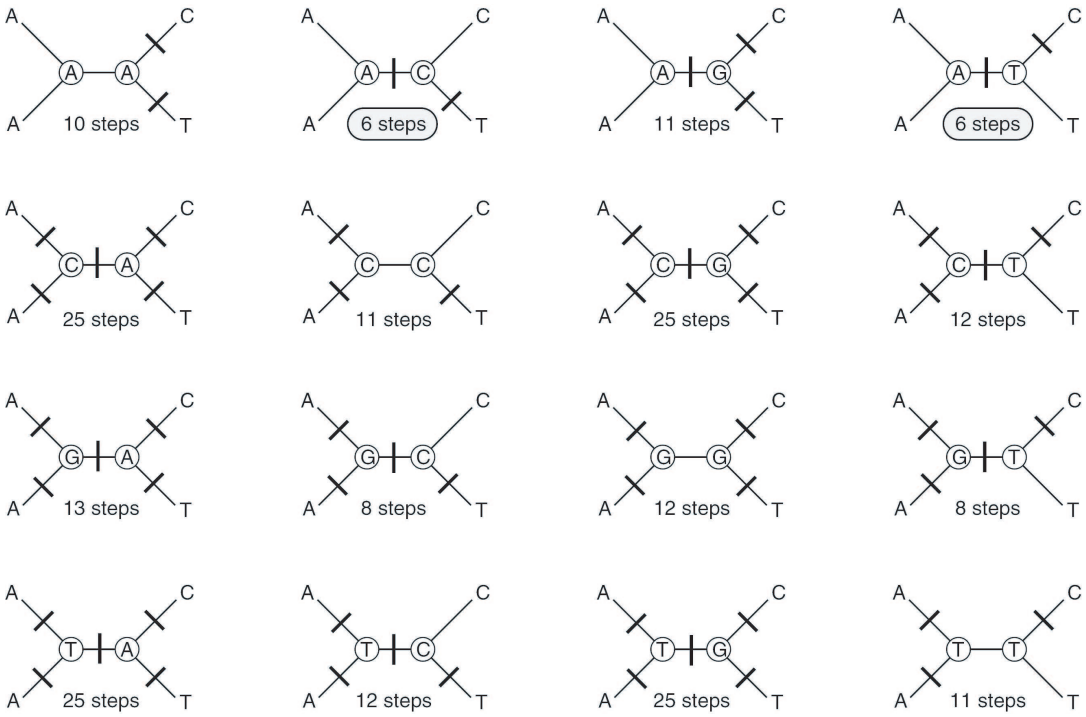
		To A	C	G	T
	A	0	1	1	1
From	C	1	0	1	1
	G	1	1	0	1
	T	1	1	1	0

Stepmatrices

Transversion-Transition 5:1 Stepmatrix

		To A	C	G	T
	A	0	5	1	5
From	C	5	0	5	1
	G	1	5	0	5
	T	5	1	5	0

5:1 Transversion:Transition parsimony



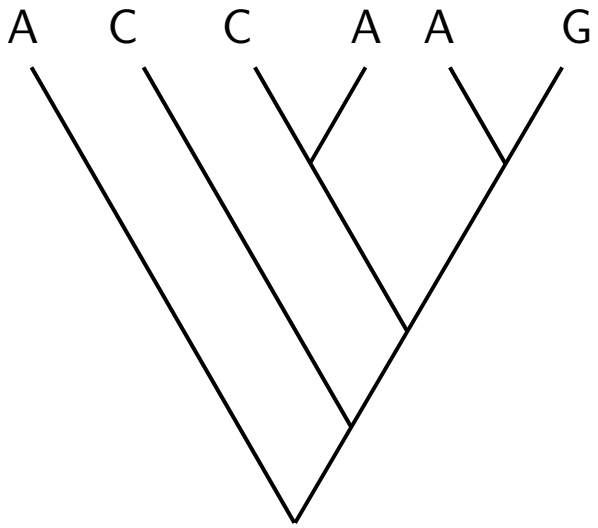
Stepmatrix considerations

- Parsimony scores from different stepmatrices cannot be meaningfully compared (31 under Fitch is not "better" than 45 under a transversion:transition stepmatrix)
- Parsimony cannot be used to infer the stepmatrix weights

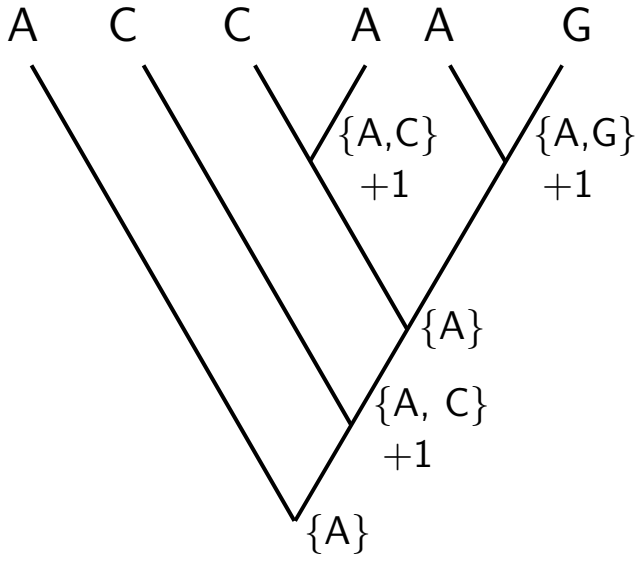
Other Parsimony variants

- *Dollo* derived state can only arise once, but reversals can be frequent (*e.g.* restriction enzyme sites).
- "weighted" - usually means that different characters are weighted differently (slower, more reliable characters usually given higher weights).
- implied weights Goloboff (1993)

Scoring trees under parsimony is fast



Scoring trees under parsimony is fast – Fitch algorithm



3 steps

Scoring trees under parsimony is fast

The "down-pass state sets" calculated in the Fitch algorithm can be stored at an internal node.

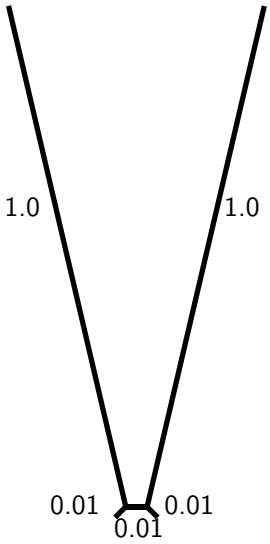
This lets you treat those internal nodes as pseudo-tips:

- avoid rescoring the entire tree if you make a small change, and
- break up the tree into smaller subtrees (Goloboff's sectorial searching).

Qualitative description of parsimony

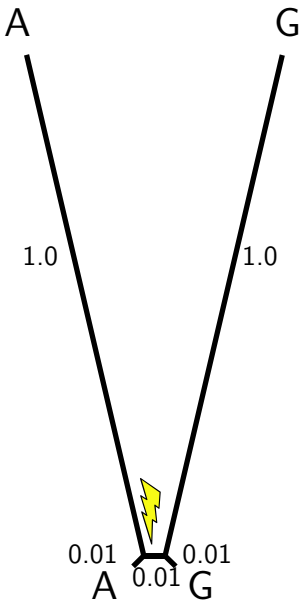
- Enables estimation of ancestral sequences.
- Even though parsimony always seeks to minimizes the number of changes, it can perform well even when changes are not rare.
- Does not "prefer" to put changes on one branch over another
- Hard to characterize statistically
  - the set of conditions in which parsimony is guaranteed to work well is very restrictive (low probability of change and not too much branch length heterogeneity);
  - Parsimony often performs well in simulation studies (even when outside the zones in which it is guaranteed to work);
  - Estimates of the tree can be extremely biased.

Long branch attraction



Felsenstein, J. 1978. Cases in which parsimony or compatibility methods will be positively misleading. *Systematic Zoology* **27**: 401-410.

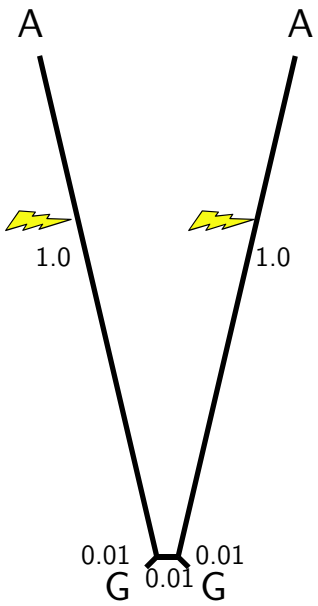
Long branch attraction



Felsenstein, J. 1978. Cases in which parsimony or compatibility methods will be positively misleading. *Systematic Zoology* **27**: 401-410.

The probability of a parsimony informative site due to inheritance is very low, (roughly 0.0003).

Long branch attraction



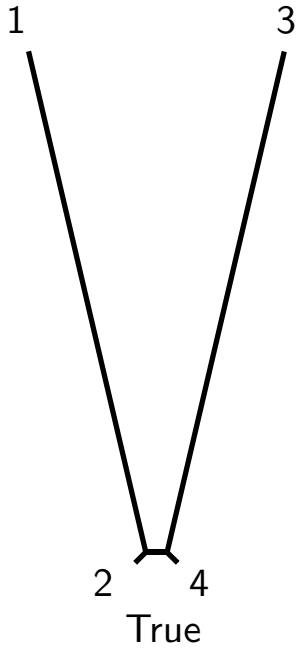
Felsenstein, J. 1978. Cases in which parsimony or compatibility methods will be positively misleading. *Systematic Zoology* **27**: 401-410.

The probability of a parsimony informative site due to inheritance is very low, (roughly 0.0003).

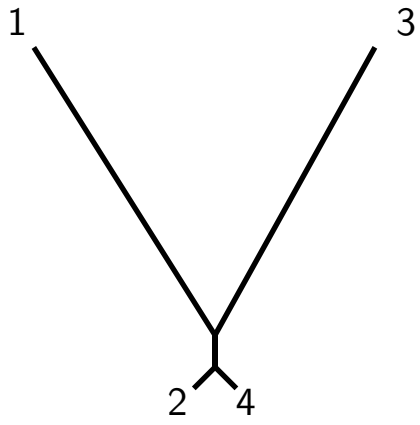
The probability of a misleading parsimony informative site due to parallelism is much higher (roughly 0.008).

Long branch attraction

Parsimony is almost guaranteed to get this tree wrong.



True



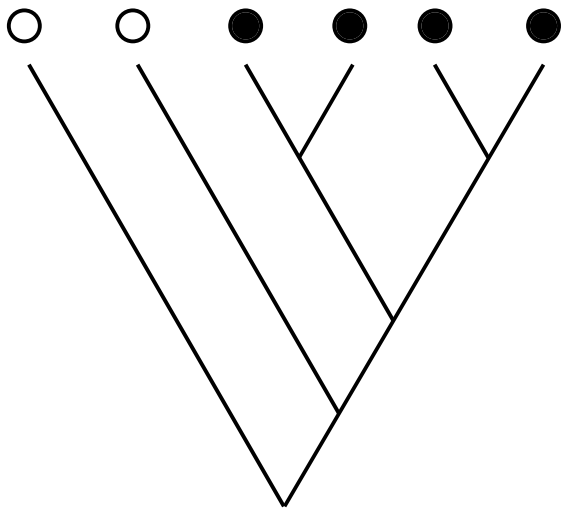
Inferred

Inconsistency

- Statistical Consistency (roughly speaking) is converging to the true answer as the amount of data goes to ∞.
- Parsimony based tree inference is *not* consistent for some tree shapes. In fact it can be "positively misleading":
  - "Felsenstein zone" tree
  - Many clocklike trees with short internal branch lengths and long terminal branches (Penny *et al.*, 1989, Huelsenbeck and Lander, 2003).
- Methods for assessing confidence (e.g. bootstrapping) will indicate that you should be very confident in the wrong answer.

Parsimony terms

- *synapomorphy* – a shared derived (newly acquired) character state. Evidence of monophletic groups.



Parsimony terms

- *parsimony informative* – a character with parsimony score variation across trees
  - *min* score $\neq$ *max* score
  - must be variable.
  - must have more than one *shared* state

Consistency Index (CI)

- minimum number of changes divided by the number required on the tree.
- CI=1 if there is no homoplasy
- negatively correlated with the number of species sampled

Retention Index (RI)

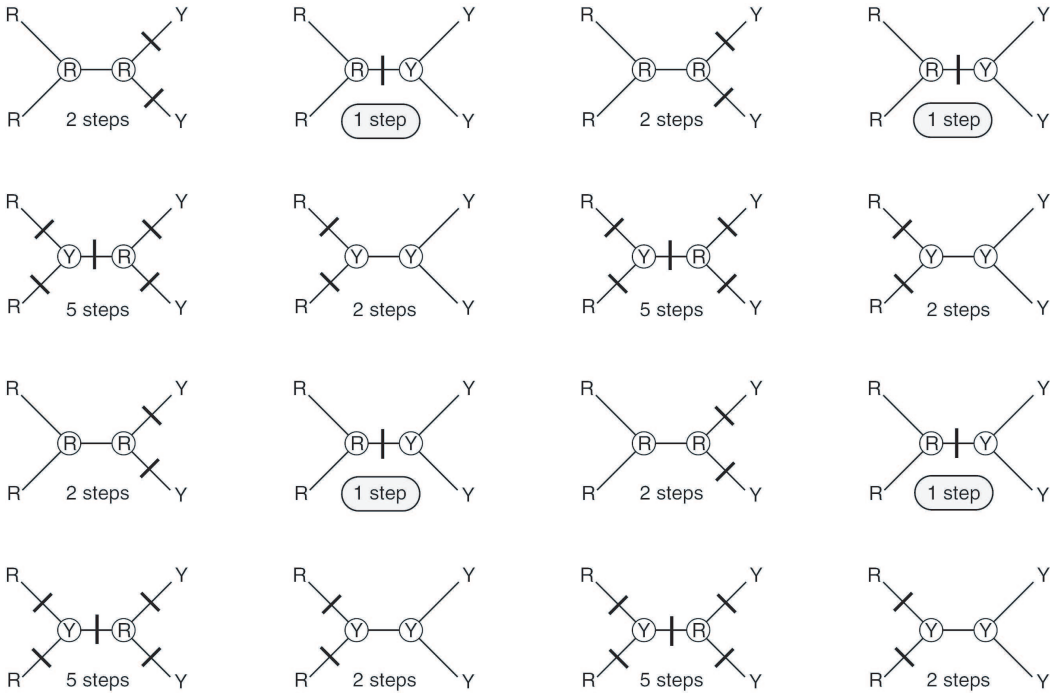
$$\text{RI} = \frac{\text{MaxSteps} - \text{ObsSteps}}{\text{MaxSteps} - \text{MinSteps}}$$

- defined to be 0 for parsimony uninformative characters
- RI=1 if the character fits perfectly
- RI=0 if the tree fits the character as poorly as possible

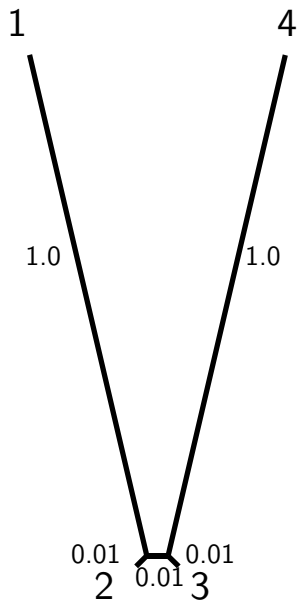
Transversion parsimony

- Transitions ($A \leftrightarrow G$, $C \leftrightarrow T$) occur more frequently than transversions (purine $\leftrightarrow$ pyrimidine)
- So, *homoplasy* involving transitions is much more common than transversions (*e.g.* $A \rightarrow G \rightarrow A$)
- Transversion parsimony (also called RY-coding) ignores all transitions

Transversion parsimony



Long branch attraction tree again



The probability of a parsimony informative site due to inheritance is very low, (roughly 0.0003).

The probability of a misleading parsimony informative site due to parallelism is much higher (roughly 0.008).

If the data is generated such that:

$$\Pr\begin{pmatrix} A \\ A \\ G \\ G \end{pmatrix} \approx 0.0003 \quad \text{and} \quad \Pr\begin{pmatrix} A \\ G \\ G \\ A \end{pmatrix} \approx 0.008$$

then how can we hope to infer the tree ((1,2),3,4) ?

Note: ((1,2),3,4) is referred to as Newick or New Hampshire notation for the tree.

You can read it by following the rules:

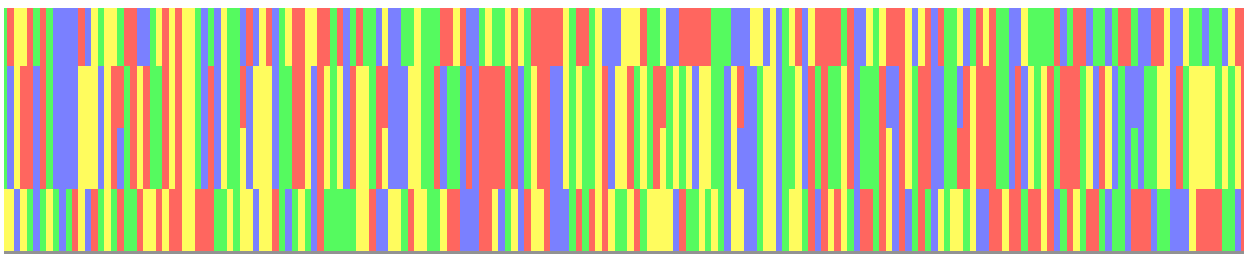
- start at a node,
- if the next symbol is '(' then add a child to the current node and move to this child,
- if the next symbol is a label, then label the node that you are at,
- if the next symbol is a comma, then move back to the current node's parent and add another child,
- if the next symbol is a ')', then move back to the current node's parent.

If the data is generated such that:

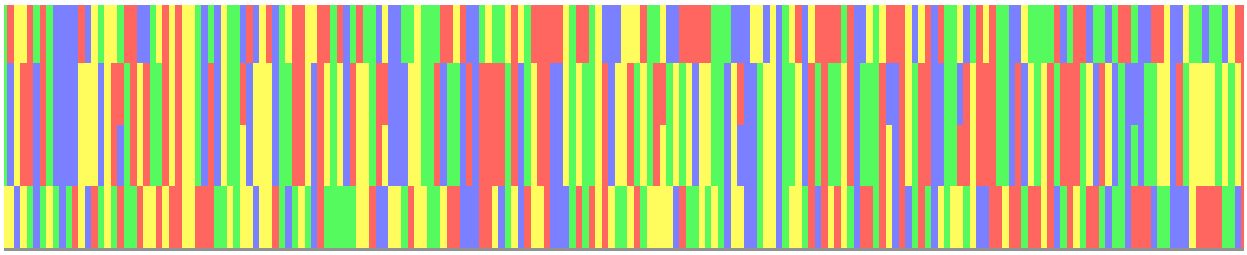
$$\Pr\begin{pmatrix} A \\ A \\ G \\ G \end{pmatrix} \approx 0.0003 \quad \text{and} \quad \Pr\begin{pmatrix} A \\ G \\ G \\ A \end{pmatrix} \approx 0.008$$

then how can we hope to infer the tree ((1,2),3,4) ?

Looking at the data in "bird's eye" view (using Mesquite):



Looking at the data in "bird's eye" view (using Mesquite):



We see that sequences 1 and 4 are clearly very different.
Perhaps we can estimate the tree if we use the branch length information from the sequences...

Distance-based approaches to inferring trees

- Convert the raw data (sequences) to a pairwise distances
- Try to find a tree that explains these distances.
- *Not* simply clustering the most similar sequences.

	1	2	3	4	5	6	7	8	9	10
Species 1	C	G	A	C	C	A	G	G	T	A
Species 2	C	G	A	C	C	A	G	G	T	A
Species 3	C	G	G	T	C	C	G	G	T	A
Species 4	C	G	G	C	C	A	T	G	T	A

Can be converted to a distance matrix:

	Species 1	Species 2	Species 3	Species 4
Species 1	0	0	0.3	0.2
Species 2	0	0	0.3	0.2
Species 3	0.3	0.3	0	0.3
Species 4	0.2	0.2	0.3	0

Note that the distance matrix is symmetric.

	Species 1	Species 2	Species 3	Species 4
Species 1	0	0	0.3	0.2
Species 2	0	0	0.3	0.2
Species 3	0.3	0.3	0	0.3
Species 4	0.2	0.2	0.3	0

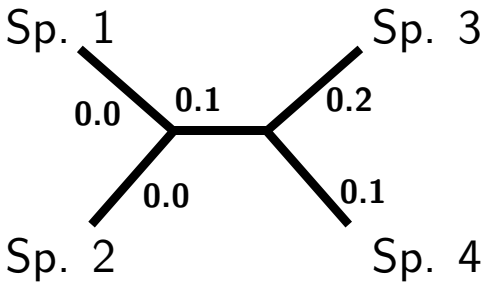
. . . so we can just use the lower triangle.

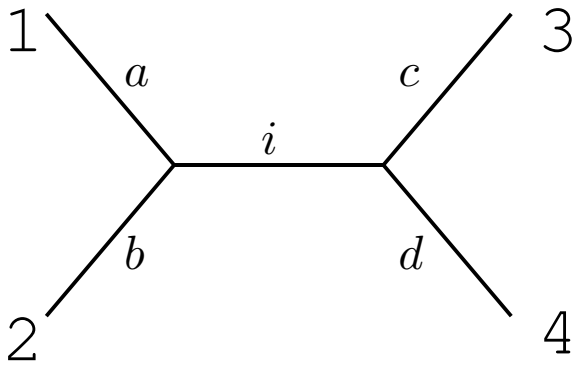
	Species 1	Species 2	Species 3
Species 2	0		
Species 3	0.3	0.3	
Species 4	0.2	0.2	0.3

Can we find a tree that would predict these observed character divergences?

	Species 1	Species 2	Species 3
Species 2	0		
Species 3	0.3	0.3	
Species 4	0.2	0.2	0.3

Can we find a tree that would predict these observed character divergences?





parameters

$p_{12} = a + b$
$p_{13} = a + i + c$
$p_{14} = a + i + d$
$p_{23} = b + i + c$
$p_{23} = b + i + d$
$p_{34} = c + d$

data

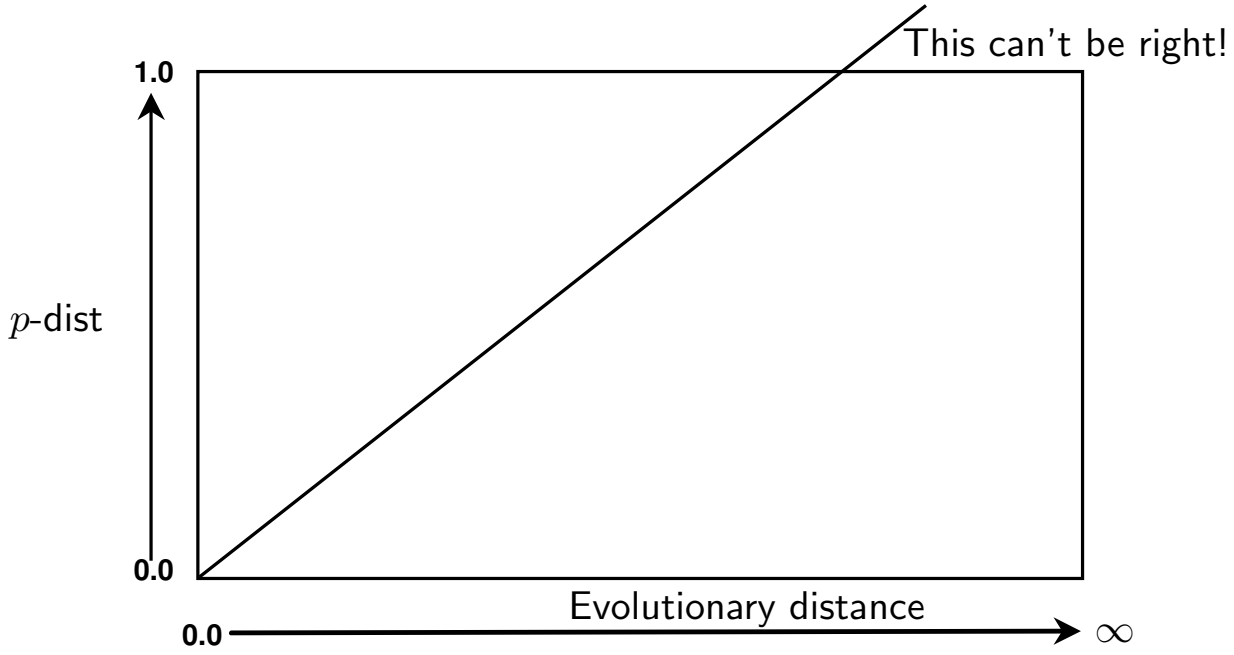
	1	2	3
2	d_{12}		
3	d_{13}	d_{23}	
4	d_{14}	d_{24}	d_{34}

If our pairwise distance measurements were error-free estimates of the *evolutionary distance* between the sequences, then we could always infer the tree from the distances.

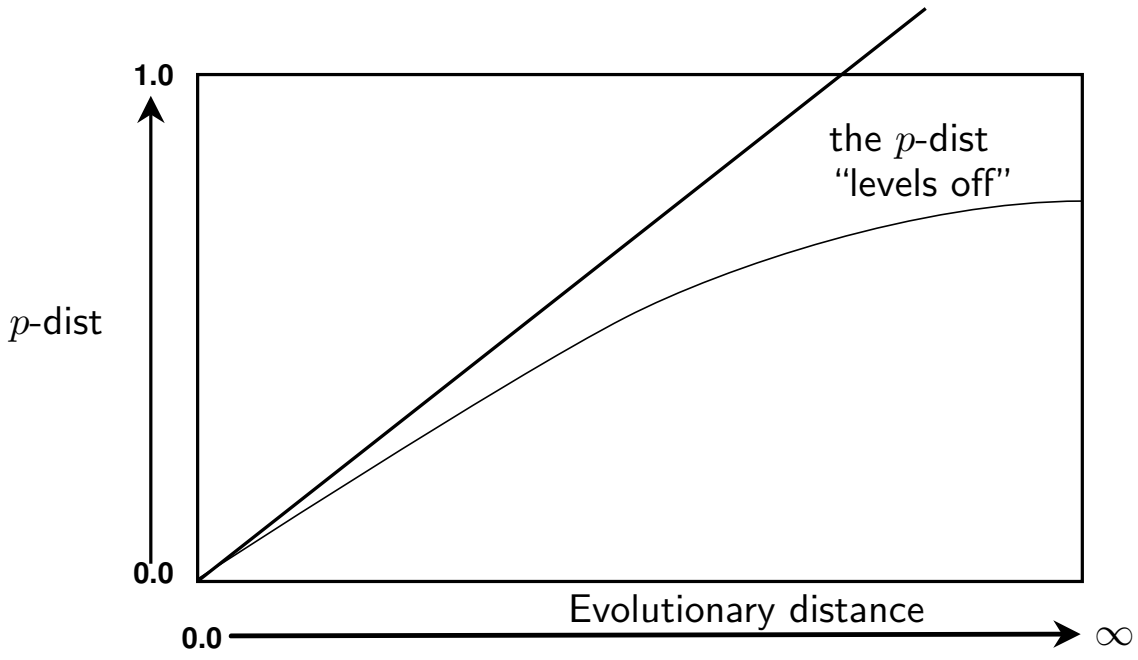
The evolutionary distance is the number of mutations that have occurred along the path that connects two tips.

We hope the distances that we measure can produce good estimates of the evolutionary distance, but we know that they cannot be perfect.

Intuition of sequence divergence vs evolutionary distance

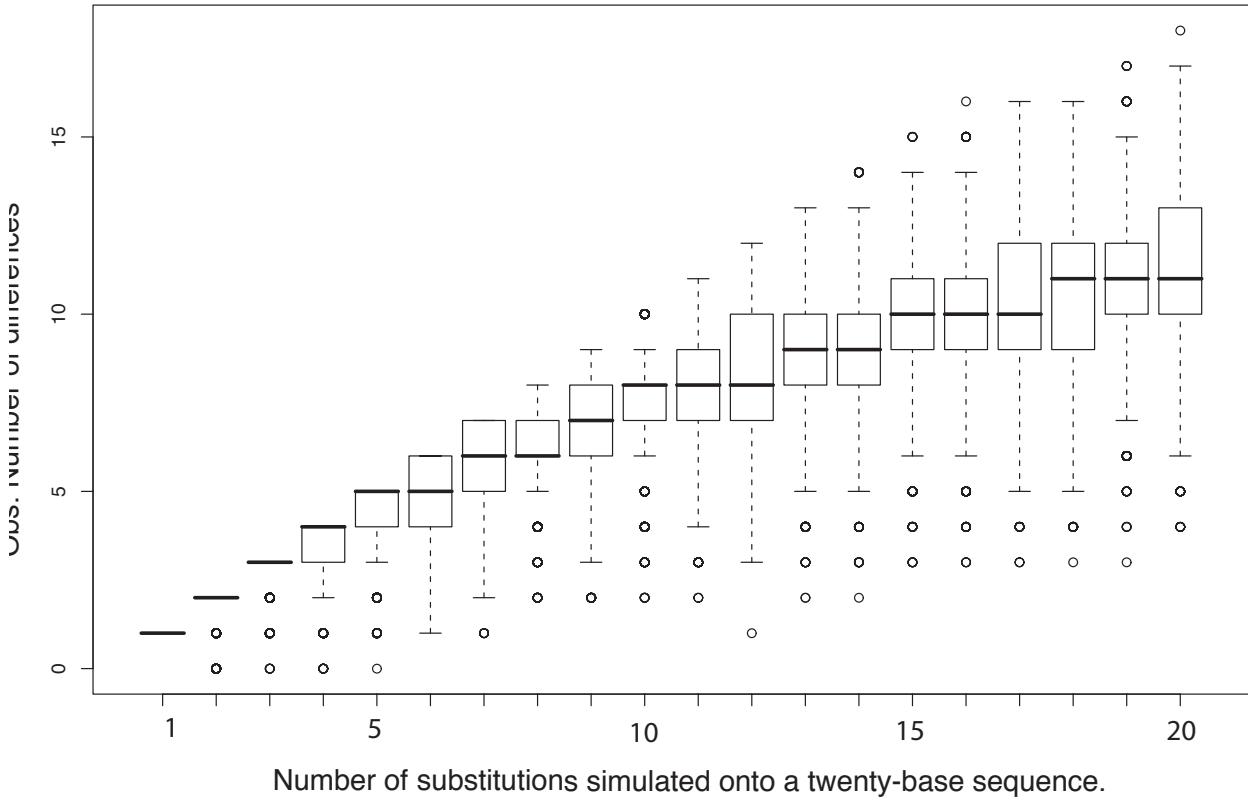


Sequence divergence vs evolutionary distance



"Multiple hits" problem (also known as saturation)

- Levelling off of sequence divergence vs time plot is caused by multiple substitutions affecting the same site in the DNA.
- At large distances the "raw" sequence divergence (also known as the p-distance or Hamming distance) is a poor estimate of the true evolutionary distance.
- Statistical models must be used to correct for unobservable substitutions (much more on these models tomorrow!)
- Large p-distances respond more to model-based correction – and there is a larger error associated with the correction.



Distance corrections

- applied to distances before tree estimation,
- converts raw distances to an estimate of the evolutionary distance

$$d = -\frac{3}{4} \ln\left(1 - \frac{4c}{3}\right)$$

"raw" p-distances

	1	2	3
2	c_{12}		
3	c_{13}	c_{23}	
4	c_{14}	c_{24}	c_{34}

corrected distances

	1	2	3
2	d_{12}		
3	d_{13}	d_{23}	
4	d_{14}	d_{24}	d_{34}

$$d = -\frac{3}{4} \ln\left(1 - \frac{4c}{3}\right)$$

"raw" p-distances

	1	2	3
2	0.0		
3	0.3	0.3	
4	0.2	0.2	0.3

corrected distances

	1	2	3
2	0		
3	0.383	0.383	
4	0.233	0.233	0.383

Least Squares Branch Lengths

$$\text{Sum of Squares} = \sum_i \sum_j \frac{(p_{ij} - d_{ij})^2}{\sigma_{ij}^k}$$

- minimize discrepancy between path lengths and observed distances
- σ_{ij}^k is used to "downweight" distance estimates with high variance

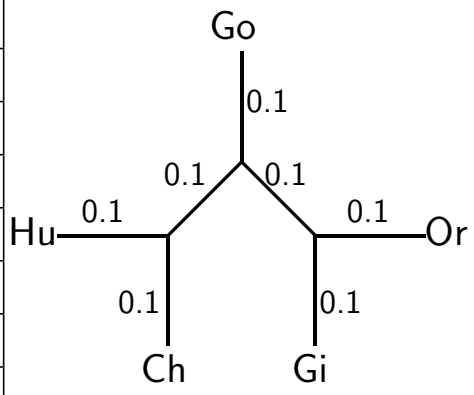
Least Squares Branch Lengths

$$\text{Sum of Squares} = \sum_i \sum_j \frac{(p_{ij} - d_{ij})^2}{\sigma_{ij}^k}$$

- in unweighted least-squares (Cavalli-Sforza & Edwards, 1967): $k = 0$
- in the method Fitch-Margoliash (1967): $k = 2$ and $\sigma_{ij} = d_{ij}$

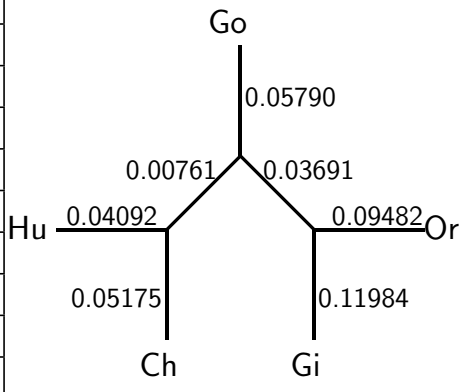
Poor fit using arbitrary branch lengths

Species	d_{ij}	p_{ij}	$(p-d)^2$
Hu-Ch	0.09267	0.2	0.01152
Hu-Go	0.10928	0.3	0.03637
Hu-Or	0.17848	0.4	0.04907
Hu-Gi	0.20420	0.4	0.03834
Ch-Go	0.11440	0.3	0.03445
Ch-Or	0.19413	0.4	0.04238
Ch-Gi	0.21591	0.4	0.03389
Go-Or	0.18836	0.3	0.01246
Go-Gi	0.21592	0.3	0.00707
Or-Gi	0.21466	0.2	0.00021
		S.S.	**0.26577**



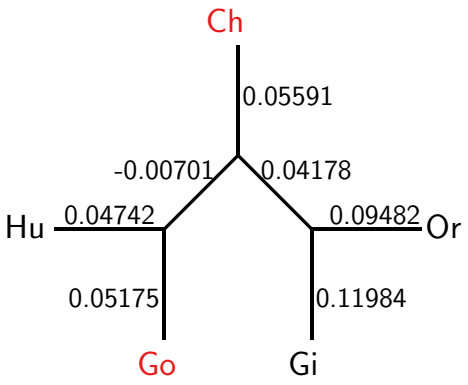
Optimizing branch lengths yields the least-squares score

Species	d_{ij}	p_{ij}	$(p-d)^2$
Hu-Ch	0.09267	0.09267	0.000000000
Hu-Go	0.10928	0.10643	0.000008123
Hu-Or	0.17848	0.18026	0.000003168
Hu-Gi	0.20420	0.20528	0.000001166
Ch-Go	0.11440	0.11726	0.000008180
Ch-Or	0.19413	0.19109	0.000009242
Ch-Gi	0.21591	0.21611	0.000000040
Go-Or	0.18836	0.18963	0.000001613
Go-Gi	0.21592	0.21465	0.000001613
Or-Gi	0.21466	0.21466	0.000000000
		S.S.	**0.000033144**

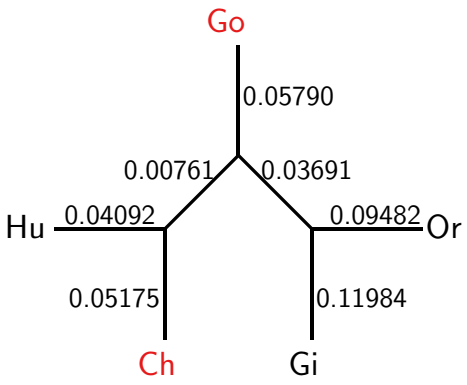


Least squares as an optimality criterion

SS = 0.00034

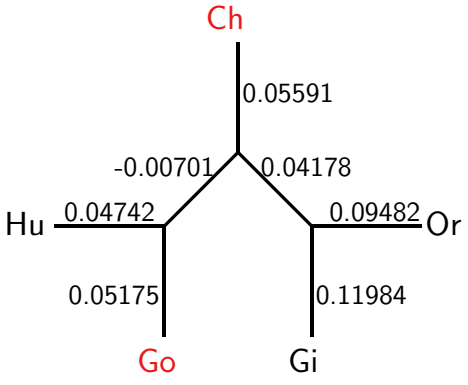


SS = 0.0003314
(best tree)

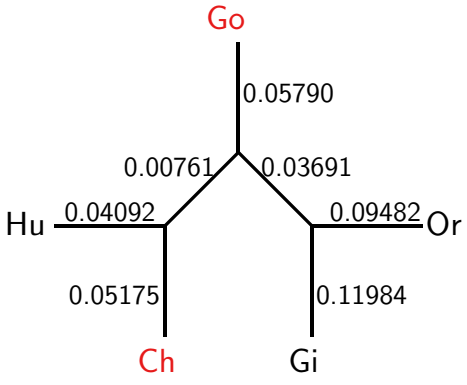


Minimum evolution optimality criterion

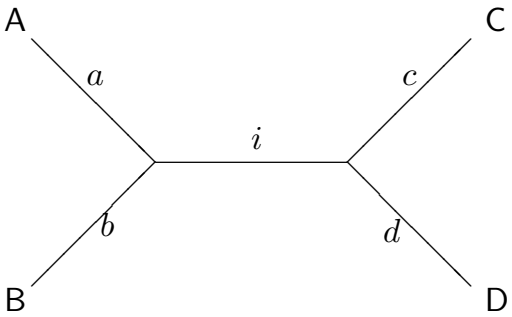
Sum of branch lengths
=0.41152



Sum of branch lengths
=0.40975
(best tree)



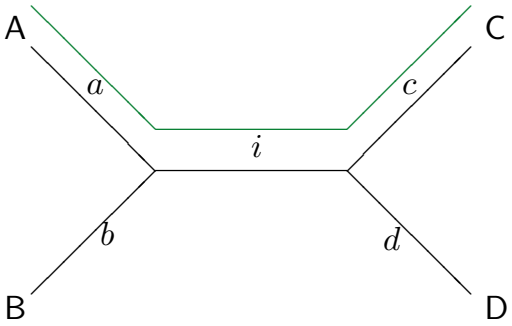
We still use least squares branch lengths when we use Minimum Evolutior



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

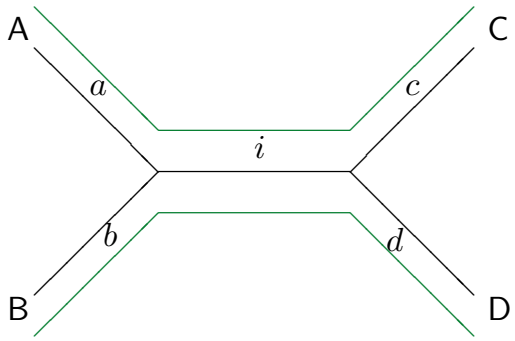
If the tree above is correct then:

$$
\begin{aligned}
p_{AB} &= a + b \\
p_{AC} &= a + i + c \\
p_{AD} &= a + i + d \\
p_{BC} &= b + i + c \\
p_{BD} &= b + i + d \\
p_{CD} &= c + d
\end{aligned}
$$



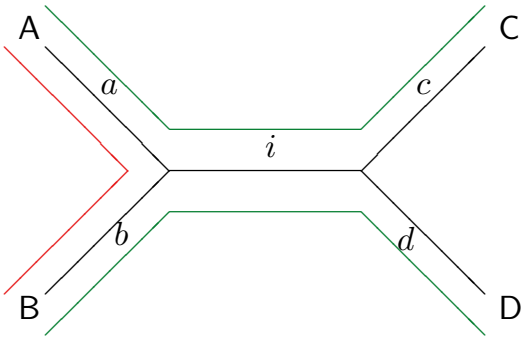
	A	B	C
B	d_{AB}		
C	d_{AC}	d_{BC}	
D	d_{AD}	d_{BD}	d_{CD}

d_{AC}



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

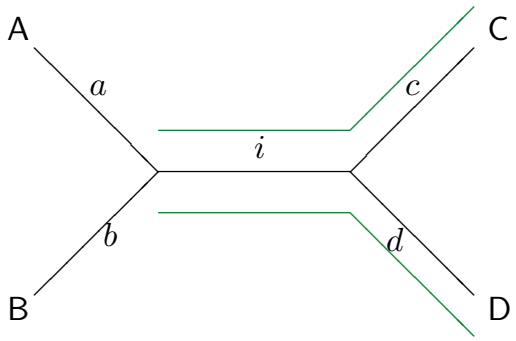
$d_{AC} + d_{BD}$



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

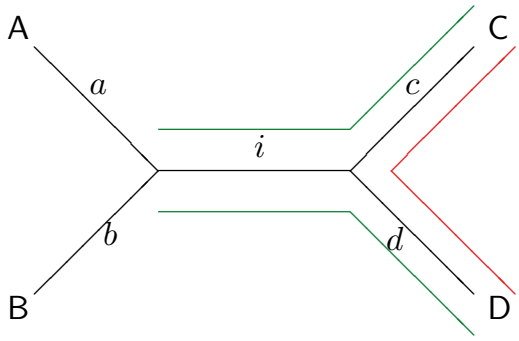
$d_{AC} + d_{BD}$

d_{AB}



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

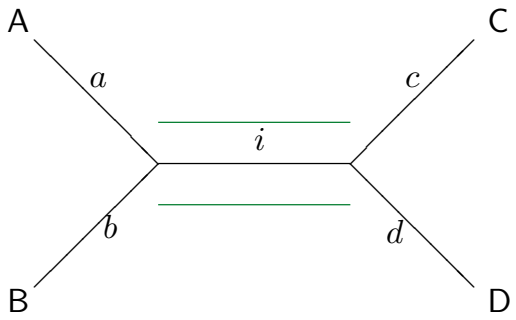
$$d_{AC} + d_{BD} - d_{AB}$$



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

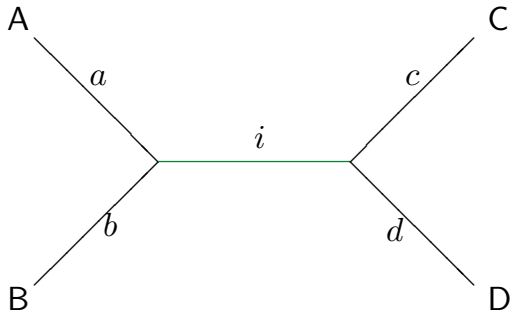
$$d_{AC} + d_{BD} - d_{AB}$$

$$d_{CD}$$



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

$$d_{AC} + d_{BD} - d_{AB} - d_{CD}$$



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

$$i^{\dagger} = \frac{d_{AC}+d_{BD}-d_{AB}-d_{CD}}{2}$$

Note that our estimate

$$i^{\dagger} = \frac{d_{AC} + d_{BD} - d_{AB} - d_{CD}}{2}$$

does not use all of our data. d_{BC} and d_{AD} are ignored!

We could have used $d_{BC} + d_{AD}$ instead of $d_{AC} + d_{BD}$ (you can see this by going through the previous slides after rotating the internal branch).

$$i^{*} = \frac{d_{BC} + d_{AD} - d_{AB} - d_{CD}}{2}$$

A better estimate than either i or i^{*} would be the average of both of them:

$$i' = \frac{d_{BC} + d_{AD} + d_{AC} + d_{BD}}{2} - d_{AB} - d_{CD}$$

This logic has been extend to trees of more than 4 taxa by Pauplin (2000) and Semple and Steel (2004).

Balanced minimum evolution

Desper and Gascuel (2002, 2004) refer to fitting the branch lengths using the estimators of Pauplin (2000) and preferring the tree with the smallest tree length "Balanced Minimum Evolution."

They that it is equivalent to a form of weighted least squares in which distances are down-weighted by an exponential function of the topological distances between the leaves.

Desper and Gascuel (2005) showed that neighbor-joining is star decomposition (more on this later) under BME. See Gascuel and Steel (2006)

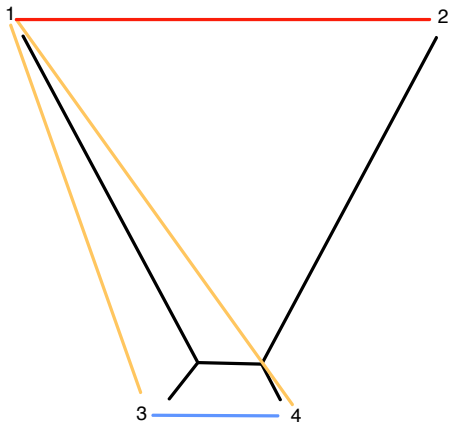
FastME

Software by Desper and Gascuel (2004) which implements searching under the balanced minimum evolution criterion.

It is extremely fast and is more accurate than neighbor-joining (based on simulation studies).

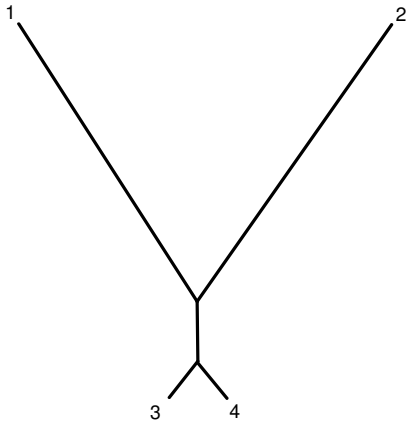
Failure to correct distance sufficiently leads to poor performance

"Under-correcting" will underestimate long evolutionary distances more than short distances



Failure to correct distance sufficiently leads to poor performance

The result is the classic "long-branch attraction" phenomenon.

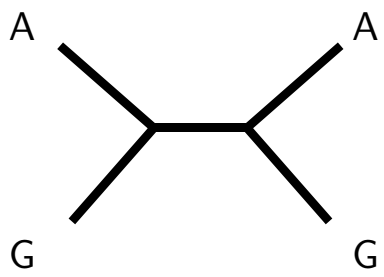


Distance methods: pros

- Fast – the new FastTree method Price et al. (2009) can calculate a tree in less time than it takes to calculate a full distance matrix!
- Can use models to correct for unobserved differences
- Works well for closely related sequences
- Works well for clock-like sequences

Distance methods: cons

- Do not use all of the information in sequences
- Do not reconstruct character histories, so they not enforce all logical constraints



References

Boussau, B., Szöllősi, G. J., Duret, L., Gouy, M., Tannier, E., and Daubin, V. (2013). Genome-scale coestimation of species and gene trees. *Genome Research*, 23(2):323–330.

Desper, R. and Gascuel, O. (2002). Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. *Journal of Computational Biology*, 9(5):687–705.

Desper, R. and Gascuel, O. (2004). Theoretical foundation of the balanced minimum evolution method of phylogenetic inference and its relationship to weighted least-squares tree fitting. *Molecular Biology and Evolution*.

Desper, R. and Gascuel, O. (2005). The minimum evolution distance-based approach to phylogenetic inference. In Gascuel, O., editor, *Mathematics of Evolution and Phylogeny*, pages 1–32. Oxford University Press.

Gascuel, O. and Steel, M. (2006). Neighbor-joining revealed. *Molecular Biology and Evolution*, 23(11):1997–2000.

Goloboff, P. (1993). Estimating character weights during tree search. *Cladistics*, 9(1):83–91.

Heled, J. and Drummond, A. (2010). Bayesian inference of species trees from multilocus data. *Molecular Biology and Evolution*, 27(3):570–580.

Pauplin, Y. (2000). Direct calculation of a tree length using a distance matrix. *Journal of Molecular Evolution*, 2000(51):41–47.

Price, M. N., Dehal, P., and Arkin, A. P. (2009). FastTree: Computing large minimum-evolution trees with profiles instead of a distance matrix. *Molecular Biology and Evolution*, 26(7):1641–1650.

Rasmussen, M. D. and Kellis, M. (2012). Unified modeling of gene duplication, loss, and coalescence using a locus tree. *Genome Research*, 22(4):755–765.

Semple, C. and Steel, M. (2004). Cyclic permutations and evolutionary trees. *Advances in Applied Mathematics*, 32(4):669–680.

Szöllősi, G. J., Tannier, E., Lartillot, N., and Daubin, V. (2013). Lateral gene transfer from the dead. *Systematic Biology*.

Tree Searching

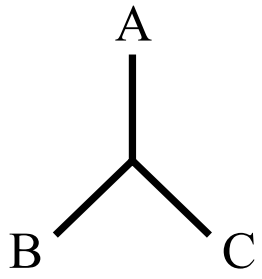
We've discussed how we rank trees

- Parsimony
- Least squares
- Minimum evolution
- Balanced minimum evolution
- Maximum likelihood (later in the course)

So we have ways of deciding what a good tree is when we see one, but . . .

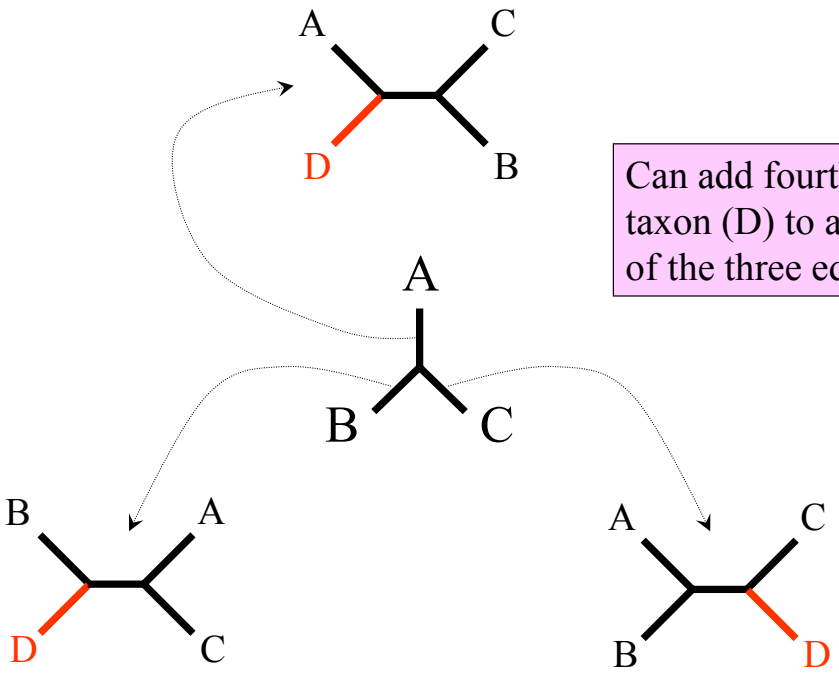
How do we find the best tree?
(or one that is good enough)

Exhaustive Enumeration



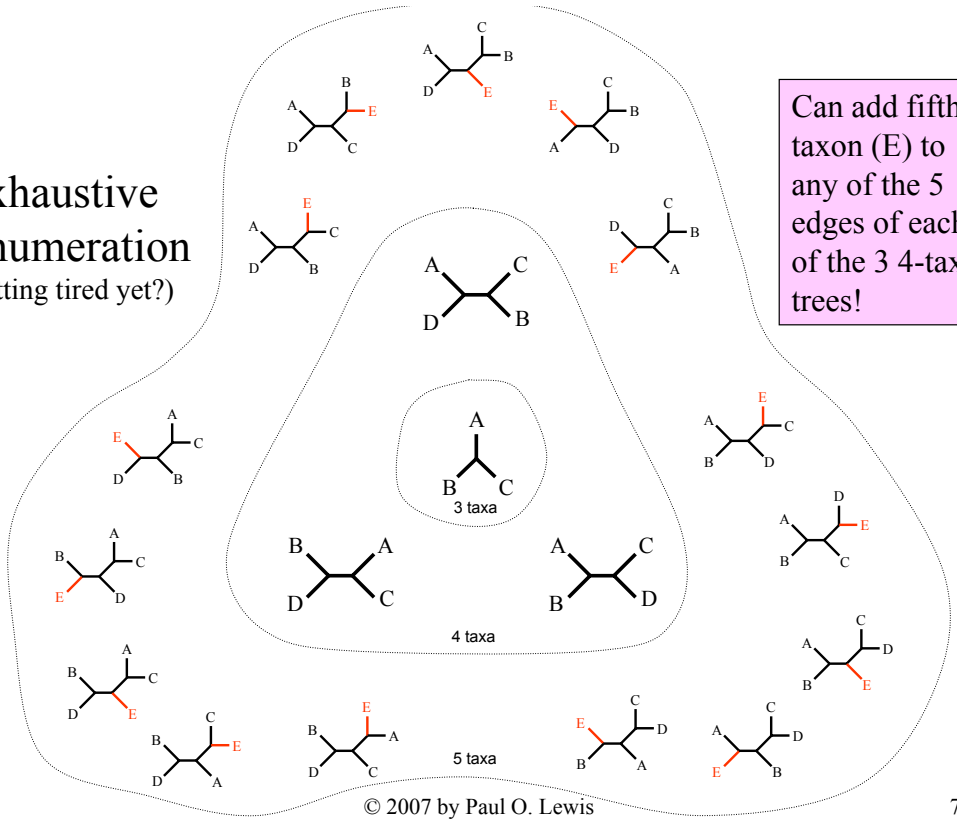
With the first three taxa, create the trivial unrooted tree

Exhaustive Enumeration...



Can add fourth taxon (D) to any of the three edges

Exhaustive Enumeration
(getting tired yet?)



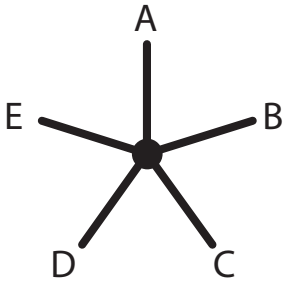
Can add fifth taxon (E) to any of the 5 edges of each of the 3 4-taxon trees!

Tips	Number of unrooted (binary) trees	
4	3	
5	15	
6	105	
7	945	
8	10,395	
9	135,135	
10	2,027,025	
11	34,459,425	
12	654,729,075	
13	13,749,310,575	
14	316,234,143,225	
15	7,905,853,580,625	
16	213,458,046,676,875	
17	6,190,283,353,629,375	
18	191,898,783,962,510,625	
19	6,332,659,870,762,850,625	
20	22,164,309,5476,699,771,875	
21	8,200,794,532,637,891,559,375	
22	319,830,986,772,877,770,815,625	
23	13,113,070,457,687,988,603,440,625	> 21 moles of trees
24	563,862,029,680,583,509,947,946,875	

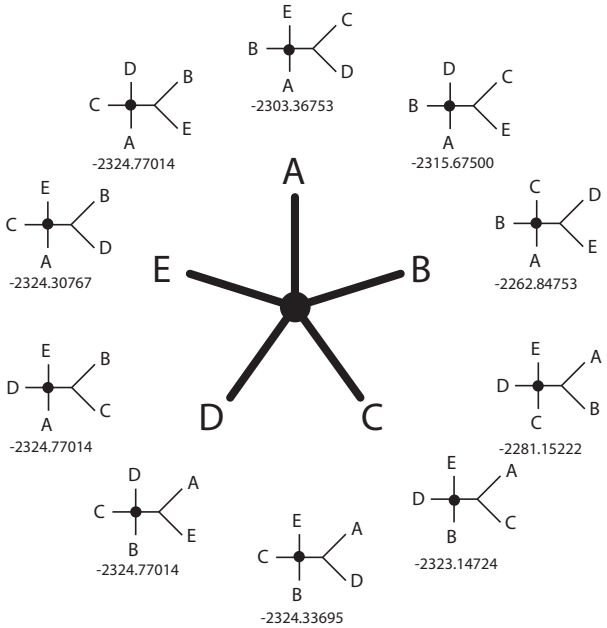
For N taxa:

$$
\begin{aligned}
\#\text{ unrooted, binary trees} &= \prod_{i=3}^{N-1}(2i-3) \\
&= \prod_{i=4}^{N}(2i-5) \\
\#\text{ rooted, binary trees} &= \prod_{i=3}^{N}(2i-3) \\
&= (2N-3)(\#\text{ unrooted, binary trees})
\end{aligned}
$$

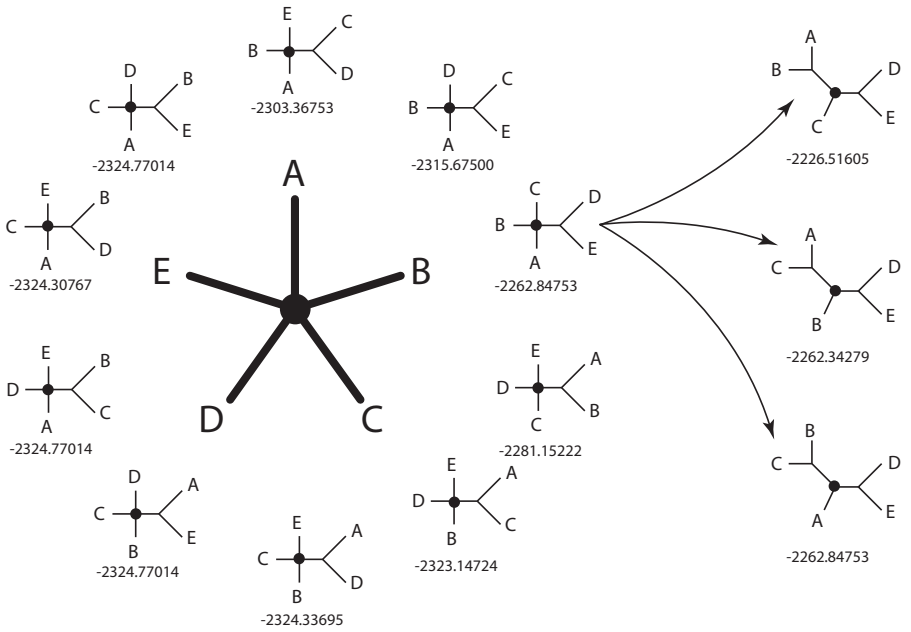
Star decomposition



Star decomposition



Star decomposition

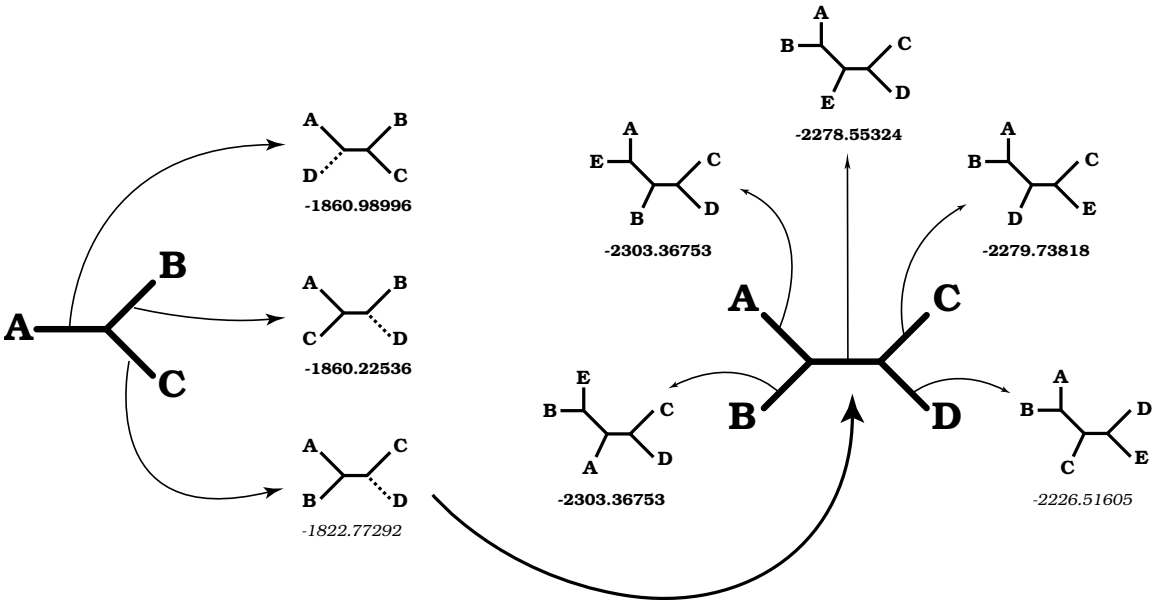


Star decomposition

- Very "greedy" – it makes the best decision at each step, but does not try to "plan ahead". Once a pair of species are joined, they will not be separated.

- Neighbor-joining (Saitou and Nei, 1987) is star decomposition under the balanced minimum evolution criterion

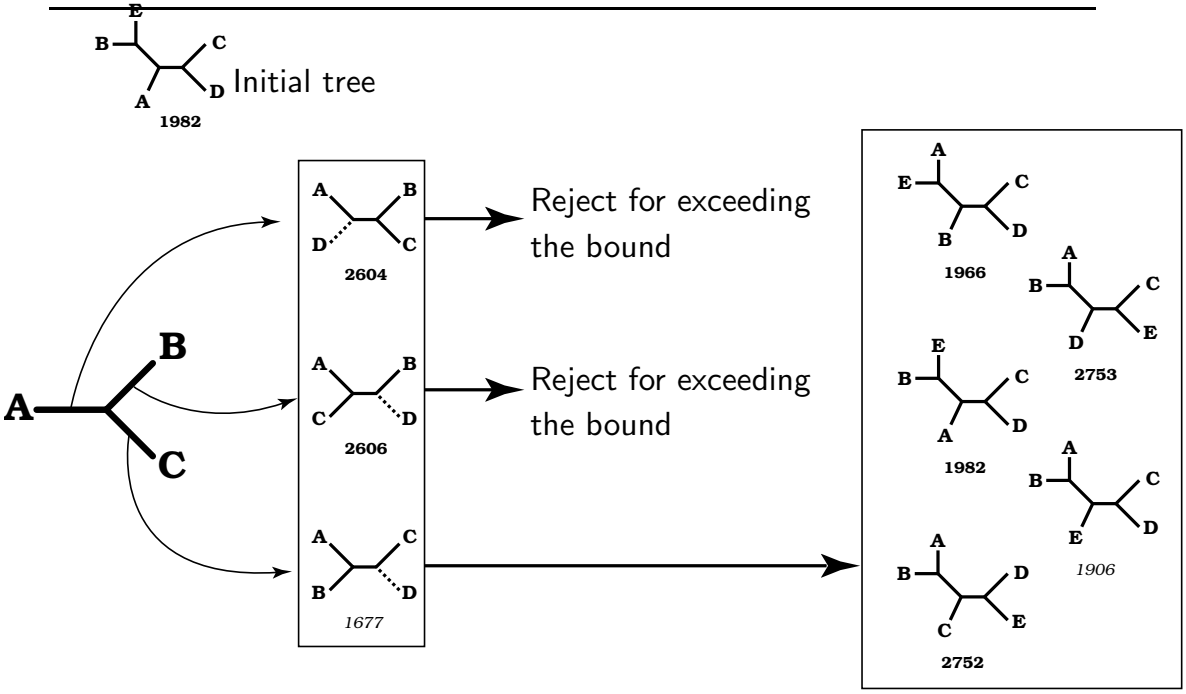
Stepwise addition



Stepwise addition

- Order-dependent (multiple random orderings can be used to give a range of starting trees for more thorough searches).
- Taxa joined initially may have intervening species added, but still fairly greedy.

Branch and bound



Branch and bound

- Guaranteed to return the best tree(s)
- Typically only a viable option for < 30 species (depends on how clean the data is)

Trying to improve a tree

Neither stepwise addition nor star decomposition is guaranteed to return the best tree(s), but branch-and-bound (or exhaustive searching) is frequently infeasible.

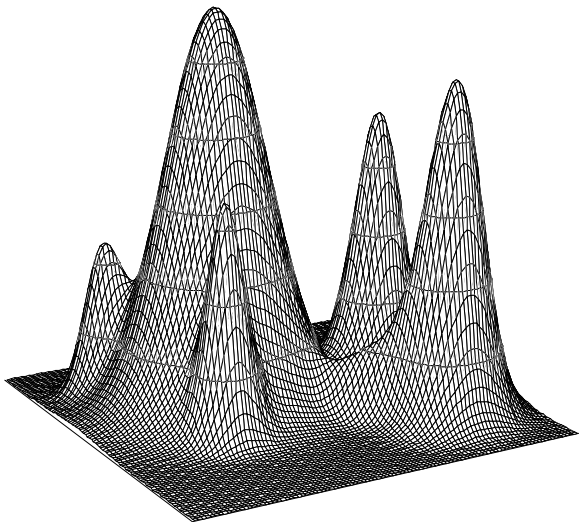
Heuristic hill-climbing searches can work quite well:

1. Start with a tree
2. Score the tree
3. Consider a new tree within the neighborhood of the current tree:
   (a) Score the new tree.
   (b) If the new tree has a better tree, use it as the "current tree"
   (c) Stop if there are no other trees within the neighborhood to consider.

These are **not** guaranteed to find even one of the optimal trees.

The most common way to explore the neighborhood of a tree is to swap the branches of the tree to construct similar trees.

Heuristics explore "Tree Space"

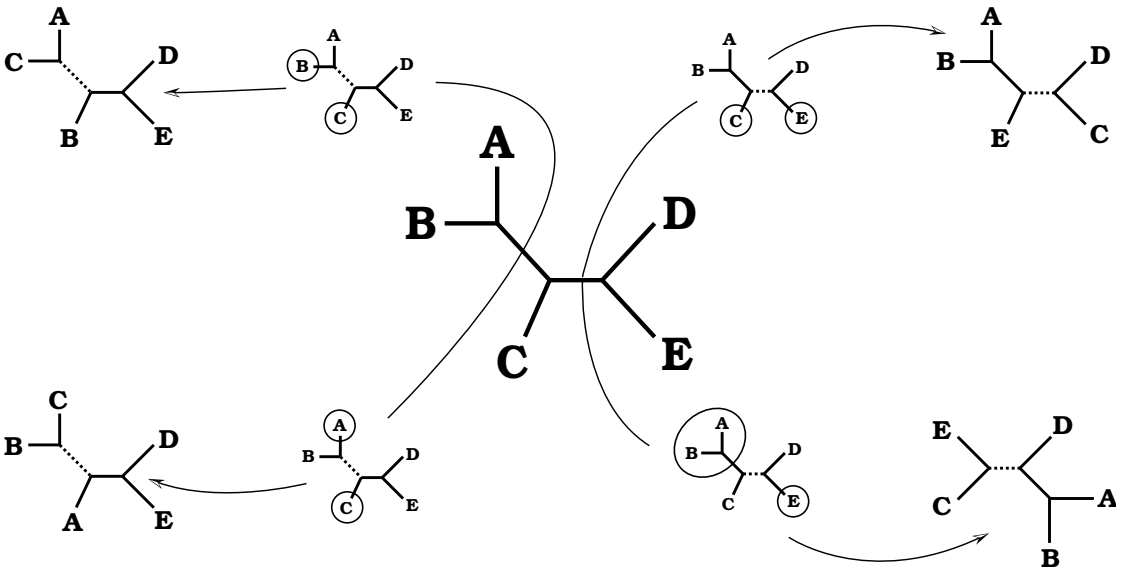


Most commonly used methods are "hill-climbers."

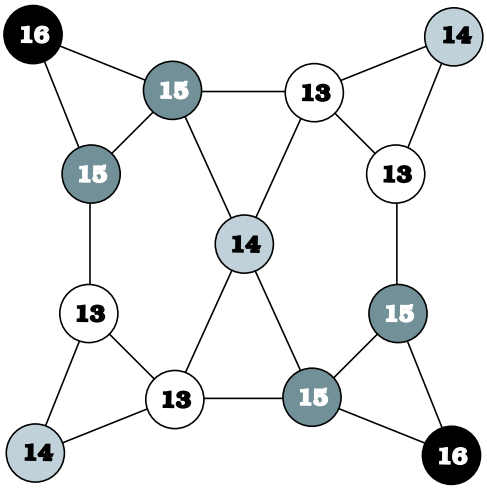
Multiple optima found by repeating searches from different origins.

Severity of the problem of multiple optima depends on step size.

Nearest Neighbor Interchange (NNI)



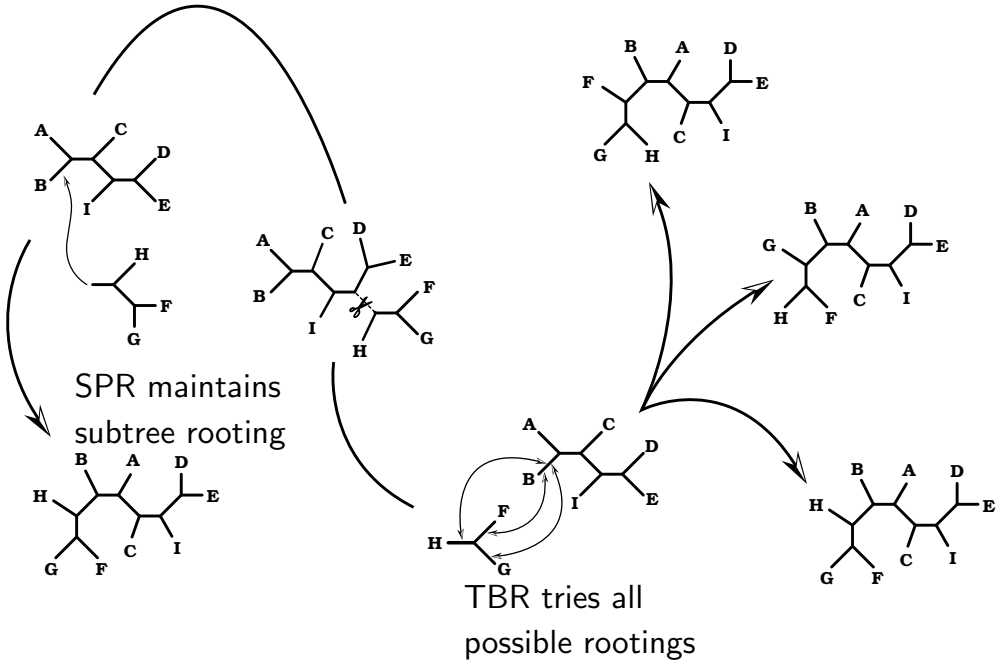
Nearest Neighbor Interchange (NNI)



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

Contrived matrix with
2 NNI islands

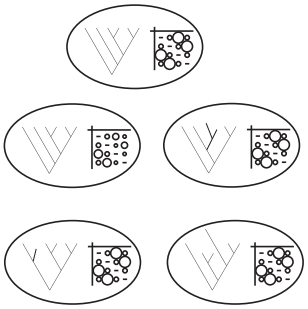
Subtree Pruning Regrafting (SPR) and Tree Bisection Reconnection (TBR)



Many other heuristic strategies proposed

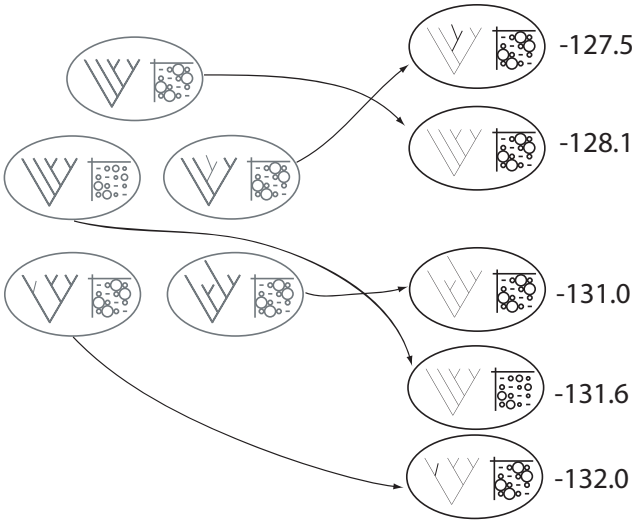
- Swapping need not include *all* neighbors (RAxML, `reconlimit` in PAUP*)
- "lazy" scoring of swaps (RAxML)
- Ignoring (at some stage) interactions between different branch swaps (PHYML)
- Stochastic searches
  - Genetic algorithms (GAML, MetaPIGA, GARLI)
  - Simulated annealing
- Divide and conquer methods (the sectortial searching of Goloboff, 1999; Rec-I-DCM3 Roshan 2004)
- Data perturbation methods (*e.g.* Kevin Nixon's "ratchet")

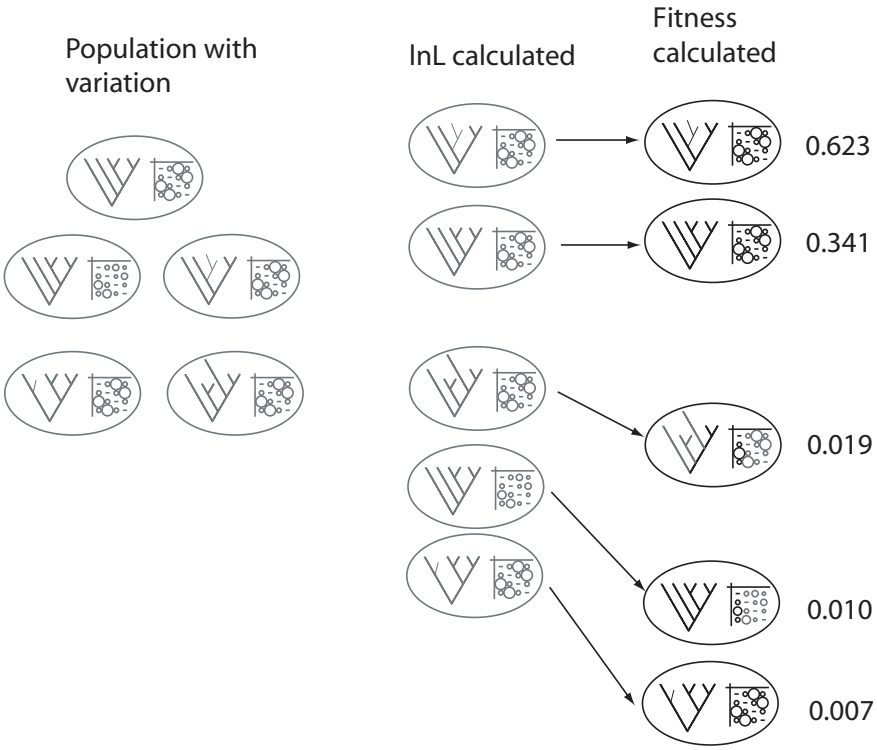
Population with variation

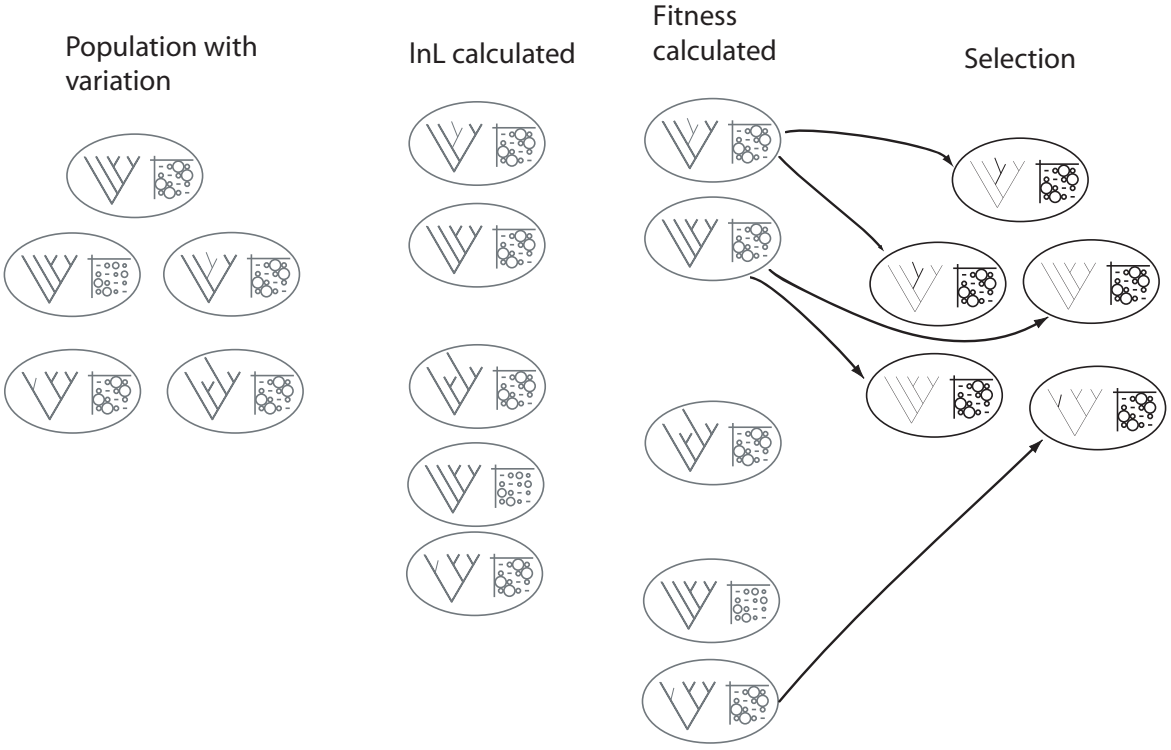


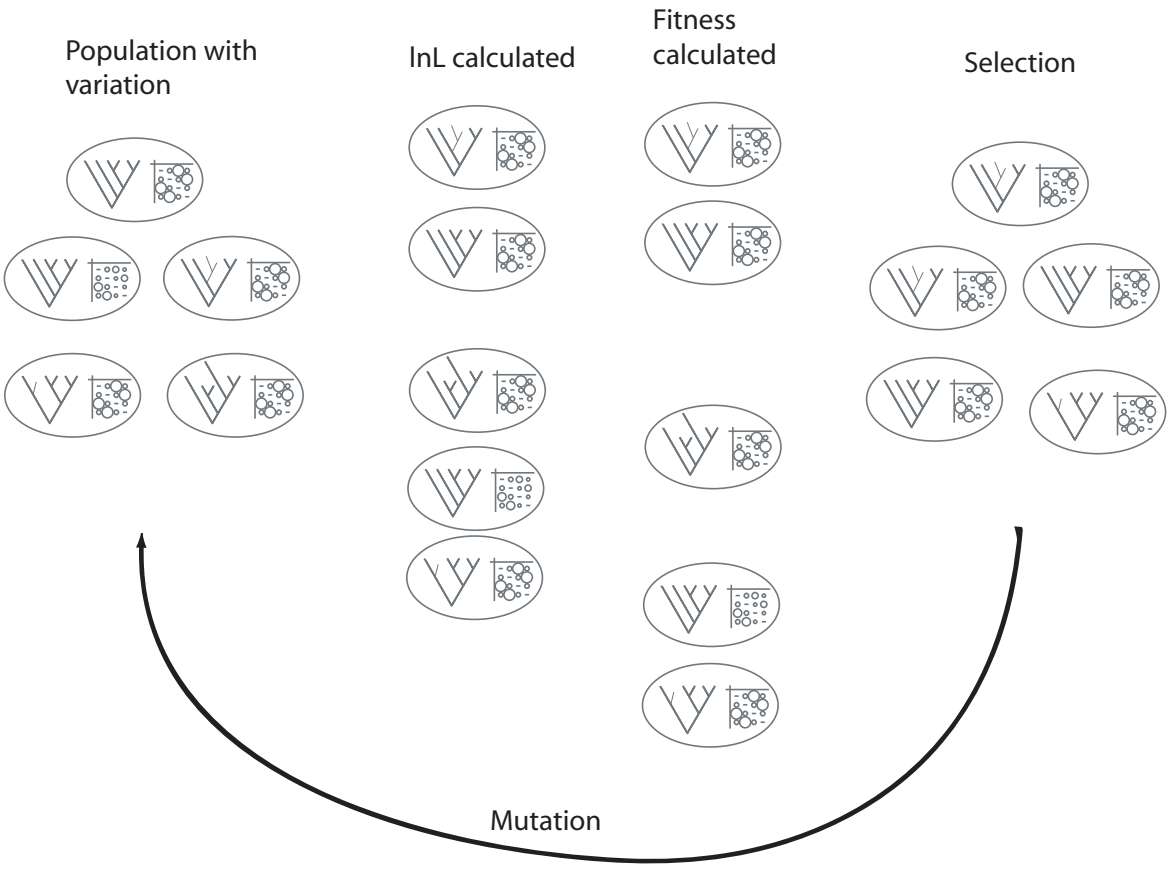
Population with variation

InL calculated




Population with variation
lnL calculated
Fitness calculated
0.623
0.341
0.019
0.010
0.007


Population with variation
lnL calculated
Fitness calculated
Selection



Software for searching under different criteria

Fast tree searching:

- Maximum likelihood – RAxML, FastTree. GARLI, phyml, Leaphy
- Distances – FastME (balanced minimum evolution); FastTree (profile approximation to balanced minimum evolution); PAUP (other distance-based criteria).
- Parsimony – TNT

Conclusions on searching

1. The large number of trees make it infeasible to evaluate every tree;

2. Intuitive, hill climbing routines often perform well;

3. Repeated searching from multiple starting points helps give you a sense of how difficult searching is for your dataset.

4. The ease of tree searching is a separate issue from statistical support. Well-supported clades are often easy to find, but we do **not** simply use the repeatability of a trees in independent searches as a measure of support (we'll talk about assessing support tomorrow).

References

Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425.

SISG "Short" PAUP* Lab

Note: Parts of this computer lab exercise wer written by Paul O. Lewis. Paul has graciously allowed Mark Holder to use and modify the lab for the Summer Institute in Statistical Genetics. Thanks, Paul!

This computer lab you will introduce you to some basic aspects of PAUP*. Versions of PAUP* exist for several different operating systems (MacIntosh, Windows, Linux, etc.), with the MacIntosh version being the most flexible and user-friendly. Most of you will be using the Windows version today.

The PAUP* Home Page is the best place to go for continuing updates on the progress being made toward the final release, and for information about purchasing the program: http://paup.csit.fsu.edu/

We are going to work from free (but expiring) versions available here: http://people.sc.fsu.edu/~swofford/paup_test

You can work through this tutorial at your own pace, asking questions whenever something needs to be clarified. Please let us know if you think another approach would be better, and if anything about this tutorial is unclear. The goals for this tutorial are to:

- Become familiar with the NEXUS data file format used by PAUP* (as well as several other prominent phylogeny programs such as Mesquite and MrBayes)
- Learn how to conduct various types of searches (exhaustive, branch-and-bound, heuristic using NNI and TBR branch swapping, and algorithmic approaches such as star decomposition and stepwise addition)
- Learn how to set up PAUP* to perform parsimony, minimum evolution, least squares searches (we will cover ML in the next lab).

Questions that you should be able to answer from looking at the output are in *italics*. Answers to the questions are provided in footnotes. If you do not understand one of these questions, or need help figuring out the answer, please do not hesitate to raise your hand.

Searching under the parsimony criterion

1. **Create the data file** Download the file angio35b.nex from http://www.people.ku.edu/~mtholder/848/data/angio35b.nex and save it on the machine that you are working on and take a quick look at it. It contains a data block with the sequence matrix; A sets block that describes where the breaks between the different genes fall; and an assumptions block that tells PAUP* to exclude some characters that may not be aligned reliably.
2. **Create a command file.** Create a blank file, then type in the following commands, and save the file as run.nex in the same directory that holds the angio35b.nex file. Here are the PAUP* commands:

```
#NEXUS
begin paup;
    Log file=output.txt start replace;
    Execute angio35b.nex;
end;
```

3. **Execute run.nex, which will in turn execute angio35b.nex**. There at least two advantages to creating little NEXUS files like run.nex. For now, the only advantage is that executing run.nex automatically starts a log file so that you will have a record of what you did. Later, when you get in the habit of putting commands in paup blocks, you will appreciate the separation of the data from the commands that initiate analyses (I have many times opened a data file, forgetting about the embedded paup block that then starts a long search, overwrites my previous log file, and otherwise creates havoc).

   Note that because we used the `replace` keyword in the `log` command, the file output.txt will be overwritten without warning if it exists. This is a bit dangerous, so you may want to refrain from using the replace keyword so that PAUP* asks before overwriting files.

4. **Delete all taxa except the first five**. The command `delete 6 - .` will cause PAUP* to ignore all taxa except Ephedrasinica, Gnetum_gnemJS, WelwitschiaJS, Ginkgo_biloba, and Pinus_ellCH. Type that command into the command line of PAUP*. Note the `.` is part of the command – it stands for 'the last member in the list' (in this context it is 'the last taxon in the data matrix'.

5. **Perform an exhaustive search using parsimony**. Use the `alltrees` command for this. This should go fast because you now have only 5 taxa.

   - *How many separate tree topologies did PAUP* examine?*[1]
   - *What is the parsimony treelength of the best tree? The worst tree?*[2]
   - *How many steps separate the best tree from the next best?*[3]

6. **Perform an heuristic search using NNI branch swapping**. Before you start, use the `describe` command to show you the tree obtained from the exhaustive enumeration. Draw this tree on a piece of paper and then draw the 4 possible NNI rearrangements

   Find all NNI rearrangements of the best tree. Note that because we performed an exhaustive enumeration, we now know which tree is the globally most parsimonious tree. We are thus guaranteed to never find a better tree were we to start an heuristic search with this tree. Let's do an experiment: perform an NNI heuristic search, starting with the best tree, and have PAUP* save all the trees it encounters in this search. In the end, PAUP* will have in memory 5 trees: the starting tree and the 4 trees corresponding to all possible NNI rearrangements of that starting tree:
   `hsearch start=1 swap=nni nbest=15`

   - `start = 1` starts the search from the tree currently in memory (i.e., the best tree resulting from your exhaustive search using the parsimony criterion)
   - `swap = nni` causes the Nearest-Neighbor Interchange (NNI) method to be used for branch swapping
   - `nbest = 15` saves the 15 best trees found during the search. Thus, were PAUP* to examine every possible tree, we would end up saving all of them in memory. The reason this command is needed is that PAUP* ordinarily does not save trees that are worse than the best one it has seen thus far. Here, we are interested in seeing the trees that are examined during the course of the search, even if they are not as good as the starting tree.

   Show all 5 trees in memory. Use the `describe all` command to plot the 5 trees currently in memory. The reason we are using the `describe` command rather than the `showtrees` command is

[1] 15 topologies
[2] 1110 was the best score, and 1247 was the worst
[3] 13 steps – this is hard to see in the versions of PAUP posted on June/2011.

because we want PAUP* to show us the numbers it has assigned to the internal nodes, something that `showtrees` doesn't do.

- *Which tree was the original tree?*[4]
- *Which trees correspond to NNI rearrangments of which internal edges on the original tree?*[5]

7. **Find the most parsimonious tree for all 35 taxa**. Restore all taxa using the `restore all` command (this will wipe out the 5 trees you currently have stored in memory, but that is ok), then conduct a heuristic search having the following characteristics:

   - The starting trees are each generated by the stepwise addition method, using random addition of sequences
   - Swap using NNI branch swapping
   - Reset the `nbest` option to `all` because we want to be saving just the best trees, not suboptimal trees (yes, this option is a little confusing).
   - Set the random number seed to 5555 (this determines the sequence of pseudorandom numbers used for the random additions; ordinarily you would not need to set the random number seed, but we will do this here to ensure that we all get the same results)
   - Do 500 replicate searches; each replicate represents an independent search starting from a different random-addition tree

   Here is the full command implementing this search:
   `hsearch start=stepwise addseq=random swap=nni nbest=all rseed=5555 nreps=500`

   - *How many tree islands were found?*[6]
   - *How long did the search take?*[7]
   - *How many rearrangements were tried?*[8]

8. **Conduct a second search with SPR swapping**. Be sure to reset the random number seed to 5555. You should be able to figure out how to do this using the output from `hsearch ?` command. Note that to save typing you can call up previously entered commands using the little buttons on the right of the command line edit control (or using the arrow up key).

   - *How many tree islands were found?*[9]
   - *What are the scores of the trees in each island?*[10]
   - *How long did the search take?*[11]
   - *How many rearrangements were tried?*[12]

9. **Now conduct a third search with TBR swapping.**

[4] It should be the first one – the tree with score 1110.
[5] It is hard to describe in the footnote – but ask me if you have questions about this
[6] 70 islands in old versions of PAUP. 58 islands on the version of PAUP posted June/2011
[7] 1.08 seconds on my laptop
[8] 147,531 rearrangements in old versions of PAUP. 155,616 rearrangments on the version of PAUP posted June/2011
[9] 4 islands in old versions of PAUP. 3 islands on the version of PAUP posted June/2011
[10] two at 5689, one at 5693 and one at 5697 (the version of PAUP posted June/2011 does not find the tree with score 5693 for this seed)
[11] 8 seconds on my laptop
[12] 5,023,936 rearrangements in old versions of PAUP. 3,960,984 rearrangements on the version of PAUP posted June/2011

- *How many tree islands were found?*[13]
- *What are the scores of the trees in each island?*[14]
- *How long did the search take?*[15]
- *How many rearrangements were tried?*[16]
- *How many trees are currently in memory (use the* `treeinfo` *command)?*[17]
- *Has PAUP* saved trees from all islands discovered during this search? (Hint: compare "Number of trees retained" to the sum of the "Size" column in the Tree-island profile.) Do you know why PAUP* saved the number of trees that it did?*[18]

  Wondering about that warning "Multiple hits on islands of unsaved trees may in fact represent different islands"? When PAUP* encounters a new island, it will find all trees composing that particular island in the process of branch swapping. Thus, if (in a new search) it encounters any trees already stored in memory, it knows that it has hit an island that it found previously. Note that it would be pointless to continue on this tack, because it will only find all the trees on that island again. For trees retained in memory, PAUP* can keep track of which island they belong to (remember that it is possible for trees with the same parsimony score to be in different tree islands!). But for trees that are not retained in memory, PAUP* only knows that it has encountered an island of trees having score X; it has no way of finding out how many islands are actually represented amongst the trees having score X.

If you want any of these commands to happen whenever you execute this file, then you can simply add the commands that you typed into the PAUP block of run.nex, add a semicolon after the command, and save the file.

[13] 4 islands in old versions of PAUP. 3 islands on the version of PAUP posted June/2011

[14] two at 5689, two at 5693 and one at 5697 (the version of PAUP posted June/2011 does not find the tree with score 5693 for this seed)

[15] 9 seconds on my laptop

[16] 14,790,674 rearrangements in old versions of PAUP. 10,698,858 rearrangements on the version of PAUP posted June/2011

[17] two

[18] No, it only save the trees from the best islands

Some of the distance methods in PAUP* and FastME

The goal of this part of the lab exercise is to show you how to conduct distance-based analyses in PAUP* and FastME.

Basic distance analyses in PAUP*

1. We will use the same angio35b.nex file that we used for the parsimony part of the lab.

2. Use a text editor to create a new file. Save it as rund.nex in the same directory as the angiob.nex file. This new file will be a NEXUS file that contains the PAUP block with the commands for PAUP. Here are the commands:

```
#NEXUS

begin paup;
  execute angio35b.nex;
  dset dist=abs;
  delete 5-.;
  exclude missambig;
  showdist;
end;
```

   This file will tell PAUP* to:

   - use the absolute number of nucleotide differences between taxa as the distance measure (`dset dist=abs`);
   - delete all taxa except the first four (`delete 5-.`);
   - exclude all sites that have missing or ambiguous data (`exclude missambig`); and
   - show the distance matrix (`showdist`)

   Save the rund.nex file. Then execute it PAUP*, and examine the output.

3. **Calculate p-distances and JC distances** To see the distances as a proportion of sites that differ for the sequences just change the distance measure from `abs` to `p` and re-execute rund.nex. Examine the resulting data matrix, change the distance correction in rund.nex from `p` to `jc` to tell PAUP to use the Jukes-Cantor distance (this is the simplest model for correcting distances – Jeff will discuss the models tomorrow, but for this exercise it is only necessary to know that this is the name of a model used to correct the observed distances for multiple hits). Re-execute the file.

   - *How do the JC distances compare with the p-distances? Does the ordering of distances change?*[19]
   - *You have seen two distance measures that PAUP* can calculate, but how could you get a list of all the distance measures it can compute?*[20]

[19] The ordering does not change, but the JC distances are larger (and the larger *p*-distances are corrected more when you use the model-based correction).

[20] `dset ?`

4. Execute the `dset ?` command to see all of the distance settings that are available in PAUP. If you are confused by an option, you can check it out by downloading the PAUP manual from: http://paup.csit.fsu.edu/Cmd_ref_v2.pdf (or by asking me about an option).

5. **Estimate edge lengths using weighted least squares** Next, perform a search using weighted least squares (weighted usually implies that the power is 2 in the denominator of the sum-of-squares formula). Add the following line to your `run.nex` file, just below the `execute angiob35.nex;` line:

   ```
   set criterion=distance ;
   ```

   This command tells PAUP* to use the distance optimality criterion specified by the `objective` option of the `dset` command during the search. If you were to leave this out, PAUP* would use the default optimality criterion (parsimony). Now issue the command `dset ?` in PAUP* to get a listing of the current values of all distance settings.

   - What is the current setting for the `power` option? If your answer is not 2, then add this line to your paup block, below the `execute` command: `dset power=2;`
   - What is the current setting for the `objective` option? If your answer is not `lsfit`, then add this line to your paup block, also below the `execute` command: `dset objective=lsfit;`

   Finally, add the following two lines to the end of your paup block to perform the search and show the resulting best tree, including the estimated branch lengths:

   ```
   alltrees;
   describe 1 / brlens;
   ```

6. Save and re-execute the file.

7. Look for the line that begins "Weighted sum-of-squares =". This is the least-squares score for the tree. In its output, PAUP* also gives you the score that would have been used were we using the minimum evolution criterion. *Can you find it?* Ask for help if you need to; PAUP* does not make this obvious. Write the value down for comparison with the value you will obtain in the next section.

8. **Searching under the minimum evolution criterion**. Before moving on to the next exercise, repeat the above search using the minimum evolution (or ME) criterion. To do this, you will need to add the `objective=ME` option to your `dset` command (be sure to remove the previous `dset objective=lsfit` if you had to put it in) and re-execute `rund.nex`.
   Is the result what you expected based on your answer to the last question in the previous section?[21]

Felsenstein zone example from lecture (optional)

You can download the file that I showed the "birds-eye view" of during lecture from:
http://www.people.ku.edu/ mtholder/848/data/fzone_sim.nex
It is an example of a "Felsenstein-zone" tree in which the taxa with long branches are not sister to each other.

1. Use PAUP* to find the least-squares tree for this data set using the p-distances.

[21] Hopefully. (Note: it is ok if the results differ slightly in the 5th. decimal place.)

2. *Is the tree the true tree, or the "long-branch attraction" tree preferred?*[22]

3. The data was simulated using the Jukes-Cantor model. Tell PAUP* to use the JC distance correction and do another search.

4. *Is the tree the true tree, or the "long-branch attraction" tree preferred?*[23]

Compare NJ with a comparable heuristic search (optional)

In this exercise, you will conduct a Neighbor-joining (NJ) analysis using JC distances and compare this with an heuristic search using the minimum evolution criterion. The goal of this section is to demonstrate that it is possible for heuristic searching to find a better tree than NJ, even using (almost) the same optimality criterion.

1. Please quit PAUP* and start it again. The reason for this will be made clear later, but mainly the purpose is to return all settings to their default values.

2. Put the commands below in a paup block in a new file. Note that we are again using the angio35b.nex file:

```
execute angio35b.nex;
dset distance=jc objective=me;
nj;
dscores 1;
set Criterion = dist;
hsearch start=1;
dscores 1;
```

3. *What is the minimum evolution score for the NJ tree?* (scroll down from the beginning of the PAUP* output looking for the phrase "ME-score" right after point where the NJ tree is displayed)

4. *What is the minimum evolution score for the tree found by heuristic search starting with the NJ tree?*

5. *What is wrong? Why is the minimum evolution score of the heuristic search worse than that of the starting tree?* (Hint: take a look at the "Heuristic search settings" section of the output.)

6. Once you have figured out what is going on (ask me for help if you are stumped), fix your paup block and re-execute the file.

7. In your reanalysis, you should find that the heuristic search starting with the NJ tree found a better tree according to minimum evolution than NJ. Neighbor-joining is very popular, but you should be wary of NJ trees involving large numbers of taxa. This analysis involved 35 taxa; for problem of this size or larger, it is almost certain that NJ will not find the best tree.

8. *How much do the trees differ from each other?* To figure out, we'll need to get PAUP* to save the nj tree so that when we do the search we do not lose the NJ tree. Change your command file to say:

[22] the "long-branch attraction" tree
[23] the true tree

```
execute angio35b.nex;
set Criterion = dist;
dset distance=jc objective=me;
nj;
dscores 1;
savetree file = nj.tre ;
hsearch start=1;
dscores 1;
gettrees file = nj.tre mode = 7;
treedist ;
```

Here is the explanation:

- The `savetree` command writes the tree to a file (in the newick representation).
- The `gettrees` command reads the nj tree back into memor.
- The `mode=7` option to the `gettrees` command means "read the trees from the file, but do not throw away the trees that are currently in memory
- After `gettrees mode=7` command, you'll have the ME tree and the NJ tree in memory.
- The `TreeDist` command calculates tree-to-tree distances. The default is the symmetric-difference – the number of edges in tree #1 that are not in tree #2 plus the number of edges in tree #2 that are not in tree #1. It would be 0 if the trees were identical.

Balanced minimum evolution search in FastME

FastME is a program written by Desper and Gascuel. You can download it from
http://www.atgc-montpellier.fr/fastme/
Among other analyses (such as BIONJ, an algorithm that is similar to NJ, but does a better job with highly divergent sequences), FastME implements fast NNI searching under the balanced minimum evolution criterion. NJ is a quick and dirty search under this criterion, and FastME can do branch swapping to find even better trees. The program produces trees very quickly, and this is best demonstrated on large datasets.

1. Download 567Tx2153C.nex from
   http://www.people.ku.edu/~mtholder/848/data/567Tx2153C.nex
   to get a large (567 taxon, 2153 character) data file.
2. FastME does not analyze sequences directly. Instead we have to give it a distance matrix. We will use PAUP* to create the input distance matrix.
3. Execute the 567Tx2153C.nex file in PAUP*.
4. Tell PAUP* to use the JC distance
5. Export the data file in a format that FastME can read using the command:
   `savedist format = phylip triangle = both diagonal file=dists.txt`
6. While you have PAUP open, do a NJ search, score the tree using minimum evolution, ordinary (unweighted) least squares, and weighted least squares.

7. Conduct a search under the weighted least squares criterion using the command: `HSearch NoMulTrees;` Score this tree under minimum evolution, ordinary (unweighted) least squares, and weighted least squares and note the amount of time the search took.

8. Now we will run FastME on the distance matrix. Open a command terminal and change the working directory of the terminal to the directory where you saved the dists.txt file. To invoke FastME, you simply type its name. On Mac, the command is `FastME_MacOS` (I must admit that I've never run it on Windows).

9. The program should prompt you for the name of the input file. At this point enter dists.txt and hit return. You should see a menu of options that let you choose what criterion to optimize and which algorithm to use. Conduct a Balanced Generalized Minimum Evolution search using the balanced NNI search. The program exits when it is finished (and it won't take long).

10. It should produce two output files: dists.txt_stat.txt and dists.txt_tree.txt. Open both in a text editor.

11. To get the tree into PAUP* we will have to change the tree file into a NEXUS file. Fortunately all you have to do to accomplish this is add:

    ```
    #NEXUS
    begin trees;
    tree fastme = [&U]
    ```

    to the file before the parenthetical notation, and

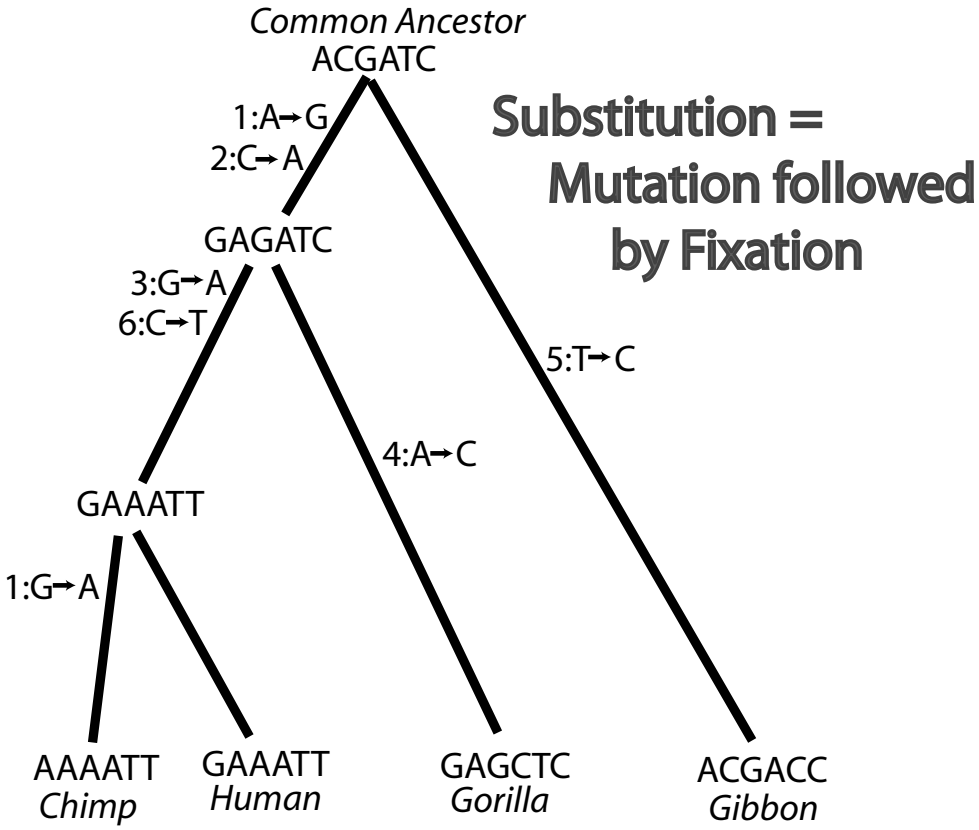
    ```
    end;
    ```

    to the end of the file.

12. Execute the 567Tx2153C.nex file and set the distance back to the JC corrected distance.

13. Use the `GetTrees` command to do get the FastME trees into memory in PAUP*. Score this tree under minimum evolution, ordinary (unweighted) least squares, and weighted least squares and note the amount of time the search took.

14. How did the tree from FastME compare to the trees that you obtained by searching in PAUP? (recall that balanced minimum evolution is not the same as minimum evolution or least-squares, so it is not too surprising if you get different trees).



AAAATT
Chimp

GAAATT
Human

GAGCTC
Gorilla

ACGACC
Gibbon

Likelihood (Prob. of data given model & parameter values) =

AAAATT *Chimp* GAAATT *Human* GAGCTC *Gorilla* ACGACC *Gibbon*

Likelihood for Site 1 X

AAAATT *Chimp* GAAATT *Human* GAGCTC *Gorilla* ACGACC *Gibbon*

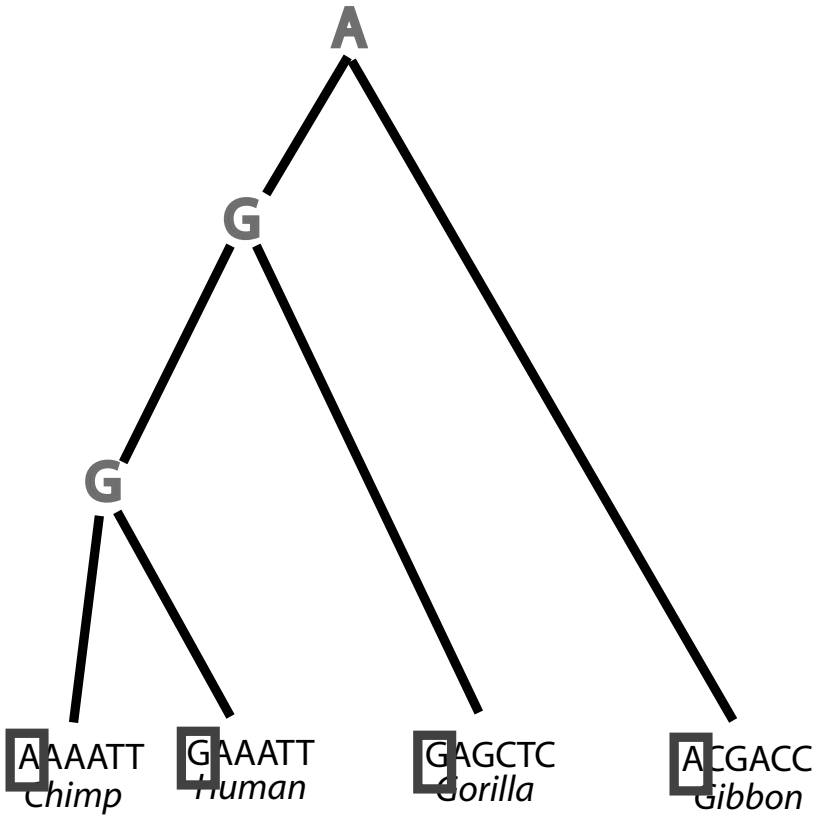
Likelihood for Site 2 X

AAAATT *Chimp* GAAATT *Human* GAGCTC *Gorilla* ACGACC *Gibbon*

... X ... X ... X

Likelihood for Site 6

AAAATT *Chimp* GAAATT *Human* GAGCTC *Gorilla* ACGACC *Gibbon*



Probabilistic models of nucleotide change (independently and identically evolving sites)

Let q_{ij} be the instantaneous rate of change at a site from nucleotide type i to j

Q is matrix of instantaneous rates (Q will have 4 rows and 4 columns because i and j can each by any of 4 nucleotide types)

For nucleotide starting as type i at time 0, probability nucleotide is type j at time t is denoted $p_{ij}(t)$.

$p_{ij}(t)$ is referred to as a *transition probability*.

Consider a **very very** small amount of evolutionary time Δt. When $i \neq j$,

$$p_{ij}(\Delta t) \doteq q_{ij}\Delta t$$

$$p_{ii}(\Delta t) \doteq 1 - \sum_{j,j\neq i} q_{ij}\Delta t$$

$$p_{ii}(\Delta t) \doteq 1 + q_{ii}\Delta t$$

where

$$q_{ii} = - \sum_{j,j\neq i} q_{ij}$$

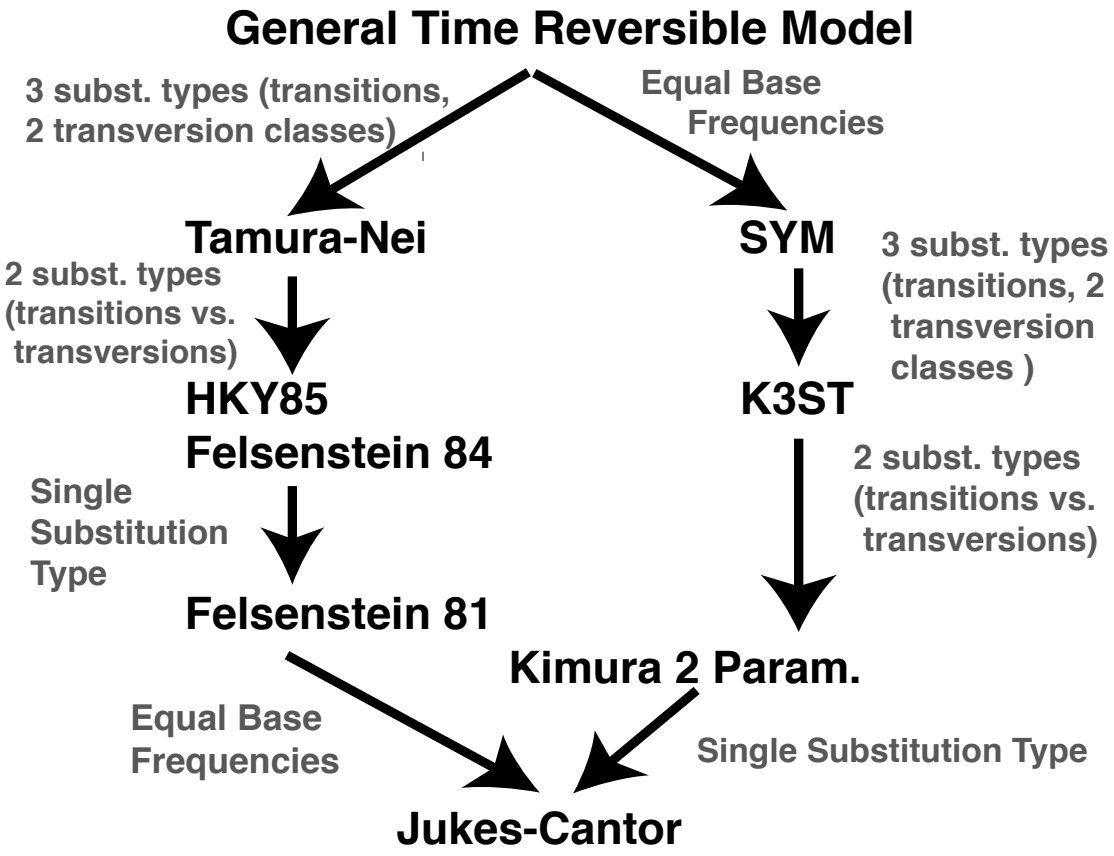
(in preceding equations, $\doteq$ can be replaced by $=$ when the limit as Δt approaches 0 is taken)

Jukes-Cantor model is simplest model of nucleotide substitution.
It assumes sequence positions evolve independently and it assumes that all possible changes at a position are equally likely.

Let π_j be probability a residue is type j. π_j is called the equilibrium probability of type j.

$$p_{ij}(\infty) = \pi_j$$

For Jukes-Cantor model, $\pi_j = 1/4$ for all 4 nucleotide types j.



based on Figure 11 from Swofford et al. Chapter in **Molecular Systematics** (Sinauer, 2nd ed. 1996)

Rate Matrix for Jukes-Cantor Model

FROM \ To	A	C	G	T
A	-3μ	μ	μ	μ
C	μ	-3μ	μ	μ
G	μ	μ	-3μ	μ
T	μ	μ	μ	-3μ

Note 1: Diagonal matrix elements are **rate away** from nucleotide type of that row.

Note 2: In later slide on Jukes-Cantor model, we write $s/3$ rather than μ.

Rate Matrix for Kimura 2-Parameter Model

FROM \ To	A	C	G	T
A	$-\alpha-2\beta$	β	α	β
C	β	$-\alpha-2\beta$	β	α
G	α	β	$-\alpha-2\beta$	β
T	β	α	β	$-\alpha-2\beta$

Changes involving only purines (i.e., A and G) or only pyrimidines (i.e., C and T) are transitions. Changes involving one purine and one pyrimidine are transversions.

Rate Matrix for Felsenstein 1981 Model

FROM \ To	A	C	G	T
A	$-\mu(\pi_C + \pi_G + \pi_T)$	$\mu\pi_C$	$\mu\pi_G$	$\mu\pi_T$
C	$\mu\pi_A$	$-\mu(\pi_A + \pi_G + \pi_T)$	$\mu\pi_G$	$\mu\pi_T$
G	$\mu\pi_A$	$\mu\pi_C$	$-\mu(\pi_A + \pi_C + \pi_T)$	$\mu\pi_T$
T	$\mu\pi_A$	$\mu\pi_C$	$\mu\pi_G$	$-\mu(\pi_A + \pi_C + \pi_G)$

Rate Matrix for Hasegawa-Kishino-Yano (a.k.a. HKY or HKY85) Model

FROM \ To	A	C	G	T
A	$-\mu(\pi_C + \kappa\pi_G + \pi_T)$	$\mu\pi_C$	$\mu\kappa\pi_G$	$\mu\pi_T$
C	$\mu\pi_A$	$-\mu(\pi_A + \pi_G + \kappa\pi_T)$	$\mu\pi_G$	$\mu\kappa\pi_T$
G	$\mu\kappa\pi_A$	$\mu\pi_C$	$-\mu(\kappa\pi_A + \pi_C + \pi_T)$	$\mu\pi_T$
T	$\mu\pi_A$	$\mu\kappa\pi_C$	$\mu\pi_G$	$-\mu(\pi_A + \kappa\pi_C + \pi_G)$

Rate Matrix for General Time Reversible Model

FROM \ To	A	C	G	T
A	$-\mu(a\pi_C + b\pi_G + c\pi_T)$	$\mu a\pi_C$	$\mu b\pi_G$	$\mu c\pi_T$
C	$\mu a\pi_A$	$-\mu(a\pi_A + d\pi_G + e\pi_T)$	$\mu d\pi_G$	$\mu e\pi_T$
G	$\mu b\pi_A$	$\mu d\pi_C$	$-\mu(b\pi_A + d\pi_C + f\pi_T)$	$\mu f\pi_T$
T	$\mu c\pi_A$	$\mu e\pi_C$	$\mu f\pi_G$	$-\mu(c\pi_A + e\pi_C + f\pi_G)$

Time Reversibility is a common property of models of sequence evolution.

Time reversibility means that $\pi_i p_{ij}(t) = \pi_j p_{ji}(t)$ for all i, j, and t.

$\pi_i q_{ij} = \pi_j q_{ji}$ for all i and j.

For phylogeny reconstruction, time reversibility means that we cannot (on the basis of sequence data alone) hope to distinguish which of two sequence is ancestral and which is the descendant.

The practical implication of time reversibility for phylogeny reconstruction is that maximum likelihood cannot infer the position of the root of the tree unless additional information information exists (e.g., which taxa are the outgroups) or additional assumptions are made (e.g., a molecular clock).

Q will represent matrix of instantaneous rates of change. For general time reversible model, entries of Q are:

		To		
From	A	C	G	T
A	$-(a\pi_C + b\pi_G + c\pi_T)$	$a\pi_c$	$b\pi_G$	$c\pi_T$
C	$a\pi_A$	$-(a\pi_A + d\pi_G + e\pi_T)$	$d\pi_G$	$e\pi_T$
G	$b\pi_A$	$d\pi_C$	$-(b\pi_A + d\pi_C + f\pi_T)$	$f\pi_T$
T	$c\pi_A$	$e\pi_C$	$f\pi_G$	$-(c\pi_A + e\pi_C + f\pi_G)$

In above matrix: a, b, c, d, e, and f cannot be negative. With any rate matrix (including above), the transition probabilities $P(t)$ can be determined from the rate matrix Q and the amount of evolution t via

$$P(t) = e^{Qt} = I + \frac{(Qt)}{1!} + \frac{(Qt)^2}{2!} + \frac{(Qt)^3}{3!} + \ldots,$$

where I is the identity matrix.

Computing $p_{ij}(t)$ for the Jukes-Cantor model

The Jukes–Cantor model assumes that this is how nucleotide substitution occurs:

0. $\pi_A = \pi_G = \pi_C = \pi_T = \frac{1}{4}$.

1. For each site in the sequence, an "event" will occur with probability $\frac{4}{3}s$ per unit evolutionary time.

2. If no event occurs, the residue at the site does not change.

3. If an event occurs, the probability that a residue is type i after the event is π_i.

What is the probability that no event occurs in t units of evolutionary time?

$$(1-\frac{4}{3}s) \times (1-\frac{4}{3}s) \times (1-\frac{4}{3}s) \ldots (1-\frac{4}{3}s) = (1-\frac{4}{3}s)^t.$$

When $\frac{4}{3}s$ is close to 0,

$$1-\frac{4}{3}s \doteq e^{-\frac{4}{3}s}.$$

$$\Pr(\text{no event}) = (1 - \frac{4}{3}s)^t \doteq e^{-\frac{4}{3}st}.$$

When s is redefined as an instantaneous rate per unit evolutionary time, the approximation becomes an equality:

$$\Pr(\text{no event}) = e^{-\frac{4}{3}st}.$$

$$\Pr(\text{at least one event}) = 1 - \Pr(\text{no event}) = 1 - e^{-\frac{4}{3}st}.$$

If there have been no "events", then the residue cannot possibly have changed after an amount of evolution t.

If there has been at least one event, then the residue is type j with probability π_j.

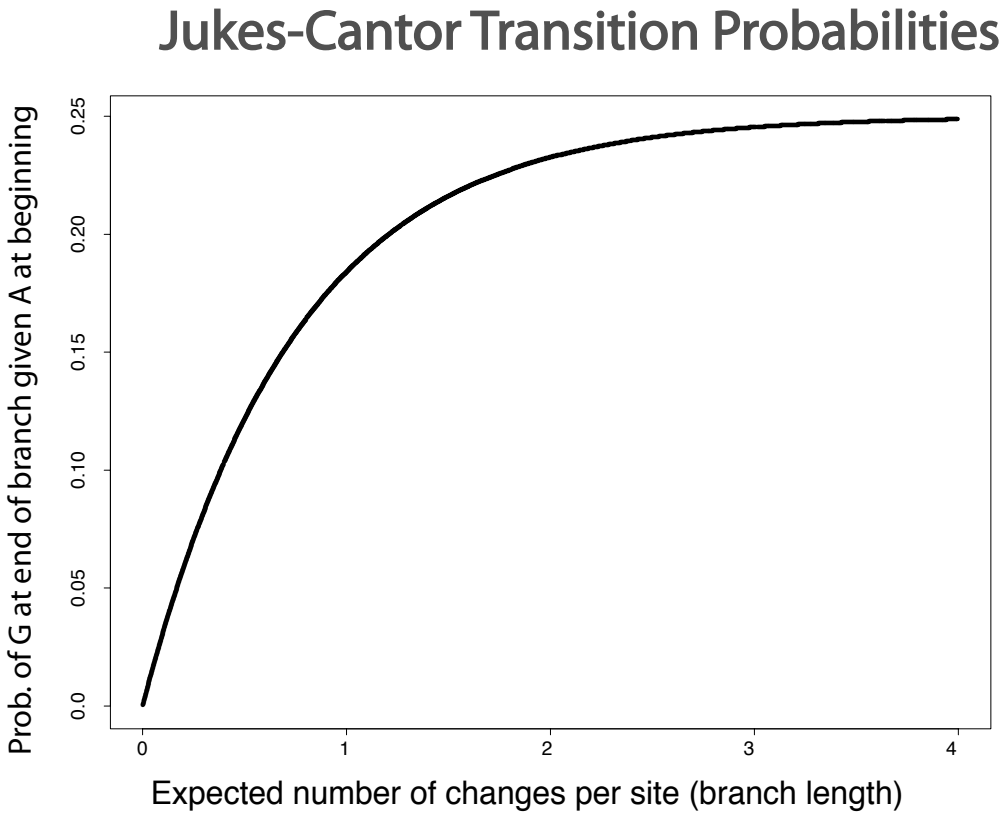
$$\begin{aligned} p_{ii}(t) &= \Pr(\text{no events}) + \Pr(\text{at least one event})\pi_i \\ &= e^{-\frac{4}{3}st} + (1 - e^{-\frac{4}{3}st})\pi_i. \end{aligned}$$

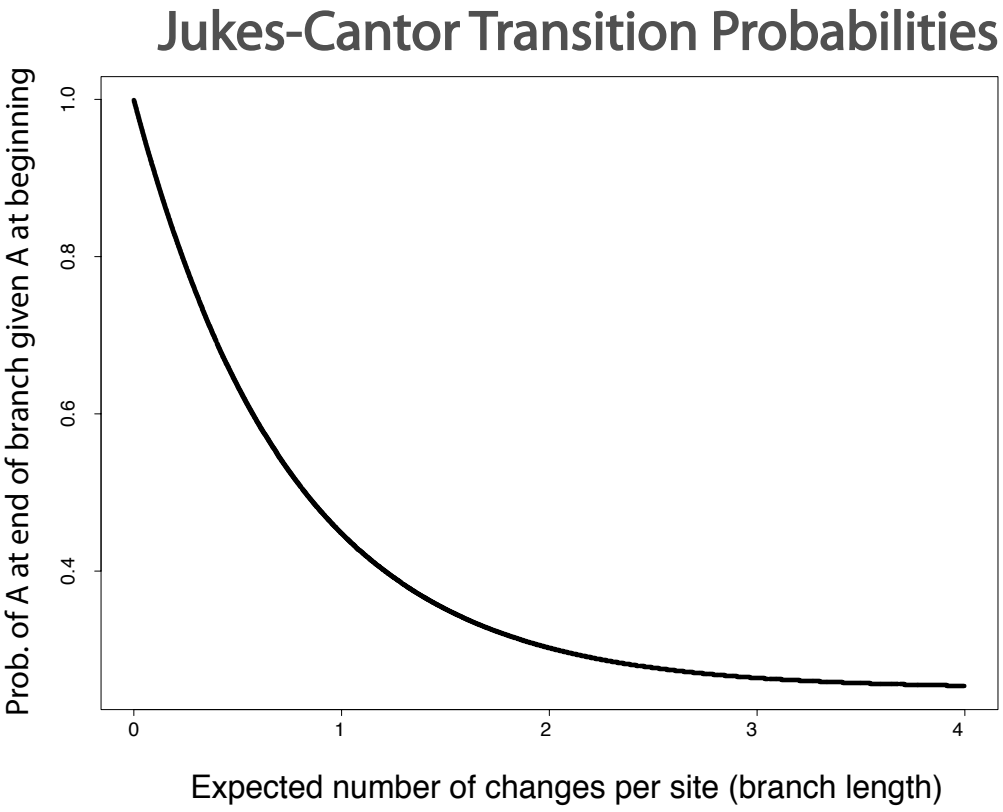
For $i \neq j$,

$$\begin{aligned} p_{ij}(t) &= \Pr(\text{at least one event})\pi_j \\ &= (1 - e^{-\frac{4}{3}st})\pi_j. \end{aligned}$$

Notice that $\frac{4}{3}s$ and t appear only as a product. $\frac{4}{3}s$ and t cannot be separately estimated. Only their product can be estimated.

Note: A generalization of the Jukes–Cantor model, the "Felsenstein 1981" model does not require $\pi_A = \pi_G = \pi_C = \pi_T = \frac{1}{4}$.




Jukes-Cantor Transition Probabilities
Prob. of A at end of branch given A at beginning
1.0
0.8
0.6
0.4
0
1
2
3
4
Expected number of changes per site (branch length)

Likelihood and phylogenies

Joe Felsenstein

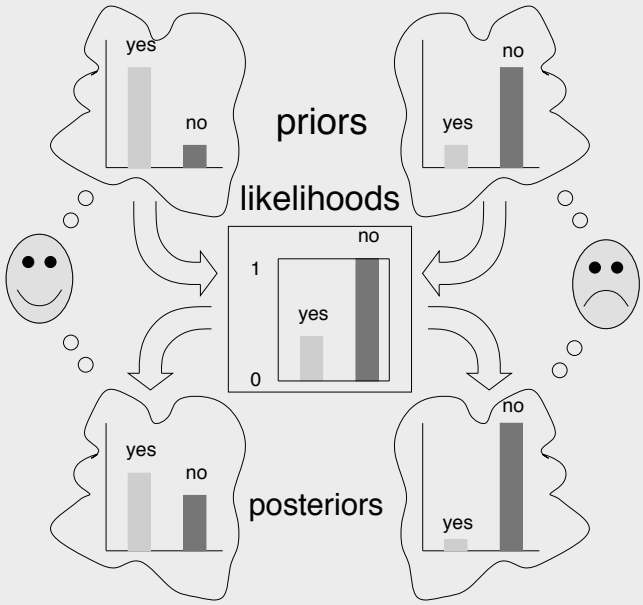
Depts. of Genome Sciences and of Biology, University of Washington

Odds ratio justification for maximum likelihood

D	**the data**
H_1	**Hypothesis 1**
H_2	**Hypothesis 2**
$\mid$	**the symbol for "given"**

$$\underbrace{\frac{\text{Prob}(H_1)}{\text{Prob}(H_2)}}_{\text{Prior odds ratio}} \quad \underbrace{\frac{\text{Prob}(D \mid H_1)}{\text{Prob}(D \mid H_2)}}_{\text{Likelihood ratio}} = \underbrace{\frac{\text{Prob}(H_1 \mid D)}{\text{Prob}(H_2 \mid D)}}_{\text{Posterior odds ratio}}$$

If a space probe finds no Little Green Men on Mars



$$\frac{4}{1} \times \frac{1/3}{1} = \frac{4}{3} \qquad \frac{1}{4} \times \frac{1/3}{1} = \frac{1}{12}$$

The likelihood ratio term ultimately dominates

If we see one Little Green Man, the likelihood calculation does the right thing:

$$\frac{1}{4} \times \frac{2/3}{0} = \frac{\infty}{1}$$

(put this way, this is OK but not mathematically kosher)

If we send n space probes and keep seeing none, the likelihood ratio term is

$$\left(\frac{1}{3}\right)^n$$

It dominates the calculation, overwhelming the prior.
Thus even if we don't have a prior we can believe in, we may be interested in knowing which hypothesis the likelihood ratio is recommending ...

Likelihood in Simple Coin-Tossing

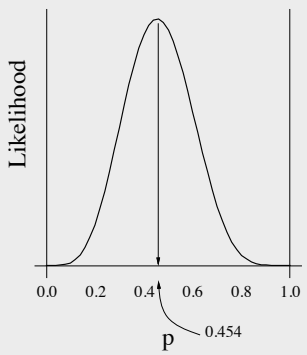
Tossing a coin n times, with probability p of heads, the probability of outcome HHTHTTTTHTTH is

$$pp(1-p)p(1-p)(1-p)(1-p)p(1-p)(1-p)p$$

which is

$$L = p^5(1-p)^6$$

Plotting L against p to find its maximum:



Differentiating to find the maximum:

Differentiating the expression for L with respect to p and equating the derivative to 0, the value of p that is at the peak is found (not surprisingly) to be $p = 5/11$:

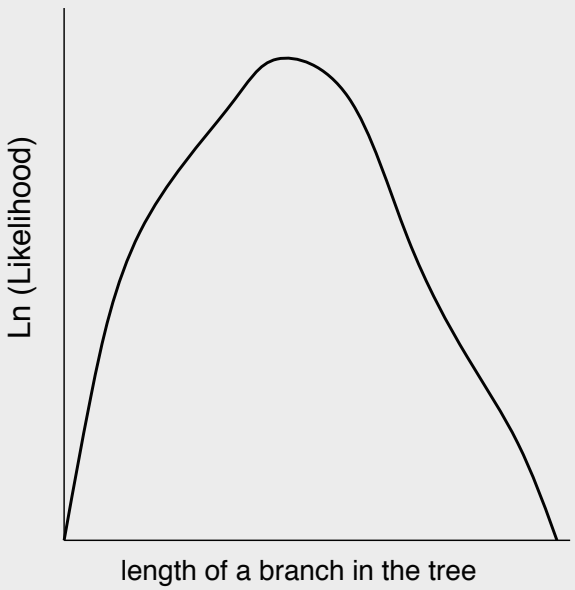
$$\frac{\partial L}{\partial p} = \left(\frac{5}{p} - \frac{6}{1-p}\right) p^5(1-p)^6 = 0$$

$$5 - 11\,p = 0$$

$$\hat{p} = \frac{5}{11}$$

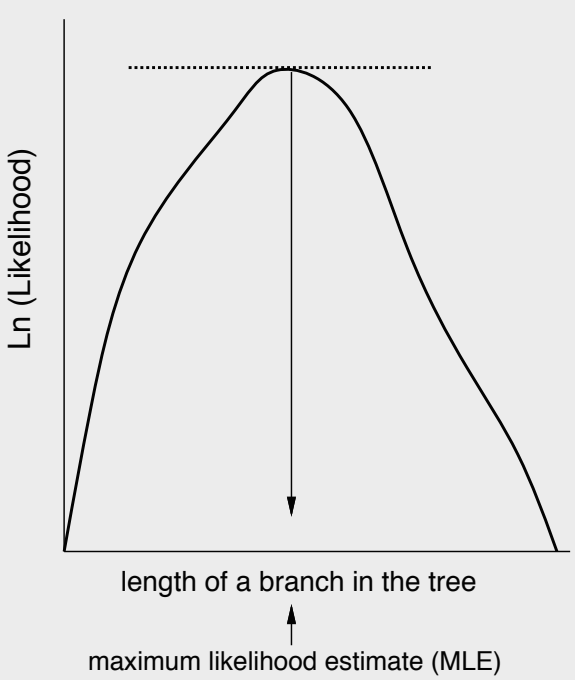
A log-likelihood curve

A log−likelihood curve in one parameter



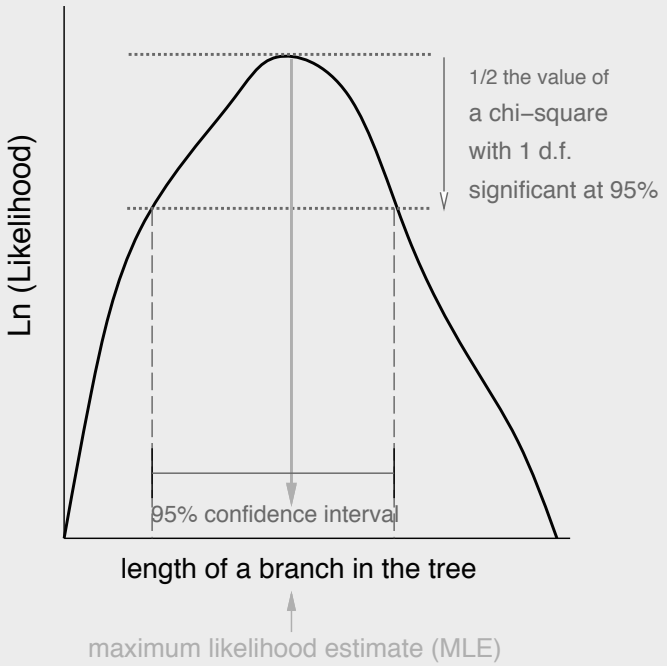
Its maximum likelihood estimate

A log−likelihood curve in one parameter
and the maximum likelihood estimate

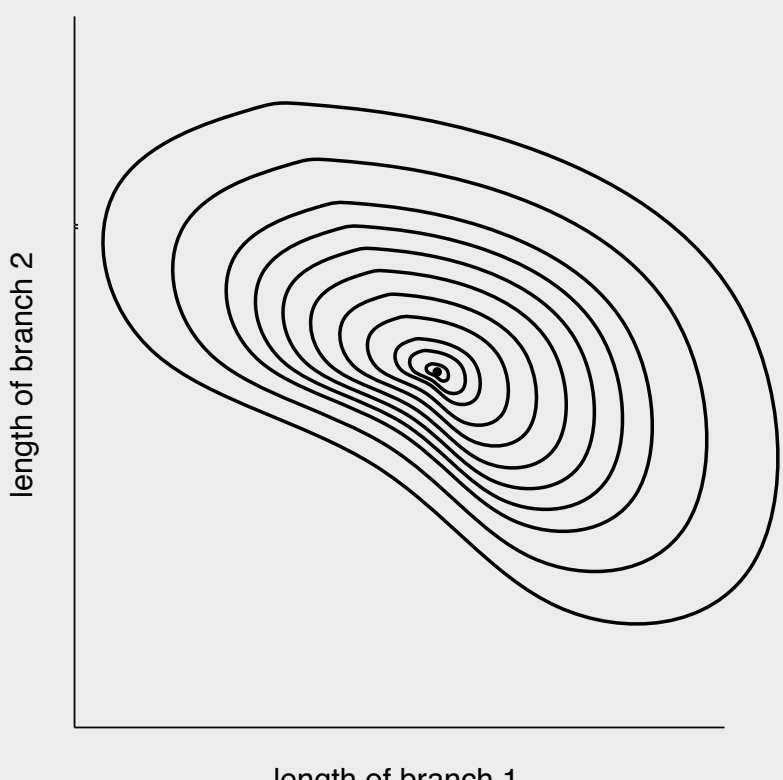


The (approximate, asymptotic) confidence interval

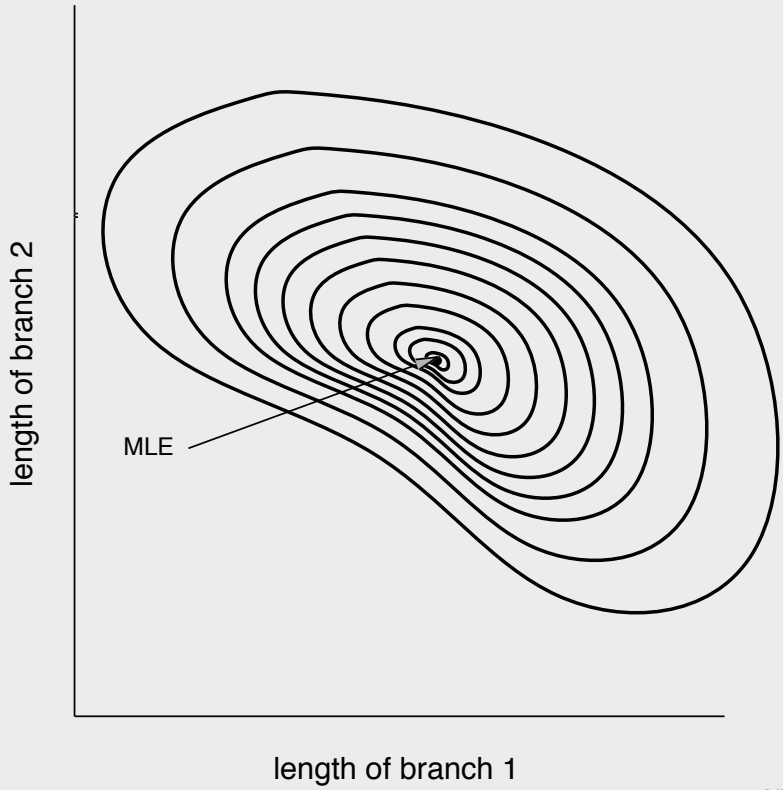
A log–likelihood curve in one parameter and the maximum likelihood estimate and confidence interval derived from it



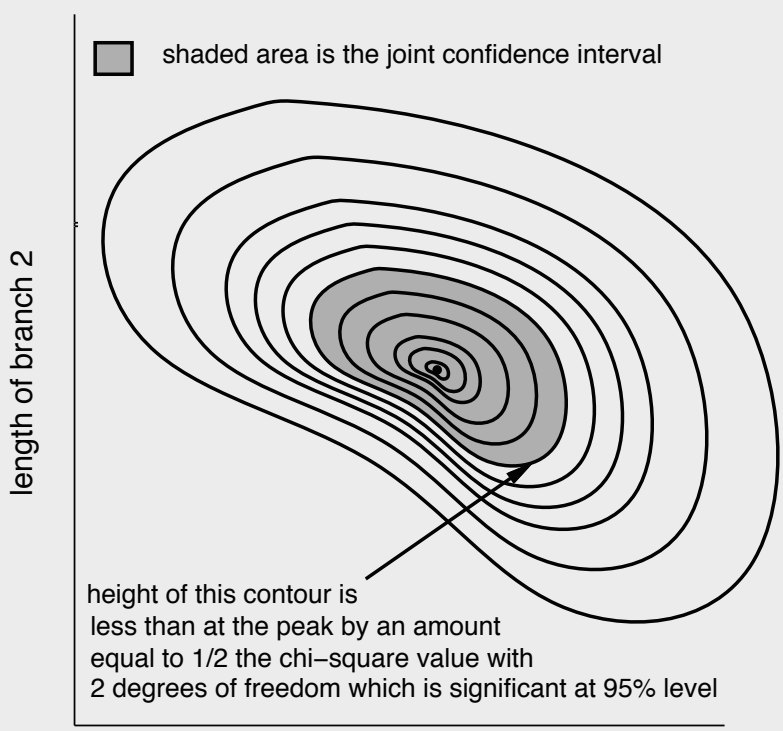
Contours of a log-likelihood surface in two dimensions



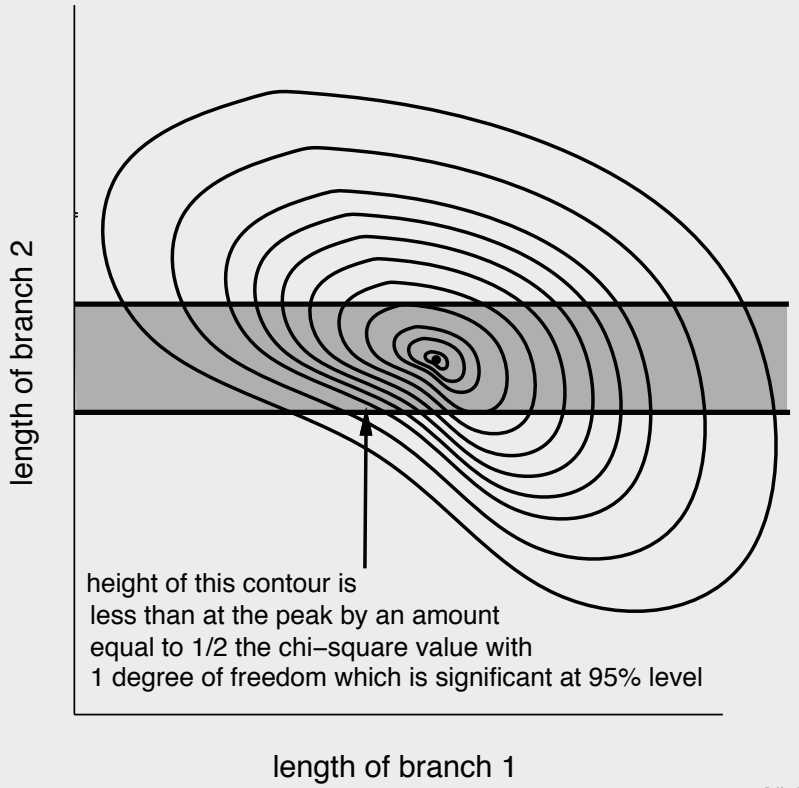
Contours of a log-likelihood surface in two dimensions



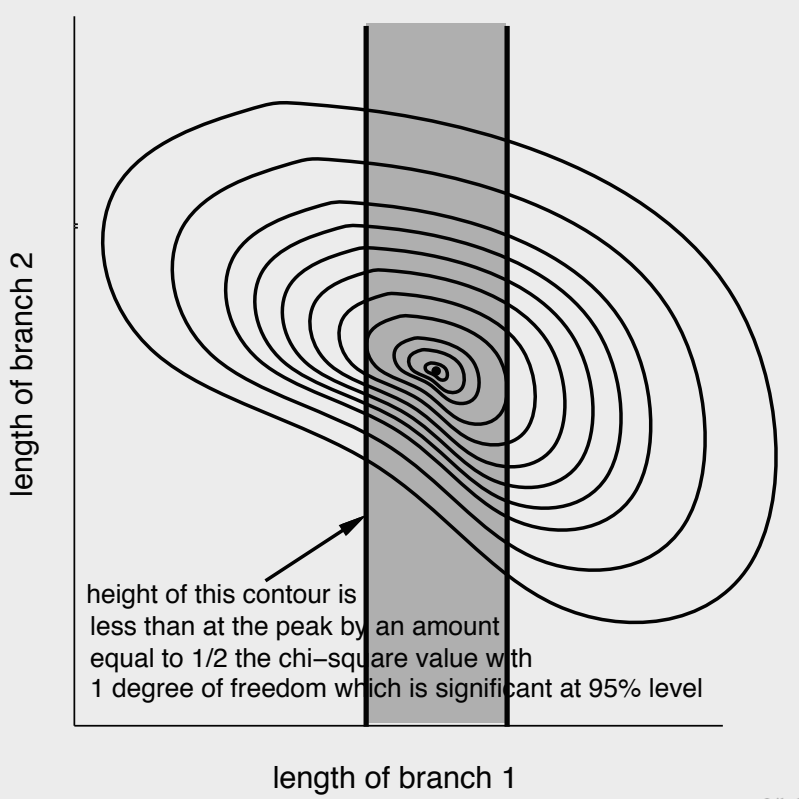
Log-likelihood-based confidence set for two variables



Confidence interval for one variable



Confidence interval for the other variable



Calculating the likelihood of a tree

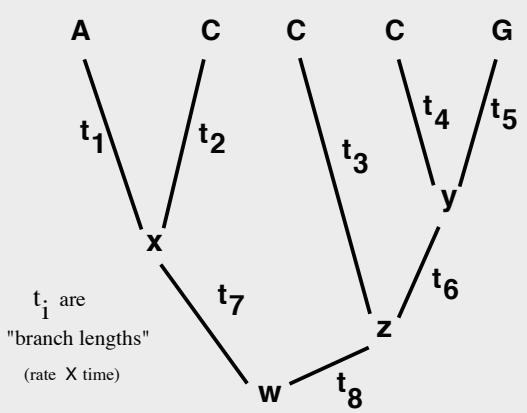
If we have molecular sequences on a tree, the likelihood is the product over sites of the data $D^{[i]}$ for each site (if those evolve independently):

$$L = \text{Prob}(D \mid T) = \prod_{i=1}^{\text{sites}} \text{Prob}(D^{[i]} \mid T)$$

With log-likelihoods, the product becomes a sum:

$$\ln L = \ln \text{Prob}(D \mid T) = \sum_{i=1}^{\text{sites}} \ln \text{Prob}(D^{[i]} \mid T)$$

Calculating the likelihood for site i on a tree



Sum over all possible states (bases) at interior nodes:

$$\begin{aligned} L^{(i)} = & \sum_x \sum_y \sum_z \sum_w \text{Prob}(w)\ \text{Prob}(x \mid w, t_7) \\ & \times \text{Prob}(A \mid x, t_1)\ \text{Prob}(C \mid x, t_2)\ \text{Prob}(z \mid w, t_8) \\ & \times \text{Prob}(C \mid z, t_3)\ \text{Prob}(y \mid z, t_6)\ \text{Prob}(C \mid y, t_4)\ \text{Prob}(G \mid y, t_5) \end{aligned}$$

Calculating the likelihood for site i on a tree

We use the conditional likelihoods: $L_j^{(i)}(s)$

These compute the probability of everything at site i at or above node j on the tree, given that node j is in state s. Thus it assumes something (s) that we don't know in practice – so we compute these for all states s.

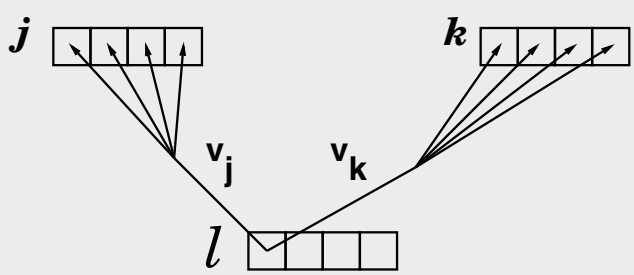
At the tips we can define these quantities: if the observed state is (say) C, the vector of L's is

$$(0, 1, 0, 0)$$

.
If we observe an ambiguity, say R (purine), they are

$$(1, 0, 1, 0), \qquad not \quad (1/2, 0, 1/2, 0)$$

The "pruning" algorithm:



$$L_\ell^{(i)}(s) = \left[\sum_{s_j} \text{Prob}\left(s_j \mid s, v_j\right) L_j^{(i)}(s_j)\right] \times \left[\sum_{s_k} \text{Prob}\left(s_k \mid s, v_k\right) L_k^{(i)}(s_k)\right]$$

(Felsenstein, 1973; 1981).

and at the bottom of the tree:

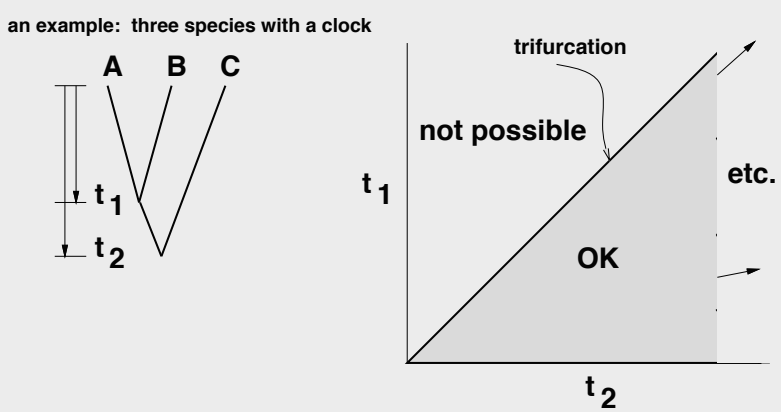
$$L_0^{(i)} = \sum_s \pi_s \, L_0^{(i)}(s)$$

(Felsenstein, 1973, 1981)

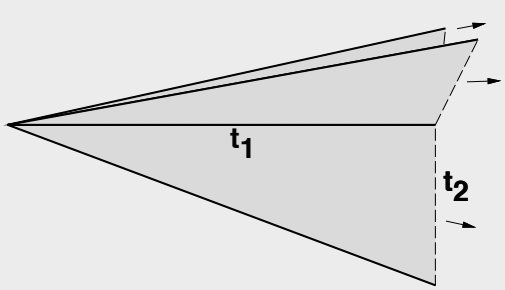
and having gotten the likelihoods for each site:

$$L = \prod_{i=1}^{\text{sites}} L_0^{(i)}$$

What does “tree space" (with branch lengths) look like?

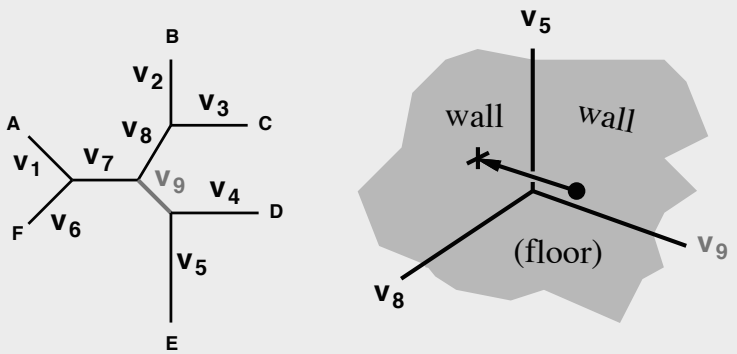


when we consider all three possible topologies, the space looks like:



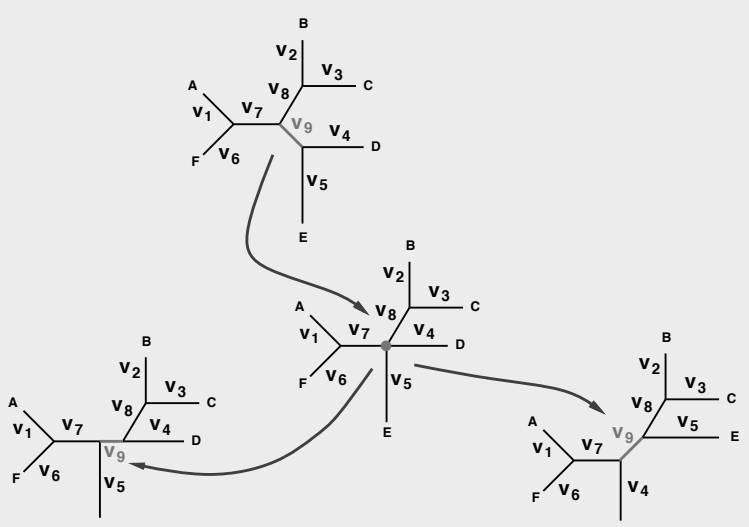
For one tree topology

The space of trees varying all $2n-3$ branch lengths, each a nonegative number, defines an “orthant" (open corner) of a $(2n-3)$-dimensional real space:



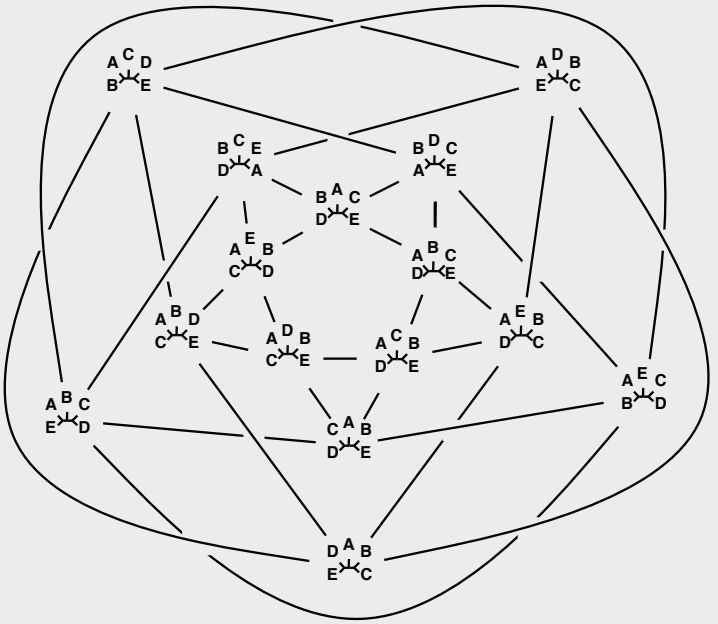
Through the looking-glass

Shrinking one of the $n-1$ interior branches to 0, we arrive at a trifurcation:



Here, as we pass “through the looking glass" we are also touch the space for two other tree topologies, and we could enter either.

The graph of all trees of 5 species



The Schoenberg graph (all 15 trees of size 5 connected by NNI's)

A data example: mitochondrial D-loop sequences

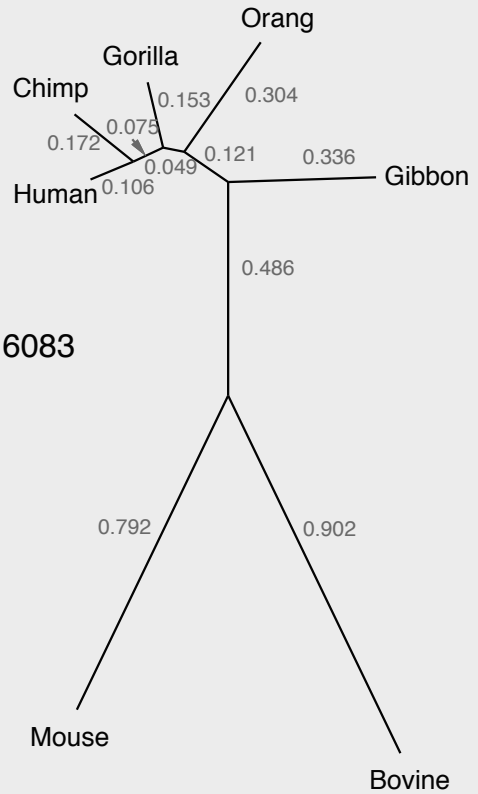
```
Bovine    CCAAACCTGT CCCCACCATC TAACACCAAC CCACATATAC AAGCTAAACC AAAAATACCA
Mouse     CCAAAAAAAC ATCCAAACAC CAACCCCAGC CCTTACGCAA TAGCCATACA AAGAATATTA
Gibbon    CTATACCCAC CCAACTCGAC CTACACCAAT CCCCACATAG CACACAGACC AACAACCTCC
Orang     CCCCACCCGT CTACACCAGC CAACACCAAC CCCCACCTAC TATACCAACC AATAACCTCT
Gorilla   CCCCATTTAT CCATAAAAAC CAACACCAAC CCCCATCTAA CACACAAACT AATGACCCCC
Chimp     CCCCATCCAC CCATACAAAC CAACATTACC CTCCATCCAA TATACAAACT AACAACCTCC
Human     CCCCACTCAC CCATACAAAC CAACACCACT CTCCACCTAA TATACAAATT AATAACCTCC

          TACTACTAAA AACTCAAATT AACTCTTTAA TCTTTATACA ACATTCCACC AACCTATCCA
          TACAACCATA AATAAGACTA ATCTATTAAA ATAACCCATT ACGATACAAA ATCCCTTTCG
          CACCTTCCAT ACCAAGCCCC GACTTTACCG CCAACGCACC TCATCAAAAC ATACCTACAA
          CAACCCCTAA ACCAAACACT ATCCCCAAAA CCAACACACT CTACCAAAAT ACACCCCCAA
          CACCCTCAAA GCCAAACACC AACCCTATAA TCAATACGCC TTATCAAAAC ACACCCCCAA
          CACTCTTCAG ACCGAACACC AATCTCACAA CCAACACGCC CCGTCAAAAC ACCCCTTCAG
          CACCTTCAGA ACTGAACGCC AATCTCATAA CCAACACACC CCATCAAAGC ACCCCTCCAA

          CACAAAAAAA CTCATATTTA TCTAAATACG AACTTCACAC AACCTTAACA CATAAACATA
          TCTAGATACA AACCACAACA CACAATTAAT ACACACCACA ATTACAATAC TAAACTCCCA
          CACAAACAAA TGCCCCCCCA CCCTCCTTCT TCAAGCCCAC TAGACCATCC TACCTTCCTA
          TTCACATCCG CACACCCCCA CCCCCCCTGC CCACGTCCAT CCCATCACCC TCTCCTCCCA
          CATAAACCCA CGCACCCCCA CCCCTTCCGC CCATGCTCAC CACATCATCT CTCCCCTTCA
          CACAAATTCA TACACCCCTA CCTTTCCTAC CCACGTTCAC CACATCATCC CCCCCTCTCA
          CACAAACCCG CACACCTCCA CCCCCCTCGT CTACGCTTAC CACGTCATCC CTCCCTCTCA

          CCCCAGCCCA ACACCCTTCC ACAAATCCTT AATATACGCA CCATAAATAA CA
          TCCCACCAAA TCACCCTCCA TCAAATCCAC AAATTACACA ACCATTAACC CA
          GCACGCCAAG CTCTCTACCA TCAAACGCAC AACTTACACA TACAGAACCA CA
          ACACCCTAAG CCACCTTCCT CAAAATCCAA AACCCACACA ACCGAAACAA CA
```

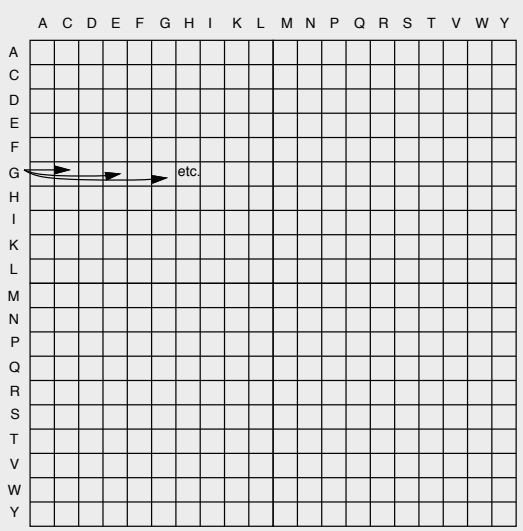
which gives the ML tree



ln L = −1405.6083

Maximum likelihood tree for the Hasegawa 232-site mitochondrial D-loop data set, with Ts/Tn set to 2, analyzed with maximum likelihood (DNAML)

Models with amino acids



Dayhoff PAM model

Jones–Taylor–Thornton model

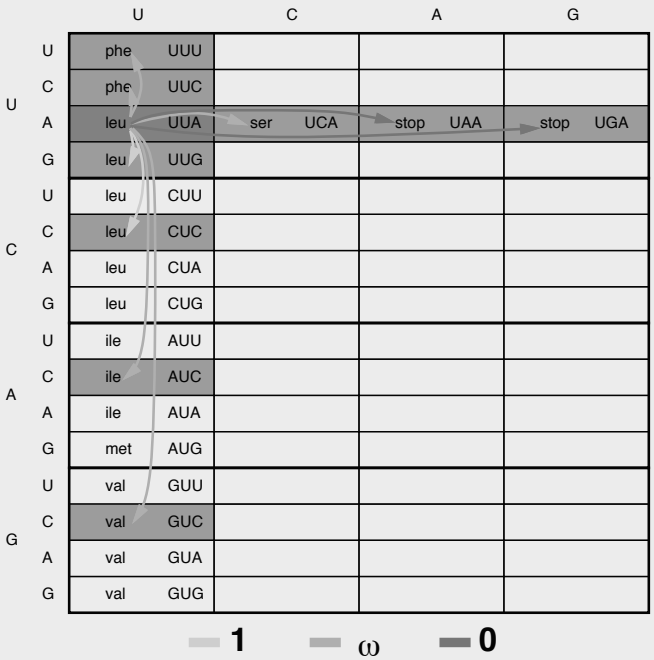
specific models for secondary–structure contexts or membrane proteins

Models adapted from Henikoff BLOSUM scoring

But ... how to take DNA sequence into account? Constraints of code?

Codon models

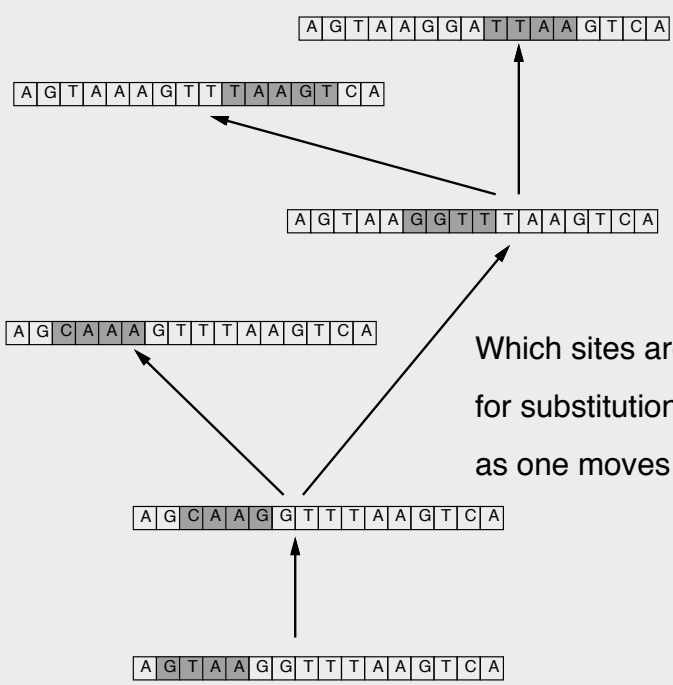
Goldman & Yang, 1994; Muse & Gaut, 1994)



Probabilities of change vary depending on whether amino acid is changing, and to what

Covarion models?

(Fitch and Markowitz, 1970)



Which sites are available

for substitutions changes

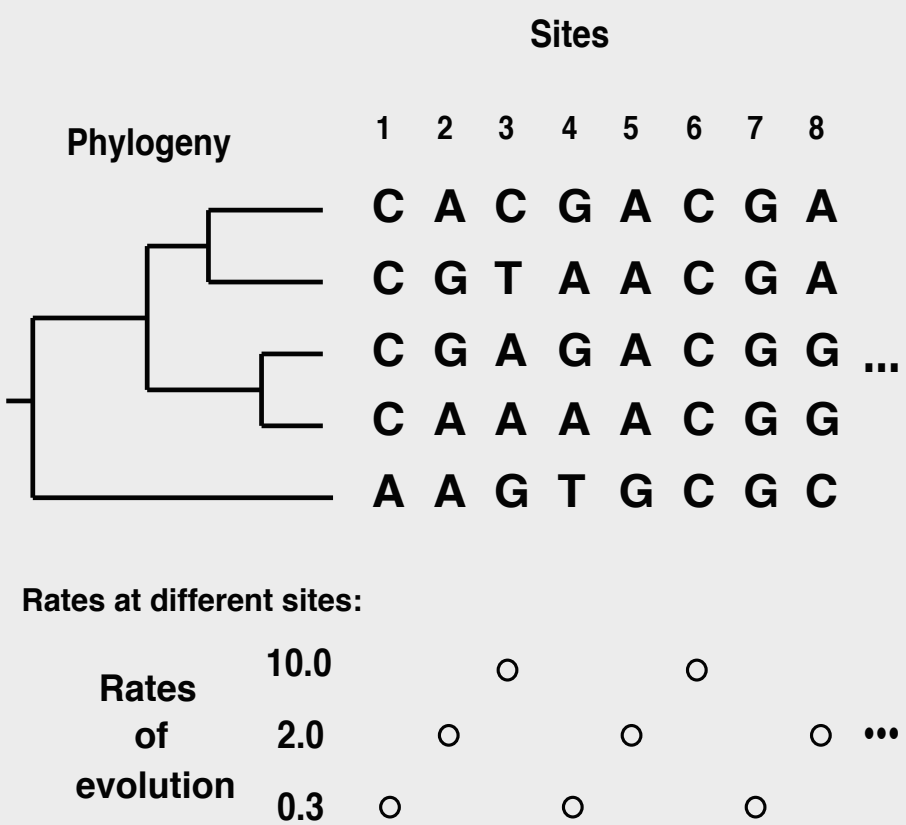
as one moves along the tree

How to calculate likelihood with rate variation

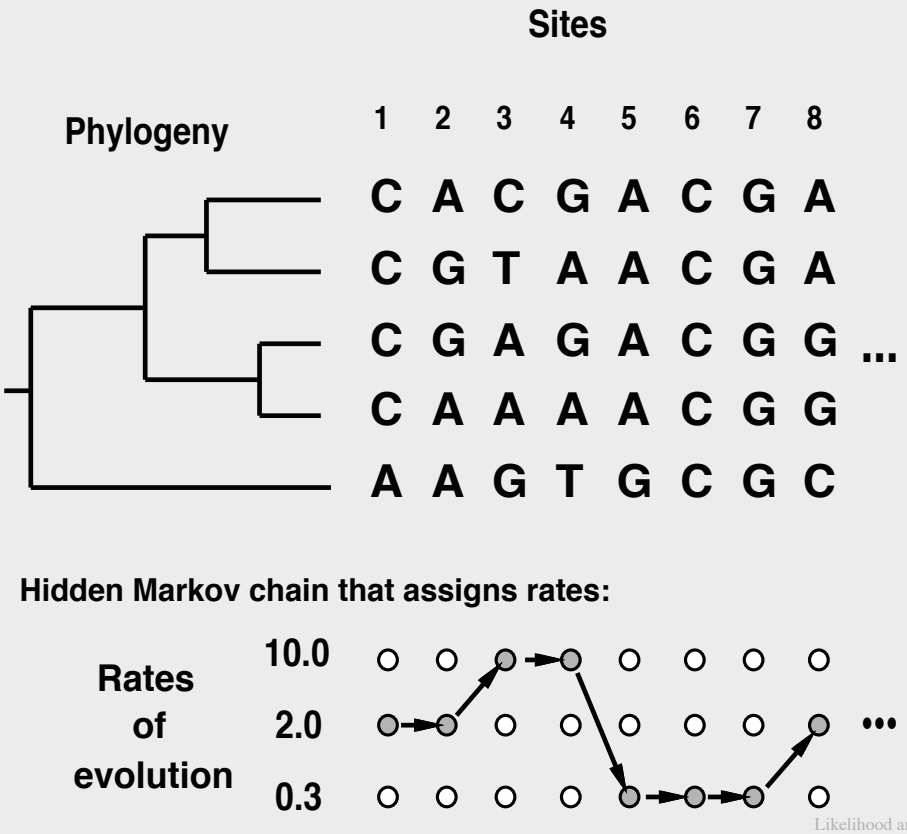
Easy! Since branch lengths always come into transition probability formulas as $r \times t$, can just multiply lengths of branches by the appropriate factor to calculate the likelihood for a site.

(Branch lengths are usually scaled by assuming a rate of 1.)

Rate variation among sites



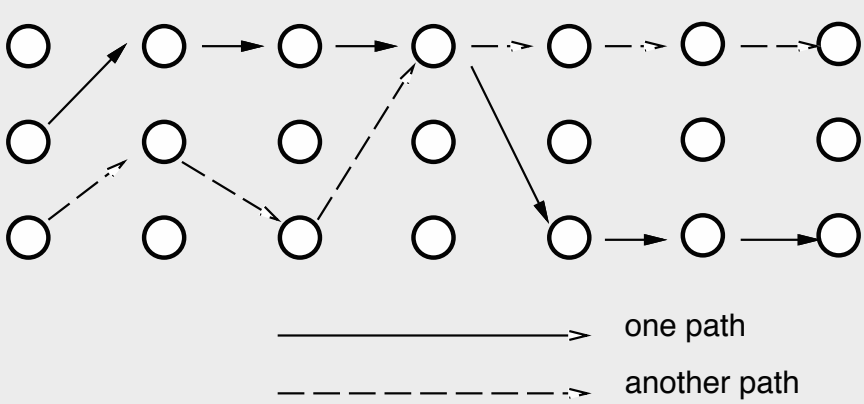
Hidden Markov Model of rate variation among sites



Hidden Markov Models sum up over all paths

The Hidden Markov Chain method sums up likelihoods over all possible paths through the states:

$$\text{Prob (Data I tree)} = \sum_{\text{paths}} \text{Prob(DataI tree, path)}\ \text{Prob(path)}$$

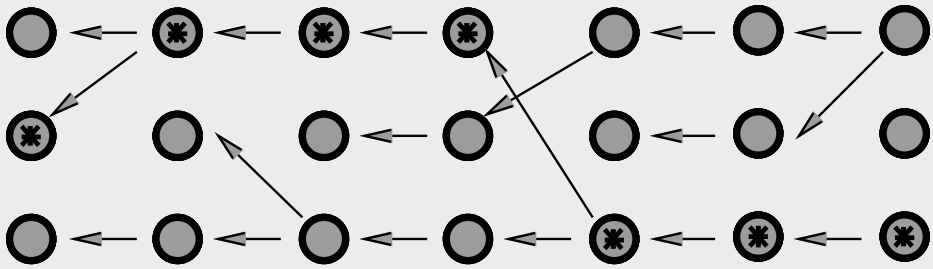


This is done using a recursive algorithm known as the Forwards

The rate combination contributing the most:

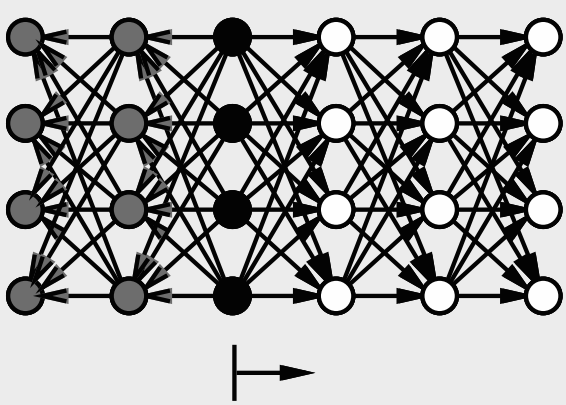
We can leave behind pointers that allow us to backtrack

This can be done by a dynamic programming algorithm called the Viterbi Algorithm, well-known in the HMM literature.



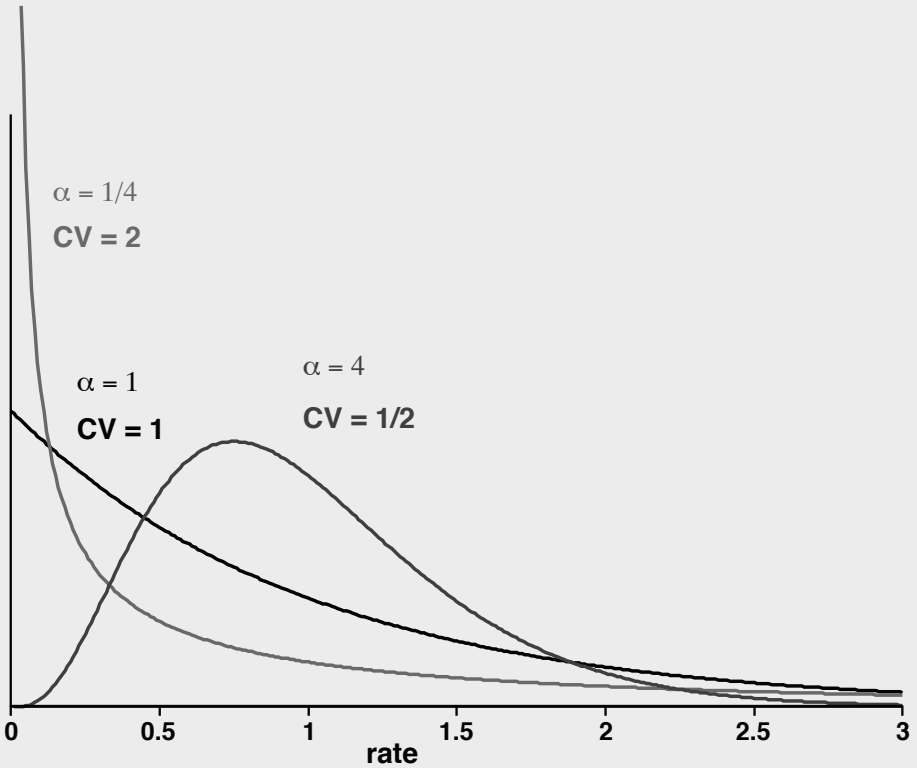
(Of course, this one might account for only 0.001 of the likelihood)

Forwards-Backwards algorithm (marginal probabilities)



The Forwards–Backwards algorithm
can calculate the contribution of one rate
at a given site to the overall likelihood
(a little different from the Viterbi calculation)

The Gamma distribution, used for rates



A numerical example. Cyochrome B

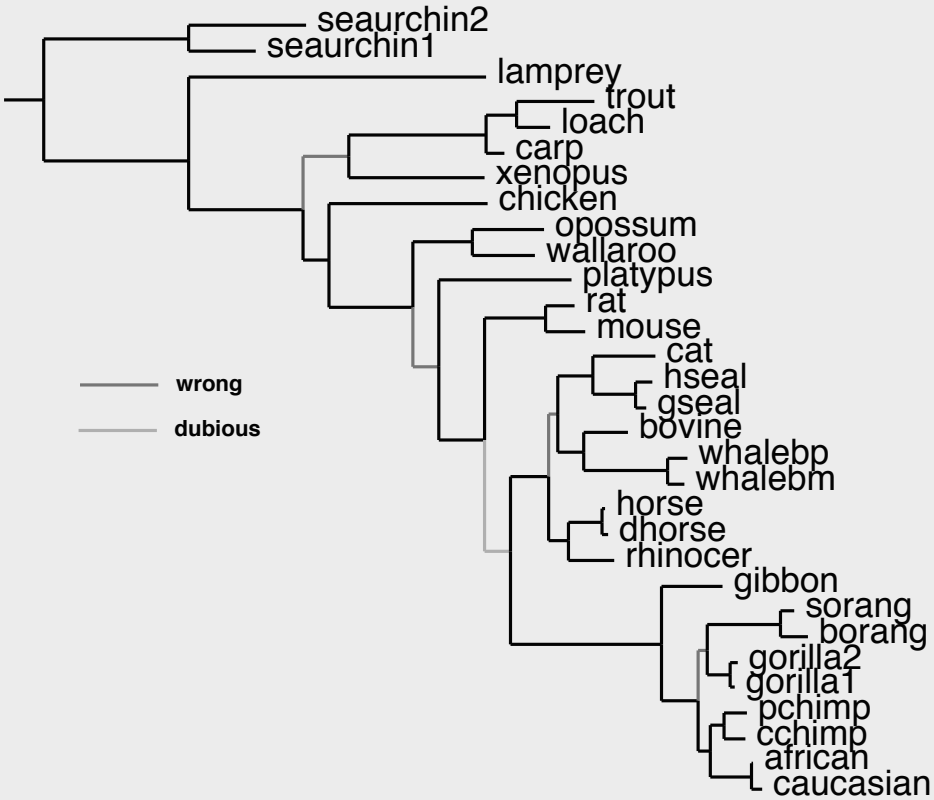
We analyze 31 cytochrome B sequences, aligned by Naoko Takezaki, using the Proml protein maximum likelihood program. Assume a Hidden Markov Model with 3 states, rates:

category	rate	probability
1	0.0	0.2
2	1.0	0.4
3	3.0	0.4

and expected block length 3.

We get a reasonable, but not perfect, tree with the best rate combination inferred to be

The cytochrome B tree from the above run



(It's not perfect).

Rates inferred from Cytochrome B

```
                1333333311 3222322313 3321113222 2133111111 1331133123 112211111
                
african         M-----TPMRK INPLMKLINH SFIDLPTPSN ISAWWNFGSL LGACLILQIT TGLFLAMH
caucasian       .......... .........R .......... .......... ..T....... ..........
cchimp          .......T.. .......... .......... .......... .......... ..........
pchimp          .......T.. .......... .......... ..T....... .......... ..........
gorilla1        .......... T...A..... .......... ..T....... .......... ..........
gorilla2        .......... T...A..... .......... ..T....... .......... ..........
borang          .......... T......... .L........ .......... ......I.TI ..........
sorang          ......ST.. T......... .L........ .......... ......I... ..........
gibbon          .......L.. T......... .L....A... ..M....... .........I ..........
bovine          ......NI.. SH....IV.N A.....A... ..S....... ..I......L ..........
whalebm         ......NI.. TH....I..D A......... ..S....... ..L...V..L ..........
whalebp         ......NI.. TH....IV.D A.V....... ..S....... ..L...M..L ..........
dhorse          ......NI.. SH..I.I... ......A... ..S....... ..I......L ..........
horse           ......NI.. SH..I.I... .......... ..S....... ..I......L ..........
rhinocer        ......NI.. SH..V.I... .......... ..S....... ..I......L ..........
cat             ......NI.. SH..I.I... ......A... .......... ..V..T...L ..........
gseal           ......NI.. TH....I..N .......... .......... ..I......L ..........
hseal           ......NI.. TH....I..N .......... .......... ..I......L ..........
mouse           ......N... TH..F.I... ......A... ..S....... ..V..MV..I ..........
rat             ......NI.. SH..F.I... ......A... ..S....... ..V..MV..L ..........
platypus        .....NNL.. TH..I.IV.. .......... ..S....... ..L...I..L ..........
wallaroo        ......NL.. SH..I.IV.. ......A... .......... ......I..L ..........
opossum         ......NI.. TH....I..D .......... .......... ..V...I..L ..........
chicken         ....APNI.. SH..L.M..N .L....A... .......... .AV..MT..L ...L......
xenopus         ....APNI.. SH..I.I..N .......... ..SL...... ..V...A..I ..........
carp            ....A-SL.. TH..I.IA.D ALV....... .......... ..L...T..L ..........
loach           ....A-SL.. TH..I.IA.D ALV...A... ..V....... ..L...T..L ..........
trout           ....A-NL.. TH..L.IA.D ALV...A... ..V....... ..L..AT..L ..........
lamprey         .SHQPSII.. TH..LS.G.S MLV...S.A. .......... .SL......I ...I......
seaurchin1      -...LG.L.. EH.IFRIL.S T.V...L... L.I....... ..L...T..L ..........
seaurchin2      -...AG.L.. EH.IFRIL.S T.V...L... L.M....... ..L...I.LI ..I.......
```

Rates inferred from Cytochrome B

```
             2223311112 2222222222 2222232112 2222222223 1222221112 333311112
african      PDASTAFSSI AHITRDVNYG WIIRYLHANG ASMFFICLFL HIGRGLYYGS FLYSETWNI
caucasian    .......... .......... .......... .......... .......... .........
cchimp       .......... .......... .......... .......... .......... ...L.....
pchimp       .......... .......... .......... ...L...... .V........ ...L.....
gorilla1     .......... .......... .T........ .......... .......... ..HQ.....
gorilla2     .......... .......... .T........ .......... .......... ..HQ.....
borang       ...T...... .......... .M..H..... ...L...... .......... .THL.....
sorang       .......... .......... .M..H..... .......... .......... .THL.....
gibbon       .........V .......... .......... .......... .......... ...L.....
bovine       S.TT.....V T..C...... .....M.... .........YM .V........ YTFL.....
whalebm      ..TM.....V T..C...... .V........ .........YA .M........ HAFR.....
whalebp      ..TT.....V T..C...... .......... .........YA .M........ YAFR.....
dhorse       S.TT.....V T..C...... .......... ..........I .V........ YTFL.....
horse        S.TT.....V T..C...... .......... ..........I .V........ YTFL.....
rhinocer     ..TT.....V T..C...... .M........ ..........I .V........ YTFL.....
cat          S.TM.....V T..C...... .......... .........YM .V...M.... YTF......
gseal        S.TT.....V T..C...... .......... .........YM .V........ YTFT.....
hseal        S.TT.....V T..C...... .......... .........YM .V........ YTFT.....
mouse        S.TM.....V T..C...... .L...M.... .......... .V........ YTFM.....
rat          S.TM.....V T..C...... .L....Q... .......... .V........ YTFL.....
platypus     S.T......V ...C...... .L...M.... ..L..M.I.. .......... YTQT.....
wallaroo     S.TL.....V ...C...... .L..N..... .....M.... .V...I.... Y..K.....
opossum      S.TL.....V ...C...... .L..NI.... .....M.... .V...I.... Y..K.....
chicken      A.T.L....V ..TC.N.Q.. .L..N..... ..F....I.. .......... Y..K....T
xenopus      A.T.M....V ...CF..... LL..N..... L.F....IY. .......... ...K.....
carp         S.I......V T..C...... .L..NV.... ..F....IYM ..A....... Y..K.....
loach        S.I......V ...C...... .L..NI.... ..F.....Y. ..A....... Y..K.....
trout        S.I......V C..C...S.. .L..NI.... ..F....IYM ..A....... Y..K.....
lamprey      ANTEL....V M..C....N. .LM.N..... .......IYA .....I.... Y..K....V
seaurchin1   A.I.L....A S..C...... .LL.NV.... ..L....MYC .........G SNKI....V
seaurchin2   A.INL....V S..C...... .LL.NV...C ..L....MYC .........L TNKI....V
```


References

Likelihood

Edwards, A. W. F. and L. L. Cavalli-Sforza. 1964. Reconstruction of evolutionary trees. pp. 67-76 in *Phenetic and Phylogenetic Classification*, ed. V. H. Heywood and J. McNeill. Systematics Association Publication No. 6. Systematics Association, London. **[The founding paper for parsimony and likelihood for phylogenies, using gene frequencies]**

Jukes, T. H. and C. Cantor. 1969. Evolution of protein molecules. pp. 21-132 in *Mammalian Protein Metabolism,* ed. M. N. Munro. Academic Press, New York. **[The Jukes-Cantor model, in one formula and a couple of sentences]**

Neyman, J. 1971. Molecular studies of evolution: a source of novel statistical problems. In *Statistical Decision Theory and Related Topics,* ed. S. S. Gupta and J. Yackel, pp. 1-27. New York: Academic Press. **[First paper on likelihood for molecular sequences. Neyman was a famous statistician.]**

Felsenstein, J. 1973. Maximum-likelihood and minimum-steps methods for estimating evolutionary trees from data on discrete characters. *Systematic Zoology* **22:** 240-249. **[The pruning algorithm, parsimony is not same as likelihood]**

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution* **17:** 368-376. **[Making likelihood useable for molecular sequences]**

(more references)

Yang, Z. 1994. Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Molecular Biology and Evolution* **10:** 1396-1401. **[Use of gamma distribution of rate variation in ML phylogenies]**

Yang, Z. 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *Journal of Molecular Evolution* **39:** 306-314. **[Approximating gamma distribution in ML phylogenies by an HMM]**

Yang, Z. 1995. A space-time process model for the evolution of DNA sequences. *Genetics* **139:** 993-1005. **[Allowing for autocorrelated rates along the molecule using an HMM for ML phylogenies]**

Felsenstein, J. and G. A. Churchill. 1996. A Hidden Markov Model approach to variation among sites in rate of evolution *Molecular Biology and Evolution* **13:** 93-104. **[HMM approach to evolutionary rate variation]**

Thorne, J. L., N. Goldman, and D. T. Jones. 1996. Combining protein evolution and secondary structure. *Molecular Biology and Evolution* **13** 666-673. **[HMM for secondary structure of proteins, with phylogenies]**

(more references)

General reading

Felsenstein, J. 2004. *Inferring Phylogenies*. Sinauer Associates, Sunderland, Massachusetts. **[Book you and all your friends must rush out and buy]**

Yang, Z. 2006. *Computational Molecular Evolution*. Oxford University Press, Oxford. **[Well-thought-out book on molecular phylogenies]**

Semple, C. and M. Steel. 2003. *Phylogenetics*. Oxford University Press, Oxford. **[Good for a mathematical audience]**

PHYLIP

Joe Felsenstein

Depts. of Genome Sciences and of Biology, University of Washington

Software for this lab

This lab is intended to introduce the PHYLIP package and a number of major phylogeny methods.

1. You should download the programs from the as-yet-unreleased version 4.0 of PHYLIP. They will be found at

   `http://evolution.gs.washington.edu/sisg/2014/programs/index.html`

   To run the GUI front end of the v4.0 programs need you to have a recent version of Oracle Java installed, for the Java front ends. If you have a Windows machine or a Linux system, it can be downloaded and installed free from `http://java.com` (Mac OS X Java should be good enough).
2. If you have come with a tablet with the iPad or Android operating systems, there is no version of PHYLIP available for that. Instead we will try to give you a URL allowing you to log in to one of our local machines so that, if you have an SSH client, you can open a terminal window and run PHYLIP on our system through that.

PHYLIP

- Distributed since 1980
- Originally in Pascal, now in C
- Intended to provide “basic transportation”
- Intended to provide a wide variety of methods
- Freely available (unless you try to charge others for it)

Advantages of PHYLIP

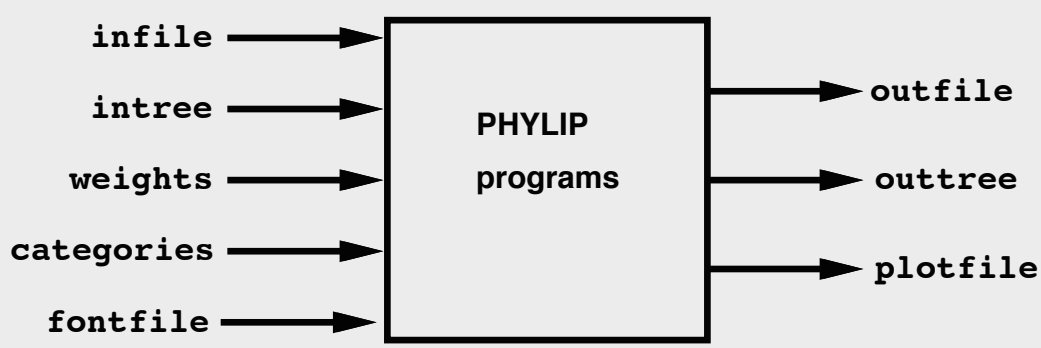
1. Free (in the sense of “free beer”), easily obtainable
2. Runs on all major platforms
3. Very good documentation
4. Lots of people around who know how to use it
5. Often used in teaching about phylogenies.
6. Runs can be automated by using input redirection and command files
7. Support for PHYLIP-format files by many other programs including phylogeny programs and sequence-alignment programs.

Over 31,000 registered users in over 50 countries including: Fiji, Cuba, Papua New Guinea, Iran, Iceland. Large numbers of users in countries such as India, Brazil, Argentina, Russia, and China where even modest cash prices for software can be a major burden.

Disadvantages of PHYLIP

1. Tree search less thorough than some other packages such as `PAUP*`.
2. Much, much slower than packages such as `PAUP*` and `RAxML`
3. Manual steps such as renaming file names can be tedious
4. Still no: codon model (but coming very soon), Bayesian inference.
5. Not as many options available as in other programs
6. Cannot read NEXUS standard files

PHYLIP programs



These are the default file names. If the input files do not exist (or if the output files exist and you choose not to overwrite them), you will be asked for the file name. **This is not a bug.**

Input format for PHYLIP (DNA, Interleaved)

```
    7  112
Bovine     CCAAACCTGT CCCCACCATC TAACACCAAC CCACATATAC AAGCTAAACC AAAAATACCA
Mouse      CCAAAAAAAC ATCCAAACAC CAACCCCAGC CCTTACGCAA TAGCCATACA AAGAATATTA
Gibbon     CTATACCCAC CCAACTCGAC CTACACCAAT CCCCACATAG CACACAGACC AACAACCTCC
Orang      CCCCACCCGT CTACACCAGC CAACACCAAC CCCCACCTAC TATACCAACC AATAACCTCT
Gorilla    CCCCATTTAT CCATAAAAAC CAACACCAAC CCCCATCTAA CACACAAACT AATGACCCCC
Chimp      CCCCATCCAC CCATACAAAC CAACATTACC CTCCATCCAA TATACAAACT AACAACCTCC
Human      CCCCACTCAC CCATACAAAC CAACACCACT CTCCACCTAA TATACAAATT AATAACCTCC

           CCCCAGCCCA ACACCCTTCC ACAAATCCTT AATATACGCA CCATAAATAA CA
           TCCCACCAAA TCACCCTCCA TCAAATCCAC AAATTACACA ACCATTAACC CA
           GCACGCCAAG CTCTCTACCA TCAAACGCAC AACTTACACA TACAGAACCA CA
           ACACCCTAAG CCACCTTCCT CAAAATCCAA AACCCACACA ACCGAAACAA CA
           ACACCTCAAT CCACCTCCCC CCAAATACAC AATTCACACA AACAATACCA CA
           ACATCTTGAC TCGCCTCTCT CCAAACACAC AATTCACGCA AACAACGCCA CA
           ACACCTTAAC TCACCTTCTC CCAAACGCAC AATTCGCACA CACAACGCCA CA
```


Format for trees in tree files (Newick standard)

(Mouse:0.87231,Bovine:0.49807,(Gibbon:0.25930,(Orang:0.24166,
(Gorilla:0.12322,(Chimp:0.13846,
Human:0.08571):0.06026):0.04405):0.10815):0.39538);

More than such tree can be placed end-to-end in the same tree file.

The Newick standard was defined by an informal standards committee in 1986. It is described on this web page:

`http://evolution.gs.washington.edu/phylip/newicktree.html`

It is very widely used by phylogeny programs. For example, although the tree block format of the NEXUS file format is a competitor, inside of it one will actually find ... a Newick tree.

PHYLIP guide

A useful guide to using PHYLIP with molecular sequences has been produced by Jarno Tuimala. It can be downloaded as a PDF from

`http://koti.mbnet.fi/tuimala/oppaat/phylip2.pdf`

or using the link to it on the main PHYLIP web page.

For more information on many other programs

... at my PHYLIP web site there is a master list of over 390 phylogeny programs, with descriptions and links.

To find it simply put the phrase “Phylogeny Programs” into your favorite search engine.

However, it is not really up-to-date. I have had to stop work on it as I have no one to help me on that.

What to do in the PHYLIP likelihood lab exercise

1. Get a DNA or protein sequence data set of aligned sequences. You can use one of the ones provided by the course if you wish. They are also at
   `http://evolution.gs.washington.edu/sisg/2014/data/`
2. Make a copy the data file as file `infile` , and then run either `Dnaml` or `Proml`, whichever is appropriate. Use the `R` to do a "Gamma distributed rates" analysis and then the `A` options to set it to a mean block length of about 3. After you accept the menu settings, you will be asked for a coefficient of variation of rates (you could set this at 2.0) and for the number of rate categories used to approximate the Gamma distribution (about 5-6 would be good).
3. Look at the tree by looking at the output file `outfile` (when you examine that file, you will need to make sure the font is a fixed-width one such as Courier) and also by renaming `outtree` to `intree` and then using `Drawgram` (perhaps with font file `font1`). You can also try `Drawtree`. (In using these, when you get a preview of the graph, use the `File` menu to choose whether you want to change settings. The final plot will be called `plotfile` .

More to do: the PHYLIP distance lab exercise

Use your data set and analyze it by the Neighbor-Joining method:

1. Make a copy of your sequences and call that file `infile`
2. Run `Dnadist` or `Protdist`, whichever is appropriate.
3. The distance matrix is in file `outfile`
4. Rename that `infile`
5. Run `Neighbor`, using the default options except maybe the outgroup-rooting option.
6. The output file `outfile` will show your tree, and the output tree file `treefile` has the Newick-format representation of it. Save them by renaming them. When examining the output file, use a constant-width font to avoid distortion of the tree.

More to do: the PHYLIP bootstrap lab exercise

Use that distance matrix method to do a bootstrap analysis:

1. (use `Seqboot`, then renaming `outfile` to `infile`, (You can use 1000 replicates if you have DNA sequences (use menu option R), but don't do 1000 replicates for a protein data set as this will be too slow). When asked for the random number seed, provide any odd number whose last two digits give a remainder of 1 when divided by 4 (for example, they might be 45).
2. Use that `infile` of many data sets as an input for `Dnadist` or `Protdist`, using the M (Multiple input data sets) option (with multiple data sets, not weights).
3. The multiple distance matrices are now in file `outfile`. Rename that to `infile`.
4. Now run program `Neighbor`, making sure to set the multiple data sets option M and provide the number of the bootstrap replicate distance matrices.
5. Rename the output file `outtree` (which will contain multiple bootstrap estimates of the tree) to `intree`.
6. Run program `Consense` which makes an Extended Majority-Rule Consensus Tree from these trees.
7. Look at the consensus tree by examining `outfile`, or renaming `outtree` to `intree` and running either `Drawgram` or `Drawtree`.
8. The branch lengths of this consensus tree are weird (they reflect levels of bootstrap support rather than amounts of change. Can you figure out a way, using the original sequences and the consensus tree and menu option U (User-defined tree) in the likelihood program, to get more reasonable branch lengths in that tree?

PAUP* Lab

Note: This computer lab exercise was written by Paul O. Lewis. Paul has graciously allowed Mark Holder to use and modify the lab for the Summer Institute in Statistical Genetics. Thanks, Paul!

In this computer lab you will learn the basics of using the computer program PAUP* for phylogenetic analyses of nucleotide sequences. Versions of PAUP* exist for several different operating systems (MacIntosh, Windows, Linux, etc.), with the MacIntosh version being the most flexible and user-friendly. We will be using the Windows version today. The graphical user interface (GUI) of this Windows version is not as well developed as the GUI for the MacIntosh version, but it is exactly the same program and produces results that are identical to the MacIntosh version.

The PAUP* Home Page is the best place to go for continuing updates on the progress being made toward the final release, and for information about purchasing the program: `http://paup.csit.fsu.edu/`

You can work through this tutorial at your own pace, asking questions whenever something needs to be clarified. Please let us know if you think another approach would be better, and if anything about this tutorial is unclear. The goals for this tutorial are to:

- Become familiar with the NEXUS data file format used by PAUP* (as well as several other prominent phylogeny programs such as Mesquite and MrBayes)
- Learn how to conduct various types of searches (exhaustive, branch-and-bound, heuristic using NNI and TBR branch swapping, and algorithmic approaches such as star decomposition and stepwise addition)
- Learn how to set up PAUP* to perform analyses under several different optimality criteria (maximum parsimony, minimum evolution, least squares, and maximum likelihood)
- Learn how to set up PAUP* for several different nucleotide substitution models, and to obtain maximum likelihood estimates of parameters of these models
- Learn how to create PAUP blocks in the data file so that analyses can be performed in batch mode (also learn why you might want to do this)

PAUP* Tutorial

Questions that you should be able to answer from looking at the output are in *italics*. Answers to the questions are provided in footnotes. If you do not understand one of these questions, or need help figuring out the answer, please do not hesitate to raise your hand.

About the data file

The tutorial uses one data file, `algae.nex`, which has been provided to you. This data set is distributed as one of the sample files for the program SplitsTree (`http://www.splitstree.org/`). It contains 16S rRNA sequences for a cyanobacterium (*Anacystis*), a chromophyte alga (*Olithodiscus*), a euglenoid protist (*Euglena*), and six green plants, including two green algae (*Chlorella* and *Chlamydomonas*), a liverwort

(*Marchantia*), a monocotyledonous angiosperm (*Oryza*, rice) and a dicotyledonous angiosperm (*Nicotiana*, tobacco).

This data set was used in a 1994 paper by Lockhart et al. to show how common models used in reconstructing phylogenies fail when confronted by convergence in nucleotide composition. The problem is that the common models assume stationarity of the substitution process, which leads to the assumption that base frequencies do not change across the tree. Thus, things can go wrong when the base frequencies do change from lineage to lineage, and things can go really wrong when unrelated groups tend to have similar base compositions. In this case, *Euglena* should group with the green plants because its chloroplast (whence the 16S rDNA is obtained) is homologous to green plant chloroplasts. However, as you will see, it has a strong tendency to group with the unrelated chromophyte *Olithodiscus* because of similarities in base composition. The complete reference to the Lockhart paper is

Lockhart, P. J., M. A. Steel, M. D. Hendy, and D. Penny. 1994. Recovering evolutionary trees under a more realistic model of sequence evolution. *Molecular Biology and Evolution* **11**: 605-612.

Tutorial begins here

1. Start PAUP* by double-clicking its icon. After PAUP* starts, it will present you with an **Open** dialog box. Navigate to the file `algae.nex` and click the **Open/Execute** button when the file's name has been selected.

2. Before doing any analyses, let's take a look at the data file PAUP* just executed [1] Open the `algae.nex` file for editing by choosing **File | Open...** from the main menu (**Ctrl-O** does the same thing), clicking the **Edit** radio button in the **File Open Mode** group, selecting the file name, and finally clicking the **Open/Edit** button.

   The Nexus data file format has been adopted by several phylogenetic analyses programs, including PAUP*, MacClade, Mesquite, SplitsTree, TreeView, and MrBayes, among others. Nexus data files always begin with `#nexus`, and the remainder of the file is divided into units known as **blocks**. Nexus files are (for the most part) case-insensitive, so `#nexus`, `#Nexus` and `#NEXUS` are synonyms. This file has two blocks: a **taxa block** and a **characters block**. Each block begins with the keyword **begin** and ends with the keyword **end**. Each block comprises **commands**, all of which end in a semicolon (;). Note that each block automatically has two commands: the **begin** command and the **end** command. Some commands are quite long, taking up many lines in the file (e.g., the **matrix** command in the characters block), but the extent of each command can be surmised by simply looking for that terminating semicolon. A mistake made by most everyone when first constructing a Nexus data file is to forget to end every command with a semicolon. If you do this, PAUP* will report an error when attempting to read in the data file.

   What are the four commands comprising the TAXA block? [2]

   Nexus files can contain comments. Comments are text surrounded by square brackets. Comments that you wish to have printed out in the output look like this:

[1] PAUP* uses the term **execute** to mean reading a data file for the purpose of storing the data contained therein. The term **edit** is used for the opening of a data file when the purpose is to view/modify its contents and not to perform analyses.

[2] The four commands comprising the TAXA block are: (1) "begin taxa;"; (2) "dimensions ntax=8;"; (3) "taxlabels [1] Tobacco [2] Rice ... [8] Olithodiscus;" and (4) "end;".

```
[!This is a printed comment]
```

If that initial exclamation point (!) is missing, PAUP* will simply ignore the comment entirely.

Can you find the single printed comment both in the data file and in PAUP's output?* [3]

Here is a brief explanation of some of the commands present in this data file:

Command	Meaning
`dimensions ntax=8;`	Data are provided for eight taxa
`taxlabels Tobacco Rice ··· Olithodiscus;`	Provides names for the eight taxa
`dimensions nchar=920;`	There are 920 nucleotide sites for each sequence (taxon)
`format datatype=RNA missing=? gap=- labels interleave;`	These are RNA sequences; the symbol `?` is used for missing data and `-` for gaps introduced for purposes of aligning the sequences; taxon labels are provided before each sequence; and the data are interleaved, which means that sequences are broken up into short segements for readability

For a complete reference to the Nexus data file format, please see the following paper:

> Maddison, David R., Swofford, David L. and Maddison, Wayne P. 1997. NEXUS: an extensible file format for systematic information. *Systematic Biology* **46**: 590-621

Close the editor now by clicking on the button labeled with an × in the upper right-hand corner of the editor window.

3. **Parsimony Analysis.** For n taxa, there are

$$\frac{(2n-5)!}{(n-3)!\,2^{n-3}}$$

possible unrooted, fully-bifurcating, distinct tree topologies. For our 8 taxa, this formula produces $10,395$. This is not astronomical, so we will use an exhaustive search in combination with the maximum parsimony criterion for our first analysis. Type the following commands into the edit control near the bottom of PAUP*'s main window (you can either type each one in, pressing the enter key after each, or type them both in and press the enter key only after both have been entered – the semicolons serve to keep the commands separated):

```
set criterion=parsimony;
alltrees;
```

[3] The comment is "[! Example of RNA data]".

Note: the search status dialog box has a single button that says "Stop" while the search is in progress and "Close" when the search is done. If the button says "Close", this means PAUP* is no longer doing any work, but is simply sitting there waiting for you to press the button!

The first command is not strictly necessary, since the parsimony criterion is selected by default in PAUP*; however, defaults can change, so it is wise to be explicit. Look at the output (you may have to scroll up some to see the beginning) and answer the following questions before proceeding:

How many tree topologies did PAUP look through? Was this what you expected? How many trees had the best score? How many steps longer than the most parsimonious tree(s) is the next best tree? How long, in your estimation, would it take to perform an exhaustive search for 16 taxa? (Note that the answer is not twice the amount of time required for 8.)* [4]

Use the command `showtrees all;` to tell PAUP* to show you all trees currently stored in memory. Although these are unrooted trees, they appear to be rooted at tobacco and rice, which is not a very useful rooting. Use the following command to tell PAUP* to root trees shown from now on using *Anacystis nidulans* as the outgroup:

```
outgroup Anacystis_nidulans;
```

Note the use of the underscore character (_) to make *Anacystis nidulans* a single word. PAUP* uses blank spaces (including tab and newline characters) to delimit individual items (called words or tokens). If we had left out the underscore character, PAUP* would have attempted to make both *Anacystis* and *nidulans* outgroups, but would have run into problems since *nidulans* is not a taxon. Another way to handle embedded blank spaces is to enclose the entire token in single quotes (e.g., `'Anacystis nidulans'` would also have worked).

One of the interesting features of this data set is that many analysis methods do not produce an estimated tree in which all 6 chlorophyll-b-containing taxa are together. There is considerable evidence from many sources that this should indeed be the case, so throughout this tutorial be on the watch for methods that produce a tree in which this **chlorophyll-a/b clade** is separated by an internal branch from the other two taxa: *Olithodiscus* (a chlorophyll-a/c-containing chromophyte alga) and *Anacystis* (a cyanobacterium containing only chlorophyll-a plus phycobilin accessory pigments).

Does standard (Fitch) parsimony produce the chlorophyll-a/b clade? If not, what taxon is separated from the rest of the chlorophyll-a/b clade? [5]

4. **Branch-and-bound.** Although we have already done an exhaustive search and thus have no need to perform a branch-and-bound search, feel free to use the command below to perform a branch-and-bound search anyway.

   ```
   bandb upbound=470;
   ```

   Note that an upper bound was supplied in this case. This upper bound is the score of some tree (not necessarily the best) containing all of the taxa. If you leave out the upper bound specification, PAUP* will compute an initial upper bound for you using stepwise addition. Notice how many fewer trees have to be evaluated when you used a branch-and-bound search.

[4] PAUP* looked through 10395 trees, which is the number of unrooted binary trees for 8 taxa. The best score was 411, and 2 trees had this score. The next best tree is 414 steps long, and only 1 tree had this score. On my laptop, it required 0.04 seconds to evaluate the 10,395 possible trees for 8 taxa, but there are literally trillions of possible trees for 16 taxa ($27!/(13!2^{13})$ = 213,458,046,676,875), so it would take some 26 years to score all possible trees for 16 taxa.

[5] *Euglena* is grouped with *Olithodiscus*, not with the other a/b taxa

5. **Distance Analyses: Neighbor-joining.** To obtain the neighbor-joining (NJ) tree, simply issue the command `nj`. This should produce the same tree as parsimony, although it will look a bit different since PAUP* shows the tree as a **phylogram** (with branch lengths proportional to the estimated number of substitutions) rather than the **dendrogram** format (the default for the `showtrees` command used earlier). To check that it is indeed equivalent to one of the most-parsimonious trees, issue the `pscores all` command to obtain parsimony scores for all trees in memory (but there should be just 1 tree in memory). The treelength shown should match the one obtained earlier in our parsimony searches.

6. **Distance Analyses: NNI heuristic search using ME criterion and JC distances.**

   To save time and keep this lab from being too tedious, I am combining several goals in each analysis. Note that PAUP* provides considerable flexibility in allowing you to mix-and-match optimality criteria, search strategies, and substitution models. The combinations used in this tutorial should not be viewed as necessarily the "best" combination.

   In this case we will combine a heuristic search strategy (using NNI branch swapping) with the minimum evolution (ME) optimality criterion and will use the Jukes-Cantor model for obtaining the pairwise distance matrix. Here are the commands necessary:

   ```
   set criterion=distance;
   dset distance=jc objective=me;
   hsearch swap=nni;
   ```

   The `dset` command is used for changing the **distance settings**. You may be wondering at this point "How do I know which distance settings to change?" To get PAUP* to give you a summary of any command, simply type the command's name followed by a space and then a question mark:

   ```
   dset ?;
   ```

   Doing so provides a list of all the options available for that command as well as their current settings.

   What is the only other option (besides `ME`*) for the setting called* `objective`*? How would you instruct PAUP* to use HKY85 distances?* [6]

   Get PAUP* to show you a phylogram of the best tree found using the command

   ```
   describetree 1 / plot=phylogram;
   ```

   Does this distance tree separate out the chlorophyll-a/b clade? What method did PAUP use for obtaining the starting tree? Use the method described above (i.e., the question mark approach) to obtain a list of options for the* `hsearch` *command. How would you instruct PAUP* to use the stepwise addition method for obtaining the starting tree?* [7]

   Note that NJ is an approximation to the algorithmic approach known as **star decomposition** in conjunction with the minimum evolution (ME) optimality criterion. Thus, an analysis similar to neighbor-joining but which is not approximate could be conducted as follows:

[6] The only other option for `objective` is `LSFit`. To use HKY85 distances, you would issue the command `dset distance=HKY85`

[7] No, *Euglena* is still with *Olithodiscus*. To use stepwise addition to obtain the starting tree, you would issue the command `hsearch start=stepwise`

```
set criterion=distance;
dset objective=me;
stardecomp;
```

Note that the `NJ` command uses the current distance settings to determine which model to use in computing the pairwise distance matrix.

7. **Distance Analyses: TBR heuristic search using LS criterion and LogDet distances.**

   This distance analysis will use a different optimality criterion (least-squares, or LS, rather than ME) and a different model for obtaining the pairwise distances (LogDet rather than JC). We will also use TBR branch swapping rather than NNI.

   ```
   dset distance=logdet objective=lsfit;
   hsearch swap=tbr;
   describetrees 1 / plot=phylogram;
   ```

   Note that this time the chlorophyll-a/b clade is intact! What is different about this analysis? The most important difference is the model used to compute the pairwise distances. LogDet is a very good model to use when nucleotide composition varies across the tree. Use PAUP*'s `basefreq` command to examine the nucleotide composition of all the sequences in this data file. Since *Euglena* has been the taxon jumping out, and it tends to join with *Olithodiscus* when it is misbehaving, look specifically at the nucleotide composition of *Euglena* and *Olithodiscus* compared to everything else. *Are these two taxa similar to each other in nucleotide composition?* If so, then most methods may be placing them together simply because of the similarity in their nucleotide composition.

8. **Likelihood Analyses: HKY85 model combined with stepwise addition.** For this analysis, we will use the maximum likelihood criterion and obtain a tree using the stepwise addition method. Stepwise addition is normally not considered an end in itself; it is most often used to obtain a **starting tree** for input into heuristic searches. Likelihood runs can be very time consuming if the number of taxa is large, and it is nice to be able to separate the process of obtaining the starting tree from the heuristic search proper. Stepwise addition trees can also come in handy for estimating parameters of the substitution models. These parameters can then be fixed at their estimated values for the duration of the search rather than estimated anew for each tree examined during the search. The latter is of course better, but the former may be the only practical course of action.

   ```
   set criterion=likelihood;
   lset nst=2 basefreq=empirical variant=hky tratio=estimate rates=equal;
   hsearch start=stepwise addseq=random swap=none nreps=1;
   ```

   This set of commands requires some explanation. PAUP* does not allow you to specify the substitution model by name as it does for distance analyses. You must specify instead the *characteristics* of the model you wish to use. Let's look first at the line specifying the model, then we will tackle the line specifying the search strategy:

   ```
   lset nst=2 basefreq=empirical variant=hky tratio=estimate rates=equal;
   ```

This line provides **likelihood settings** corresponding to the HKY85 model. This model allows **unequal base frequencies**, and `basefreq=empirical` instructs PAUP* that the **empirical base frequencies** should be used. The empirical base frequencies are simply the base frequencies computed from the data matrix without reference to any tree topology. These base frequencies work about as well as the maximum likelihood estimates of the base frequencies, which PAUP* can also compute (using `basefreq=estimate`), but estimating the base frequencies will add a considerable amount of time to any analysis. In either case, the base frequencies employed will almost certainly be unequal, which fits the HKY85 model and excludes models such as JC and K2P which assume all bases are equally frequent (i.e., $\pi_A = \pi_C = \pi_G = \pi_T = 0.25$). The `nst=2` part tells PAUP* to use a model employing 2 substitution rate parameters (the K2P, F84, and HKY85 models all allow transitions to occur at a potentially different rate than transversions, hence two substitution rates). The `variant=hky` statement is necessary since both the F84 model and the HKY85 model allow unequal base frequencies and transition/transversion bias. The `tratio=estimate` instructs PAUP* to estimate the transition/transversion rate for every tree. Finally, the `rates=equal` statement says that PAUP* is to assume that all sites evolve at the same overall substitution rate.

```
hsearch start=stepwise addseq=random swap=none nreps=1;
```

Specifying `swap=none` prevents PAUP* from actually carrying out the heuristic search. Instead, it will create a stepwise addition starting tree (`start=stepwise`) using a random sequence to determine the order in which taxa are added (`addseq=random`) and then simply stop, holding that stepwise addition tree in memory. The `nreps=1` statement is necessary to prevent PAUP* from performing its default number of 10 independent search replicates.

Run the `lset` and `hsearch` commands. Then use `showtrees` to see the estimated phylogeny and use the following command to obtain the maximum likelihood estimate of the transition/transversion ratio and rate ratio:

```
lscores 1;
```

This command instructs PAUP* to compute the likelihood score [8] for the 1st. tree in memory. This forces PAUP* to also spit out estimates of all parameters you asked it to estimate, which in this case is just the transition/transversion ratio.

What model did PAUP say it used for this analysis? What is the maximum likelihood estimate of the transition/transversion ratio and rate ratio? Did this analysis result in an estimated phylogeny that separates out the chlorophyll-a/b clade?* [9]

You probably noticed that this analysis took considerably longer than any analysis performed thus far, even though we specifically avoided doing a search! The main reason this one took so long was because we instructed PAUP* to estimate the transition/transversion ratio (tratio) for every tree examined during the analysis. Let's fix the value of tratio to the maximum likelihood estimate just obtained and try again to see how much faster things go:

```
lset tratio=previous;
hsearch;
```

[8] There are corresponding commands for obtaining scores for the parsimony (`pscores`) and distance (`dscores`) criteria.

[9] PAUP* reports "These settings correspond to the HKY85 model." The transition/transversion ratio was estimated to be 1.910491, whereas the transition/transversion rate ratio was estimated to be 3.637613. No, this analysis did not separate the chlorophyll-b-containing taxa from the other two.

In my case, it took about 4.5 times longer to do the search when estimating the tratio for every tree examined, so fixing the tratio at some reasonable value sped up the analysis considerably. It is common to estimate parameters on a tree that can be obtained quickly (e.g., a NJ tree), then fix the values of those parameters for the duration of a search. If you employ this strategy, it is probably a good idea to re-estimate the parameters after the search to make sure they are not appreciably different. To be safe, it is best to repeat the search using the newer estimates. If the second search returns the same tree as the first search, then you know that the initial estimates were reasonable.

9. **Maximum Likelihood Analyses: Using the GTR model.**

The HKY85 model is rather inflexible in that it allows for only two classes of substitutions – transitions and transversions. If the two types of transitions (purine $\leftrightarrow$ purine vs. pyrimidine $\leftrightarrow$ pyrimidine) occur at different rates, or if the four types of transversions occur do not all occur at the same rate, the HKY85 model might not capture important aspects of the evolution of the sequences under study. The General Time Reversible, or GTR, model allows the six possible classes of substitutions (i.e., $A \leftrightarrow C$, $A \leftrightarrow G$, $A \leftrightarrow T$, $C \leftrightarrow G$, $C \leftrightarrow T$, and , $G \leftrightarrow T$) to all occur at potentially different rates. This adds five extra parameters to the JC model rather than just the one added by the HKY85 model, and estimating all of these for each tree examined would be quite time consuming (although not that impractical for this data set because of the relatively small number of taxa). Our approach will be to estimate the five extra relative rate parameters using the tree currently in memory, then fix these values for purposes of conducting the heuristic search.

To set up the GTR model and estimate the relative rate parameters, use these commands:

```
lset nst=6 rmatrix=estimate;
lscores 1;
```

The output should show something like this:

```
Tree             1
------------------
-ln L   3251.02830
Rate matrix R:
  AC       0.62353
  AG       1.87194
  AT       0.82691
  CG       0.32529
  CT       3.68057
  GT       1.00000
```

The **R-matrix** contains the estimates of the relative rates (rows and columns both are in the order A, C, G, T). PAUP* always reports the rate of the $G \leftrightarrow T$ change as 1. Since these are relative rates having meaning only in comparison to each other, any one of them may be set to some arbitrary value or some overall constraint may be applied (i.e., their mean could be made 1.0). In PAUP*, the former method is used and the last one is arbitrarily set to the value 1. The way to interpret these relative rates is best demonstrated by example: all other things being equal, the rate at which $C \leftrightarrow T$ changes occur is 3.68 times the rate at which $G \leftrightarrow T$ changes occur, and $A \leftrightarrow C$ changes are occurring at about a third the rate of $A \leftrightarrow G$ changes ($0.62353/1.87194 = 0.33309$).

The **Q-matrix** takes account of the effects of the base frequencies. To see the Q-matrix, re-issue the `lscores` command, but this time with a couple of options:

```
lscores 1 / longfmt showqmatrix;
```

The output should now show the R-matrix (displayed in matrix format rather than a simple list) followed by the Q-matrix. In the GTR model, the rates at which the various changes occur are a function not only of these relative rates, but also involve the frequency of the base being substituted *to*. For example, $Q_{TC} = \mu R_{TC} \pi_C$, where μ is the overall rate of substitution, R_{TC} is the relative rate of the $T \leftrightarrow C$ substitution class (from the R-matrix), and π_C is the frequency of C. Thus, the rate at which $T \leftrightarrow C$ changes occur is influenced by the frequency of the base C: if Cs are common, this change will occur at a higher rate than if Cs are rare.

Now fix the rmatrix values and conduct an heuristic search using the following commands:

```
lset rmatrix=previous;
hsearch start=1 swap=tbr;
describe 1 / plot=phylogram;
```

Note that we do not need to obtain a starting tree, since we have already done that, so we tell PAUP* to simply begin with the first (and only) tree currently in memory (`start=1`) and conduct a heuristic search using TBR branch swapping.

Does the tree that PAUP estimates using the GTR model split the chlorophyll-a/b clade from the other two taxa?* [10]

10. **Maximum Likelihood Analyses: Adding Discrete Gamma Rate Heterogeneity.**

We have one more important feature to add to our model. Among-site rate variation is pronounced in most sequence data sets. A common way to model such heterogeneity in rates across sites is through the use of the discrete gamma model. This model assumes that the relative rates [11] are gamma-distributed with a shape parameter usually referred to as α. If the shape parameter is large (i.e., close to ∞), then the relative rates will all be clustered around their mean, 1.0, meaning that there is essentially **rate homogeneity**. If α is very low (i.e., close to 0.0), the relative rates are extremely dispersed with most near zero but a few very high values. This represents the case of high **rate heterogeneity**. Thus, using a gamma distribution to model rates allows us to capture much of the possible range of relative rate distributions with the addition of only one more parameter (α) to our model.

Our first step will be to estimate the amount of rate heterogeneity (and re-estimate the rmatrix, since it will be different under a rate heterogeneity model) using the tree already in memory:

```
lset rates=gamma ncat=4 shape=estimate rmatrix=estimate;
lscores 1;
```

To see graphically what the gamma distribution with this shape looks like, issue the gammaplot command:

```
gammaplot;
```

[10] No, *Euglena* is still stuck on *Olithodiscus*

[11] Note that these relative rates are different from those just discussed. Here we are referring to the relative rates at which different sites evolve; before, we were referring to the relative rates of different classes of substitutions

The shape of the distribution makes it obvious that this is rather extreme rate heterogeneity (low rate heterogeneity would be indicated by a sharp peak over the value 1.0). Now fix both the rmatrix and gamma shape parameters at their estimated values and conduct a new search, starting from a random stepwise addition tree and using TBR branch swapping.

```
lset shape=previous rmatrix=previous;
hsearch start=stepwise addseq=random swap=tbr;
describe 1;
```

This time you should see that chlorophyll-a/b clade return, so rate heterogeneity is obviously an important factor in the evolution of these sequences. The question "How important a factor is rate heterogeneity?" now arises. Is it more important, for example, than allowing for six different substitution rates? It turns out that for all but the simplest models (i.e., Jukes-Cantor and F81), allowing rate heterogeneity will produce the chlorophyll-a/b clade! To see this, here are the commands for setting up the Kimura 2-parameter model (K2P) with rate heterogeneity. This first obtains a stepwise addition tree using the plain K2P model, then estimates tratio and the gamma shape parameter on the stepwise addition tree, fixing the value of these two parameters before conducting a more thorough TBR search:

```
lset nst=2 basefreq=equal tratio=estimate rates=equal;
hsearch start=stepwise addseq=random swap=none nreps=1;
lset rates=gamma shape=estimate;
lscores 1;
lset shape=previous tratio=previous;
hsearch start=1 swap=tbr;
describe 1;
```

11. **Saving, viewing and printing trees using TreeView**

TreeView is a free program written by Rod Page that can be downloaded from this web page:

http://taxonomy.zoology.gla.ac.uk/rod/treeview.html

TreeView provides a graphical interface for viewing, editing, and printing trees produced by PAUP* or other programs. This somewhat makes up for the fact that PAUP*'s tree printing machinery has not yet been completed for the Windows version.

Alternatively, you can use FigTree by Andrew Rambaut:
http://tree.bio.ed.ac.uk/software/figtree/

To illustrate the use of TreeView (or FigTree) in conjunction with PAUP*, we will save a tree within PAUP* and then open and view it in TreeView. In PAUP*, issue this command to save the tree currently stored:

```
savetrees file=test.tre brlens;
```

This will save the tree currently stored in memory to the file `test.tre`. The `brlens` keyword tells PAUP* to store the branch lengths as well as the tree's topology. Start TreeView, and select `test.tre` in the dialog box that appears upon choosing the **File** | **Open...** menu item. Once the treefile has been opened, try switching to phylogram view (menu choice **Tree** | **Phylogram**) to take advantage of the fact that the branch lengths were stored. You can either print the tree if a printer is connected, or save the tree as a Windows metafile (which can later be imported into a Word document, for example). To save the tree as a Windows metafile, choose **File** | **Save as graphic...**.

12. **Constructing and Using PAUP Blocks.**

All commands that can be entered from the keyboard can also be placed in a NEXUS file in the form of a PAUP block. For example, the following PAUP block would enable us to automatically conduct the K2P analysis (see end of last step in the tutorial):

```
begin paup;
  log file=algae.log start replace;
  set autoclose=yes;
  execute algae.nex;
  outgroup Anacystis_nidulans;
  set criterion=likelihood;
  lset nst=2 basefreq=equal tratio=estimate rates=equal;
  hsearch start=stepwise addseq=random swap=none nreps=1;
  lset rates=gamma shape=estimate;
  lscores 1;
  lset shape=previous tratio=previous;
  hsearch start=1 swap=tbr;
  describe 1 / plot=phylogram;
  log stop;
end;
```

The first line after the `begin paup` command starts a log file named `algae.log` to which all the output will be echoed. The `replace` keyword in the `log` command tells PAUP* to automatically (i.e., without asking) overwrite any existing file named `algae.log`. The command `set autoclose=yes` tells PAUP* that we would like the progress dialog box that appears whenever a search is initiated to close without our intervention after each of the searches are completed. The `execute` command executes the `algae.nex` data file, which is a necessary prerequisite to doing an analysis. The `execute` command is equivalent to inserting the entire text of the `algae.nex` file (except for the initial `#nexus` keyword) into this file in place of this line. The only other command you have not yet seen is the last one before the `end` command, which instructs PAUP* to close the log file.

Why would one wish to place all the commands for an analysis into a separate file? I find it very useful for the simple reason that it is very easy to forget exactly what analyses you performed for a particular paper, and it is even easier to forget which file names you used for each separate analysis. If you use PAUP blocks, then you can exactly repeat an analysis at a later time. Another use for PAUP blocks is to perform initial setup steps. For example, if you always use the likelihood criterion whenever you analyze a particular data file, and if you tend to always use the HKY85 model, you could have PAUP* switch to the likelihood criterion and the HKY85 model every time you execute the data file by inserting the following PAUP block at the bottom of the file:

```
begin paup;
  log file=logfile.txt start append;
  set criterion=likelihood;
  lset nst=2 basefreq=empirical tratio=estimate rates=equal;
end;
```

Note that this time we are placing the PAUP block into the actual data file. While this is legal, I try to stay away from starting analyses inside the data file - better to keep analyses in separate NEXUS

files so that you do not accidentally start long analyses when your intention was only to open the data file.

In keeping with this suggestion, the above PAUP block does not perform any analyses, but it does start a log file and set up both the criterion and the model for you. This time I used the `append` keyword in the `log` command, which causes PAUP* to add new material to the end of the file `logfile.txt` if it already exists (`append` is safer than `replace` because existing files are never overwritten).

One further tip about PAUP blocks. I often create multiple PAUP blocks – one for each analysis done – in a single file. The ones that are currently not in use I simply "turn off" by changing the name slightly (for example, change the name from `paup` to `_paup`). If PAUP* does not recognize a block name, it skips the block and moves on to the next `begin` command. Thus, you can accumulate old PAUP blocks as a record of past analyses.

13. **Constructing Nexus Data Files and Finding Islands In Treespace**

Choose **File** | **New** from PAUP*'s menu (or press Ctrl-N) to open a new text editor window. Type the following small data set into the window, then choose **File** | **SaveAs...** from PAUP*'s menu (or press Ctrl-S) to save the file (call it `islands.nex`, or make up a name of your own).

```
#nexus

begin data;
  dimensions ntax=5 nchar=8;
  format datatype=dna;
  matrix
    A  A C G C A G G T
    B  A T G G T G A T
    C  G C T C A C G G
    D  A C T G T C G T
    E  G T T C T G A G
  ;
end;

begin paup;
  hsearch start=stepwise addseq=random nreps=100 swap=nni;
end;
```

After saving this file, use **File** | **Open...** (or press Ctrl-O) to execute the file. PAUP* should immediately perform the search since it was specified in a PAUP block in the data file itself. The `nreps=100` in the `hsearch` command tells PAUP* to perform 100 searches, each beginning with a (potentially [12] different random addition starting tree. Note that PAUP* was able to find both tree islands. This is the same data file presented in class as an example of tree islands. Look back at your notes for the figure showing the NNI connections between the trees, which shows that there are two islands in this treespace, each of which has two members.

Now redo the search but specify only one replicate. Type the following command into PAUP* command edit control at the bottom of the main window:

[12] I say potentially here since there are only 20 distinct random addition sequences for 5 taxa and thus there will be many replicates in which the random addition sequence is identical to that of previous replicates.

```
hs nreps=1;
```

Note that I have abbreviated `hsearch` as simply `hs` this time. PAUP* allows abbreviations of commands as long as enough letters are provided to distinguish the command from all other commands.

Now how many islands does PAUP find, and what are the sizes of these islands? Can you explain why PAUP* did not find all the islands this time?* [13]

Now redo the search once more, again specifying only one replicate, but this time perform an SPR search:

```
hs nreps=1 swap=spr;
```

Now how many islands does PAUP find, and what are the sizes of the islands? Can you explain why PAUP* obtained this result?*/footnoteOnly one island found, but it contains all four trees that were on two separate islands in the NNI search. Explanation: SPR search explores a different landscape in which there is no "valley" between these trees – it is able to easily hop over the divide that stopped NNI.

[13] It only finds a single island this time because, by definition, once it finds a tree in one island it will be unable to bridge the gap to find trees of equal score on other islands.

Tutorial Quick Reference

Here are the commands issued during the tutorial. This is useful if you want to proceed quickly through commands with which you are familiar, and then go back to the main tutorial when you hit something you don't understand.

Parsimony exhaustive search produces 2 trees of length 411 steps (p. 3)

```
set criterion=parsimony;
alltrees;
```

Set outgroup for display purposes (p. 4)

```
outgroup Anacystis_nidulans;
```

Branch-and-bound parsimony analysis yields same two trees of 411 steps (p. 4)

```
bandb upbound=470;
```

NNI search using JC distances produces tree with ME score 0.43085 (p. 5)

```
set criterion=distance;
dset distance=jc objective=me;
hsearch swap=nni;
```

Using question mark to get list of options for a command (p. 5)

```
dset ?;
```

Using describetree to show phylogram (p. 5)

```
describetree 1 / plot=phylogram;
```

Star-decomposition using the ME criterion yields tree with ME score 0.43085 (p. 5)

```
set criterion=distance;
dset objective=me;
stardecomp;
```

TBR search using LogDet distances and the least-squares criterion, yielding tree with score 0.03831 (p. 6)

```
dset distance=logdet objective=lsfit;
hsearch swap=tbr;
describetrees 1 / plot=phylogram;
```

Stepwise addition using likelihood and HKY model, yielding tree with score 3274.21265 (p. 6)

```
set criterion=likelihood;
lset nst=2 basefreq=empirical variant=hky tratio=estimate rates=equal;
hsearch start=stepwise addseq=random swap=none nreps=1;
```

Using lscores command to show MLEs of parameters, with ti/tv ratio 1.910491, kappa 3.637613, and the negative log-likelihood 3274.21265 (p. 7)

```
lscores 1;
```

Fixing parameter values using the previous option, search produces same tree with score 3274.21265 (p. 7)

```
lset tratio=previous;
hsearch;
```

Estimate parameters of the GTR model on the HKY tree in memory, yielding score 3251.02830 and rmatrix rAC=0.62353, rAG=1.87194, rAT=0.82691, rCG=0.32529, rCT=3.68057 and rGT=1.0 (p. 8)

```
lset nst=6 rmatrix=estimate;
lscores 1;
```

Use longfmt to show R-matrix and Q-matrix in matrix form (p. 8)

```
lscores 1 / longfmt showqmatrix;
```

TBR search using GTR model and starting with the HKY tree fails to find a better tree, score (3251.02830) identical to the HKY tree (p. 9)

```
lset rmatrix=previous;
hsearch start=1 swap=tbr;
describe 1 / plot=phylogram;
```

Estimate among-site rate heterogeneity for current tree, MLE of shape parameter is 0.255221 (p. 9)

```
lset rates=gamma ncat=4 shape=estimate rmatrix=estimate;
lscores 1;
```

The gammaplot command shows you what the gamma density function looks like for the current shape parameter (p. 9)

```
gammaplot;
```

TBR search starting with stepwise addition and allowing rate heterogeneity in the GTR model yields tree with score 3155.85106 (p. 10)

```
lset shape=previous rmatrix=previous;
hsearch start=stepwise addseq=random swap=tbr;
describe 1;
```

TBR search using K2P model with rate heterogeneity. First tree obtained (using plain K2P model) has score 3178.17621, tratio 2.435876, and kappa 4.871753. The second search with rate heterogeneity yields a tree with score 3178.07881 (p. 10)

```
lset nst=2 basefreq=equal tratio=estimate rates=equal;
hsearch start=stepwise addseq=random swap=none nreps=1;
lset rates=gamma shape=estimate;
lscores 1;
lset shape=previous tratio=previous;
hsearch start=1 swap=tbr;
describe 1;
```

Save trees to file using the savetrees command (p. 10)

```
savetrees file=test.tre brlens;
```

Using PAUP blocks to automate analyses (p. 11)

```
begin paup;
  log file=algae.log start replace;
  set autoclose=yes;
  execute algae.nex;
  outgroup Anacystis_nidulans;
  set criterion=likelihood;
  lset nst=2 basefreq=equal tratio=estimate rates=equal;
  hsearch start=stepwise addseq=random swap=none nreps=1;
  lset rates=gamma shape=estimate;
  lscores 1;
  lset shape=previous tratio=previous;
  hsearch start=1 swap=tbr;
  describe 1 / plot=phylogram;
  log stop;
end;
```

Data set for seeing tree islands resulting from NNI search. Two islands are found, both with score 13, and each of size 2 trees (p. 12)

```
#nexus

begin data;
  dimensions ntax=5 nchar=8;
  format datatype=dna;
  matrix
    A  A C G C A G G T
    B  A T G G T G A T
    C  G C T C A C G G
    D  A C T G T C G T
    E  G T T C T G A G
```

```
  ;
end;

begin paup;
  hsearch start=stepwise addseq=random nreps=100 swap=nni;
end;
```

NNI search with one replicate only finds one of the two known islands (p. 12)

```
hs nreps=1;
```

SPR search finds all four tree even though one replicate specified (p. 13)

```
hs nreps=1 swap=spr;
```

Summary

In summary, and for easy reference, here are the commands necessary to set up many of the models and search strategies discussed. There are many options for most of the commands; remember that you can use the question-mark approach to obtain a listing of all the options for any given command.

Search Strategies

Search strategy	PAUP* commands
Stepwise addition (only)	`hsearch start=stepwise addseq=random swap=no;`
Star decomposition	`stardecomp;`
Exhaustive enumeration	`alltrees;`
Branch-and-bound	`bandb;`
Stepwise addition + NNI heuristic search	`hsearch start=stepwise addseq=random swap=nni;`
Stepwise addition + SPR heuristic search	`hsearch start=stepwise addseq=random swap=spr;`
Stepwise addition + TBR heuristic search	`hsearch start=stepwise addseq=random swap=tbr;`

Distance Analyses

To get this:	Use these PAUP* commands
Neighbor-joining	`nj;`
Minimum Evolution criterion	`set criterion=distance;` `dset objective=me;`
Least-squares criterion (unweighted)	`set criterion=distance;` `dset objective=lsfit power=0;`
Fitch-Margoliash criterion (weighted least-squares)	`set criterion=distance;` `dset objective=lsfit power=2;`
JC69 model	`set criterion=distance;` `dset distance=jc;`
F81 model	`set criterion=distance;` `dset distance=f81;`
F84 model	`set criterion=distance;` `dset distance=f84;`
HKY85 model	`set criterion=distance;` `dset distance=hky85;`
GTR model	`set criterion=distance;` `dset distance=gtr;`
LogDet model	`set criterion=distance;` `dset distance=logdet;`
ML distances[a]	`set criterion=distance;` `dset distance=ml;`

[a] If ML distances are desired, use the `lset` command to set up the desired maximum likelihood model (see tables that follow), but make sure "`set criterion=distance;`" is in effect when the analysis is started.

Likelihood Models

In each of these models containing extra parameters (e.g., transition/transversion ratio), I have provided for estimating these extra parameters in the commands. Remember that this will add a lot of time to the analysis, and for each of these, there is a way to either set the parameter to the previous value or to any particular value. For example, to set `tratio` to a value previously estimated, you can use `tratio=previous` rather than `tratio=estimate`. To set `tratio` to some particular value, say 2.5, use `tratio=2.5`. The `rmatrix` setting works a little differently since there are 5 values associated with the `rmatrix` setting. To set `rmatrix` to values previously estimated, use `rmatrix=previous`; however, to set the `rmatrix` to a set of five specific values, say 0.62, 1.87, 0.82, 0.32, and 3.68, use this format: `rmatrix=(0.62 1.87 0.82 0.32 3.68 )`.

Models Assuming Rate Homogeneity

Model	PAUP* commands

JC69	`set criterion=likelihood;`
	`lset nst=1 basefreq=equal;`
	`lset rates=equal pinvar=0;`

F81	`set criterion=likelihood;`
	`lset nst=1 basefreq=empirical;`
	`lset rates=equal pinvar=0;`

K2P	`set criterion=likelihood;`
	`lset nst=2 basefreq=equal tratio=estimate;`
	`lset rates=equal pinvar=0;`

HKY85	`set criterion=likelihood;`
	`lset nst=2 basefreq=empirical variant=hky tratio=estimate;`
	`lset rates=equal pinvar=0;`

F84	`set criterion=likelihood;`
	`lset nst=2 basefreq=empirical variant=f84 tratio=estimate;`
	`lset rates=equal pinvar=0;`

GTR	`set criterion=likelihood;`
	`lset nst=6 basefreq=empirical rmatrix=estimate;`
	`lset rates=equal pinvar=0;`

Using Invariant-sites Model For Rate Heterogeneity

Model	PAUP* commands

JC69+I	`set criterion=likelihood;`
	`lset nst=1 basefreq=equal;`
	`lset rates=equal pinvar=estimate;`

F81+I	`set criterion=likelihood;`
	`lset nst=1 basefreq=empirical;`
	`lset rates=equal pinvar=estimate;`

K2P+I	`set criterion=likelihood;`
	`lset nst=2 basefreq=equal tratio=estimate;`
	`lset rates=equal pinvar=estimate;`

HKY85+I	`set criterion=likelihood;`
	`lset nst=2 basefreq=empirical variant=hky tratio=estimate;`
	`lset rates=equal pinvar=estimate;`

F84+I	`set criterion=likelihood;`
	`lset nst=2 basefreq=empirical variant=f84 tratio=estimate;`
	`lset rates=equal pinvar=estimate;`

GTR+I	`set criterion=likelihood;`
	`lset nst=6 basefreq=empirical rmatrix=estimate;`
	`lset rates=equal pinvar=estimate;`

Using Discrete Gamma Model For Rate Heterogeneity

Model	PAUP* commands

JC69+Γ	`set criterion=likelihood;`
	`lset nst=1 basefreq=equal;`
	`lset rates=gamma ncat=4 shape=estimate pinvar=0;`

F81+Γ	`set criterion=likelihood;`
	`lset nst=1 basefreq=empirical;`
	`lset rates=gamma ncat=4 shape=estimate pinvar=0;`

K2P+Γ	`set criterion=likelihood;`
	`lset nst=2 basefreq=equal tratio=estimate;`
	`lset rates=gamma ncat=4 shape=estimate pinvar=0;`

HKY85+Γ	`set criterion=likelihood;`
	`lset nst=2 basefreq=empirical variant=hky tratio=estimate;`
	`lset rates=gamma ncat=4 shape=estimate pinvar=0;`

F84+Γ	`set criterion=likelihood;`
	`lset nst=2 basefreq=empirical variant=f84 tratio=estimate;`
	`lset rates=gamma ncat=4 shape=estimate pinvar=0;`

GTR+Γ	`set criterion=likelihood;`
	`lset nst=6 basefreq=empirical rmatrix=estimate;`
	`lset rates=gamma ncat=4 shape=estimate pinvar=0;`

Using Invariant-sites and Discrete Gamma Model For Rate Heterogeneity

Model	PAUP* commands

JC69+I+Γ	`set criterion=likelihood;`
	`lset nst=1 basefreq=equal;`
	`lset rates=gamma ncat=4 shape=estimate pinvar=estimate;`

F81+I+Γ	`set criterion=likelihood;`
	`lset nst=1 basefreq=empirical;`
	`lset rates=gamma ncat=4 shape=estimate pinvar=estimate;`

K2P+I+Γ	`set criterion=likelihood;`
	`lset nst=2 basefreq=equal tratio=estimate;`
	`lset rates=gamma ncat=4 shape=estimate pinvar=estimate;`

HKY85+I+Γ	`set criterion=likelihood;`
	`lset nst=2 basefreq=empirical variant=hky tratio=estimate;`
	`lset rates=gamma ncat=4 shape=estimate pinvar=estimate;`

F84+I+Γ	`set criterion=likelihood;`
	`lset nst=2 basefreq=empirical variant=f84 tratio=estimate;`
	`lset rates=gamma ncat=4 shape=estimate pinvar=estimate;`

GTR+I+Γ	`set criterion=likelihood;`
	`lset nst=6 basefreq=empirical rmatrix=estimate;`
	`lset rates=gamma ncat=4 shape=estimate pinvar=estimate;`

Extra topic: Rate variation across classes of sites – the site-specific rates option

Earlier in the lab, we added gamma-distributed among site rate variation to our model of character evolution. This lets the model recognize the fact that (in almost all real datasets) some sites change much more frequently than others. This approach of using a generic, statistical distribution may seem unsatisfying because it does not incorporate much of our biological knowledge or intuition.

Another approach is to define different models of evolution for different classes of characters. The current version of PAUP* does not let you use multiple models simultaneously in a completely flexible way (while MrBayes does). However, PAUP**does* allow one component of the model (the rate of evolution) vary over sites. In PAUP*'s terminology this is the site-specific rates model.

This is an example of a partitioned analysis. The partitioning (the assignment of sites to different rate classes) must be done *a priori*. So, you must tell PAUP* which sites should be assumed to have the same rate of substitution. Once you have decided how to partition the sequences, you *can* ask PAUP* to infer the rates that are assigned to these categories.

One of the most common uses of the site-specific rates option in PAUP* is to let different sites in different codon positions have different rates. This agrees with ample empirical observations and the intuition that because many changes in the third position result in either no change in the amino acid sequence or a conservative change (a similar amino acid). Now we'll walk through an example of using PAUP* to assign different rates according to codon position.

1. Obtain the primate-mtDNA.nex file (it comes with PAUP*), for the duration of the workshop I have posted the file at http://www.people.ku.edu/~mtholder/sisg/
2. Open the file in a text editors (for example WordPad) and look at `assumptions` block. This section defines groups of taxa and characters (`TaxSet` s and `CharSet` s in NEXUS lingo). The `UserType` command show how you can define a stepmatrix for parsimony (this command is understood by PAUP*, MacClade, and Mesquite). For the purpose of this lab the important commands are the `CharSet` s that tell us which sites belong to each category: `noncoding, 1stpos, 2ndpos`, and 3rdpos. These `CharSet` names are (as far as PAUP* is concerned) arbitrary names – the name `1stpos` does not tell PAUP* that these are the first codon positions.
3. `Execute` the primate-mtDNA.nex file in PAUP*.
4. To tell PAUP* how to partition the characters, we use the `CharPartition` command (which should be placed in an `Assumptions` block if you want to use the partitioning scheme in more than one PAUP* session). You can define as many ways of partitioning the characters into groups as you like. Each partition is assigned a name. The general syntax is:
   `CharPartition <partition name> = <subset name> : <subset definition> , <subset name> : <subset definition> ;` Where everything in the `<>` is replaced with an appropriate value. Let's setup a character partition that groups first and second positions in one group, has another group for the third base positions, and a third subset for the non coding DNA sequence. The command is:

   ```
   CharPartition byCateg = fs: 1stpos 2ndpos , t : 3rdpos, n: noncoding ;
   ```

   This creates a partition named `byCateg`, first and second partitions are grouped in the subset `fs`, subset `t` has the 3rd base positions, and noncoding sites go into a category named `n`.

5. It is always a good idea to verify that the program did what you were expecting. We can ask PAUP* to show us how it thinks sites are assigned to subsets. Type the command:

   ```
   ShowCharParts;
   ```

   To see a table of subsets. Each character in the matrix has a column in this table (the table has to be wrapped across multiple lines because it is too long). There will be an `*` each column. This tells you which subset PAUP* thinks this character belongs to.

6. Lets get a tree in memory. A quick parsimony search should work (`BAndB` or `HSearch`).

7. Ask PAUP* to score these trees under maximum likelihood and using the HKY model with no among-site rate heterogeneity. Use the `LSet` command to set up the model:

   ```
   LSet BaseFreq=estimate NST=2 TRatio=estimate Rates = equal PInvar=0
   ```

   and `LScore` commands to get the score. Note the likelihood score.

8. Now add gamma-distributed rate heterogeneity to the model with the command:

   ```
   LSet Rates = gamma Shape = estimate;
   ```

   and rescore the trees. Were the scores significantly better?

9. Now lets see how the site-specific rate model works.

   ```
   LSet rates = siteSpec;
   LScore
   ```

   Ooops! This doesn't work does it? PAUP* does not know how to assign sites to rate categories. We have to tell it to use the `CmdPartition` that we created earlier:

   ```
   LSet rates = siteSpec SiteRates = Partition : byCateg;
   LScore;
   ```

   Note: the word "byCateg" here is the arbitrary name that we assigned to the character partition. Is the likelihood higher than the when we used the site-specific model? Do the the relative rates for each of the categories that PAUP* estimated agree with your intuition of which sites are fast and which are slow?

10. If we have *a priori* hypothesis of relative rates that we'd like to test, then we can tell PAUP* not only what the categories are, we can tell it what rates should be used for each. To do this we need a `RateSet` command. This looks like the `CharPartition` command, but we replace the names of subsets with numbers that determine the relative rates. If we thought that noncoding DNA evolved 2 times faster than third positions, and third positions evolve 2 times faster than first and second positions. Then we can test how well this matches the data by looking at the likelihood ratio between analysis in which we estimate the relative rates (the analysis we just ran) and a constrained analysis in which we specify the rates:

```
RateSet doubling = 1: 1stpos 2ndpos , 2 : 3rdpos, 4: noncoding ;
LSet rates = siteSpec SiteRates = RateSet : doubling;
LScore;
```

We can perform a significance test using 2 times the difference in log likelihoods between these hypotheses as our test statistic, and a critical values from the χ^2-distribution with 2 degrees of freedom (there are 3 categories. We can arbitrarily set one of them to 1.0 and estimate the relative rates for the other 2 categories. Thus in the unconstrained model we estimated 2 more parameters than the constrained version).

11. You can test that PAUP* is doing the right thing, by specifying equal rates for all categories:

```
RateSet eq = 1: 1stpos 2ndpos , 1 : 3rdpos, 1: noncoding ;
LSet rates = siteSpec SiteRates = RateSet : eq;
LScore;
```

This should give you the same likelihood as if you tell PAUP* not to use rate heterogeneity (as we did in step 7).

You cannot use the χ^2-distribution with the likelihood ratio test statistic to compare the gamma-distributed rate heterogeneity model and the site-specific rates model (because the models are not nested). I would also suggest that use caution when comparing these models based on their likelihoods. The site-specific rates model usually incorporates more prior biological knowledge (from your definition of partitioning), and often have a better fit. Unfortunately (in PAUP* implementation) you cannot allow for rate variation within your categories. Because some first and second sites (for instance) evolve at a high rate, this constraint may lead to worse behavior than the gamma-distributed form of rate variation. Take a look at:

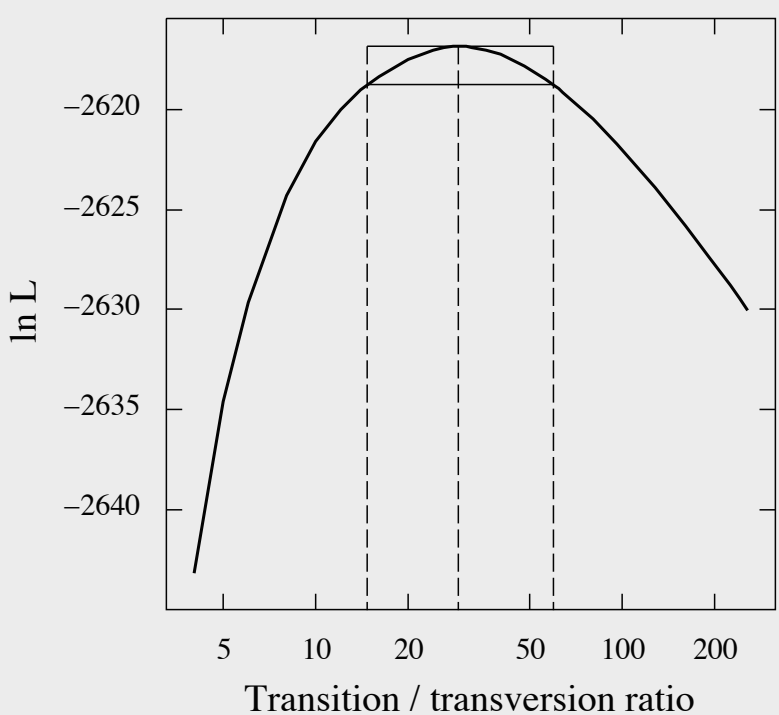
Buckley, T., C. Simon, and G. K. Chambers, "Exploring Among-Site Rate Variation Models in a Maximum Likelihood Framework Using Empirical Data: Effects of Model Assumptions on Estimates of Topology, Branch Lengths, and Bootstrap Support". *Syst. Biol.* **50**(1):6786, 2001.

Bootstraps and testing trees

Joe Felsenstein

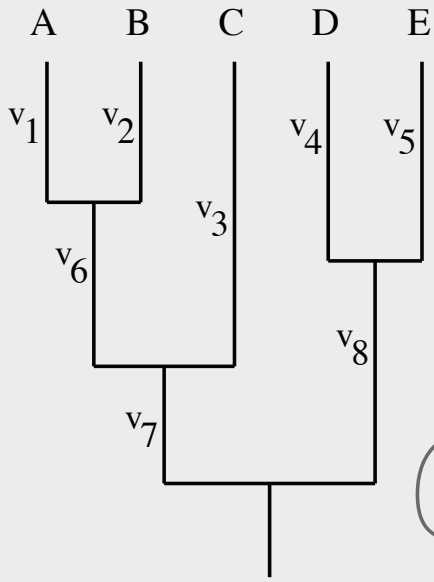
Depts. of Genome Sciences and of Biology, University of Washington

A log-likelihood curve and its confidence interval



(This is for the 14-species primates data available for download).

Constraints on a tree for a clock



Constraints for a clock

$v_1 = v_2$

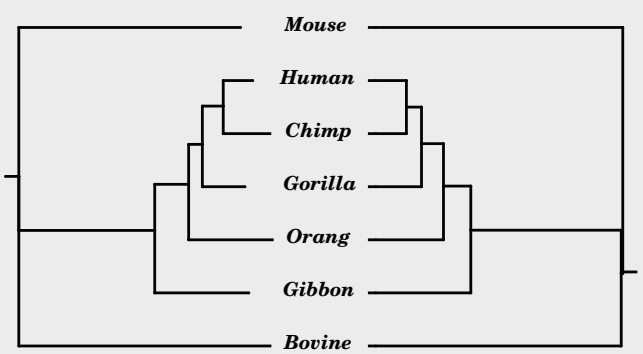
$v_4 = v_5$

$v_1 + v_6 = v_3$

$v_3 + v_7 = v_4 + v_8$

Does not constrain the branch length on the unrooted tree

Likelihood-ratio test of molecular clock



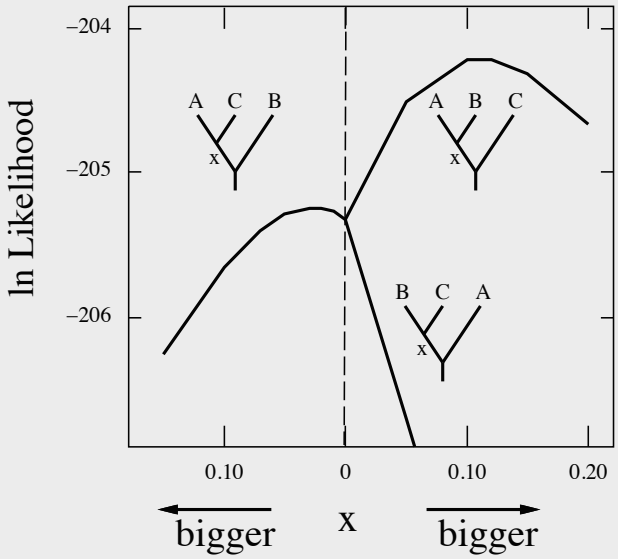
	log–likelihood	parameters
Without clock	–1405.608	11
With clock	–1407.085	6
Difference	1.477	5

$\chi^2 = 2.954$ df = 5

(non–significant)

(This is for this 7-species subset of the primates data).

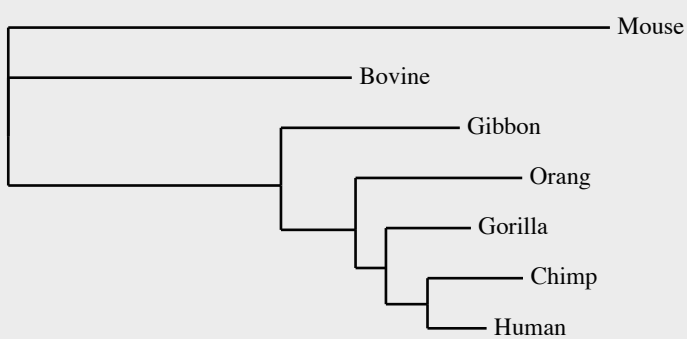
Likelihood surface for three clocklike trees



(These are “profile likelihoods" as they show the largest likelihood for that value of x, maximizing over the other branch length in the tree.)

Two trees to be tested using KHT test

Tree I



Tree II

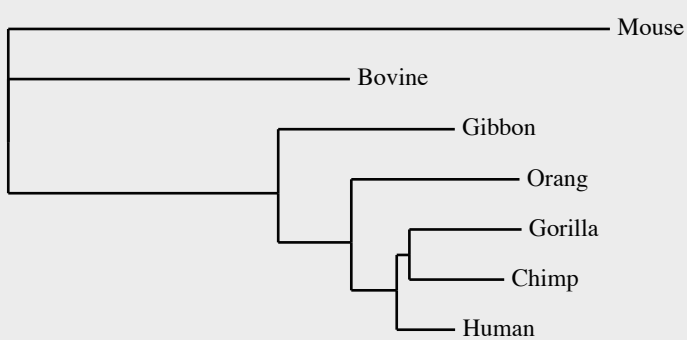
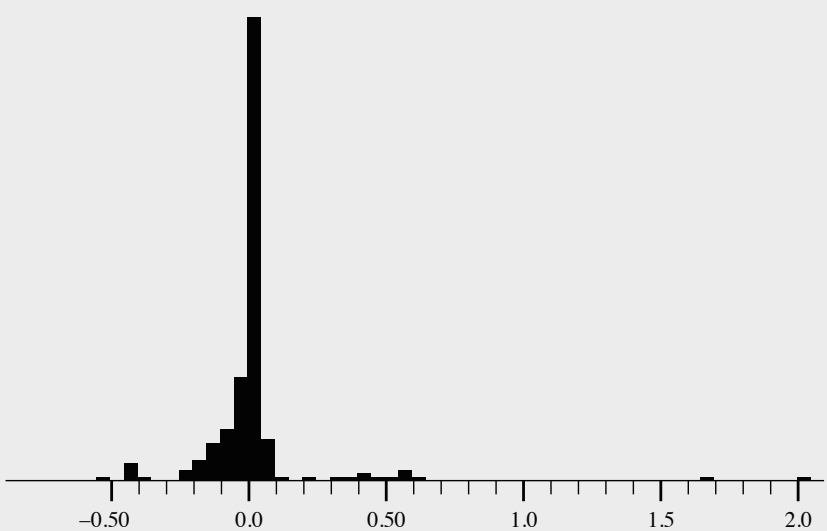


Table of differences in log-likelihood

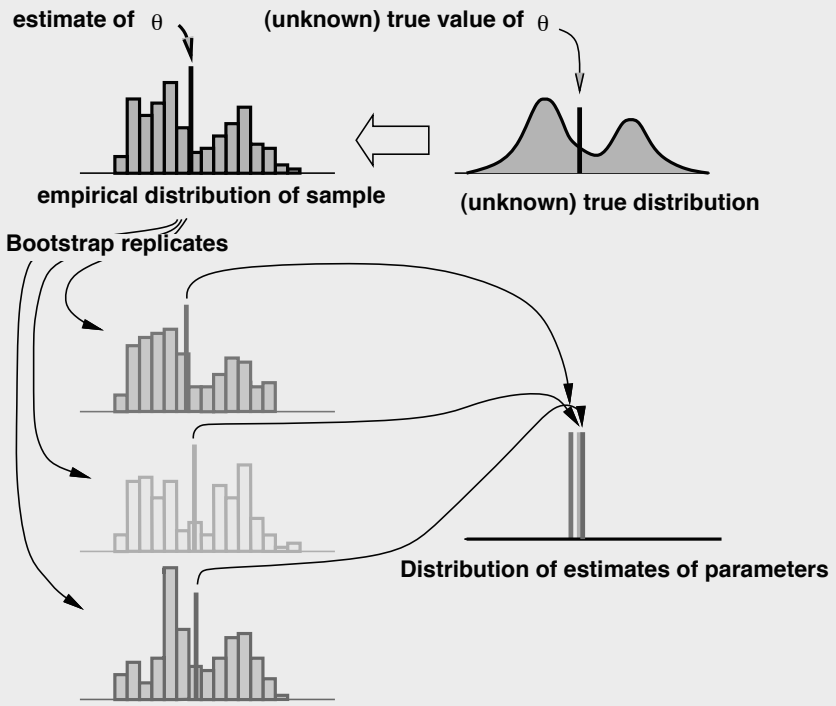
Tree \ site	1	2	3	4	5	6	…	231	232	ln L
I	–2.971	–4.483	–5.673	–5.883	–2.691	–8.003	…	–2.971	–2.691	–1405.61
II	–2.983	–4.494	–5.685	–5.898	–2.700	–7.572	…	–2.987	–2.705	–1408.80
Diff	+0.012	+0.111	+0.013	+0.015	+0.010	–0.431	…	+0.012	+0.010	+3.19

Histogram of those differences



Do sign test, or t-test, or similar nonparametric tests.

Bootstrap sampling (with mixtures of normals)

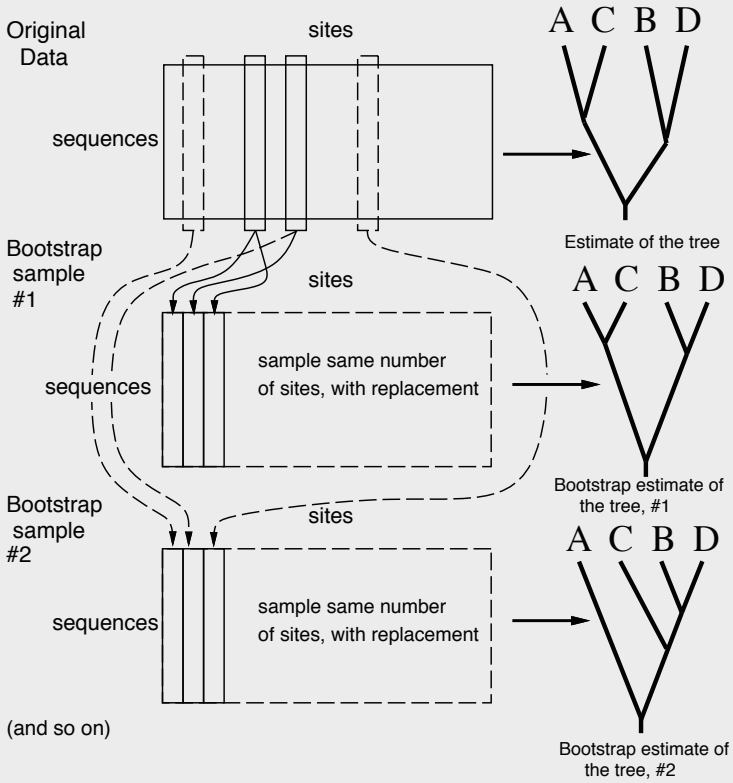


Bootstrap sampling

To infer the error in a quantity, θ, estimated from a sample of points $x_1, x_2, \ldots, x_n$ we can

- Do the following R times ($R = 1000$ or so)
- Draw a “bootstrap sample" by sampling n times with replacement from the sample. Call these $x_1^*, x_2^*, \ldots, x_n^*$. Note that some of the original points are represented more than once in the bootstrap sample, some once, some not at all.
- Estimate θ from each of the bootstrap samples, call these $\hat{\theta}_k^*$ ($k = 1, 2, \ldots, R$)
- When all R bootstrap samples have been done, the distribution of $\hat{\theta}_i^*$ estimates the distribution one would get if one were able to draw repeated samples of n data points from the unknown true distribution.

Bootstrap sampling of phylogenies



Analyzing bootstraps with phylogenies

The sites are assumed to have evolved independently given the tree. They are the entities that are sampled (the x_i). The trees play the role of the parameter. One ends up with a cloud of R sampled trees.

There are many possible ways. The one I will describe here is the most useful, but not the only way we could go.

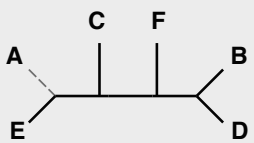
To summarize this cloud, we ask, for each branch in the tree, how frequently it appears among the cloud of trees. We make a tree that summarizes this for all the most frequently occurring branches. This is the majority rule consensus tree of the bootstrap estimates of the tree.

Partitions from branches in an (unrooted) tree

AE | BCDF

ACE | BDF

ACEF | BD

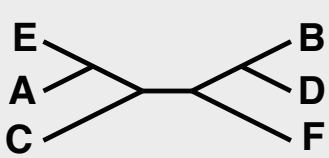


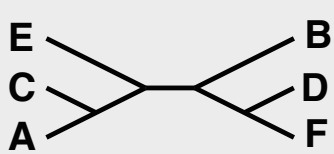
A | CEFBD

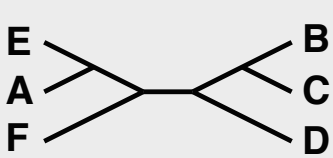
and so on for all the other external (tip) branches

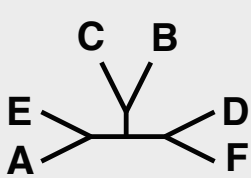
The majority-rule consensus tree

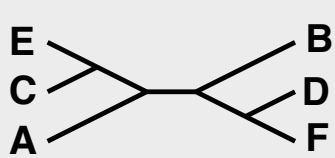
Trees:







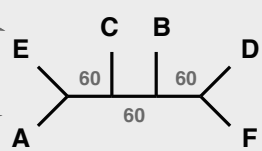




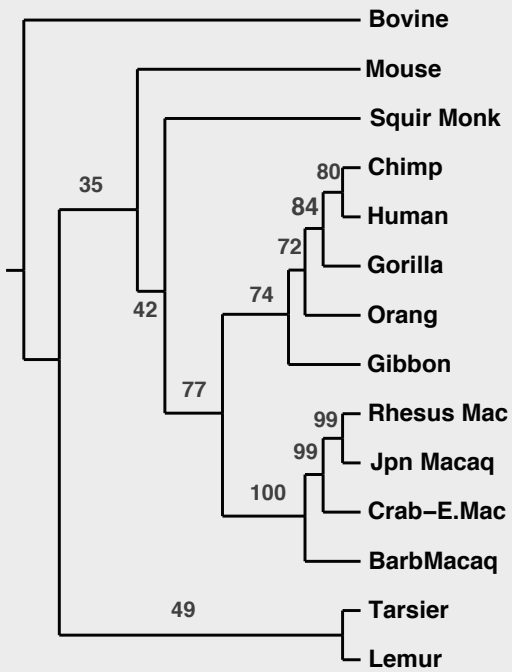
How many times each (non–tip) partition of species is found:

AE | BCDF	3
ACE | BDF	3
ACEF | BD	1
AC | BDEF	1
AEF | BCD	1
ADEF | BC	2
ABDF | EC	1
ABCE | DF	3

Majority–rule consensus tree of the unrooted trees:



Bootstrap sampling of a phylogeny

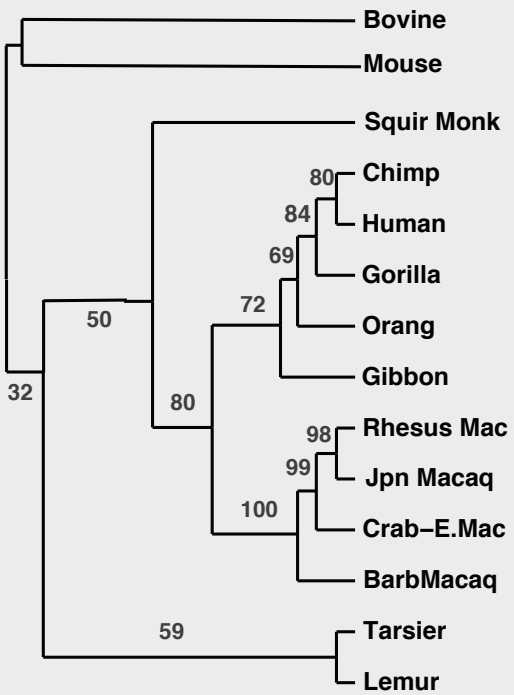


In this example, parsimony was used to infer the tree.

Potential problems with the bootstrap

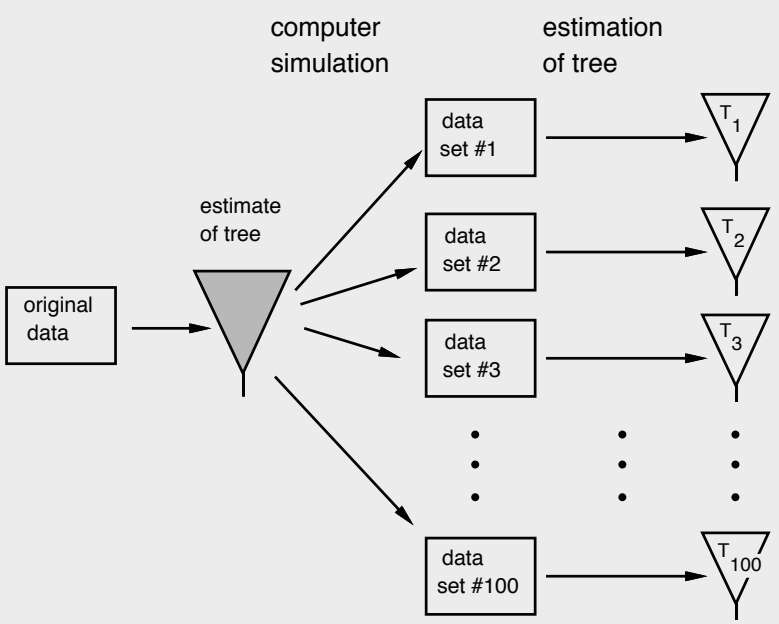
- Sites may not evolve independently
- Sites may not come from a common distribution (but you can consider them to be sampled from a mixture of possible distributions)
- If do not know which branch is of interest at the outset, a “multiple-tests" problem means that the most extreme P values are overstated
- P values are biased (too conservative)
- Bootstrapping does not correct biases in phylogeny methods

Delete-half jackknife P values



In this example, parsimony was used to infer the tree.

A diagram of the parametric bootstrap



References

Bootstraps etc.

Efron, B. 1979. Bootstrap methods: another look at the jackknife. *Annals of Statistics* **7:** 1-26. [The original bootstrap paper]

Margush, T. and F. R. McMorris. 1981. Consensus n-trees. *Bulletin of Mathematical Biology* **43:** 239-244i. [Majority-rule consensus trees]

Felsenstein, J. 1985. Confidence limits on phylogenies: an approach using the bootstrap. *Evolution* **39:** 783-791. [The bootstrap first applied to phylogenies]

Zharkikh, A., and W.-H. Li. 1992. Statistical properties of bootstrap estimation of phylogenetic variability from nucleotide sequences. I. Four taxa with a molecular clock. *Molecular Biology and Evolution* **9:** 1119-1147. [Discovery and explanation of bias in P values]

Künsch, H. R. 1989. The jackknife and the bootstrap for general stationary observations. *Annals of Statistics* **17:** 1217-1241. [The block-bootstrap]

Wu, C. F. J. 1986. Jackknife, bootstrap and other resampling plans in regression analysis. *Annals of Statistics* **14:** 1261-1295. [The delete-half jackknife]

Efron, B. 1985. Bootstrap confidence intervals for a class of parametric problems. *Biometrika* **72:** 45-58. [The parametric bootstrap]

(more references)

Templeton, A. R. 1983. Phylogenetic inference from restriction endonuclease cleavage site maps with particular reference to the evolution of humans and the apes. *Evolution* **37:** 221-224. [The first paper on the KHT test]

Goldman, N. 1993. Statistical tests of models of DNA substitution. *Journal of Molecular Evolution* **36:** 182-98. [Parametric bootstrapping for testing models]

Shimodaira, H. and M. Hasegawa. 1999. Multiple comparisons of log-likelihoods with applications to phylogenetic inference. *Molecular Biology and Evolution* **16:** 1114-1116. [Correction of KHT test for multiple hypothesis]

Prager, E. M. and A. C. Wilson. 1988. Ancient origin of lactalbumin from lysozyme: analysis of DNA and amino acid sequences. *Journal of Molecular Evolution* **27:** 326-335. [winning-sites test]

Hasegawa, M. and H. Kishino. 1994. Accuracies of the simple methods for estimating the bootstrap probability of a maximum-likelihood tree. Molecular Biology and Evolution **11:** 142-145. [RELL probabilities]

Max. likelihood & Bayesian techniques are both likelihood-based.

Weaknesses of likelihood for phylogeny reconstruction:
1) Computational tractability
2) Based on overly simplistic evolutionary models.

But,

a) All phylogeny reconstruction methods are based on assumptions but some (e.g. parsimony) are not based on explicit ones. For methods based on unstated assumptions, we need to worry not just whether the assumptions are realistic but also we need to worry about what they are.

b) Likelihood methods allow assumptions to be rigorously tested. When an assumption is found to be particularly poor, it can be replaced with a better one (i.e., models will improve over time!)

Max. likelihood & Bayesian techniques are both likelihood-based.

Weaknesses of likelihood for phylogeny reconstruction:
1) Computational tractability
2) Based on overly simplistic evolutionary models.

But,

a) All phylogeny reconstruction methods are based on assumptions but some (e.g. parsimony) are not based on explicit ones. For methods based on unstated assumptions, we need to worry not just whether the assumptions are realistic but also we need to worry about what they are.

b) Likelihood methods allow assumptions to be rigorously tested. When an assumption is found to be particularly poor, it can be replaced with a better one (i.e., models will improve over time!)

Strengths of likelihood methods:

1. Explicit Assumptions – we know what we're assuming.
2. Use **all** information in a data set. Distance methods, for example, do not. This is part of the explanation for success of likelihood methods in simulations – they tend to yield estimates that are closer to the truth than other methods.
3. Likelihood approaches are consistent. Estimates get better as amount of data increases. (Caveat: violation of model assumptions may cause loss of consistency property)
4. Because likelihood applied to so many statistical situations in addition to phylogenetics, powerful theory & tools for performing likelihood analyses have developed. This theory and these tools (e.g., tools for hypothesis testing) can be applied to phylogenetics.
5. Likelihood lets you know how good estimate is, in addition to what estimate is.

Mechanistic versus Phenomenological Models of Sequence Evolution

see Ph.D. thesis by Nicolas Rodrigue ("Phylogenetic structural modeling of molecular evolution", 2008, University of Montreal)

(see also Rodrigue & Philippe. 2010. Trends in Genetics 26:248-252)

Mutation-Selection Balance:

For change from i to j, evolutionary rate is R_{ij} where

R_{ij} = (Mutation Rate) x (Fixation Probability)

(see Halpern & Bruno. 1998. MBE 15:910-917)

With low mutation rates, this depends on effective pop'n size "N" and relative fitness of j minus i (call this difference "s")

Population Genetic formulae for fixation probability allows estimation of Ns

One good idea for more realistic models ...

TUFFLEY, C., and M. A. STEEL. 1998. Modeling the covarion hypothesis of nucleotide substitution. Math. Biosci. 147:63–91.

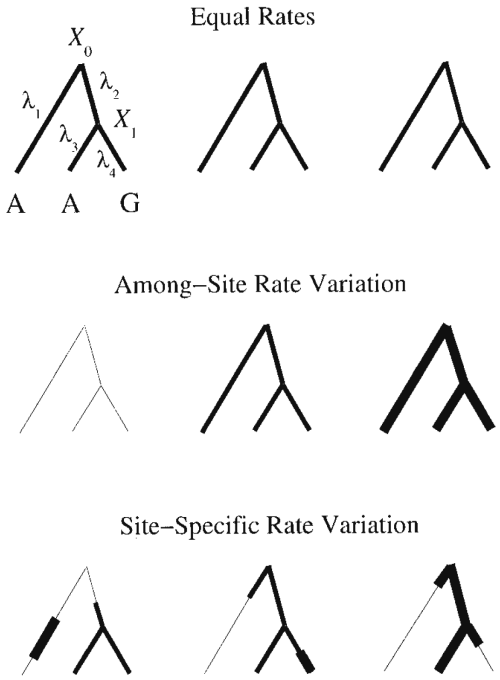


FIG. 1.—Distribution of rates across sites and lineages under three models of evolution. Each tree plot describes the distribution of rates across lineages for a particular site under the considered model. Three categories of rate are assumed, represented by different line thicknesses. Under the equal-rates (ER) model, all sites evolve at a constant, unique, moderate rate. Under the among-site rate variation (ASRV) model, each site has its own rate (low, moderate, or high), which is constant between lineages. Under the site-specific rate variation (SSRV) model, the rate of a site can switch between categories; a site has distinct rates in distinct lineages.

From Galtier. 2001. Mol. Biol. Evol. 18(5):866-873.

Tuffley/Steel -type model

		Slow				*Fast*			
		A	**C**	**G**	**T**	**A**	**C**	**G**	**T**
Slow	**A**	**-**	**r**	**r**	**r**	**f**	**0**	**0**	**0**
	C	**r**	**-**	**r**	**r**	**0**	**f**	**0**	**0**
	G	**r**	**r**	**-**	**r**	**0**	**0**	**f**	**0**
	T	**r**	**r**	**r**	**-**	**0**	**0**	**0**	**f**
Fast	**A**	**s**	**0**	**0**	**0**	**-**	**q**	**q**	**q**
	C	**0**	**s**	**0**	**0**	**q**	**-**	**q**	**q**
	G	**0**	**0**	**s**	**0**	**q**	**q**	**-**	**q**
	T	**0**	**0**	**0**	**s**	**q**	**q**	**q**	**-**

Substitution Rates: q>r

Switching rates: f (slow to fast), s (fast to slow)

Inspired by Lartillot and Philippe's CAT model of amino acid replacement that permits variation of preferred residues among sites, there is active development of sequence evolution models that allow variation of evolutionary processes among sites without prespecifying the number of categories, the nature of categories, or which sites are in which categories.

Key Ingredient: "Dirichlet Process" as a prior for the number of categories and for the probabilities of the categories.

Nicolas Lartillot and Hervé Philippe. 2004. A Bayesian Mixture Model for Across-Site Heterogeneities in the Amino-Acid Replacement Process. Mol. Biol. Evol. 21(6):1095-1109. 2004

Dirichlet Process Priors ("Chinese restaurant process", not same as Dirichlet distribution):

Useful to specify prior distribution for situations when number of categories is unknown and where prior probability of each possible category needs determination.

Additional applications in Evolution Include:

Characterization of population structure
Huelsenbeck and Andolfatto. 2007. Genetics. 175:1787-1802.

Variation in nonsyn. and synonymous rates among sites
Huelsenbeck et al. 2006. PNAS 103(16): 6263-6268.

Variation in evolutionary rate across a phylogeny
Heath et al. 2012. Mol. Biol. Evol. 29(3): 939-955.

Codon Models: Evolution occurs at the DNA level rather than at the amino acid level.

It makes sense to frame a model of protein evolution in terms of codons rather than amino acid types (Schoniger et al. 1990; Goldman and Yang 1994; Muse and Gaut 1994).

Codon-based models are typically framed in terms of 61 codon-states rather than 64 codon-states because the common genetic codes have three stop codons, and the possibility that a stop codon may appear or disappear from a sequence is not allowed.

One simplification that is often adopted holds that changes from one codon to another are only possible when the two codons differ at exactly one of the three codon positions.

The instantaneous rates of other changes between codons are set to 0.

Typical parameterization of a codon model when physicochemical differences between amino acids are ignored...

Instantaneous rate $\alpha_{i,j}$ from codon i to codon j is set to 0 if i and j differ at more than one nucleotide or if j encodes a premature stop codon. For cases where i and j differ by exactly one nucleotide, rate matrix entries are:

$$\alpha_{i,j} = \begin{cases} u\pi_j & \text{for a synonymous transversion} \\ u\pi_j\kappa & \text{for a synonymous transition} \\ u\pi_j\omega & \text{for a nonsynonymous transversion} \\ u\pi_j\kappa\omega & \text{for a nonsynonymous transition} \end{cases}$$

u, π_j, and κ reflect mutation rates

$\omega > 1$ means positive **diversifying** selection (i.e., nonsyn. rates higher than they would be if changes were synonymous)

Other kinds of positive selection exist (e.g., positive directional selection)

The previous rate matrix can be modified so that each codon k has its own parameter ω_k. The rates then become:

$$\alpha_{i,j} = \begin{cases} u\pi_h & \text{for a synonymous transversion} \\ u\pi_j\kappa & \text{for a synonymous transition} \\ u\pi_j\omega_k & \text{for a nonsynonymous transversion} \\ u\pi_j\kappa\omega_k & \text{for a nonsynonymous transition} \end{cases}$$

As with the rate heterogeneity among sites treatment, the distribution of ω_k values among codons can be modelled. Often, we want to know if certain codons have ω_k values that exceed 1.

Alternatively, we can assume all codons share the same value of ω but that ω values vary among branches on the tree. The rate matrix then becomes:

$$\alpha_{i,j} = \begin{cases} u\pi_j & \text{for a synonymous transversion} \\ u\pi_j\kappa & \text{for a synonymous transition} \\ u\pi_j\omega_B & \text{for a nonsynonymous transversion} \\ u\pi_j\kappa\omega_B & \text{for a nonsynonymous transition} \end{cases}$$

where ω_B is the parameter value for branch B. Many other possibilities for parameterizing codon models exist. and codon models can become very elaborate.

For example, Pedersen and colleagues (1998) carefully designed a codon model to reflect the fact that CpG dinucleotide levels are depressed in lentiviral genes.

Codon models have received attention for their potential ability to detect positive selection (Nielsen and Yang 1998).

Early methods for detecting positive selection from protein-coding DNA sequence data were designed to looked for an "excess" of nonsynonymous amino acid replacements throughout the sequence.

Codon methods offer the potential of detecting positive selection at individual sites and for detecting the existence of a small proportion of sites at which positive selection may operate.

Best statistical technique for detecting positive selection is a contentious issue at the moment...

Some future directions for codon-based models ...

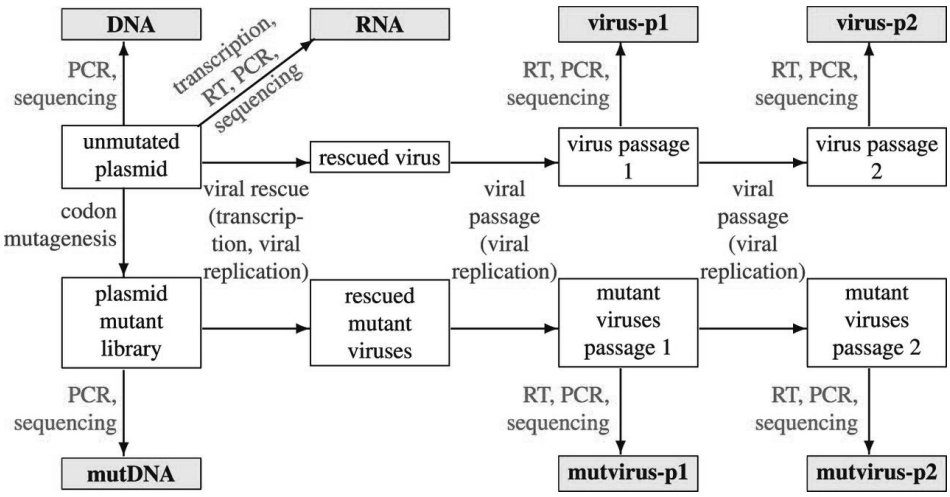
Evolutionary changes that simultaneously affect two consecutive positions could be allowed (Averof et al. 2000 have claimed empirical evidence for these kinds of changes).

Reconciliation of codon-based models with classical population genetic models – some progress has been made (see Halpern and Bruno 1998).

Improved treatment of effects of chemical similarity of amino acids on protein evolution

Experimental Evolution Can Inform Models

Fig. 2 from Bloom J.D. MolBiol Evol 2014;molbev.msu173



"**Design of the deep mutational scanning experiment.** The sequenced samples are in yellow. Blue text indicates sources of mutation and selection; red text indicates sources of errors. The comparison of interest is between the mutation frequencies in the mutDNA and mutvirus samples, because changes in frequencies between these samples represent the action of selection. However, because some of the experimental techniques have the potential to introduce errors, the other samples are also sequenced to quantify these unintended sources of error. "

Databases can inform models ...

Dayhoff model of protein evolution (see Dayhoff et al. 1972; Dayhoff et al. 1978) operates at the level of the 20 amino acid types.

π_i is the probability of amino acid type i

α_{ij} is the instantaneous rate of replacement from amino acid i to amino acid j

Dayhoff model is most general time-reversible 20-state model of amino acid replacement.

This means $\pi_i \alpha_{ij} = \pi_j \alpha_{ji}$ for all i and j.

It is important to separate the Dayhoff model of protein evolution from:

1. The procedure used by Dayhoff and collaborators to estimate the α_{ij} AND

2. The data set upon which the α_{ij} estimates were based.

Dayhoff and collaborators exploited the fact that the probability of replacements from amino acid type i to type j (i not equal to j) is approximately linear in time for small amounts of time.

In other words, the probability of a replacement from amino acid type i to a different type j is approximately $\alpha_{ij}t$ if t represents some small amount of time.

Subsequent studies (e.g., Jones et al. 1992) adopted the Dayhoff model but employed different data sets and parameter estimation procedures.

A	Ala																		
R	Arg	30																	
N	Asn	109	17																
D	Asp	154	0	532															
C	Cys	33	10	0	0														
Q	Gln	93	120	50	76	0													
E	Glu	266	0	94	831	0	422												
G	Gly	579	10	156	162	10	30	112											
H	His	21	103	226	43	10	243	23	10										
I	Ile	66	30	36	13	17	8	35	0	3									
L	Leu	95	17	37	0	0	75	15	17	40	253								
K	Lys	57	477	322	85	0	147	104	60	23	43	39							
M	Met	29	17	0	0	0	20	7	7	0	57	207	90						
F	Phe	20	7	7	0	0	0	0	17	20	90	167	0	17					
P	Pro	345	67	27	10	10	93	40	49	50	7	43	43	4	7				
S	Ser	772	137	432	98	117	47	86	450	26	20	32	168	20	40	269			
T	Thr	590	20	169	57	10	37	31	50	14	129	52	200	28	10	73	696		
W	Trp	0	27	3	0	0	0	0	0	3	0	13	0	0	10	0	17	0	
Y	Tyr	20	3	36	0	30	0	10	0	40	13	23	10	0	260	0	22	23	6
V	Val	365	20	13	17	33	27	37	97	30	661	303	17	77	10	50	43	186	0	17
		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y
		Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val

Figure 80. Numbers of accepted point mutations (X 10) accumulated from closely related sequences. Fifteen hundred and seventy-two exchanges are shown. Fractional exchanges result when ancestral sequences are ambiguous.

ORIGINAL AMINO ACID

REPLACEMENT AMINO ACID	A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile	L Leu	K Lys	M Met	F Phe	P Pro	S Ser	T Thr	W Trp	Y Tyr	V Val
A Ala	9867	2	9	10	3	8	17	21	2	6	4	2	6	2	22	35	32	0	2	18
R Arg	1	9913	1	0	1	10	0	0	10	3	1	19	4	1	4	6	1	8	0	1
N Asn	4	1	9822	36	0	4	6	6	21	3	1	13	0	1	2	20	9	1	4	1
D Asp	6	0	42	9859	0	6	53	6	4	1	0	3	0	0	1	5	3	0	0	1
C Cys	1	1	0	0	9973	0	0	0	1	1	0	0	0	0	1	5	1	0	3	2
Q Gln	3	9	4	5	0	9876	27	1	23	1	3	6	4	0	6	2	2	0	0	1
E Glu	10	0	7	56	0	35	9865	4	2	3	1	4	1	0	3	4	2	0	1	2
G Gly	21	1	12	11	1	3	7	9935	1	0	1	2	1	1	3	21	3	0	0	5
H His	1	8	18	3	1	20	1	0	9912	0	1	1	0	2	3	1	1	1	4	1
I Ile	2	2	3	1	2	1	2	0	0	9872	9	2	12	7	0	1	7	0	1	33
L Leu	3	1	3	0	0	6	1	1	4	22	9947	2	45	13	3	1	3	4	2	15
K Lys	2	37	25	6	0	12	7	2	2	4	1	9926	20	0	3	8	11	0	1	1
M Met	1	1	0	0	0	2	0	0	0	5	8	4	9874	1	0	1	2	0	0	4
F Phe	1	1	1	0	0	0	0	1	2	8	6	0	4	9946	0	2	1	3	28	0
P Pro	13	5	2	1	1	8	3	2	5	1	2	2	1	1	9926	12	4	0	0	2
S Ser	28	11	34	7	11	4	6	16	2	2	1	7	4	3	17	9840	38	5	2	2
T Thr	22	2	13	4	1	3	2	2	1	11	2	8	6	1	5	32	9871	0	2	9
W Trp	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	9976	1	0
Y Tyr	1	0	3	0	3	0	1	0	4	1	1	0	0	21	0	1	1	2	9945	1
V Val	13	2	1	1	3	2	2	3	3	57	11	1	17	1	3	2	10	0	2	9901

Figure 82. Mutation probability matrix for the evolutionary distance of 1 PAM. An element of this matrix, M_{ij}, gives the probability that the amino acid in column j will be replaced by the amino acid in row i after a given evolutionary interval, in this case 1 accepted point mutation per 100 amino acids. Thus, there is a 0.56% probability that Asp will be replaced by Glu. To simplify the appearance, the elements are shown multiplied by 10,000.

ORIGINAL AMINO ACID

REPLACEMENT AMINO ACID	A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile	L Leu	K Lys	M Met	F Phe	P Pro	S Ser	T Thr	W Trp	Y Tyr	V Val
A Ala	13	6	9	9	5	8	9	12	6	8	6	7	7	4	11	11	11	2	4	9
R Arg	3	17	4	3	2	5	3	2	6	3	2	9	4	1	4	4	3	7	2	2
N Asn	4	4	6	7	2	5	6	4	6	3	2	5	3	2	4	5	4	2	3	3
D Asp	5	4	8	11	1	7	10	5	6	3	2	5	3	1	4	5	5	1	2	3
C Cys	2	1	1	1	52	1	1	2	2	2	1	1	1	1	2	3	2	1	4	2
Q Gln	3	5	5	6	1	10	7	3	7	2	3	5	3	1	4	3	3	1	2	3
E Glu	5	4	7	11	1	9	12	5	6	3	2	5	3	1	4	5	5	1	2	3
G Gly	12	5	10	10	4	7	9	27	5	5	4	6	5	3	8	11	9	2	3	7
H His	2	5	5	4	2	7	4	2	15	2	2	3	2	2	3	3	2	2	3	2
I Ile	3	2	2	2	2	2	2	2	2	10	6	2	6	5	2	3	4	1	3	9
L Leu	6	4	4	3	2	6	4	3	5	15	34	4	20	13	5	4	6	6	7	13
K Lys	6	18	10	8	2	10	8	5	8	5	4	24	9	2	6	8	8	4	3	5
M Met	1	1	1	1	0	1	1	1	1	2	3	2	6	2	1	1	1	1	1	2
F Phe	2	1	2	1	1	1	1	1	3	5	6	1	4	32	1	2	2	4	20	3
P Pro	7	5	5	4	3	5	4	5	5	3	3	4	3	2	20	6	5	1	2	4
S Ser	9	6	8	7	7	6	7	9	6	5	4	7	5	3	9	10	9	4	4	6
T Thr	8	5	6	6	4	5	5	6	4	6	4	6	5	3	6	8	11	2	3	6
W Trp	0	2	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	55	1	0
Y Tyr	1	1	2	1	3	1	1	1	3	2	2	1	2	15	1	2	2	3	31	2
V Val	7	4	4	4	4	4	4	5	4	15	10	4	10	5	5	5	7	2	4	17

Figure 83. Mutation probability matrix for the evolutionary distance of 250 PAMs. To simplify the appearance, the elements are shown multiplied by 100. In comparing two sequences of average amino acid frequency at this evolutionary distance, there is a 13% probability that a position containing Ala in the first sequence will contain Ala in the second. There is a 3% chance that it will contain Arg, and so forth. The relationship of two sequences at a distance of 250 PAMs can be demonstrated by statistical methods.

C	Cys	12																				
S	Ser	0	2																			
T	Thr	-2	1	3																		
P	Pro	-3	1	0	6																	
A	Ala	-2	1	1	1	2																
G	Gly	-3	1	0	-1	1	5															
N	Asn	-4	1	0	-1	0	0	2														
D	Asp	-5	0	0	-1	0	1	2	4													
E	Glu	-5	0	0	-1	0	0	1	3	4												
Q	Gln	-5	-1	-1	0	0	-1	1	2	2	4											
H	His	-3	-1	-1	0	-1	-2	2	1	1	3	6										
R	Arg	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6									
K	Lys	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5								
M	Met	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6							
I	Ile	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5						
L	Leu	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6					
V	Val	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4				
F	Phe	-4	-3	-3	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9			
Y	Tyr	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10		
W	Trp	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17	
		C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
		Cys	Ser	Thr	Pro	Ala	Gly	Asn	Asp	Glu	Gln	His	Arg	Lys	Met	Ile	Leu	Val	Phe	Tyr	Trp

Figure 84. Log odds matrix for 250 PAMs. Elements are shown multiplied by 10. The neutral score is zero. A score of -10 means that the pair would be expected to occur only one-tenth as frequently in related sequences as random chance would predict, and a score of +2 means that the pair would be expected to occur 1.6 times as frequently. The order of the amino acids has been arranged to illustrate the patterns in the mutation data.

Table 23
Correspondence between Observed Differences and the Evolutionary Distance

Observed Percent Difference	Evolutionary Distance in PAMs
1	1
5	5
10	11
15	17
20	23
25	30
30	38
35	47
40	56
45	67
50	80
55	94
60	112
65	133
70	159
75	195
80	246
85	328

Table 21

Relative Mutabilities of the Amino Acids[a]

Asn	134	His	66
Ser	120	Arg	65
Asp	106	Lys	56
Glu	102	Pro	56
Ala	100	Gly	49
Thr	97	Tyr	41
Ile	96	Phe	41
Met	94	Leu	40
Gln	93	Cys	20
Val	74	Trp	18

[a]The value for Ala has been arbitrarily set at 100.

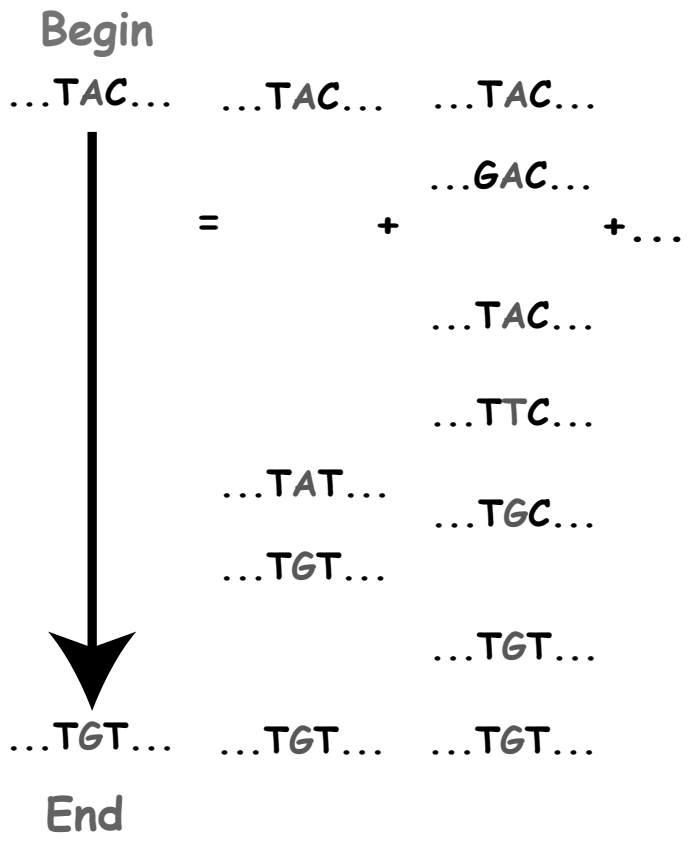
Table 22

Normalized Frequencies of the Amino Acids in the Accepted Point Mutation Data

Gly	0.089	Arg	0.041
Ala	0.087	Asn	0.040
Leu	0.085	Phe	0.040
Lys	0.081	Gln	0.038
Ser	0.070	Ile	0.037
Val	0.065	His	0.034
Thr	0.058	Cys	0.033
Pro	0.051	Tyr	0.030
Glu	0.050	Met	0.015
Asp	0.047	Trp	0.010

An infinite number of possible evolutionary histories are consistent with sequences at the beginning and end of a branch on a tree.

transition probabilities add up all these possible histories...



4-state substitution model

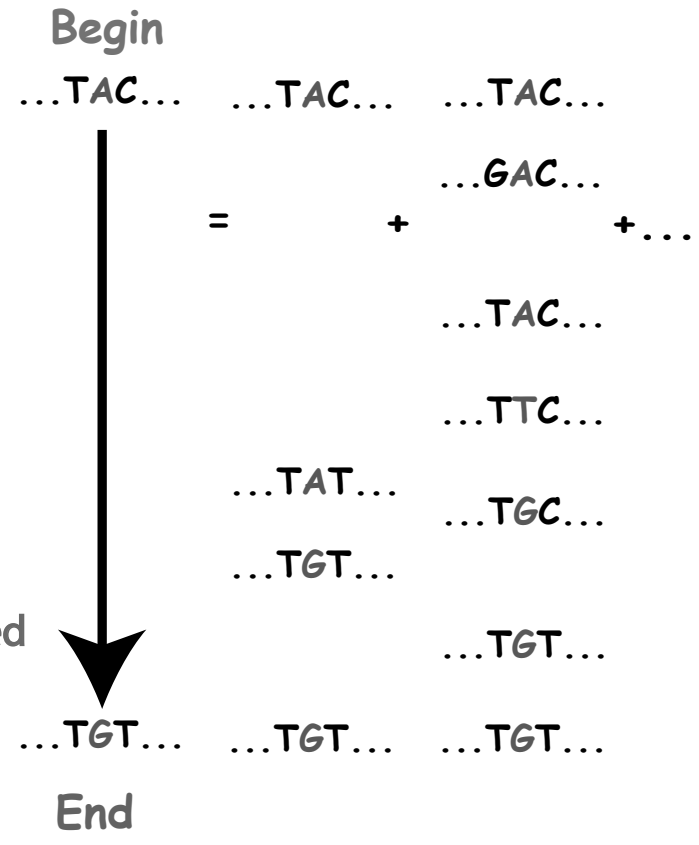
From \ To	A	C	G	T
A	–	+	+	+
C	+	–	+	+
G	+	+	–	+
T	+	+	+	–

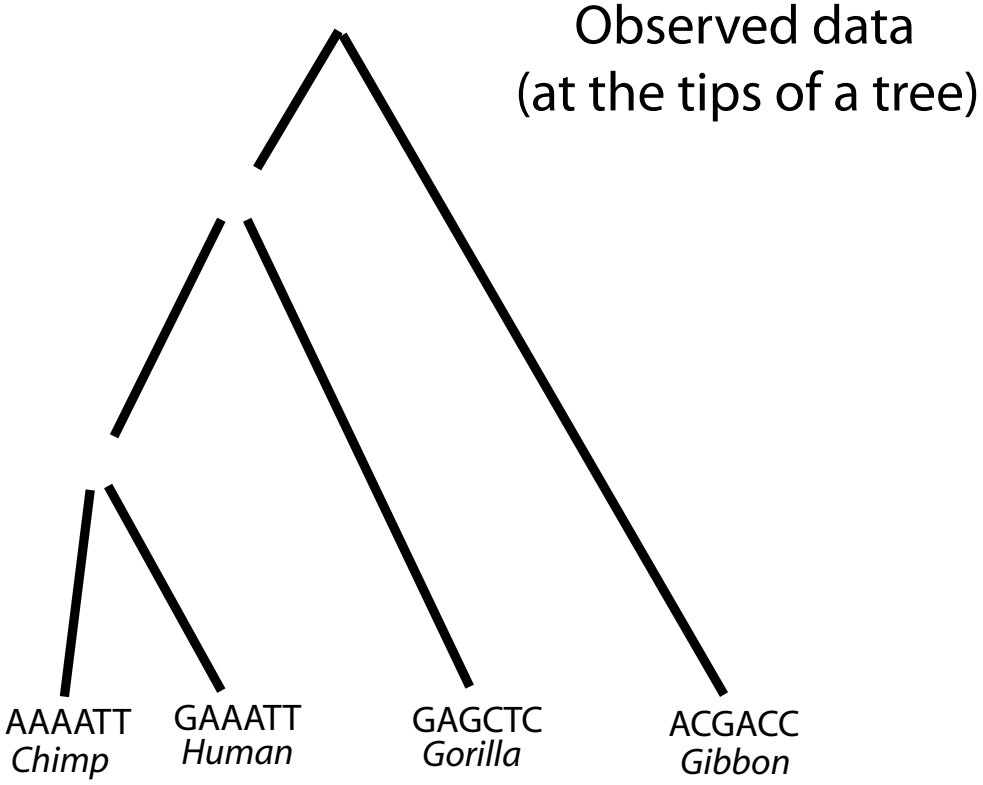
4^N by 4^N rate matrix

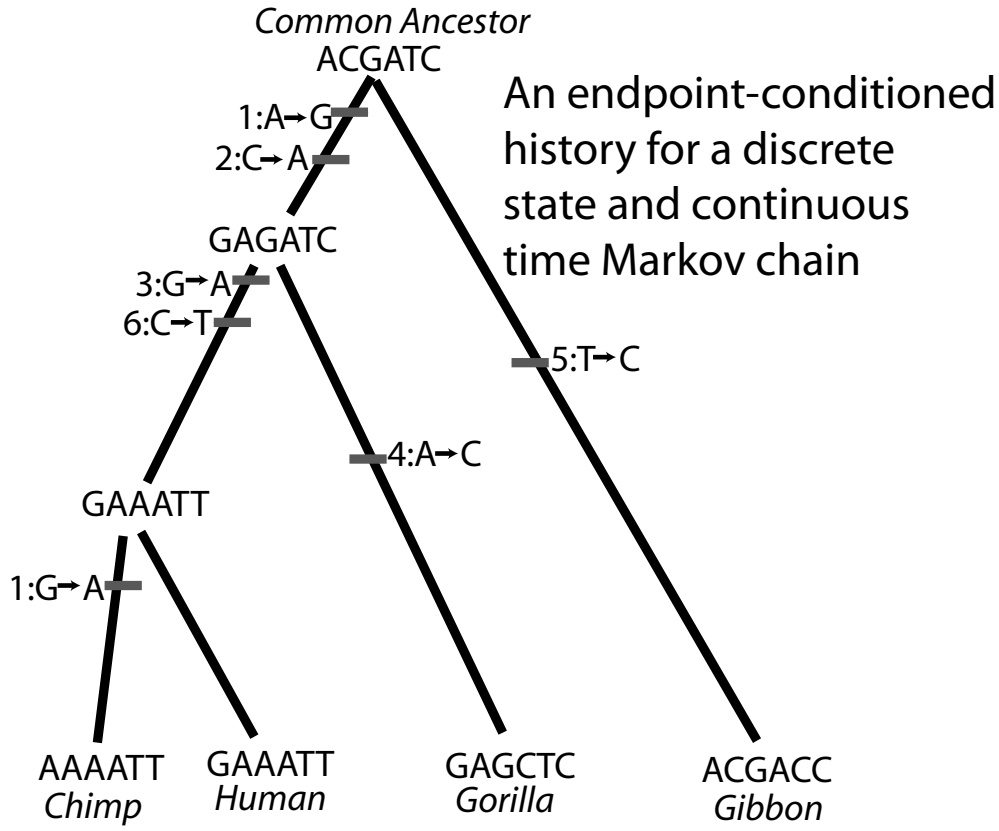
From \ *To*	*AA…AA*	*AA…AC*	*AA…AG*	*AA…AT*	*AA…CA*	…	*TT…GT*	*TT…TA*	*TT…TC*	*TT…TG*	*TT…TT*
AA…AA	–	+	+	+	+		0	0	0	0	0
AA…AC	+	–	+	+	0		0	0	0	0	0
AA…AG	+	+	–	+	0		0	0	0	0	0
AA…AT	+	+	+	–	0		0	0	0	0	0
AA…CA	+	0	0	0	–		0	0	0	0	0
…											
TT…GT	0	0	0	0	0		–	0	0	0	+
TT…TA	0	0	0	0	0		0	–	+	+	+
TT…TC	0	0	0	0	0		0	+	–	+	+
TT…TG	0	0	0	0	0		0	+	+	–	+
TT…TT	0	0	0	0	0		+	+	+	+	–

An infinite number of possible evolutionary histories are consistent with sequences at the beginning and end of a branch on a tree.

If we cannot add up all of these histories, then maybe we can still sample these histories according to their probabilities (this is called "endpoint-conditioned sampling")




Observed data
(at the tips of a tree)
AAAATT
Chimp
GAAATT
Human
GAGCTC
Gorilla
ACGACC
Gibbon


Common Ancestor
ACGATC
1:A→G
2:C→A
GAGATC
3:G→A
6:C→T
5:T→C
4:A→C
GAAATT
1:G→A
AAAATT
Chimp
GAAATT
Human
GAGCTC
Gorilla
ACGACC
Gibbon
An endpoint-conditioned history for a discrete state and continuous time Markov chain

Example Rate Matrix (Continuous Time)

FROM \ To	A	C	G	T
A	−5	2	2	1
C	1	−2	0	1
G	3	3	−10	4
T	1	3	1	−5

Exponentially distributed waiting time for change ...

from A has mean 1/5
from C has mean 1/2
from G has mean 1/10
from T has mean 1/5

Respective change probabilities to (A,C,G,T) from ...

A are (0, 0.4, 0.4, 0.2)
C are (0.5, 0, 0, 0.5)
G are (0.3, 0.3, 0, 0.4)
T are (0.2, 0.6, 0.2, 0)

FROM \ To	A	C	G	T
A	−5	2	2	1
C	1	−2	0	1
G	3	3	−10	4
T	1	3	1	−5

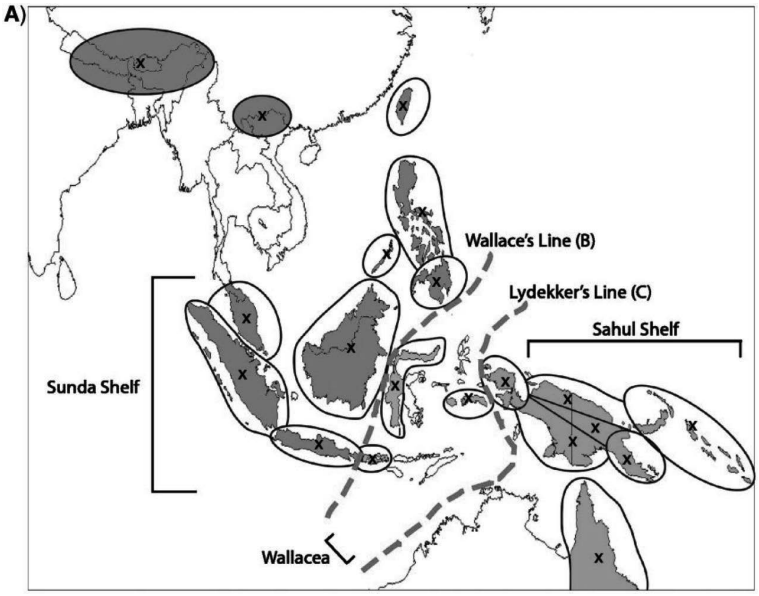
Uniformization **where waiting time to events are exponential with mean 10* (events that do not change state are known as virtual events)**

Exponentially distrib. waiting time for change ...	Respective change probs to (A,C,G,T) from ...	**Uniformized change probs to (A,C,G,T) from ...**
from A has mean 1/5	A are (0, 0.4, 0.4, 0.2)	**A are (0.5, 0.2, 0.2, 0.1)**
from C has mean 1/2	C are (0.5, 0, 0, 0.5)	**C are (0.1, 0.8, 0, 0.1)**
from G has mean 1/10	G are (0.3, 0.3, 0, 0.4)	**G are (0.3, 0.3, 0, 0.4)**
from T has mean 1/5	T are (0.2, 0.6, 0.2, 0)	**T are (0.1, 0.3, 0.1, 0.5)**

Uniformization **Idea: Convert process to Poisson process by making waiting time distributions identical among states. Do this by adding "virtual events" that do not alter the state.**

***Note: Any number >= 10 could have been chosen**

Biogeographic history of Malesian Rhododendron.



Landis et al. employ endpoint conditioned sampling to infer species ranges change on a phylogeny (total geographic area divided into 20 discrete ranges for this example)

Figure 8a from Landis M J et al. Syst Biol 2013;62:789-804

Data augmentation strategies employed to study protein evolution with dependence due to protein structure

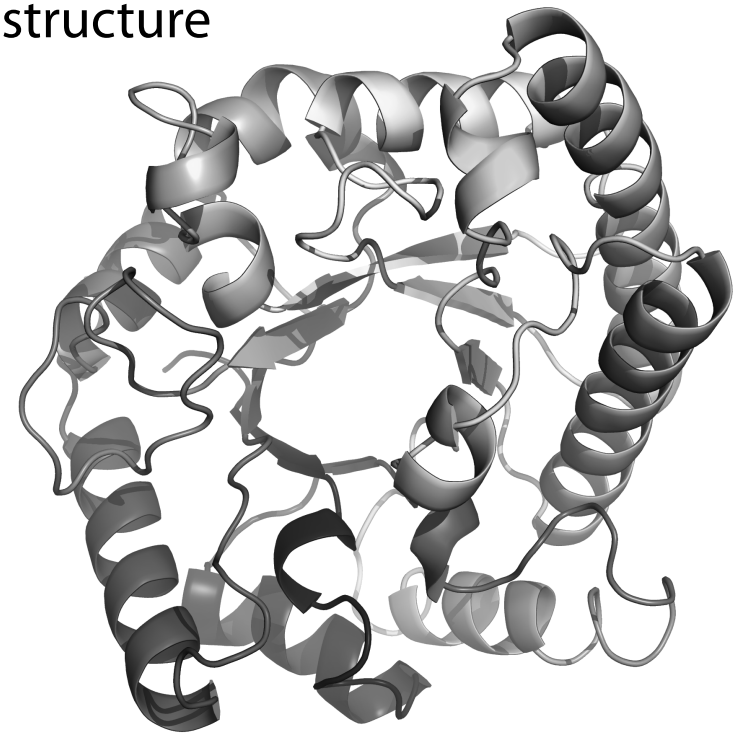
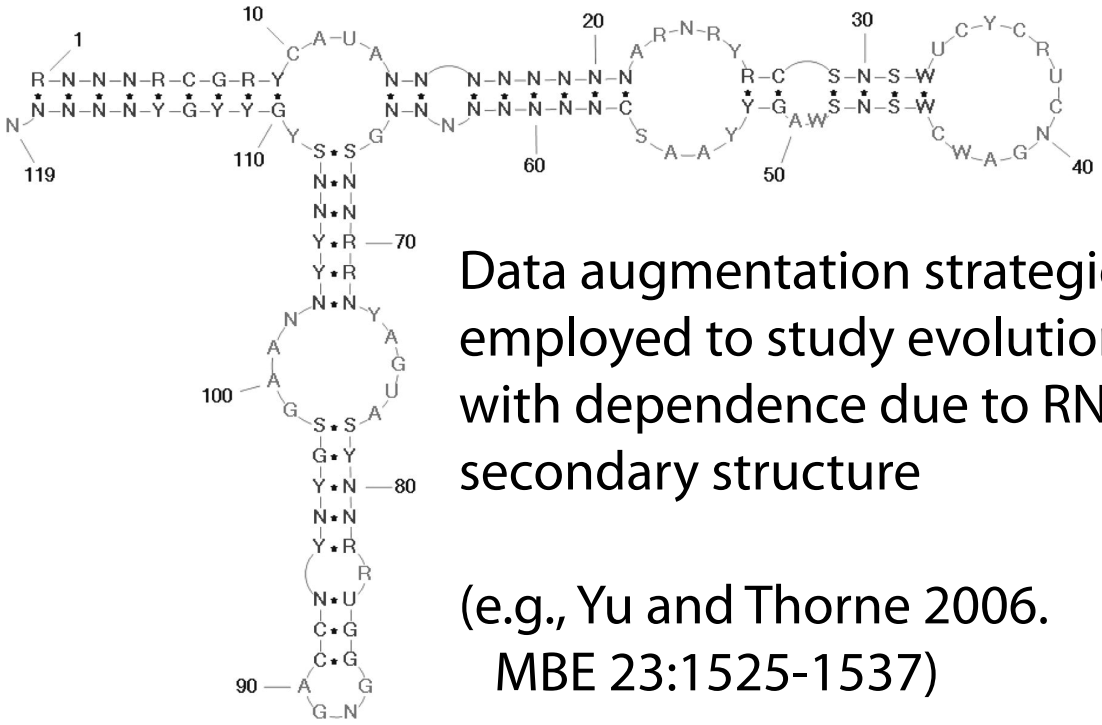


image from http://www.topsan.org/Proteins/JCSG/3qxb

see...

Robinson et al. 2003. MBE 18: 1692-1704

Rodrigue et al. MBE 2006 23:1762-1775 and Gene 2005 347:207-217.



Data augmentation strategies employed to study evolution with dependence due to RNA secondary structure

(e.g., Yu and Thorne 2006. MBE 23:1525-1537)

Data augmentation strategies employed to study context-dependent substitution in mammals

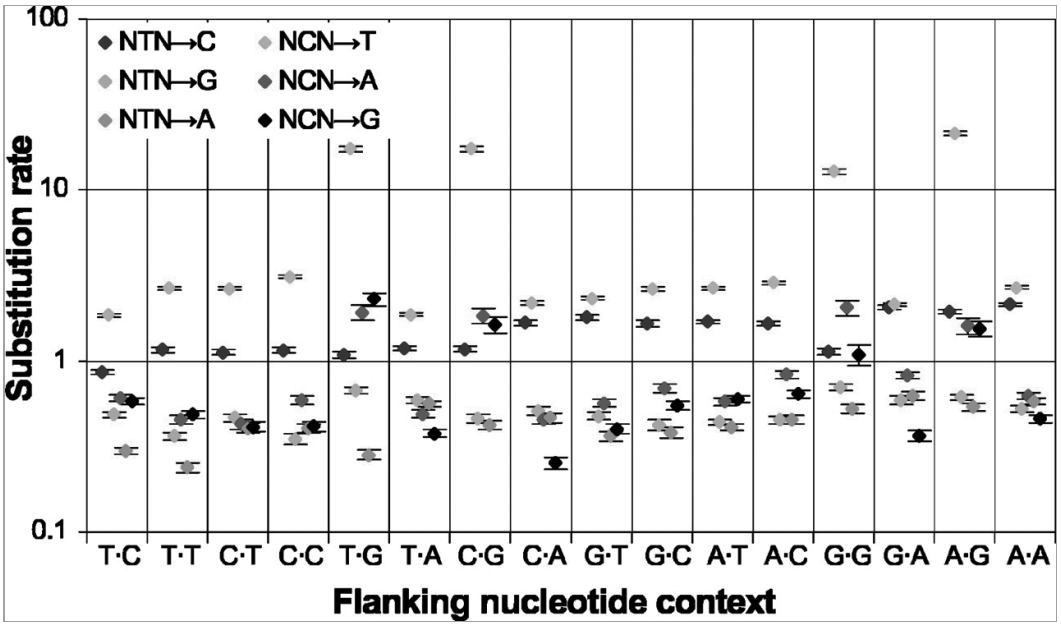


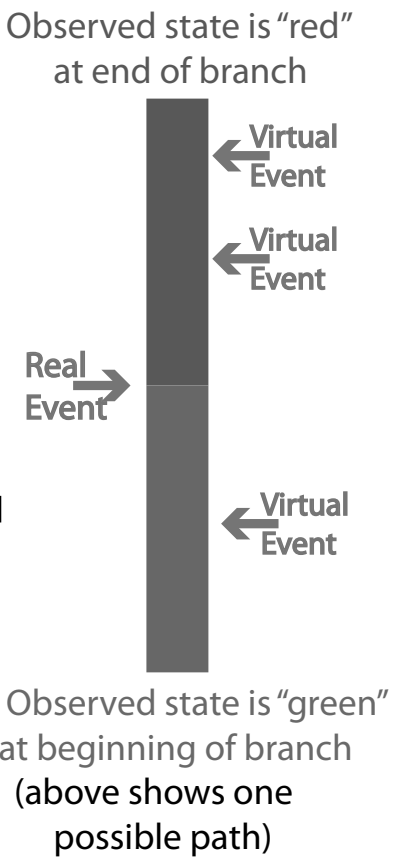
Figure 4 from Hwang and Green. 2004. PNAS 101:13994-14001.

Rao-Teh algorithm: Combines Gibbs Sampling and Uniformization to yield endpoint-conditioned samples.

See Rao and Teh. 2013.
Journal of Machine Learning 14:3295-3320.

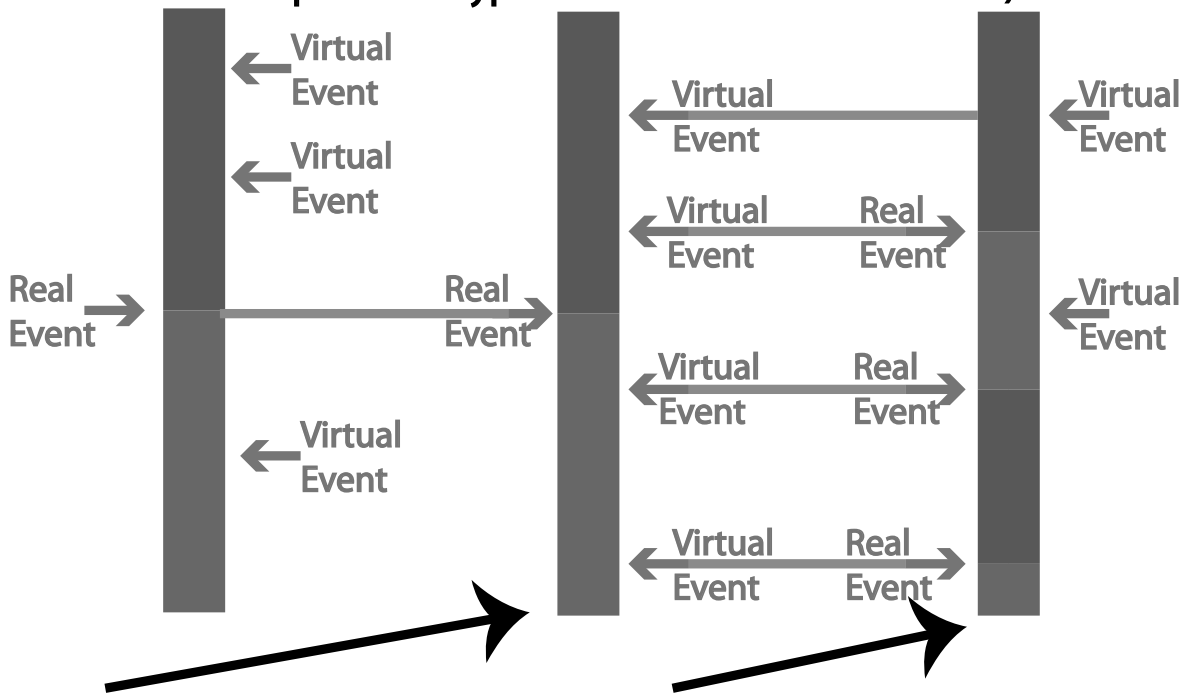
"Usual" uniformization may not scale well to large state space because requires calculation of transition probabilities.

Rao-Teh uniformization is well-suited to evolutionary inference with large state space and sparse rate matrices (computation proportional to product of state space size and number of "neighbors" of typical state).



"Virtual" events do not actually change character state. Real events do change character state.

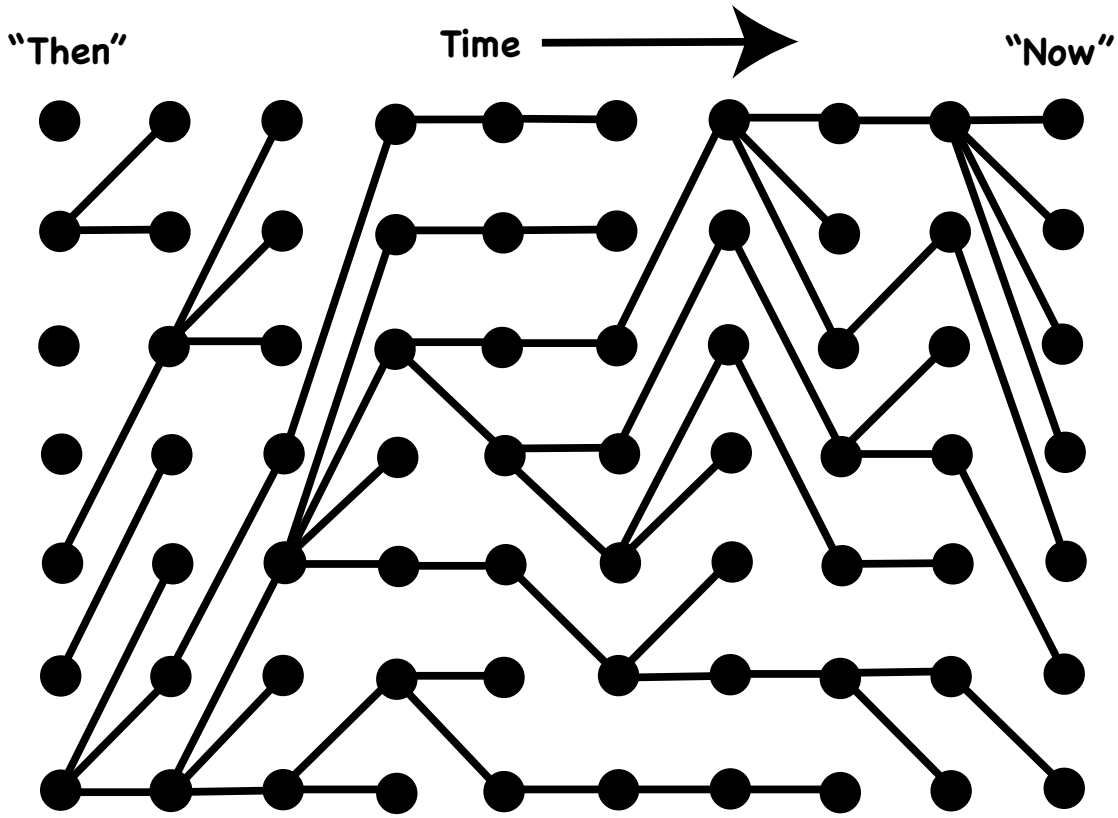
Rao-Teh algorithm (1. Resample virtual events conditional on real ones. 2. Resample event types conditional on event times)

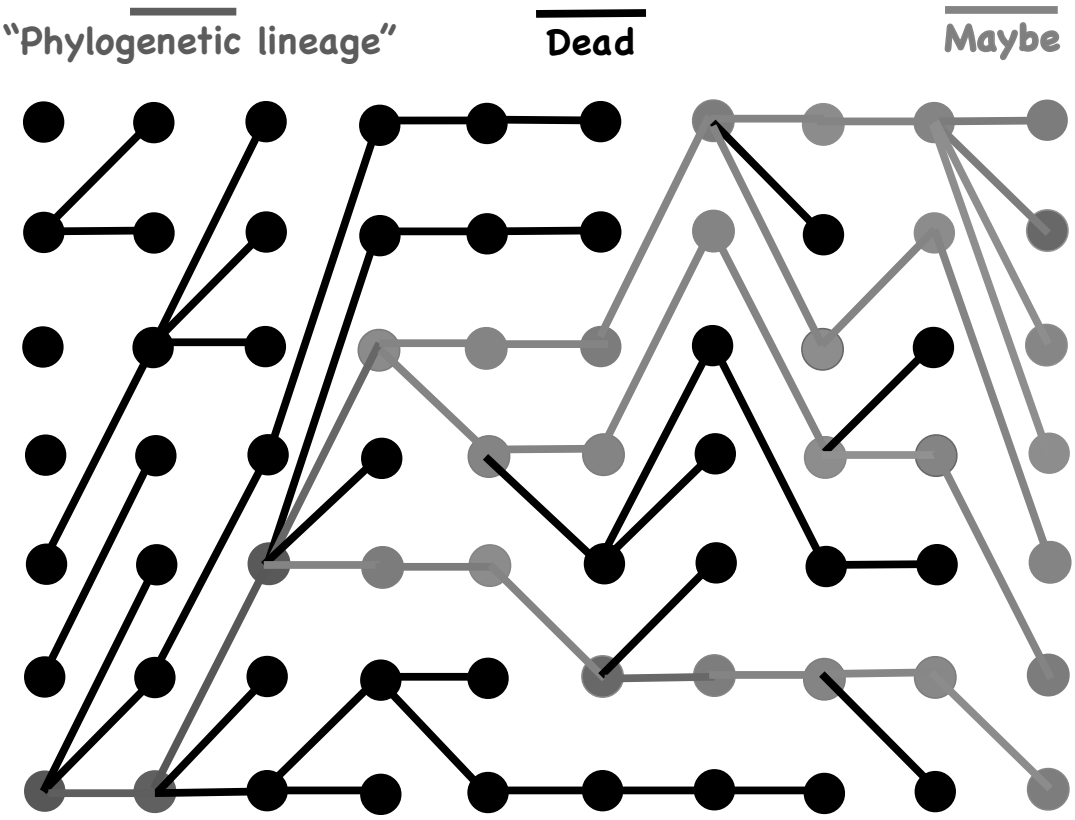


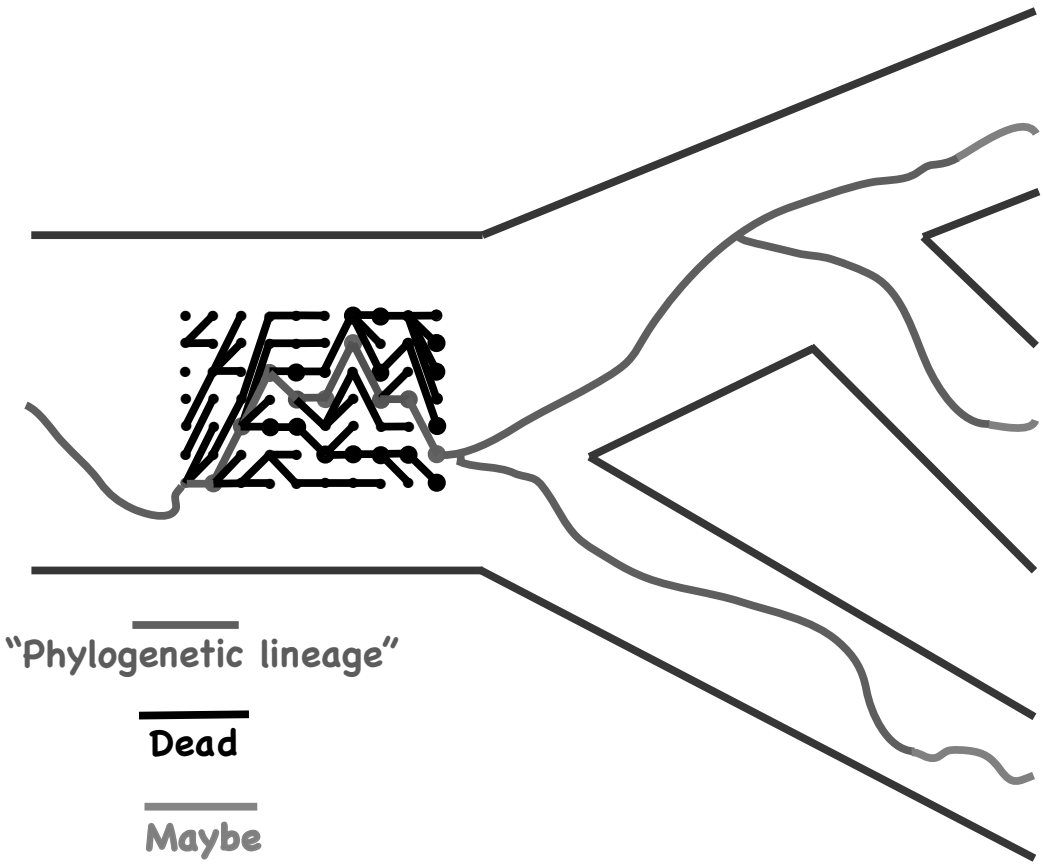
First, use Poisson distribution to resample virtual events conditional on real events

Second, resample event type conditional on event times and endpoints (dynamic programming)

What justifies the assumption of phylogenetic models that sequences change over time according to a Markov process?




"Phylogenetic lineage"
Dead
Maybe


"Phylogenetic lineage"
Dead
Maybe

Fixation probabilities depend on the other alleles in the population

Protein Evolution References

Averof, M., A. Rokas, K.H. Wolfe, and P.M. Sharp. 2000. Evidence for a high frequency of simultaneous double-nucleotide substitutions. Science. **287**:1283–1286.

Cao Y, Adachi J, Janke A, Paabo S, Hasegawa M (1994) Phylogenetic relationships among eutherian orders estimated from inferred sequences of mitochondrial proteins: Instability of a tree based on a single gene. J Mol Evol 39: 519–527

Dayhoff, M.O., R.V. Eck, and C.M. Park. 1972. A model of evolutionary change in proteins. Pp. 89–99 in M.O. Dayhoff, ed. Atlas of protein sequence and structure, vol. 5, National Biomedical Research Foundation, Washington D.C.

Dayhoff MO, Schwartz RM, Orcutt BC (1978) A model of evolutionary change in proteins. Pp. 345–352 in M.O. Dayhoff, ed. Atlas of protein sequence structure, vol. 5, suppl. 3. National Biomedical Research Foundation, Washington D.C.

Goldman N, Yang Z. 1994. A codon–based model of nucleotide substitution for protein–coding DNA sequences. Mol. Biol. Evol. 11:725–736.

Gonnet, G.H., M.A. Cohen, and S.A. Benner. 1992. Exhaustive matching of the entire protein sequence database. Science 256:1443–1445.

Halpern, A., and W.J. Bruno. 1998. Evolutionary distances for protein-coding sequences: Modeling site-specific residue frequencies. Mol. Biol. Evol. **15**:910–917.

Jones DT, Taylor WR, Thornton JM (1992) The rapid generation of mutation data matrices from protein sequences. CABIOS 8:275–282

Kishino H, Miyata T, Hasegawa M (1990) Maximum likelihood inference of protein phylogeny and the origin of chloroplasts. J Mol Evol 31:151–160

Muse, S.V. 1996. Estimating synonymous and nonsynonymous substitution rates. Mol. Biol. Evol. **13**:105–114.

Muse SV, Gaut BS. 1994. A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with applications to the chloroplast genome. Mol. Biol. Evol. 11:715–724.

Nielsen, R., and Z. Yang. 1998. Likelihood models for detecting positively selected amino acid sites and applications to the HIV-1 envelope gene. Genetics **148**:929–936.

Parisi G. and J. Echave. 2001. Structural Constraints and Emergence of Sequence Patterns in Protein Evolution. Mol. Biol. Evol. 18(5):750-756.

Pedersen, A-M. K., C. Wiuf, and F.B Christiansen. 1998. A codon-based model designed to describe lentiviral evolution. Mol. Biol. Evol. 15:1069-1081

Pollock, D.D., W.R. Taylor, and N. Goldman. 1999. Coevolving protein residues: maximum likelihood identification and relationship to structure. J. Mol. Biol. **287**:187–198.

Robinson, D.M., D.T. Jones, H. Kishino, N. Goldman, and J.L. Thorne. 2003. Protein evolution with dependence among codons due to tertiary structure. Mol. Biol. Evol. 20(10):1692-1704.

Schöniger, M., G.L. Hofacker, and B. Borstnik. Stochastic traits of molecular evolution – acceptance of point mutations in native actin genes. J. Theor. Biol. **143**:287–306.

Models of Sequence Evolution: Nucleotide Substitution

Churchill GA (1989) Stochastic models for heterogeneous DNA sequences. Bull Math Biol 51:79–94

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 17:368–376. **(the paper that made maximum likelihood practical for phylogenies**

Felsenstein J., and G.A. Churchill. (1996) A hidden Markov model approach to variation among sites in rate of evolution. Mol. Biol. Evol. 13:93–104

Jensen, J.L., and A.-M. K. Pedersen. 2000. Probabilistic models of DNA sequence evolution with context dependent rates of substitution. Adv. Appl. Prob. 32:499–517.

Lockhart PJ, MA Steel, MD Hendy, D Penny. 1994. Recovering evolutionary trees under a more realistic model of sequence evolution. Mol Biol Evol 11:605-612 **(the LogDet)**

Pedersen, A.-M.K., and J.L. Jensen. 2001. A dependent-rates model and an MCMC-based methodology for the maximum likelihood analysis of sequences with overlapping reading frames. Mol. Biol. Evol. 18:763-776.

Yang Z (1993) Maximum–likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. Mol Biol Evol 10:1396–1401

Yang Z (1994) Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. J Mol Evol 39:306–314

Yang Z (1995) A space–time process model for the evolution of DNA sequences. Genetics 139:993–1005.

Towards more general dependence among sequence positions in molecular evolution...

Hwang, D.G., and P. Green. 2004. Bayesian Markov chain Monte Carlo sequence analysis reveals varying neutral substitution patterns in mammalian evolution. Proc. Natl. Acad. Sci. U.S.A. 101(39):13994-14001

Jensen, J. L., and A. K. Pedersen. 2000. Probabilistic models of DNA sequence evolution with context dependent rates of substitution. Adv. Appl. Prob. 32:499-517

Pedersen A. -M. K. and J. L. Jensen. 2001. A Dependent-Rates Model and an MCMC-Based Methodology for the Maximum-Likelihood Analysis of Sequences with Overlapping Reading Frames. Mol. Biol. Evol. 18(5):763-776.

Siepel, A., and D. Haussler. 2004a. Phylogenetic Estimation of Context-Dependent Substitution Rates by Maximum Likelihood. Mol. Biol. Evol. 21:468-488.

Siepel, A., and D. Haussler. 2004b. Combining phylogenetic and hidden Markov models in biosequence analysis. J Comput Biol. 11:413-428.

Bayesian inference & Markov chain Monte Carlo

Note 1: Many slides for this lecture were kindly provided by Paul Lewis and Mark Holder

Note 2: Paul Lewis has written nice software for demonstrating Markov chain Monte Carlo idea. Software is called "MCRobot" and is freely available at:

http://hydrodictyon.eeb.uconn.edu/people/plewis/software.php

(unfortunately software only works for windows operating system, but see the iMCMC program by John Huelsenbeck at: http://cteg.berkeley.edu/software.html ... also try the MCMC robot ipad app)

Assume we want to estimate a parameter θ with data X.

Maximum likelihood approach to estimating θ finds value of θ that maximizes $\Pr(X \mid \theta)$.

Before observing data, we may have some idea of how plausible are values of θ. This idea is called our prior distribution of θ and we'll denote it $\Pr(\theta)$.

Bayesians base estimate of θ on the posterior distribution $\Pr(\theta \mid X)$.

$$\Pr(\theta \mid X) = \frac{\Pr(\theta, X)}{\Pr(X)} = \frac{\Pr(X \mid \theta)\Pr(\theta)}{\int_\theta \Pr(X, \theta) d\theta}$$

$$= \frac{\Pr(X \mid \theta)\Pr(\theta)}{\int_\theta \Pr(X \mid \theta)\Pr(\theta) d\theta}$$

$$= \frac{\textbf{likelihood} \times \textbf{prior}}{\textbf{difficult quantity to calculate}}$$

Often, determining the exact value of the above integral is difficult.

Problems with Bayesian approachs in general:

1. Disagreements about philosophy of inference
&
Disagreements over priors

2. Heavy Computational Requirements
(problem 2 is rapidly becoming less noteworthy)

Potential advantages of Bayesian phylogeny inference

Interpretation of posterior probabilities of topologies is more straightforward than interpretation of bootstrap support.

If prior distributions for parameters are far from diffuse, very complicated and realistic models can be used and the problem of overparameterization can be simultaneously avoided.

MrBayes software for phylogeny inference is at:

http://mrbayes.sourceforge.net

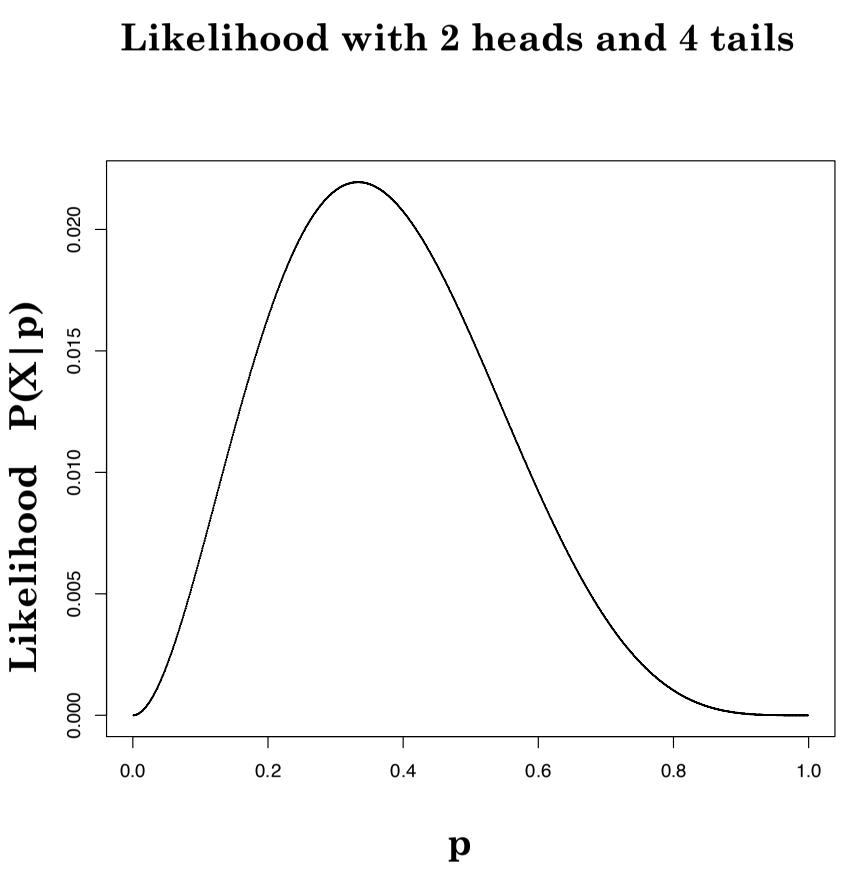
Let p be the probability of heads.

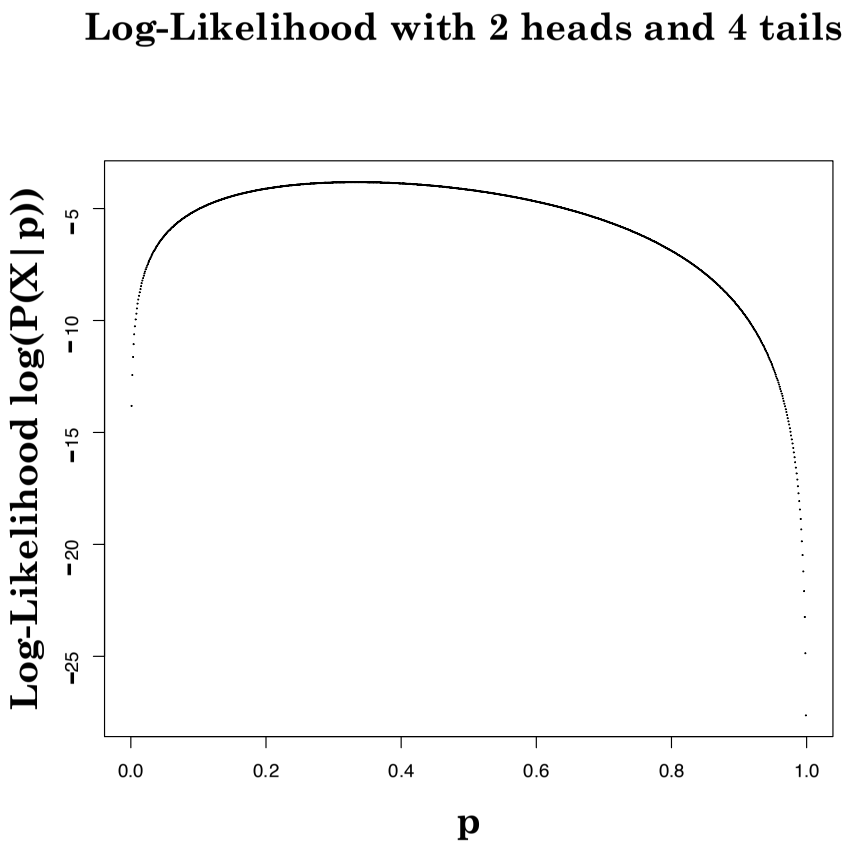
Then 1-p is the probability of tails

Imagine a data set X with these results from flipping a coin

Toss	**1**	**2**	**3**	**4**	**5**	**6**
Result	**H**	**T**	**H**	**T**	**T**	**T**
Probability	**p**	**1-p**	**p**	**1-p**	**1-p**	**1-p**

$$P(X|p) = p^2(1-p)^4$$ ← **almost binomial distribution form**


Likelihood with 2 heads and 4 tails
Likelihood P(X|p)
0.000
0.005
0.010
0.015
0.020
0.0
0.2
0.4
0.6
0.8
1.0
p


Log-Likelihood with 2 heads and 4 tails
Log-Likelihood log(P(X|p))
−5
−10
−15
−20
−25
0.0
0.2
0.4
0.6
0.8
1.0
p

For integers a and b, Beta density B(a,b) is

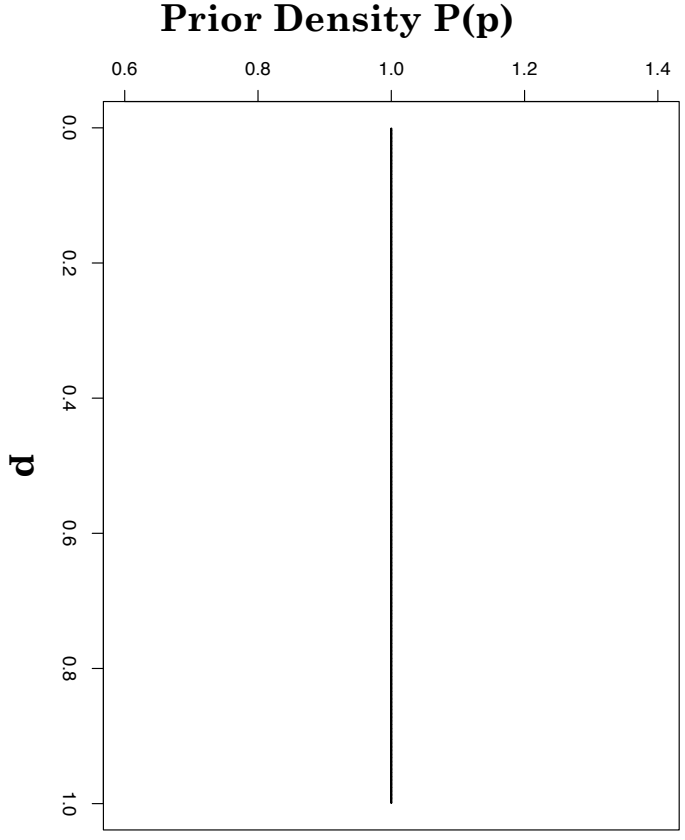
$$P(p) = (a+b-1)!/((a-1)!(b-1)!)\, p^{a-1}(1-p)^{b-1}$$

where p is between 0 and 1.

Expected value of p is a/(a+b)

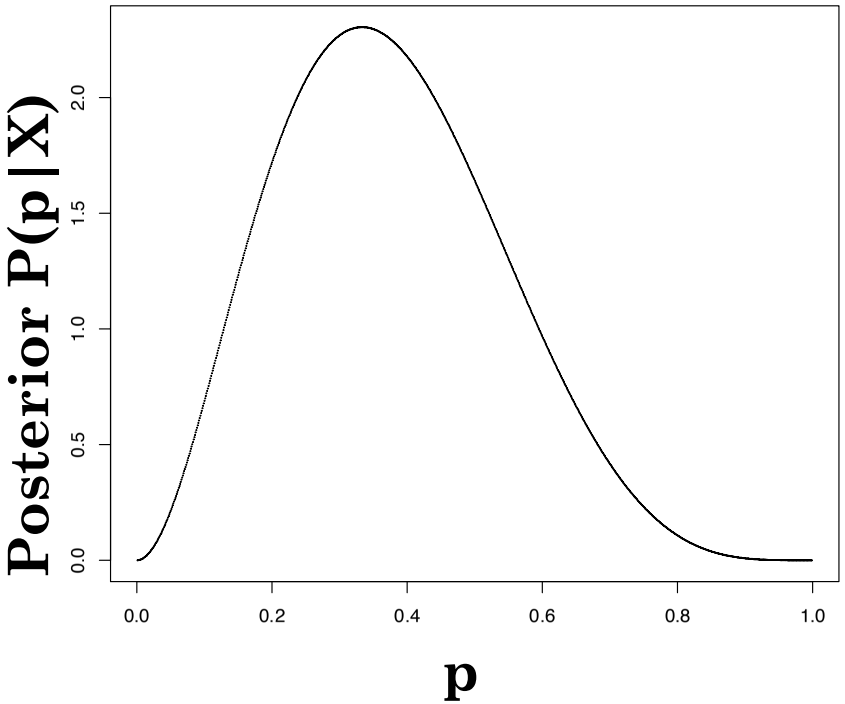
$$\text{Variance of p is } ab/((a+b+1)(a+b)^2)$$

- **Beta distribution is conjugate prior for**
  - **data from binomial distribution**



Uniform Prior Distribution (i.e., Beta(1,1) distribution)

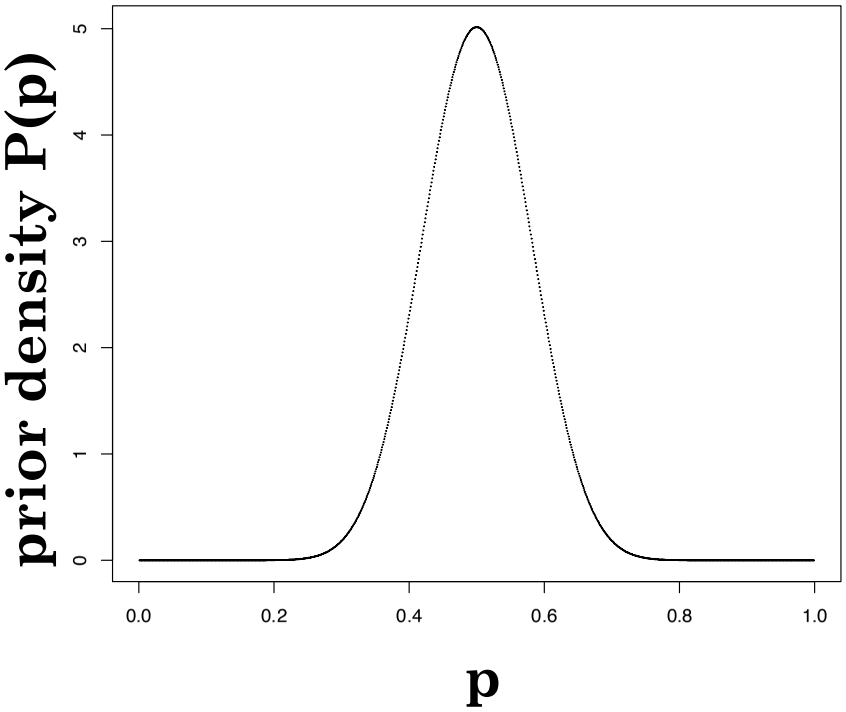
Beta(3,5) posterior from Uniform prior + data (2 heads and 4 tails)



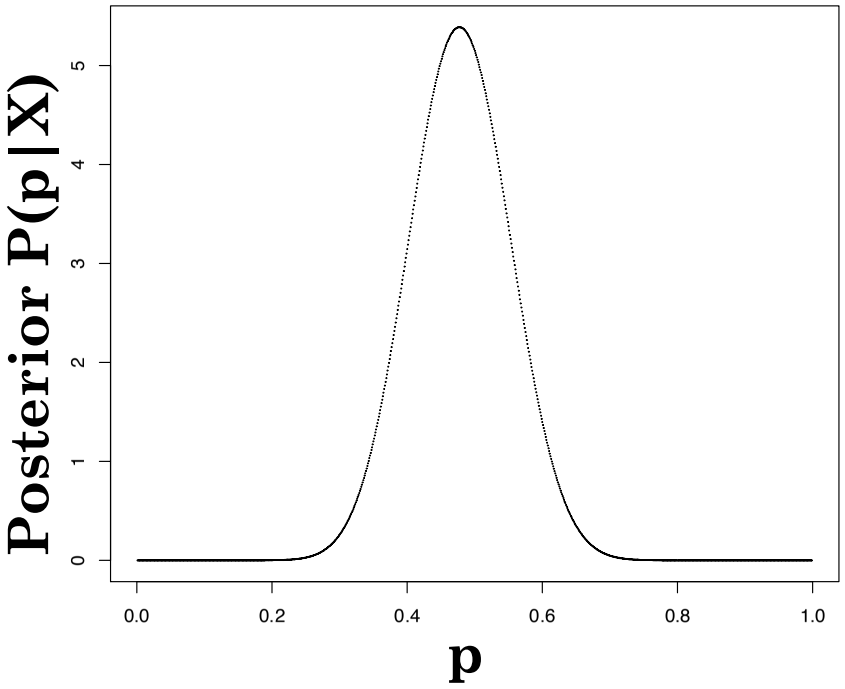
Posterior Mean = 3/(3+5)

Beta(20,20) prior distribution

Prior Mean = 0.5

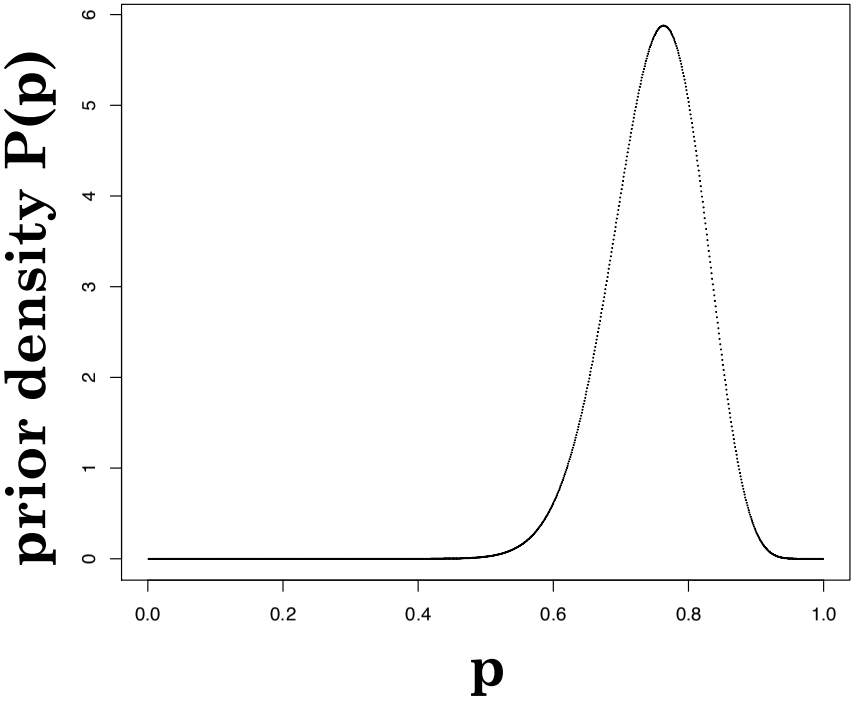


Beta(22,24) posterior from Beta(20,20) prior + data (2 heads and 4 tails)

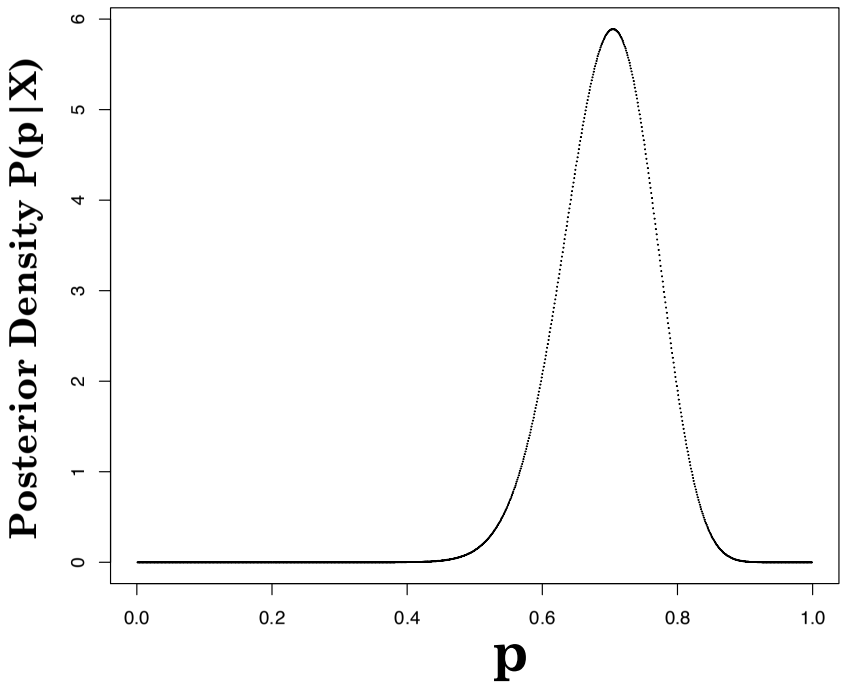


Beta(30,10) prior distribution

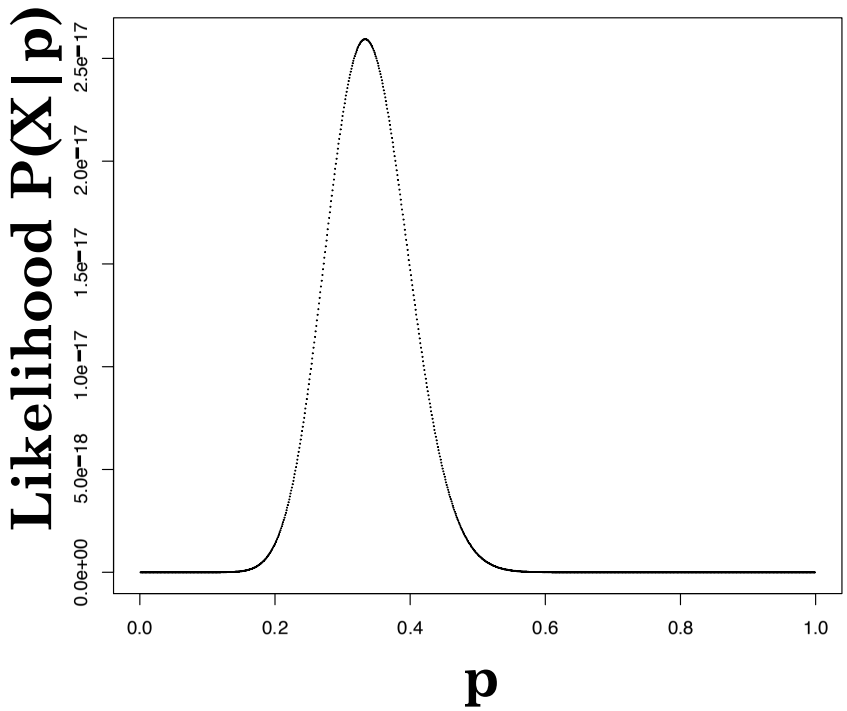
Prior Mean = 0.75



Beta(32,14) posterior from Beta(30,10) prior + data (2 heads and 4 tails)

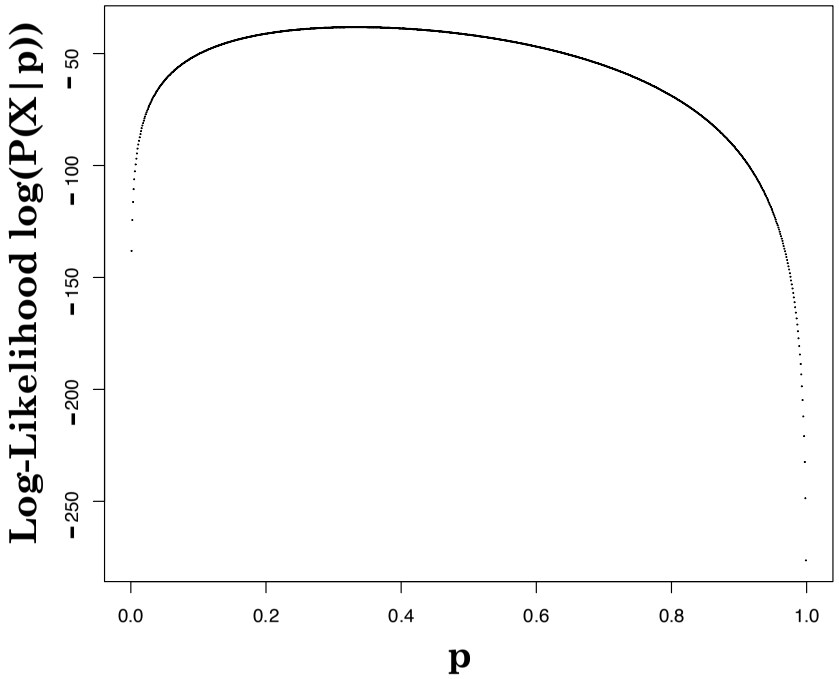


Likelihood with 20 Heads and 40 Tails

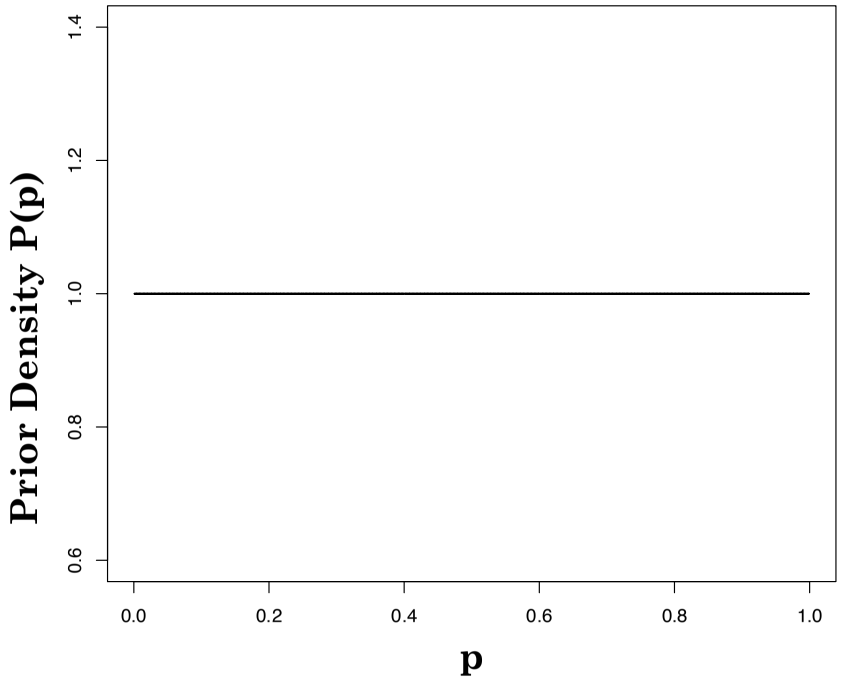


Posterior Mean = 32/(32+14)

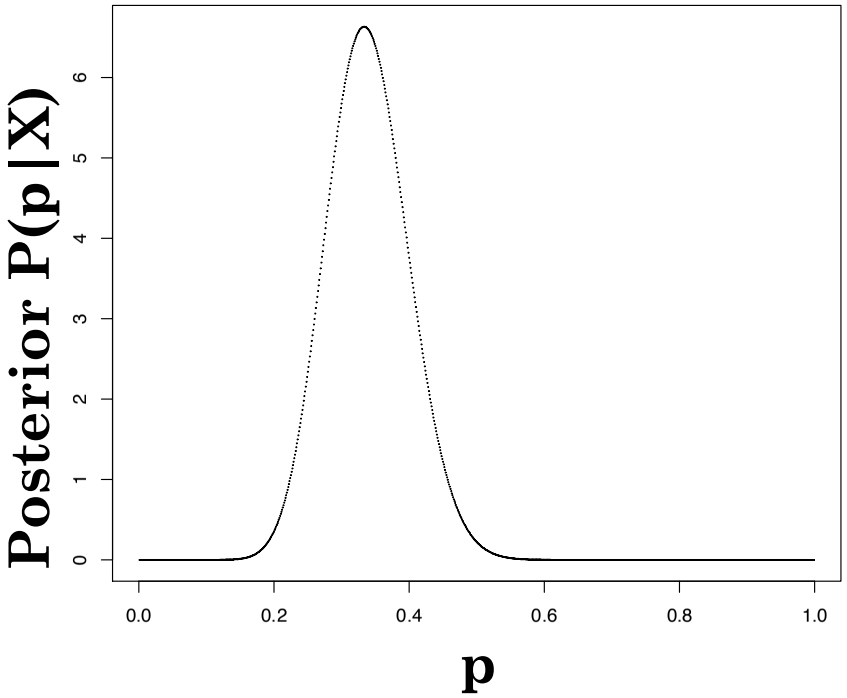
Log-Likelihood with 20 heads and 40 tails



Uniform Prior Distribution (i.e., Beta(1,1) distribution)

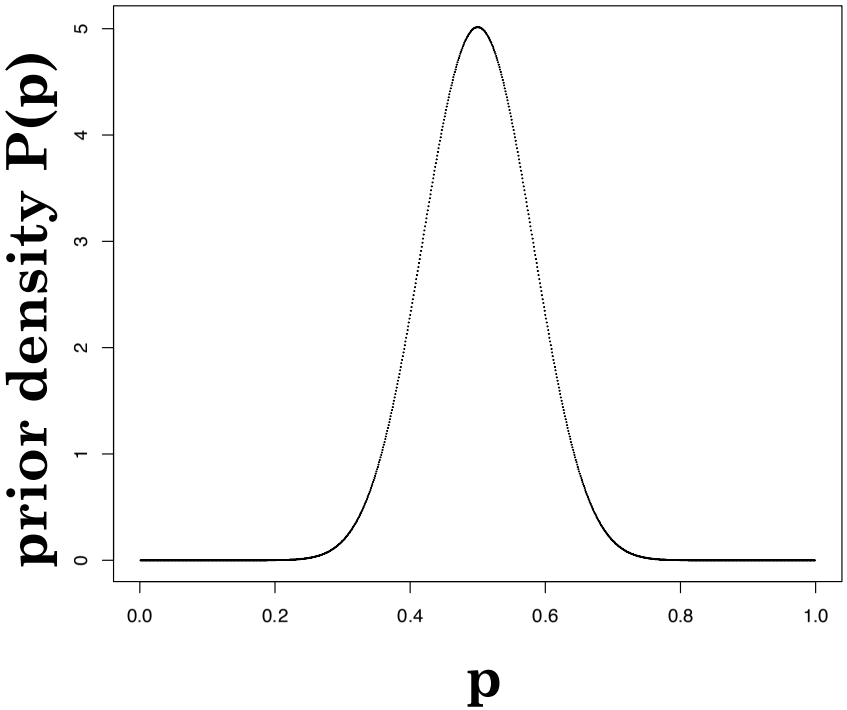


Beta(21,41) posterior from Uniform prior + data (20 heads and 40 tails)

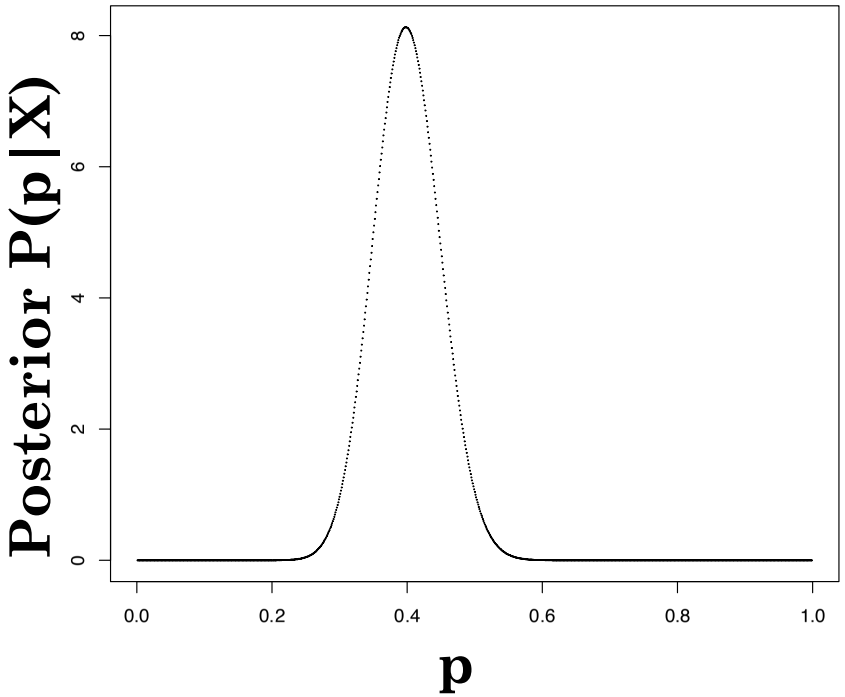


Beta(20,20) prior distribution

Prior Mean = 0.5

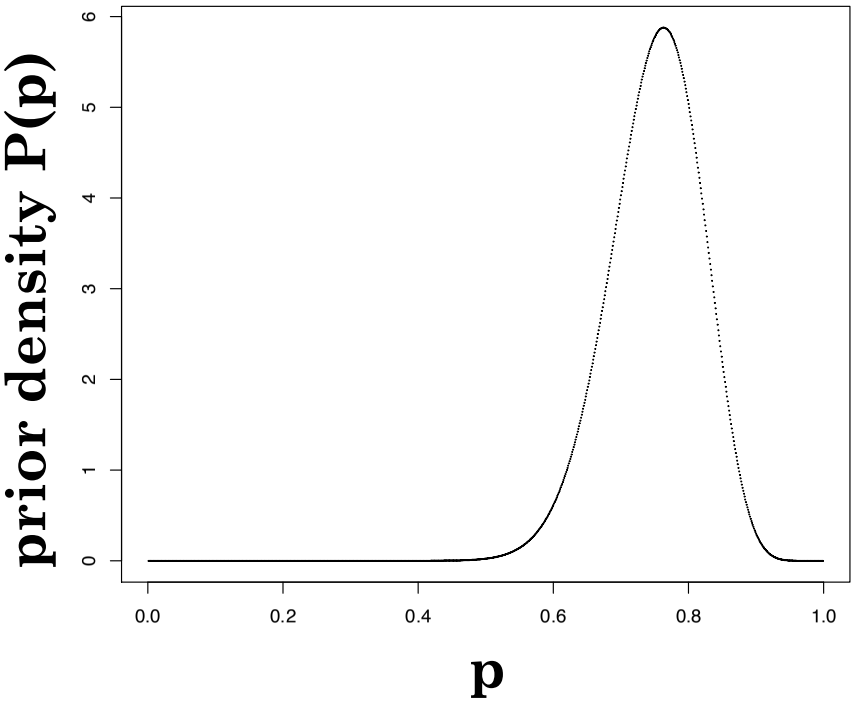


Beta(40,60) posterior from Beta(20,20) prior + data (20 heads and 40 tails)



Beta(30,10) prior distribution

Prior Mean = 0.75



Beta(50,50) posterior from Beta(30,10) prior + data (20 heads and 40 tails)

Markov chain Monte Carlo (MCMC) idea approximates $\Pr(\theta \mid X)$ by sampling a large number of θ values from $\Pr(\theta \mid X)$.

So, θ values with higher posterior probability are more likely to be sampled than θ values with low posterior probability.

Question: How is this sampling achieved?

Answer: A Markov chain is constructed and simulated. The states of this chain represent values of θ. The stationary distribution of this chain is $\Pr(\theta \mid X)$.

In other words, we start chain at some initial value of θ. After running chain for a long enough time, the probability of the chain being at some particular state will be approximately equal to the posterior probability of the state.

Let $\theta^{(t)}$ be the value of θ after t steps of the Markov chain where $\theta^{(0)}$ is the initial value.

Each step of the Markov chain involves randomly proposing a new value of θ based on the current value of θ. Call the proposed value θ^*.

We decide with some probability to either accept θ^* as our new state or to reject the proposed θ^* and remain at our current state.

The Hastings (Hastings 1970) algorithm is a way to make this decision and force the stationary distribution of the chain to be $\Pr(\theta \mid X)$.

According to the Hastings algorithm, what state should we adopt at step $t+1$ if $\theta^{(t)}$ is the current state and θ^* is the proposed state?

Let $J(\theta^*|\theta^{(t)})$ be the "jumping" distribution, i.e. the probability of proposing θ^* given that the current state is $\theta^{(t)}$.

Define r as

$$r = \frac{\Pr(X \mid \theta^*)\Pr(\theta^*)J(\theta^{(t)}|\theta^*)}{\Pr\left(X \mid \theta^{(t)}\right)\Pr\left(\theta^{(t)}\right)J(\theta^*|\theta^{(t)})}$$

With probability equal to the minimum of r and 1, we set

$$\theta^{(t+1)} = \theta^*.$$

Otherwise, we set

$$\theta^{(t+1)} = \theta^{(t)}.$$

For the Hastings algorithm to yield the stationary distribution $\Pr(\theta \mid X)$, there are a few required conditions. The most important condition is that it must be possible to reach each state from any other in a finite number of steps. Also, the Markov chain can't be periodic.

MCMC implementation details:

The Markov chain should be run as long as possible.

We may have T total samples after running our Markov chain. They would be $\theta^{(1)}$, $\theta^{(2)}$, ..., $\theta^{(T)}$. The first B ($1 \leq B < T$) of these samples are often discarded (i.e. not used to approximate the posterior). The period before the chain has gotten these B samples that will be discarded is referred to as the "burn–in" period.

The reason for discarding these samples is that the early samples typically are largely dependent on the initial state of the Markov chain and often the initial state of the chain is (either intentionally or unintentionally) atypical with respect to the posterior distribution.

The remaining samples $\theta^{(B+1)}$, $\theta^{(B+2)}$, ..., $\theta^{(T)}$ are used to approximate the posterior distribution. For example, the average among the sampled values for a parameter might be a good estimate of its posterior mean.

Markov Chain Monte Carlo and Relatives (some important papers)

CARLIN, B.P., and T.A. LOUIS. 1996. Bayes and Empirical Bayes Methods for Data Analysis. Chapman and Hall, London.

GELMAN, A., J.B. CARLIN, H.S. STERN, and D.B. RUBIN. 1995. Bayesian Data Analysis. Chapman and Hall, London.

GEYER, C. 1991. Markov chain Monte Carlo maximum likelihood. Pages 156-163 in Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface. Keramidas, ed. Fairfax Station: Interface Foundation

HASTINGS, W.K. (1970) Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57:97–109

METROPOLIS, N., A.W. ROSENBLUTH, M.N. ROSENBLUTH, A.H. TELLER, and E. TELLER. 1953. Equations of state calculations by fast computing machines. J. Chem. Phys. 21: 1087–1092.

Posterior predictive inference (notice resemblance to parametric bootstrap)

1. Via MCMC or some other technique, get N sampled parameter sets $\theta^{(1)}, \ldots, \theta^{(N)}$ from posterior distribution $p(\theta|X)$

2. For each sampled parameter set $\theta^{(k)}$, simulate a new data set $X^{(k)}$ from $p(X|\theta^{(k)})$

3. Calculate a test statistic value $T(X^{(k)})$ from each simulated data set and see where test statistic value for actual data $T(X)$ is relative to simulated distribution of test statistic values.

From Huelsenbeck et al. 2003. Syst Biol 52(2): 131-158

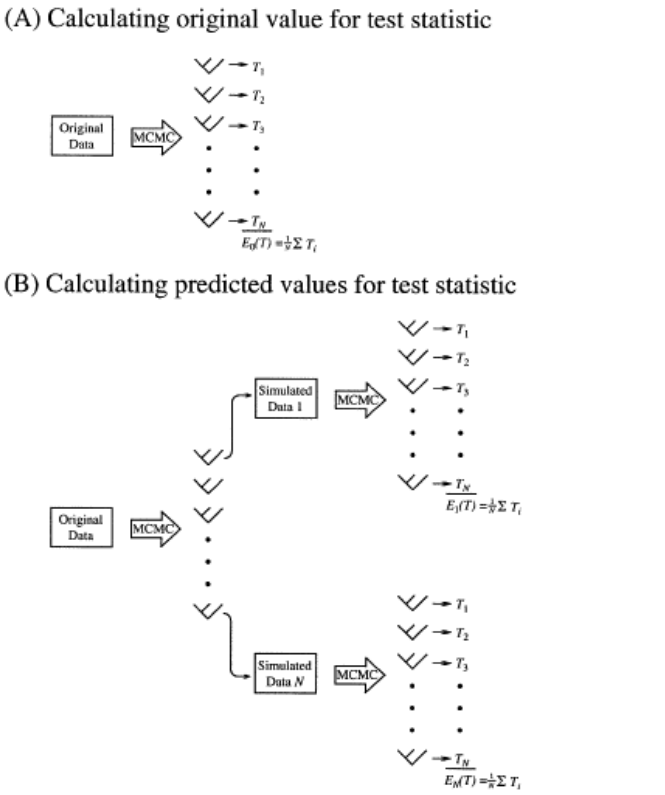


FIGURE 4. An example of how posterior predictive P values are calculated for a test statistic, T. The observed value for the test statistic is calculated by averaging over the posterior probability distribution of parameters. We use MCMC to draw parameter values from the posterior probability distribution. The predictive distribution is calculated by simulating new data using parameter values from the posterior probability distribution of parameters. Each simulated data set is treated exactly as was the original data. The predictive P value is the proportion of the test statistics from the simulated data that exceed the observed value.

Notation for following pages:

X data

M_i, M_j: Models i and j

θ_i, θ_j: parameters for models i and j

$p(X|\theta_i, M_i), p(X|\theta_j, M_j)$: likelihoods

Bayes factor

$$\frac{p(M_i|X)}{p(M_j|X)} = \frac{p(M_i)p(X|M_i)/p(X)}{p(M_j)p(X|M_j)/p(X)}$$

$$= \frac{p(M_i)}{p(M_j)} \times \frac{p(X|M_i)}{p(X|M_j)}$$

Left factor is called *prior odds* and right factor is called *Bayes factor*.

Bayes factor is ratio of *marginal likelihoods* of the two models.

$$BF_{ij} = \frac{p(X|M_i)}{p(X|M_j)}$$

According to wikipedia, Jeffreys (1961) interpretation of BF_{12} (1 representing one model and 2 being the other):

BF_{12}	Interpretation
$<$ 1 : 1	Negative (supports M_2)
1 : 1 to 3 : 1	Barely worth mentioning
3 : 1 to 10 : 1	Substantial
10 : 1 to 30 : 1	Strong
30 : 1 to 100 : 1	Very Strong
$>$ 100 : 1	Decisive

$$BF_{ij} = \frac{p(X|M_i)}{p(X|M_j)}$$

Bayes factors hard to compute because marginal likelihoods hard to compute:

$$p(X|M_i) = \int_{\theta_i} p(X|M_i, \theta_i) p(\theta_i|M_i) d\theta_i$$

Important point to note from above: Bayes factors depend on priors $p(\theta_i|M_i)$ because marginal likelihoods depend on priors!

How to approximate/compute marginal likelihood?

$$p(X|M_i) = \int_{\theta_i} p(X|M_i, \theta_i) p(\theta_i|M_i) d\theta_i$$

Harmonic mean estimator of marginal likelihood (widely used but likely to be terrible and should be avoided):

$$\frac{1}{p(X|M_i)} \doteq \frac{1}{N} \sum_{k=1}^{N} \frac{1}{p(X|\theta_i^{(k)}, M_i)}$$

where $\theta_i^{(k)}$ are sampled from posterior $p(\theta_i|X, M_i)$.

Important papers regarding Bayesian Model Comparison ...

Posterior Predictive Inference in Phylogenetics: J.P. Bollback. 2002. Molecular Biology and Evolution. 19:1171-1180

Harmonic Mean and other techniques for estimating Bayes factors: Newton and Raftery. 1994. Journal of the Royal Statistical Society. Series B. 56(1):3-48.

more reliable ways to approximate marginal likelihood

Thermodynamic Integration to Approximate Bayes Factors (adapted to molecular evolution data): Lartillot and Philippe. 2006. Syst. Biol. 55:195-207

Improving marginal likelihood estimation for Bayesian phylogenetic model selection. W. Xie, P.O. Lewis, Y. Fan, L. Kao, M-H Chen. 2011. Syst Biol. 60(2):150-160.

Choosing among partition models in Bayesian phylogenetics. Y. Fan, R. Wu, M-H Chen, L Kuo, P.O. Lewis. 2011. Mol. Biol. Evol. 28(1):523-532.

Markov chain Monte Carlo without likelihoods. P. Marjoram, J. Molitor, V. Plagnol, and S. Tavare. 2003. PNAS USA. 100(26): 15324-15328.

H. Jeffreys. The Theory of Probability (3e). Oxford (1961); p. 432

M.A. Beaumont, W. Zhang, D.J. Balding. Approximate Bayesian Computation in Population Genetics. 2002. Genetics 162:2025-2035.

Paul Lewis' MCMC Robot Demo

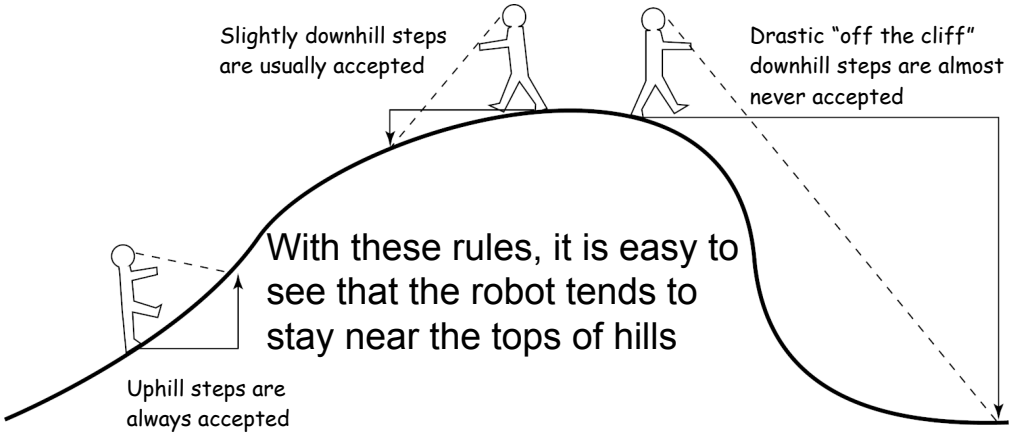
Target distribution:

- Mixture of bivariate normal "hills"
- inner contours: 50% of the probability
- outer contours: 95%

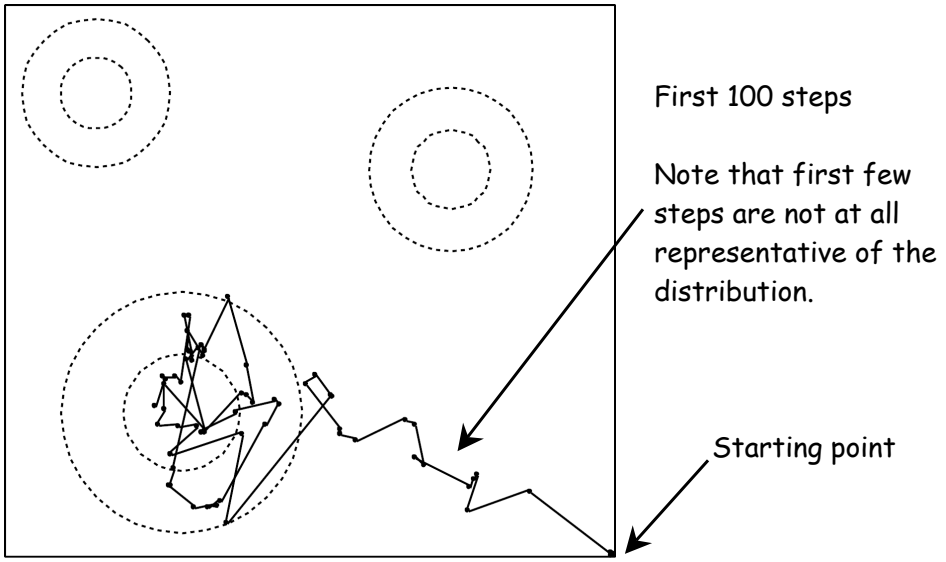
Proposal scheme:

- random direction
- gamma-distributed step length
  (mean 45 pixels, s.d. 40 pixels)
- reflection at edges

MCMC robot rules



Burn-in



Problems with MCMC approaches:

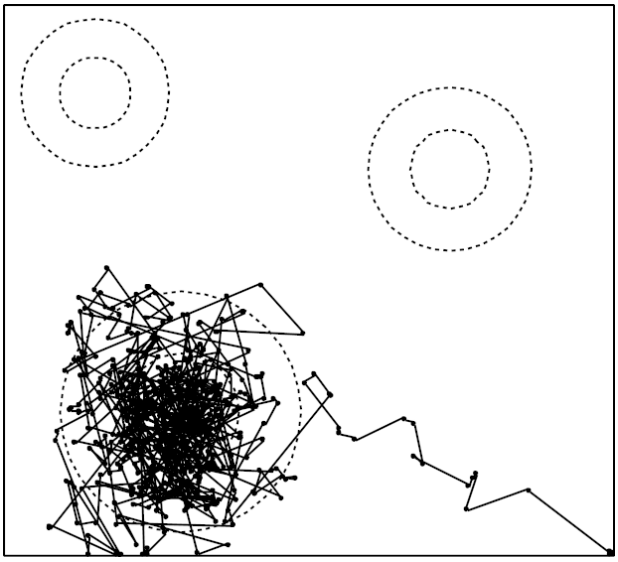
1. They are difficult to implement. Implementation may need to be clever to be computationally tractable and programming bugs are a serious possibility.

2. For the kinds of complicated situations that biologists face, it may be very difficult to know how fast the Markov chain converges to the desired posterior distribution.

There are diagnostics for evaluating whether a chain has converged to the posterior distribution but the diagnostics do not provide a guarantee of convergence.

A GOOD DIAGNOSTIC : MULTIPLE RUNS !!

Just how long is a long run?

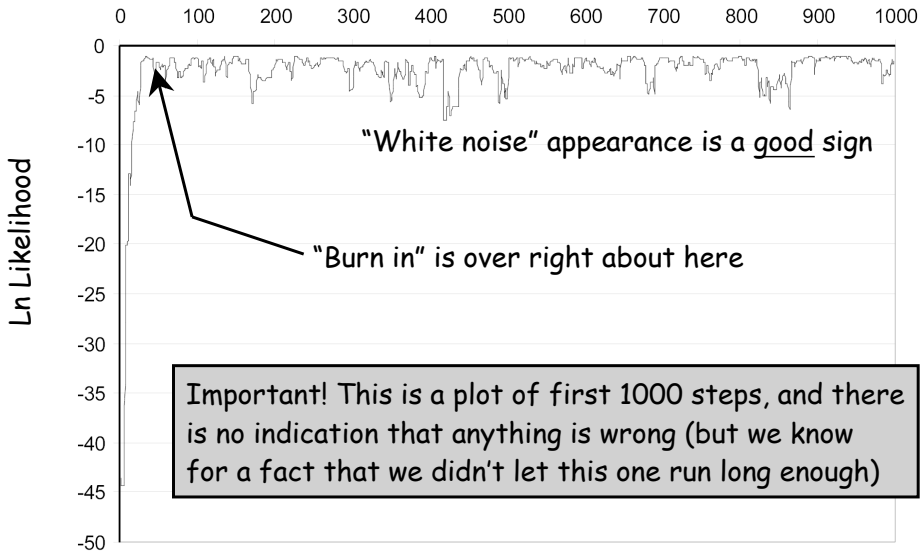


What would you conclude about the target distribution had you stopped the robot at this point?

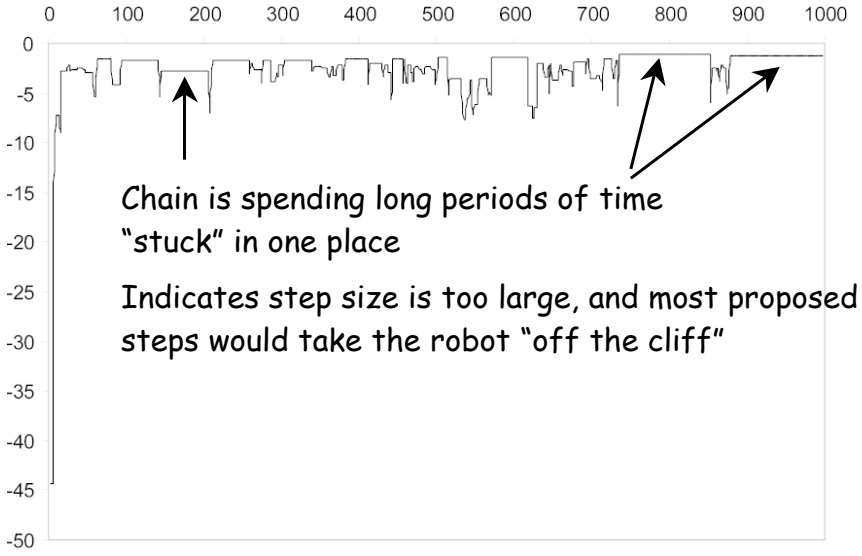
One way to detect this mistake is to perform several independent runs.

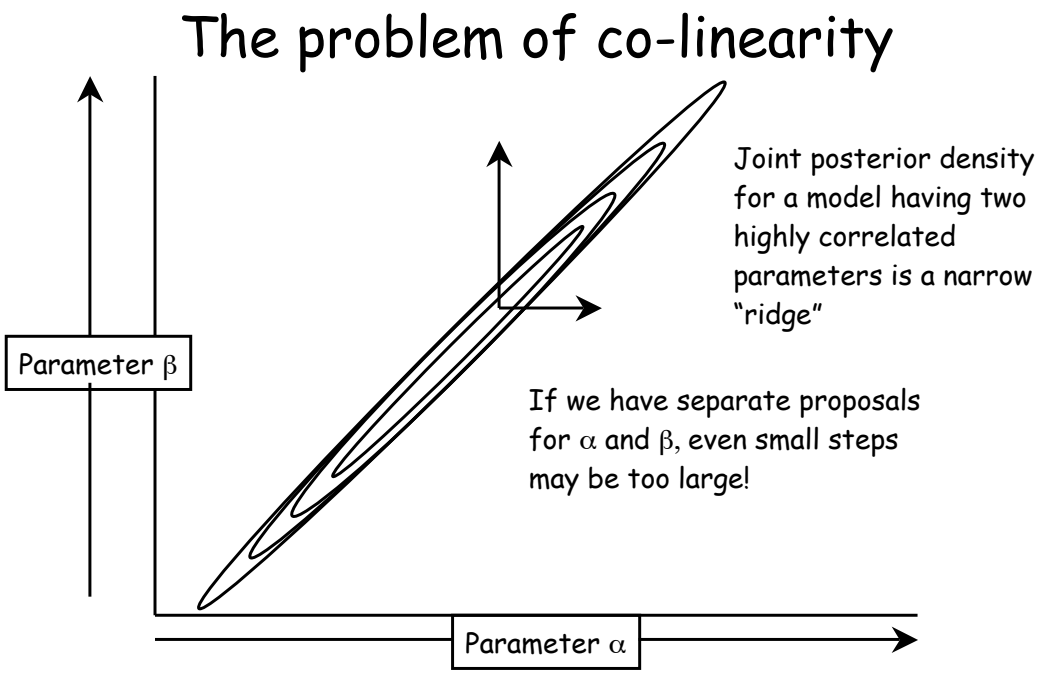
Results different among runs? Probably none of them were run long enough!

History plots



Slow mixing





Some material on Bayesian model comparison and hypothesis testing

1. Some Bayesians dislike much hypothesis testing because null hypotheses often are known *a priori* to be false and $p-value$ depends both on "how" wrong null is and on amount of data.

2. Posterior predictive inference for assessing fit of models (see next pages)

3. Bayes factors for comparing models (see next pages)

The Tradeoff

- *Pro:* Proposing big steps helps in jumping from one “island” in the posterior density to another
- *Con:* Proposing big steps often results in poor mixing
- Solution: Better proposals - MCMCMC

Huelsenbeck has found that a technique called Metropolis-Coupled Markov chain Monte Carlo (i.e., MCMCMC !! or MC^3) suggested by C.J. Geyer is useful for getting convergence with phylogeny reconstruction.

The idea of MCMCMC is to run multiple Markov chains in parallel.

One chain will have stationary distribution that is the posterior of interest.

The other chains will approximate posterior distributions that are various degrees more smooth that than the posterior distribution of interest.

Each chain is run separately, except that occasionally 2 chains are randomly picked and a proposal to switch the states of these two chains is made. This proposal is randomly accepted or reject with the appropriate probability

Metropolis-coupled Markov chain Monte Carlo (*MCMCMC*, or MC^3)

- MC^3 involves running **several chains simultaneously**
- The **cold chain** is the one that counts, the rest are **heated chains.**

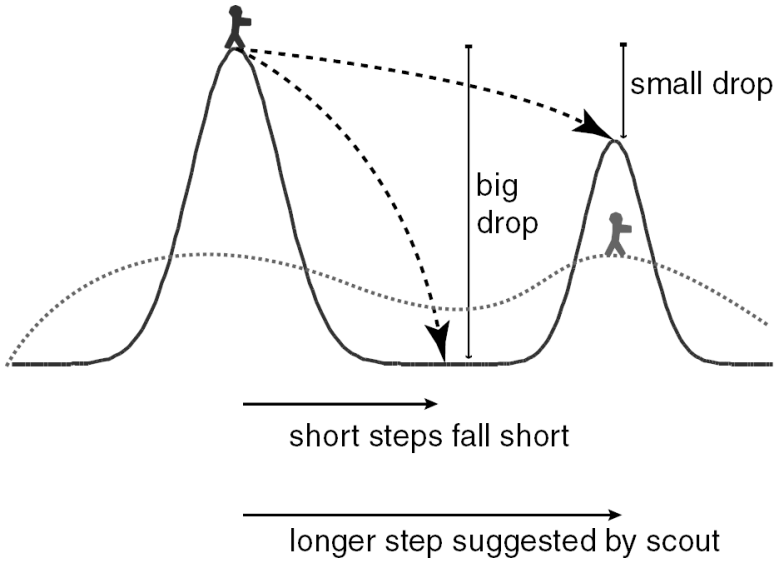
What is a heated chain?

- Instead of using R, to make acceptance or rejection decisions, heated chains use:

$$R^{\frac{1}{1+H}}$$

- In MrBayes: H = Temperature*(Chain's index)
- The cold chain has index 0
- Heated chains explore the surface more freely
- Occasionally, you propose to switch the positions of 2 of the chains

Heated chains act as scouts



Phylogeny Priors: For phylogeny inference, parameters might represent topology, branch lengths, base frequencies, transition-transversion ratio, etc.

Each parameter needs specified prior distribution.
For example...

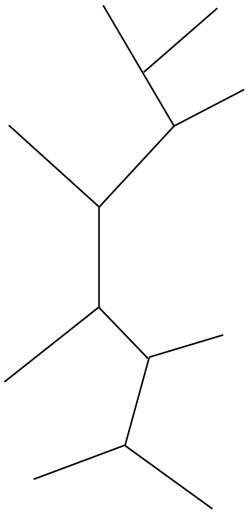
1. All unrooted topologies can be considered equally probable a priori. Given topology, all branch lengths between 0 and some big number could be considered equally likely a priori

2. All combinations of base frequencies could be considered equally likely a priori

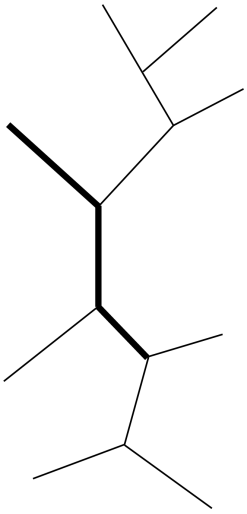
3. The transition-transversion ratio could have a prior distribution that is uniform between 0 & some big number.

... and so on.

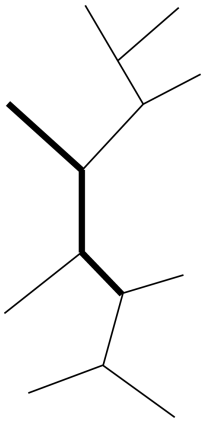
Moving through Tree Space
Larget Simon Local Move



Step 1: Randomly select an internal branch and 2 of its neighbors

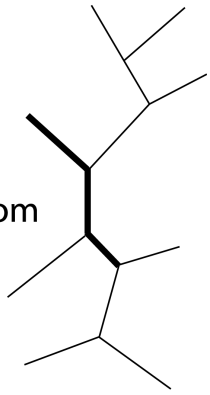


Moving through Tree Space
Larget Simon Local Move

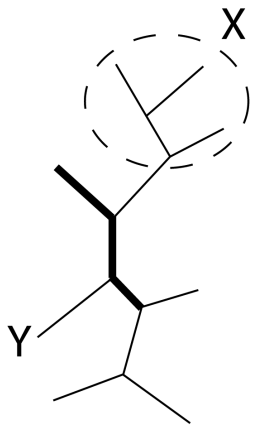


Step 2: Shrink or expand the selected segment by a random amount

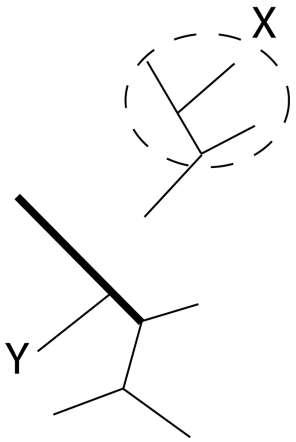
$$m^* = m\, e^{\lambda(u-.5)}$$



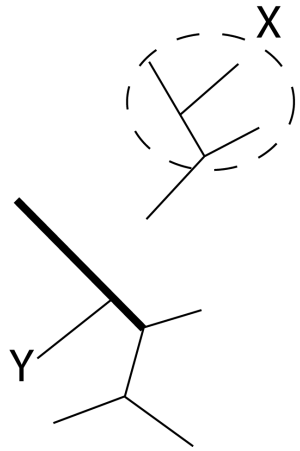
Moving through Tree Space
Larget Simon Local Move



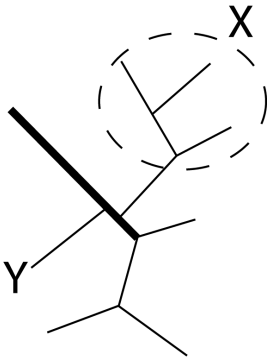
Step 3: Randomly select 1 of the 2 branches that intersect with the selected segment, and detach it



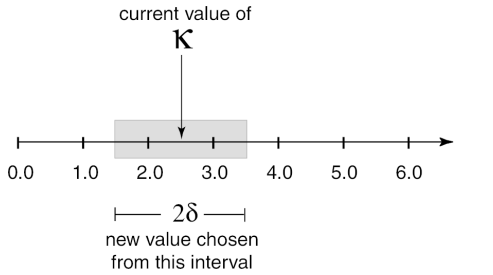
Moving through Tree Space
Larget Simon Local Move



Step 4: Randomly reattach the clade X somewhere along the selected segment. This might result in a new topology.

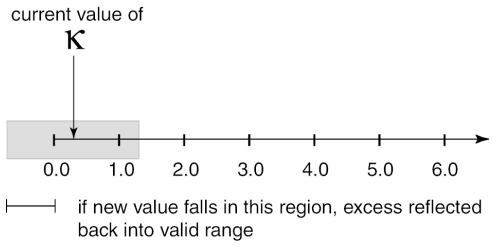


Moving through parameter space



Using κ (ratio of the transition rate to the transversion rate) as an example of a model parameter.

Proposal distribution is the uniform distribution on the interval $(\kappa-\delta, \kappa+\delta)$



A larger δ means the sampler will attempt to make larger jumps on average.

Putting it all together

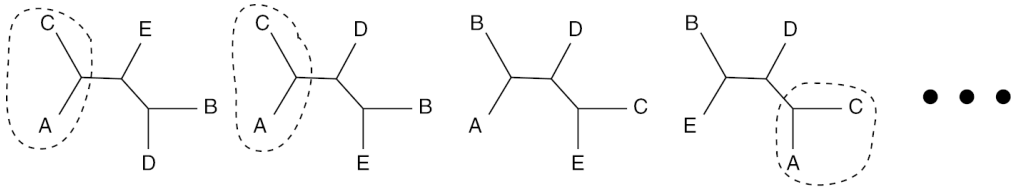
- Start with an initial tree and model parameters (often chosen randomly).
- Propose a new, randomly-selected move. Accept or reject the move (**Walking**).
- Every *k* generations, save tree, branch lengths and all model parameters (**Thinning**).
- After *n* generations, summarize the sample using histograms, means, credibility intervals, etc. (**Summarizing**).

Sampling the chain tells us:

- Which tree has the highest posterior probability?
- What is the probability that "tree X" is the true tree?
- What values of the parameters are most probable?

What if we are only interested in one grouping?

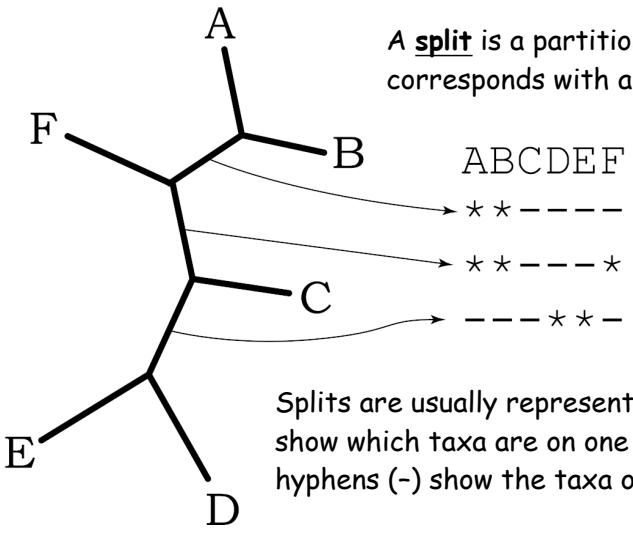
Which of the trees in the MCMC run contained the clade (e.g. *A* + *C*) ?



1 1 0 1

The *proportion* of trees with *A* and *C* together in our sample approximates the posterior probability that *A* and *C* are sister to each other.

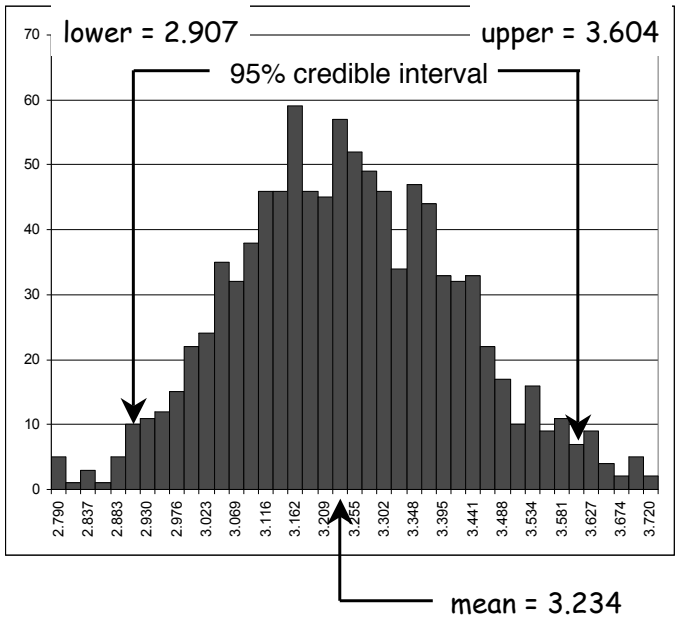
Split (a.k.a. clade) probabilities



A **split** is a partitioning of taxa that corresponds with a particular branch.

Splits are usually represented by strings: asterisks (*) show which taxa are on one side of the branch, and the hyphens (–) show the taxa on the other side.

Posteriors of model parameters



Histogram created from a sample of 1000 kappa values.

From: Lewis, L., & Flechtner, V. (2002. *Taxon* **51**:443-451)

MrBayes Lab

Note: This computer lab exercise was written by Paul O. Lewis. Paul has graciously allowed us to use and modify the lab for the Summer Institute in Statistical Genetics. Thanks, Paul!

The main goal is to become familiar with running MrBayes (Ronquist and Huelsenbeck, 2003) and interpreting its output. Don't rush things. If you don't finish the lab, you can always download MrBayes when you get home and finish it there.

Stylistic conventions used in this document:

Commands understood by the PAUP* and/or MrBayes programs are in `this fixed width font`. Questions for you to answer are in *italics*. Important things to note are in **bold face**. Names of files are in this font. Web site URLs look `http://like_this` and are clickable links.

Exercise 1: Basics

1. You can download MrBayes 3.2.2 (Ronquist et al., 2012) for your platform from `http://mrbayes.net`. If you are using Windows, you have to install a .NET framework from Microsoft in order to run the MrBayes installer.

2. On Mac, you can then run MrBayes by launching the /Applications/Utilities/Terminal and then typing mb. It is easiest to copy the algaemb.nex file into your home directory.

   On Windows, it is easiest to put the algaemb.nex file into the C:\Program Files\mrbayes32 directory which contains the mrbayes executable. Then you can launch MrBayes by double-clicking on the icon.

   MrBayes by double-clicking the file name. When you see the "MrBayes >" prompt, type the following and then press the Enter key:

   ```
   execute algaemb.nex
   ```

   The algaemb.nex file is essentially the same as the one used for the PAUP* lab, and you should refer to that handout for a description of the data file. This version differs only in formatting: some features of the NEXUS file format that are understood by PAUP* are not handled cleanly by MrBayes. For example, the CHARACTERS block is not recognized by MrBayes, and the `symbols` and `labels` statements within the `format` command are not recognized.

   Assuming MrBayes reads the file without errors, set up an MCMC run using the HKY85 model with discrete gamma among site rate heterogeneity. There are three commands in particular that you will need to know something about in order to set up a typical run: `lset`, `prset` and `mcmc`. The command `mcmcp` is identical to `mcmc` except that it does not actually start a run.

   For each of these commands you can obtain online information by typing help followed by the command name: for example, `help prset`. This may spew more information than will fit on your screen, however. If this happens you can tell the Windows command prompt to remember more of the text that it displays:

   (a) click the icon at the left edge of the title bar of the MrBayes window,

   (b) choose "Properties",

   (c) click the "Layout" tab, and

(d) change "Screen Buffer Size, Height" to a large number (e.g. 10000).

After describing every parameter of a particular command, the `help` command, displays a table of the current values for the parameters.

3. First, establish the prior distributions that will be used for the model parameters. Use:

```
help prset
```

to see the default set of priors.

Here is a table of the parameters and the prior distributions that we will use for this exercise:

Base frequencies	"flat" Dirichlet distribution
Shape parameter	Exponential distribution, mean 1.0
TRatio	"flat" Beta distribution – also known as Uniform[0,1.0]

The prior distribution over the tree topologies and branch lengths are:

Topologies	Uniform prior over all (fully-resolved) tree topologies.
Branch lengths	Exponential distribution, mean 0.1

```
prset statefreqpr=dirichlet(1.0, 1.0, 1.0, 1.0)
prset shapepr=exp(1)
prset tratiopr=beta(1,1)
prset topologypr = uniform
prset brlenspr=unconstrained:exp(10)
```

A Dirichlet distribution with all four parameters 1.0 assigns equal prior probability density to all (legal) combinations of base frequencies. To make the prior on base frequencies favor equal base frequencies, you could use a value such as Dirichlet(50.0, 50.0, 50.0, 50.0). Such a prior would tend to tie down the base frequencies close to 0.25 unless the data strongly contradicted equal base frequencies.

The `tratiopr` is poorly named. In previous versions of MrBayes, the setting controlled the prior on the rate ratio between transitions and transversions. This parameter, κ, is defined over the range [0,∞). In recent versions of MrBayes, the prior is described in terms of a [0,1.0] parameter that is a transformation of κ. If we call this parameter x then $\frac{x}{1-x} = \kappa$. Thus, using a Beta(10,1) prior for "`tratiopr`" has an expectation that the rate of a transition is 10 times the rate of a transversion.

The `unconstrained` keyword in the prior for branch lengths indicates that a non-clock model is being used (i.e. the branch lengths are not constrained to obey the molecular clock). Note that MrBayes expects you to specify the hazard parameter for exponential distributions, not the mean. The hazard parameter is the inverse of the mean, so specifying `exp(10)` is the same as specifying an exponential prior distribution having mean 0.1.

4. Next, set up the HKY-gamma model. This amounts to telling MrBayes that we will be using a two-parameter model (a model that distinguishes between transitions and transversions) and that we want gamma-distributed rate heterogeneity with 4 categories.

```
lset nst=2 rates=gamma ngammacat=4
```

5. Specifying the outgroup uses the `outgroup` command (similar to PAUP*):

```
outgroup Anacystis_nidulans
```

6. Now set up the MCMC run itself:
   `mcmcp ngen=50000 filename=algaeout samplefreq=50 printfreq=200 savebrlens=yes nruns = 1`

   The command parameters used here (and their meanings) are:

   - `ngen` – the total number of generations,
   - `filename` – the **prefix** to use for output file names. Trees will be written to algaeout.t; samples of parameter values will be in a tab-delimited format in algaeout.p; and statistics about the MCMC algorithm itself will be written in algaeout.mcmc
   - `samplefreq` – the frequency with which to sample trees and parameters,
   - `printfreq` – the frequency with which to print a one-line progress report.
   - `savebrlens` – tells MrBayes to save branch length information in the tree file that is output.
   - `nruns=1` – tells MrBayes to run only one simultaneous MCMCMC analysis. This disables the automatic run stopping criterion (more on this feature later) and means that MrBayes will pay attention to the `ngen` parameter setting that we give it. Note: this parameter does *not* control the number of heated chains in MCMCMC (that is the `nchains` parameter and the default is to run 4 chains: 3 "heated" chains, and the "cold" chain that is sampled).

7. All that is left now is to start the run:

   `mcmc`

8. While MrBayes runs, it shows one-line progress reports. The first column is the generation number. The next few columns show the log-likelihoods of the separate chains that are running, with the cold chain indicated by square brackets rather than parentheses. The last complete column is the best estimate of the time remaining until the run completes.

   After it finishes, type `no` in response to the query "Continue with chain? (yes/no):" and then press the Enter key.

9. MrBayes will now report various statistics about the run, such as the percentage of the time it was able to accept proposed changes of various sorts, or the percentage of proposed swaps between chains that it accepted. These percentages should, ideally, all be between about 20% and 40%, but as long as they are not extreme (e.g. 1% or 99%) then things went well.

   MrBayes saves information in several files. One of these should be called algaeout.p. This is the file in which the model parameter values were saved. This file is saved in tab-delimited format so that it is easy to import into a spreadsheet program such as Excel. This would allow you to create history plots showing variation in the parameter values over the course of the run. This file is important because it shows the likelihood values (which can be used as a crude indication of how long to make the burnin period). My own examination of the file I produced using MrBayes suggests that the burnin should end at or around 2,000 iterations ("generations" in MrBayes lingo). By this point, MrBayes has wandered into the heart of the posterior distribution after starting with random tree topology (which is almost certainly a very poor fit to the data).

   The second file of importance is the algaeout.t file, which contains the sampled trees. This is an ordinary NEXUS tree file, and could be read into another program (e.g. PAUP* or TreeView) that understands the NEXUS file format. We will ask MrBayes to summarize this file for us today using the `sumt` command:

```
sumt file=algaeout burnin=41 nruns=1
```

Why did we specify 41 as the burnin?[1]

10. You should now see a tree showing all compatible splits (a.k.a. bipartitions) with branches labeled with their split posterior probabilities. The last tree shown will be the same topology, but will make the lengths of the branches proportional to the posterior mean branch length for each split present in the tree. The posterior mean branch length is a simple average. Every time a tree with some split S is visited during the running of the chain, its current branch length is added to a sum and the posterior mean branch length is obtained by simply dividing this sum by the total number of times that split S was present in the current tree during the run.

    Do organisms with green plant chloroplasts (all species other than Olithodiscus *and* Anacystis*) form a group in this tree? What is the posterior probability of the split separating the green plant chloroplast group from the chromophyte + cyanobacterium group?*[2]

    The `sumt` produces 3 useful files:

    (a) algaeout.con has the majority-rule consensus of the sampled trees in NEXUS format. The tree occurs twice. In the first version, internal nodes are given name that are the estimates of the posterior probability that the branch occurs in the true tree. The branch lengths are the posterior mean branch lengths described above. PAUP*, Mesquite, or TreeView can be used to display these trees.

    (b) algaeout.parts contains a table of splits and their associated posterior probabilities (and branch lengths). This file can be used to get probabilities for groupings that do not appear in the majority rule consensus tree.

    (c) algaeout.trprobs contains the sampled topologies. Redundant trees are excluded and the trees are sorted by their posterior probability. The trees are sorted so that the tree with the highest posterior probability is the first tree (the last tree has the lowest probability among the sampled trees – of course there are many unsampled trees that are not written to the file). After the tree name there will be a NEXUS comment such as `[p = 0.015, P = 0.922]`. `p` is the (estimate) of the posterior probability of this tree. `P` is the cumulative posterior probability that the true tree is **this tree or any one of the trees before it in the file**. Thus, `P` is simply the sum of all of the `p`'s before and including this tree. The tree's posterior probabilities are also stored as NEXUS tree weights.

11. To terminate the MrBayes program.

    ```
    quit
    ```

Exercise 2 (optional): Bayes Factors

This section demonstrates how to approximate the marginal likelihood of a model using the harmonic mean estimator. You may see this in the literature, but it is a terrible estimate of the marginal likelihood of the

[1] We ran the chain 50,000 steps, sampling every 50. Thus, there should be 50,000 / 50 = 1,000 trees saved in the tree file. Actually, there will be 1001 trees in the file because MrBayes saves the starting tree too. I said that burnin should stop at 2,000 steps, which translates to 40 sampled trees, so I am telling MrBayes to not include the starting tree and the next 40 trees in its summary.

[2] The answer to the first question is yes, and the posterior probability I obtained was 0.88 (your results may differ a little).

model. You almost certainly should be using the stepping stone method in MrBayes rather than the harmonic mean estimate. Depdending on how fast your laptop is, you may not be able to complete a stepping stone run during this exercise – by default it takes about 50 times as long as a normal MCMC run. If you do want to try to use the stepping stone method, you simply use the command:

```
ss
```

after you have executed the data file and configured the model and prior settings in MrBayes. If you don't want to run the stepping stone method, you should probably skip to Exercise 3. The steps in exercise 2 below here are mainly of interest if you are curious about how the harmonic mean estimator works.

1. Edit the algaemb.nex file (PAUP* provides a nice text editor), adding the following MrBayes block at the end of the file:

   ```
   begin MrBayes;
     set autoclose=yes;
     outgroup Anacystis_nidulans ;
     prset statefreqpr=fixed(equal);
     lset nst=1;
     mcmcp ngen=510000 samplefreq=100 printfreq=1000 nruns=1;
     mcmcp nchains=4 savebrlens=yes;
     mcmc file=jc;
     sump file=jc burnin=101 nruns=1;
   end;
   ```

   This sets up the Jukes-Cantor model in MrBayes. The `sump` command estimates the marginal likelihood of the JC model for this data set. You need to make sure that the `nruns` option of the `sump` command is set to the same number as the number of runs you requested in the `mcmc` command. The marginal likelihood of the model represents the average probability of the data using the model, where the average is a weighted average and the weights are provided by the prior distribution. Think of this as the average ability of the model to explain the data. Models that explain the data poorly over the parameter space defined by the prior have a lower marginal likelihood than models that explain the data better on average.

2. Execute this file in MrBayes, and write down the estimated marginal likelihood (harmonic mean), labeling it "log JC marginal likelihood". Type
   `quit` to terminate MrBayes.

3. Edit the algaemb.nex file again, this time replacing

   ```
   lset nst=1;
   ```

   with

   ```
   lset nst=2;
   ```

   and replacing

   ```
   mcmc file=jc;
   sump file=jc burnin=101 nruns=1;
   ```

   with

   ```
   mcmc file=k2p;
   sump file=k2p burnin=101 nruns=1;
   ```

   This time we will be using the K2P model, which differs from the JC model only in allowing transitions to occur at a different rate than transversions. The `nst=2` tells MrBayes that you want the rate matrix

to have two different rates (one for transitions and the other for transversions) rather than just one (`nst=1`)[3]. If the transition rate is indeed different than the transversion rate, then on average the K2P model should be able to explain the data better than the JC model, and the Bayes Factor in favor of the K2P model should be greater than 1. The `prset tratiopr` prior specifiication will now be used (this setting is simply ignored when `nst` is not set to 2).

4. Execute this file again in MrBayes, and again write down the estimated marginal likelihood (harmonic mean), labeling it "log K2P marginal likelihood". Type `quit`
to terminate MrBayes.

5. Compute the Bayes Factor in favor of the K2P model over the JC model as follows:

$$\text{BF} = \frac{\bar{L}_1}{\bar{L}_0} = e^{\ln \bar{L}_1 - \ln \bar{L}_0}$$

where $\ln \bar{L}_1$ is the log harmonic mean you wrote down for the K2P model, and $\ln \bar{L}_0$ is the log harmonic mean you wrote down for the JC model. In words, simply subtract the log harmonic mean for JC from the log harmonic mean for K2P. The Bayes Factor is the constant e raised to this value.[4]

What is the Bayes Factor for this comparison of models? Does the K2P model explain the data better than the JC model for this data set?[5]

6. If you have time and wish to explore Bayes Factors further, you can compare the K2P model to the HKY model. This involves simply replacing the

```
prset statefreqpr=fixed(equal)
```

with

```
prset statefreqpr=dirichlet(1.0, 1.0, 1.0, 1.0)
```

Rather than fixing the base frequencies all at 0.25, they will now be allowed to vary during the MCMC run, which effectively switches the model from K2P to HKY. I found that the difference in log marginal likelihoods was about 6.34 for this comparison (with HKY on top), and thus the Bayes Factor in favor of HKY over K2P is about 567. Much less impressive than the previous comparison, but the fact that HKY explains the data more than 500 times better than K2P means that the base frequencies really aren't equal and a model (such as HKY) that allows unequal base frequencies is preferable.

Exercise 3: Marginal Distributions

Exercise 3 shows you how to examine the posterior distributions for parameters in Excel and with the `sumt` command. You may prefer to just use Tracer for this (and skip to Exercise 4)

One of the benefits of the Bayesian approach is the ability to obtain marginal distributions of just about any quantity of interest. For example the marginal distribution of the transition/transversion rate ratio, κ, is the distribution of this parameter with all other parameters in the model integrated out.

[3] "nst" stands for number of substitution types.

[4] Unfortunately this simple harmonic mean estimator for the marginal likelihood is a poor estimator – it has high variance. Despite the fact that these estimates of the marginal likelihoods are not necessarily very reliable, they have been widely used. See (Lartillot and Philippe, 2006) and the Stepping-stone method of Paul Lewis and collaborators for more reliable methods of estimating the marginal likelihood. The version 3.2 of MrBayes (Ronquist et al., 2012) supports stepping-stone sampling. The manual pdf that comes with MrBayes describes how to conduct a stepping-stone sampling analysis using the `ss` command in MrBayes.

[5] The difference in log marginal likelihoods was 80.68, making the Bayes Factor equal to $e^{80.68} = 1.09 \times 10^{35}$, which is an impressive number! This indicates that the K2P model does indeed explain the data much better than the JC model.

If we pretend that the field the MCRobot explores is a bivariate parameter space, with κ on one axis and the evolutionary distance v on the other, then the joint probability density of these two parameters would be represented by a single bivariate normal hill in the field. The marginal distribution of κ would be a univariate normal distribution, and can be visualized by imagining that the "hill" is made up by many tiny balls inside a box, and marginalization involves tilting the field so that the marbles pile up against one side of the box.

The data in jc.nex was simulated using a JC69 model, so we know in this case that the true value of κ is 1.0. How certain of this could we be, however, if we didn't know the truth but only had these data to work with.? What is the 95% credible region for κ given this dataset? This is the type of question that can be answered easily using MrBayes.

1. Create a MrBayes block at the bottom of the jc.nex file so that the Kimura 2-parameter (1980) model is specified.

```
begin MrBayes;
  set autoclose=yes;
  lset nst=2;
  prset statefreqpr=fixed(equal) tratiopr=beta(1,1);
  mcmcp ngen=110000 samplefreq=100 printfreq=1000 nruns=1;
  mcmc;
  sump burnin=101 nruns=1;
end;
```

2. Execute this file and at the end of the output you should find a parameter summary that shows the posterior mean, variance and 95% credible intervals for both tree length (TL) and κ (kappa).

   Does the 95% credible region for κ include 1.0? Is this result consistent with the fact that JC was the generating model?[6]

3. Read the jc.nex.p file into Excel if Excel is available. For the 95% credible region, we need to find the 2.5 and 97.5 percentiles. Assuming D is the column containing the κ values, the following formulas can be used to obtain, respectively, the number of sampled values, the posterior mean of kappa, the 2.5 percentile and the 97.5 percentile. The formulas assume that you have copied the entire contents of the jc.nex file and pasted it into the upper left corner of an Excel worksheet. The cell D104 thus corresponds to generation 10100, the first cell considered if the top 101 samples are discared as burnin.

```
=COUNT(D104:D1103)
=AVERAGE(D104:D1103)
=PERCENTILE(D104:D1103,0.025)
=PERCENTILE(D104:D1103,0.975)
```

   Do these values agree with those computed by MrBayes?[7]

4. If Tracer (http://tree.bio.ed.ac.uk/software/tracer) is available, exploring marginal distributions of parameters is easy. Just choose File > Import... from the main menu of Tracer, then navigate

[6] Yes, the credible region includes 1.0 (the interval I obtained stretched from 0.906 to 1.149), which is consistent with JC being the model used to generate the data.

[7] If not, be sure you did not include the first 101 values, which were discarded as burnin when the `sump` command was run.

to the jc.nex.p file, select it and click the Open button. Then, click on kappa in the "Traces" panel on the lower left. (If you do not see the work kappa anywhere on the left, it may be that the "Traces" panel is off the screen on the bottom. In this case, look for a horizontal bar on the lower left; drag this bar upward to reveal the Traces panel.)

5. If Tracer is not available, you can create a histogram of column D (the column labeled kappa) in Excel. This histogram represents the marginal distribution of kappa. In Excel, you must first create a column of bin boundaries. I used the series of 40 values 0.05, 0.1, 0.15, ..., 2.0 for this purpose. Then, from the main menu choose Tools > Data Analysis > Histogram and then specify `$D$104:$D$1103` for the input range (again, assuming κ is in column D) and `$J$1:$J$40` for the bin range (assuming the 40 bin boundaries are in column J).

Exercise 4: Stopping rules

MCMC algorithms can produce arbitrarily precise estimates of the distributions that they sample from **if they are run long enough**. Assessing when an MCMC simulation has been run long enough can be very difficult. There are wide variety of tools that can detect failure to converge (e.g. the Google search for "convergence diagnostics" + MCMC returned 17,200 hits for me in June 2006). An inherent weakness in these diagnostics is that they look at where the chain(s) has/have been. If a whole region of high posterior probability has been missed in every run, this error will not be detected. Unfortunately, MCMC errors of this type usually lead us to be too confident in our results (we will underestimate the variance because we have not sampled the landscape thoroughly).

The general defense against inflated confidence due to MCMC error is to run several independent MCMC analyses and compare the results. In particular, if the starting points for different runs are chosen such that they are spread out more than by random ("over-dispersed") then different runs are expected to yield different answers as long as they have been run for an insufficient number of iterations. Given that the runs are all sampling the same posterior distribution, they will all agree *eventually*. Thus, a general strategy for generating reliable results from MCMC analyses is to perform multiple runs making them all long enough for them to agree with each other.

Fortunately, versions of MrBayes after version 3 include an automatic stopping rule. Instead of specifying a fixed number of generations and then assessing whether or not the analysis was sufficiently long, we can ask MrBayes to run the chain until a condition has been met. The convergence diagnostic implemented in MrBayes 3.1.2 is the **average standard deviation in split frequencies**. The standard deviation of the frequency of a particular split (or "clade" or "grouping") across all runs is calculated. These standard deviations are then averaged over all splits which had a frequency (in at least one run) above a user-specified cutoff frequency. As the runs proceed, they should converge to similar split frequencies for all of the splits in the tree. When the average standard deviation in split frequencies drops below the `stopval`, then MrBayes will terminate the MCMC runs.

1. Add the command to use automatic stopping rule to the last version of the algaemb.nex. Replace:

```
mcmcp ngen=510000 samplefreq=100 printfreq=1000 nruns=1;
```

with

```
mcmcp samplefreq=100 printfreq=1000;
mcmcp nruns=2 stoprule=YES burninfrac=.25 ;
mcmcp stopval=0.01 minpartfreq=0.05;
```

```
mcmcp mcmcdiagn=YES diagnfreq=100;
```

These new options mean:

- `nruns=2` – the automatic stopping rule needs more than one run so that it can compare split frequencies between runs, so 2 is the minimum we can get away with.
- `stoprule=YES` – *Guess what this one means.*[8]
- `burninfrac=.25` – every time MrBayes calculates convergence diagnostics, it will discard the first 25% of the run. Alternatively you can specify `burnin=1000` to specify a constant burnin length.
- `stopval=0.01` – Stop the runs when the average standard deviation of split frequencies (across different runs) is ≤ 0.01.
- `minpartfreq=0.05` – Determines the (lower) cutoff for split frequencies included when calculating the average standard deviation of split frequencies.
- `mcmcdiagn=YES` – Write MCMC diagnostics in <prefix>.mcmc
- `diagnfreq=100` – Calculate the MCMC diagnostics every 100 iterations.

2. Redirect the output to a new set of files by replacing:
```
mcmc file=k2p;
```
with
```
mcmc file=autoStop;
```

3. Finally comment out the `sump` command by replacing:
```
sump burnin=101 nruns=1;
```
with
```
[sump burnin=101 nruns=2;]
```

4. Execute the file with the `Execute` command.
How does the progress reported by MrBayes during the run look different from previous runs? What is the trend in the average standard deviation of split frequencies during the run?[9]

5. Unfortunately, if we are using `burninfrac` we do not know the number of samples to discard from the beginning of the run, so we cannot put the `sump` in our MrBayes block. After the chain has run, look at one of the `.t` or `.p` files and see how many samples were taken. Multiply this number by the `burninfrac`. This is your burnin

6. Comment out the MrBayes block in algaemb.nex by surrounding it in [and] or by changing the name of the block so that it is skipped (e.g. change it to `xMrBayes`).

7. Launch MrBayes, execute algaemb.nex and then type:
`sumt file=autoStop nruns=2 burnin=X`
where `X` is the burnin that you calculated above. This should give you results that are similar to those you obtained in earlier when we used a (somewhat arbitrary) number of generations.

[8] Yup, you're right.

[9] There are now lines like "Average standard deviation of split frequencies: 0.010775" reported every 100 iterations. The average standard deviation of split frequencies may bounce around, but the trend should be to decrease. The run will terminate when the diagnostic falls below the stopval (0.01)

8. If you have time, you can repeat the run with the automatic stopping rule. You will probably get a different chain length, but the results should be compatible.

The web site http://ceb.csit.fsu.edu/awty is a useful tool for producing graphics that help you assess convergence. Due to bandwidth concerns, we may not all be able to use the site at the same time, but I encourage you to check it out on your own time.

Exercise 5: Partitioned models

MrBayes is extremely flexible in terms of the number of models that it supports. It allows you to partition the data into subsets and apply different models to different subsets of data. Beyond that you can even force some model parameters to be shared across the different model. For instance you could allow two different subsets to have different rate heterogeneity parameters, but force them to share base frequency parameters.

This will be a very brief introduction to how to set up partitioned model analyses in MrBayes. For simplicity sake, we'll use the same algaemb.nex that we used above. This is a data set is a group of ribosomal RNA sequences. Unfortunately, I do not know which sites code for stem and which code for loops (if we did we could use MrBayes models for coevolution of nucleotides that pair in stems in the rRNA secondary structures). Instead we'll use an arbitrary partitioning scheme. This will lead to an unsatisfying example, but focus on the commands that we use – you can apply these commands to your own data with more sensible partitions.

1. Relaunch MrBayes and `Execute bglobin.nex`

2. The command to define a partition of the data is very similar to the `CharPartition` used in PAUP*. The general form is `Partition <partition name> = <number of subsets> : <subset definition> , <subset definition> ;` Where everything in the `<>` is replaced with an appropriate value. Unlike the commands we used with PAUP*, the subsets are not named, and we have to tell the program "up front" how many subset there are. Open `bglobin.nex`
   in a text editor. You should see a MrBayes block with the following content:

   ```
   charset non_coding = 1-90 358-432 ;

   charset first_pos  = 91 - 357 \ 3 ;

   charset second_pos = 92 - 357 \ 3 ;

   charset third_pos  = 93 - 357 \ 3 ;

   partition region = 4:non_coding,first_pos,second_pos,third_pos;

   set partition = region;
   ```

   This sets up 4 subsets. The syntax

   ```
   91 - 357 \ 3
   ```

means "every third character from 91 up to 357." So

```
91 - 357 \ 3
```

means characters $91, 94, 97, 100, \ldots$. This partitioning command would be the one you would use if you wanted to use different models for different coding positions (and the entire data matrix was an in-frame protein coding gene sequences).

3. To tell MrBayes that it needs to use this partition in the next analysis, we use the command:
   `set partition = bc`

4. I find that the key to setting up partitioned models correctly is to verify after every step that you and MrBayes agree about the model that you have set. Fortunately, you can enter:
   `ShowModel`
   to see a report from MrBayes about what models are currently in effect. The report ends with a section that lists all of the parameters of any active model. It shows what subsets (referred to "Partitions" in the report) use that parameter.

5. MrBayes uses the `LSet` and `PrSet` to adjust the model itself (what parameters are used) and the prior distributions for each of these parameters. Because there are multiple models in effect, we have to qualify these commands with an `applyto` command parameter that lists which models the command is intended for. The `applyto` uses the number of the data partition as the identifier of the model. So if you enter:

   ```
   LSet ApplyTo=(1) nst= 6;
   ShowModel;
   ```

   The first partition is now assigned a 6 substitution type model (the GTR model). Make sure to put the `applyto` option *before* the other options in the `LSet` and `PrSet` commands.

6. If you examine the report from the previous `ShowModel` command you will see that we have MrBayes configured to use 3 different models, but partitions 2 and 3 are sharing all of their parameters. In MrBayes terminology, these parameters are linked. We can use the `Unlink` command to give the different partitions their own parameters to estimate. This means that we will actually be using different models for each subset. To make MrBayes infer separate equilibrium state frequencies for each subset use the command:

   ```
   Unlink StateFreq=(all);
   ShowModel;
   ```

7. There are several other model changing commands shown in a comment in the data file. Particularly important is the `prset ratepr = Dirichlet(3, 1, 1, 3)` which says that the different subsets of the data can have different rates of evolution. In our example of codon positions, it is very likely that the different codon positions have different mean rates, so we certainly want to let the software explore models where these mean rates vary. In this we think that the non-coding region and the third-base positions are likely to evolve at a higher rate, so we are putting a prior on the rate multiplier for each subset such that the 1st and 4th subset are expected to be evolving three times faster than the other two subsets.

8. Note that these model conditions may be overparameterized – I just set up this model to demonstrate the syntax needed to configure MrBayes

9. Now you can use the `MCMC`, `SumT`, and `SumP` commands to perform the MCMC approximation, and summarize the results.

References

Lartillot, N. and Philippe, H. (2006). Computing Bayes factors using thermodynamic integration. *Systematic Biology*, 55(2):195–207.

Ronquist, F., Teslenko, M., van der Mark, P., Ayres, D. L., Darling, A., Höhna, S., Larget, B., Liu, L., Suchard, M. A., and Huelsenbeck, J. P. (2012). Mrbayes 3.2: Efficient bayesian phylogenetic inference and model choice across a large model space. *Systematic Biology*, page *in press*.

Ronquist, F. R. and Huelsenbeck, J. P. (2003). Mrbayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19(12):1574–1575.

Using phylogenetics to estimate species divergence times ...

More accurately ...

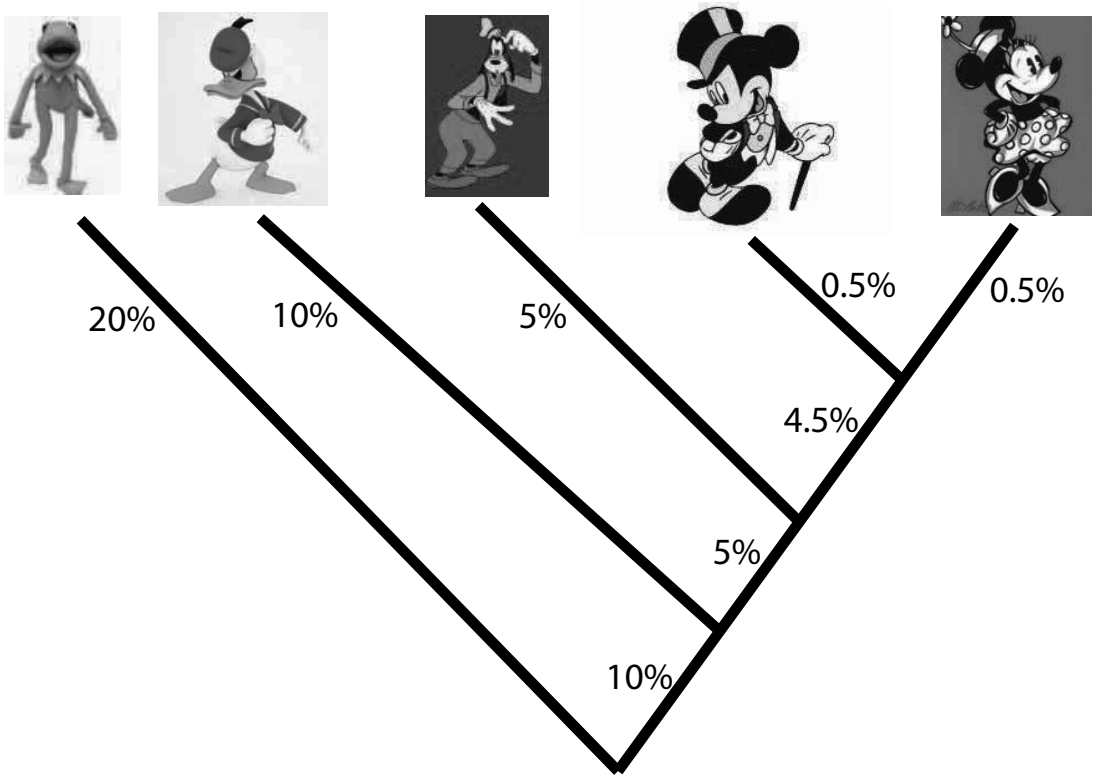
Basics and basic issues for Bayesian inference of divergence times (plus some digression)

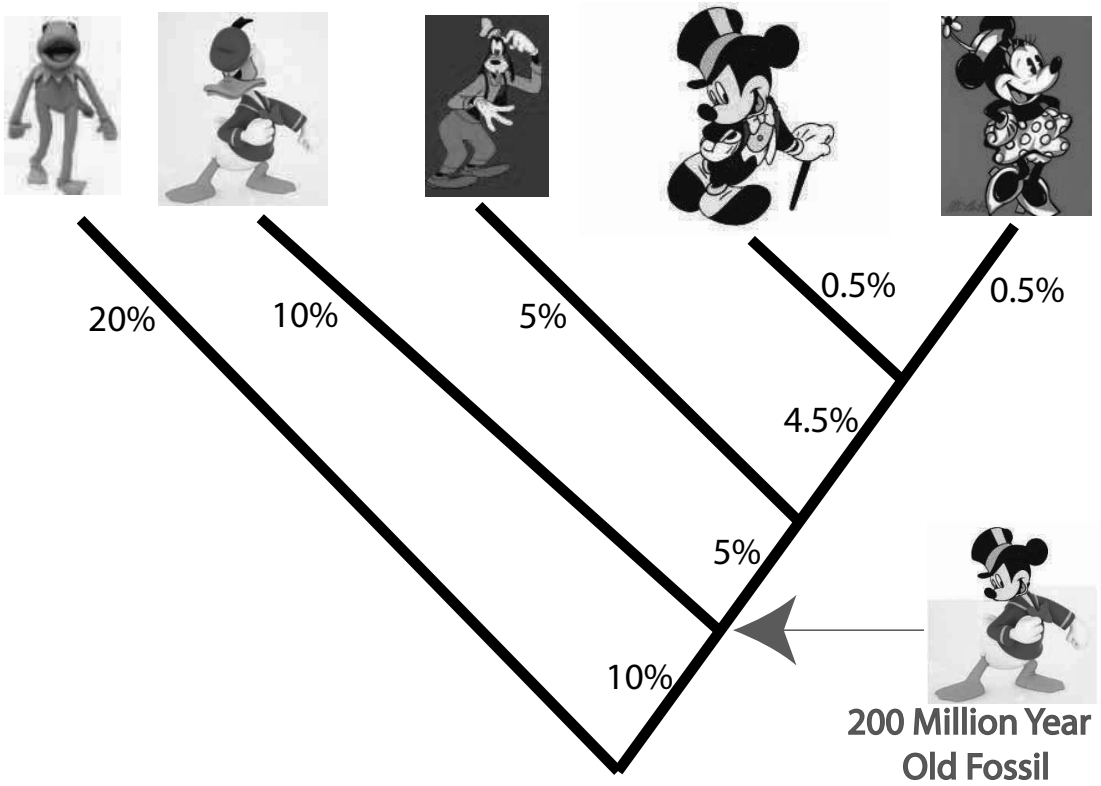
"*A comparison of the structures of homologous proteins ... from different species is important, therefore, for two reasons. First, the similarities found give a measure of the minimum structure for biological function. Second, the differences found may give us important clues to the rate at which successful mutations have occurred throughout evolutionary time and may also serve as an additional basis for establishing phylogenetic relationships.*"

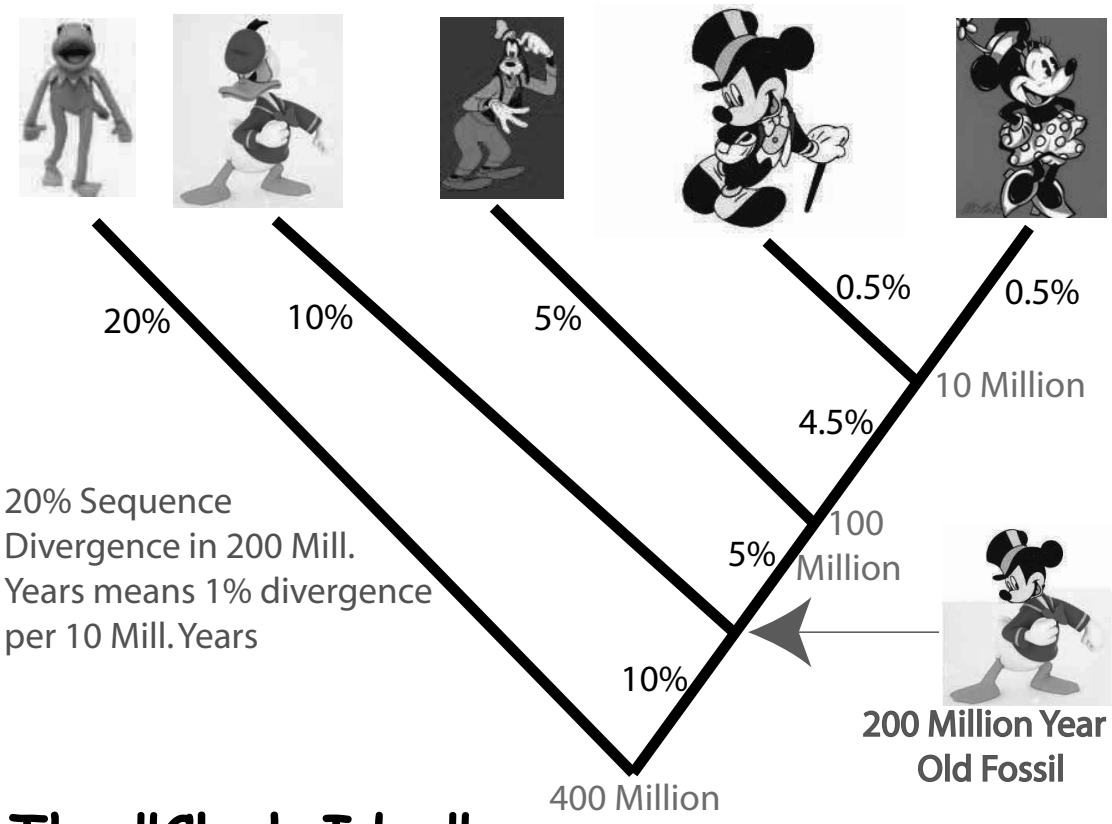
From p. 143 of

The Molecular Basis of Evolution

by Dr. Christian B. Anfinsen (Wiley, 1959)


20%
10%
5%
0.5%
0.5%
4.5%
5%
10%

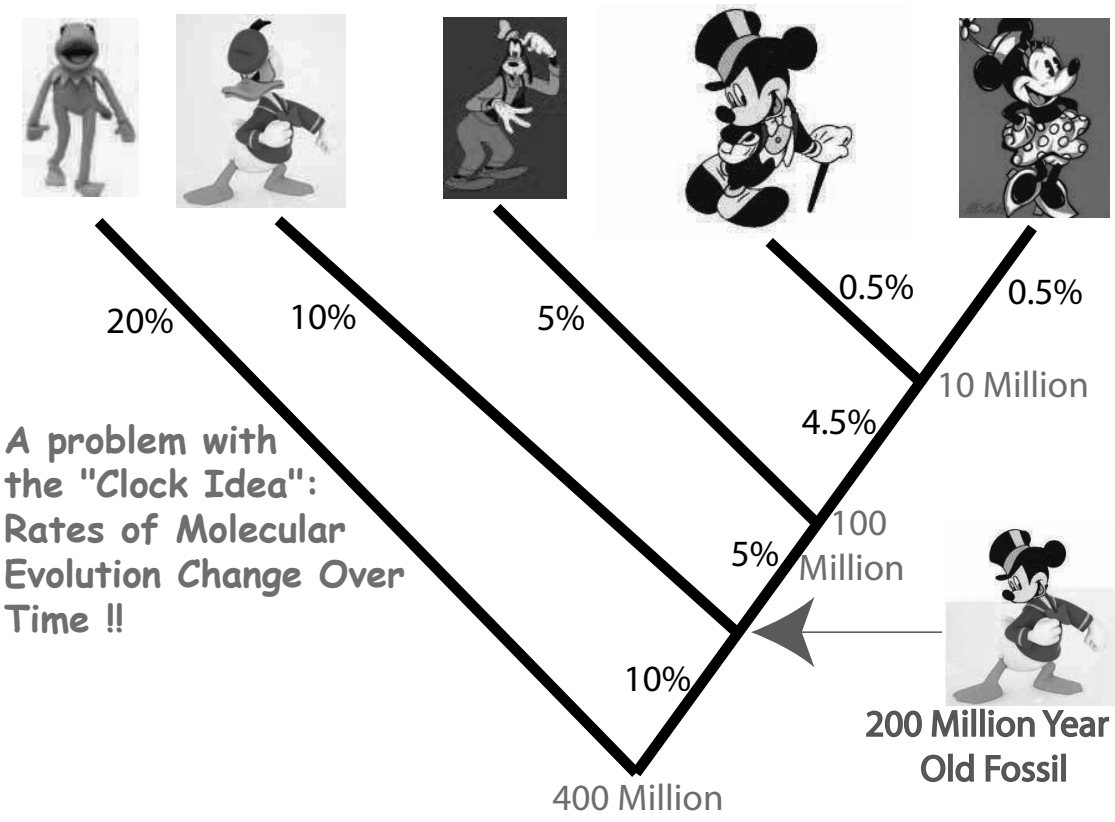

20%
10%
5%
0.5%
0.5%
4.5%
5%
10%
200 Million Year
Old Fossil



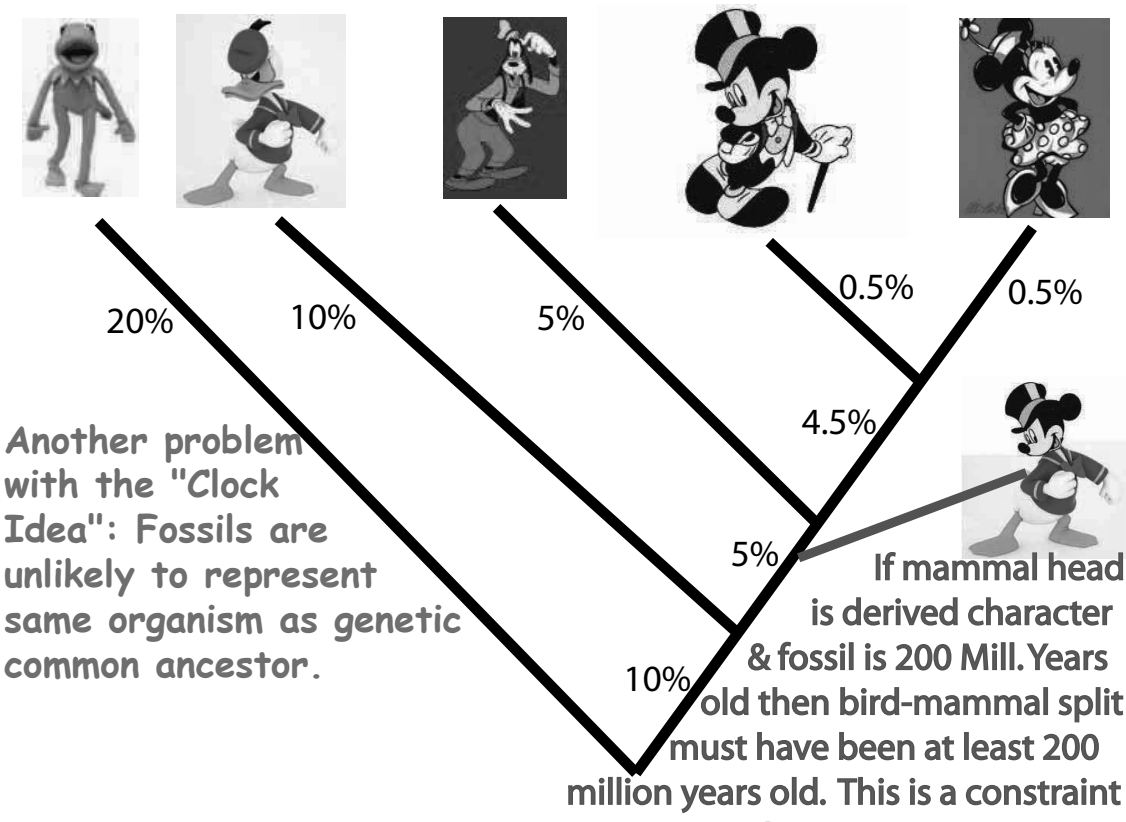
The "Clock Idea"

"Ernst Mayr recalled at this meeting that there are two distinct aspects to phylogeny: the splitting of lines, and what happens to the lines subsequently by divergence. He emphasized that, after splitting, the resulting lines may evolve at very different rates... How can one then expect a given type of protein to display constant rates of evolutionary modification along different lines of descent?"

(Evolving Genes and Proteins. Zuckerkandl and Pauling, 1965, p. 138).



A problem with the "Clock Idea": Rates of Molecular Evolution Change Over Time !!



Another problem with the "Clock Idea": Fossils are unlikely to represent same organism as genetic common ancestor.

If mammal head is derived character & fossil is 200 Mill. Years old then bird-mammal split must have been at least 200 million years old. This is a constraint on a divergence time.

Bayesian Idea:

(Prior Information)

X

(Information from data)

= Posterior Information

Basic Idea for Bayesian Divergence Time Inference

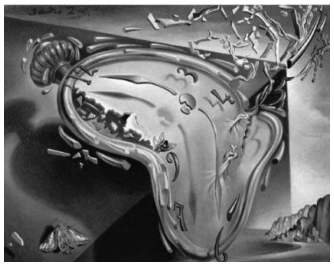
R: rates
T: node times
C: Fossil Evidence (constraints)
S: Sequence Data

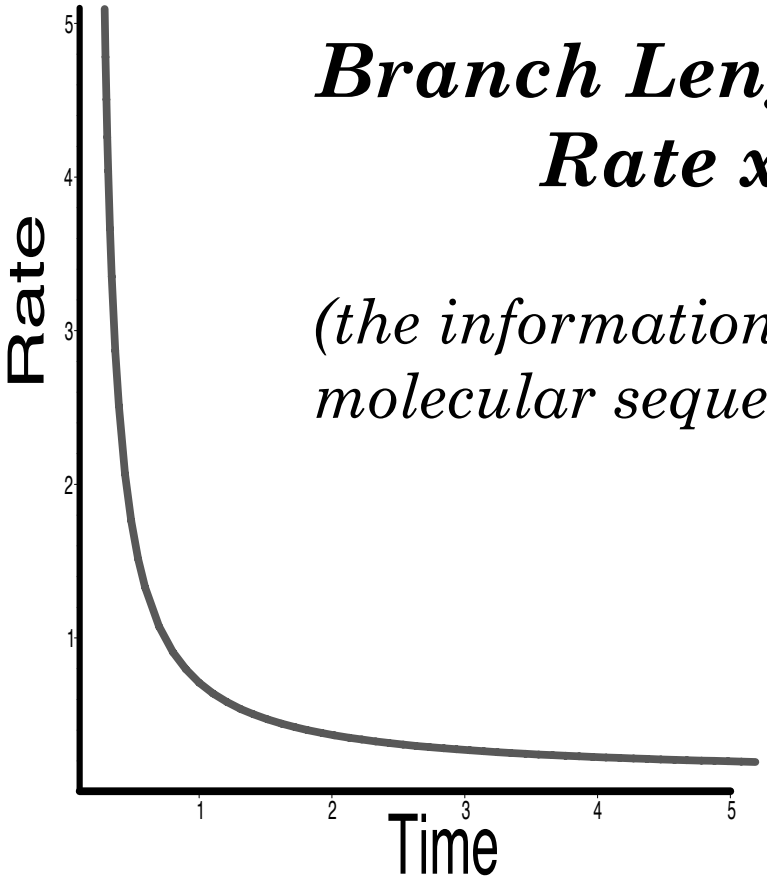
$$P(R,T|S,C) = \frac{P(S,R,T|C)}{P(S|C)} = \frac{P(S|R,T,C)\,P(R|T,C)\,P(T|C)}{P(S|C)}$$

$$= \frac{P(S|R,T)\,P(R|T)\,P(T|C)}{P(S|C)}$$

(Relaxed Clock) Bayesian Divergence Time Components

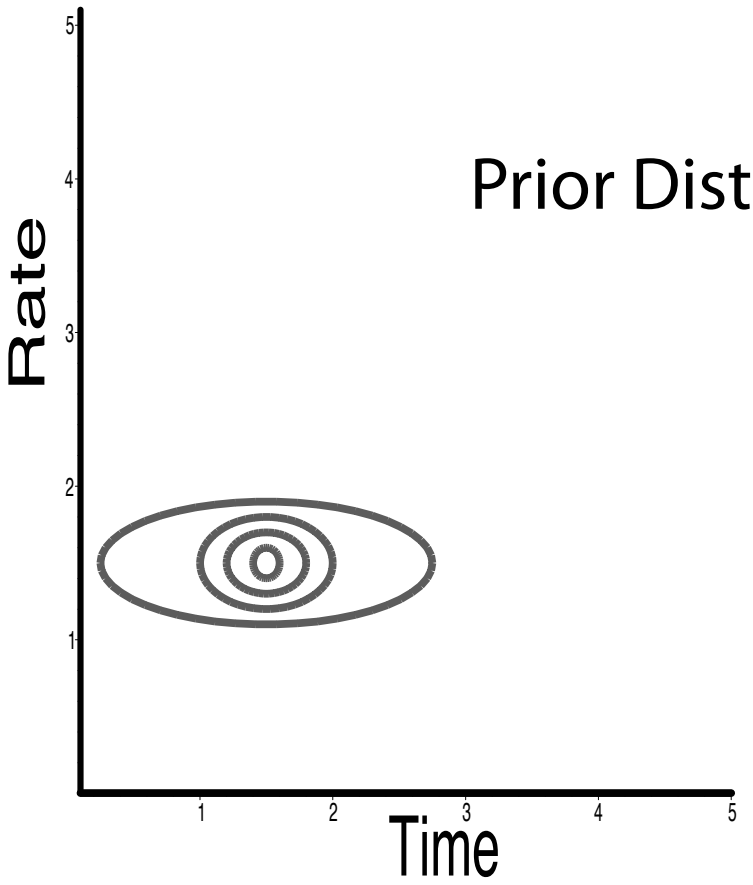
1. DNA or protein sequence data
2. Model of Sequence Change
3. Model of Rate Change
4. Prior Distributions for Rates, Times, etc.
5. Fossil or other information

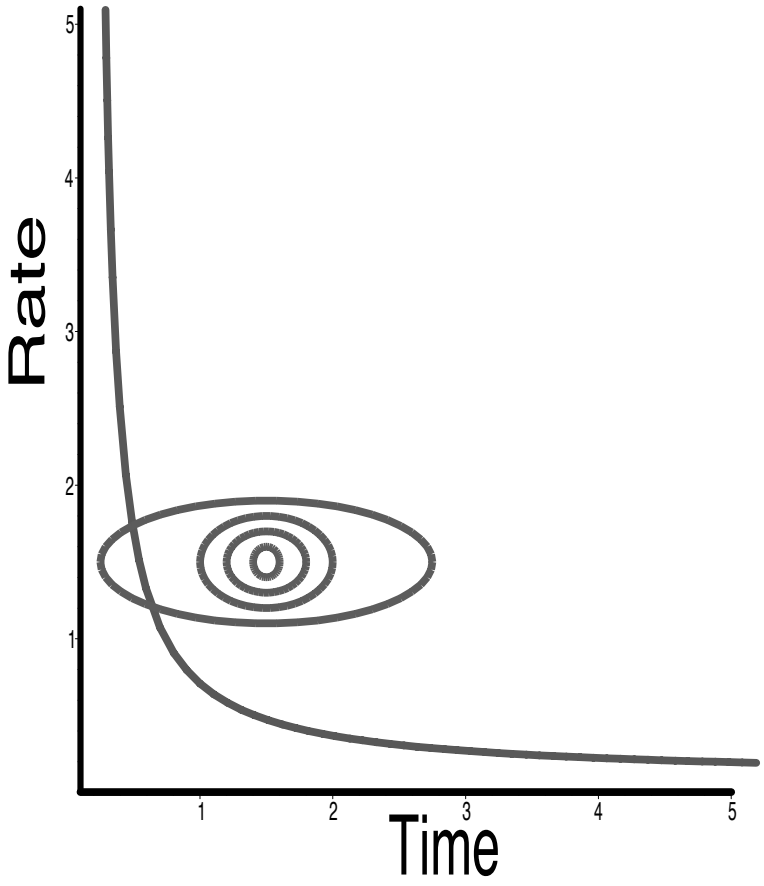


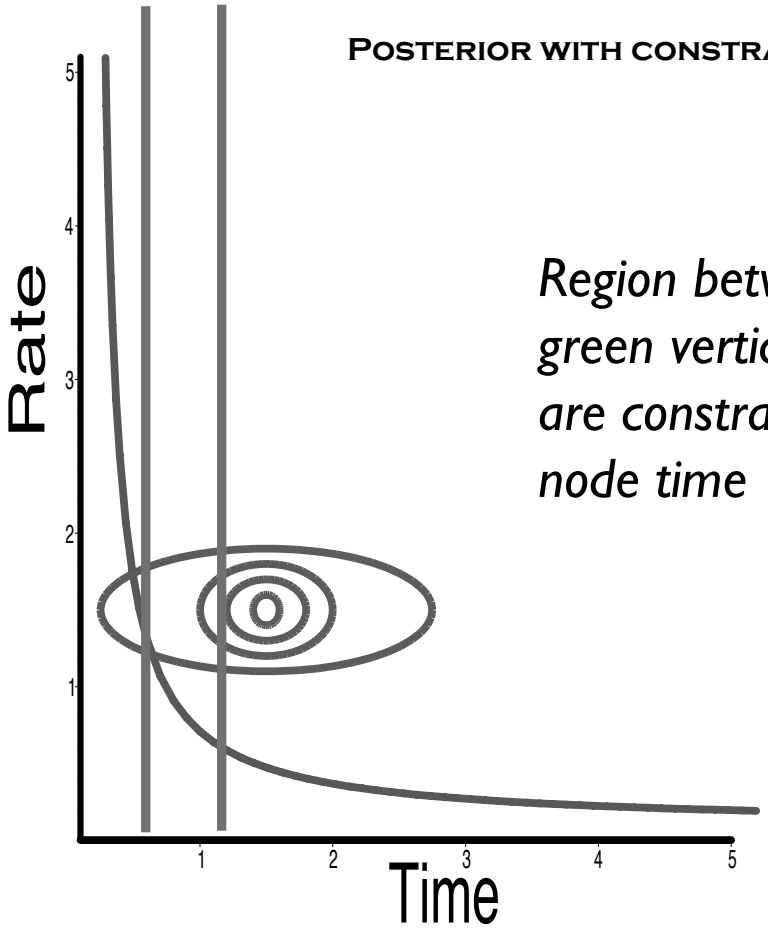


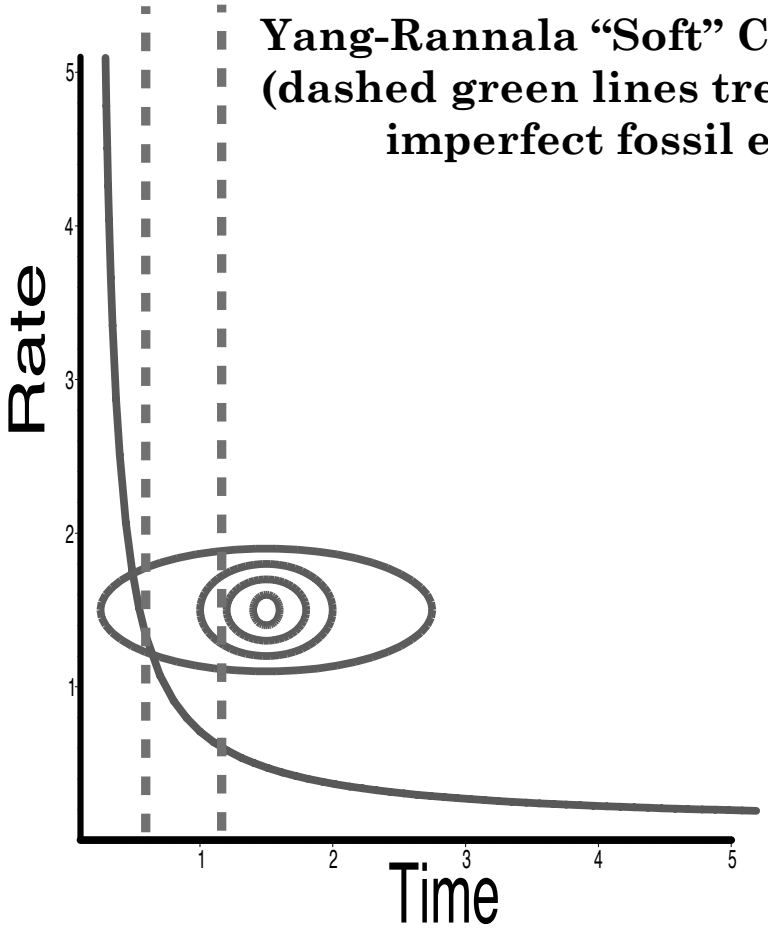
Branch Length = Rate x Time

(the information from molecular sequence data)


Prior Distribution
Rate
Time
1
2
3
4
5


Rate
Time
1
2
3
4
5


Posterior with constraints
Region between green vertical lines are constraints on node time
Rate
Time
1
2
3
4
5


Yang-Rannala "Soft" Constraints (dashed green lines treated as imperfect fossil evidence)
Rate
Time
1
2
3
4
5

Bayesian Divergence Time Components

1. DNA or protein sequence data

Sequence data is needed for branch length (rate x time) estimation.

Sequence data does not separate rates and times.

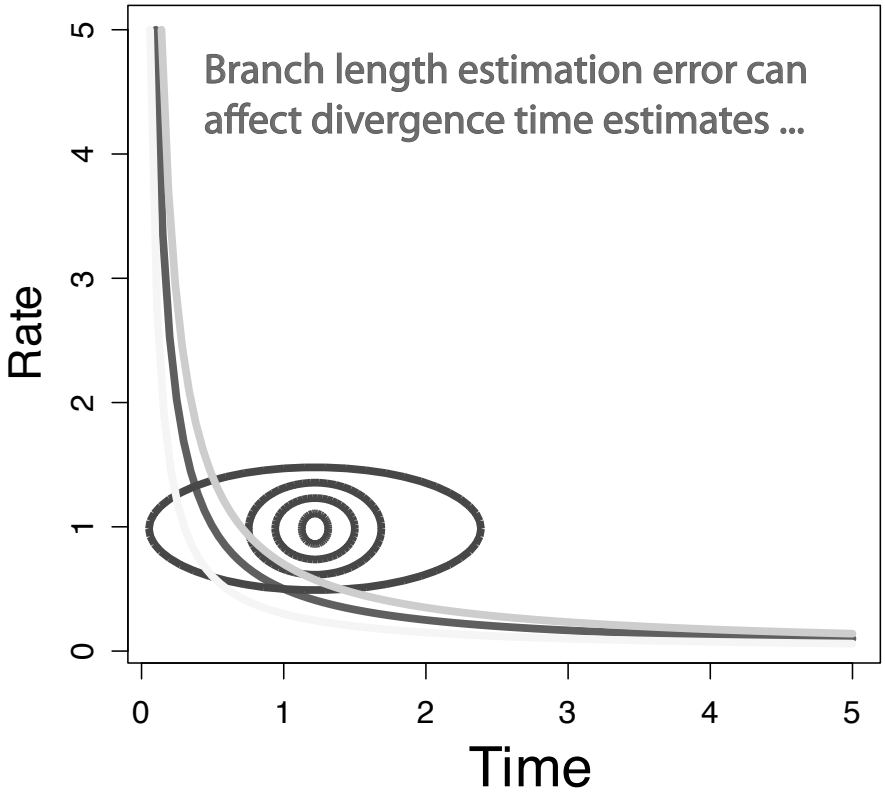
Better to invest in improving other time estimation components?

Bayesian Divergence Time Components

2. Model of Sequence Change

Branch Length (BL) Errors → **Divergence Time Errors**

Posterior distributions for times are compromise between branch length information from sequence data and prior information and fossil information.



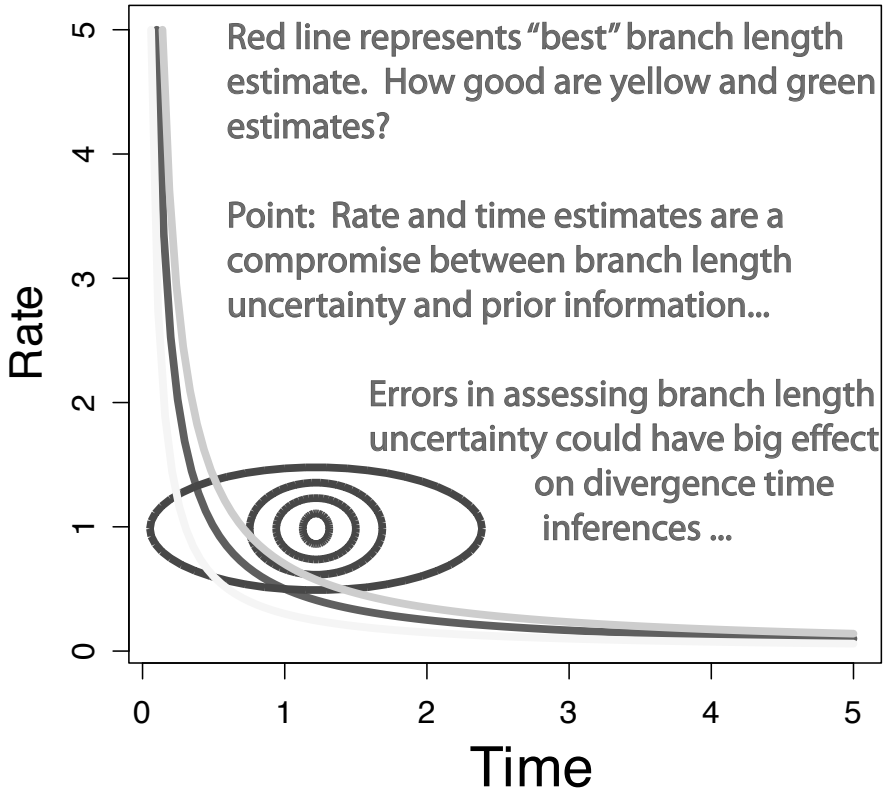
Bayesian Divergence Time Components

2. Model of Sequence Change

Branch Length (BL) Errors → **Divergence Time Errors**

Errors in BL uncertainty → **Divergence Time Errors**

Posterior distributions for times are compromise between branch length information from sequence data and prior information and fossil information.



Errors in BL uncertainty have more serious consequences for divergence time estimation than for phylogeny inference.

Sources of these errors include failure to account for dependent change among sequence positions.

Context-Dependent Mutation
Codons
Protein Tertiary Structure
RNA Secondary Structure
Other Genotype-Phenotype Connections

Bayesian Divergence Time Components

3. Model of Rate Change

How much of what appears to be rate change really is rate change?

see

Cutler, D.J. (2000) Estimating divergence times in the presence of an overdispersed molecular clock. Mol. Biol. Evol. 17:1647-1660.

A point made well by Cutler (2000)

..Rejection of constant rate hypothesis may not be due to variation of rates over time as much as being due to poor models of sequence evolution that may mislead us about how confident we can be regarding branch length estimates ...

(my viewpoint... "first principles" of evolutionary biology mean constant rate hypothesis must be formally wrong even though it may sometimes be nearly right)

Why might rates of molecular evolution change over time? Candidates include changes in ...

mutation rate per generation

generation time

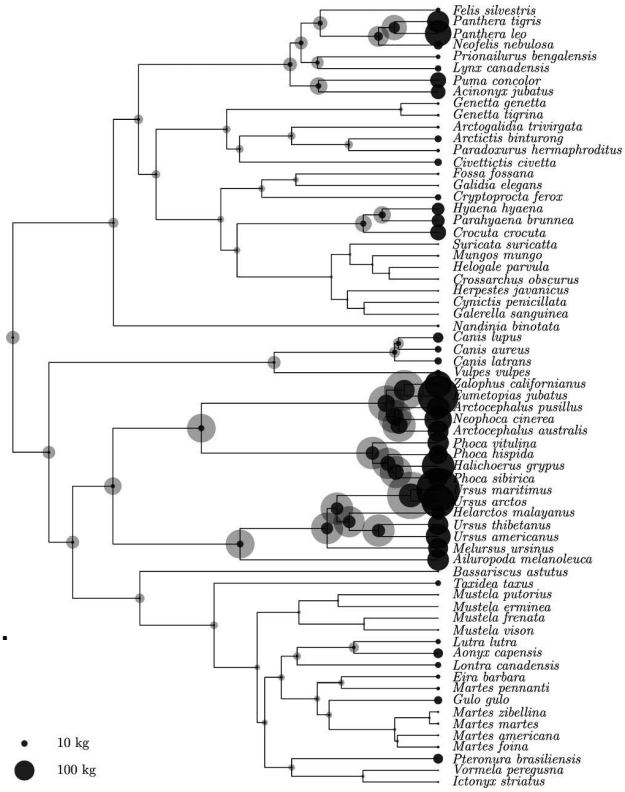
natural selection (including effects due to duplication)

population size (higher rates for small pop. size)

MODELING RATE VARIATION AMONG LINEAGES

- Global molecular clock (Zuckerkandl & Pauling, 1962)
- Local molecular clocks (Hasegawa, Kishino & Yano 1989; Kishino & Hasegawa 1990; Yoder & Yang 2000; Yang & Yoder 2003, Drummond & Suchard 2010)
- **Autocorrelated Rate Change (Huelsenbeck, Larget & Swofford 2000; Thorne, Kishino, & Painter 1998; Kishino, Thorne & Bruno 2001; LePage, Bryant, Philippe, & Lartillot 2007)**
- Uncorrelated/independent rates models (Drummond et al. 2006; Rannala & Yang 2007)
- Mixture models on branch rates (Heath, Holder, & Huelsenbeck 2012)

A promising idea:
By allowing them to evolve along with substitution rates, phenotypic characters that may be correlated with substitution rates can be leveraged to improved divergence time estimates



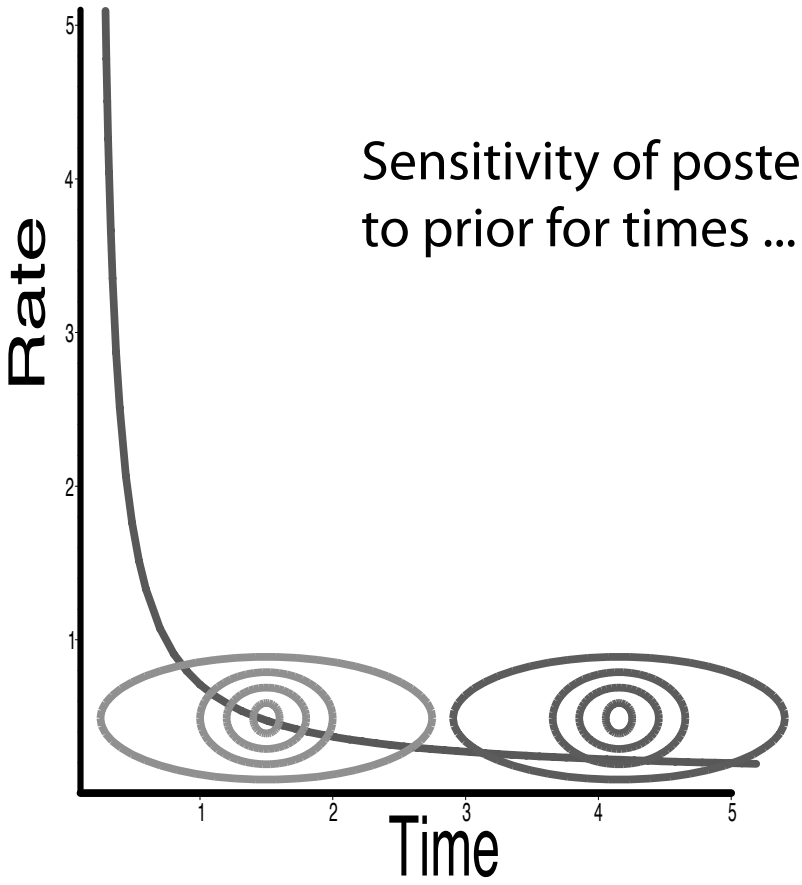
From: Lartillot N , Poujol R. 2011. Reconstruction of the evolution of body mass in carnivores. Mol Biol Evol 28:729-744

Bayesian Divergence Time Components

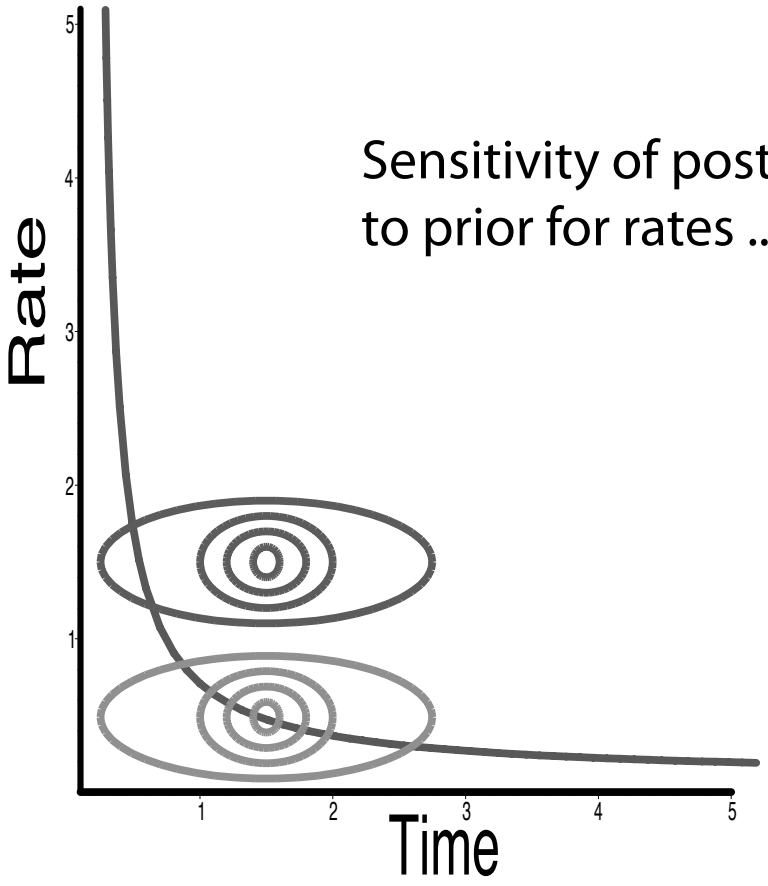
4. Prior Distributions for Rates, Times, etc.

Difficulty in specifying appropriate prior distributions is arguably the biggest obstacle for Bayesian inference and this difficulty is especially great for divergence time estimation.

In many situations, prior distribution is not too important if data set is large. However, large amounts of sequence data do not overcome need for good rate and time priors here ...

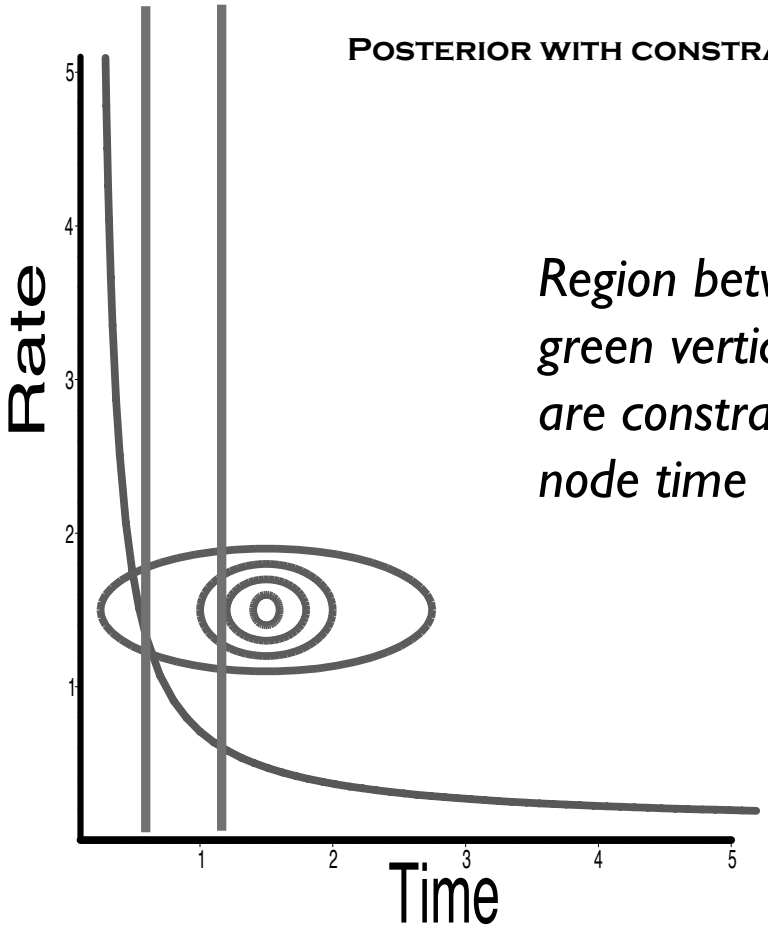


Sensitivity of posterior to prior for times ...

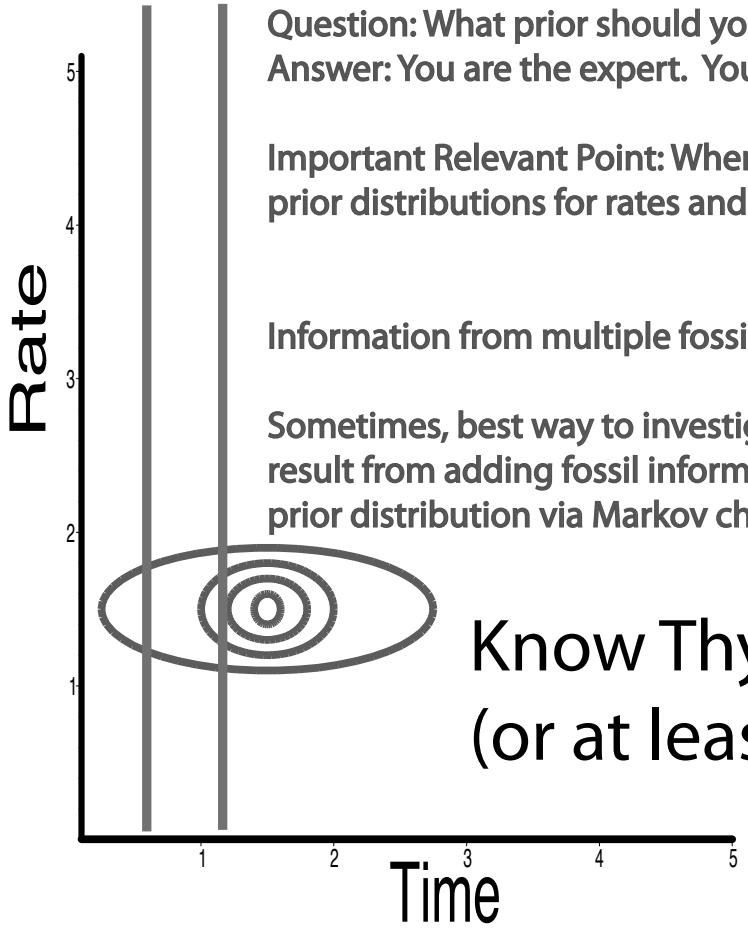


Sensitivity of posterior to prior for rates ...

Posterior with constraints



Region between green vertical lines are constraints on node time



Question: What prior should you use?
Answer: You are the expert. You decide.

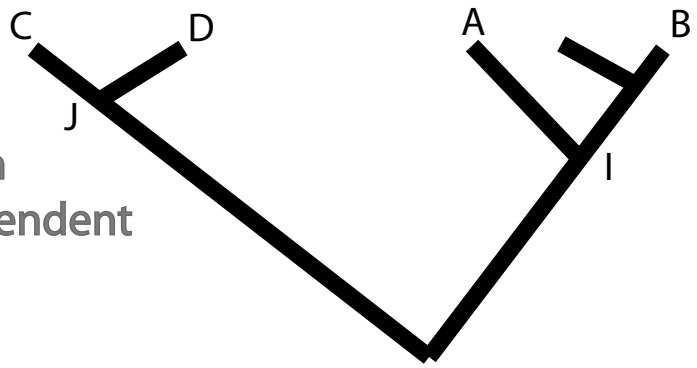
Important Relevant Point: When adding fossil information, prior distributions for rates and times can be complicated.

Information from multiple fossils can interact !

Sometimes, best way to investigate prior distributions that result from adding fossil information is to approximate prior distribution via Markov chain Monte Carlo.

Know Thy Prior!
(or at least learn it!)

Branch length between Nodes A & I and between Nodes B & I should be correlated even if rates on these branches are independent of each other.



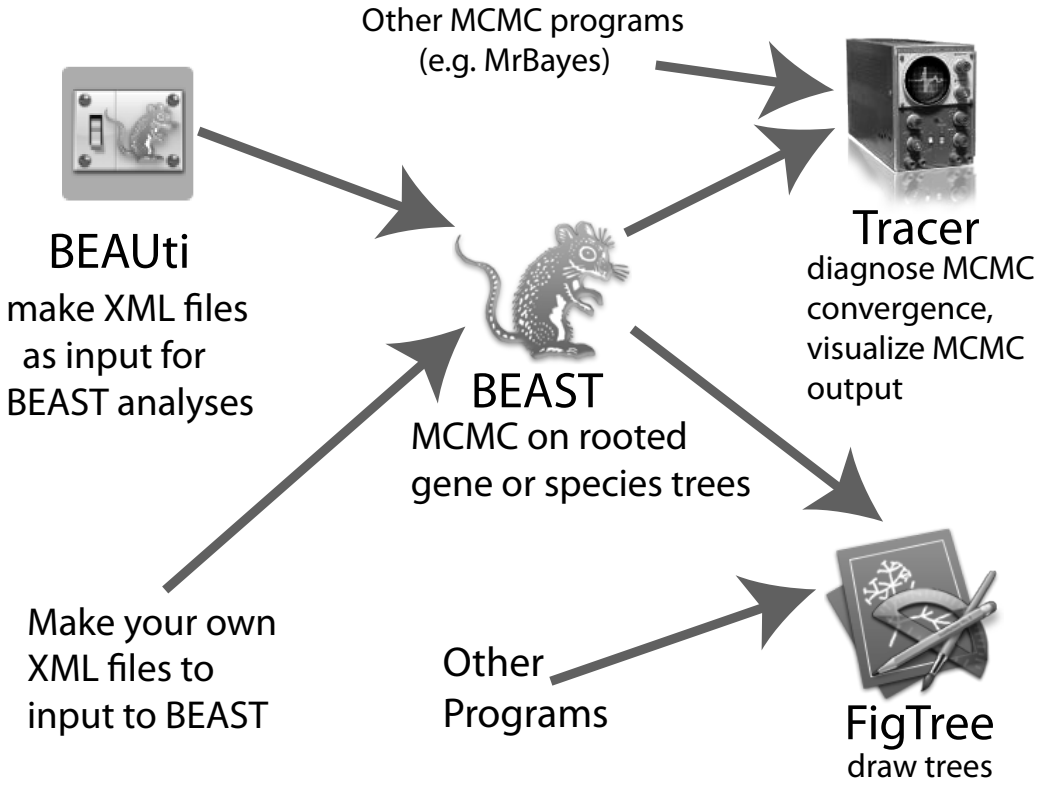
Reason: These branches represent the same amount of time.

A nice paper ...

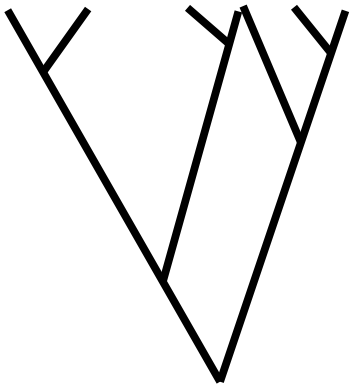
Drummond, Ho, Phillips, and Rambaut. 2006. Relaxed Phylogenetics and Dating With Confidence. PLOS Biology 4(5):e88 (see also their BEAST software)

(i) Divergence time estimation without prespecified topology
(ii) Phylogeny inference incorporating models of rate evolution

BEAST & relatives (see http://tree.bio.ed.ac.uk/software/)



Priors on node times
(and sometimes on rooted topologies):



(1) Phenomenological: Choose a hopefully flexible probability distribution (e.g., put a prior distribution on the root age and put a prior on the proportional ages of all other internal nodes relative to root age)

(2) Mechanistic: Invoke some biology to justify the prior

Yule Process (Birth process): Only speciation considered

Birth-Death Process: Speciation and Extinction considered

Taxon Sampling can also be considered (i.e., how does one decide which extant species to include in data set?)

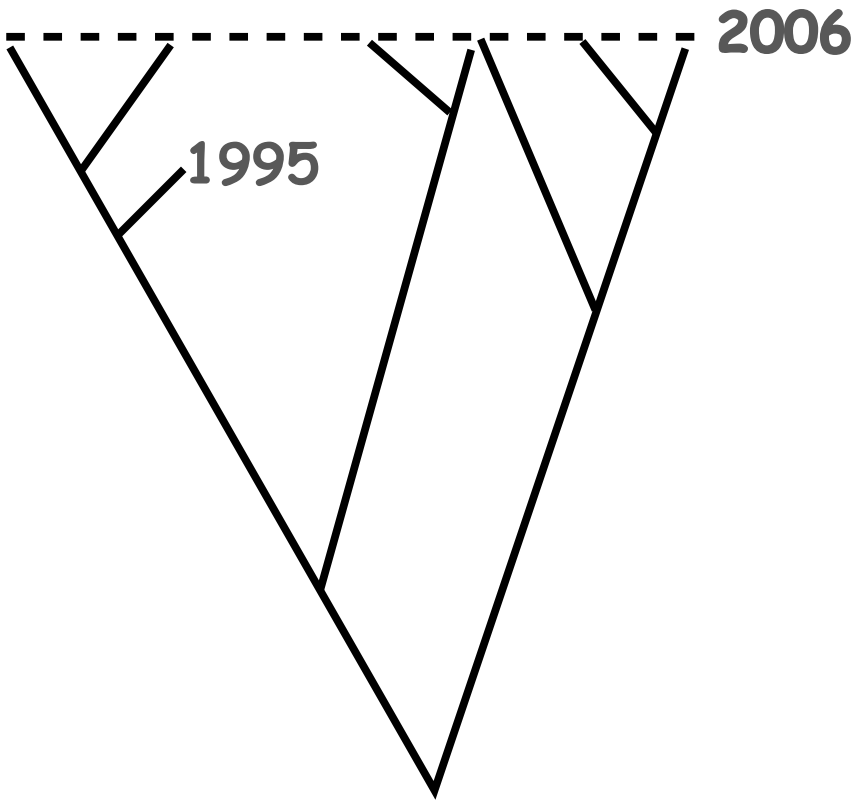
Bayesian Divergence Time Components

5. Fossil or other information

Prospects for much improved treatment of fossil evidence are good

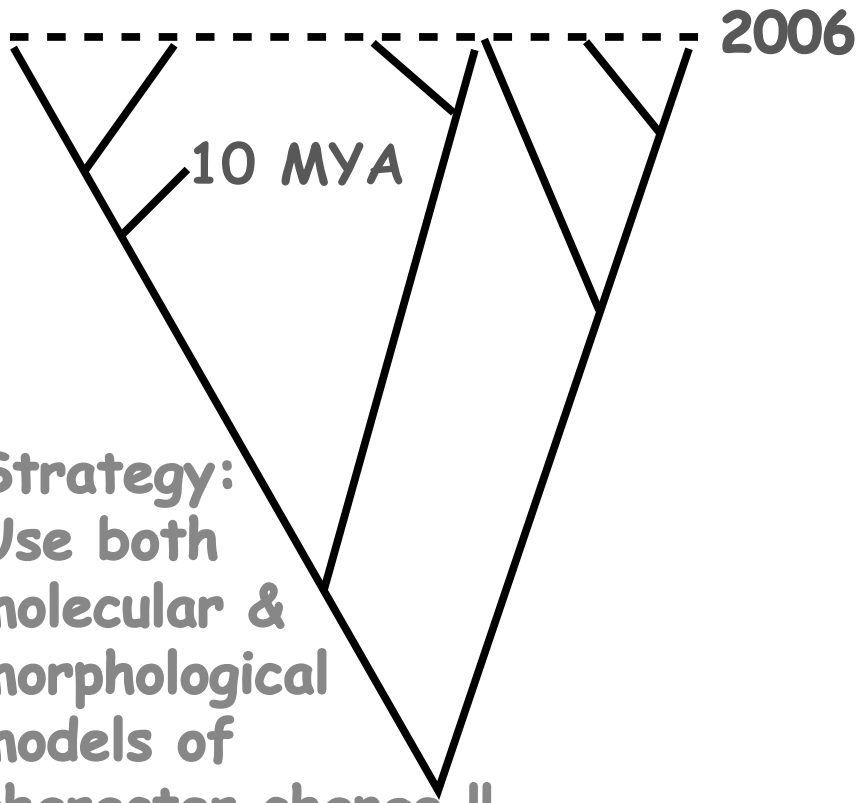
(particular progress by Ronquist et al. 2012. Syst. Biol. 61:973-999; see also Lee et al. 2009. Mol. Phylo. Evol. 50:661-666)

Can separate rates and times for quickly evolving (e.g., viral) lineages but cannot for slow lineages.



Serially Sampled Data

Can get sequence data and morphological data for 2006. Can get morphological (fossil) data for 10 million years ago!



Strategy: Use both molecular & morphological models of character change !!

Bayesian Divergence Time Components

1. DNA or protein sequence data - Bountiful
2. Model of Sequence Change - Difficult
3. Model of Rate Change - Difficult
4. Prior Distributions for Rates, Times, etc. - ? ? ?
5. Fossil or other information - Progress !!

THE END!

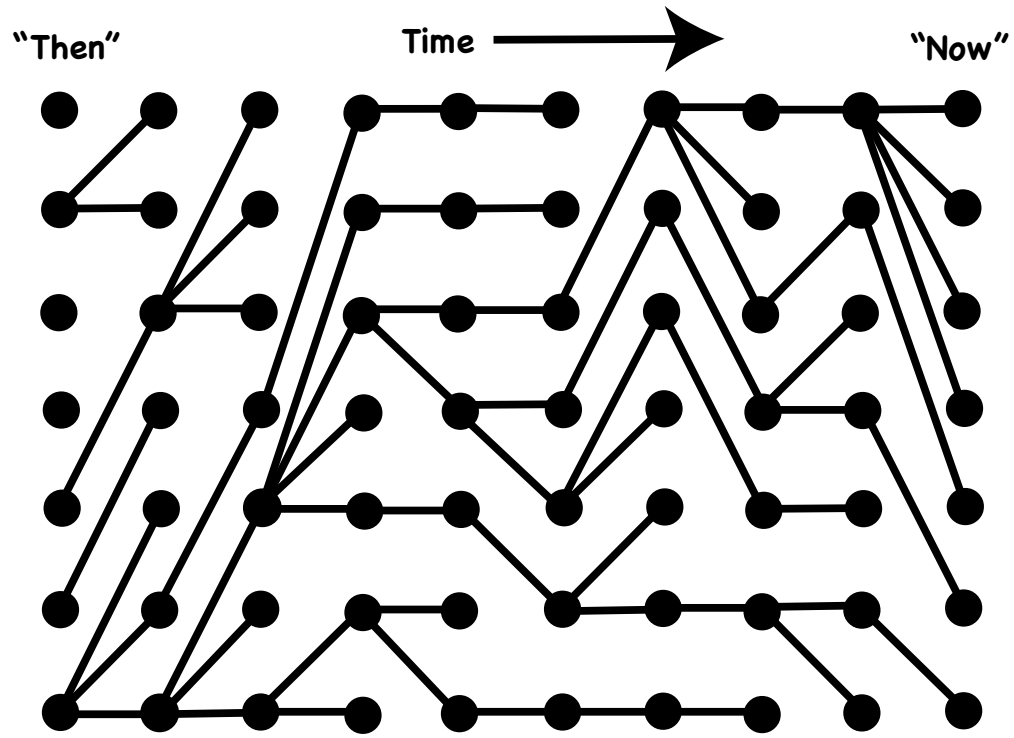
Some divergence time inference software:

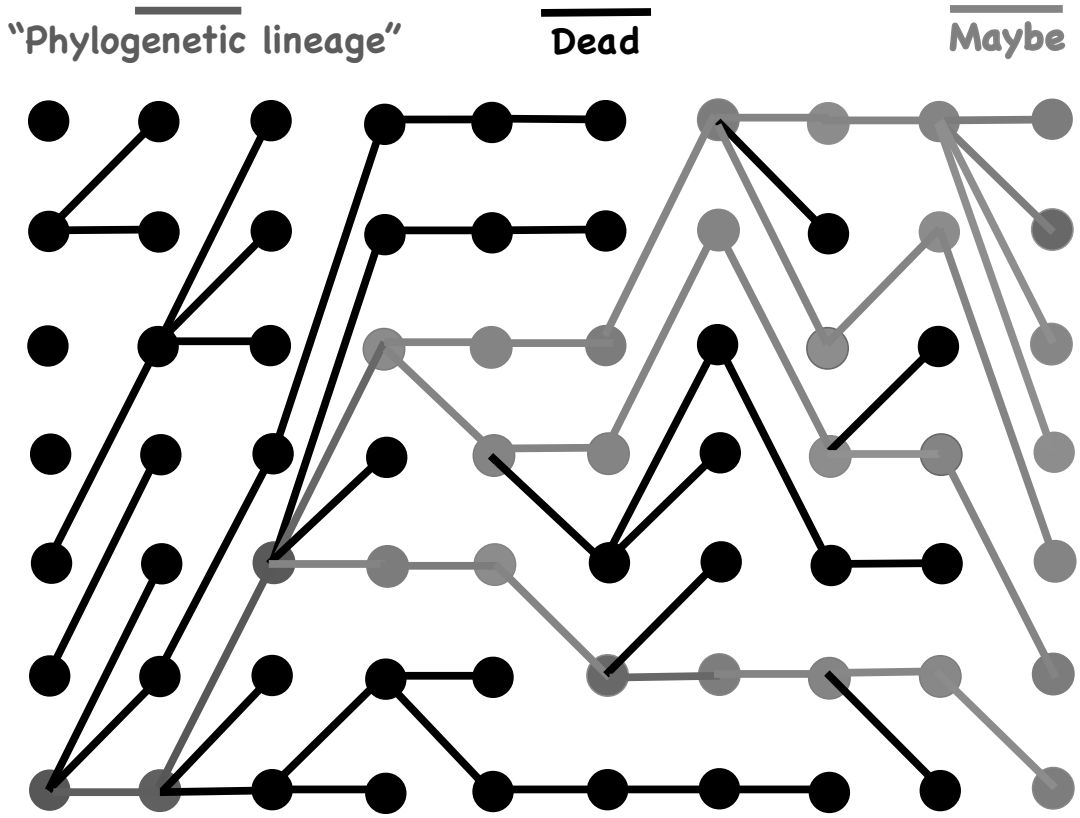
MrBayes	http://mrbayes.sourceforge.net
Beast	http://beast.bio.ed.ac.uk/
CoEvol	www.phylobayes.org/
DPPDiv	http://phylo.bio.ku.edu/content/tracy-heath-dppdiv
PAML	http://abacus.gene.ucl.ac.uk/software/paml.html

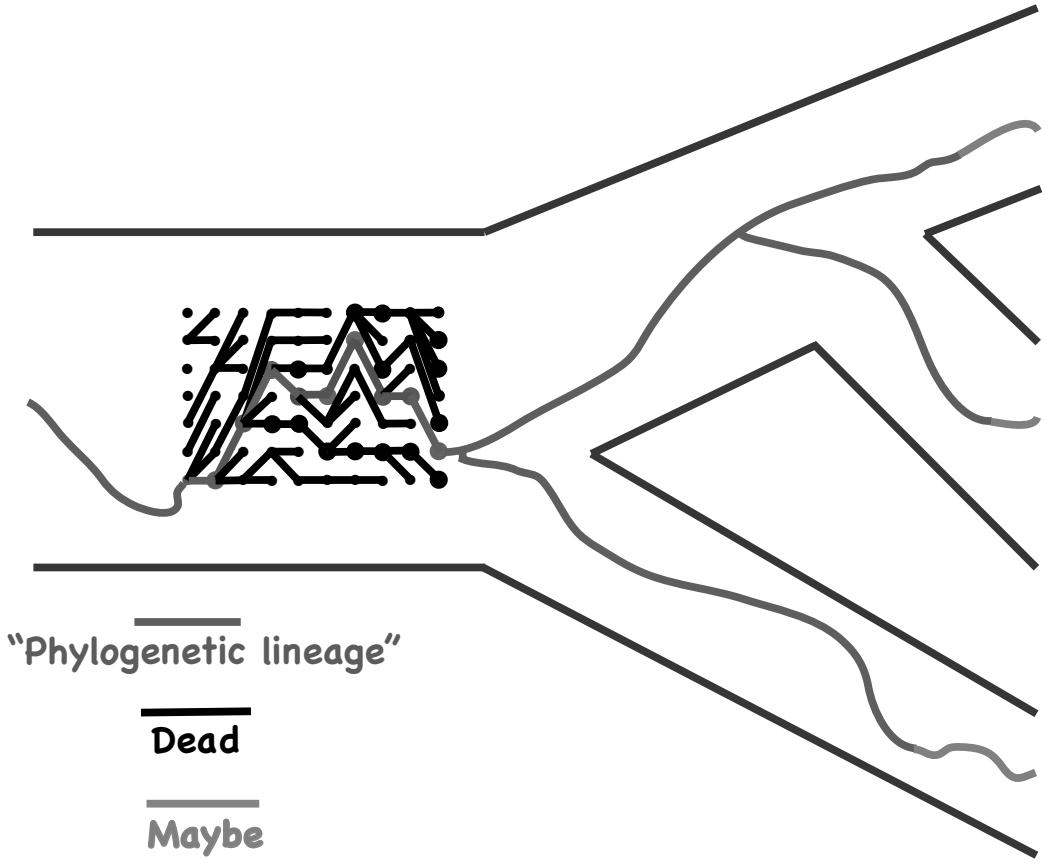
A digression:

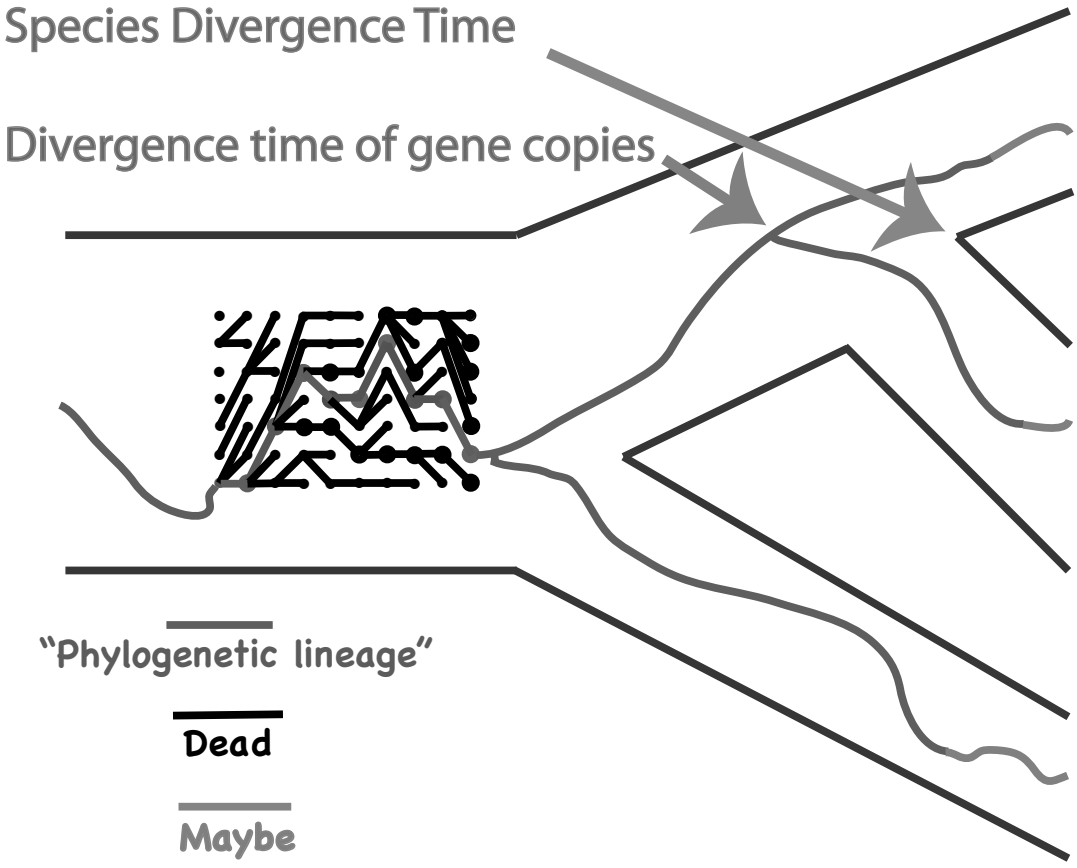
What are we really estimating when we estimate "divergence" times?

History of gene copies in a population

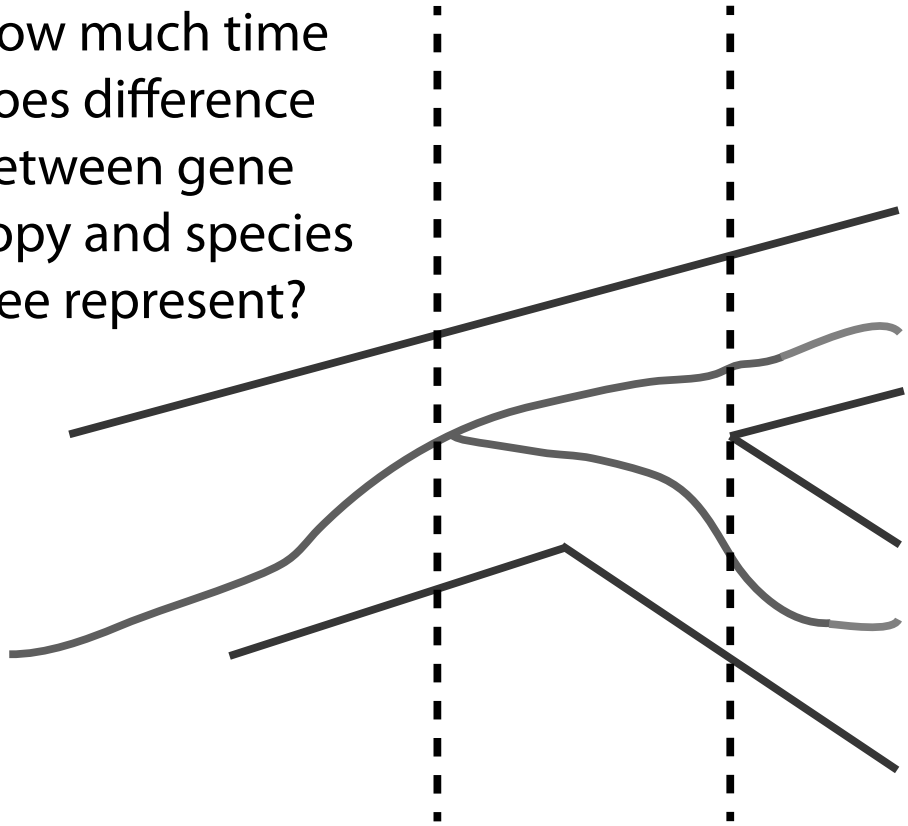





"Phylogenetic lineage"
Dead
Maybe

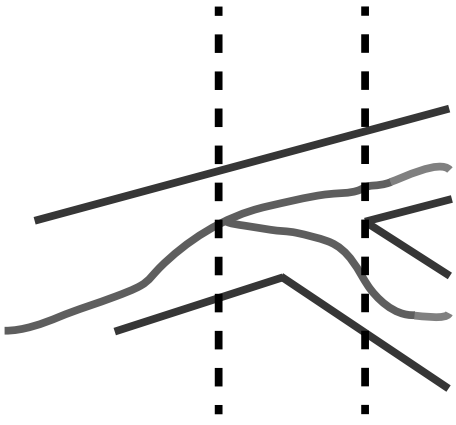

Species Divergence Time
Divergence time of gene copies
"Phylogenetic lineage"
Dead
Maybe

How much time does difference between gene copy and species tree represent?



How much time does difference between gene copy and species tree represent?

(N_e is effective population size)



For a coalescent process with diploid organisms, average time difference is $2N_e$ generations and standard deviation is also $2N_e$ generations ...

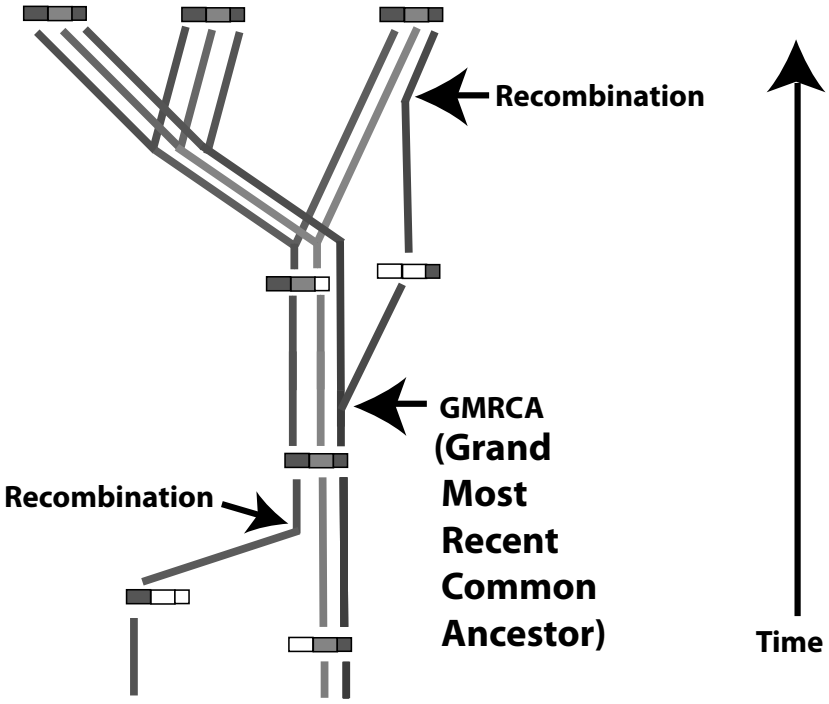
When time needed for $2N_e$ generations is large relative to species divergence times, be careful ...

and try *BEAST or BEST software?

See:

Heled & Drummond. 2012. MBE 27:570-580
Liu. 2008. Bioinformatics 24:2542-2543.

Recombination is another divergence time (and phylogenetic) challenge!



End of digression on ...

What are we really estimating when we estimate "divergence" times?

Demo of BEAST2 for divergence time estimation

Dr. Tracy Heath has graciously allowed Mark to present a demo based on her BEAST lab.

We won't have time for a full lab in which each student works through the steps on his/her own. However if you want to go through each step on your own, you'll find the materials at:

http://treethinkers.org/divergence-time-estimation-using-beast/

The steps that Mark will demonstrate are described in:
http://treethinkers.org/wp-content/uploads/2013/04/DivTime_BEAST2_tutorial_2014.pdf
That document also has a very nice, detailed description of the options and a great discussion of the issues surrounding divergence time estimation.

The data is at:
http://treethinkers.org/wp-content/uploads/2013/04/divtime_beast_data.tar.gz

The demonstration will use the software:
BEAST2: http://www.beast2.org/

FigTree http://tree.bio.ed.ac.uk/software/figtree/

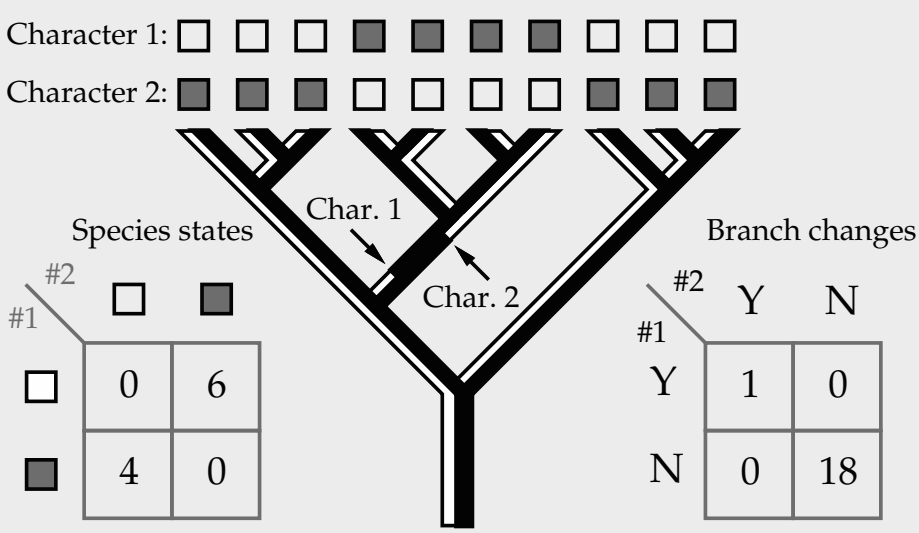
Tracer: http://tree.bio.ed.ac.uk/software/tracer/

Comparative method, coalescents, and the future

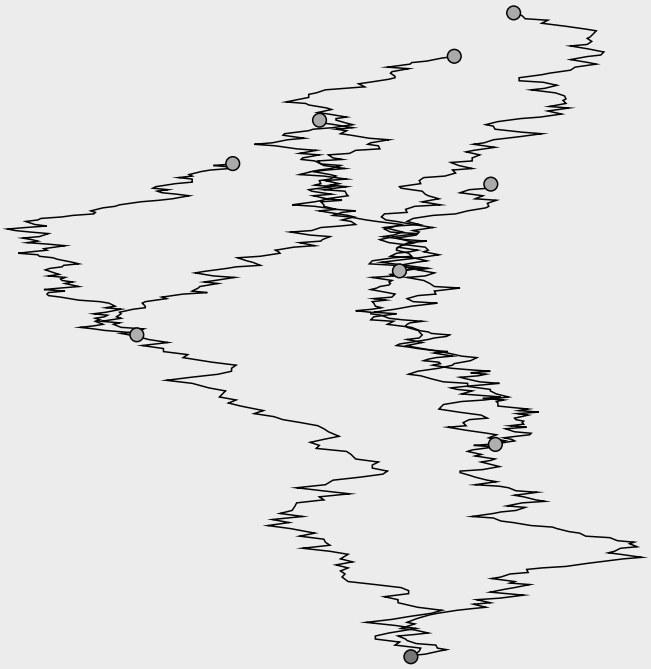
Joe Felsenstein

Depts. of Genome Sciences and of Biology, University of Washington

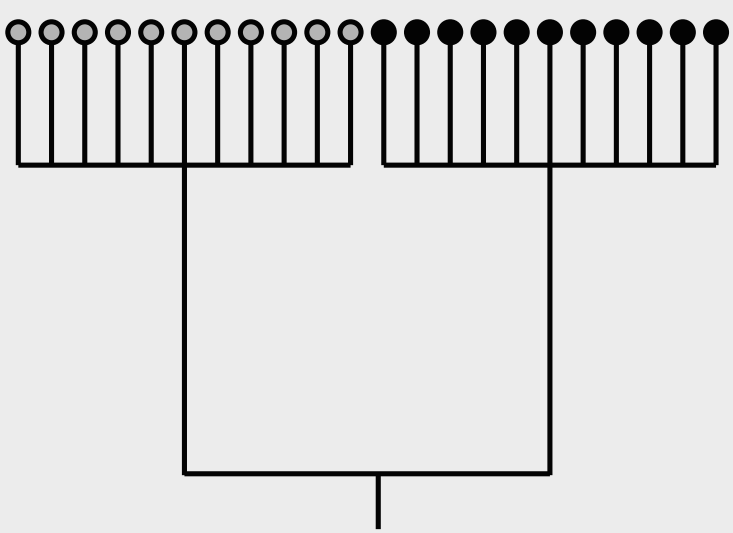
Correlation of states in a discrete-state model



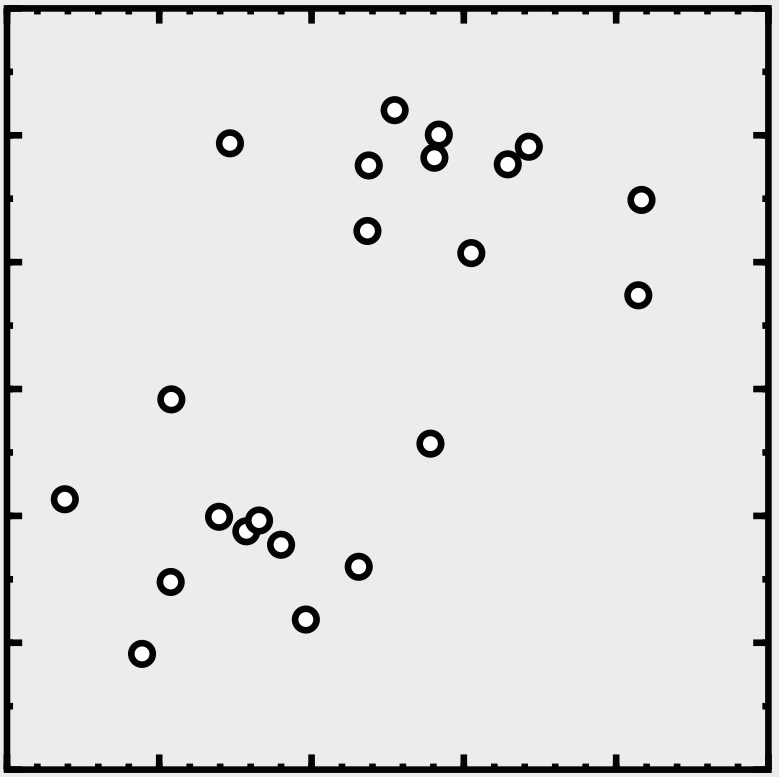
A simple model: Brownian motion



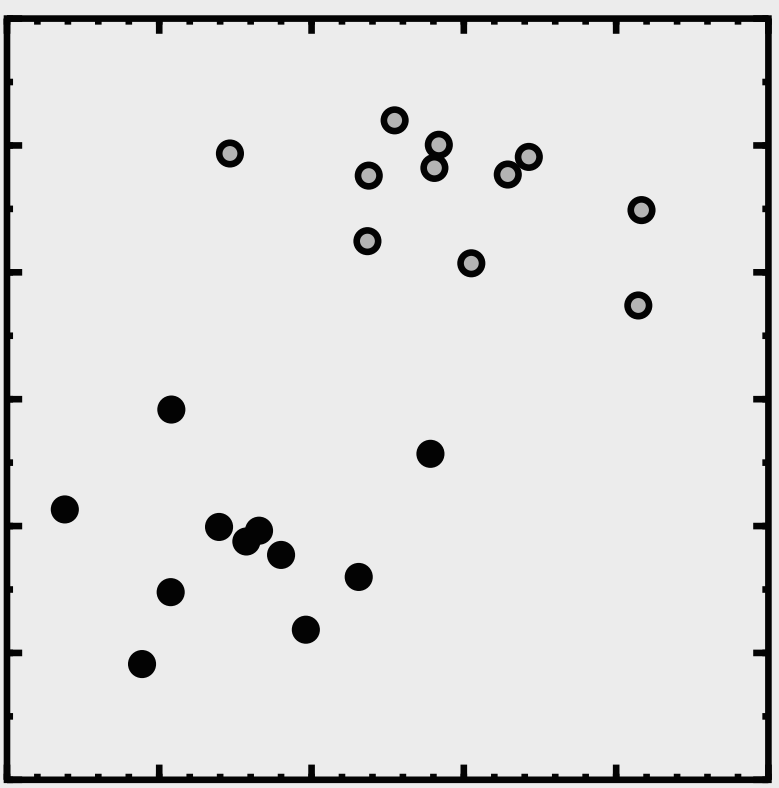
A simple case to show effects of phylogeny



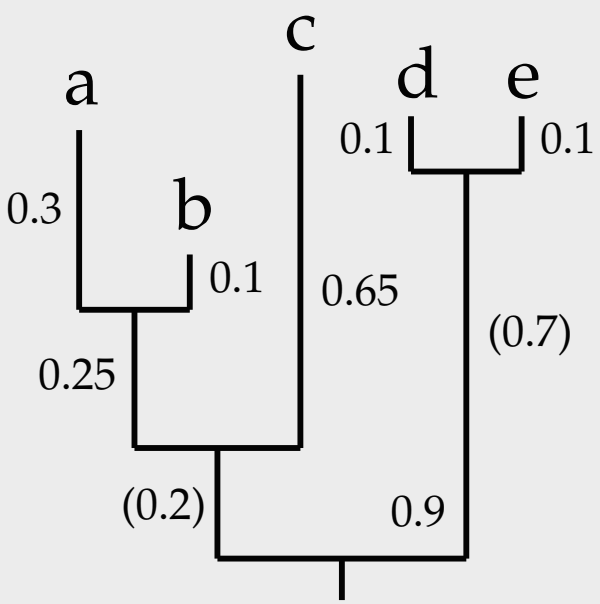
Two uncorrelated characters evolving on that tree



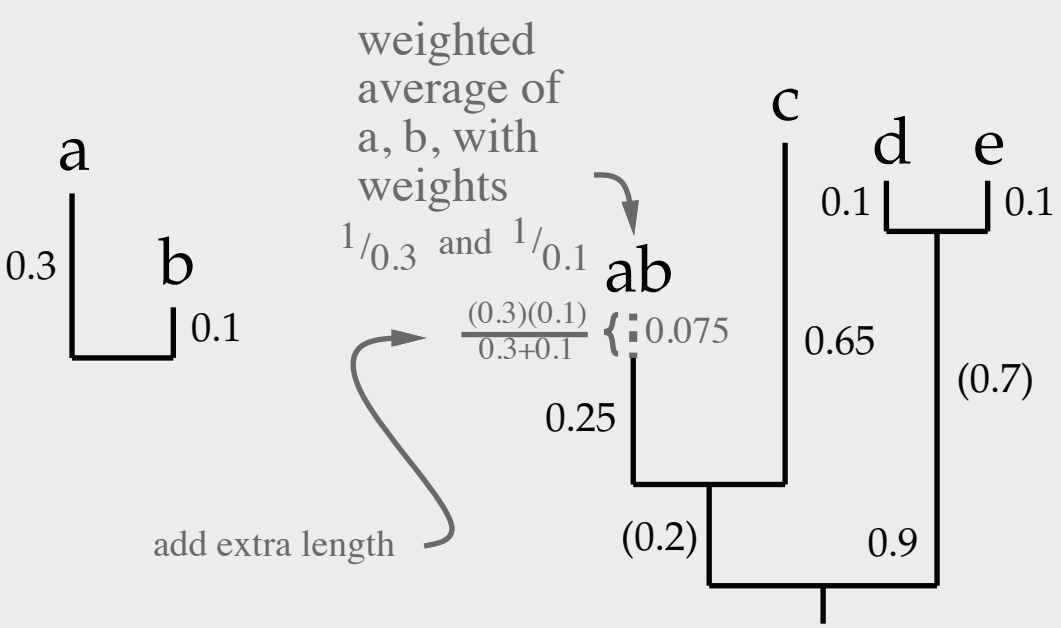
Identifying the two clades



A tree on which we are to observe two characters



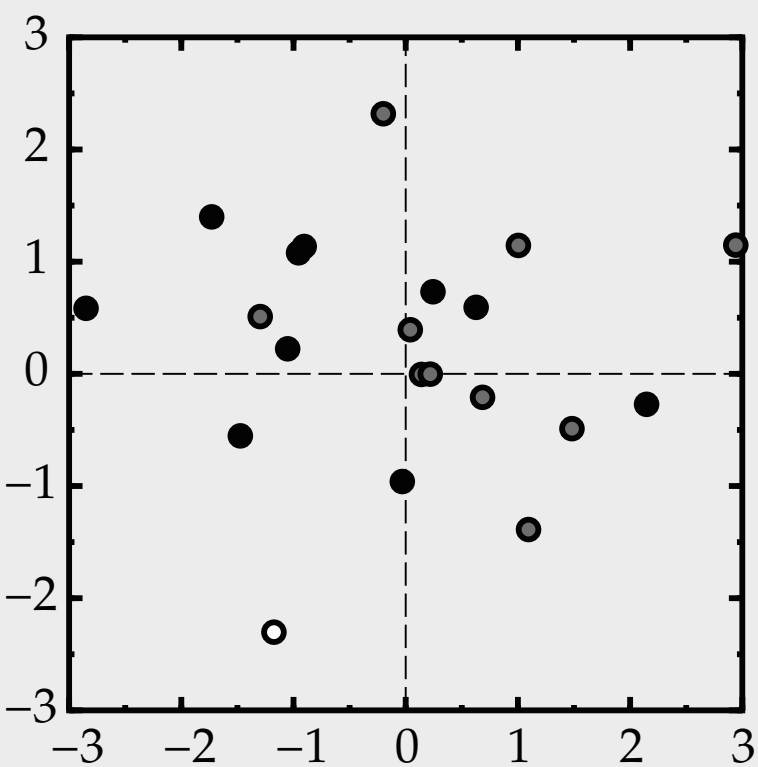
This turns out to be statistically equivalent to ...



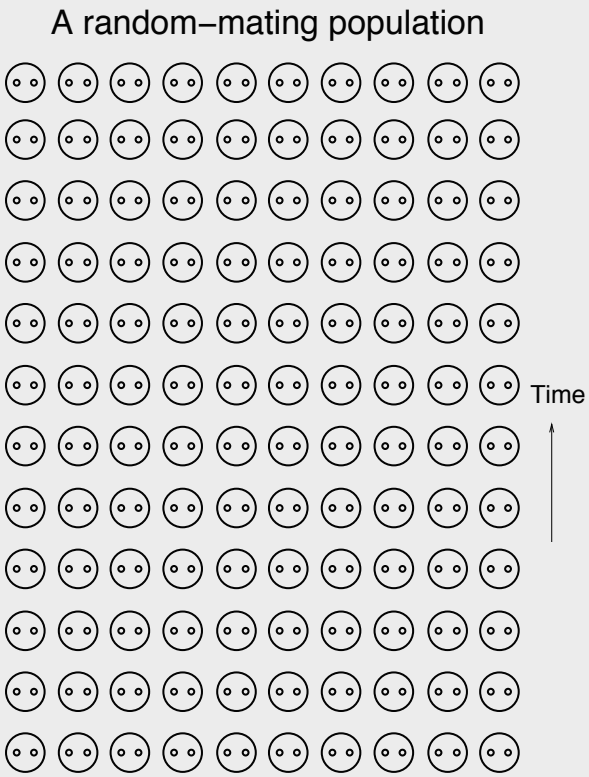
Contrasts on that tree

Contrast												Variance proportional to
y_1	$=$	x_a	$-$	x_b								0.4
y_2	$=$	$\frac{1}{4}x_a$	$+$	$\frac{3}{4}x_b$	$-$	x_c						0.975
y_3	$=$							x_d	$-$	x_e		0.2
y_4	$=$	$\frac{1}{6}x_a$	$+$	$\frac{1}{2}x_b$	$+$	$\frac{1}{3}x_c$	$-$	$\frac{1}{2}x_d$	$-$	$\frac{1}{2}x_e$		1.11666

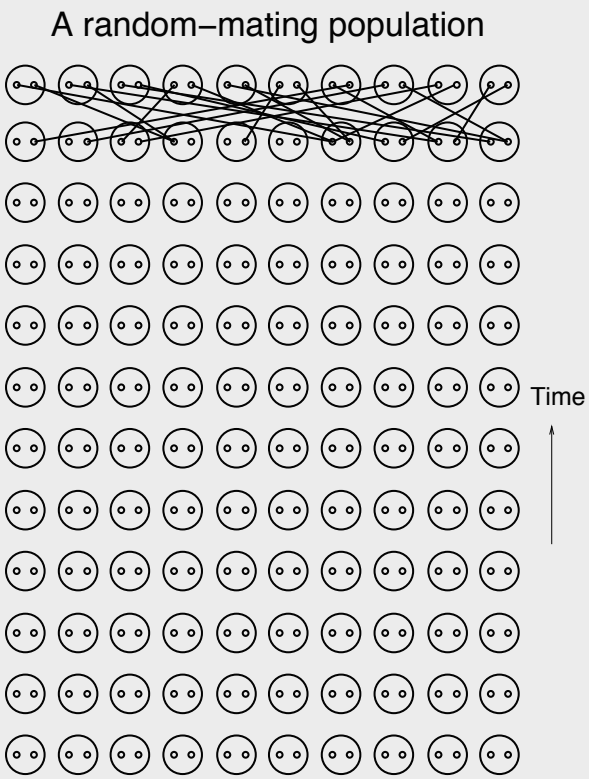
Plotting the contrasts against each other



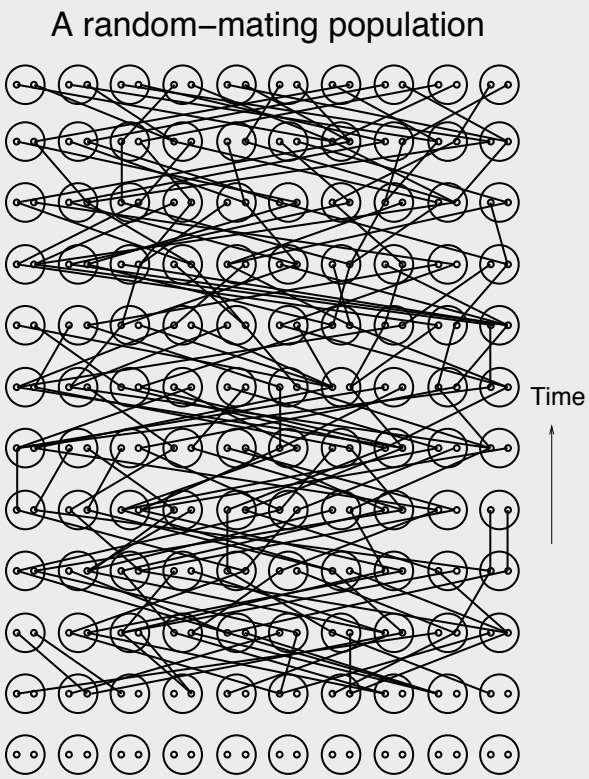
Gene copies in a population of 10 individuals



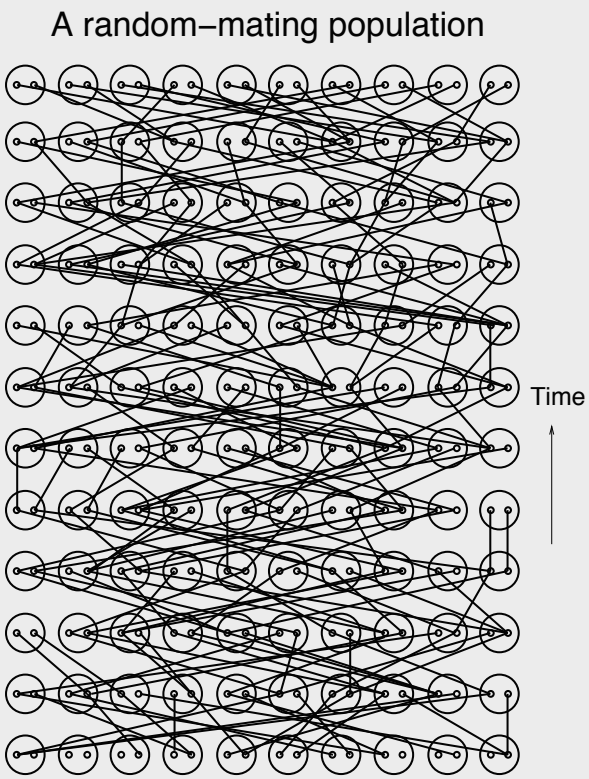
Going back one generation



... and one more

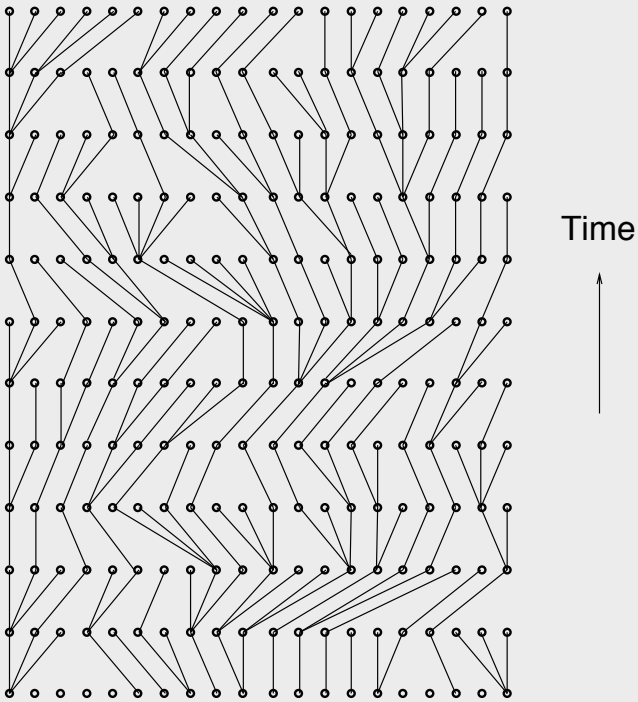


showing ancestry of gene copies



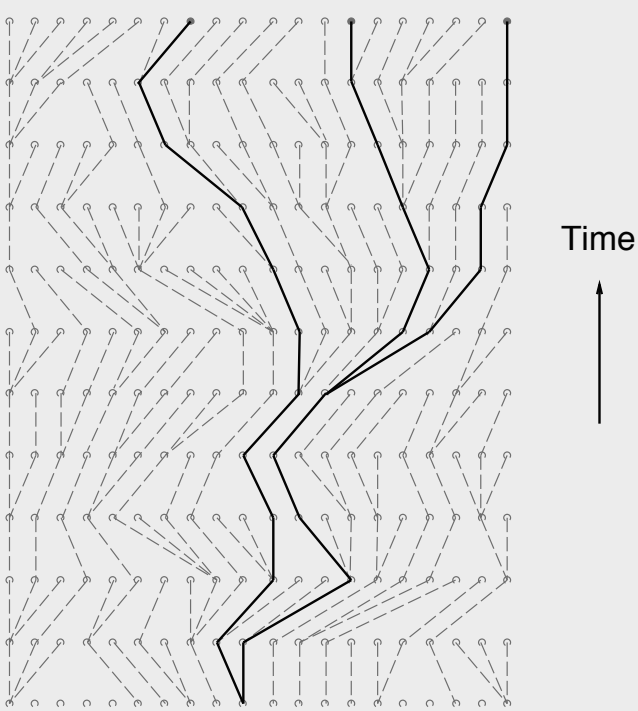
The genealogy of gene copies is a tree

Genealogy of gene copies, after reordering the copies

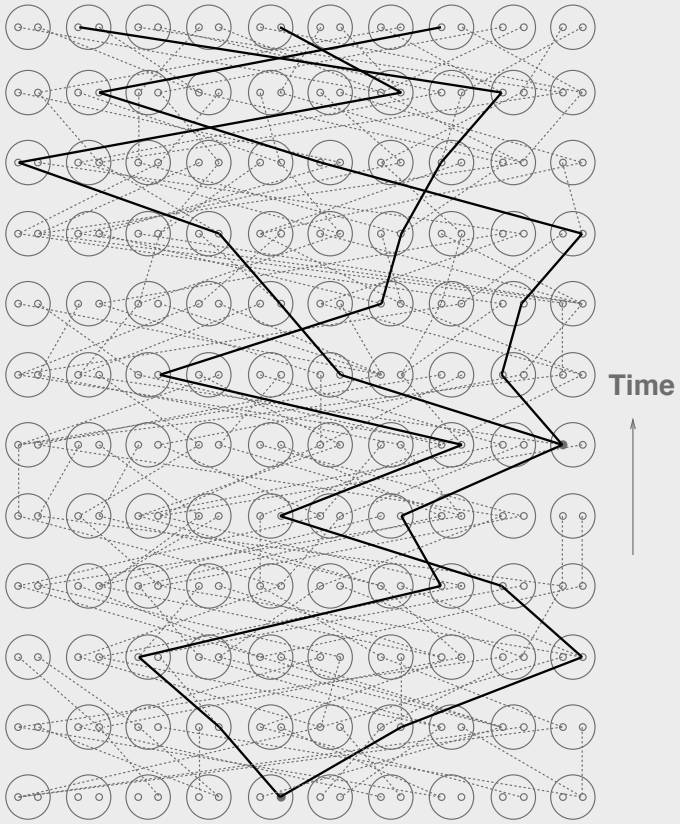


Ancestry of a sample of 3 copies

Genealogy of a small sample of genes from the population



Here is that tree of 3 copies in the pedigree



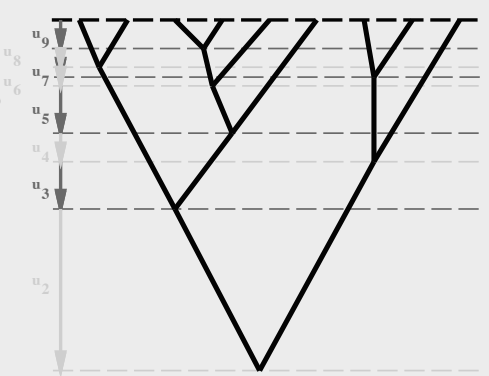
Kingman's coalescent

Coalescent trees of gene copies within species (Kingman, 1982)

Random collision of lineages as go back in time (sans recombination)
Collision is faster the smaller the effective population size

Average time for
k copies to coalesce to
k–1 = $\frac{4N}{k(k-1)}$

Average time for
two copies to coalesce
= 2N generations

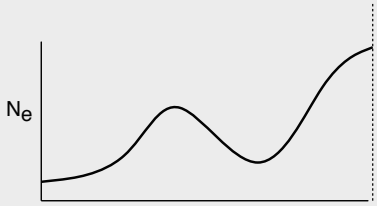


In a diploid population of
effective population size N,

Average time for n
copies to coalesce
= $4N\left(1-\frac{1}{n}\right)$ generations

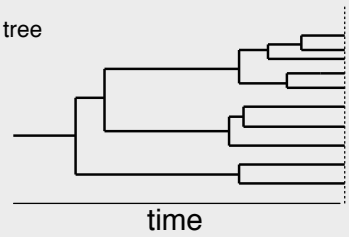
Coalescence is faster in small populations

Change of population size and coalescents

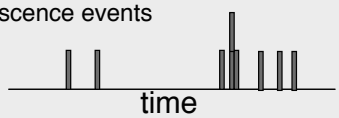


the changes in population size will produce waves of coalescence

the tree

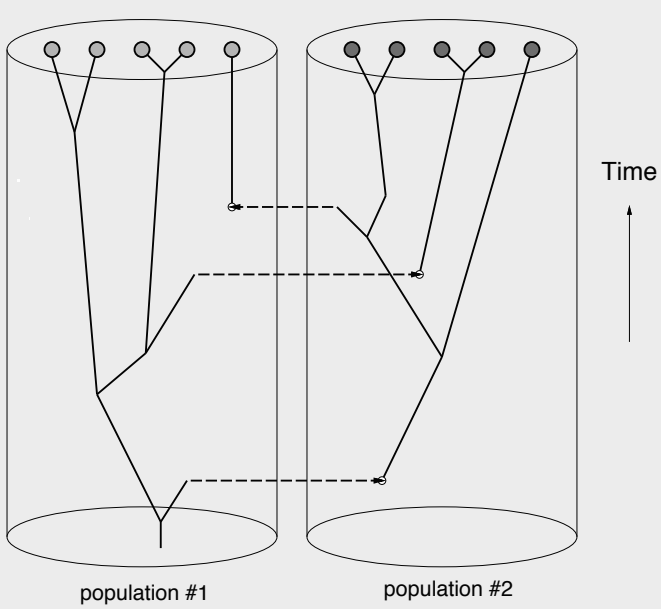


Coalescence events

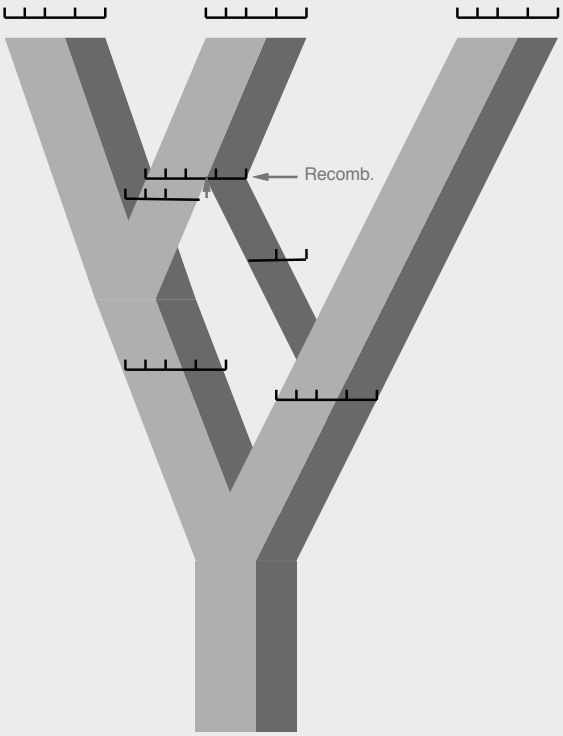


The parameters of the growth curve for N_e can be inferred by likelihood methods as they affect the prior probabilities of those trees that fit the data.

Migration can be taken into account

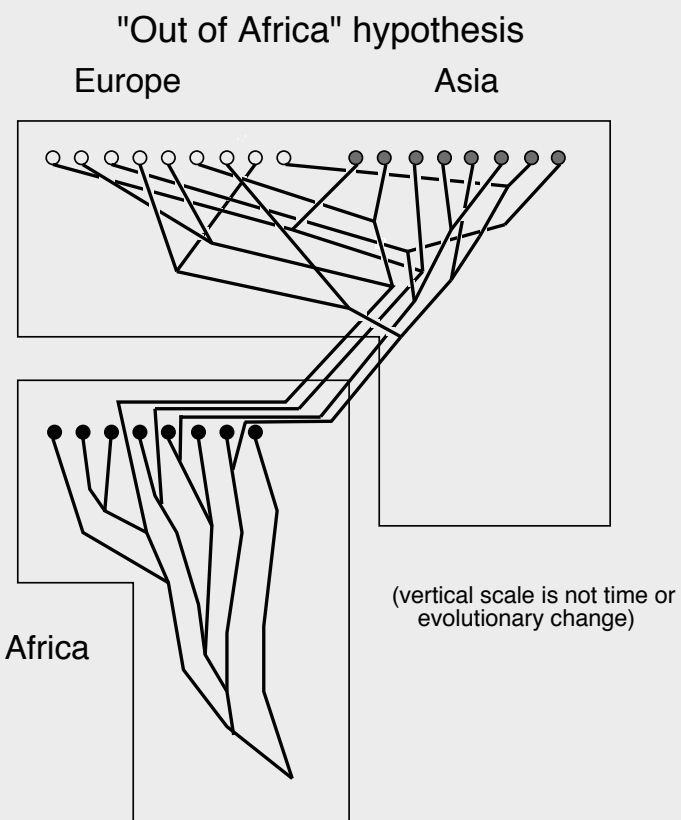


Recombination creates loops



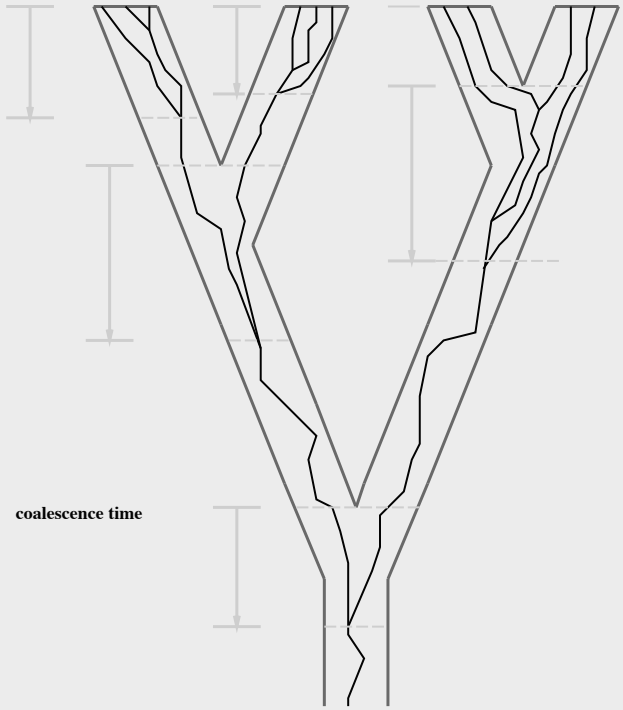
Different markers have slightly different coalescent trees

We want to be able to analyze human evolution



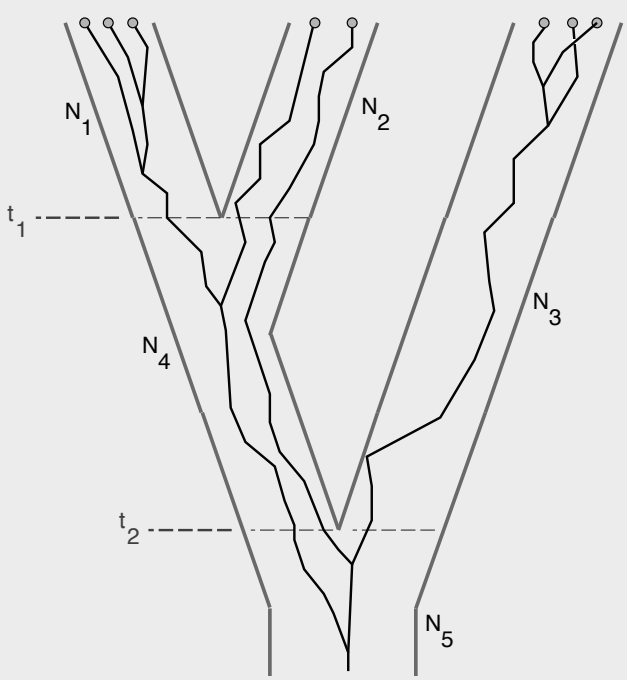
coalescent and “gene trees” versus species trees

Consistency of gene tree with species tree



If the branch is more than N_e generations long ...

Gene tree and Species tree



What to do with coalescents?

- They are poorly estimated (often only a modest number of sites is available for each tree).
- Our interest is *not* in the coalescent tree itself, it is in the population and genetic parameters (population size, mutation rate, migration rate, population growth rate, rate of recombination).
- So we want to sum up likelihoods over our uncertainty about the tree, or do the equivalent in Bayesian terms.
- Got that? Our objective is *not* to "get the tree"! We don't end up with a tree!
- This can be done by Markov Chain Monte Carlo (MCMC) methods, in programs such as LAMARC, BEAST, MIGRATE, IMa or BEST (there are others too).

Topics for the future ...

- Use of many loci
- Use of SNP data on a large scale (if relevant)
- Use of whole-genome sequences (in the longer run)
- Integration of between-species and between-population studies with multiple loci across multiple species. IMPORTANT: If you are within a species, not all loci will have the same tree (we have just explained why, in the discussion of recombination). So you ought to consider coalescents that differ between loci, between SNPs and *not* just infer "the tree". (Also, please do *not* do phylogenies of individuals).
- Integration of between-species and between-population studies with QTL mapping
- Integration of between-species and between-population studies with morphological characters.
- Inferences of, and using, genomic changes (comparative genomics)
- More rigorous statistical models for quantitative traits, especially in fossils (hominoid fossils, anyone?)
- ~~Using phylogenies to analyze multispecies microarray data~~

References

Comparative methods

Felsenstein, J. 1985. Phylogenies and the comparative method. *American Naturalist* **125:** 1-15. [The contrasts method]

Harvey, P. H. and M. D. Pagel. 1991. *The Comparative Method in Evolutionary Biology*. Oxford University Press, Oxford. [Reviews early work by me, Marl Ridley and the authors on comparative methods]

Pagel, M. 1994. Detecting correlated evolution on phylogenies: A general method for the comparative analysis of discrete characters. *Proceedings of the Royal Society of London, Series B* **255:** 37-45. [Method for two-state discrete characters]

Felsenstein, J. 2004. *Inferring Phylogenies*. Sinauer Associates, Sunderland, Massachusetts. [Especially chapter 25 which covers comparative methods]

Felsenstein, J. 2012. A comparative method for both discrete and continuous characters using the threshold model. *American Naturalist* **179:** 145-156. [Using Sewall Wright's 1934 "threshold model" to get a comparative method that can handle both discrete and continuous characaters]

The coalescent

Griffiths, R. C. and S. Tavaré. 1994a. Sampling theory for neutral alleles in a varying environment. *Philosophical Transactions of the Royal Socety of London, Series B (Biological Sciences)* **344:** 403-10. [The pioneering sampling method]

(continued)

Kuhner, M. K., J. Yamato, and J. Felsenstein. 1995. Effective population size and mutation rate from sequence data using Metropolis-Hastings sampling. *Genetics* **140:** 1421-1430. [Our MCMC coalescent likelihood method]

Hein, J., M. Schierup, and C. Wiuf. 2005, *Gene Genealogies, Variation and Evolution: A Primer in Coalescent Theory*. Oxford University Press, Oxford. [One of two books so far on coalescents. Light on estimation issues]

Wakeley, J. 2008. *Coalescent Theory*. Roberts and Co., Greenwood Village, Colorado. [One of two books so far on coalescents. Light on estimation issues.]

Nielsen, R. and M. Slatkin. 2013. An Introduction to Population Genetics. Theory and Applications. Sinauer Associates, Sunderland, Massachusetts. Population genetics textbook with more coverage of coalescents than usual.

Felsenstein, J. 2004. *Inferring Phylogenies*. Sinauer Associates, Sunderland, Massachusetts. [Especially chapter 27 which covers MCMC likelihood approaches (but explanation of logic of Griffiths/Tavaré method is wrong)]

Felsenstein, J. 2007. Trees of genes in populations. pp. 3-29 in *Reconstructing Evolution. New Mathematical and Computational Advances,* pp. 3-27 in by O. Gascuel and M. Steel. Oxford University Press, Oxford. [Review of coalescents including MCMC, for a somewhat mathematical audience]