Evolutionary Models

Edit Operators

- In standard pairwise alignment, what are the allowed “edit operators” that transform one sequence into the other?
- Describe how each of these edit operations are represented on a sequence alignment matrix.

Lessons

- Most of you got this right.
- Remember that global alignment considers the *entire length* of *both* sequences (unlike local alignment, say). So if the first letter of sequence X aligns to letter 21 of sequence Y, global alignment needs to *first* insert letters 1-20 of sequence Y before considering alignment of letter 1 of X to letter 21 of Y. Those insertions are considered by cells with X coordinate of 0.
- Another way of saying this is that global paths must start at (0,0).
- It may seem confusing that *match* and *substitution* are represented by the same move (diagonal). Just think of this move as emitting a letter for *both* sequences (which could agree or disagree).

Alignment Paths

- Using the three edit operator "moves" $\leftarrow, \swarrow, \downarrow$, write the list of all possible paths representing global alignments of $\vec{X}^2$ to $\vec{Y}^1$. (For simplicity, you can write the moves as -, /, and |).
- If any of these paths correspond to the same alignment (i.e the same letter(s) of $\vec{X}^m, \vec{Y}^n$ are aligned to each other), group those paths together as one group.

Alignment Paths Lessons

This question was designed to highlight details about exactly where global paths start.

- don't forget that global paths start at (0,0) because they must consider the possibility of inserting letters from one sequence before starting to align against the first letter of the other sequence!
- remember that different gap paths can be "degenerate" (i.e. represent the same alignment).

Viterbi Alignment

- Given the two sequences $\vec{X} = AGC\ldots$ and $\vec{Y} = ACC\ldots$
- Find the optimal global alignment of $\vec{X}^2$ to $\vec{Y}^1$ by filling out the corresponding Viterbi alignment matrix.
- Assume the following scores:
  - exact match: +5
  - substitution: -2
  - gap: -3

Viterbi Alignment Lessons

- Most of you got this right.
- a few of you just copied the substitution scores for the letters X_t, Y_u. Remember that Viterbi considers all possible moves $\leftarrow, \swarrow, \downarrow$, and chooses the best one.
- Remember that $\leftarrow, \swarrow$ and $\swarrow, \leftarrow$ are different paths and different alignments.

Alignment Scoring

- Say you are given the likelihood $p(X_t, Y_u|\text{aligned})$ derived from "true alignments" of sequences that are known to be evolutionarily related to each other.
- Based on the hypothesis testing principles we have learned, propose a scoring function for assessing whether X_t, Y_u should be aligned to each other or not.
- How does your scoring function indicate they *should* be aligned vs. *should not* be aligned?

Alignment Scoring Lessons

- Most of you got this right or very close.
- A few of you didn't think of this in probability terms, though the question emphasized that very clearly. Just remember that "hypothesis testing" should generally use either an odds ratio or a p-value.
- Two of you proposed a p-value test, which is valid but would ignore the information $p(X_t, Y_u|\text{aligned})$ that the question asked you to take into account.
- Some of you proposed the odds ratio test but were unsure what to use for the "unrelated" model. As in many previous problems we've worked, independence is the standard model that two variables are unrelated.
- It would have been fine to just propose the odds ratio (instead of its log), but in that case note that you have to *multiply* (rather than add) the individual score components.
- Many of you considered your answer "wrong" because you didn't use log, but actually the question did not require that!

Intron Alignment

Assume that you know exactly the positions that are aligned at the end of exons A, A' and at the beginning of exons B, B'.

- What would be the easiest way to obtain the optimal intron alignment subject to the proposed constraint?
- State the proposed constraint in precise terms of the allowed paths on the alignment matrix.

Intron Alignment Lessons

- Few of you explicitly made the connection to global alignment that this problem required.
- You can think of global alignment as requiring the best alignment *conditioned* on a particular endpoint constraint -- which is exactly what this problem was explicitly asking for.
- Just extracting the intron sequences for alignment is not enough; e.g. local alignment will not enforce this constraint.

Intron Alignment Lessons

- Note that constraining the endpoints of the alignment path does *not* imply there can be no gaps at the the beginning or end of the alignment! (Recall that global alignment reserves a "zero row" and a "zero column" for scoring precisely these gaps). Global alignment simply forces such gaps to be *included in the score* (whereas local alignment allows them to occur without penalty).
- Some of you made the converse error of interpreting the exon information as meaning that only diagonal moves should be allowed at the beginning / end of the introns. But this gets the constraint wrong: you were told that last part of the *exons* aligned, not that the first part of the *introns* aligned.

Profile HMM Computation

If we run a Viterbi (or Forward-Backward calculation) on this profile HMM, what will the actual calculation look like?

- draw the basic data structure that stores the state of the Viterbi algorithm during the calculation;
- indicate precisely how the different aspects of the problem (i.e. the observation sequence, the HMM states) are represented on this data structure.

Profile HMM Computation Lessons

- some of you said "I didn't understand what the question was asking". But the question was unambiguous. I suspect what caused you trouble was that it's *open-ended*, i.e. it requires you to explain in your own words how this alignment problem would be expressed as a Viterbi matrix.
- Even if you memorized the definition of a Viterbi matrix, and (separately) the definition of a profile HMM, you wouldn't necessarily be able to answer this. It required you to think about how to apply the one to the other.
- some of you thought of the HMM as having just three states M, D, I. But the question concerned a profile HMM, which has these three states for every position in the profile. Furthermore, even if you use a three-state HMM, your *Viterbi matrix* still is $O(3nm)$ because it must consider every possible letter of $\vec{X}^n$ vs. every possible letter of $\vec{Y}^m$.

Profile HMM “Moves”

- Whether we’re doing Viterbi or forward-backward, we follow our standard process.
- I.e. consider all possible moves to (from) a given cell, and write the *max* (or *sum*) of those moves in this cell.

Draw the “moves” incoming to each of the possible states in terms of how they are represented on the matrix.

Profile HMM Move Lessons

This question highlighted the correspondence between a transition to a state and a specific “move” on the matrix. I.e. the “move” depends on the state you’re transitioning *to*, not the state you’re transitioning *from*. That was all it asked for.

- some of you “regurgitated” the definition of the state graph, but the question asked you to think about how those states map to moves on the matrix. Again, this requires you to think (a bit) about how to apply the states onto the matrix.
- some of you found it hard to think about this problem as a single matrix. You’re right -- it’s much easier to think about as a separate matrix for each of the three states.
- some of you asked that I draw an example...

Probability Summation with Silent States

- Note that in previous HMM matrices we never had “vertical moves” (i.e. “silent states” that emit nothing) before.

Does this affect how we would calculate the probability of the observation sequence via the forward-backward algorithm?

- if not, why not?
- if so, how?

Lessons

- some of you made the nice suggestion of treating the emission probability of the D state as always 1. That's fine; same as the emission likelihood of an empty observation sequence.
- by contrast, another suggestion of setting $p(X_t|\theta_t = D) = 0$ is wrong. That would eliminate all paths that include D states (by assigning them zero probability).
- many of you didn't consider the problem of "path overlap" within a column, i.e. that since a path can pass through multiple cells in the same column, the probabilities in these cells are no longer disjoint.
- one of you raised the question: if a silent state emits no observation, how can we infer it from the observations? We can only infer it from the context of *surrounding alignment states*.

Total Likelihood Computation

Propose what you think would be the simplest way to compute $p(\vec{X})$ over this matrix structure.

Lessons

- many of you proposed a correct probability summation method, e.g. summing the probability exiting the last column to the END state, or summing over M and I states. Any valid sum would be considered a correct answer to this question.
- The most common error was not thinking about how to get a disjoint set of probabilities that are valid to sum. Just summing a column is no good, as we had just discussed in the previous problem.
- Easy answer: use START or END, because all the paths converge there, so no need to sum multiple cells.
- Another valid answer: use a set of states that are guaranteed disjoint (no overlap) and complete (include all possible paths). For example, a global alignment path must emit any given letter X_t from exactly one M_u state or I_u state.
- Some of you asked to see an example...

Remembering vs. Thinking

Once again, some comments are asking why I'm asking you questions I have not already shown you the answers for.

- If I first show you the answer, then re-ask the same question (with a few numbers changed), I am just testing your ability to *remember* the answer, not your ability to use the concepts (i.e. think) for yourself.
- Real life is not going to just ask you the same questions I put in your homework or lecture notes. It's going to challenge you with somewhat different questions, and you're going to need to *think* to figure out how to use the concepts you know to solve the new question.
- I'm happy to show you how to solve each question (in as much detail as you need to understand it) *after* you try doing it yourself.
- But bear in mind that I (and real life) will continue to ask you new questions. So memorization and plug-and-chug will not be enough.

Forward Computation

- Draw the move(s) that must be taken into account for computing the forward probability f_{t,M_u} (i.e. for aligning letter X_t to profile letter M_u).
- Write the corresponding recursion equation for f_{t,M_u}.

Forward Computation Lessons

- most of you mistakenly thought of the move direction as depending on the origin state. As we emphasized before, it depends only on the *destination state*. Since the destination was a match state in this question, all incoming moves are *diagonal*.
- You can think of this as the M state "consuming" one letter of X and one letter of the profile. So it has to advance both t and u indices.
- You can think of t,u as indicating that $\vec{X}^t$, $\vec{Y}^u$ have *already* been emitted on the path up to (and including) cell (t,u).
- Note that the forward probability's emission term comes from the *destination state* (in this case, M), so that term must be included even if the *origin state* is D.
- All of these confusions seem like the same thing to me. Just remember that the emission "move" is driven by the *destination* state.
- Easy case to remember: START $\rightarrow M_1$, D_1 or I_0.

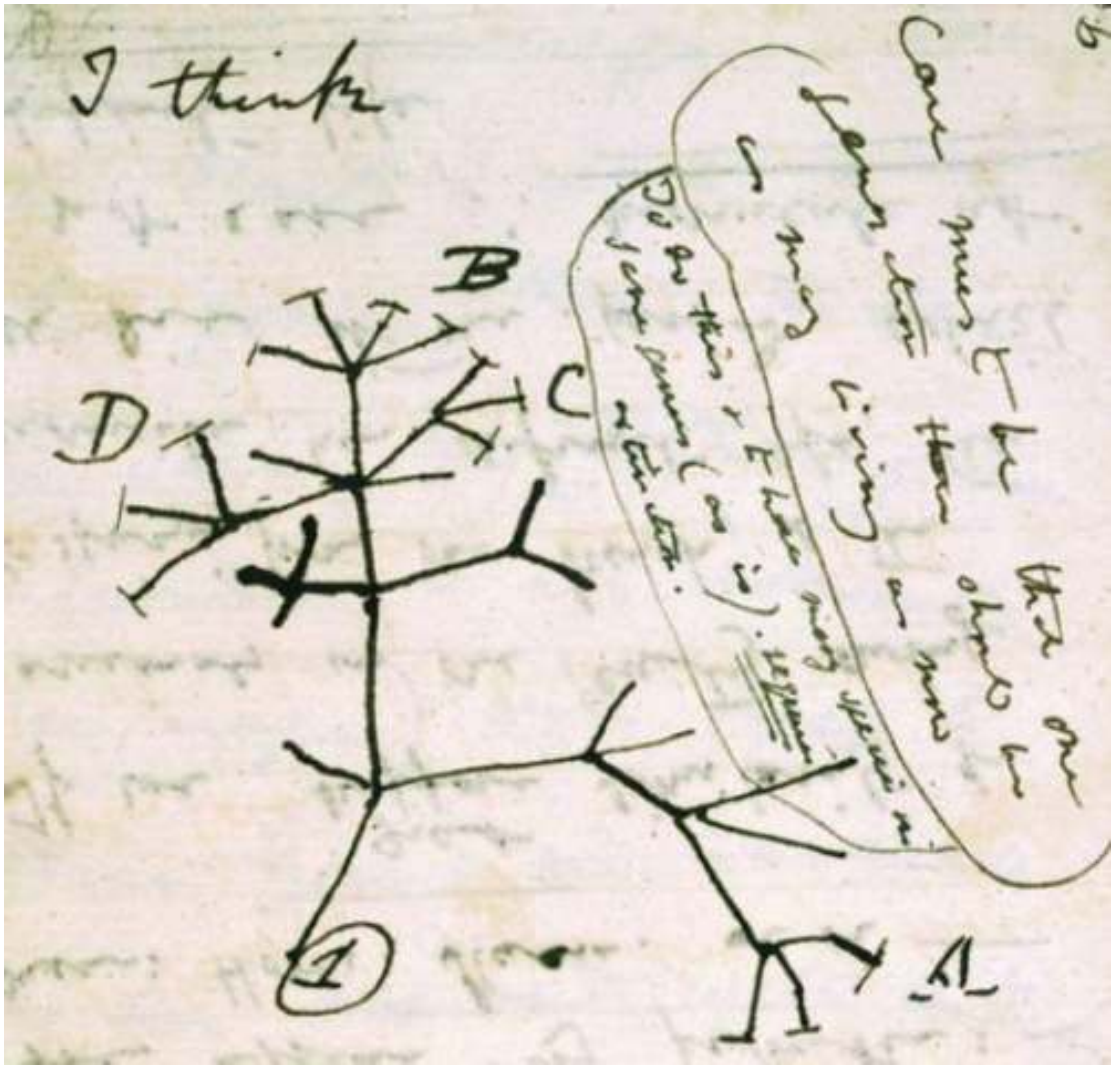
Backward Computation

- Draw the move(s) that must be taken into account for computing the backward probability b_{t,M_u} (i.e. for aligning letter X_t to profile letter M_u).
- Write the corresponding recursion equation for b_{t,M_u}.

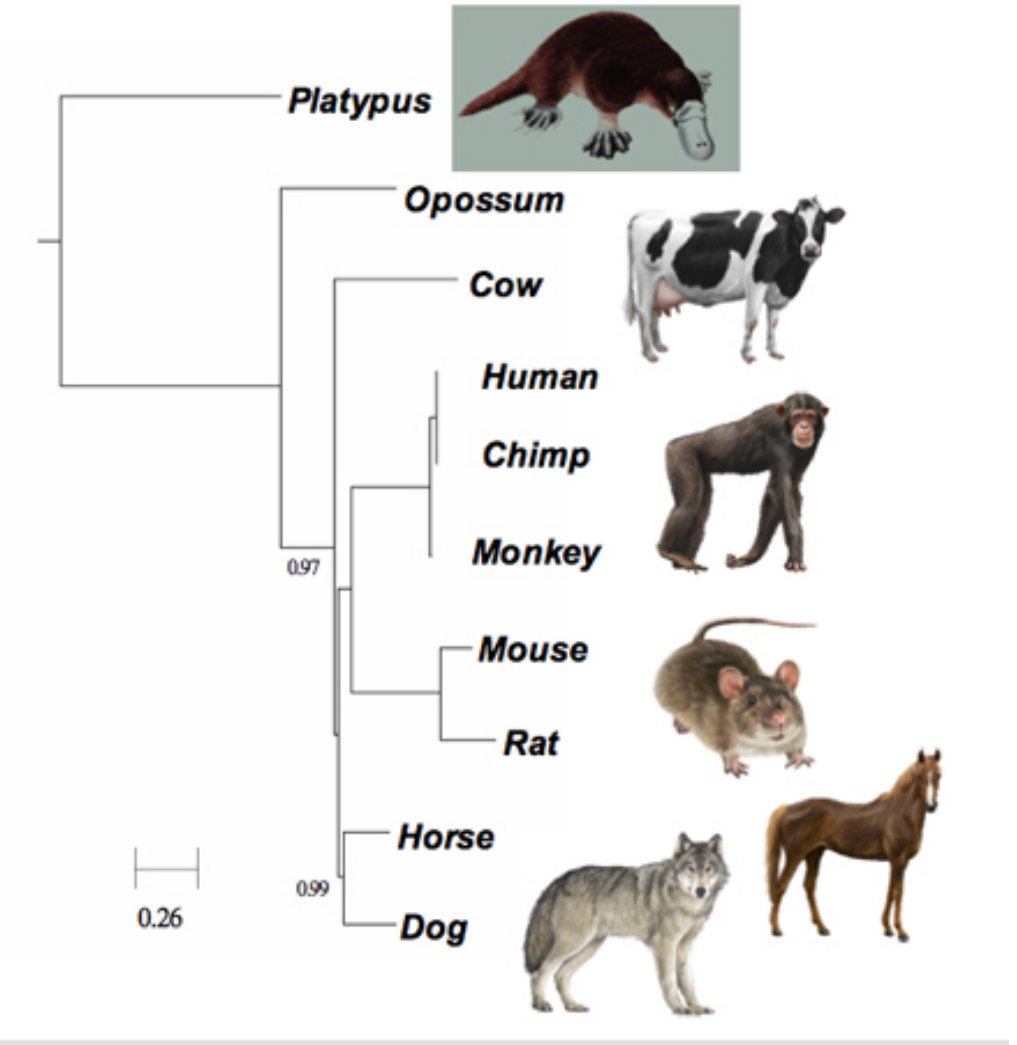
Backward Computation Lessons

- It's tempting to think of the backward probability as the exact "mirror image" of the forward probability. But we have to be careful about the details.
- For example, the backward probability sums over the possible destination states, so each of those are *different* moves (e.g. vertical for the D state, horizontal for the I state, etc.)
- Let's draw the calculation on an example...

Darwin Draws the First Phylogeny, 1838



A Whole-Genome Based Tree



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

How to model gene evolution?

atggggctcagcgacggggagtggcagcaggtgctgaacgtctgggggaa

atggggctcagtgatggggagtggcagatggtgctgaacatctgggggaa

atggctgatcatgatctggttctgaagtgctggggagccgtggaggccga

atggctaactatgacatggttctgcagtgctgggggccagtggaggctga

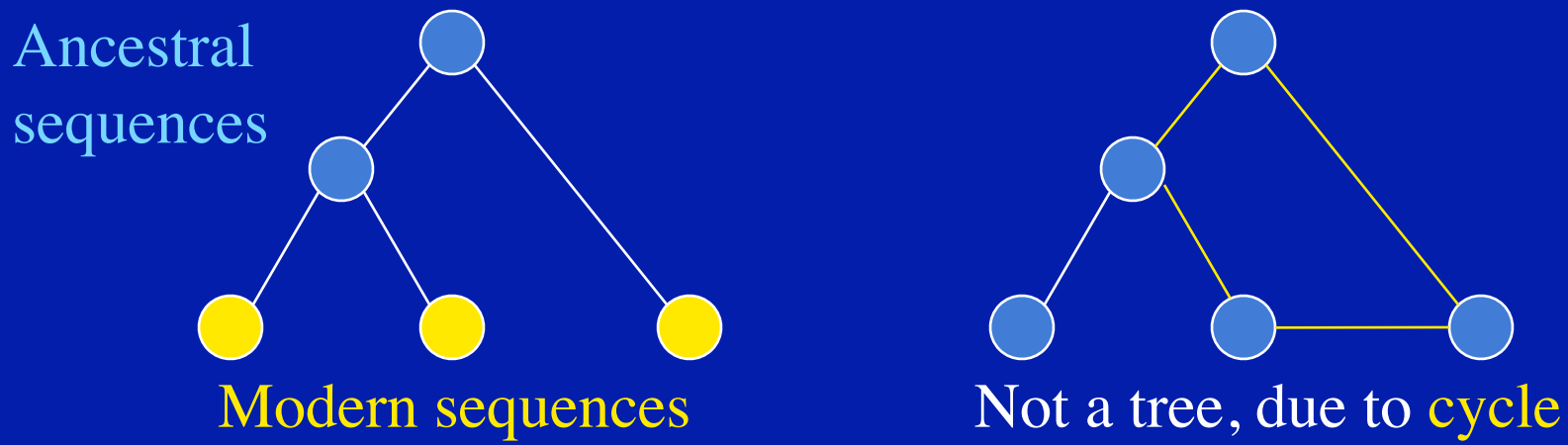
Evolution as a Markov Chain?

Presumably, evolution from a given ancestor A depends only on sequence of A, not on *its* ancestors.

But discrete time assumption no longer tenable.

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.



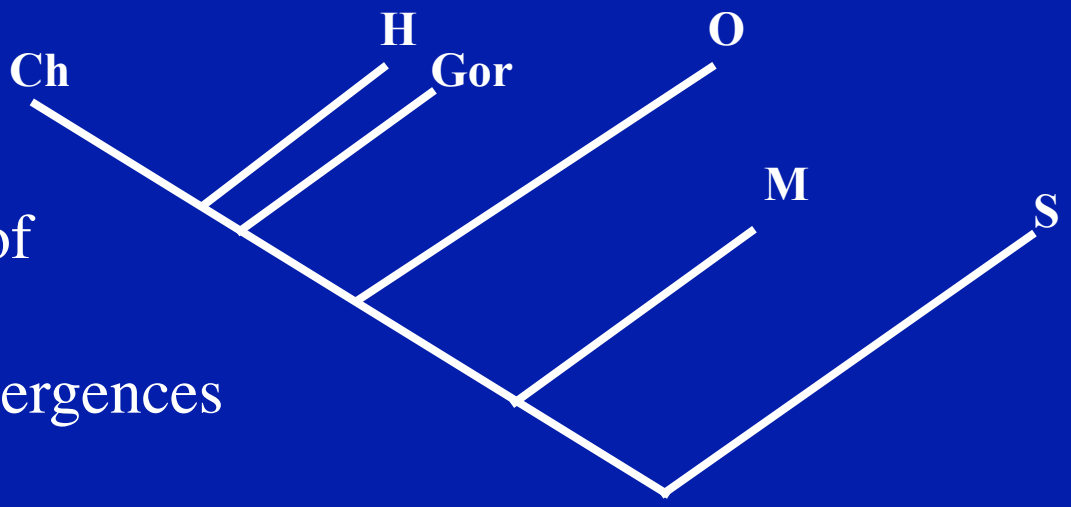
Character Differences → Branch Lengths and Order

Characters

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

Phylogenetic Tree

- Length of a branch is proportional to number of differences
- Tree shows order of divergences



Branch length represents the extent of change - expected number of substitutions per nucleotide site. The longer the branch the more change

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.

Character Differences → Sequence Distances

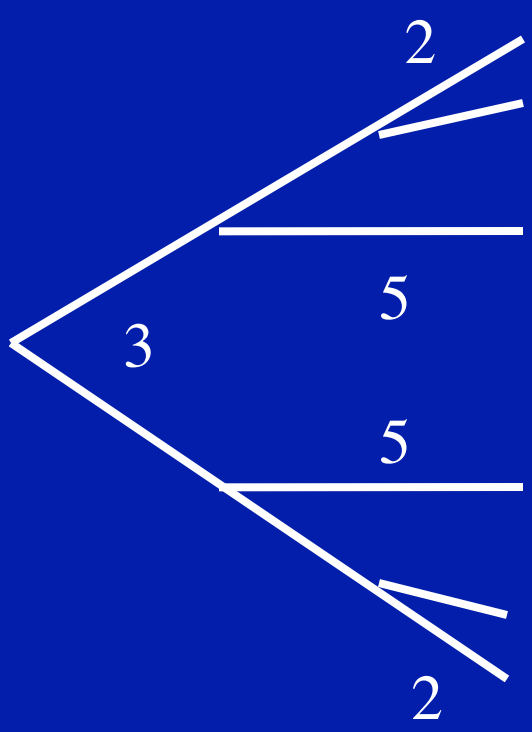
Characters

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

Distances

	Chimp	Gor	Hum	Mac	Orang	Sq
Chimp		5	2	8	8	9
Gor			5	7	6	8
Hum				9	8	9
Mac					2	4
Orang						5

Sequence Distances → Branch Lengths and Order




```
Chimp      ..AGCTAAAGGGTCAGGGGAAGGGCA..
Human      ..AGCAAAAGGGTCAGGGGAAGGGGA..

Gorilla    ..AGCATAGGGGTCAGGGGAAAGGCT..


Squirrel   ..AGCGGACCGGTAAGGAGAAAGGAC..

Macaque    ..AGCTCATCGGTAAGGAGAAAGGAT..
Orangutan  ..AGCCCATCGGTCAGGAGAAAGGAT..
```

Continuous Time Markov Chains

For a homogeneous process; define state probability vector $\vec{\pi}(t)$ and transition matrix at time t as $\mathbf{T}(\mathbf{t})$:

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

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

Define instantaneous rate matrix Λ:

$$\Lambda = \lim_{\Delta t \to 0} \frac{\mathbf{T}(\Delta t) - \mathbf{I}}{\Delta t}$$

Matrix Exponential

We can then calculate $\mathbf{T}(\mathbf{t})$ as

$$\mathbf{T}(t) = e^{\Lambda t}$$

where for a square matrix M

$$e^{\mathbf{M}} = \mathbf{I} + \mathbf{M} + \frac{\mathbf{M}^2}{2!} + \ldots + \frac{\mathbf{M}^i}{i!} + \ldots = \sum_{i=0}^{\infty} \frac{\mathbf{M}^i}{i!}$$

Elegant, but not easy to calculate generally...

Simple Mutation Model Assumptions

Neutral: mutation only; no selection

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

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

F81 Model (Felsenstein)

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

For $j \neq i$: $\tau_{ij}(t) = (1 - e^{-\mu t})\pi_j$

$$\tau_{ii}(t) = e^{-\mu t} + (1 - e^{-\mu t})\pi_i$$

Jukes-Cantor: assume $\pi_i = \frac{1}{4}$

Ancestral State Likelihood

Say you are given a rooted tree of a set of modern sequences $X, Y, Z, \ldots$. 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.

- 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
  - $\tau_{ii}\ t \quad \pi \quad e^{-\lambda t}\ 1 - \pi$ if $\alpha \quad \beta$
  - $\tau_{ij}\ t \quad 1 - e^{-\lambda t}\ \pi$ 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 $p\ \alpha \quad \pi \quad 1/4$).