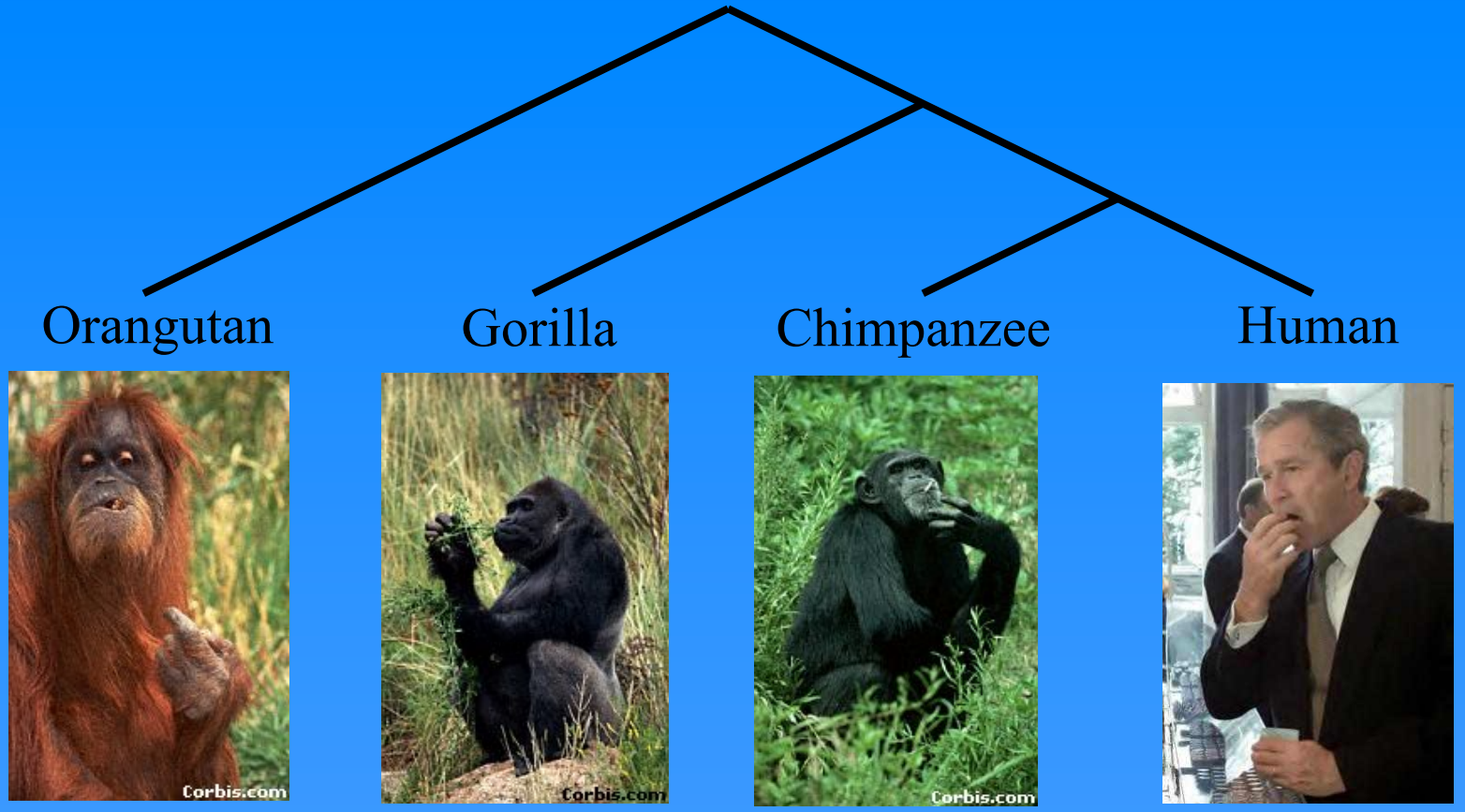
Introduction to Phylogenetics III



From the Tree of the Life Website, University of Arizona

Sagi Snir

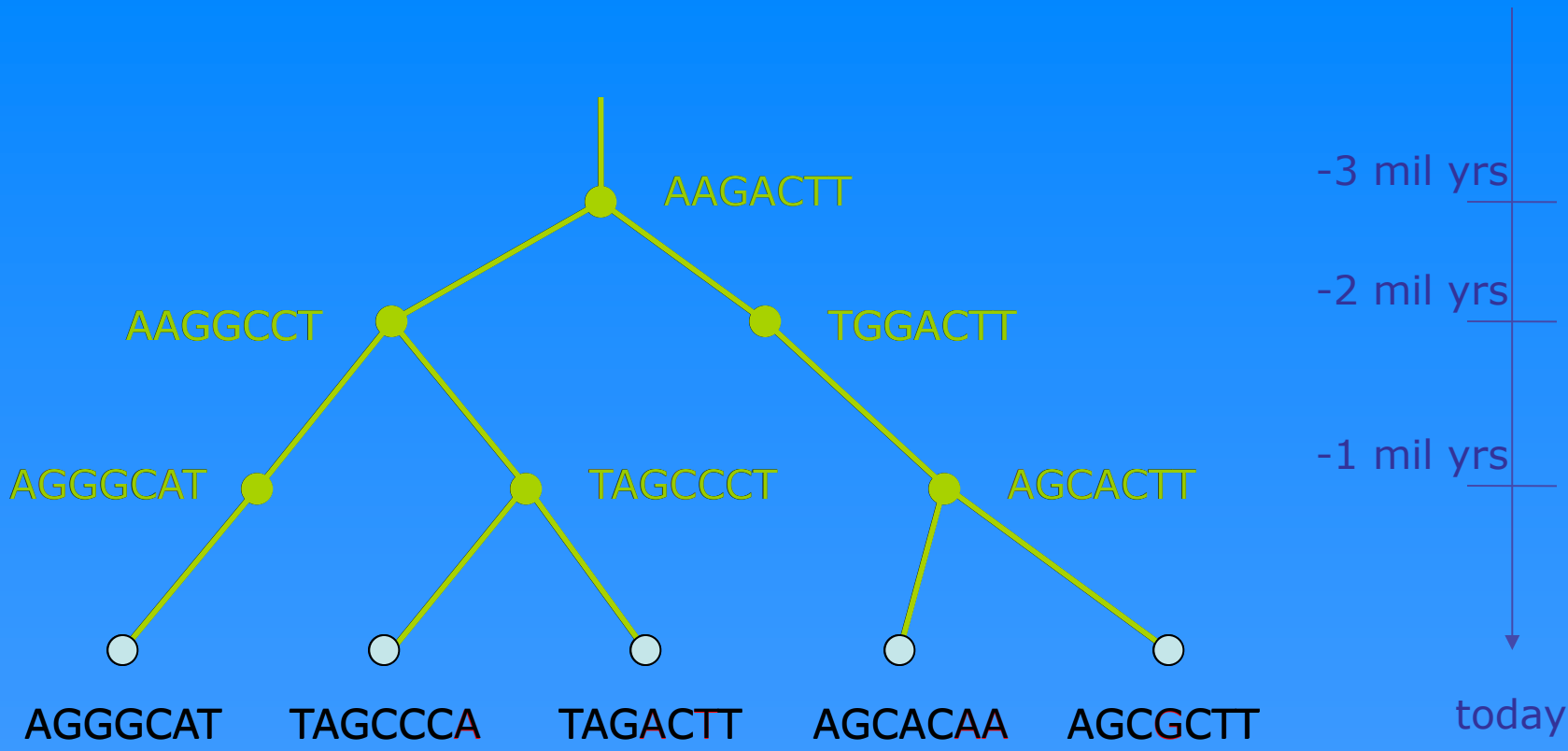
Dept. of Evol. Env. Biol. and The Inst. of Evolution,

University of Haifa

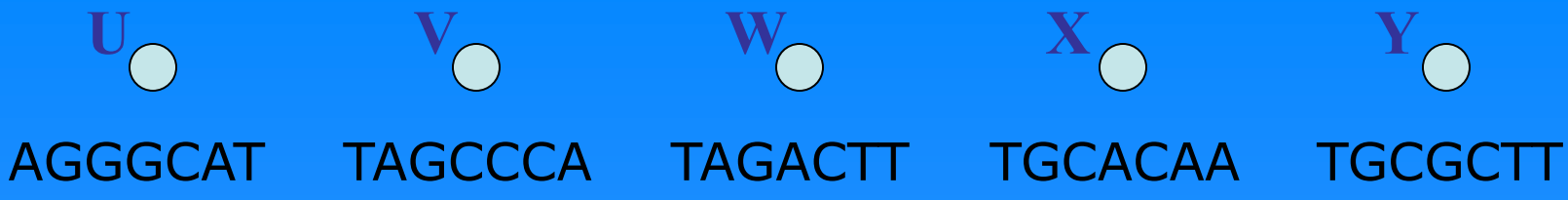
Introduction to Phylogenetics
Distance Based Methods – Neighbor Joining

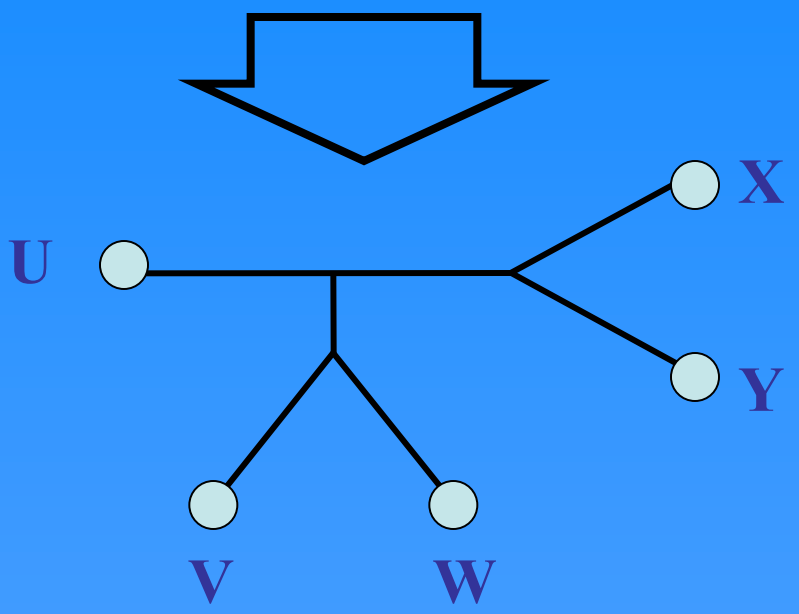
- Sequence based methods:
  - Two main categories:
    - *Character based methods*: Trees are constructed by comparing the characters of the corresponding sequences. Characters are mainly molecular (nucleotides in homologous DNA).
    - *Distance based methods*: Input is a square symmetric distance matrix. Seeks trees (edge-weighted) best-describing these distances.
- Supertree methods:
  - Construct small (reliable) trees form any data and combine it to a complete tree by combinatorial algorithms.
  - Quartet based methods.

Sequence Evolution
(substantially simplified)



Reconstructing the Tree





Unrooted trees!

Distance-based Methods for Constructing Phylogenies

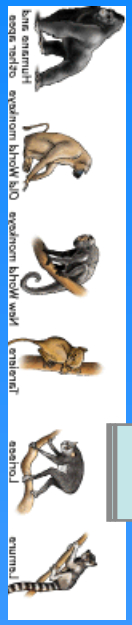
This approach attempts to overcome the two weaknesses of maximum parsimony:

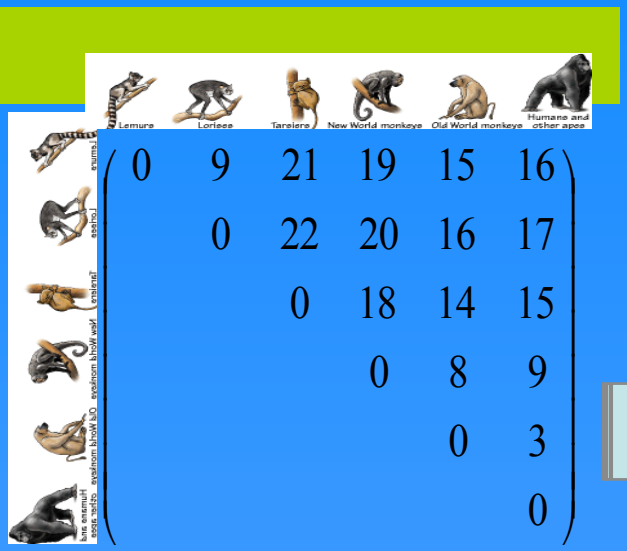
1. It start by estimating inter-taxa distances from a well defined statistical model of evolution (distances correspond to probability of changes)
2. It provides efficient algorithms for the big problem.

Basic idea: The differences between species (usually represented by sequences of characters) are transformed to numerical distances, and an ***edge weighted tree realizing these distances*** is constructed.

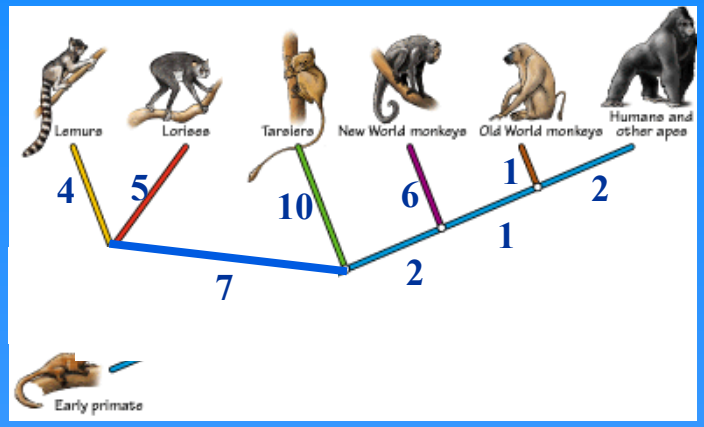
Distance-Based Reconstruction

- Compute distances between all taxon-pairs
- Find a tree (edge-weighted) best-describing the distances





$$D = \begin{pmatrix} 0 & 9 & 21 & 19 & 15 & 16 \\ & 0 & 22 & 20 & 16 & 17 \\ & & 0 & 18 & 14 & 15 \\ & & & 0 & 8 & 9 \\ & & & & 0 & 3 \\ & & & & & 0 \end{pmatrix}$$



Distance-based methods for constructing phylogenies

Common issues:

•Evolutionary model: molecular clocks vs. variable rates of evolution

• Algorithms for exact distances: do not handle real data.

• Algorithms for noisy distances.

Tree Metric (aka Additive Distances)

A ***distance metric*** on a set M of L objects is a function

$$d : M \times M \rightarrow R^{+}$$

(represented by a symmetric matrix) satisfying:

- $d(i,i)=0$, and for $i \neq j$, $d(i,j)>0$
- $d(i,j)=d(j,i)$.
- For all i,j,k it holds that $d(i,k) \leq d(i,j)+d(j,k)$.

If there is a weighted tree which ***realizes*** these distances, then the distance form a tree-metric.

Additive Distances

Definition: A distance metric d is **additive** if there is a tree T with positive weights on the edges, such that for all i,j, $d(i,j) = d_T(i,j)$, the length of the path from i to j in T.

Related topics:

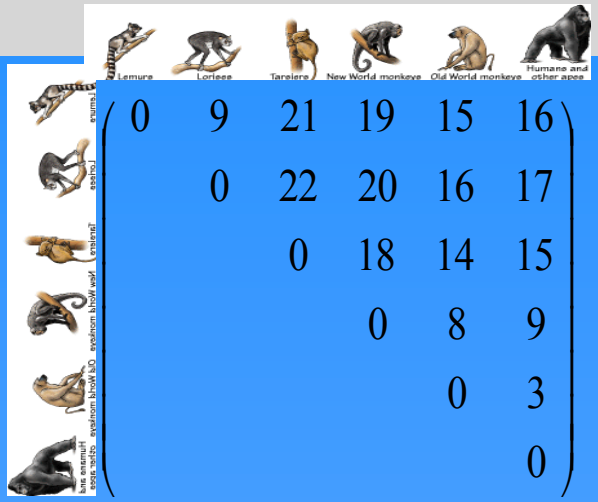
1. Characterize the additive metrics.
2. Given additive metric, construct a tree which realizes its distances.
3. Given a non-additive metric, construct a tree which "approximates" it

We'll start with 2 and then discuss 1.

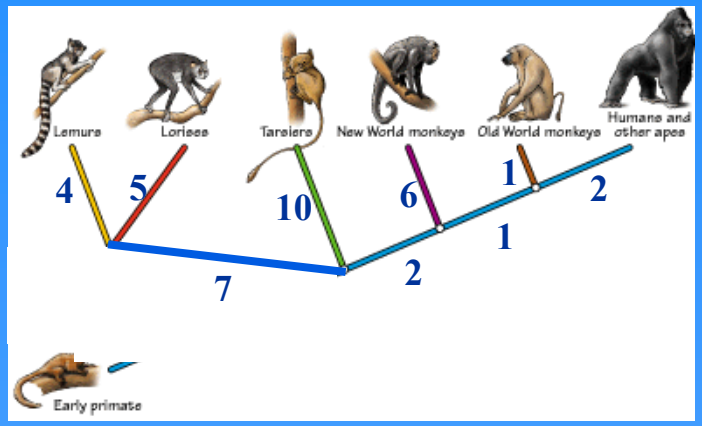
The Reconstruction Task

- Input: a Distance matrix D.
- Output: If D is additive, return a tree which realize its distances.

$D =$



$$\begin{pmatrix} 0 & 9 & 21 & 19 & 15 & 16 \\ & 0 & 22 & 20 & 16 & 17 \\ & & 0 & 18 & 14 & 15 \\ & & & 0 & 8 & 9 \\ & & & & 0 & 3 \\ & & & & & 0 \end{pmatrix}$$



$= T$

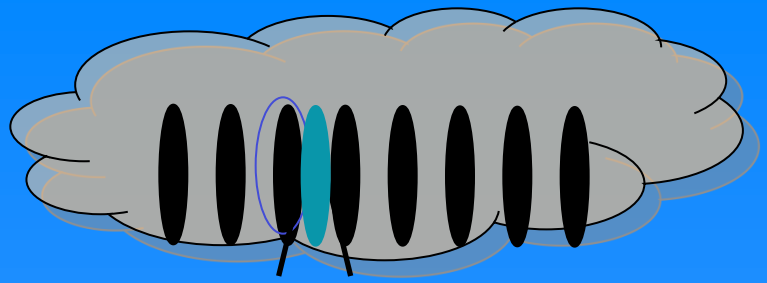
Requirement from Distance-based Tree-Reconstruction Algorithms

1. Consistency: If the input metric is **additive**, i.e. fits a tree metric, the returned tree should be the (unique) tree which fits this metric.
2. Efficiency: poly-time, preferably no more than $O(n^3)$ (as opposed to MaxPars that is exponential)
3. Robustness: if the input matrix is "close" to additive, the algorithm should return the correct tree. We distinguish between
   - Robust in theory
   - Robust in practice (eg in simulations)

A natural family of algorithms which satisfy 1 and 2 is called "Neighbor Joining".

The Neighbor Joining Tree-Reconstruction Scheme

Start with n singletons, and each iteration join two neighboring leaves (**cherries**):



- Select pair i,j and replace them by a new vertex v
- Make v the parent of the cherries i,j
- Remove i,j and insert v to the distance matrix
- Method recursively applied on reduced matrix

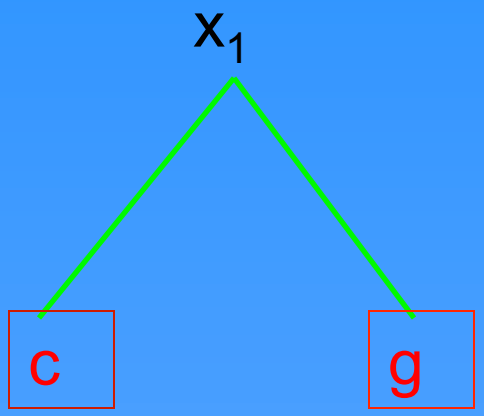
Two issues:

- How do we find i,j which are indeed cherries?
- How do we compute distances from the new vertex v?

Recursive NJ

- Selected c and g.
- A cherry on c and g in the tree is created.

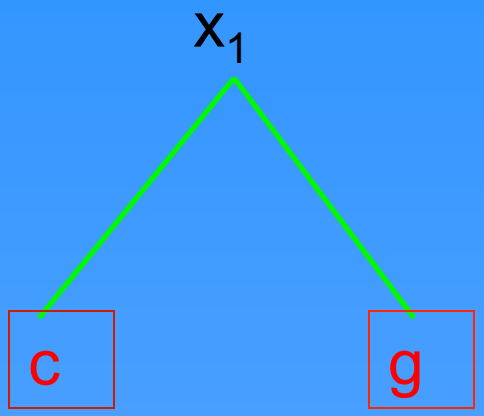
	a	b	c	d	e	f	g	h	i
a	x	x	x	x	x	x	x	x	x
b	x	x	x	x	x	x	x	x	x
c	x	x	x	x	x	x	x	x	x
d	x	x	x	x	x	x	x	x	x
e	x	x	x	x	x	x	x	x	x
f	x	x	x	x	x	x	x	x	x
g	x	x	x	x	x	x	x	x	x
h	x	x	x	x	x	x	x	x	x
i	x	x	x	x	x	x	x	x	x



Recursive NJ

- Selected c and g.
- A cherry on c and g in the tree is created.
- Rows and columns of c and g are removed.

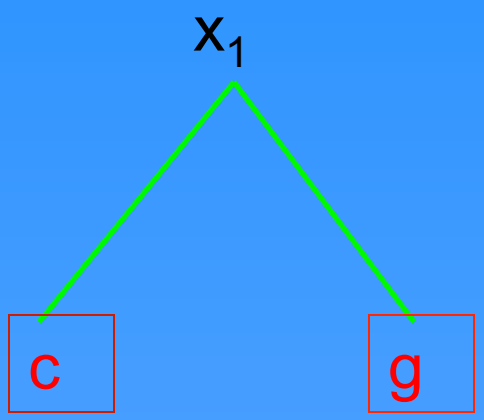
	a	b	c	d	e	f	g	h	i
a	x	x		x	x	x		x	x
b	x	x		x	x	x		x	x
d	x	x		x	x	x		x	x
e	x	x		x	x	x		x	x
f	x	x		x	x	x		x	x
h	x	x		x	x	x		x	x
i	x	x		x	x	x		x	x



Recursive NJ

- Selected c and g.
- A cherry on c and g in the tree is created.
- Rows and columns of c and g are removed.
- Distances from x_1 to c and g (in the tree) are computed.
- Distances from x_1 to all taxa still in the table are computed.

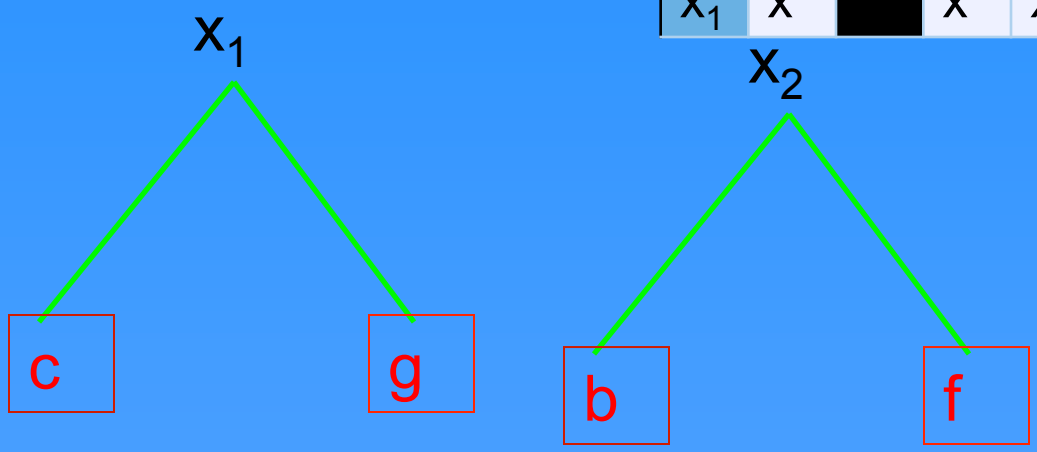
	a	b	d	e	f	h	i	x_1
a	x	x	x	x	x	x	x	x
b	x	x	x	x	x	x	x	x
d	x	x	x	x	x	x	x	x
e	x	x	x	x	x	x	x	x
f	x	x	x	x	x	x	x	x
h	x	x	x	x	x	x	x	x
i	x	x	x	x	x	x	x	x
x_1	x	x	x	x	x	x	x	x



Recursive NJ

- Selected b and f.
- A cherry on b and f in the tree is created.
- Rows and columns of b and f are removed.
- Distances from x_2 to b and f (in the tree) are computed.
- Distances from x_2 to all taxa still in the table are computed.

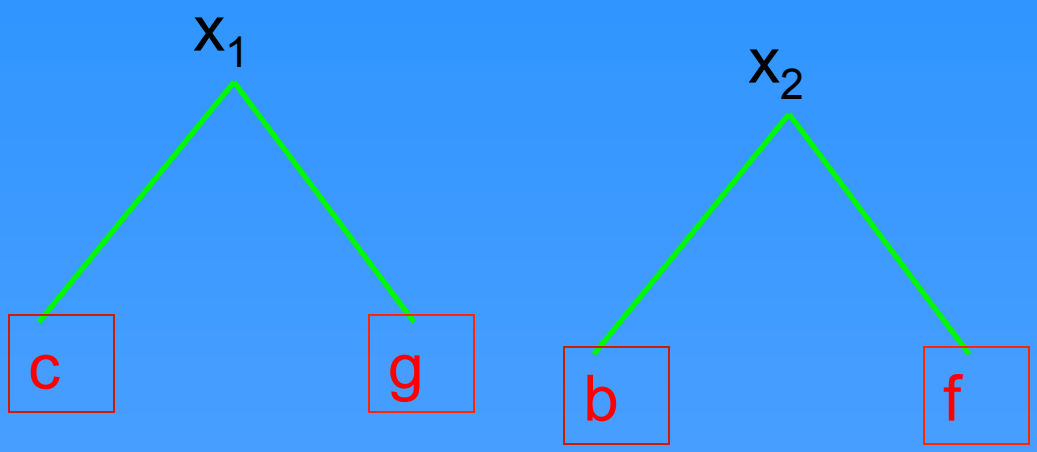
	a	b	d	e	f	h	i	x_1
a	x		x	x		x	x	x
d	x		x	x		x	x	x
e	x		x	x		x	x	x
h	x		x	x		x	x	x
i	x		x	x		x	x	x
x_1	x		x	x		x	x	x



Recursive NJ

- Selected b and f.
- A cherry on b and f in the tree is created.
- Rows and columns of b and f are removed.
- Distances from x_2 to b and f (in the tree) are computed.
- Distances from x_2 to all taxa still in the table are computed.

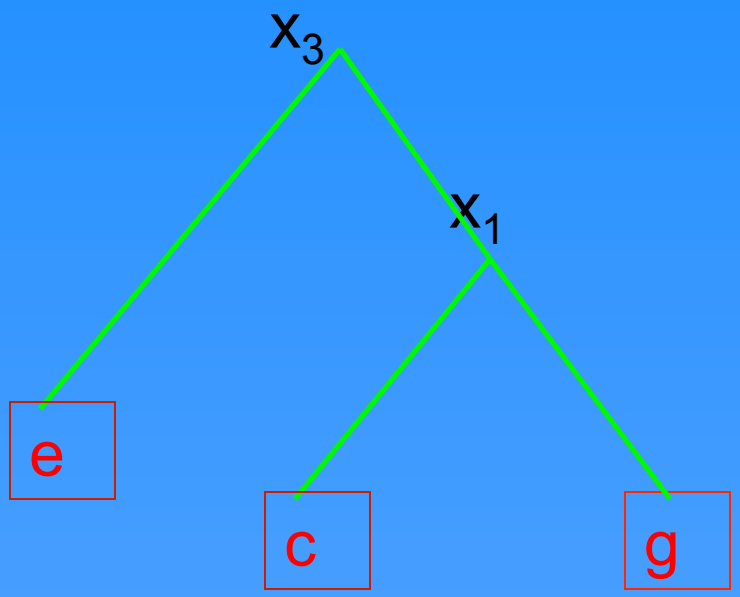
	a	d	e	h	i	x_1	x_2
a	x	x	x	x	x	x	x
d	x	x	x	x	x	x	x
e	x	x	x	x	x	x	x
h	x	x	x	x	x	x	x
i	x	x	x	x	x	x	x
x_1	x	x	x	x	x	x	x
x_2	x	x	x	x	x	x	x

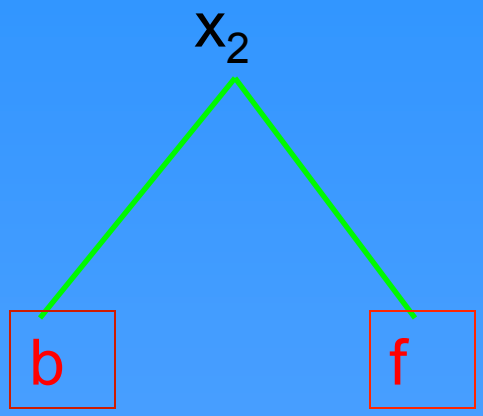


Recursive NJ

- Selected e and x_1 and x_3 is created.

	a	d	e	h	i	x_1	x_2
a							
d							
e							
h							
i							
x_1							
x_2							

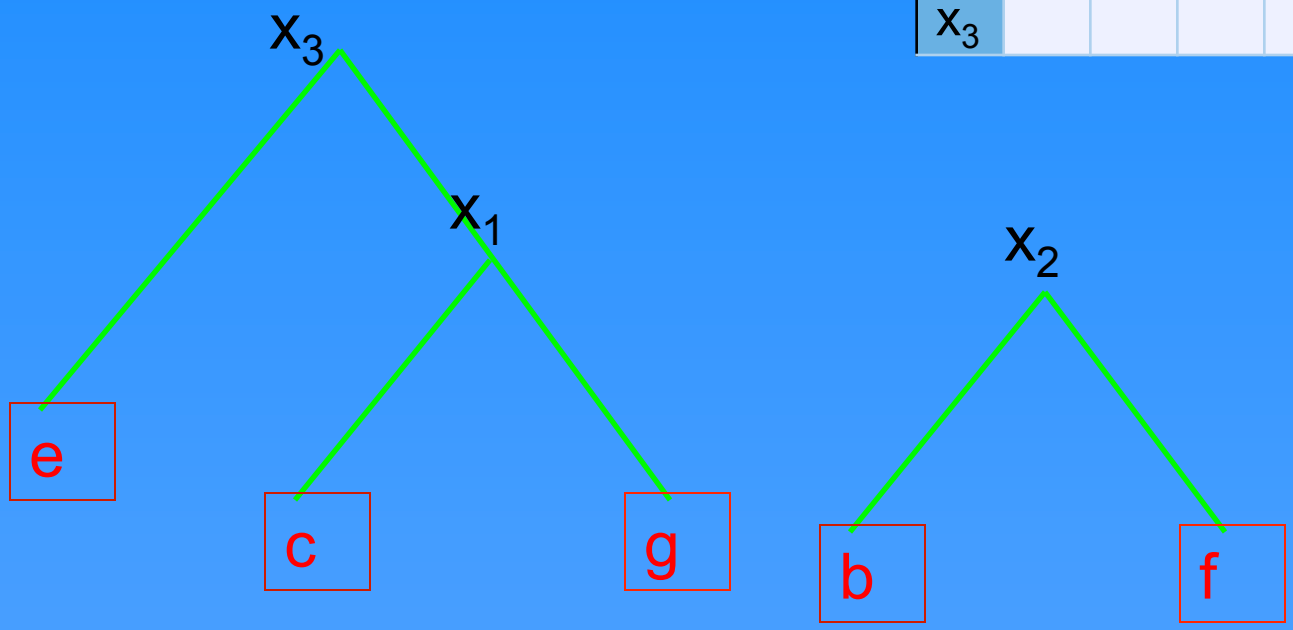




Recursive NJ

- Selected e and x_1 and x_3 is created.

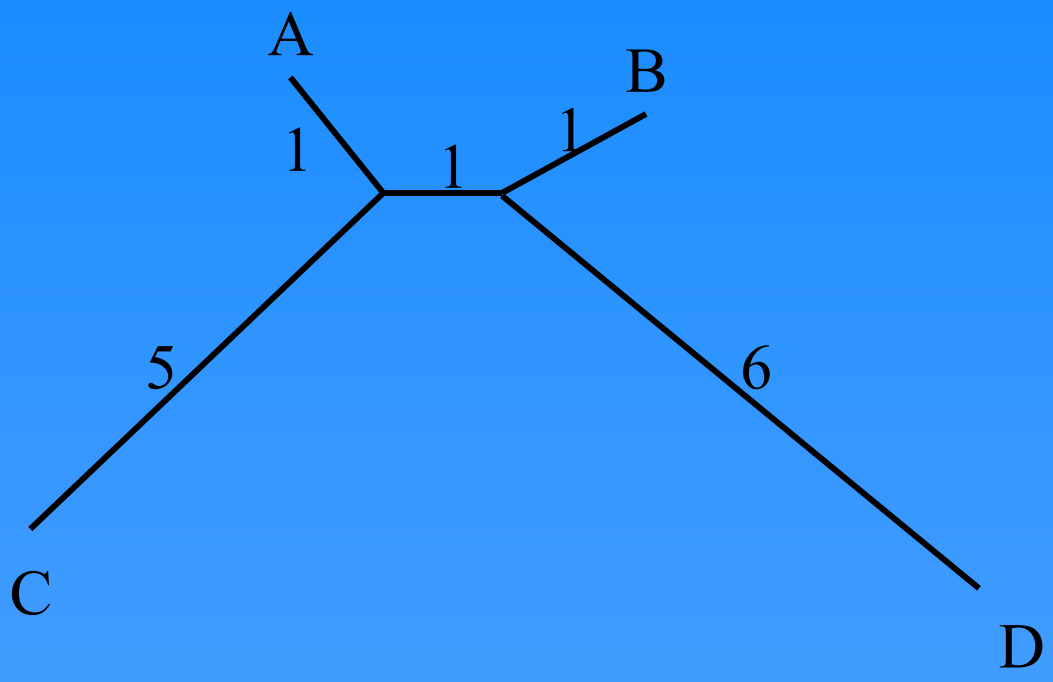
	a	d	h	i	x_2	x_3
a						
d						
h						
i						
x_2						
x_3						



Neighbor Selecting

How can we find (from distances alone) a pair of nodes which are neighboring leaves ("cherries")?

Unlike in ultrametric trees, closest nodes aren't necessarily cherries.



Saitou & Nei's Neighbor Joining Algorithm (1987)

- ~13,000 citations (*Science Citation Index*)
- Implemented in numerous phylogenetic packages
- Fastest implementation - $\theta(n^3)$
- Usually referred to as "the *NJ* algorithm"
- Identified by its neigbor selection criterion

select i, j which maximize the sum

$$Q(i,j) = \sum_r D(r,i) + \sum_r D(r,j) - (n-2)D(i,j)$$

← Saitou & Nei's neighbor-selection criterion

Saitou & Nei's NJ Algorithm (since 1987)

$$\max_{i,j}\left\{\sum_r D(r,i)+\sum_r D(r,j)-(L-2)D(i,j)\right\}$$

← Saitou & Nei's neighbor-selection criterion

- What makes Saitou&Nei's neighbor selection criterion so good?
- Is there any simpler consistent criterion?

✓ Numerous papers studying virtues of NJ

["Why does NJ work?", *Mihaescu, Levy and Pachter '06*]

["Neighbor-Joining Revealed", *Gascuel and Steel '06*]

✓ No other consistent, symmetric and linear neighbor selection criterion [Charleston et al '93] [Bryant '05]

Saitou & Nei's neighbor-selection Criterion

Select i,j which maximize

$$Q(i,j) = \sum_r D(r,i) + \sum_r D(r,j) - (n-2)D(i,j)$$

Intuition: NJ "tries" to select taxon-pairs which are *farthest* from all the rest.

Next we prove the criterion finds a cherry.

Let us denote $R_i = \sum_r D_{i,r}$

Hence: $Q(i,j) = R_i + R_j - (n-2)D(i,j)$

.

Proof of equality

$$Q(i,j) = \overbrace{\left(D(i,j) + \sum_{u \neq i,j} D(i,u) \right)}^{R_i} + \overbrace{\left(D(j,i) + \sum_{u \neq i,j} D(j,u) \right)}^{R_j} - (n-2)D(i,j)$$

$$= \sum_{u \neq i,j} [D(i,u) + D(j,u) - D(i,j)] + 2D(i,j)$$

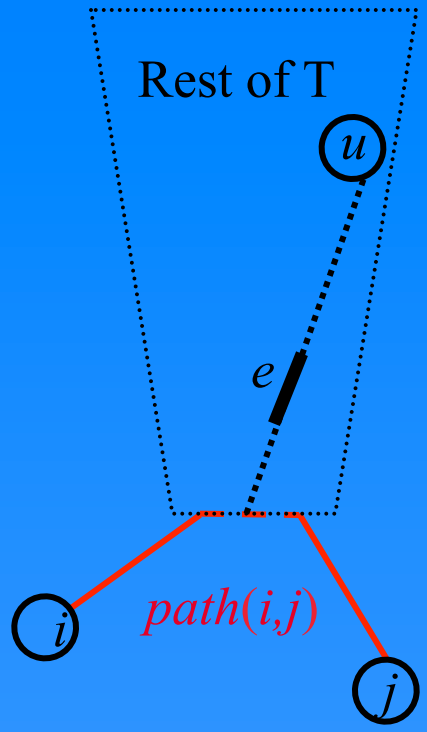
Seitou&Nei proof (cont.)

It remains to show that

$$Q(i,j) = \sum_{u \neq i,j} [D(i,u) + D(j,u) - D(i,j)] + 2D(i,j)$$

is maximized only when *i,j* are cherries.

Note that $[D(i,u) + D(j,u) - D(i,j)]$ is twice (the length of) the path emanating from path(i,j) going to leaf *u*.



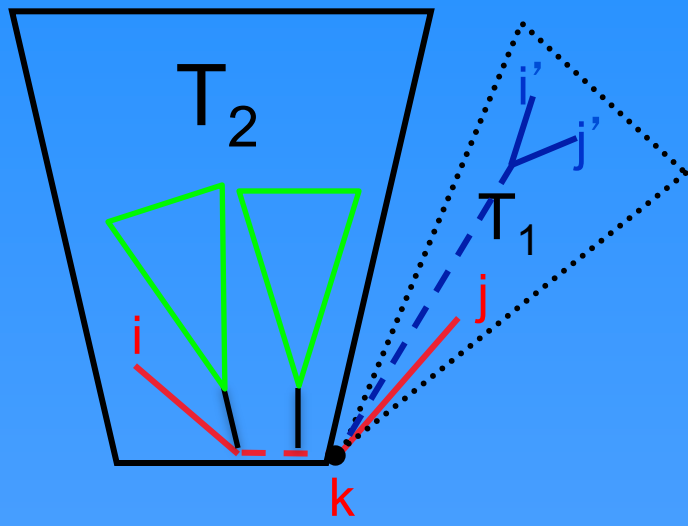
For a vertex *i*, and an edge *e* we define:
$N_i(e) = |\{u : e \text{ is on } path(i,u)\}|$
Then:

$$Q(i,j) = \overbrace{2\sum_{e \in path(i,j)} w(e)}^{2D(i,j)} + \overbrace{2\sum_{e \notin path(i,j)} N_i(e) w(e)}^{\sum_{u \neq i,j} [D(i,u) + D(j,u) - D(i,j)]}$$

Could also be $N_j(e)w(e)$

Seitou&Nei Proof Idea

- We decompose the tree as follows.
- T_2 is larger than T_1.
- There must be *i*’ ,*j*’ that are cherries in T_1(may be *i*’ or *j*’ equals *j*).
- The proof shows that Q(i’ ,j’) > Q(i,j).



Felsenstein example

D

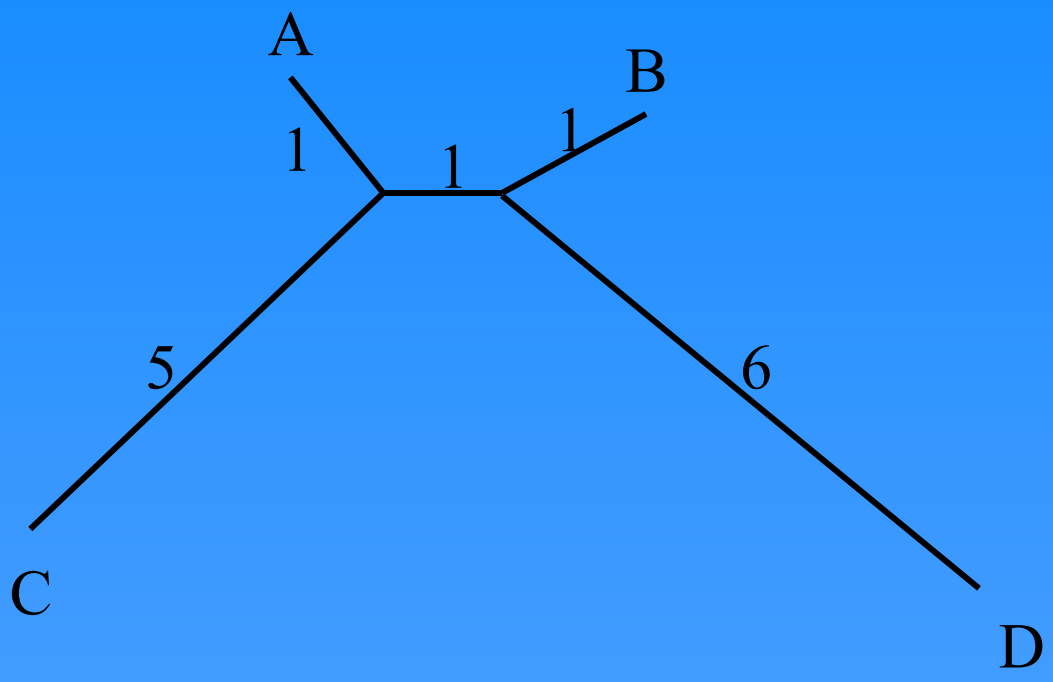
	A	B	C	D
A	0	3	6	8
B	3	0	7	7
C	6	7	0	12
D	8	7	12	0

R

A	B	C	D
17	17	25	27

Q

	A	B	C	D
A	0	28	30	28
B		0	28	30
C			0	28
D				0

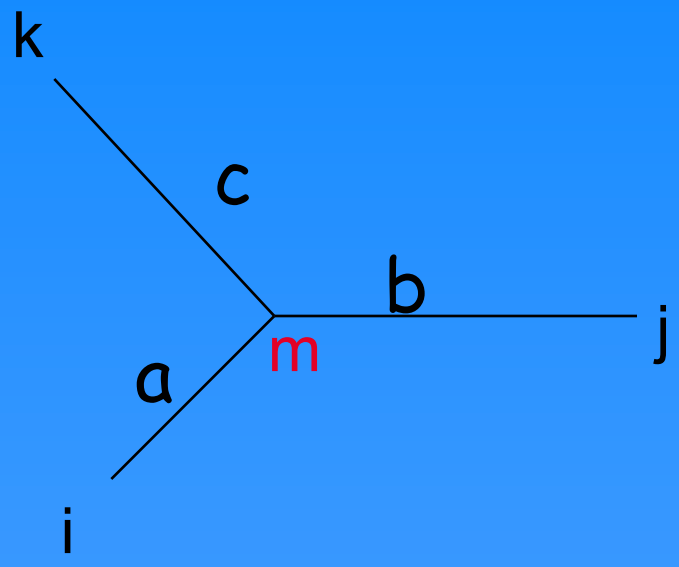


The Complete NJ Algorithm

- For each taxon *i*, $R_i = \sum_{k \neq i} D_{i,k}$
- While the matrix > 3x3
  - For *i,j* s.t. ($R_i + R_j - (n-2)D_{i,j}$) is the largest,
    - ★Create a node *u* with two leaves *i,j.*
    - ★Set $d_{u,i} \leftarrow \frac{1}{2} D_{i,j} + \frac{R_i - R_j}{2(n-2)}$
    - ★Replace i,j in the matrix *D* with *u* with distances ($D_{i,k}+D_{j,k}-D_{i,j}$)/2 for every entry *k.*

A characterization of additive metrics: the 4 points condition

Distances on 3 objects are always realizable by a (unique) tree with one internal node.



	i	j	k
i	0	a+b	a+c
j		0	b+c
k			0

$$d(i,j) = a + b$$
$$d(i,k) = a + c$$
$$d(j,k) = b + c$$

For instance

$$c = d(k,m) = \frac{1}{2}[d(i,k) + d(j,k) - d(i,j)] \geq 0$$

How about four objects?

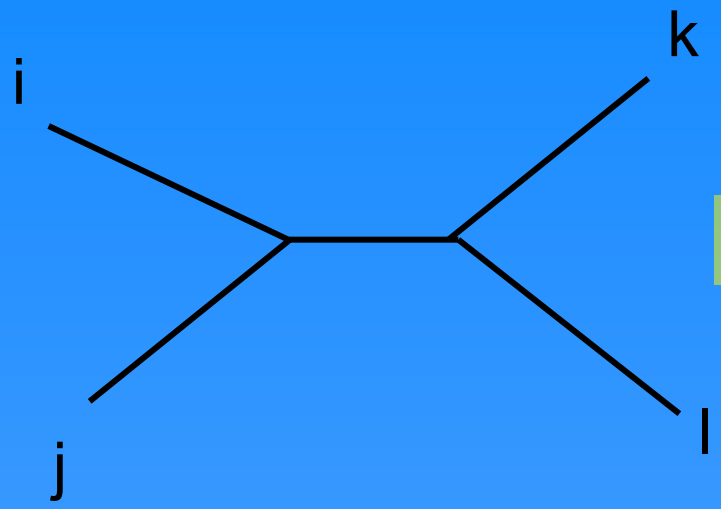
L=4: Not all distance metrics on 4 objects are additive: eg, there is no tree which realizes the below distances.

	i	*j*	*k*	*l*
i	0	2	2	2
j		0	2	2
k			0	3
l				0

The Four Points Condition

A necessary condition for distances on four objects to be additive: its objects can be labeled i,j,k,l so that:

$$d(i,k) + d(j,l) = d(i,l) + d(k,j) \geq d(i,j) + d(k,l)$$



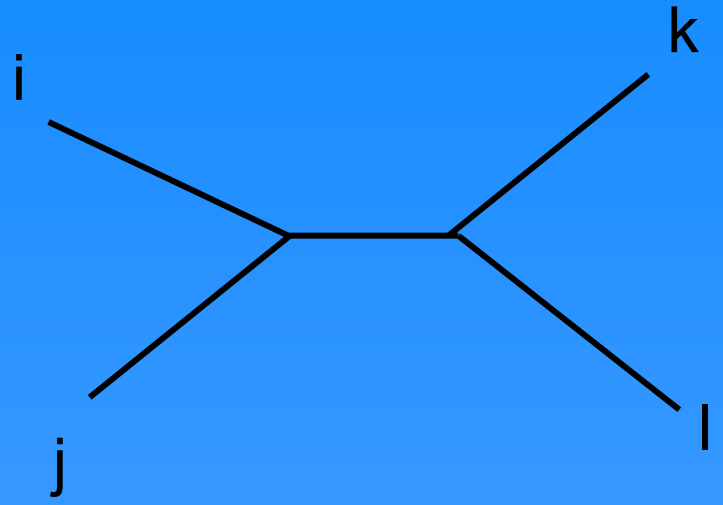
{{i,j},{k,l}} is a "**split**" of {i,j,k,l}.

Proof: By the figure...

The Four Points Condition

Definition: A distance metric satisfies the four points condition iff *any* subset of four objects can be labeled i,j,k,l so that:

$$d(i,k) + d(j,l) = d(i,l) + d(k,j) \geq d(i,j) + d(k,l)$$



Theorem: A distance matrix D on a set M is additive iff D satisfies the four points condition for all quartets in M.

Proof: → trivial, from the figure.

← By a straightforward construction (quartet algorithm).

The Complete NJ Algorithm

- For each taxon *i*, $R_i = \sum_{k \neq i} D_{i,k}$
- While the matrix > 3x3
  - For *i,j* s.t. ($R_i + R_j - (n-2)D_{i,j}$) is the largest,
    - ★Create a node *u* with two leaves *i,j.*
    - ★Set $d_{u,i} \leftarrow \frac{1}{2} D_{i,j} + \frac{(R_i - R_j)}{2(n-2)}$
    - ★Replace *i,j* in the matrix *D* with *u* with distances ($D_{i,k}+D_{j,k}-D_{i,j}$)/2 for every entry *k.*

- Claim: The distances from *u* to its children are correct.

We will show for a single node *k*.

Recall $d_{u,i} \leftarrow \frac{1}{2} D_{i,j} + \frac{(R_i - R_j)}{2(n-2)}$

Now:

$R_i - R_j =$

$D_{i,j} + \Sigma_{k \neq j} D_{i,k} - (D_{j,i} + \Sigma_{k \neq i} D_{j,k}) =$

$(n-2)(D_{i,u} - D_{j,u}) + \Sigma_{k \neq i,j}(D_{u,k} - D_{u,k}) =$

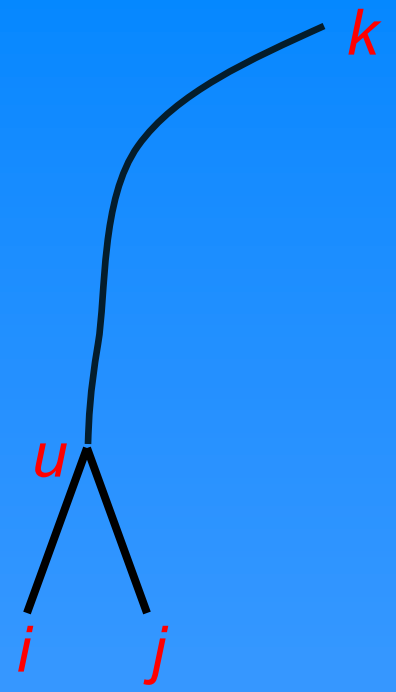
$(n-2)(D_{i,u} - D_{j,u})$.

On the other side:

$\frac{1}{2} D_{i,j} + \frac{(R_i - R_j)}{2(n-2)} = \frac{(n-2)(D_{i,u} - D_{j,u})}{2(n-2)} =$

$\frac{1}{2} * D_{i,j} + \frac{1}{2} (D_{i,u} - D_{j,u}) =$

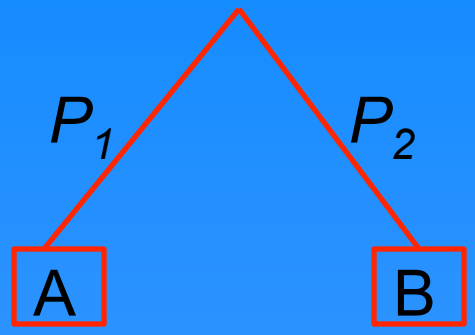
$\frac{1}{2} (D_{i,u} + D_{j,u} + D_{i,u} - D_{j,u}) = D_{i,u}$



The claim regarding the distances from *u* to all other remaining leaves *k*, $(D_{i,k} + D_{j,k} - D_{i,j})/2$ is straightforward.

Where distances come from?

- We spoke about distances.
- Where distances come from?
- In reality we have probabilities on tree branches.



- These in turn induce differences between the sequences at the leaves.

Molecular Evolutionary Models

- We started with edge substitution probabilities matrices (recall the evolutionary model)
- We then spoke about distances and algorithms converting a distance matrix to a tree realizing the matrix's distances
- We now link between the two terms by introducing another term – *substitution rate*, that will put things in the proper context.

Simulating a changing sequence

1. Begin with a sequence of 10,000 nucleotides.

TCAGAAAAACAGTTTATTTTCTTTTTTTCTGAGAGAGAGGGTCTTATTTTGTTGCCCAGGCTGGTGTGCAATGGTGCA

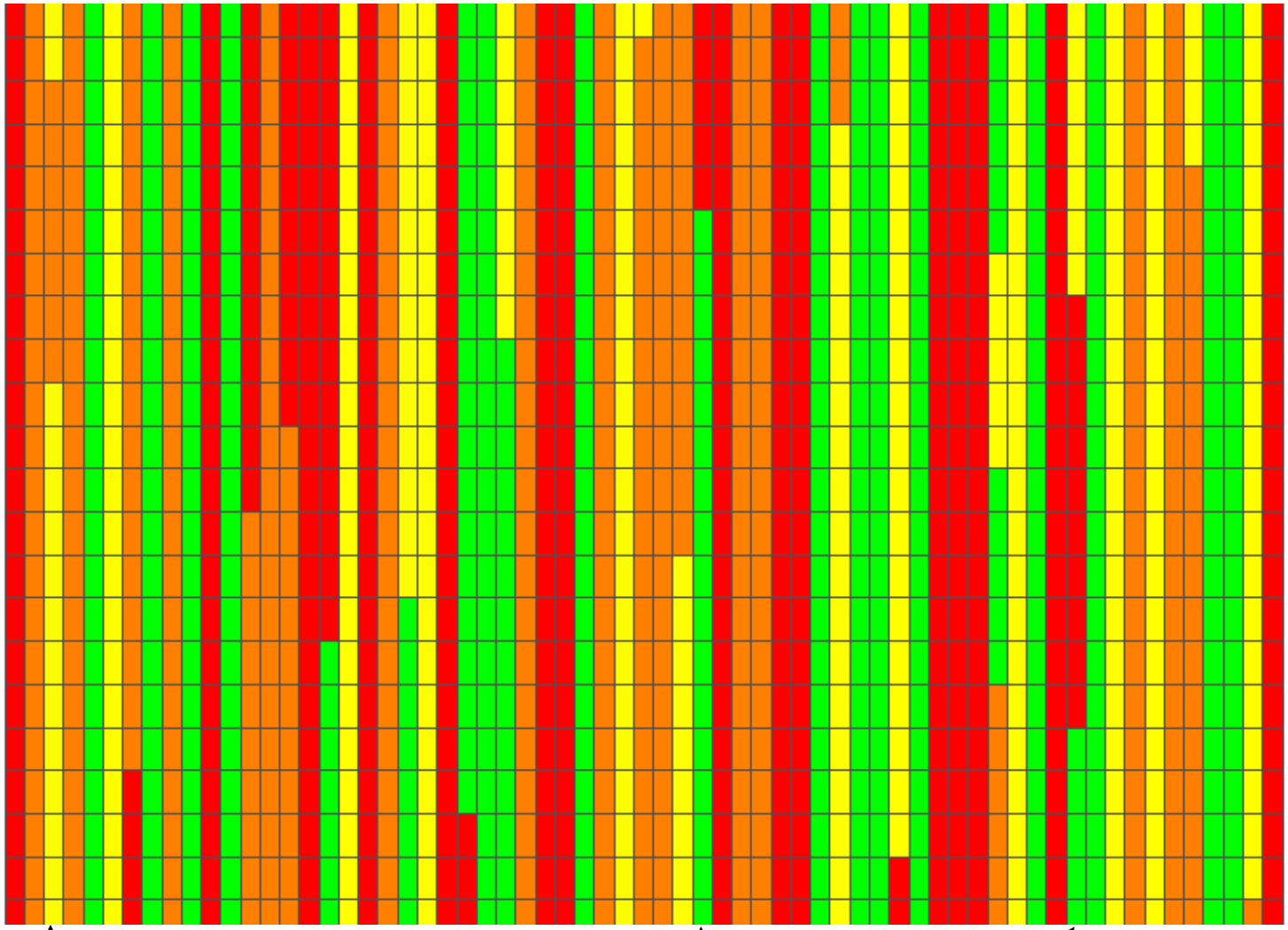
2. Choose a nucleotide at random and mutate it to another nucleotide.

TCAGAAAAACAGTTTATTTTCTTTTTTTCTGAGAGAGAGGGTCTTATTTTGTTGCCCAGGCTGGTGTGCAATGGTGCA
TCAGAAAAACAGTTTATTTTCTTTTTTTCTGAGTGAGAGGGTCTTATTTTGTTGCCCAGGCTGGTGTGCAATGGTGCA

3. Repeat 10,000 times. How many differences accumulate?

Taken from Itai Yanai

A sequence mutating at random

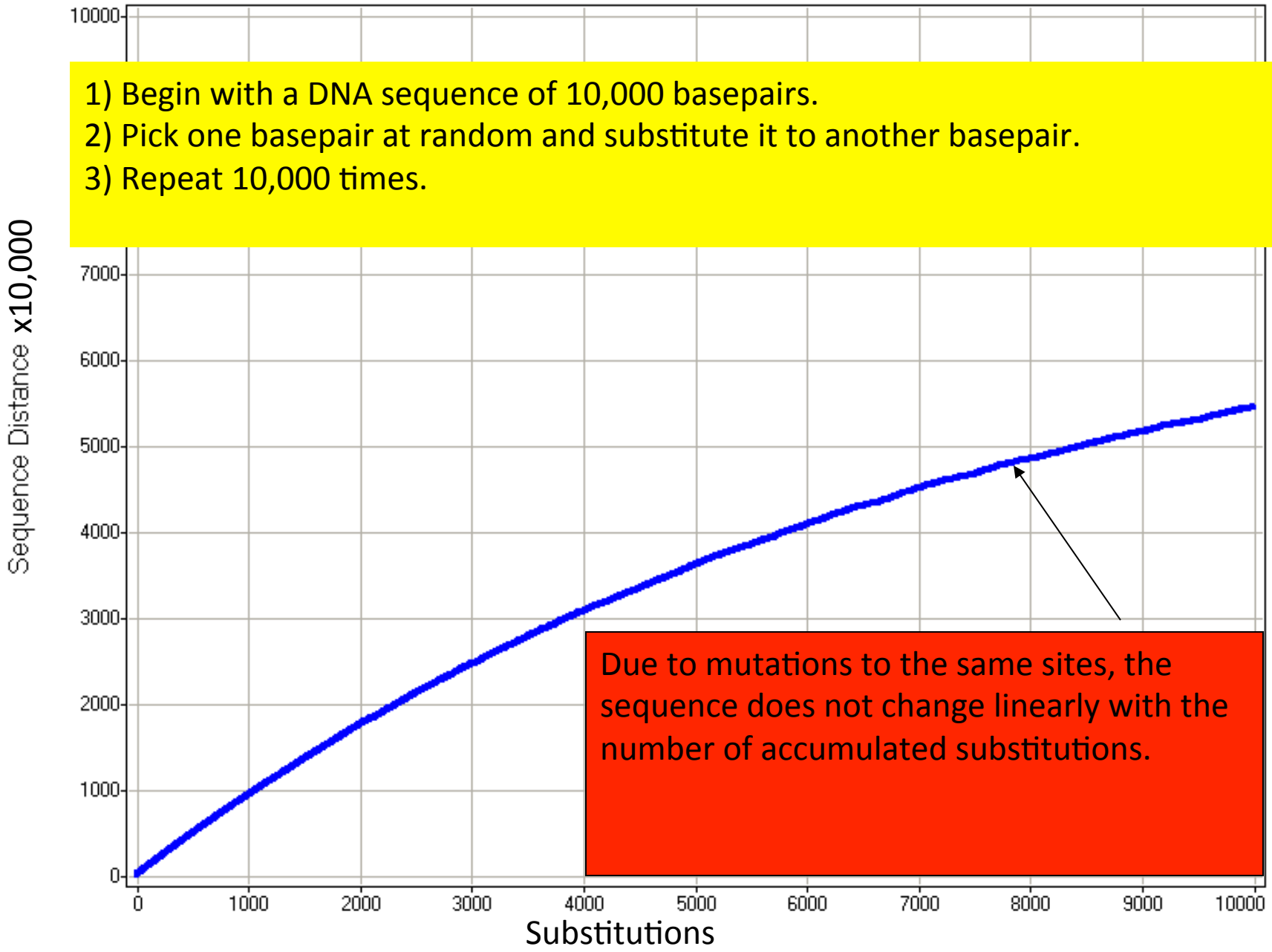


21 changes but only 17 differences

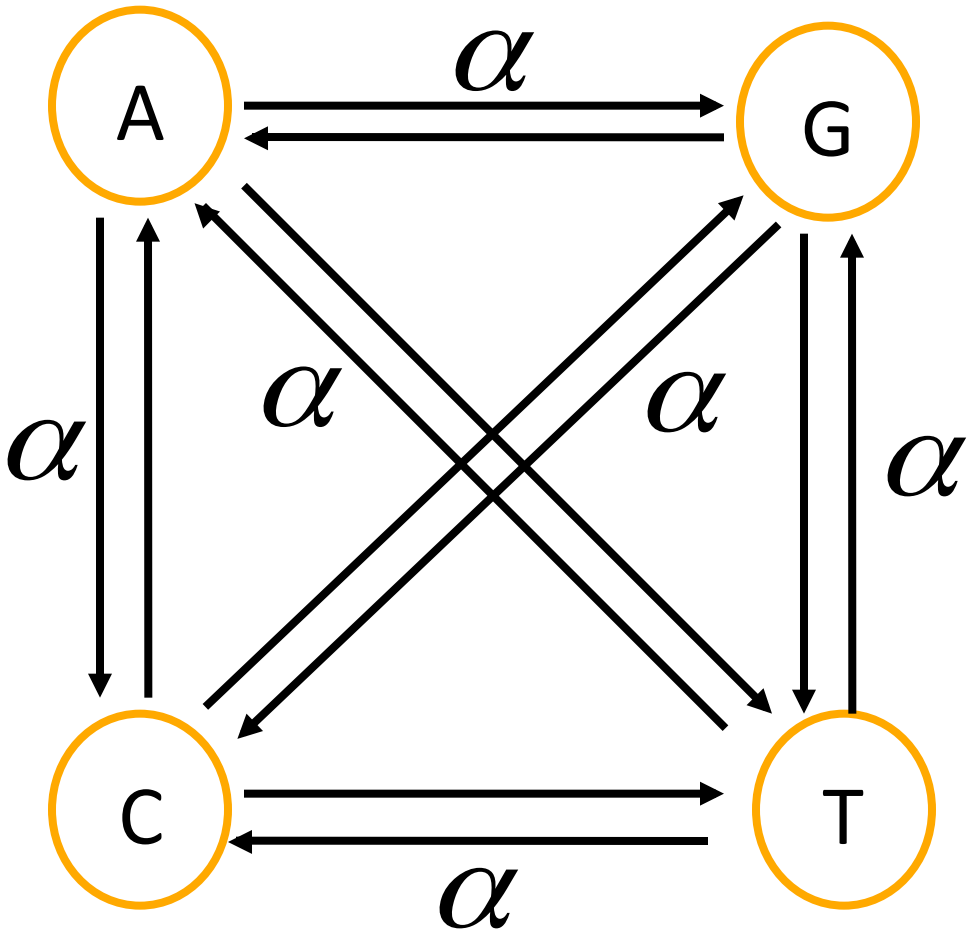
Taken from Itai Yanai

Simulating a changing sequence

1) Begin with a DNA sequence of 10,000 basepairs.
2) Pick one basepair at random and substitute it to another basepair.
3) Repeat 10,000 times.

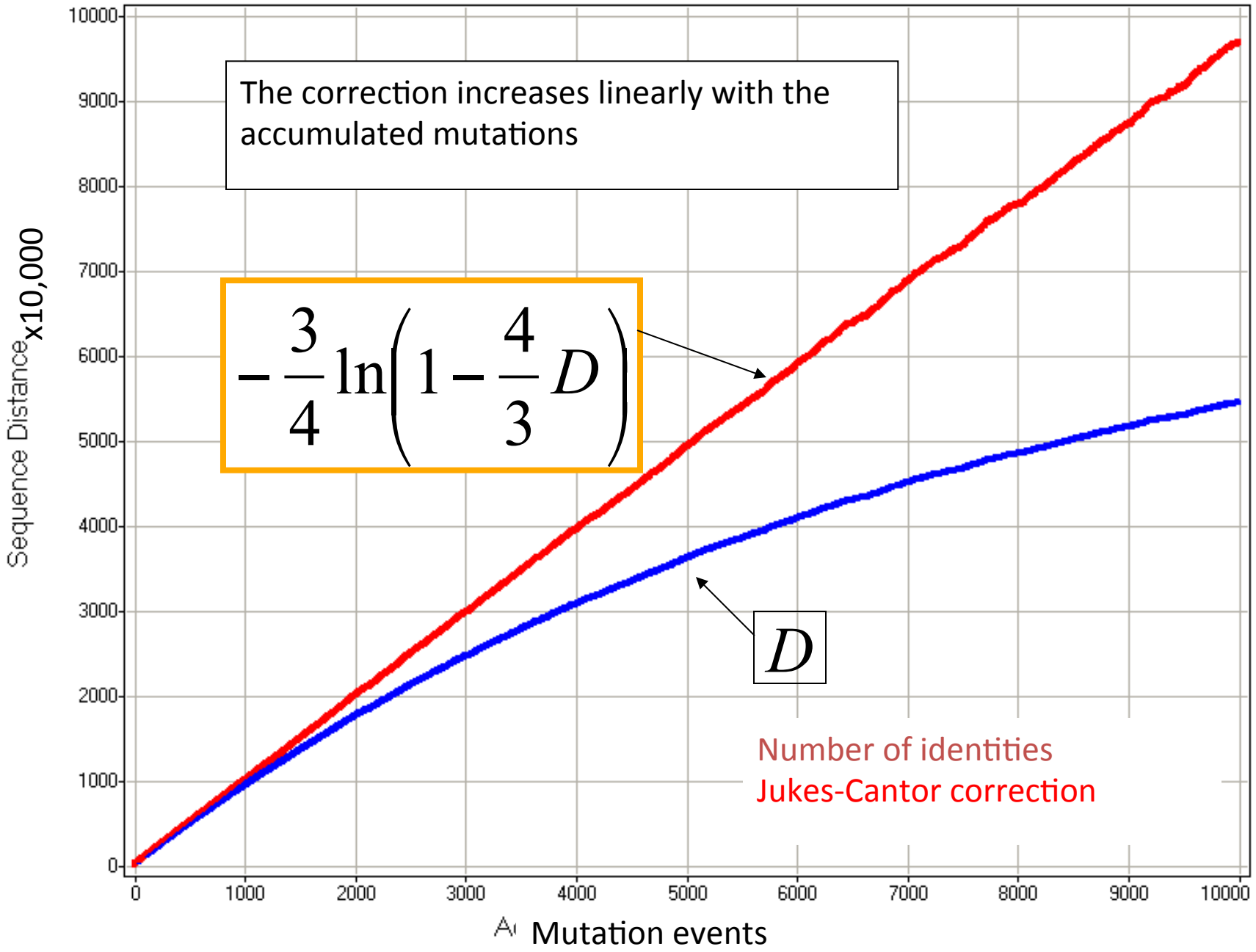


Jukes-Cantor model



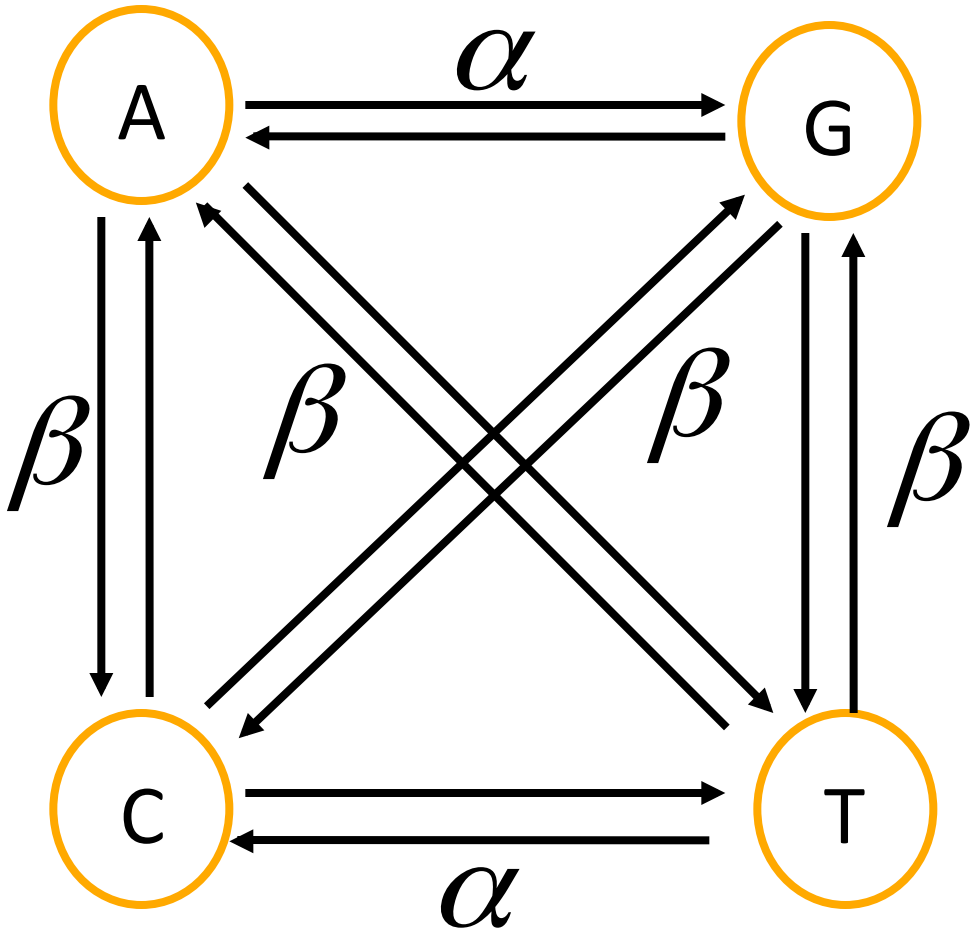
In this simulation we assumed that all changes occur at equal probabilities

The Jukes-Cantor correction



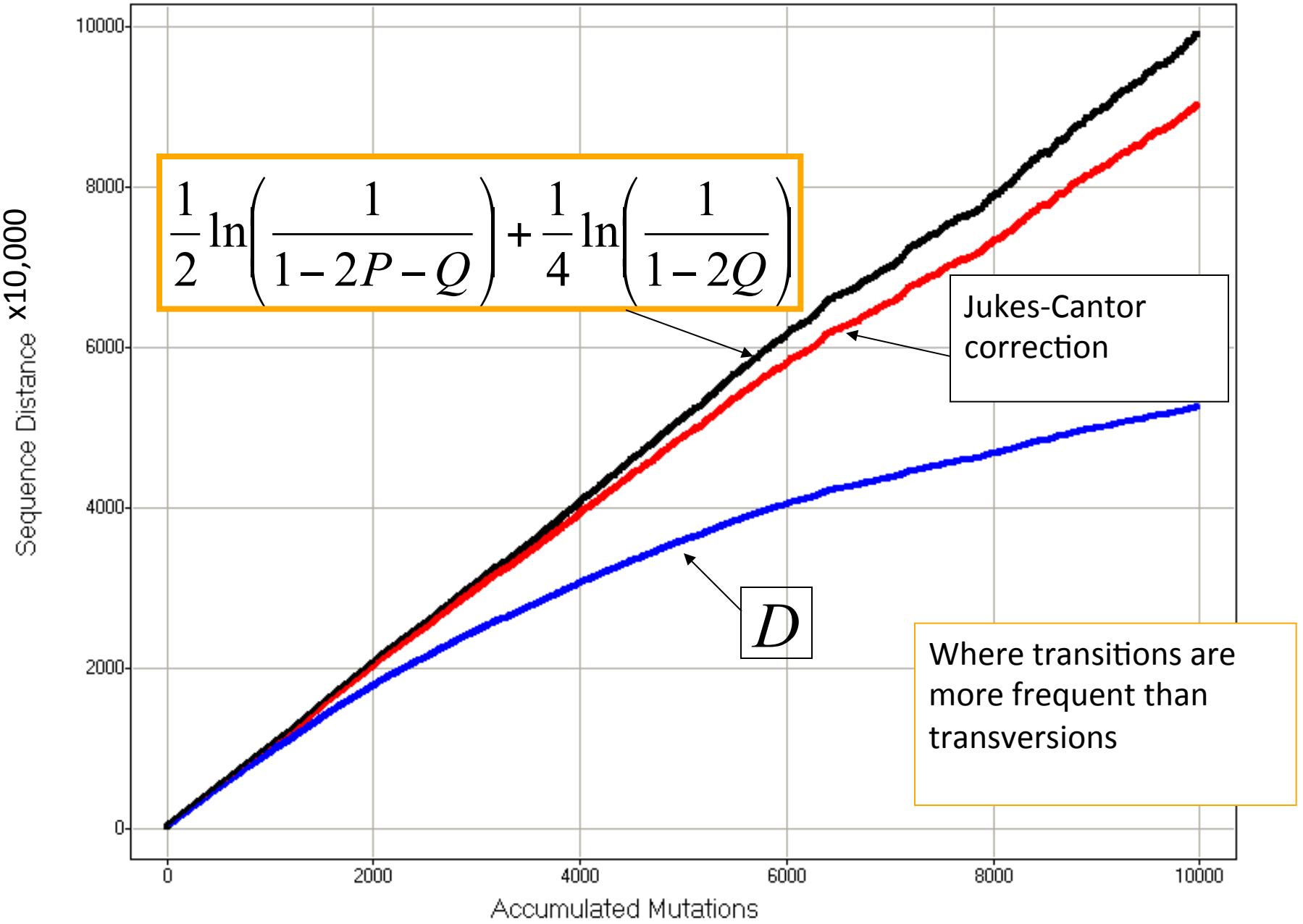
Kimura model

A more realistic simulation represents different probabilities for transitions than to transversions



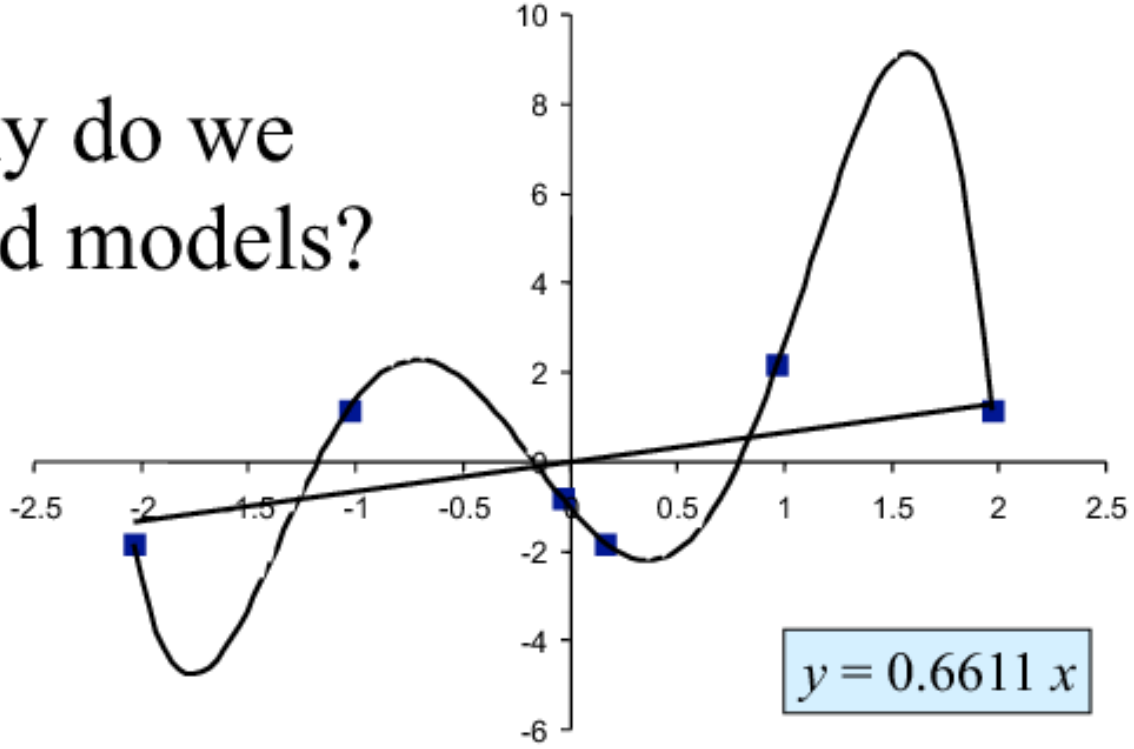
α = transitions β = transversions

The Kimura correction



$$y = -1.5972\,x^5 + 23.167\,x^4 - 126.18\,x^3 + 319.17\,x^2 - 369.22\,x + 155.67$$

Why do we need models?



Models

- Models help us intelligently **interpolate between our observations** for purposes of **making predictions**
- **Adding parameters** to a model generally increases its fit to the data
- **Underparameterized** models lead to poor fit to observed data points
- **Overparameterized** models lead to poor prediction of future observations
- Criteria for choosing models include likelihood ratio tests, AIC, BIC, Bayes Factors, etc.
  - all provide a way to choose a model that is neither underparameterized nor overparameterized

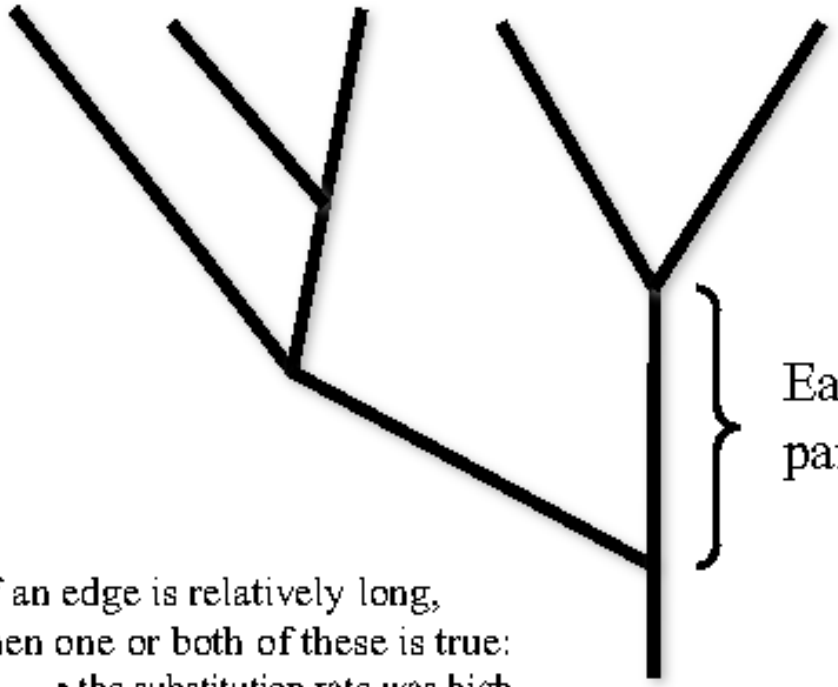
Evolutionary Model Parameters

- Mutation Rate between every two bases
- Base frequency at each node
- Time duration between two nodes

Simplifying Assumptions

- Same rate on all branches
- Same rates between sets of bases
- Uniform base frequency

Substitution Models



Each edge length is itself a function of substitution rate (α) and time (t)

$$\nu = 3\,\alpha\,t$$

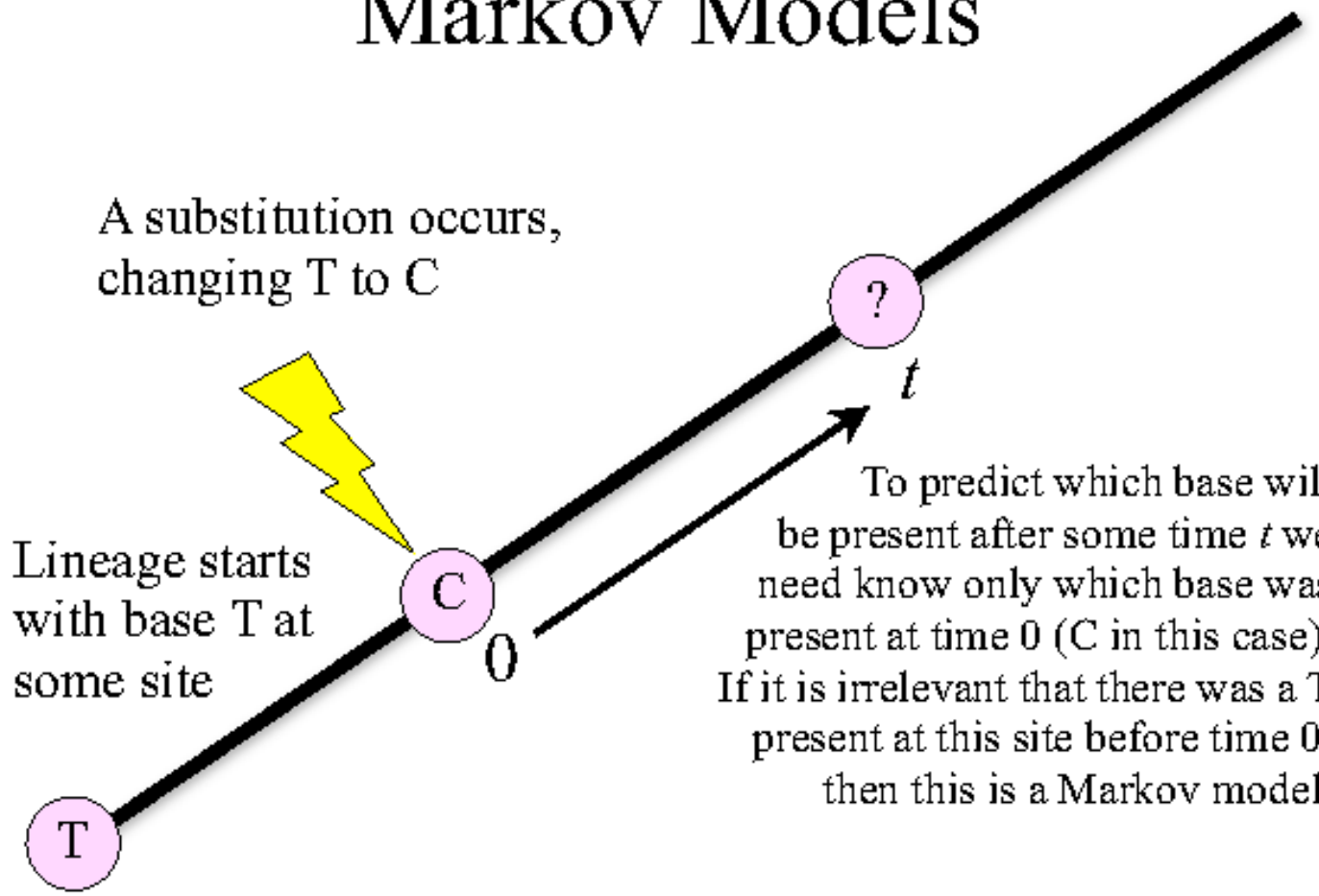
(Jukes-Cantor model)

Each edge length (ν) is a parameter in the model

If an edge is relatively long, then one or both of these is true:

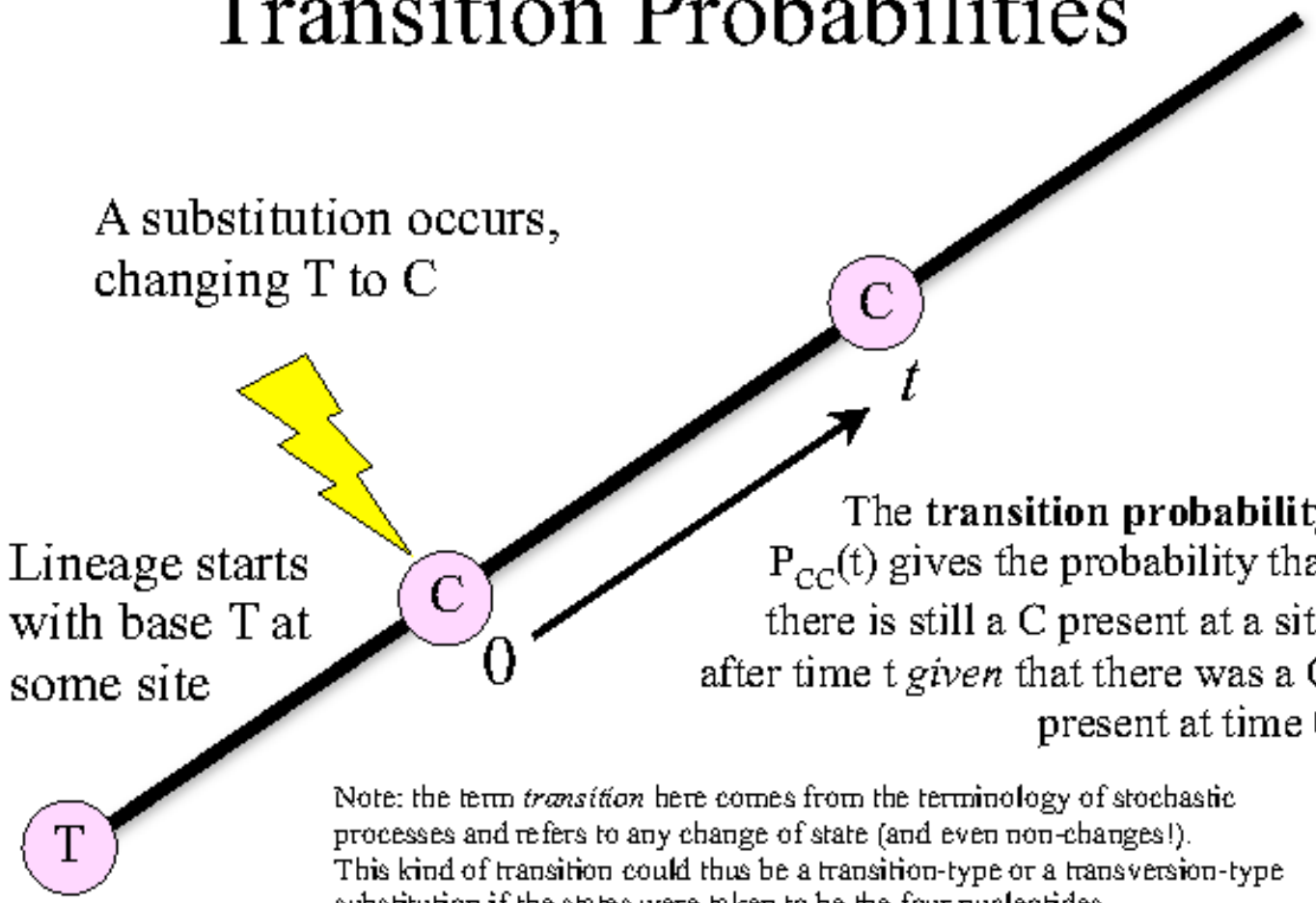
- the substitution rate was high
- the lineage was in existence for a long time

Markov Models



To predict which base will be present after some time *t* we need know only which base was present at time 0 (C in this case). If it is irrelevant that there was a T present at this site before time 0, then this is a Markov model.

Transition Probabilities



Note: the term *transition* here comes from the terminology of stochastic processes and refers to any change of state (and even non-changes!). This kind of transition could thus be a transition-type or a transversion-type substitution if the states were taken to be the four nucleotides

JC Transition Probability

Here is the probability that a site starting in state T will end up in state G after time *t* when the substitution rate is α :

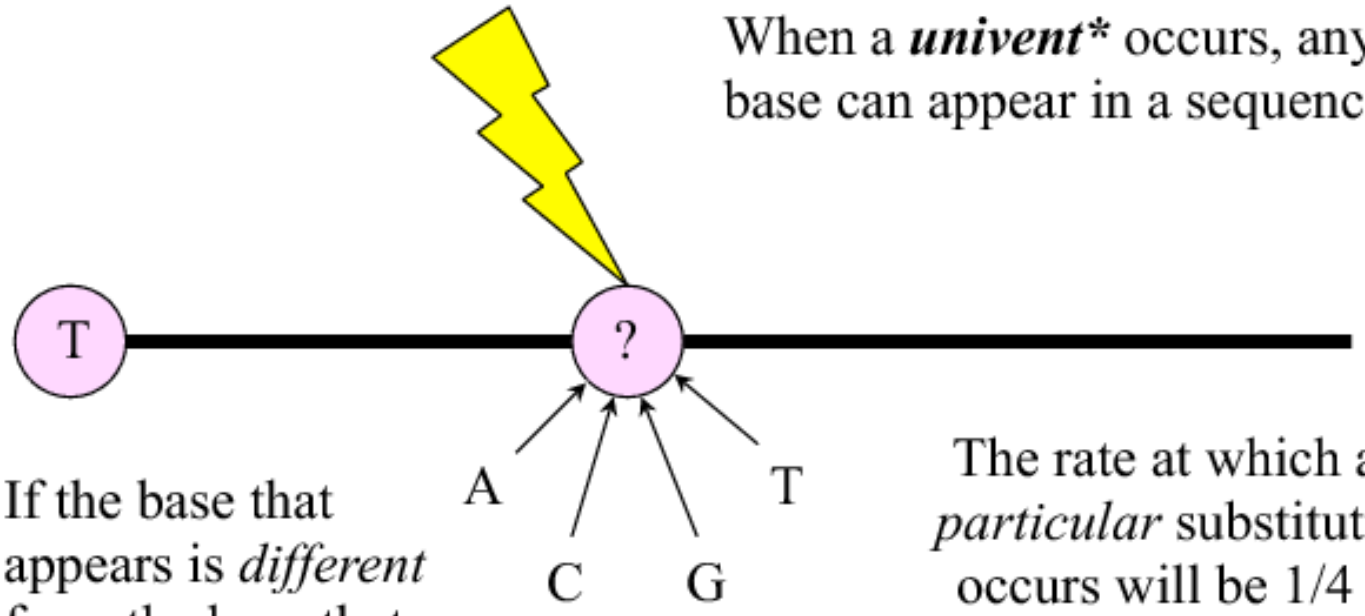
$$P_{TG}(t) = \frac{1}{4}\left(1 - e^{-4\alpha t}\right)$$

The JC model has only 1 parameter: αt

(the symbol e is the base of the natural logarithms and is thus a constant: 2.718281828459045...)

Where does a transition probability formula such as this come from?

"Univents" vs. substitutions



When a ***univent**** occurs, any base can appear in a sequence.

If the base that appears is *different* from the base that was already there, then a **substitution** event has occurred.

The rate at which any *particular* substitution occurs will be 1/4 the univent rate

*You will not find the word *univent* in the literature; however, this concept plays an important role in a technique called *uniformization*, which hold some promise for making complex models more practical. We will discuss uniformization later in the semester.

Poisson Processes

$$\Pr(x \text{ events} \mid \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$$

Suppose the events we were interested in were accidents at Four Corners, CT. Suppose accidents occur at a rate of $\mu = 0.2$/day and let's consider a time period of $t = 7$ days. λ is the expected number of accidents per week: $\lambda = \mu t = 1.4$. The probability of seeing exactly one accident (i.e. $x = 1$) in a week is thus:

$$\Pr(1 \text{ event} \mid \lambda = 1.4) = \frac{(1.4)^1 e^{-1.4}}{1!} = 0.345$$

Poisson Processes

$$\Pr(0 \text{ events} \mid \lambda) = \frac{\lambda^0 e^{-\lambda}}{0!} = e^{-\lambda}$$

$$\Pr(\text{at least 1 event} \mid \lambda) = 1 - e^{-\lambda}$$

Deriving a transition probabilities

Calculate the probability that a site currently T will change to G over time *t* when the rate of this particular substitution is α:

Pr(zero univents) = $e^{-\mu t}$

JC69 model

- Bases are assumed to be equally frequent (all 0.25)
- Assumes rate of substitution (α) is the same for all possible substitutions
- Usually described as a 1-parameter model (the parameter being αt)
- Remember, however, that each edge in a tree can have its own αt, so there are really as many parameters in the model as there are edges in the tree!

Jukes, T. H., and C. R. Cantor. 1969. Evolution of protein molecules. Pages 21-132 *in* H. N. Munro (ed.), *Mammalian Protein Metabolism*. Academic Press, New York.

Transition Probabilities: Remarks

$$P_{TA}(t) = 0.25\,(1 - e^{-4\alpha t})$$
$$P_{TC}(t) = 0.25\,(1 - e^{-4\alpha t})$$
$$P_{TG}(t) = 0.25\,(1 - e^{-4\alpha t})$$
$$\underline{P_{TT}(t) = 0.25\,(1 - e^{-4\alpha t})}$$
$$= 1 - e^{-4\alpha t}$$

Oops! Should be 1.0 because T must either stay the same or change to A, C or G. What are we forgetting?

Transition Probabilities: Remarks

$$P_{TA}(t) = 0.25\,(1 - e^{-4\alpha t})$$

$$P_{TC}(t) = 0.25\,(1 - e^{-4\alpha t})$$

$$P_{TG}(t) = 0.25\,(1 - e^{-4\alpha t})$$

$$P_{TT}(t) = e^{-4\alpha t} + 0.25\,(1 - e^{-4\alpha t})$$

$$= e^{-4\alpha t} + (1 - e^{-4\alpha t})$$

$$= 1$$

Forgot to account for the possibility that the base could stay the same even if there were *no* disruptions over time t

More on Transition Probabilities

$$P_{ij}(t) = 0.25\,(1 - e^{-4\alpha t})$$

$$P_{ii}(t) = 0.25 + 0.75\,e^{-4\alpha t}$$

Consider an edge representing an amount of time t and a substitution rate α.

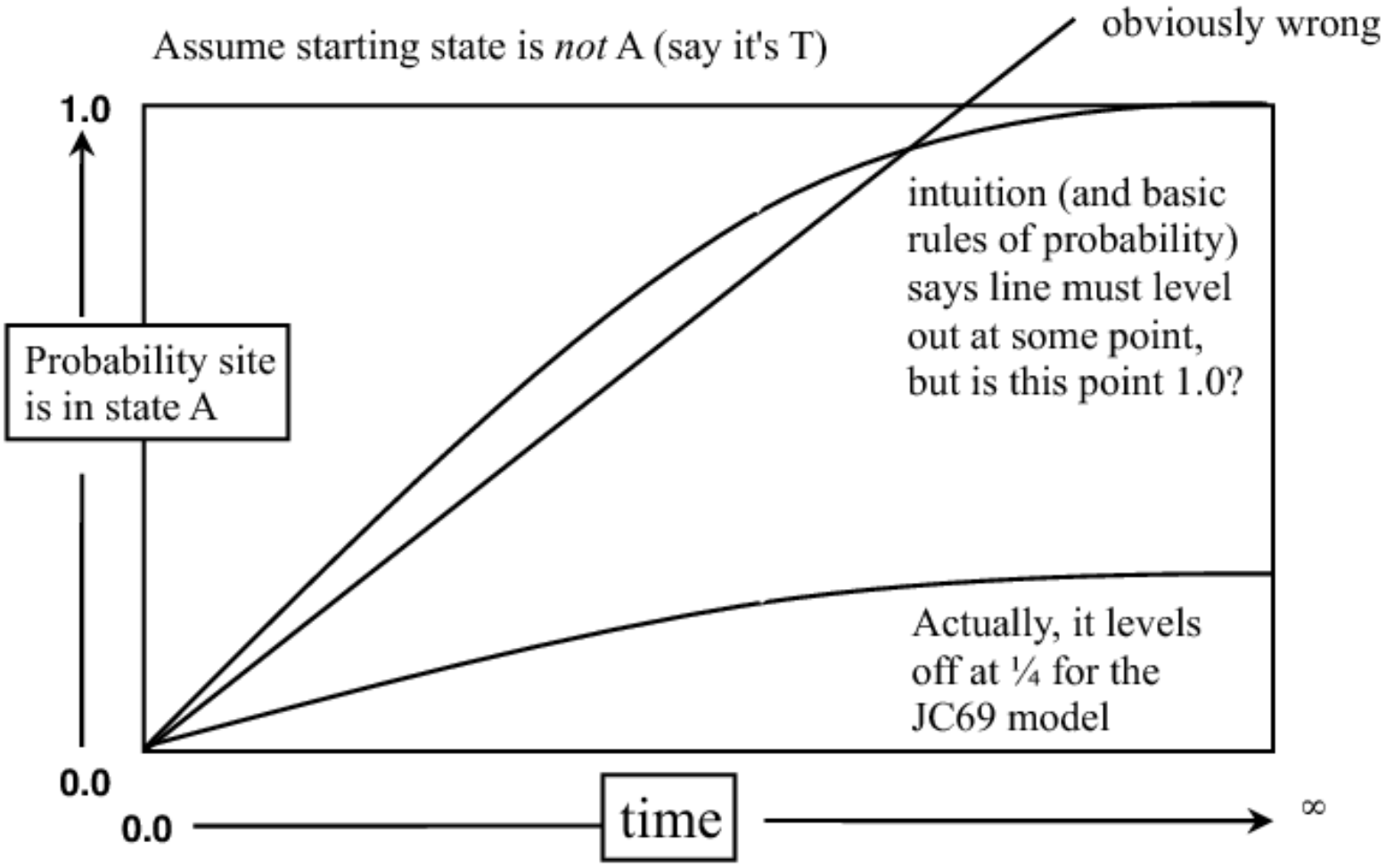
What are the transition probabilities if $t = \infty$?

$$P_{ij}(\infty) = P_{ii}(\infty) = 0.25$$

What are the transition probabilities if $t = 0$?

$$P_{ij}(0) = 0.0,\ P_{ii}(0) = 1.0$$

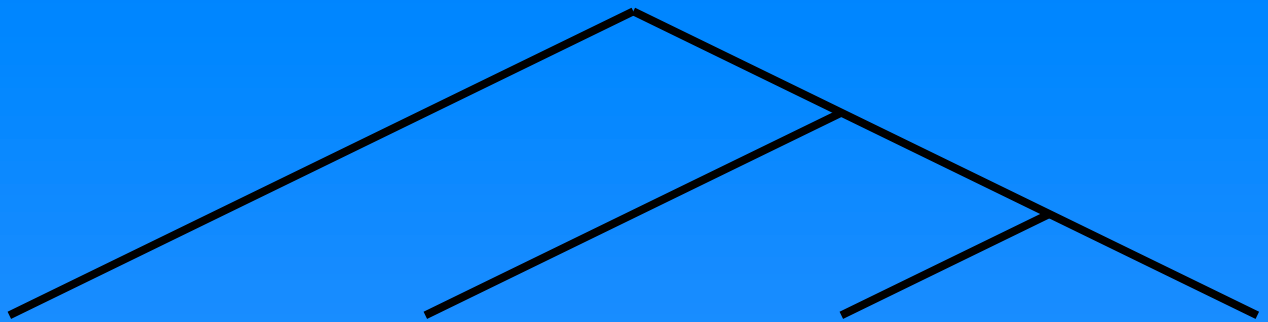
Transition probability intuition

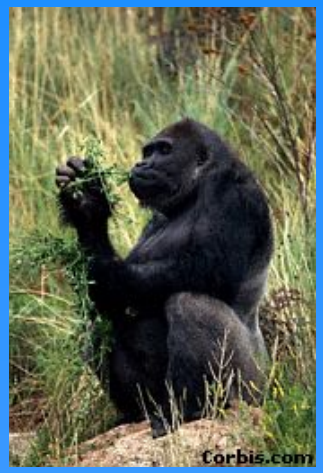


Closing Remarks

- We started with probabilities by which we could simulate evolution on a tree.
- These probabilities were derived from a rate (Poisson) process occurring in nature.
- In order to apply distance approaches, that are consistent and efficient, we transformed the non linear process to linear distances.
- All this space is basically *Maximum Likelihood*.

Introduction to Phylogenetics





Thank You