Week 3: Parsimony variants, compatibility, statistics and parsimony

Genome 570

January, 2016

Weighting using a function

Suppose that each step is to be weighted $1/(n+2)$ if the character has a total of n steps.

Then the total weight of steps when there are n steps in that character is $n \times 1/(n+2)$ which is $n/(n+2)$

Total steps in the character	Weight of each step	Total weight of all steps
0	0.5	0
1	0.3333	0.3333
2	0.25	0.5
3	0.2	0.6
4	0.1667	0.6667
5	0.142857	0.71428

The rationale for doing this is to somewhat discount the information from more rapidly changing characters.

Successive weighting

J.S. Farris suggested (1969):

1. Infer a tree with unweighted parsimony
2. Calculate weights for each character (or site) based on the number of changes in that character on that tree
3. Use those weights to infer a new tree
4. Unless the tree hasn't changed, go back to step 2.

This method, which was put forward in the first modern quantitative paper on weighting, was intended to discount rapidly-evolving characters.

Successive weighting

There are 15 possible unrooted trees, which fall into 5 types according to how many changes they have in each character. The table shows the total weighted number of changes when each tree type is evaluated using the weights implied by the 5 different tree types.

number of trees	have pattern of changes:	type of tree	tree type used for weights I	II	III	IV	V
1	(1,1,1,2,2,1)	I	2.333	2.250	2.167	2.417	2.083
2	(1,2,1,2,2,1)	II	2.667	2.500	2.500	2.667	2.333
2	(2,1,2,2,2,1)	III	3.000	2.917	2.667	2.917	2.583
3	(2,2,2,1,1,1)	IV	2.833	2.667	2.500	2.500	2.333
7	(2,2,2,2,2,1)	V	3.333	3.167	3.000	3.167	2.833

From this table you can figure out what the sequence of trees will be if you start from one of these types. Do you get different ultimate outcomes?

Ties in successive weighting

An example of successive weighting that would show the difficulty it has in detecting ties. The table shows the total weighted number of changes when each tree type is evaluated using the weights implied by the different tree types.

number of trees	have pattern of changes:	type of tree	tree type used for weights I	II	III
1	(1,1,2,2)	I	1.667	1.833	1.5
1	(2,2,1,1)	II	1.833	1.667	1.5
1	(2,2,2,2)	III	2.333	2.333	2

Nonsuccessive weighting

We can use a different algorithm to avoid the issue of dependence on starting point which is seen in successive weighting:

- Search over tree space in the usual way
- For each tree, evaluate the number of steps in each character (or site)
- Calculate the weight for the character from its number of steps on the current tree in the search.
- Add up the weighted steps across characters to evaluate that tree.

This is equivalent to looking only at the diagonals in the preceding tables of trees. Weights are never based on a different tree.

Evaluating compatibility with 0/1 characters

E. O. Wilson (*Systematic Zoology, 1965*) pointed out that for two characters, if all four combinations 00, 01, 10 and 11 exist in one or another species, the two characters must be incompatible with each other, in the sense that there can be no tree on which they each change only once (so the derived state is uniquely derived).

Compatible characters

	0	1
0	X	X
1		X

Incompatible characters

	0	1
0	X	X
1	X	X

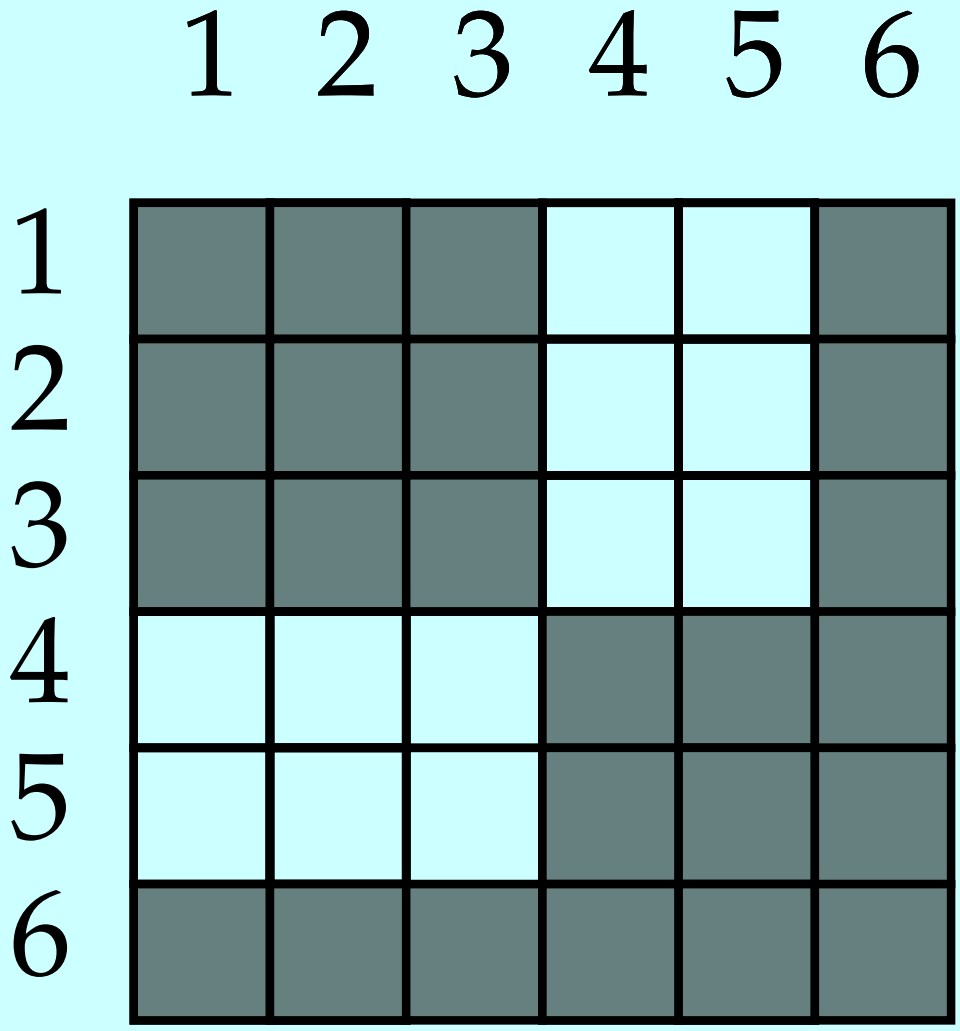
The logic is straightforward: to originate a new combination requires a step in one character (or both). With four combinations there are 3 steps required, one more than the number that would be needed if both characters were incompatible. (In genomics this is called the "three-gamete condition" – out of the four possibilities there must be three or fewer for there to have been no recombination and no recurrent mutations).

Data example for compatibility

The data set Table 1.1 with an added species all of whose characters are 0.

	Characters					
Species	1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0
Omega	0	0	0	0	0	0

The compatibility matrix for this data set



The darkly-shaded boxes are the combinations that are compatible.

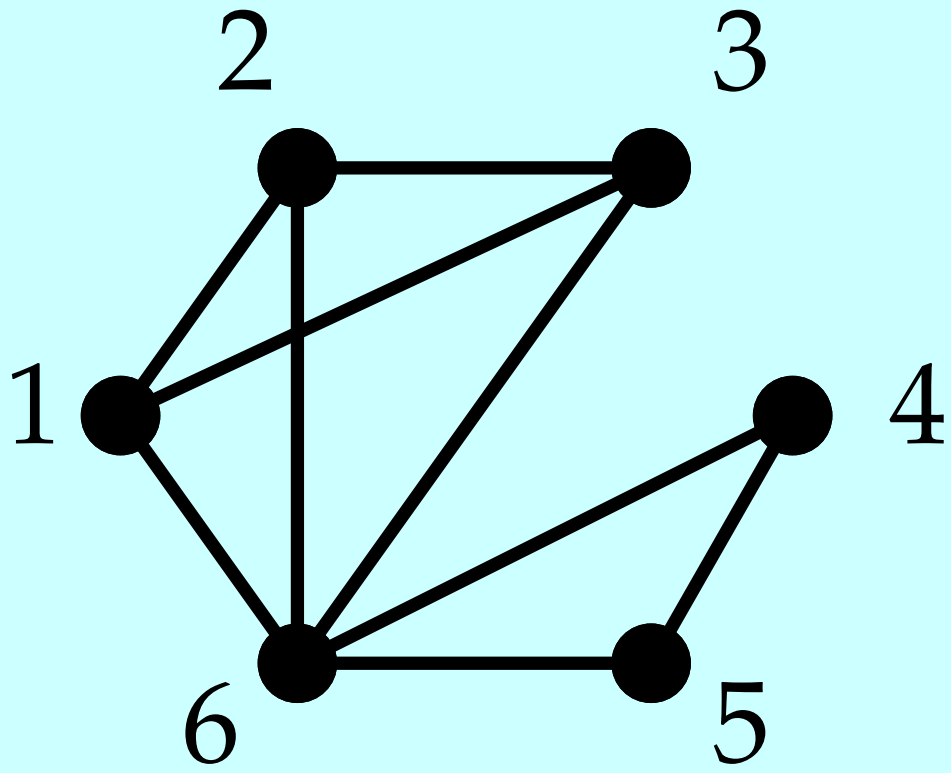
The Pairwise Compatibility Theorem

> A set S of characters has all pairs of characters compatible with each other if and only if all of the characters in the set are jointly compatible (in that there exists a tree with which all of them are compatible).

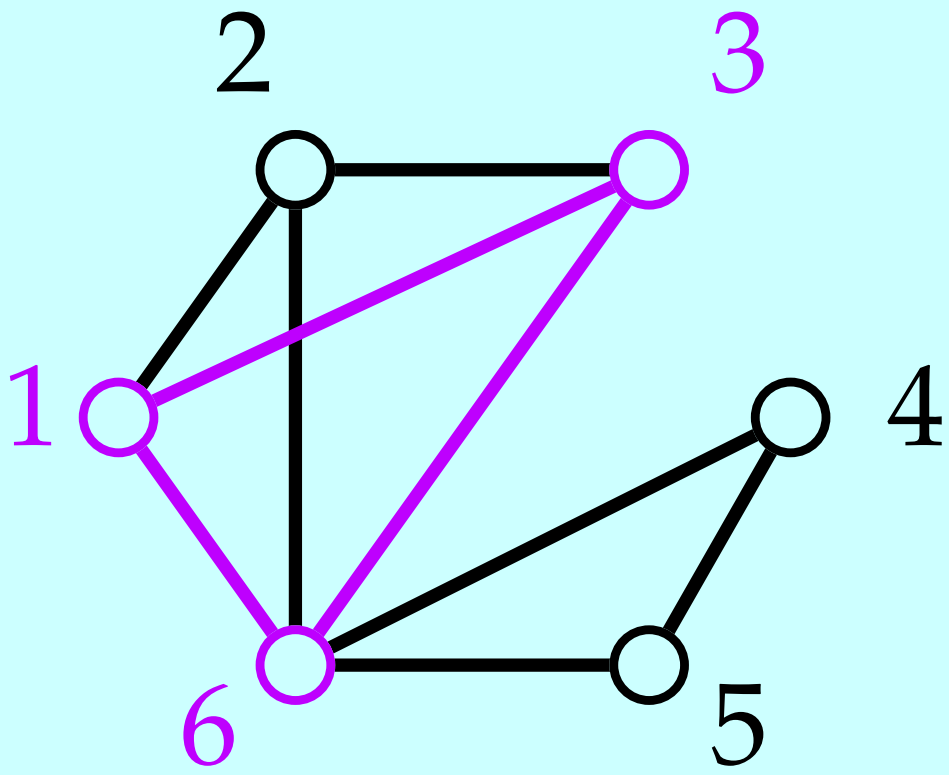
This theorem is true, given the right conditions, which we will explain.

The compatibility graph

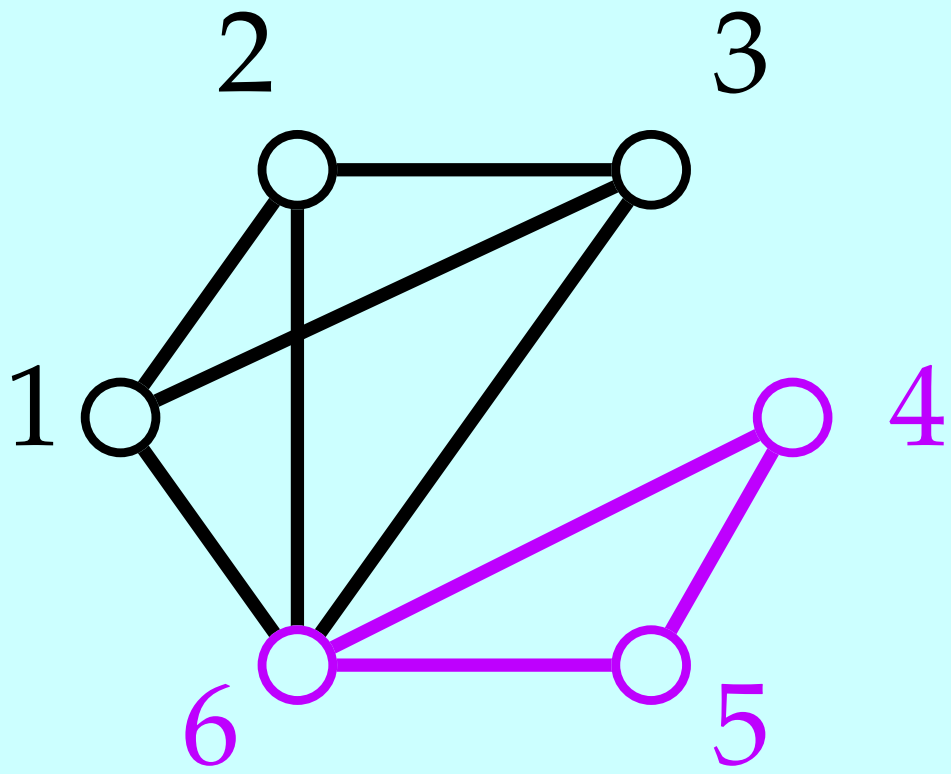
Maximal cliques? Largest?



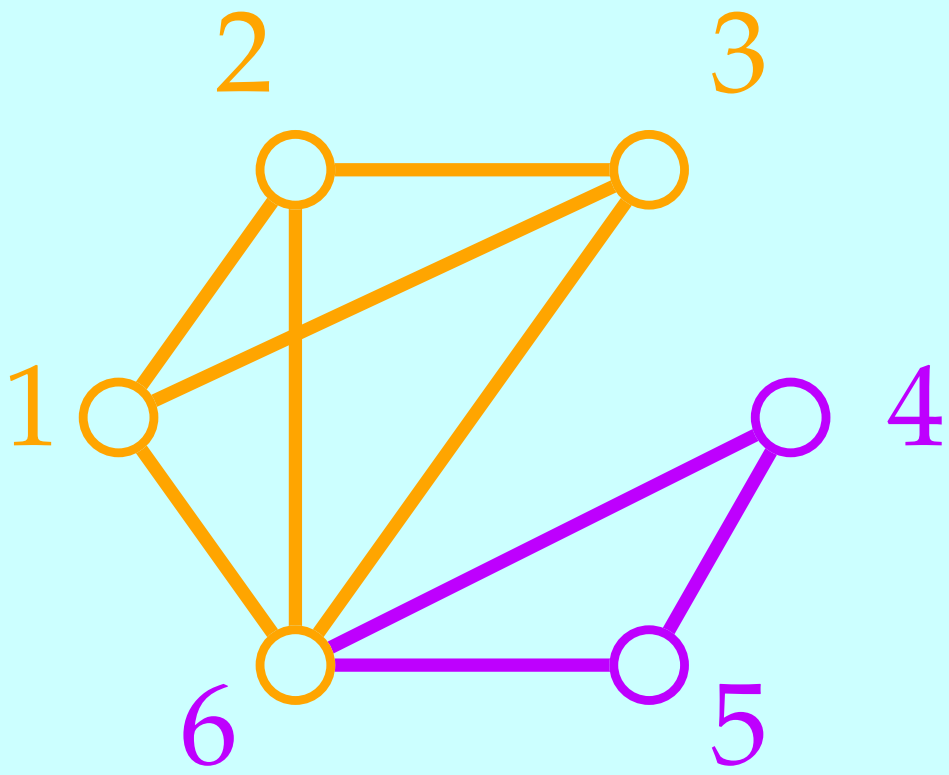
One of the cliques it contains



A maximal clique

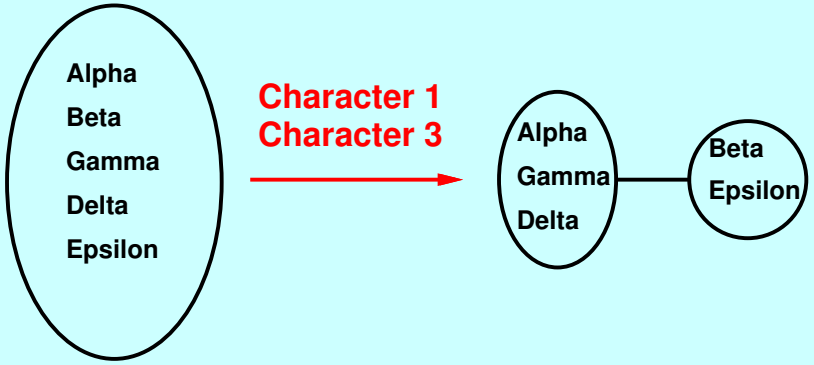


The largest maximal clique



Making the tree by “tree popping”

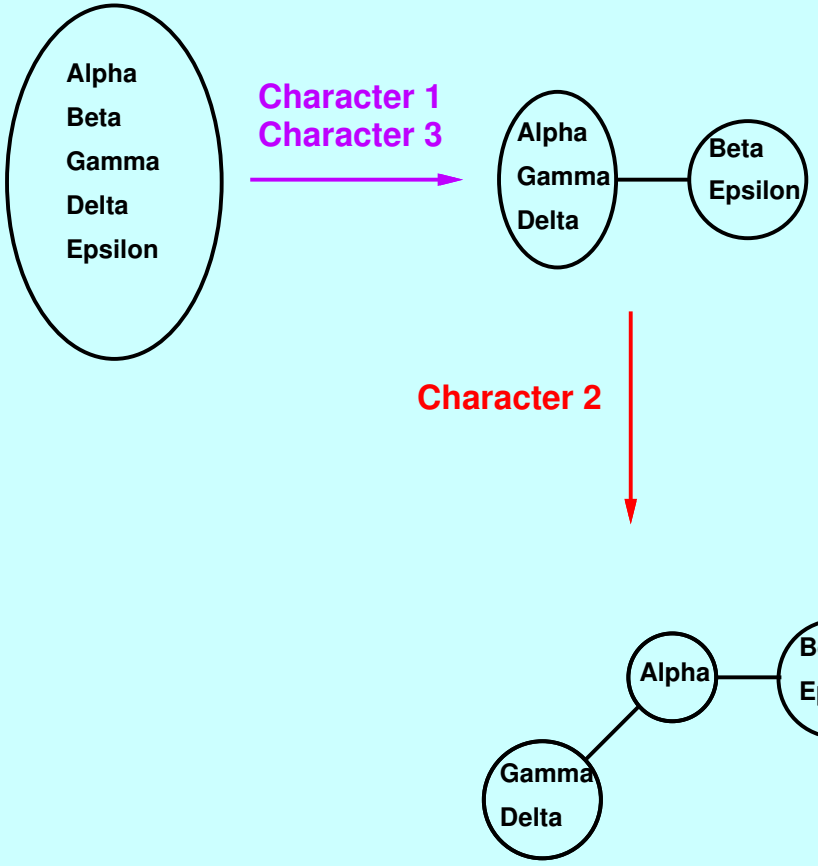
(for clique $\{1, 2, 3, 6\}$)



	1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0

Making the tree by “tree popping”

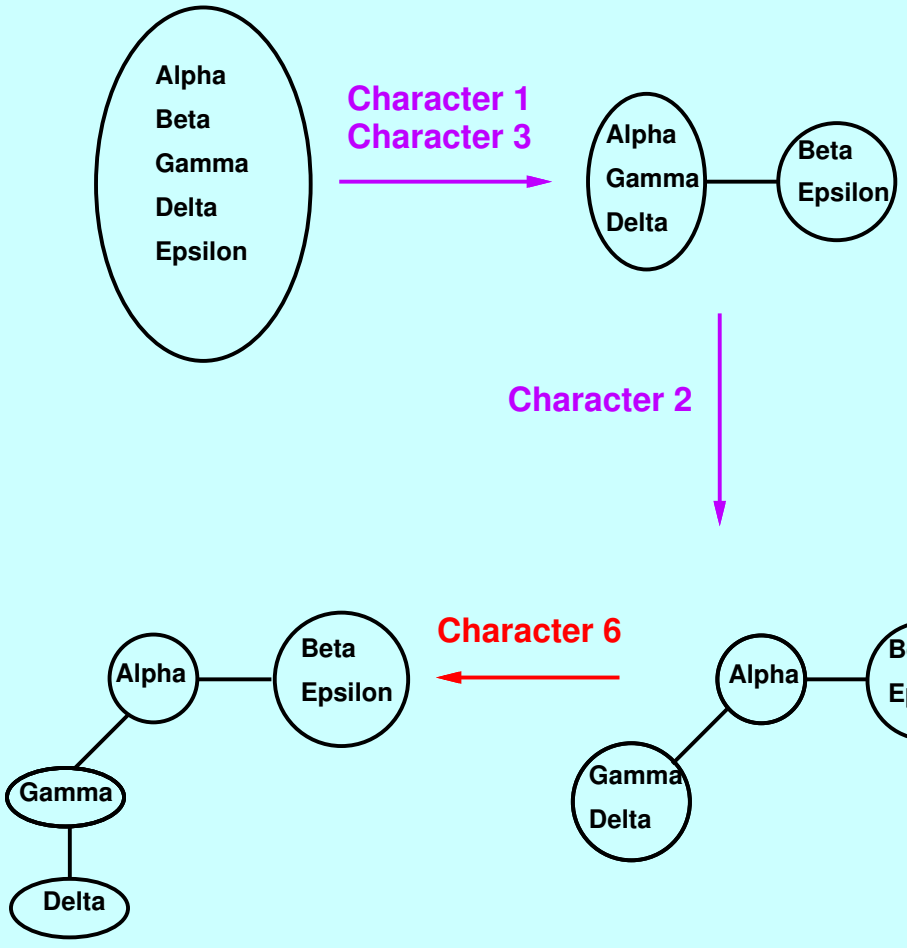
(for clique $\{1, 2, 3, 6\}$)



	1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0

Making the tree by “tree popping”

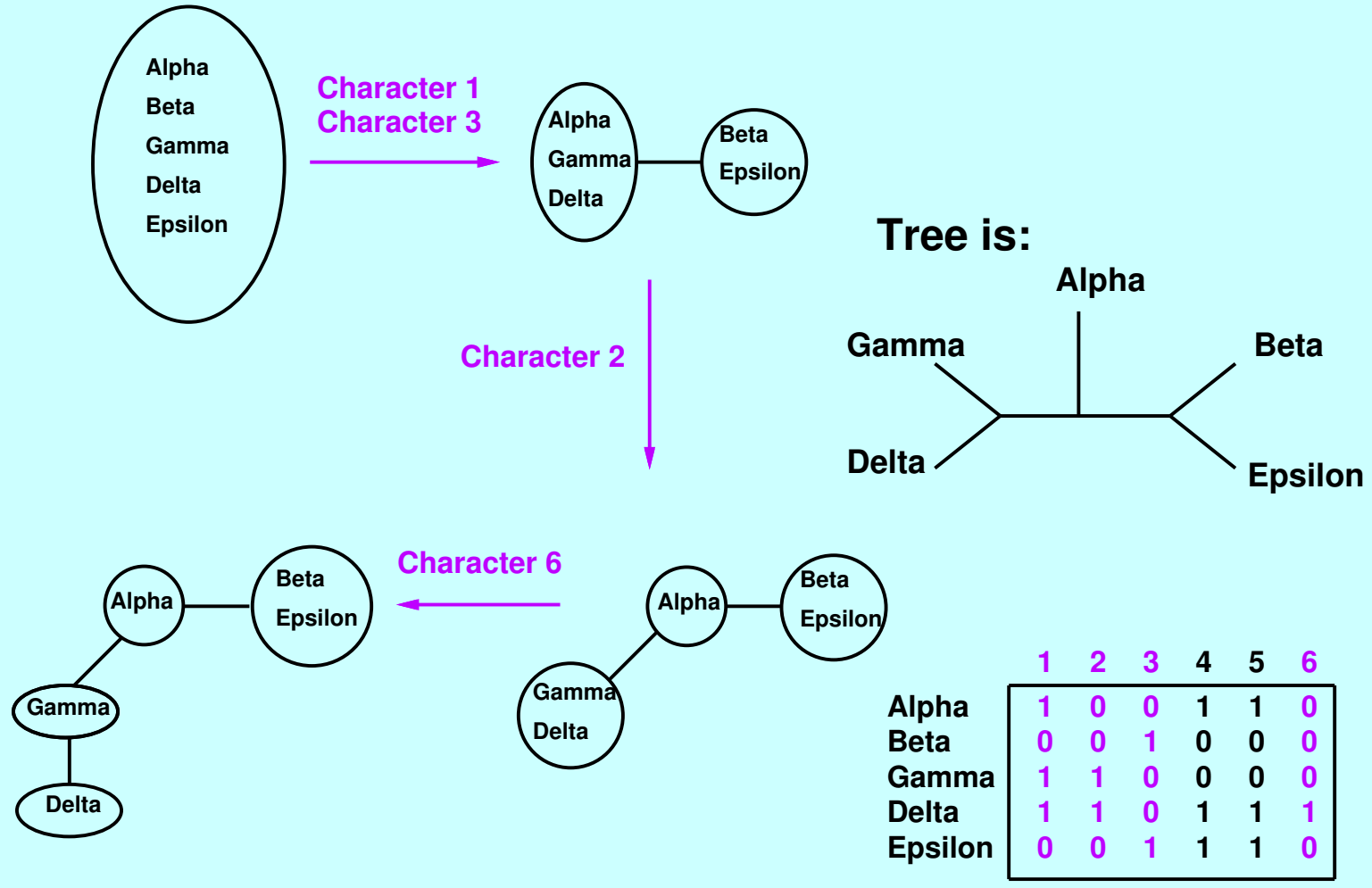
(for clique $\{1, 2, 3, 6\}$)



	1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0

Making the tree by "tree popping"

(for clique $\{1, 2, 3, 6\}$)



	1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0

Unknown states and compatibility

A data set that has all pairs of characters compatible, but which cannot have all characters compatible with the same tree. This violates the Pairwise Compatibility Theorem, owing to the unknown ("?") states.

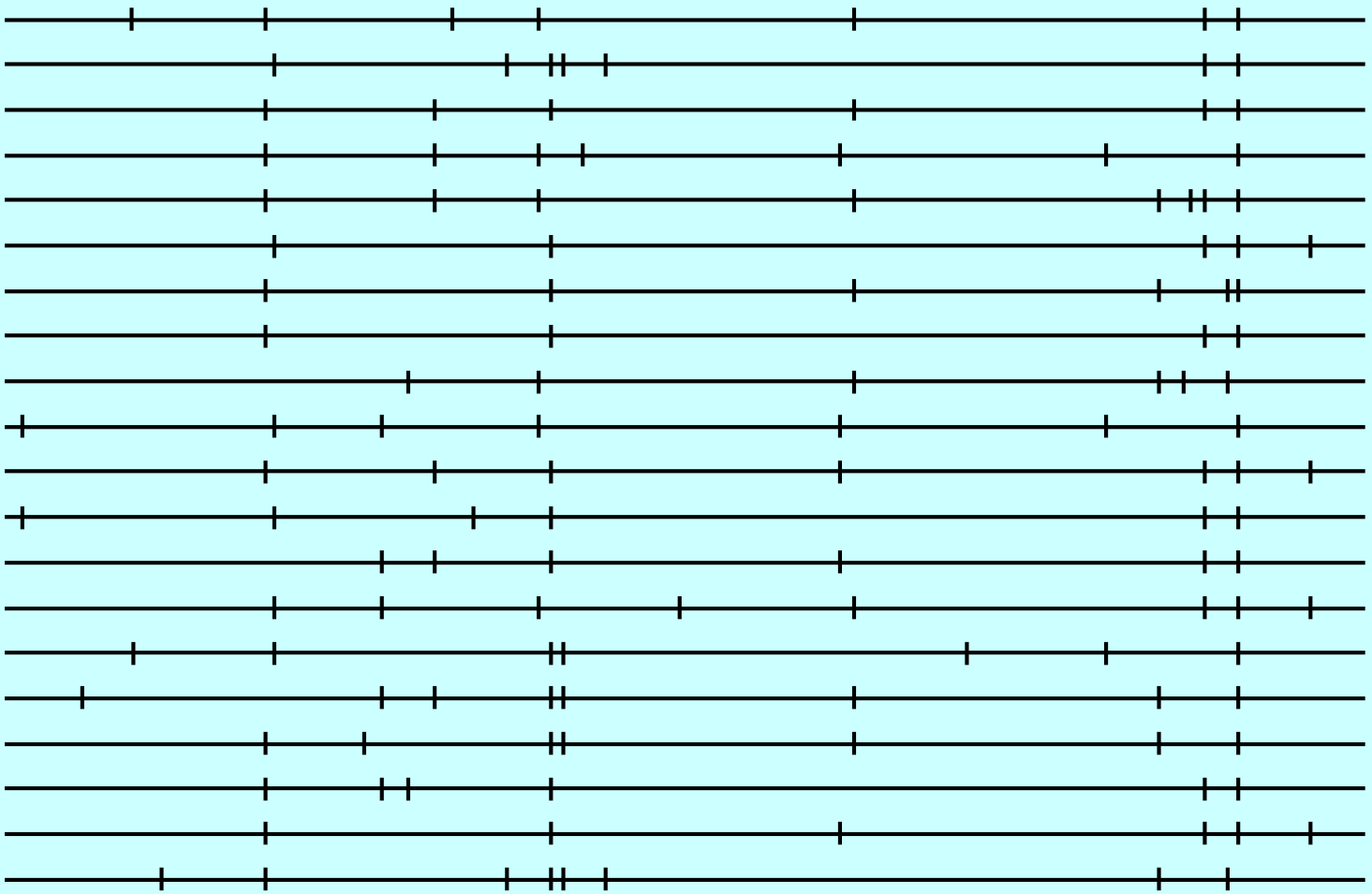
Alpha	0	0	0
Beta	?	0	1
Gamma	1	?	0
Delta	0	1	?
Epsilon	1	1	1

Multiple character states and compatibility

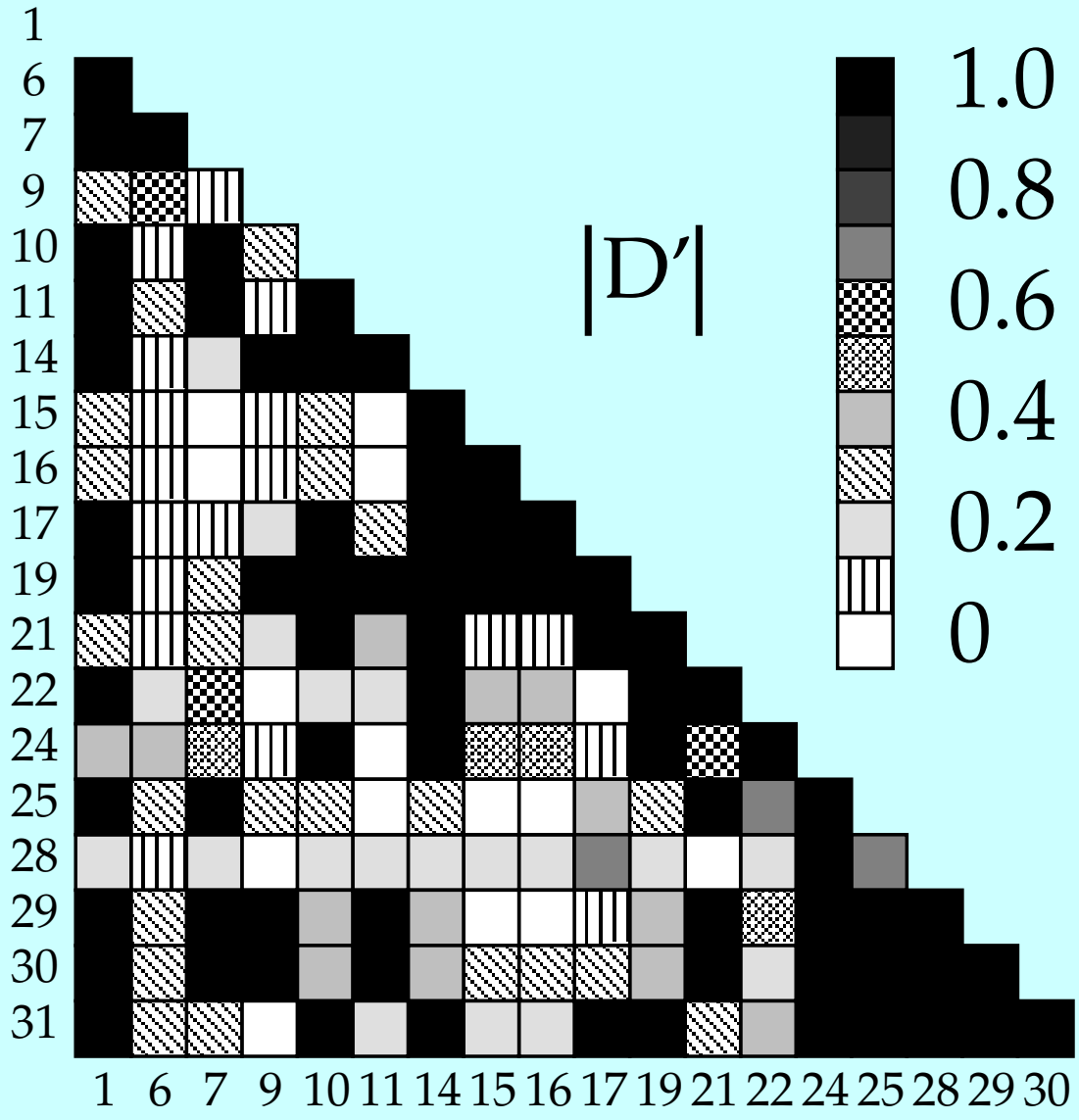
Walter Fitch's set of nucleotide sequences that have each pair of sites compatible, but which are not all compatible with the same tree.

Alpha	A	A	A
Beta	A	C	C
Gamma	C	G	C
Delta	C	C	G
Epsilon	G	A	G

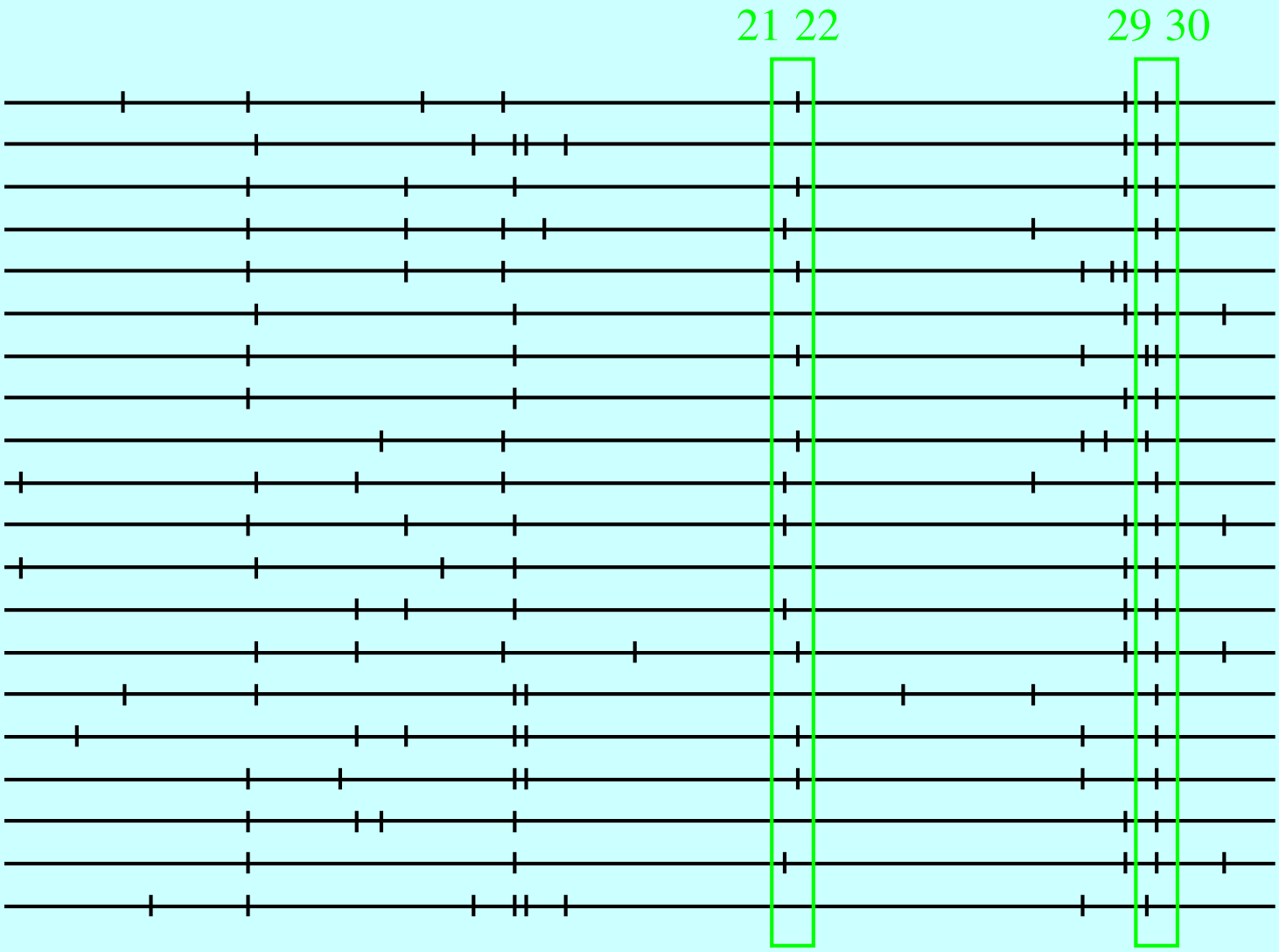
Haplotypes of SNPs in a simulated population



Values of the LD measure $|D'|$



Two of the pairs of SNPs that show $|D'| = 1$

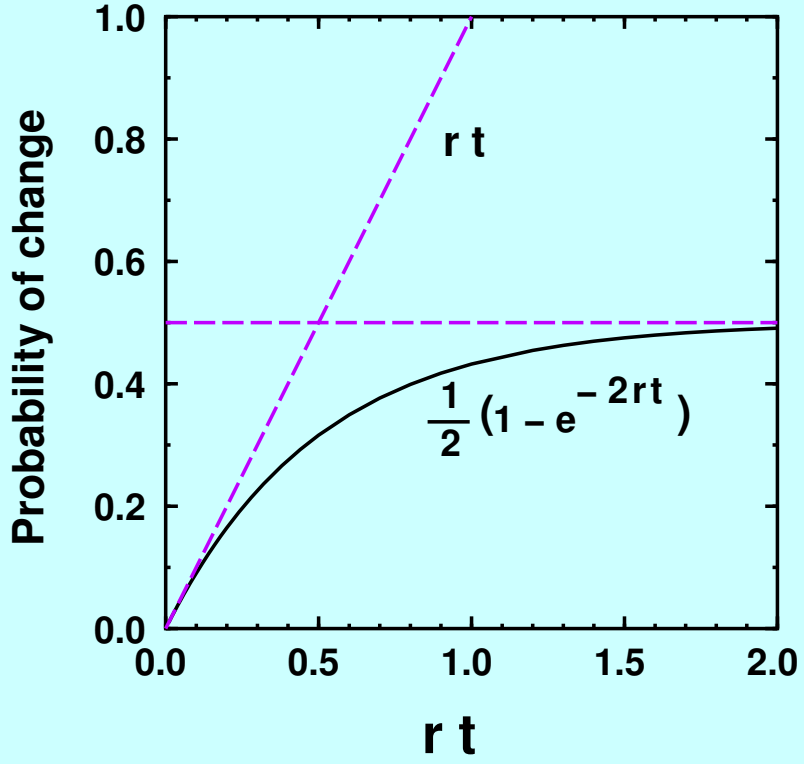


Transition probabilities in a two-state model

In a symmetric two-state model with rate of change r per unit time, where

$$0 \underset{r}{\overset{r}{\rightleftharpoons}} 1$$

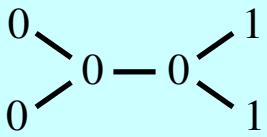
we have $\text{Prob}\,(1 \mid 0, t, r) = \frac{1}{2}\left(1 - e^{-2rt}\right)$



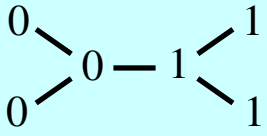
When rt is small then to very good approximation the probability of the events in the branch is rt or $1 - rt$. The latter is nearly 1.

Likelihood and parsimony

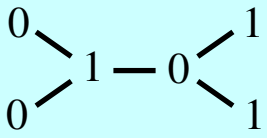
Approximating all the terms $1 - r\,t_i$ by 1, we find the probabilities of these data for the four possible choices of interior node states:



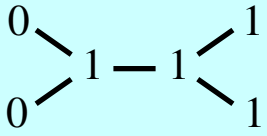
$$\tfrac{1}{2}(r\,t_3)(r\,t_4)$$



$$\tfrac{1}{2}(r\,t_5)$$



$$\tfrac{1}{2}(r\,t_1)(r\,t_2)(r\,t_5)(r\,t_3)(r\,t_4)$$



$$\tfrac{1}{2}(r\,t_1)(r\,t_2)$$

Likelihood and parsimony

$$\begin{aligned} L &= \text{Prob}\,(\text{Data}|\text{Tree}) \\ &= \prod_{i=1}^{\text{chars}} \sum_{\substack{\text{recon}- \\ \text{structions}}} \left(\frac{1}{2} \prod_{j=1}^{B} \begin{cases} r_i t_j & \text{if this character changes} \\ 1 - r_i t_j & \text{if it does not change} \end{cases} \right) \end{aligned}$$

The sum is over all reconstructions of changes in that character that result in the observed states at the tips of the tree. It is *not* just over all most parsimonious reconstructions of changes.

We will see (later in the course) that maximizing the likelihood is a Good Thing. We want to show that in the case where changes are infrequent, there is a relationship between that and minimizing the (weighted) parsimony score, and what the proper weights turn out to be.

approximating ...

If we toss out, in each character, all the terms that have more than the smallest possible number of terms rt, for each character (i) the likelihood is approximately the product over all branches (the jth of which has length t_j)

$$\frac{1}{2}\prod_{i=1}^{B}(r_i t_j)^{n_{ij}}\,.$$

where n_{ij} is either 0 or 1, for all characters the likelihood is then approximately

$$L \simeq \prod_{i=1}^{\text{chars}}\left(\frac{1}{2}\prod_{j=1}^{\text{branches}}(r_i t_j)^{n_{ij}}\right).$$

Tossing out the $1/2$ terms and taking logarithms, if the number of changes in character i in branch j is n_{ij},

$$-\ln L \simeq \sum_{i=1}^{\text{chars}}\sum_{j=1}^{\text{branches}} n_{ij}\left(-\ln\left(r_i t_j\right)\right).$$

Weights and likelihood

Probabilities of change and resulting weights for an imaginary case.

Character	r_i t_j	changes	total weight
1	0.01	1	4.605
2	0.01	2	9.210
3	0.00001	1	11.519

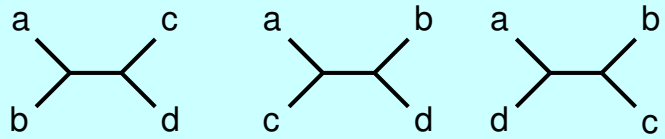
with one-tenth as much change ...

Character	r_i t_j	changes	total weight
1	0.001	1	6.908
2	0.001	2	13.816
3	0.000001	1	13.816

... and with one-tenth less again

Character	r_i t_j	changes	total weight
1	0.0001	1	9.210
2	0.0001	2	18.421
3	0.0000001	1	16.118

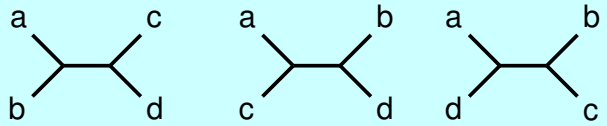
Trees, steps, and patterns



0000	0	0	0
0001	1	1	1
0010	1	1	1
0011	1	2	2
0100	1	1	1
0101	2	1	2
0110	2	2	1
0111	1	1	1
1000	1	1	1
1001	2	2	1
1010	2	1	2
1011	1	1	1
1100	1	2	2
1101	1	1	1
1110	1	1	1
1111	0	0	0

This table shows how many changes of state (steps) are needed for each of the 16 possible character patterns on each of the three unrooted tree topologies, under a parsimony criterion.

Trees, steps, and patterns with DNA



AAAA	0	0	0
AAAC	1	1	1
AAAG	1	1	1
AAAT	1	1	1
AACA	1	1	1
AACC	1	2	2
AACG	2	2	2
AACT	2	2	2
AAGA	1	1	1
AAGC	2	2	2
AAGG	1	2	2
AAGT	2	2	2
AATA	1	1	1
AATC	2	2	2
AATG	2	2	2
AATT	1	2	2
ACAA	1	1	1
ACAC	2	1	2
ACAG	2	2	2
ACAT	2	2	2
ACCA	2	2	1
ACCC	1	1	1
⋮		⋮	
TTTT	0	0	0

Parsimony and patterns

Ignoring all the patterns that have the same number of steps on all topologies, the ones that matter to a parsimony method have one of these three sums:

$$n_{xxyy} + 2n_{xyxy} + 2n_{xyyx} = 2(n_{xxyy} + n_{xyxy} + n_{xyyx}) - n_{xxyy}$$

$$2n_{xxyy} + n_{xyxy} + 2n_{xyyx} = 2(n_{xxyy} + n_{xyxy} + n_{xyyx}) - n_{xyxy}$$

$$2n_{xxyy} + 2n_{xyxy} + n_{xyyx} = 2(n_{xxyy} + n_{xyxy} + n_{xyyx}) - n_{xyyx}.$$