Evolutionary Tree Analysis

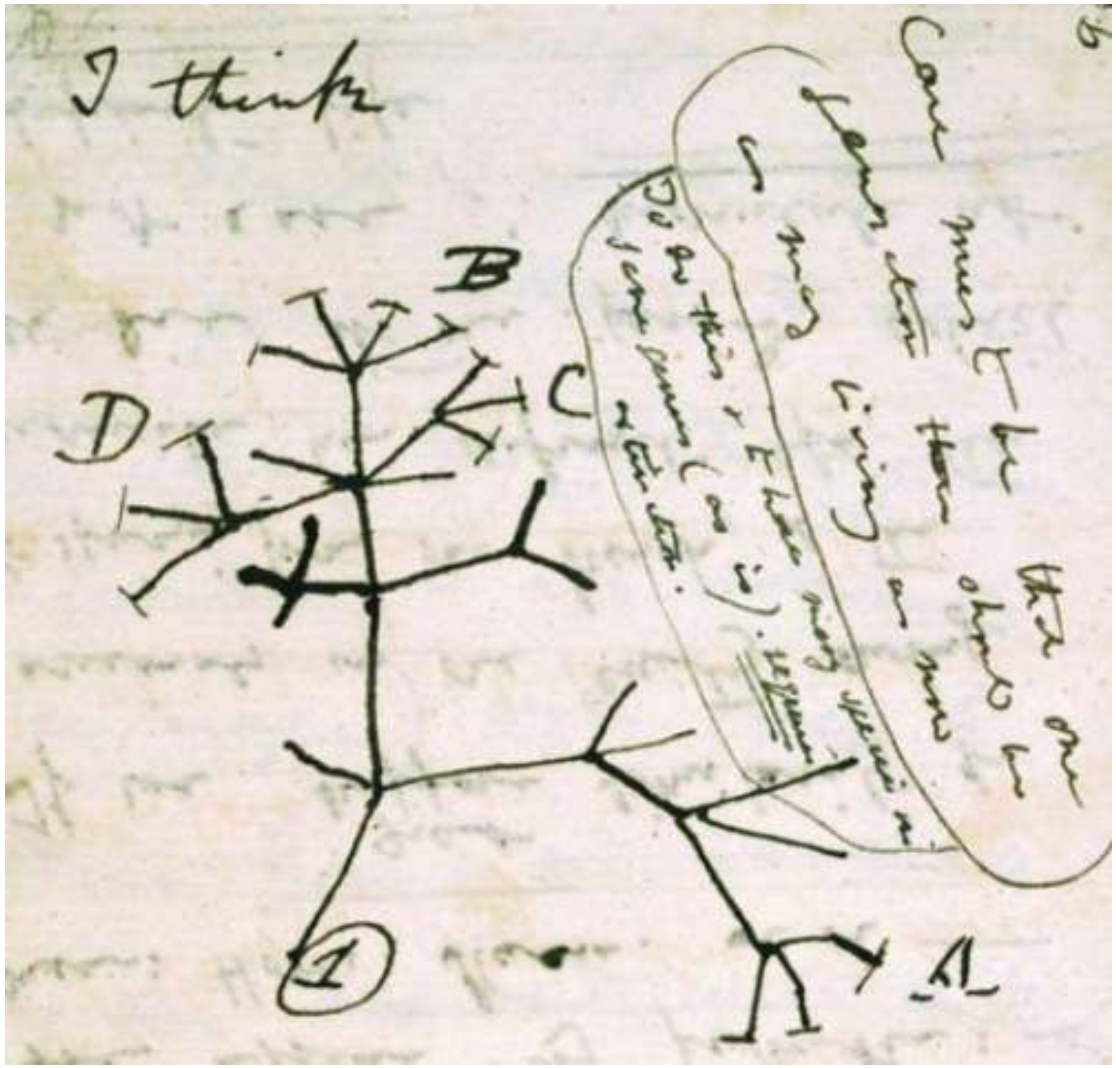
Christopher Lee

Department of Chemistry and Biochemistry, UCLA

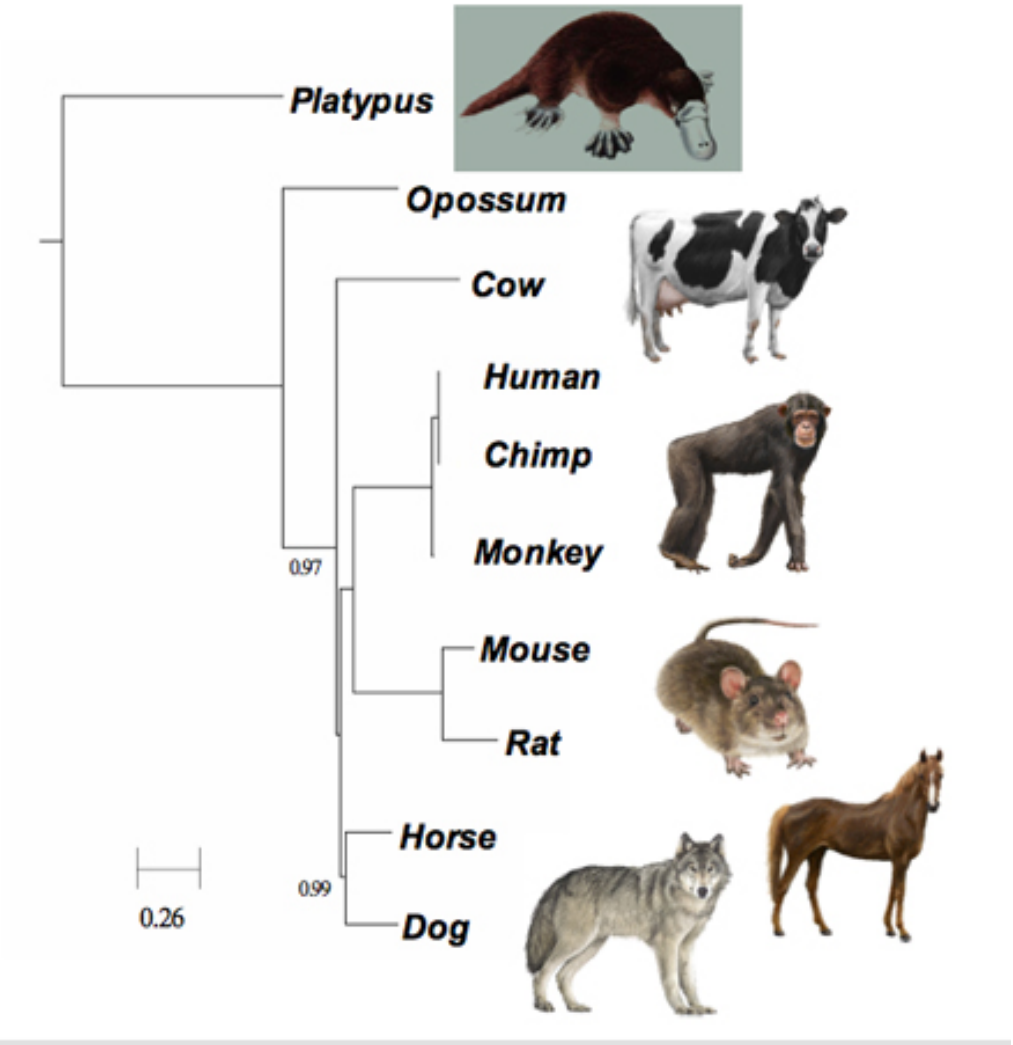
Announcements

- Mission Training due **Friday**.
- No project this week.

Darwin Draws the First Phylogeny, 1838



A Whole-Genome Based Tree

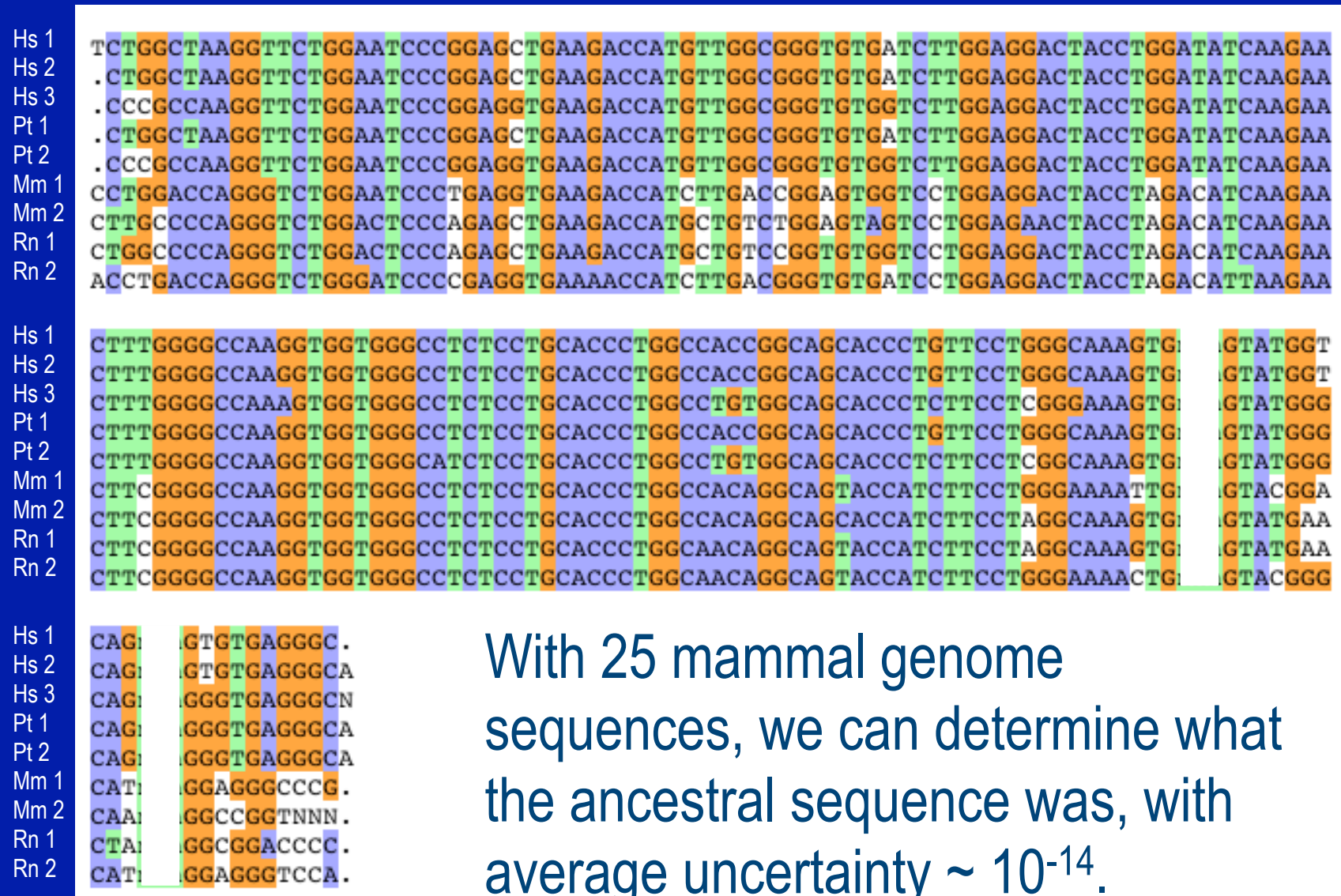


(S.H. Kim & co-workers, 2009)

Sequencing the ancestor of all mammals (100 million

- Already sequenced: human, mouse, rat, chimp, dog, cat…
- 25 mammalian genomes will be sequenced, carefully selected to give best coverage of families (i.e. choose mammals that are as unrelated to each other as possible).
- From the alignment, we can reconstruct the genome of the ancestor of all mammals!

Example: Human, Chimp, Mouse, Rat



With 25 mammal genome sequences, we can determine what the ancestral sequence was, with average uncertainty ~ 10^{-14}.

Evolution Challenges

- Given data for modern organisms (e.g. sequences), can we figure out the history (tree) of life?
- Can we measure which parts of the tree are confident, and which are not?
- Can we reconstruct ancestral genome sequences?
- Can we use comparisons of genome sequences to figure out which parts of the genome have the most important functions (and are most conserved by evolution)?

Your Mission: Reconstruct Evolutionary History

- *Jurassic Park* not withstanding, in general our only source of sequence data is modern species samples.
- Hence, we try to reconstruct the past evolutionary history of modern species by comparing their sequences.

Your mission this week is to construct such a **phylogenetic tree** for a set of modern sequences, given their "evolutionary distances" from each other.

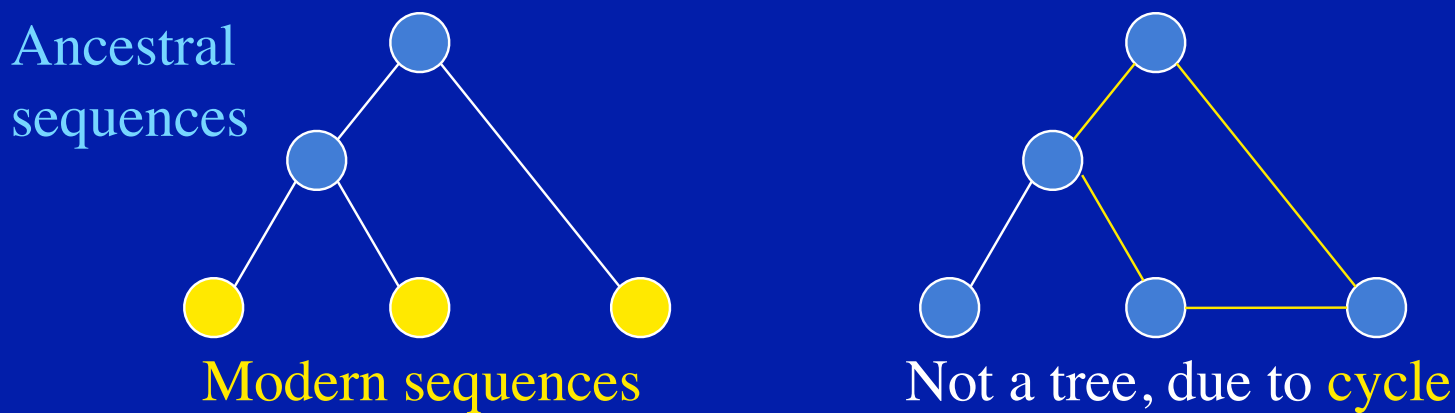
- This mission training is due next Monday (noon), see the link in CCLE Week 9.
- No programming is required.

Phylogeny: Standard Assumptions

- Sequences diverge by bifurcation events (no trifurcations etc.). (Model forms a tree).
- Sequences are essentially independent once they diverge from their common ancestor.
- The probability of observing nucleotide k at site j in the future depends only on the current nucleotide at site j. (Markov Chain assumption).
- Different sites (characters) within a sequence evolve independently.

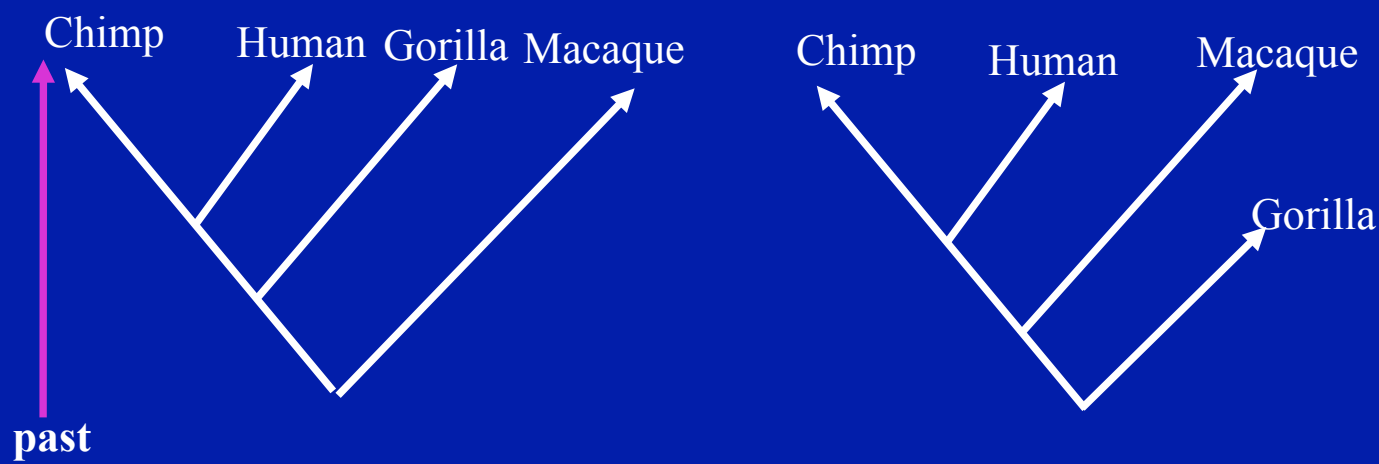
What is a Tree?

- A tree is a *connected graph* consisting of *nodes* (sequences) and *edges* (that each connect a pair of nodes) with no *cycles* (path of edges that returns to its starting point).
- There exists a single unique path between any pair of nodes.



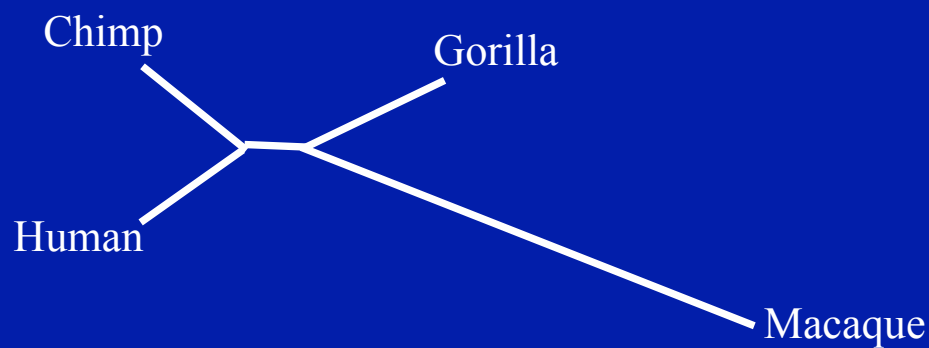
Rooted Versus Unrooted Trees

Rooted Trees



These two rooted trees are different
But both have the same Unrooted tree:

Unrooted Tree

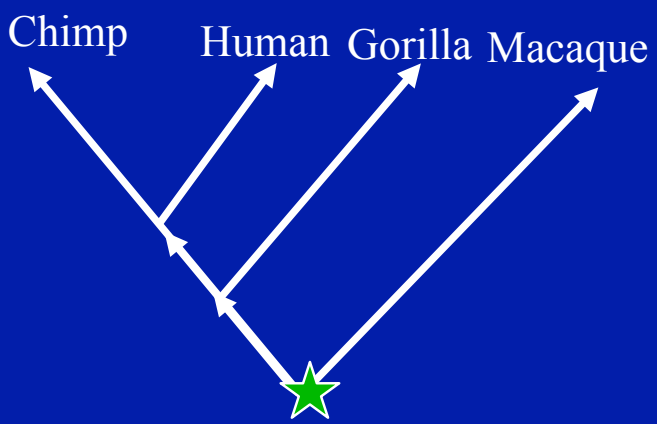


The direction of evolution is specified in a rooted tree but not in an unrooted tree

Rooted Trees Are Directed Graphs

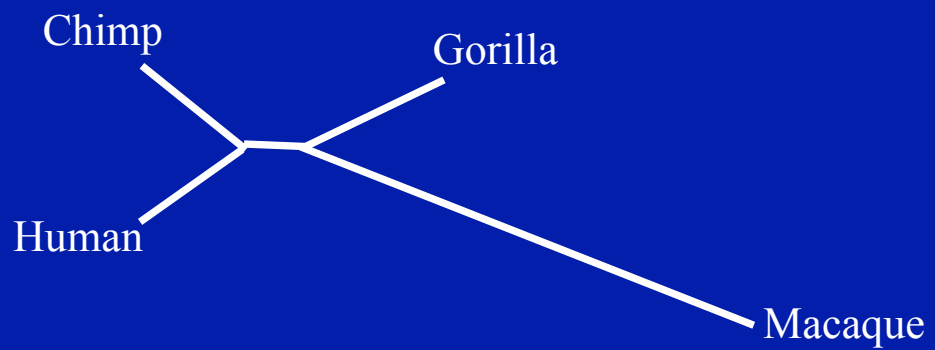
In a rooted tree each edge is directed, pointing from root to "leaves".

Rooted Tree



In an unrooted tree, the edges are "undirected".

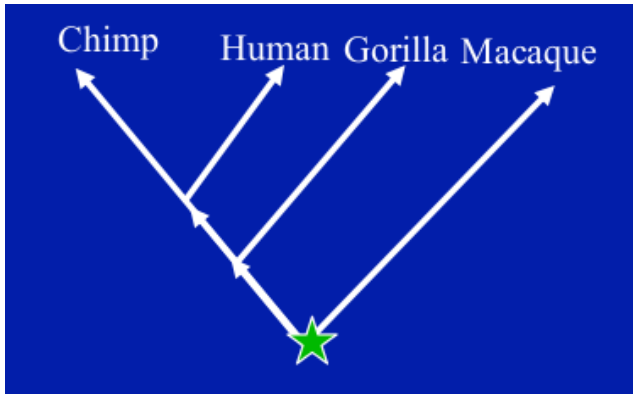
Unrooted Tree



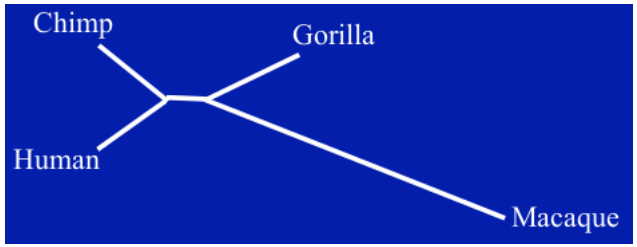
Ancestors of Human and Chimp?

Consider the following rooted and unrooted trees.

- Which point(s) in the rooted tree could be considered an ancestor of both human and chimp?



- Which point(s) in the unrooted tree could be considered an ancestor of both human and chimp?



Ancestors of Human and Chimp? Answer

- in a rooted tree, the edge direction represents the direction of time. Hence any point that has a directed path to both human and chimp is an ancestor of human and chimp. Concretely, this is the set of points from the vertex where the branches to human and chimp meet, up to the tree root.
- in an unrooted tree, the direction of time on each edge is unknown. Hence an ancestor of human and chimp could not only be anywhere on the path between them, but also anywhere else within the tree (internal point).

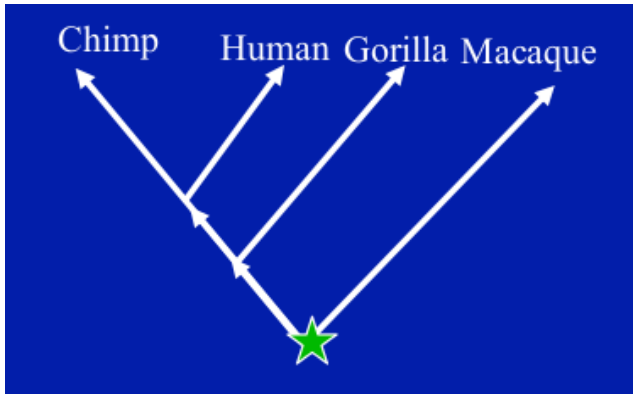
Tree Root = Most Recent Common Ancestor

- When we talk about a given group of sequences, usually the ancestor we are most interested in is their **most recent common ancestor**.
- Indeed this is so important that it has its own acronym (**MRCA**) and is one of the most common terms in phylogenetics.

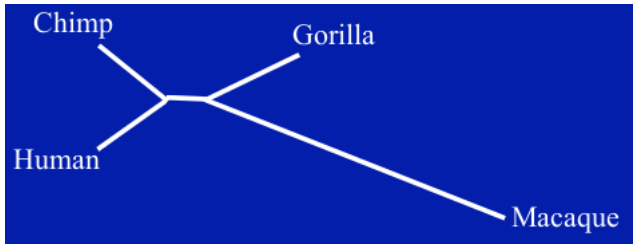
Most Recent Common Ancestor?

Consider the following rooted and unrooted trees.

- Which point(s) in the rooted tree could be considered the MRCA of human and chimp?



- Which point(s) in the unrooted tree could be considered the MRCA of human and chimp?



Most Recent Common Ancestor? Answer

- in a rooted tree, the edge direction represents the direction of time. Hence the vertex where the branches to human and chimp meet is the "first" (i.e. most recent) point that is an ancestor both. This is the MRCA. Note that this is the point that "cuts" the subtree containing human and chimp from the rest of the tree of life.
- in an unrooted tree, the direction of time on each edge is unknown. Hence the MRCA of human and chimp could be anywhere on the path between them. Note that while the root of the whole tree could be *anywhere* in the entire tree, the **most recent** common ancestor of human and chimp cannot be off the *direct path* between them (such a point off the direct path *could* be an ancestor of human and chimp, but not their *most recent* ancestor).

Which Sequences Are Most Related?

Say we have a tree of a set of modern sequences S, and a subset of these sequences G are all connected by a single vertex V that is *not* the MRCA of the tree. Are the sequences in G more closely related to each other in evolutionary time than to any other sequences in S? I.e. is their *time of divergence* more recent than their times of divergence from other sequences in S? (This is called a **monophyletic group**).

- in a rooted tree, for sequences G all reachable via two of the directed edges from vertex V?
- in a unrooted tree, for sequences G all reachable via two of the undirected edges from vertex V?

Which Sequences Are Most Related? Answer

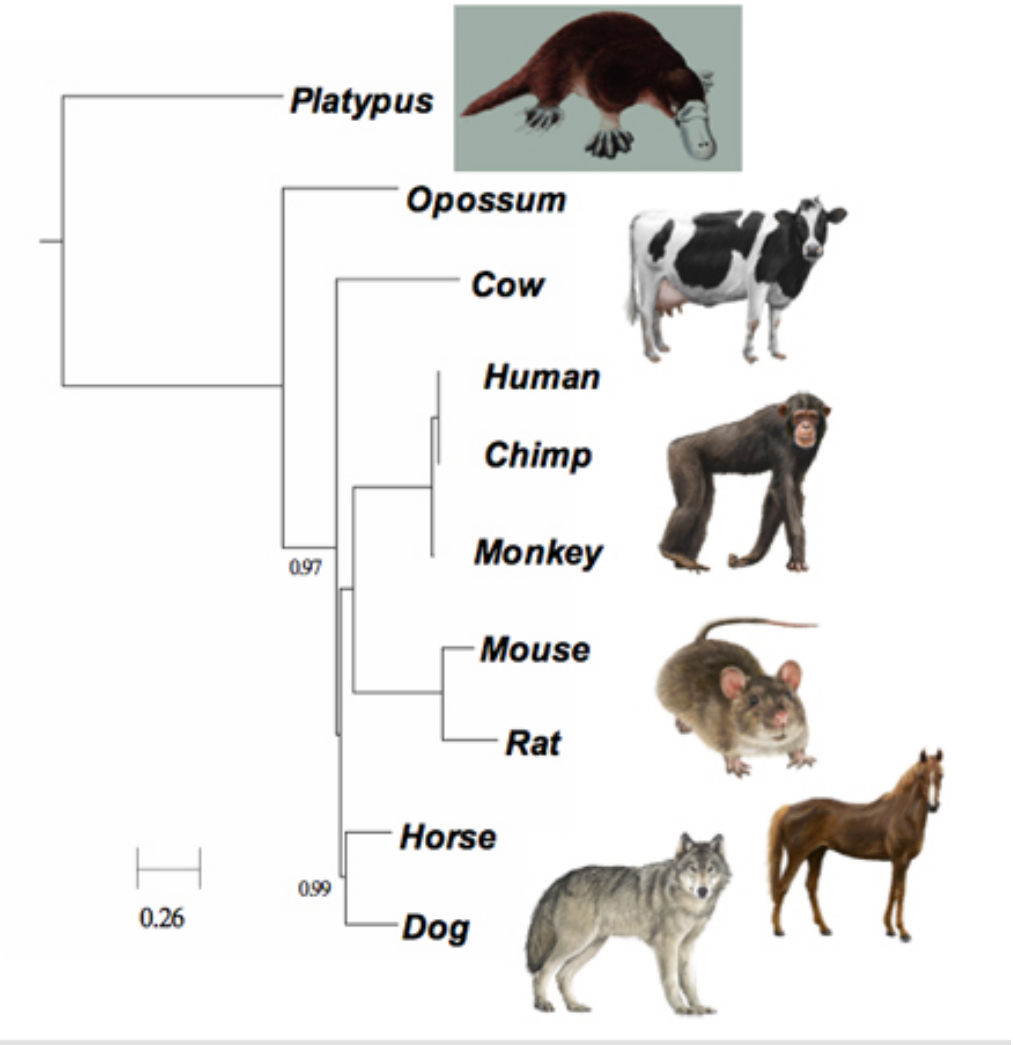
- in a rooted tree, the edge direction represents the *forward* direction of time. For the sequences in G to reach the other sequences requires traversing the edge "above" V in reverse time direction, i.e. going back to an older time of divergence. So yes, G is a monophyletic group.
- in an unrooted tree, the MRCA could be on *any edge* in the tree. If the MRCA is *inside* the subtree representing G, then there are sequences in G that are as distant from each other (in time) as they are from any other sequences in S. In that case G is *not* monophyletic (to make it monophyletic, we'd have to add all the sequences in S). If the MRCA is outside the G subtree, then G is monophyletic (by definition).

Monophyletic Groups

- Let $\tau(X, Y)$ be the *time of divergence* of sequences X, Y.
- Then for a specified set of sequences S, a subset G of those sequences is a **monophyletic group** iff

$$\tau(X, Y) < \tau(X, Z) \text{ for all } X, Y \in G \text{ and all } Z \in S \text{ and } Z \notin G$$

A Whole-Genome Based Tree



(S.H. Kim & co-workers, 2009)

Constant Evolutionary Rates?

- Imagine evolution is a *homogeneous* process, i.e. the same rates of mutation **at all times** during evolution.
- Say we take three related modern sequences (e.g. sequence of the same gene in three species descended from a common ancestor), and compare them by aligning all three sequences (assume this can be done accurately), and counting the differences between each pair of sequences.
- Assuming constant mutation rates, what approximately would you expect to see?

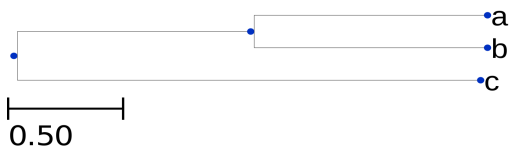
Constant Evolutionary Rates? Answer

If the mutation rates are constant at all times, we expect the number of mutations that occur on a given branch of evolution to be a measure of *evolutionary time*. So for two sequences that diverged a given amount of time ago, we expect them to have each accumulated an approximately equal number of mutations since they diverged.

- so for three related sequences A,B,C, the pair A, B that diverged most recently should have the fewest differences,
- and the number of differences from A and B to the third sequence C should be approximately equal.

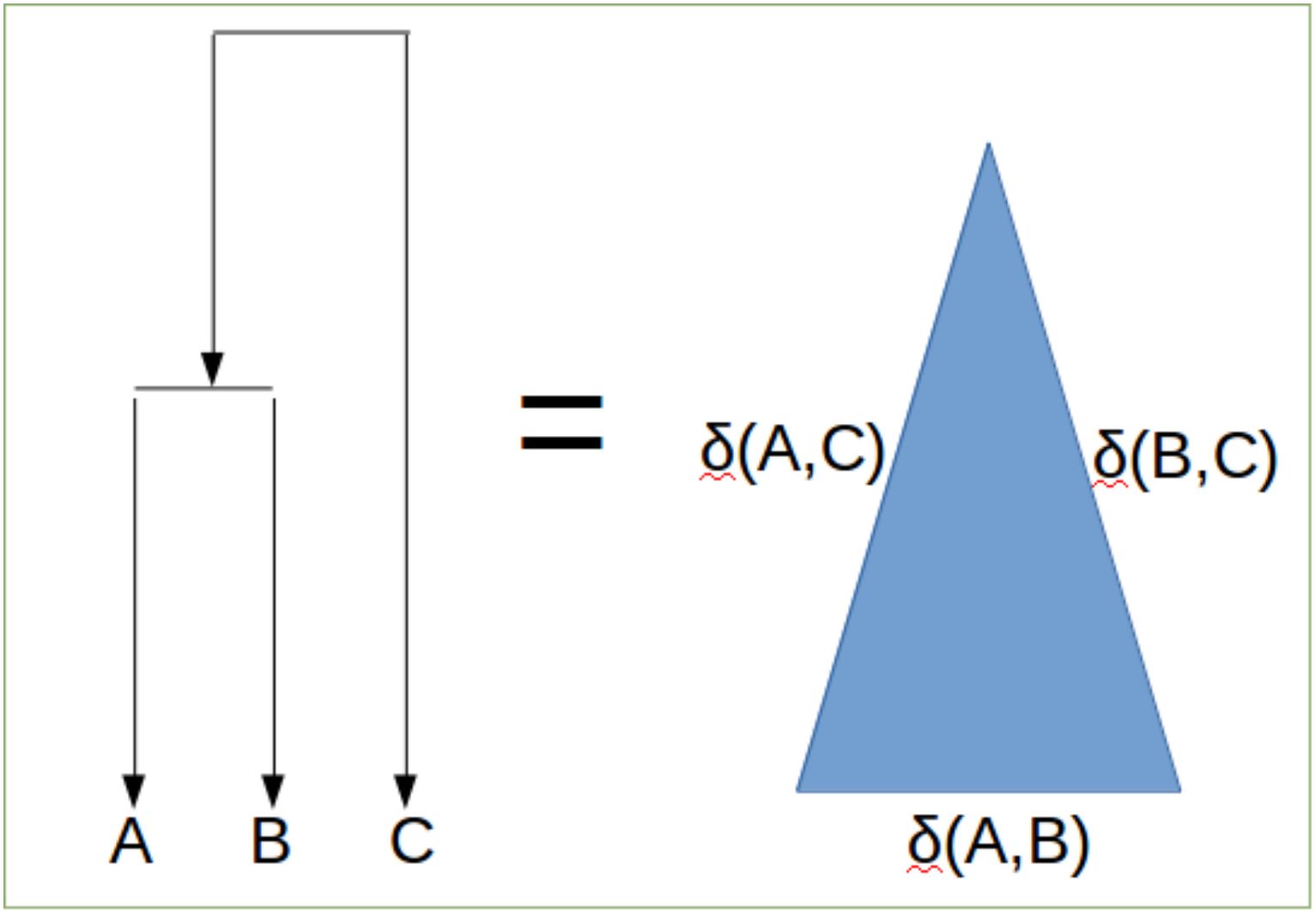
$$\delta(A,B) < \delta(A,C) \approx \delta(B,C)$$

The hypothesis of constant mutation rates is called the **molecular clock hypothesis**.



Molecular Clock "Slim Isosceles" Inequality

Easy to remember this molecular clock distance inequality as a *slim isosceles triangle*:



Inferring the Root Location

You wish to reconstruct the most recent common ancestor (MRCA) of a set of n sequences of a conserved gene from diverse mammal species. Say you are given an *unrooted* tree of these sequences. If the sequences obeyed the molecular clock assumption, suggest the simplest possible principle for identifying the correct location of the root, on the unrooted tree that you were given.

Inferring the Root Location Answer

- Under the molecular clock assumption, the distances of every sequence from their MRCA should be equal.
- Thus we could take the maximum pairwise distance in the tree, and find the midpoint on the tree between that pair of sequences. That should be the MRCA location.

Inferring the Root Location Errors

- Some people understood the MRCA should be equidistant from all leaf nodes, but didn't come up with a procedure for finding that point (e.g. take longest path and divide in half).

Tree Reconstruction under constant mutation rates?

Assume that a set of sequences evolved under constant mutation rates, and we have pairwise distances (e.g. the count of number of sequence differences between every pair of sequences).

- Can you think of a way to start reconstructing their evolutionary tree?
- Can you think of a way to use this idea over and over to reconstruct the entire tree?

Tree Reconstruction under constant mutation rates? Answer

- We should be able to identify the pair that diverged most recently by selecting the pair X,Y with **minimum distance**.
- We can replace that pair by a single "cluster sequence" C:
  - the pair's distances to any other sequence Z should be equal (i.e. $\delta(X,Z) \approx \delta(Y,Z)$).
  - so we can simply set $\delta(C,Z) = \text{mean}(\delta(X,Z), \delta(Y,Z))$.
- We then just repeat the whole procedure on the *next* pair with minimum distance.
- we keep "clustering" this way until all sequences have been joined into a single cluster.

Announcements

- NO discussion section this Friday (holiday)
- Friday Dec. 8 discussion section will be REVIEW session for the final exam.
- Final exam Wed. Dec. 7, 3-6 PM. (location will be announced this Friday)
- Practice exam problems will be posted in CCLE Week 10, so please work through them and bring your questions to the Review Session.

Your Mission: Reconstruct Evolutionary Trees

- given **modern sequences**, can you reconstruct the branching order (and times) of their evolutionary relationships?
- In this Mission Training, you will do this using two different methods.
- Principle 1: constant evolutionary rates, aka "molecular clock"
- Draw this as a picture with time as the vertical axis, modern sequences at bottom (t=0).
- minimum distance pair must be neighbors!

Ultrametric Tree Reduction

The Ultrametric phylogeny algorithm depends on a *reduction* step that replaces the selected pair of clusters with a new cluster, prior to applying the tree construction algorithm recursively on the "reduced" set of clusters.

Let's consider the details of the reduction step for the algorithm for a simple tree of four sequences X,Y,Z,W. Say sequences X,Y are chosen as nearest neighbors. Indicate precisely how they would be replaced with a new cluster vertex C, such that the assumptions of the ultrametric phylogeny algorithm are still satisfied and we can simply repeat its basic steps to find the next pair of neighbors. Specifically, state a basic principle for how the distance of C to its MRCAs with Z or W should compare with the corresponding distances for X,Y to their MRCAs with Z or W. Hint: draw a representative tree *before* and *after* replacing X,Y with C.

Ultrametric Tree Reduction Answer

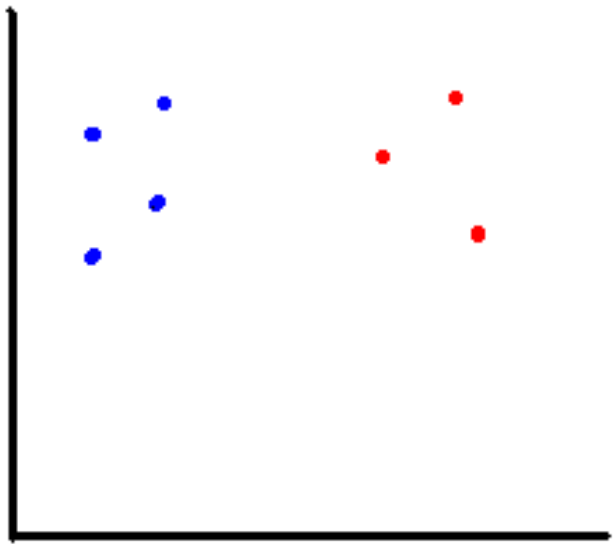
To satisfy the ultrametric phylogeny assumption, C must be the same distance from any MRCA as its other descendants are. That is, it should be the same distance from those MRCAs as X and Y were.

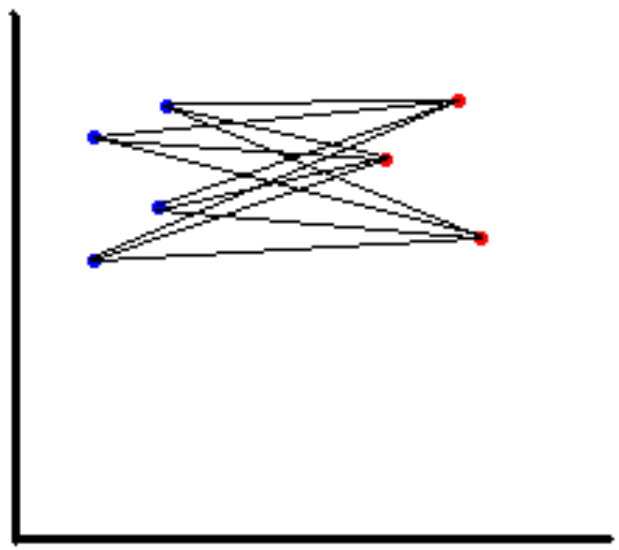
- Concretely, X,Y are replaced by a new node C whose distances to Z,W are just the averages of the corresponding distances from X and Y.
- Preserving distances in this way is vital for the correctness of subsequent clustering steps (which again look for the minimal distance pair).
- Note that C will itself (eventually) be clustered with some other cluster, to connect it to the rest of the tree.

Ultrametric Tree Reduction Errors

- some people forgot that UPGMA produces a rooted tree.
- some people forgot that rooted trees are binary trees (each node has only two children)
- some people forgot that UPGMA replaces X,Y with C at the same (average) distance to all the other sequences, as would be required for recursion on reduced tree with C.

Unweighted Pair Group Method with Arithmetic mean (UPGMA)





$$d_{GA}(G,H) = \sum_{i \in G} \sum_{i' \in H} d_{ii'} / (N_G N_H)$$

Neighbor-Join + Reduce Algorithm

We can abstract the tree construction process as an iterative cycle of two steps that takes a pairwise "distance" metric table as input:

- JOIN: use a minimization principle to identify a pair X, Y that must be **neighbors** (i.e. nodes connected by a **single** internal node) in the correct tree structure. We mark this pair as being "joined" (concretely, we add an edge from each of them to the new internal node that joins them).
- REDUCE: we reduce our pairwise-metric table by replacing the rows representing X, Y with a single row representing the merged "cluster" C.

The reduced table constitutes a new "neighbor joining" subproblem that we can solve by repeating the exact JOIN+REDUCE procedure.

What if the Mutation Rate Can Vary?

Our UPGMA tree building algorithm made use of the molecular clock assumption (i.e. identifying neighbors by *closest distance*).

- will that algorithm still work if the rate of evolution can vary on different branches? (you can picture this as stretching each edge in a tree by an arbitrary amount).

What if the Mutation Rate Can Vary? Answer

No, if we took one of the neighbor edges and stretched it, that would eventually make the true neighbor pair X,Y no longer have the minimum pairwise distance. The UPGMA algorithm would no longer be guaranteed to join true neighbors.

Sensitivity to Length of External Edges?

- If we drop the molecular clock assumption, we must expect that any edge in the tree can be stretched (or compressed) arbitrarily.
- Say somebody proposes a *neighbor metric* $\nu(X, Y)$ whose minimum value must guarantee that that pair are truly neighbors.
- Now consider the true (but unknown) length ε_X of the **external edge** joining sequence X to the rest of the tree. Can the *neighbor metric* $\nu(\cdot, \cdot)$ be sensitive to the value of ε_X, i.e. stretching ε_X could change which pair has the minimum value of $\nu(\cdot, \cdot)$?

Sensitivity to Length of External Edges? Answer

No. If changing the value of ε_X can alter which pair has the minimum value of our neighbor metric, it could again be tricked into choosing a non-neighbor pair.

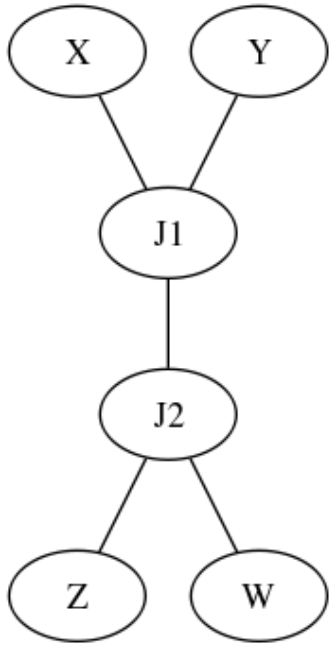
- we want $\nu(X, Y)$ to be **insensitive** to the edge lengths of *all* external edges.
- specifically, for any external edge length ε_A, we want its weight to be the same in $\nu(X, Y)$ for *all* possible pairs *X,Y*:

$$\nu(X, Y) = \ldots + \omega\varepsilon_A + \ldots \text{ for all } X, Y$$

where ω is a constant that is the *same* for all *X,Y*.

Neighbor Joining Simplest Case?

- As a concrete example, consider the following tree of four sequences, which does not obey the molecular clock assumption (i.e. edges can be stretched by any arbitrary amount).
- Assuming that the total distance between any pair $\delta(X, Y)$ is just the **sum of the lengths of the individual edges** connecting X, Y, can you propose any function of the pairwise distances that would pick out the correct neighbors?



Neighbor Joining Simplest Case? Answer

- for four sequences, choosing a tree is just a matter of choosing neighbors (e.g. (A,B) & (C,D)).
- What's special about neighbors is that they are connected solely by external edges and **no internal edges**.
- We already know we want our neighbor metric to be insensitive to (constant weight for) external edges; evidently we want it to be sensitive to internal edges (since that's what distinguishes neighbors from non-neighbors).
- So for any proposed neighbor-pairing ((X,Y) & (Z,W)) just set

$$\nu((X,Y)\&(Z,W)) = \delta(X,Y) + \delta(Z,W)$$

Since every sequence is always included exactly once, every external edge is included exactly once (hence insensitive), whereas *internal edges* will only be included if the chosen pairs are **not neighbors**.

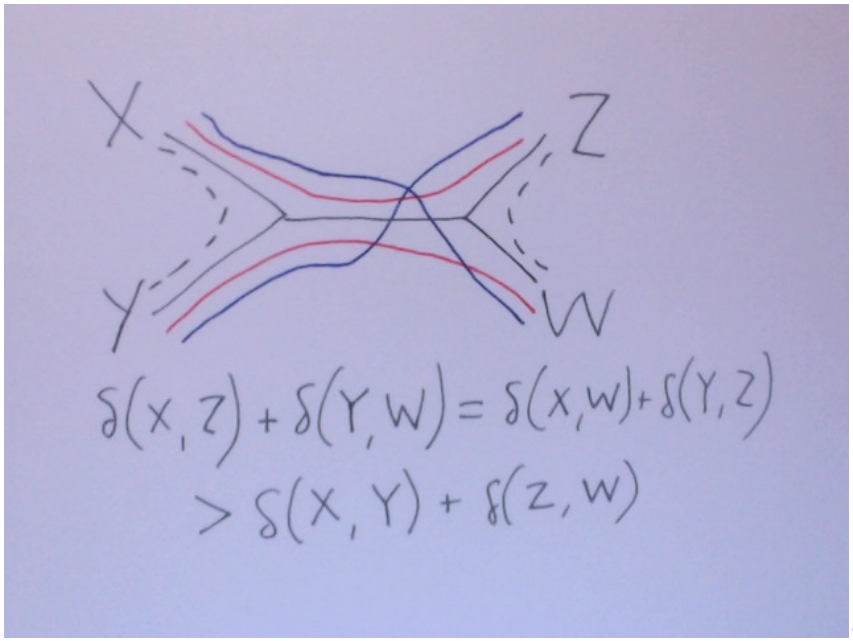
The Four Point Condition for Additive Distances

- In general for any four sequences X, Y, Z, W that obey an additive distance metric, we expect a "tall isosceles triangle":

$$\delta(X,Y)+\delta(Z,W) < \delta(X,Z)+\delta(Y,W) = \delta(X,W)+\delta(Y,Z)$$

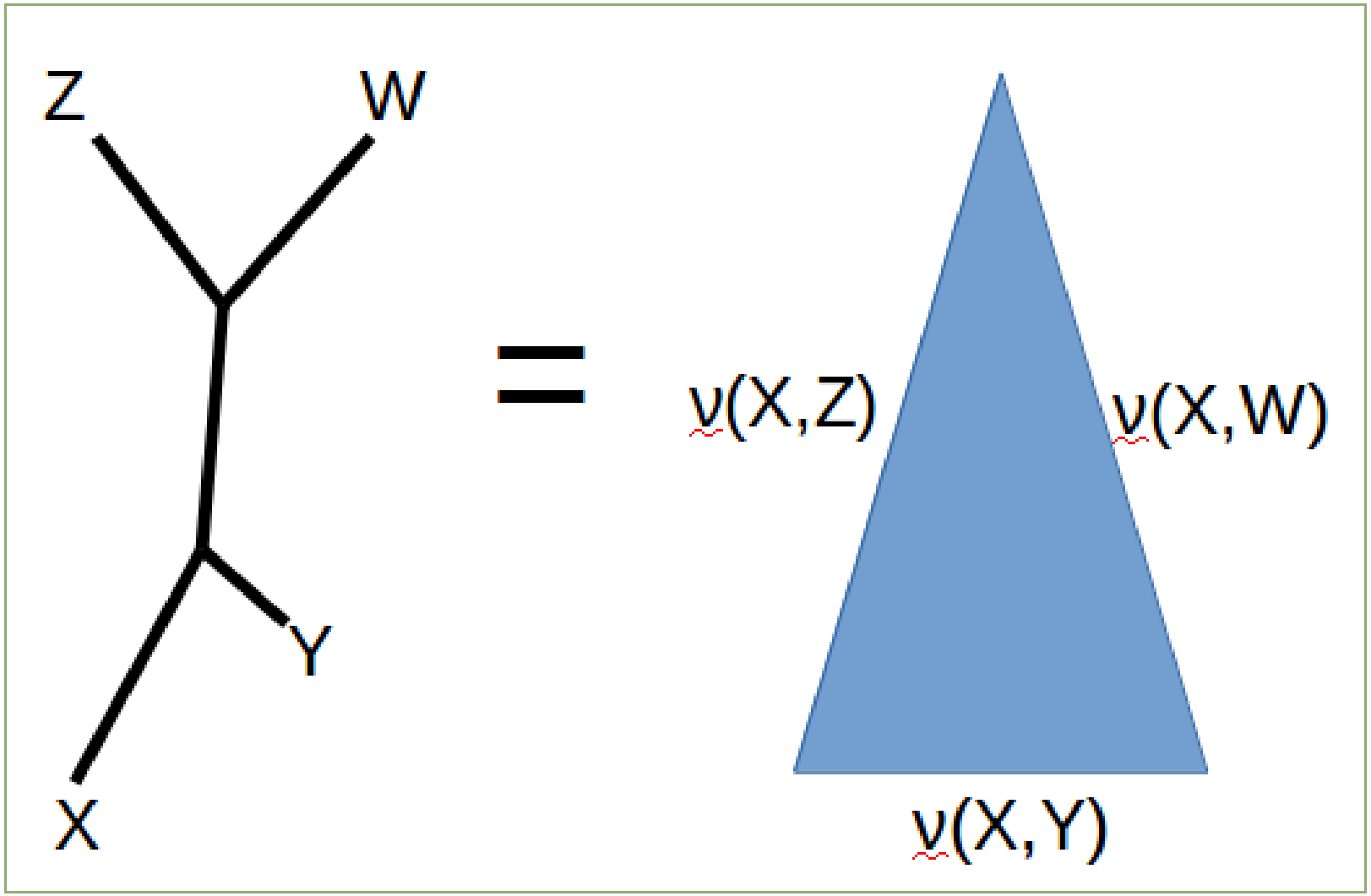
where the minimum sum represents the true neighbor pairing.

- This is called the **four point condition**.



Additive Distance "Slim Isosceles" Inequality

Easy to remember this additive distance inequality as a *slim isosceles triangle*:



Making Neighbor Joining Insensitive to External Edge Lengths

We know we want $v(X,Y)$ to weight any given external edge length ε_A equally over all possible choices of X,Y. For our candidate metric,

$$v(X,Y) = \delta(X,Y) + \omega\left(\sum_{Z\neq X}\delta(X,Z) + \sum_{Z\neq Y}\delta(Y,Z)\right)$$

is there a weight ω that accomplishes this?

Making Neighbor Joining Insensitive to External Edge Lengths Answer

- For X (or Y), the total weight of its external edge ε_X will be

$$\omega_X = 1 + \omega((N-1)+1) = 1 + N\omega$$

  where N is the total number of sequences
- For any Z (not X or Y), the total weight of its external edge ε_Z will be

$$\omega_Z = \omega(1+1) = 2\omega$$

Hence, we achieve constant weighting of external edges by setting $\omega_X = \omega_Z$, i.e. $1 + N\omega = 2\omega$, so

$$\omega = -\frac{1}{N-2}$$

The Neighbor Joining Metric = the Four Point Condition

For *N=4* sequences *X,Y,Z,W*, the Neighbor Joining Metric is equivalent to the Four Point Condition minus a constant term (which is irrelevant to our minimization):

$$\nu(X,Y) = \delta(X,Y) + \delta(Z,W) - 2\sum_i \varepsilon_i$$

where the summation is over all edge lengths in the tree.

Neighbor-Join + Reduce Algorithm

We can abstract the tree construction process as an iterative cycle of two steps that takes a pairwise "distance" metric table as input:

- JOIN: use a minimization principle to identify a pair X, Y that must be **neighbors** (i.e. nodes connected by a **single** internal node) in the correct tree structure. We mark this pair as being "joined" (concretely, we add an edge from each of them to the new internal node that joins them).
- REDUCE: we reduce our pairwise-metric table by replacing the rows representing X, Y with a single row representing the merged "cluster" C.

The reduced table constitutes a new "neighbor joining" subproblem that we can solve by repeating the exact JOIN+REDUCE procedure.

Tree Reduction

Neighbor Joining is a recursive algorithm that depends on a *reduction* step that replaces the selected pair of clusters with a new cluster, prior to applying the tree construction algorithm recursively on the "reduced" set of clusters.

Let's consider the details of the reduction step for the algorithm for a simple tree of four sequences X,Y,Z,W. Say sequences X,Y are chosen as nearest neighbors. Indicate precisely how they would be replaced with a new cluster vertex C, by drawing a representative tree *before* and *after* this replacement.

Tree Reduction Answer

- In NJ, X,Y are replaced by the vertex representing their junction (C). We must compute its distance versus all other sequences Z. We do this using additivity. Let

$$\delta(X,Z) = \alpha + \gamma;\ \delta(Y,Z) = \beta + \gamma;\ \delta(X,Y) = \alpha + \beta$$

  Then

$$\delta(C,Z) = \gamma = \frac{\delta(A,Z) + \delta(B,Z) - \delta(A,B)}{2}$$

- This leaves us with a smaller tree to infer, using the exact same procedure (i.e. neighbor joining based on finding the pair with minimal pseudodistance).
- Note that C will itself (eventually) be clustered with some other cluster, to connect it to the rest of the tree.

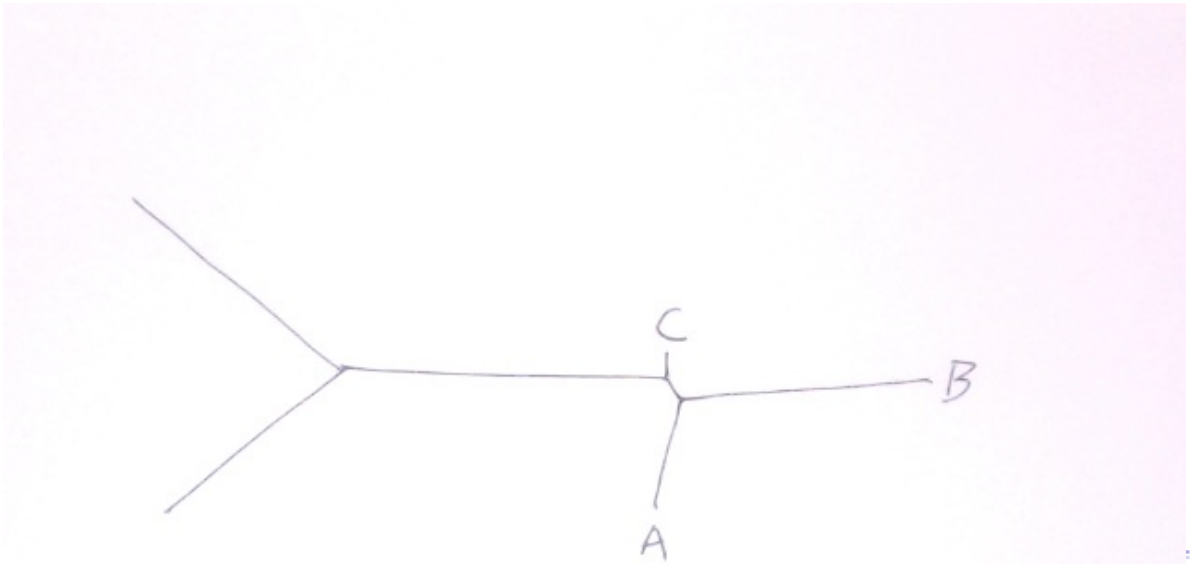
Tree Reduction Errors

- some people assumed NJ tree reduction is no different from UPGMA tree reduction. But we're trying to build a different kind of tree (unrooted/additive vs. rooted/ultrametric).
- some people assumed rooted tree, even though NJ is for unrooted trees (no ultrametric assumption).

NJ Tree Connectivity Significance?

You are testing a new tree construction program, which produced the following unrooted tree (the length of each edge in the diagram represents evolutionary distance δ). Note that sequences A,B are neighbors, even though the edge lengths indicate they are both closer to sequence C (i.e. $\delta_{AC} < \delta_{AB}$ and also $\delta_{BC} < \delta_{AB}$).

- Does this indicate that the program has made an evolutionarily incorrect tree?
- Alternatively, does the fact that the tree directly connects A and B have any valid evolutionary meaning?



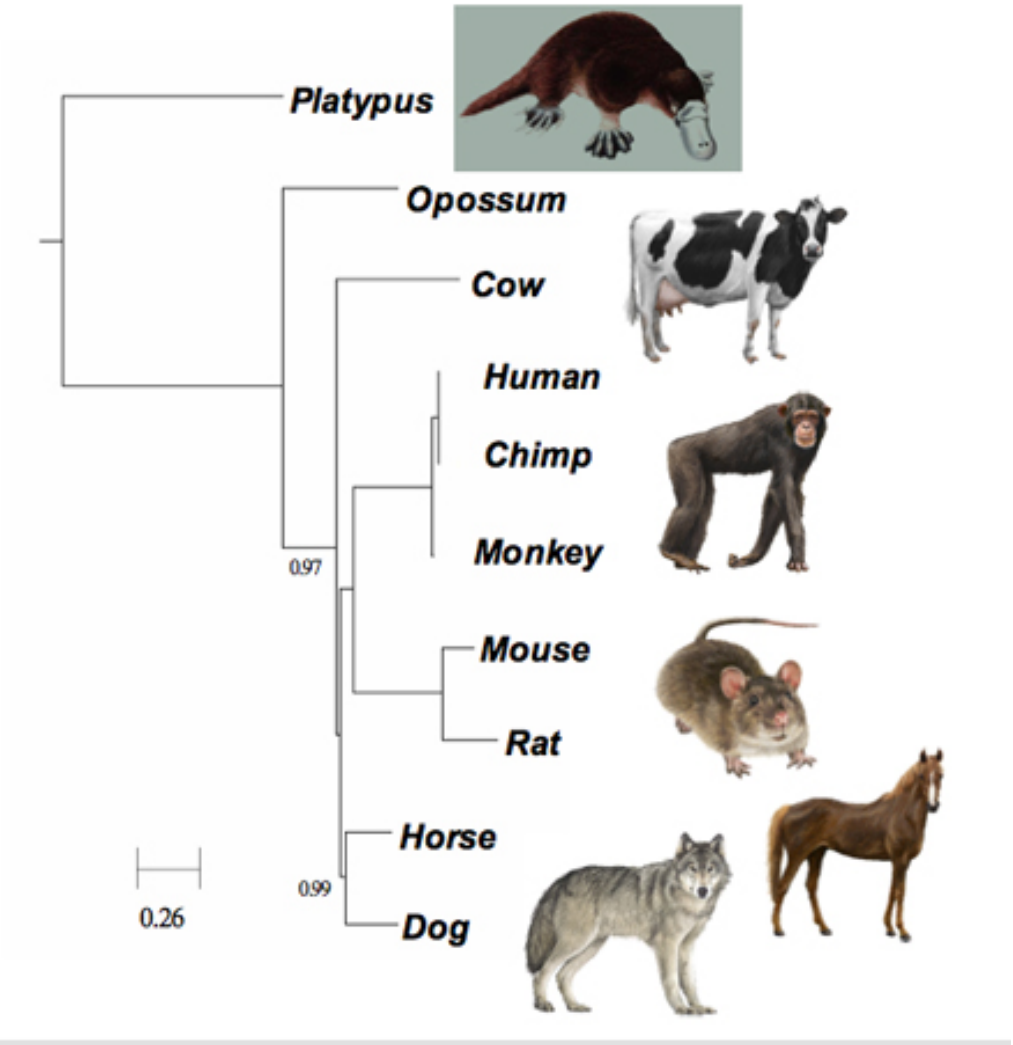
NJ Tree Connectivity Significance? Answer

- This is a valid unrooted tree. Neighbor Joining is completely insensitive to the lengths of the terminal edges. After all, the mutation rate could be different on different edges, stretching or shrinking them; we do not want that to fool the algorithm into inferring a different tree.
- The connectivity of the tree should reflect the actual evolutionary history of how they branched from each other. Specifically, we can think of the edge as a "cut" that divides the tree into two groups (A,B on one side; and all the other sequences on the other side). If the MRCA is on one side, then the other group must be *monophyletic*; if the MRCA is on the cut edge itself, then *both* groups are monophyletic.
- In other words, (A,B) is a monophyletic group unless the tree MRCA is actually on the path between them (which is usually unlikely).

NJ Tree Connectivity Significance? Errors

- some assumed that NJ should choose minimum distance pairs as neighbors. No, this forgets the "correction factor" in NJ, which makes NJ insensitive to external edge lengths.
- some asserted that "direct connection" (i.e. neighbors) has no evolutionary meaning. No, it records the actual branching history (connectivity) of the sequences.
- Some people claimed C is directly connected to A and B. No, A and B are neighbors (connected by a single internal node), whereas C is not a neighbor of either A or B.

A Whole-Genome Based Tree



(S.H. Kim & co-workers, 2009)

What Do We Want to Know from the Tree of Life?

A tree gives us a picture of two key features of the history of life:

- **branching order**: the *order* of the branch points in a tree shows who is most closely connected (most related). For example, are we more closely related to chimp or gorilla?
- **branch length**: intuitively, we'd like the length of each edge in a tree drawing to represent (be proportional to) a well-defined *distance metric* on our evolutionary process. I.e. we'd like it to be a "scale drawing" of the actual history, just the way an architect's blueprint is a scale drawing of a real building.
  Another worry: if our distance metric does not obey **additivity**, our whole neighbor-joining strategy goes out the window!

What Distance Metric(s) To Show on a Tree?

Thinking in basic terms, what measure(s) of evolutionary distance do you think would be useful to draw as the branch lengths when we look at an evolutionary tree? For any metric you propose, highlight its *purpose* (what it shows us) and main *measurement challenge*.

What Distance Metric(s) To Show on a Tree? Answer

Any of the following are good candidates:

- *evolutionary divergence time t*: this would show us the **actual history** of evolution, i.e. we could project the tree onto a fixed axis representing Time Before Present. The obvious challenge is that time is hard to estimate unless we know the precise rate of mutation per unit time μ (strictly speaking, on each edge!).
- *observed letter differences per site f*: shows *observed* distance. Disadvantage: highly non-linear vs. time, i.e. it "saturates" as it approaches its maximum value, so an increase of 0.01 means something quite different near zero vs. near the maximum value.
- *inferred mutation events per site* μt: shows us the "effective time" (proportional to evolutionary time, eliminating the non-linearity problem), free of misleading assumptions about μ.
  Disadvantage: requires some modeling assumptions.

Is Evolution a Markov Chain?

Consider the evolution of a gene sequence over time.

- do you think this process fits our definition of a Markov Chain?
- if so, explain why, and highlight any technical differences you see vs. how we have modeled Markov chains so far.
- if not, explain why not, i.e. identify what specific features of our definition are violated.

Is Evolution a Markov Chain? Answer

The core definition of a Markov chain is that the random variable representing the state of a process at a given time is conditionally independent of all previous history given its *immediate predecessor state*.

- the mutation process we've discussed has exactly that structure, i.e. the probability of the next state is conditioned only on the *current state*. So it fits the definition of a Markov chain.
- however, the assumption of **discrete time steps** no longer seems tenable here.

Continuous Time Markov Chains

Say we have a set of discrete states whose probabilities at time t are given by the state probability vector $\vec{p}(t)$. We define the transition matrix for a time duration t as $\mathbf{T}(t)$, which we can just apply in the usual linear algebra way, to compute the state probabilities after time t has elapsed:

$$\vec{p}(t) = \vec{p}(t=0)\mathbf{T}(t)$$

If the process represented by the time-dependent transition matrix $\mathbf{T}(t)$ is truly a Markov chain, what if anything can we say about the transition matrix for "adding" two time intervals $\mathbf{T}(t+\Delta t)$?

Continuous Time Markov Chains Answer

- The continuous-time version of the Markov property is that "adding" two time intervals is equivalent to just **multiplying their transition matrices**.
- i.e. we can cut time at any point, and everything *after* that time is conditionally independent of everything *before* that time, which is why we can just multiply the probabilities.

Hence

$$\mathbf{T}(t+\Delta t) = \mathbf{T}(t)\mathbf{T}(\Delta t)$$

Homogeneous Continuous Time Markov Chains

- A **homogeneous Markov process** is one that uses the same transition rates *at all times*.
- we can best characterize this "constant rate of change" by defining an **instantaneous rate matrix** Λ whose matrix elements λ_{ij} are the rates of flow from $s_i \rightarrow s_j$. I.e.

$$\frac{dp_j}{dt} = \sum_i \lambda_{ij} p_i$$

Mutation Rate Matrix Element Values?

For a *mutation process* where any given letter s_i can mutate to a different letter s_j, what if anything can you say about the range of values you'd expect

- for **off-diagonal rates** λ_{ij}, where $i \neq j$?
- for **diagonal rates** λ_{ii}?

Mutation Rate Matrix Element Values? Answer

- Due to mutation, for $i \neq j$, we expect $\lambda_{ij} > 0$, i.e. there is positive flow from nucleotide $s_i \rightarrow s_j$.
- If probability is flowing *away* from s_i in proportion to its own probability, then its probability will **decrease** at a rate proportional to its own probability, i.e. $\lambda_{ii} < 0$.
- In particular, since probability is conserved ($\vec{p}(t)$ remains normalized at all times t), the diagonal λ_{ii} and off-diagonal elements λ_{ij} in a given row i must sum to zero:

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

Note: if Λ were just a scalar value, the solution would just be an exponential of the form $e^{\lambda t}$.

Simple Mutation Model Assumptions

Some common assumptions for simple evolutionary rate models:

- Neutral: mutation only; no selection
- Reversible, also called the *detailed balance* condition

$$\pi_i \lambda_{ij} = \pi_j \lambda_{ji}$$

- Independence (Felsenstein 81): the rate of a given transition is independent of the starting point:

$$\lambda_{ij} = \pi_j \mu$$

  where μ is the mutation rate per unit time.

What would such a mutation rate matrix look like?

Simple Mutation Model Assumptions Answer

A simple example is the Felsenstein 81 model:

$$\Lambda = \mu \begin{pmatrix} -(1-\pi_1) & \pi_2 & \pi_3 & \pi_4 \\ \pi_1 & -(1-\pi_2) & \pi_3 & \pi_4 \\ \pi_1 & \pi_2 & -(1-\pi_3) & \pi_4 \\ \pi_1 & \pi_2 & \pi_3 & -(1-\pi_4) \end{pmatrix}$$

where μ is the mutation rate per unit time, and $\vec{\pi}$ is the stationary distribution.

- An even further simplification is the Jukes-Cantor model which simply asserts all four nucleotides have equal stationary probability i.e. $\pi_1 = \pi_2 = \pi_3 = \pi_4$.

Inferred mutations per site Distance Metric?

The inferred Number of Mutation Events per Site represents a useful evolutionary distance metric (e.g. for drawing branch lengths on a tree diagram; or reconstructing a tree based on distances). Say we have a constant density of mutations per unit time μ. What is the expected number (i.e. expectation value) of mutation events in a time duration t?

Inferred mutations per site Distance Metric? Answer

The Poisson distribution $p(k) = \frac{\lambda^k}{k!} e^{-\lambda}$ models a constant density of events over a continuous time interval. The expectation value for the number of events k at $\lambda = \mu t$ is just

$$E(k) = \sum_{k=0}^{\infty} kp(k) = \sum_{k=0}^{\infty} k \frac{(\mu t)^k}{k!} e^{-\mu t} = \mu t$$

Jukes-Cantor Distance vs. Observed Distance?

To get a feeling for the behavior of a simple mutation model, let's formulate

- what are the basic incoming and outgoing edges and rates for a given nucleotide s_i?
- the basic rate equation for the simple Jukes-Cantor model (reversible, independent, equal), assuming we start in a specific nucleotide $p(s_i) = 1$ at time *t=0*?
- how its distance metric relates to the *observed distance* (observed fraction of mutations / site)?

Jukes-Cantor Distance vs. Observed Distance? Answer

- state s_i has three outgoing edges and three incoming edges, all with the same rate constant $\pi\mu$.
- the other 3 states will grow at the same rate, i.e. $p_j = (1 - p_i)/3$.
- so the basic rate equation is:

$$\frac{dp_i}{dt} = -3\pi\mu p_i + \pi\mu(1 - p_i)$$

- the solution is

$$p_i(t) = \pi + (1 - \pi)e^{-\mu t}$$

- since $f = 1 - p_i(t)$ represents the expected fraction of observed mutations per site, it's useful to derive our **inferred distance** metric μt from it:

$$\mu t = -\log\left(1 - \frac{f}{1 - \pi}\right)$$

Jukes-Cantor Distance Rescaling

Note that $\log(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \cdots$, so for $f \to 0$,

$$-\log\left(1 - \frac{f}{1-\pi}\right) \to \frac{f}{1-\pi}$$

It's convenient to define the Jukes-Cantor distance metric δ so it converges to f at short time intervals, by multiplying by the denominator $1-\pi$:

$$\delta = (1-\pi)\mu t = -(1-\pi)\log\left(1 - \frac{f}{1-\pi}\right)$$

Conclusion: for any set of sequences *A,B,C...* we can compute their pairwise distances δ_{AB} from the observed mutation fractions f_{AB}.

Distance Metric Behavior

Under the Jukes-Cantor model (all nucleotides equally probable; all transitions between them equally probable), how do you expect the following distance metrics to behave as a function of evolutionary time t? (sketch a basic graph with t as the horizontal axis and distance as the vertical axis):

- *f: observed letter differences per site*
- δ: *inferred mutation events per site*

Distance Metric Behavior Answer

- f grows linearly from zero for small t.
- f follows exponential decay to $1 - \pi = 3/4$ as $t \to \infty$.
- $\delta \approx f$ for small t.
- δ grows linearly for all times t.

Distance Metric Behavior Errors

- assumed observed mutations / site just proportional to rate * time; did not consider how this turns into exponential decay at longer time scales

Molecular Clock vs. Additive

- Can the Jukes-Cantor model fit the *additive distance* assumption?
- Can the Jukes-Cantor model fit the *molecular clock* assumption?

If so, how? If not, why not?

Molecular Clock vs. Additive Answer

- Yes, it produces an additive distance metric. Let $d = \mu t$, and Λ be the JC unit-distance rate matrix. Then the transition matrix for two edges d_1, d_2 are $e^{d_1\Lambda}, e^{d_2\Lambda}$ respectively. The transition matrix for traversing both edges is

$$e^{d_3\Lambda} = e^{d_1\Lambda}e^{d_2\Lambda} = e^{(d_1+d_2)\Lambda}$$

  So the effective total distance is just $d_3 = d_1 + d_2$. The key is that JC assumes that same rate matrix on all edges; if that weren't the case, it wouldn't give this simple, additive form.
- Yes, it can also produce an ultrametric distance obeying the molecular clock hypothesis, in the case where the value of μ is the same on all edges. Then the distance is always the total time multiplied by a constant, as required by the molecular clock assumption.

Molecular Clock vs. Additive Errors

- some people intrepreted JC as implying the same transition matrix for all time intervals.
- some people interpreted mutation rate parameter as being the same on all edges, implying that JC is constrained to the ultrametric case.
- some people interpreted additive distance assumption as meaning that mutation rate different at different times. No: it means that total distance is equal to the sum of the distances over the edges between any two sequences.
- Some people did not consider how to prove that distances are guaranteed to be additive.

Measuring Random vs. Systematic Errors

- *random errors* can be broadly categorized as problems of insufficient sampling.
- Need to test whether our tree is likely to be quite different if we had a complete sample.
- We also must consider the possibility of *systematic errors* in the data / our assumptions.

Bootstrap Test: check internal agreement among multiple characters

Characters

```
Chimp       ..AGCTAAAGGGTCAGGGGAAGGGCA..
Gorilla     ..AGCATAGGGGTCAGGGGAAAGGCT..
Human       ..AGCAAAAGGGTCAGGGGAAGGGGA..
Macaque     ..AGCTCATCGGTAAGGAGAAAGGAT..
Orangutan   ..AGCCCATCGGTCAGGAGAAAGGAT..
Squirrel    ..AGCGGACCGGTAAGGAGAAAGGAC..
```

Each character yields independent information about the correct phylogeny. While it's not expected that all subsets of the characters would yield the exact same tree, we'd like to know if there are major disagreements within the set of characters. The Bootstrap Test addresses this, by *resampling* the set of characters.

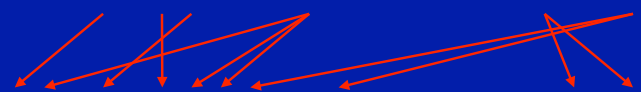
Bootstrap: make a random "sample" of characters with replacement

Characters

"original sample"

```
Chimp       ..AGCTAAAGGGTCAGGGGAAGGGCA..
Gorilla     ..AGCATAGGGGTCAGGGGAAAGGCT..
Human       ..AGCAAAAGGGTCAGGGGAAGGGGA..
Macaque     ..AGCTCATCGGTAAGGAGAAAGGAT..
Orangutan   ..AGCCCATCGGTCAGGAGAAAGGAT..
Squirrel    ..AGCGGACCGGTAAGGAGAAAGGAC..
```

Random sampling:



"new sample" is just random mix of old characters

```
Chimp       ..GACGGAACCCGGCAAAGAGGGGG..
Gorilla     ..GTCGGTGCCCGGCAAAGAGGAGAG..
Human       ..GACGGAACCGGGGAAAGAGGGGG..
Macaque     ..GCAGCCTAAAGGAAAAGAGGAGAG..
Orangutan   ..GCCGCCTCCAGGAAAAGAGGAGAG..
Squirrel    ..GGAGCGCAAAGGAAAAGAGGAGAG..
```

Sampling with replacement means each character can be sampled any number of times (0,1,2,3…).

Measuring Confidence for Individual Edges?

Say we have a set of 1000 "bootstrap trees" generated via resampling, and we want to estimate the confidence of **each edge** in the tree (even if the edges *below* it are not confident).

- Note that we do not expect the bootstrap trees to be identical.
- How would you measure the confidence of individual edges? Be specific about how your metric would measure just that edge without being sensitive to whether its "subedges" (subtrees) are confident.

Measuring Confidence for Individual Edges? Answer

- each edge is a "cut" that separates one group of sequences (e.g. the sequences below it) from the rest of the tree. So we can represent each edge by a code consisting of the unordered list of sequences in that group.
- Note this *composition code* is insensitive to the structure of that subtree.
- We can enumerate the set of edges in each bootstrap tree, generate the code for each edge, and count how many times each edge-code recurs in the set of bootstrap trees.
- The confidence of each edge is just *n/1000*, where *n* is the observed count for that edge-code.

Drawing the Confident Tree?

Say we have the set of confidence values $p(\varepsilon)$ for all edges observed in the set of bootstrap trees. We want to draw the tree of edges ε with confidence above some threshold C i.e. $p(\varepsilon) \geq C$. Propose an algorithm for extracting this tree, in terms of two issues:

- *ordering*: Assuming that you've filtered the subset of edges that meet the specified threshold, how would you "reassemble" (connect) them into a proper tree structure?
- *self-consistency*: say we wanted to ensure that the reassembled tree is self-consistent, i.e. only contains edges that could co-occur in a single bootstrap tree (do not conflict with each other). Propose a precise definition for a pair of edges to be self-consistent (as opposed to conflicting). Is there a value of the confidence threshold C that would ensure all the edges are self-consistent?

Drawing the Confident Tree? Answer

- An edge is defined by the subgroup of sequences G that it **cuts** from the rest of the tree.
- Hence the natural ordering on edges is simply **subset containment**, i.e. we connect edges $\varepsilon \rightarrow \gamma$ if

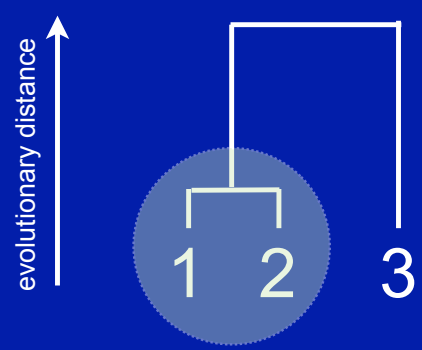
$$G_\gamma \subset G_\varepsilon$$

- Two edges conflict if they overlap in a way that is not simple containment i.e.

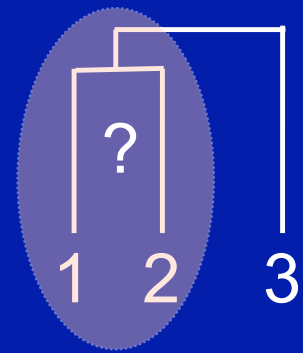
$$G_\gamma \cap G_\varepsilon \neq \emptyset \text{ and } G_\gamma \not\subset G_\varepsilon \text{ and } G_\varepsilon \not\subset G_\gamma$$

- Choosing *C>0.5* (i.e. majority) will ensure self-consistency, as the sum of of the probabilities of an edge ε and all its conflicting edges must equal 1.

Monophyletic Group Analysis



confident



uncertain

A monophyletic group is a subset of sequences that includes *all* descendants of their most recent common ancestor (MRCA). Only confident if all *other* sequences well separated from this group. Confident monophyletic groups also form a tree!

Monophy Algorithm

- Simple solution: monophyletic group reduces to *composition* (i.e. the set of children, irrespective of the subtree details)
- Represent composition as a *bitstring*, by assigning each sequence a unique bit.
- Combining two subtrees is just bitwise-OR.
- Traverse the tree with depth-first search, incrementing the count of each group found.

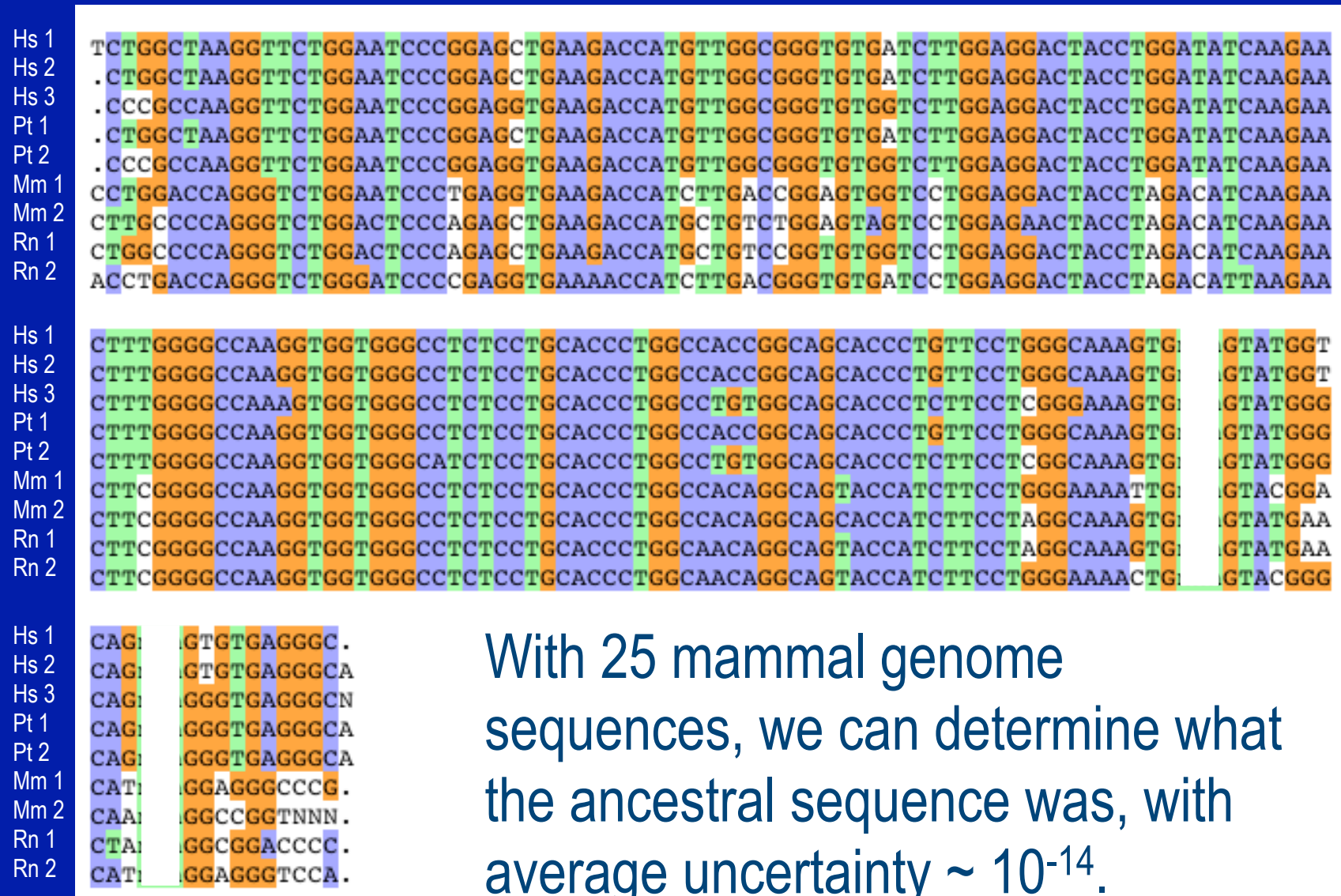
Monophy Algorithm

```
monophy(t, n): # analyze a tree or subtree
    if t is leaf: return bitcode[t]
    code1 = monophy(t->child1, n)
    code2 = monophy(t->child2, n)
    n[code1 OR code2]++
    return code1 OR code2
monophy_all(trees, n): # analyze all trees
    for tree in trees:
            monophy(tree, n)
```

Sequencing the ancestor of all mammals (100 million

- Already sequenced: human, mouse, rat, chimp, dog, cat…
- 25 mammalian genomes will be sequenced, carefully selected to give best coverage of families (i.e. choose mammals that are as unrelated to each other as possible).
- From the alignment, we can reconstruct the genome of the ancestor of all mammals!

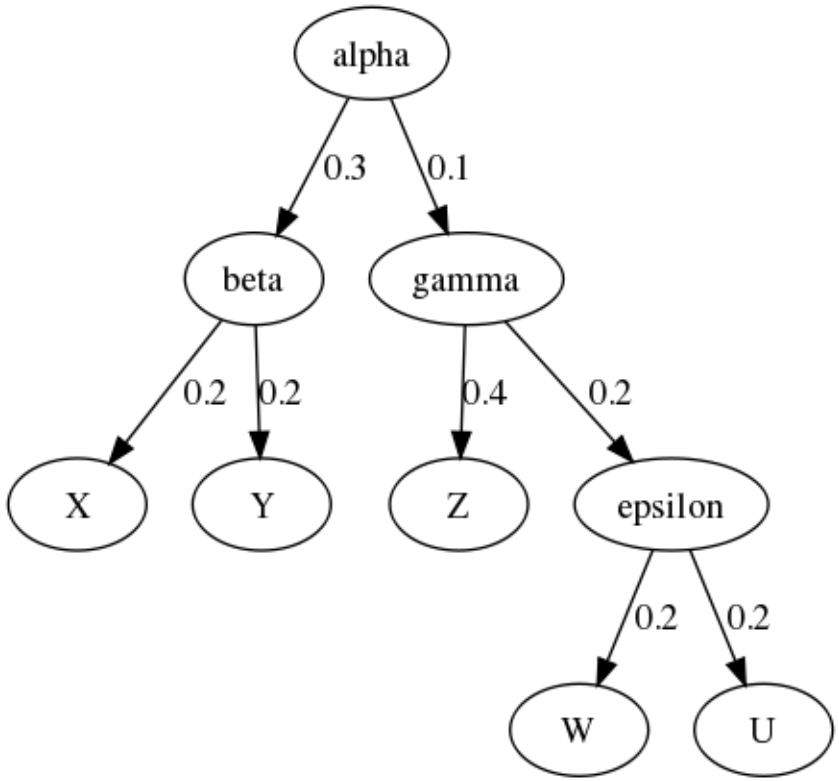
Example: Human, Chimp, Mouse, Rat



With 25 mammal genome sequences, we can determine what the ancestral sequence was, with average uncertainty ~ 10^{-14}.

Ancestral State Inference?

- Given a tree structure and edge lengths $\varepsilon = \mu t$, and observed leaf node states *X,Y,Z,W,U,...* (e.g. nucleotides);
- We would like to find the "tree traversal" (internal node states $\alpha = s_i, \beta \ldots$) that maximizes the probability of all nodes in the tree.
- What basic probabilities inputs must we be given, to solve this?



Ancestral State Inference? Answer

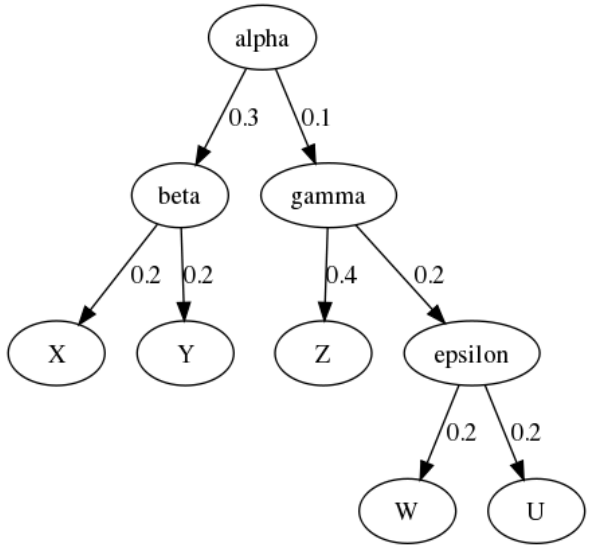
The tree of ancestral and observed (leaf node) states form a Markov chain. To compute the joint probability $p(\alpha, \beta, ..., X, Y, ...W, U)$ of the entire tree, we need the

- **transition probabilities**: $p(\beta = s_j | \alpha = s_i, \varepsilon = \mu t)$. We get these directly from our evolutionary model, e.g. Jukes-Cantor.
- **prior probabilities**: in theory, at the root of the tree we need to know the prior probability of the possible states (nucleotides). In practice, we usually just use the stationary distribution, e.g. for Jukes-Cantor, $p(\alpha = s_i) = \pi$.

Ancestral State Inference Induction?

To find the maximum likelihood traversal via the Viterbi algorithm, we need an **induction** for the ML traversal up to a given endpoint from the ML traversals up to endpoints directly connected to this endpoint.

- can you propose an induction that defines a traversal to an endpoint in terms of the traversal to endpoint(s) connected to it? Be specific about exactly what endpoint(s) you would use as the induction for endpoint $\alpha = s_i$.
- propose a Viterbi maximization for the maximum likelihood traversal for endpoint $\alpha = s_i$.



Ancestral State Inference Induction? Answer

- Define a traversal $D_{\alpha=s_i}$ as the *descendants* of endpoint $\alpha = s_i$, i.e. $\beta, \gamma, ..., X, Y, ...U$. That traversal can be defined as an induction on two sets of connected endpoints, namely D_β and D_γ.
- Let D^*_α be the traversal (set of node states) that maximizes $p(D_\alpha|\alpha)$ for a given value of α.
- We can obtain D^*_α via the following Viterbi maximization rule:

$$V(\alpha = s_i) = \max_{\beta,\gamma} V(\beta)p(\beta|\alpha = s_i, \varepsilon_\beta)V(\gamma)p(\gamma|\alpha = s_i, \varepsilon_\gamma)$$

Ancestral State Inference Traversal Path?

In addition to storing the maximum likelihood $V(\alpha)$ for a given endpoint, we also need to store the traversal path that maximized it.

- What precisely would you store the traversal path at endpoint $\alpha = s_i$?

Ancestral State Inference Traversal Path? Answer

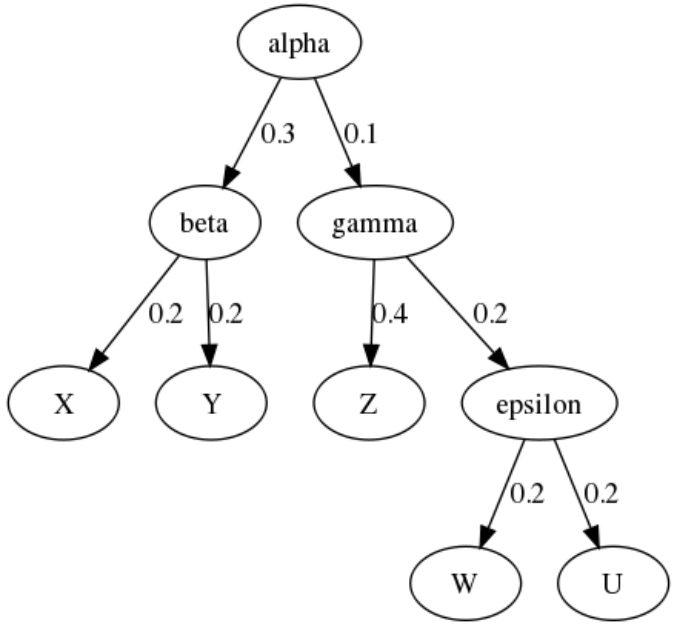
- For a given endpoint $\alpha = s_i$, we need to store the values of β, γ that maximized the Viterbi, i.e.

$$R(\alpha = s_i) = \text{argmax}_{\beta,\gamma} V(\beta) p(\beta | \alpha = s_i, \varepsilon_\beta) V(\gamma) p(\gamma | \alpha = s_i, \varepsilon_\gamma)$$

Viterbi on Trees

Differences vs. a linear HMM:

- unrooted trees are undirected graphs, whereas HMM is directed. Let's restrict our attention to rooted trees (which have directed edges).
- instead of each variable transitioning to a single "next" variable, it has two "descendant" variables in a binary tree.
- instead of each variable emiting an observation, only the "terminal" nodes (i.e. modern sequences) are observed.



Which Direction to Optimize?

- following directed edges in "forward" order, starting from the root: maximizing $p(\alpha \rightarrow \beta \rightarrow \gamma \rightarrow \ldots)$ (where α is the root) is straightforward but not meaningful (no observations on this path, so no basis for inference).
- following directed edges in "reverse" order, starting from the leaves: define the *descendants* of α as D_α (i.e. all nodes below α in the tree). Consider the probability $p(D_\alpha|\alpha)$. Since D_α always includes leaf nodes (observations), this gives us a basis for inferences about α.

Viterbi Maximization

- Define $p^*(D_\alpha|\alpha)$ as the maximum probability of the descendants D_α out of all possible values of the descendants D_α, given a specific value of the ancestor α.
- Thanks to the Markov property it can be maximized recursively.
- Say α has two child nodes β, γ. Given a specific value of α, we then find the values of β, γ that maximize the total probability of the descendants of α:

$$p^*(D_\alpha|\alpha) = \text{Max}\left[p^*(D_\beta|\beta)p(\beta|\alpha)p^*(D_\gamma|\gamma)p(\gamma|\alpha)\right]$$

Ancestral State Likelihood

Say you are given a rooted tree of a set of modern sequences $X, Y, Z, ...$ Now consider a given nucleotide site s in the aligned sequences where they differ (i.e. different nucleotides are observed). You are asked to infer the ancestral states (i.e. nucleotide) at this site in the different MRCA (ancestors) of these sequences, under the Jukes-Cantor model. Assume you are given the basic mutation rate μ and evolutionary time "length" t of each edge in the tree.

- state what hidden variables you would use, and their relation to the tree.
- state the basic form of the likelihood model you would use by specifying what you would use as the basic transition probability for an edge, and the basic form of how the transition probabilities are combined.

Ancestral State Likelihood Answer

Given a set of (hidden) ancestral nucleotides $\alpha, \beta, \gamma, \ldots$ at different internal nodes in the tree, we compute the joint probability of all the variables $p(\alpha, \beta, \gamma, \ldots, X_s, Y_s, Z_s, \ldots)$

- the transition probabilities for any edge $p(\beta|\alpha, t)$ are
  - $\tau_{ii}(t) = \pi + e^{-\mu t}(1 - \pi)$ if $\alpha = \beta$
  - $\tau_{ij}(t) = (1 - e^{-\mu t})\pi$ otherwise
- since the tree is a Markov process, the joint probability is just the product of all the conditional probabilities for the edges.
- (under Jukes-Cantor, the prior for the root α is just $p(\alpha) = \pi = 1/4$).

Ancestral State Likelihood Errors

- some people only considered ancestral state variables, forgot about time / rate variables
- some people were unclear about the likelihood model (presumably didn't know what JC model is).
- some people assumed a single ancestor rather than modeling as binary tree. Maybe if a picture of the rooted tree had been shown, wouldn't have made this mistake.
- some people assumed mutation probability just proportional to rate * time; did not consider how this turns into exponential decay at longer time scales.

Viterbi at the Leaf Nodes

Let's compute $p^*(D_\alpha|\alpha)$ for the simplest possible case, where both child nodes of α are observed (i.e. leaf nodes; call them X, Y).
Furthermore, assume that all of the variables (sequences) consist of just a single nucleotide (e.g. $X = \mathrm{A}, Y = \mathrm{G}$).
Write an equation for how you would compute the Viterbi maximum probability $p^*(D_\alpha|\alpha)$ in this case.

Viterbi at the Leaf Nodes Answer

- X, Y have no descendants, so in effect $p^*(D_X|X) = 1$.
- Furthermore, since X,Y are directly observed, only one state (the observed state) is allowed for them, so there is nothing to maximize over.

$$p^*(D_\alpha|\alpha) = p(X|\alpha)p(Y|\alpha)$$

Viterbi at the Leaf Nodes Errors

- some people forgot that X,Y were leaf-nodes, so have no descendants.
- some people forgot that X,Y were observed, so no need to consider multiple states.

Viterbi at the Root

We wish to find the values of all our variables (i.e. all internal nodes) $\alpha, \beta, \gamma, \ldots$ that maximize the total probability of the tree $p(\alpha, \beta, \gamma, \ldots, X, Y, Z, \ldots)$, where $X, Y, Z, \ldots$ are the observed sequences (leaf nodes).

- Say you are given $p^*(D_\alpha | \alpha)$ for the root node α.
- How would you find the value of α that maximizes $p(\alpha, \beta, \gamma, \ldots, X, Y, Z, \ldots)$, and the associated maximum probability $p^*(\alpha, \beta, \gamma, \ldots, X, Y, Z, \ldots)$?

Viterbi at the Root Answer

By the chain rule and Markov property,

$$p(\alpha, \beta, \gamma, ..., X, Y, Z, ...) = p(\alpha)p(\beta, \gamma, ..., X, Y, Z, ...|\alpha) = p(\alpha)p(D_\alpha|\alpha)$$

We find its maximum by finding the value of α that maximizes

$$p^*(\alpha, \beta, \gamma, ..., X, Y, Z, ...) = p(\alpha)p^*(D_\alpha|\alpha)$$

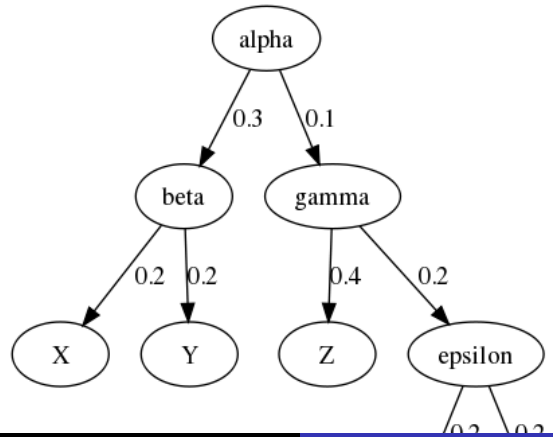
E.g. for a single nucleotide site, under the Jukes-Cantor model, $p(\alpha) = \frac{1}{4}$.

Viterbi at the Root Errors

- some people did not consider the prior $p(\alpha)$.
- some people did not use $p^*(D_\alpha|\alpha)$ given in the question.

Ancestral State "Outgroup Tie-breaker"?

Consider internal node β with descendants X,Y. If X=Y, it's easy to see that $\beta = X = Y$ will yield the biggest value of D^*_β, and thus will tend to get chosen as the Viterbi path. This matches our intuition. But what if $X \neq Y$? We can expect that $\beta = X$ and $\beta = Y$ will have equal values of D^*_β. This "tie" leaves us uncertain about which value of β is better. Intuitively, if the outgroup sequences (e.g. $Z,W,U,...$) agree with X, that should "break the tie" in favor of $\beta = X$. However D^*_β does *not* take into account $Z,W,U,...$. Does that mean our proposed Viterbi algorithm is unable to account for "outgroup tie-breakers"? If so, explain what would be required to fix this. If not, explain how the Viterbi accounts for the outgroup information.



Ancestral State "Outgroup Tie-breaker"? Answer

- D^*_β only captures the best path up to a specific endpoint (value of β).
- the Viterbi algorithm is not done until we decide what's the best *endpoint* for the whole tree, i.e. the *root node* state with maximal D^*_α.
- For $X = Z = W = U \neq Y$, that will be $\alpha = X = Z = W = U$.
- we then do "traceback" from that endpoint.
- In this case, it will backtrack through $\beta = X$.
- So our Viterbi does take into account the "outgroup tie-breaker" information.

Evaluations

- let's do the online course eval now in class
- login to **MyUCLA**
- Access your **Study List**
- Should list instructor evaluations to complete
- Also handing out informal, anonymous survey on what worked vs. didn't for you in this class.