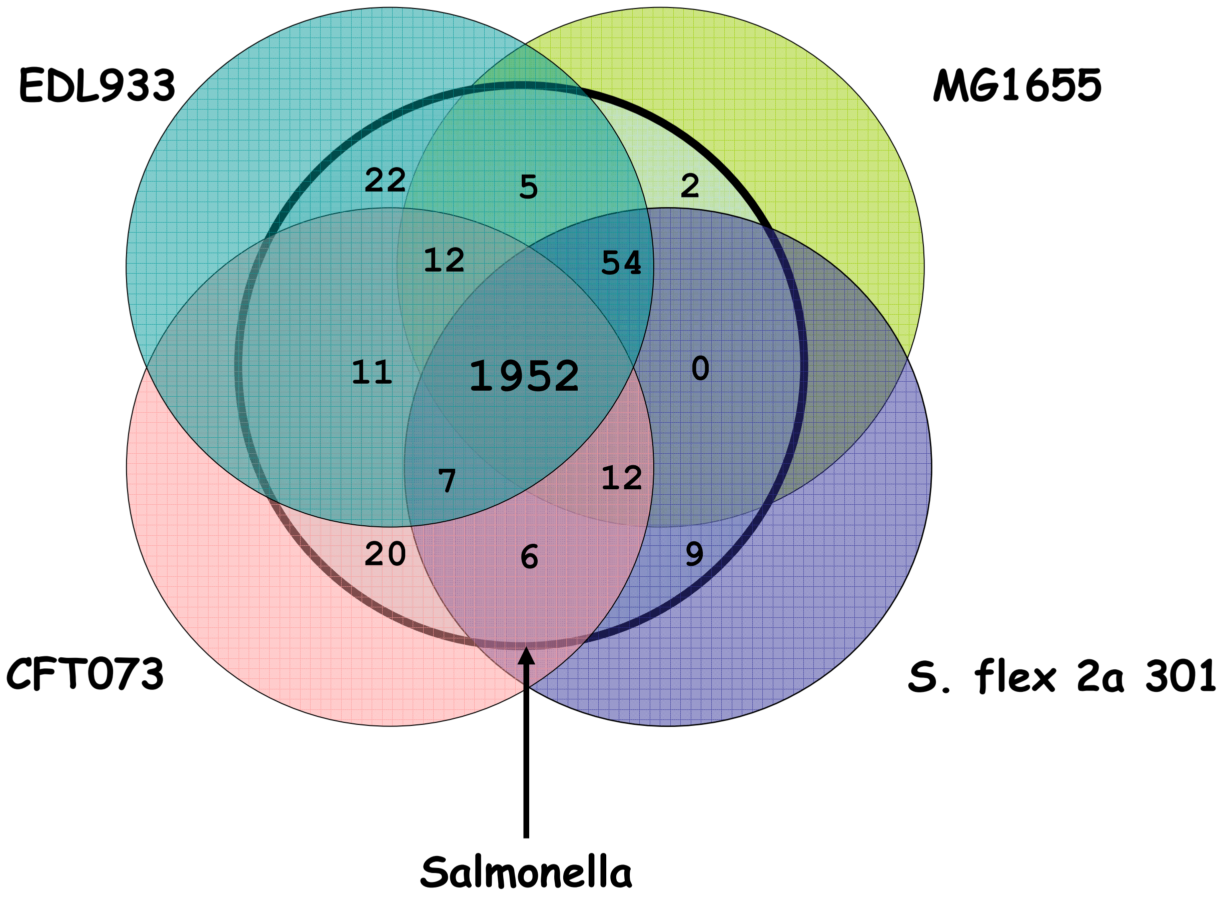
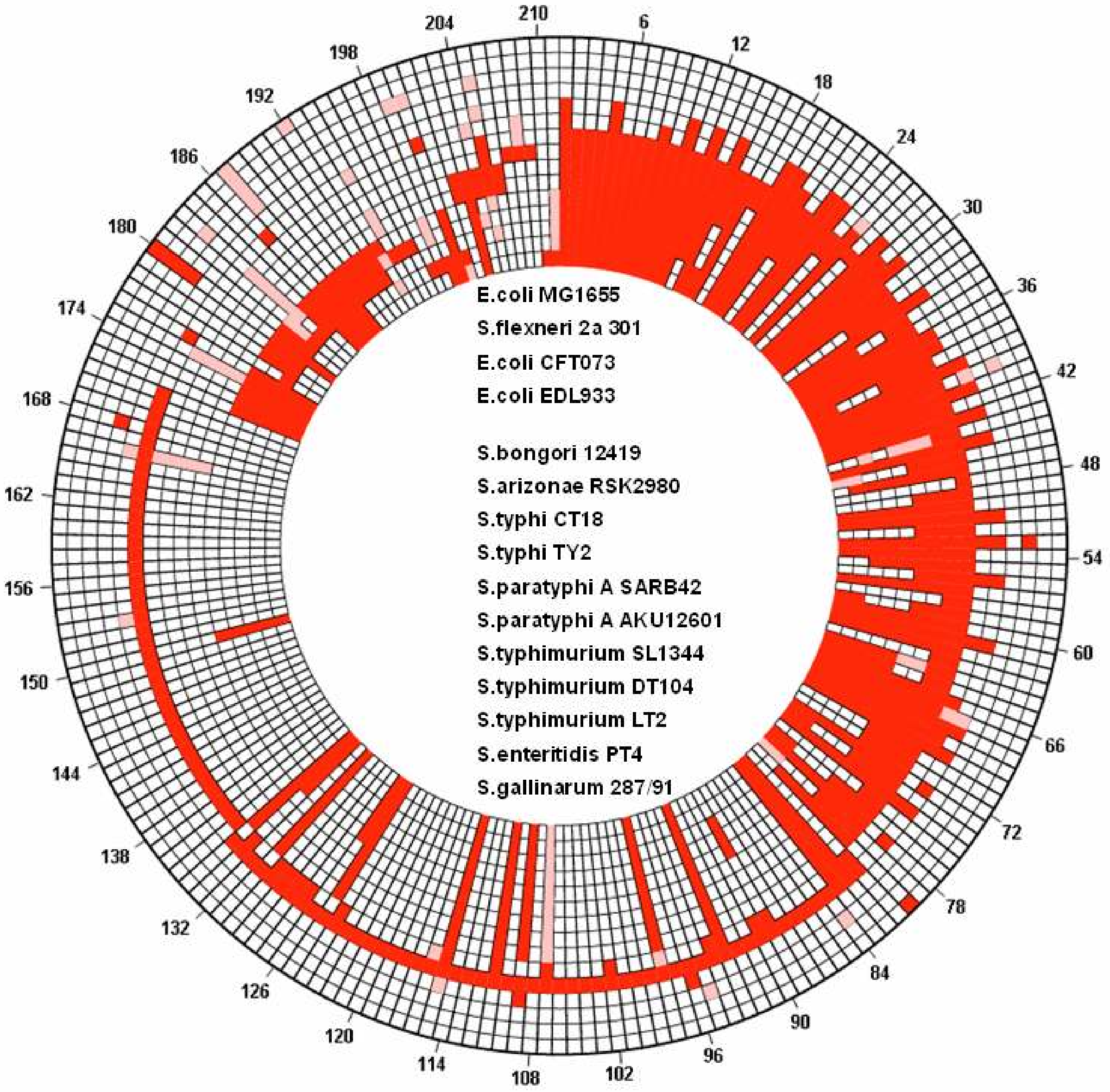
Horizontal Gene Transfer over Time

Relative

Core Gene Dataset (Venn)



Mobilome: Circular Diagram of Horizontally Acquired DNA



Methods for Comparative Analysis

Method	Genome coverage (%)	Core genes	Dispensable genes
16s rRNA	0.07[a]	Yes	No
MLST	0.2[a]	Yes	No
SNPs	2[b]	Yes	Yes
Whole-genome	100	Yes	Yes

Estimates have been calculated based on:

- [a]*Neisseria meningitidis*: genome size ~2.2 Mb (Bentley *et al.*, 2007)
    - 16S rRNA length ~1.5kb (Sacchi *et al.*, 2002)
    - length of MLST loci ~4kb (Maiden *et al.*, 1998)
- [b]*Salmonella typhi*: genome size ~4.8 Mb (Deng *et al.*, 2003)
    - SNPs on gene fragments covering ~89 Kb (Roumagnac *et al.*, 2006)

Medini, 2008

MAUVE

Algorithm: MAUVE.

1. Identify multiple maximal unique matches (multi-MUMs), i.e. local alignments of exactly matching (single-copy) sequences that are shared between 2 or more chromosomes.
2. Calculate a phylogenetic guide tree based on the multi-MUMs sequences.
3. Partition a subset (anchors) of the multi-MUMs into LCBs.
4. Do recursive anchoring to identify new anchors within and outside the LCBs.
5. Align each LCB based on the guide tree.

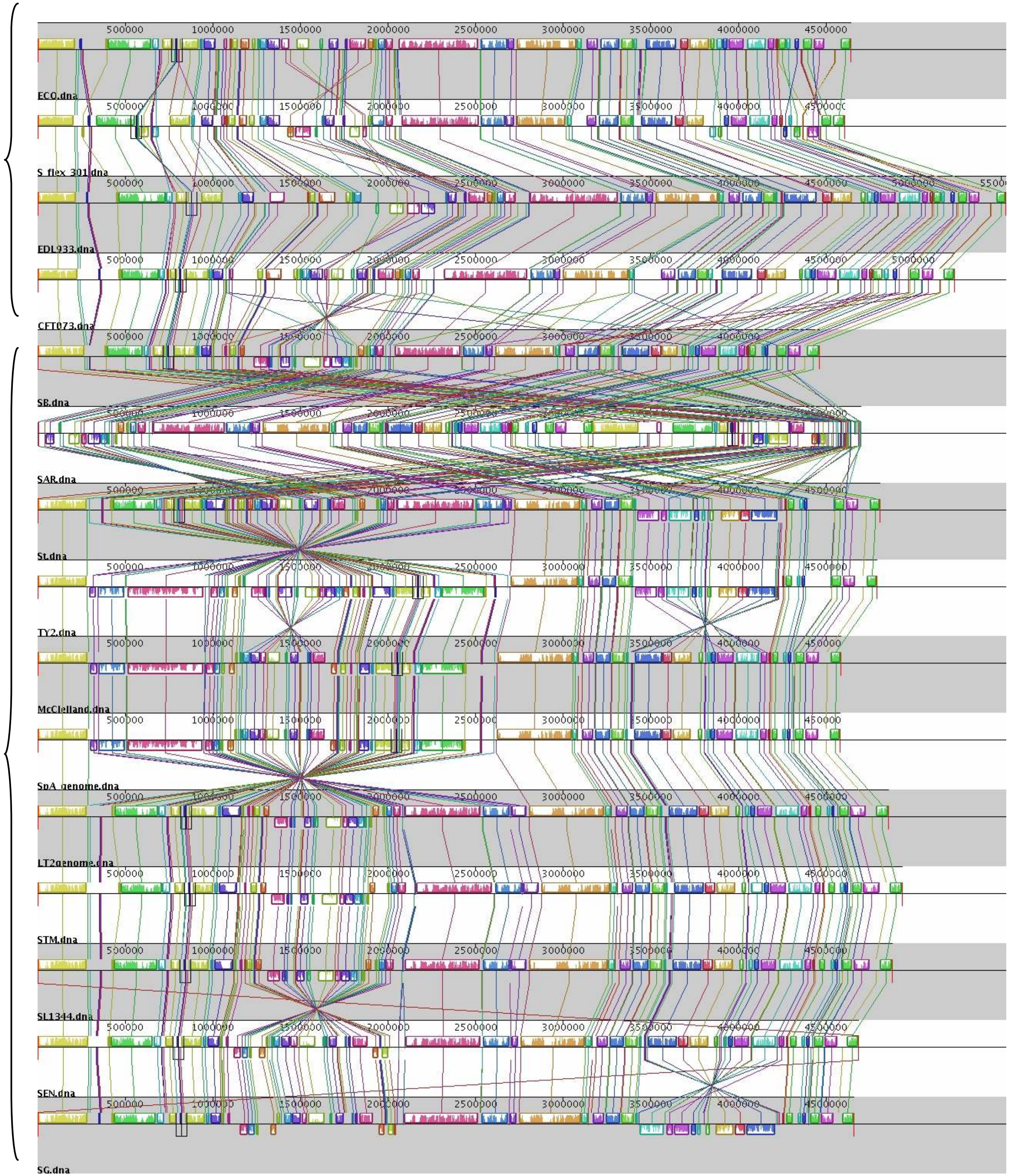
Note: Formally, an LCB is a sequence of multi-MUMs that satisfies a total ordering property, such that the left end of the *i*th multi-MUM occurs before the left end of the *i*+1 multi-MUM, for all multi-MUMs in the LCB and for all the genomes compared.

Source: (Darling et al., 2004).

MAUVE (15 Genomes)

E. coli (outgroup)

Salmonella



Tree Building Methods

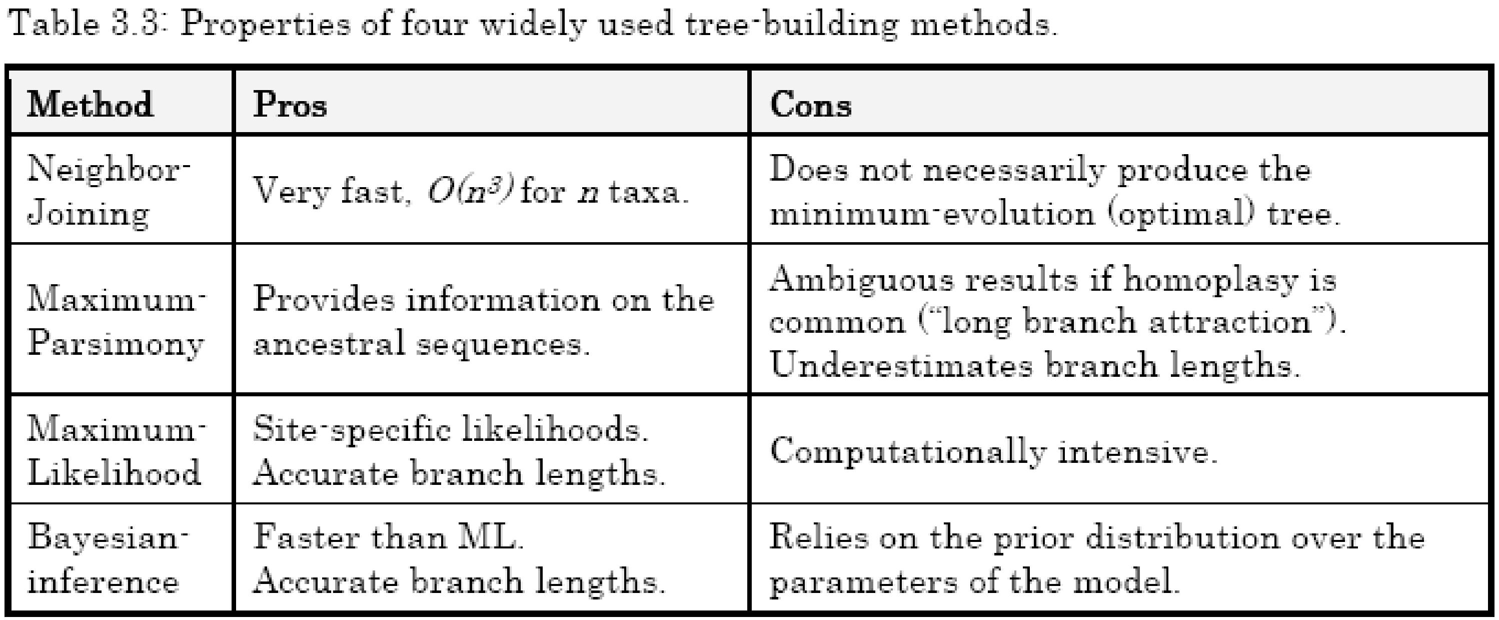
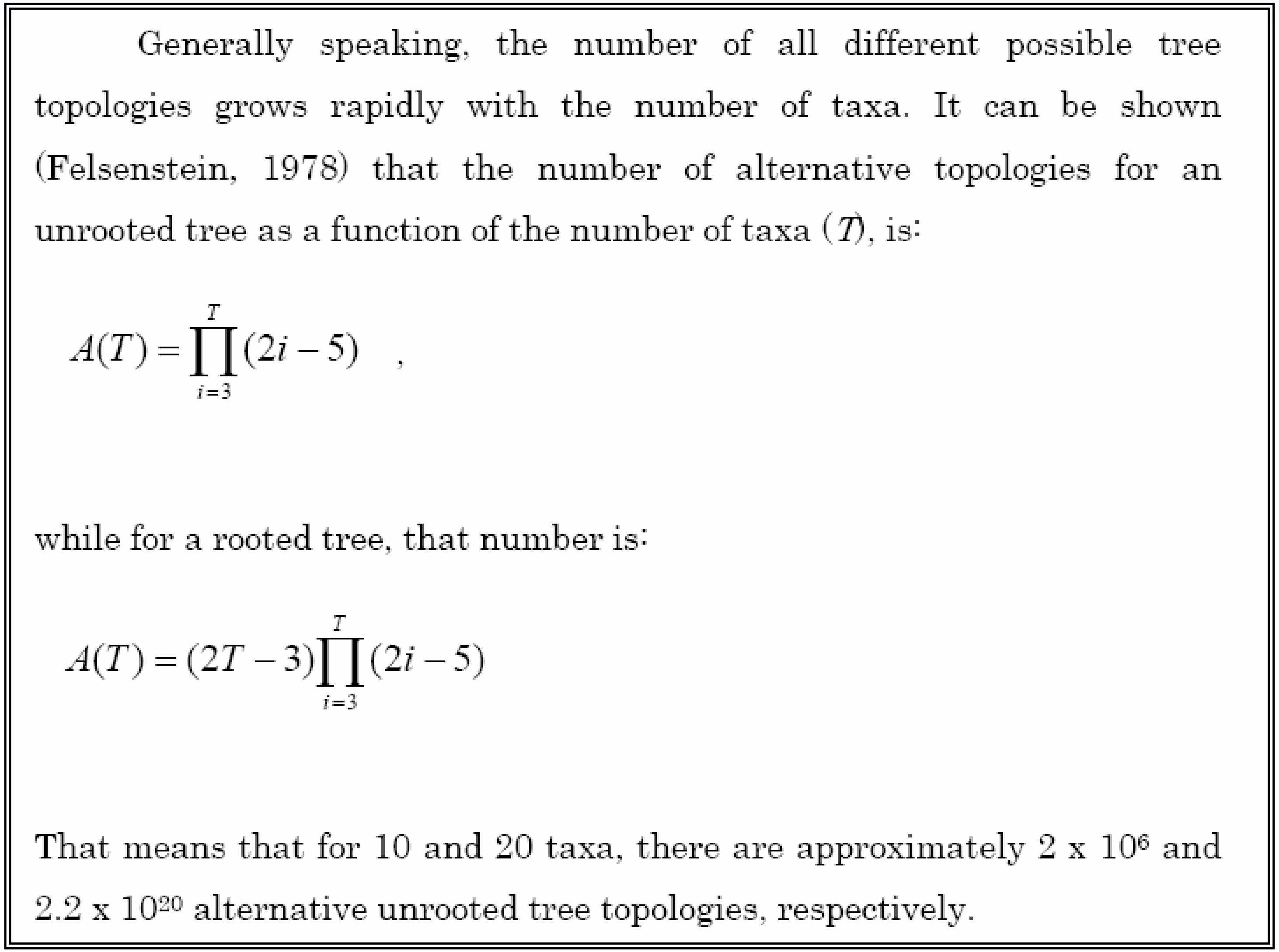


Table 3.3: Properties of four widely used tree-building methods.

Method	Pros	Cons
Neighbor-Joining	Very fast, $O(n^3)$ for n taxa.	Does not necessarily produce the minimum-evolution (optimal) tree.
Maximum-Parsimony	Provides information on the ancestral sequences.	Ambiguous results if homoplasy is common ("long branch attraction"). Underestimates branch lengths.
Maximum-Likelihood	Site-specific likelihoods. Accurate branch lengths.	Computationally intensive.
Bayesian-inference	Faster than ML. Accurate branch lengths.	Relies on the prior distribution over the parameters of the model.

Number of Tree Topologies (1)



Generally speaking, the number of all different possible tree topologies grows rapidly with the number of taxa. It can be shown (Felsenstein, 1978) that the number of alternative topologies for an unrooted tree as a function of the number of taxa (*T*), is:

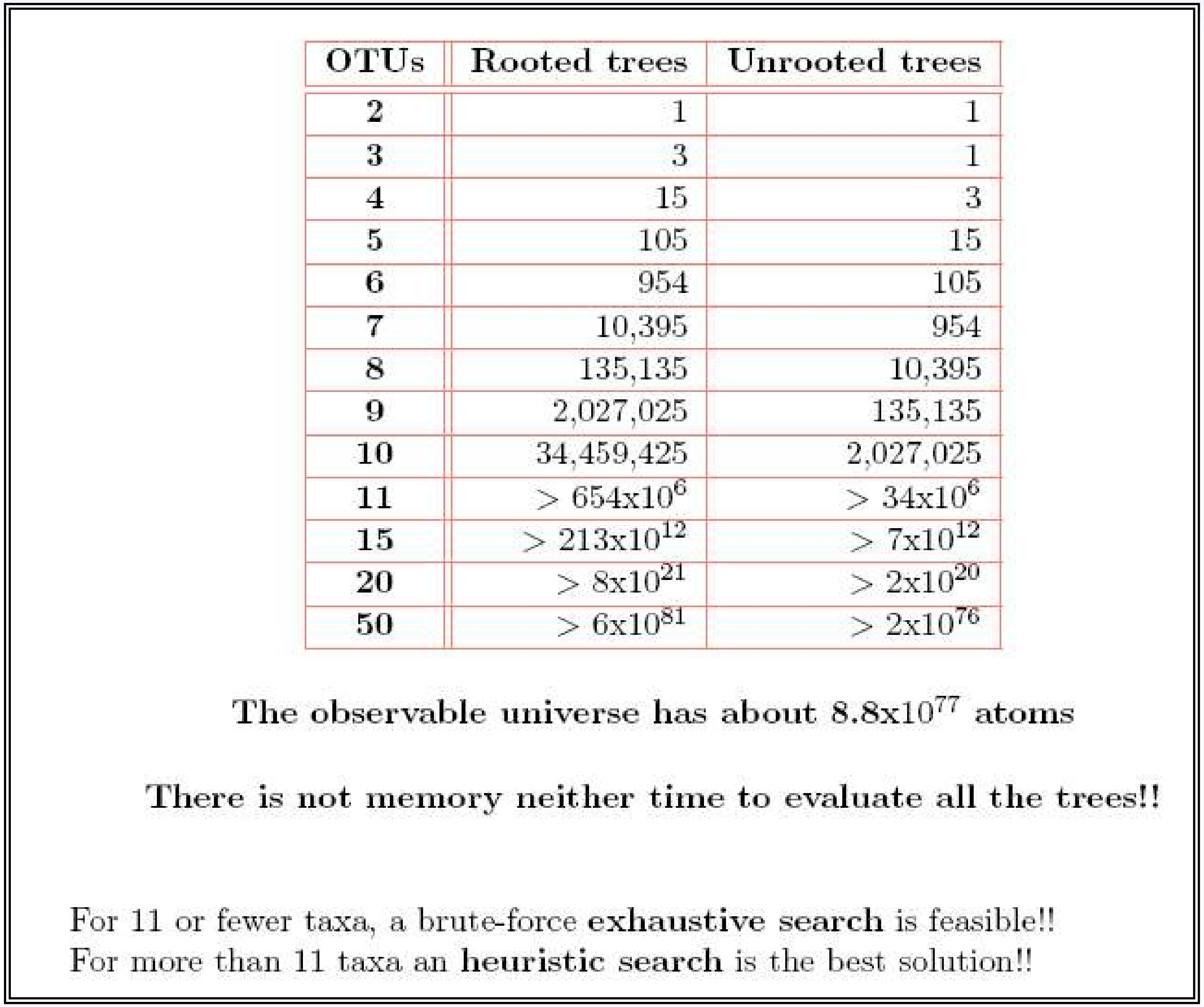
$$A(T) = \prod_{i=3}^{T} (2i-5) \quad ,$$

while for a rooted tree, that number is:

$$A(T) = (2T-3)\prod_{i=3}^{T} (2i-5)$$

That means that for 10 and 20 taxa, there are approximately 2 x 10^6 and 2.2 x 10^{20} alternative unrooted tree topologies, respectively.

Number of Tree Topologies (2)



OTUs	Rooted trees	Unrooted trees
2	1	1
3	3	1
4	15	3
5	105	15
6	954	105
7	10,395	954
8	135,135	10,395
9	2,027,025	135,135
10	34,459,425	2,027,025
11	$> 654\text{x}10^{6}$	$> 34\text{x}10^{6}$
15	$> 213\text{x}10^{12}$	$> 7\text{x}10^{12}$
20	$> 8\text{x}10^{21}$	$> 2\text{x}10^{20}$
50	$> 6\text{x}10^{81}$	$> 2\text{x}10^{76}$

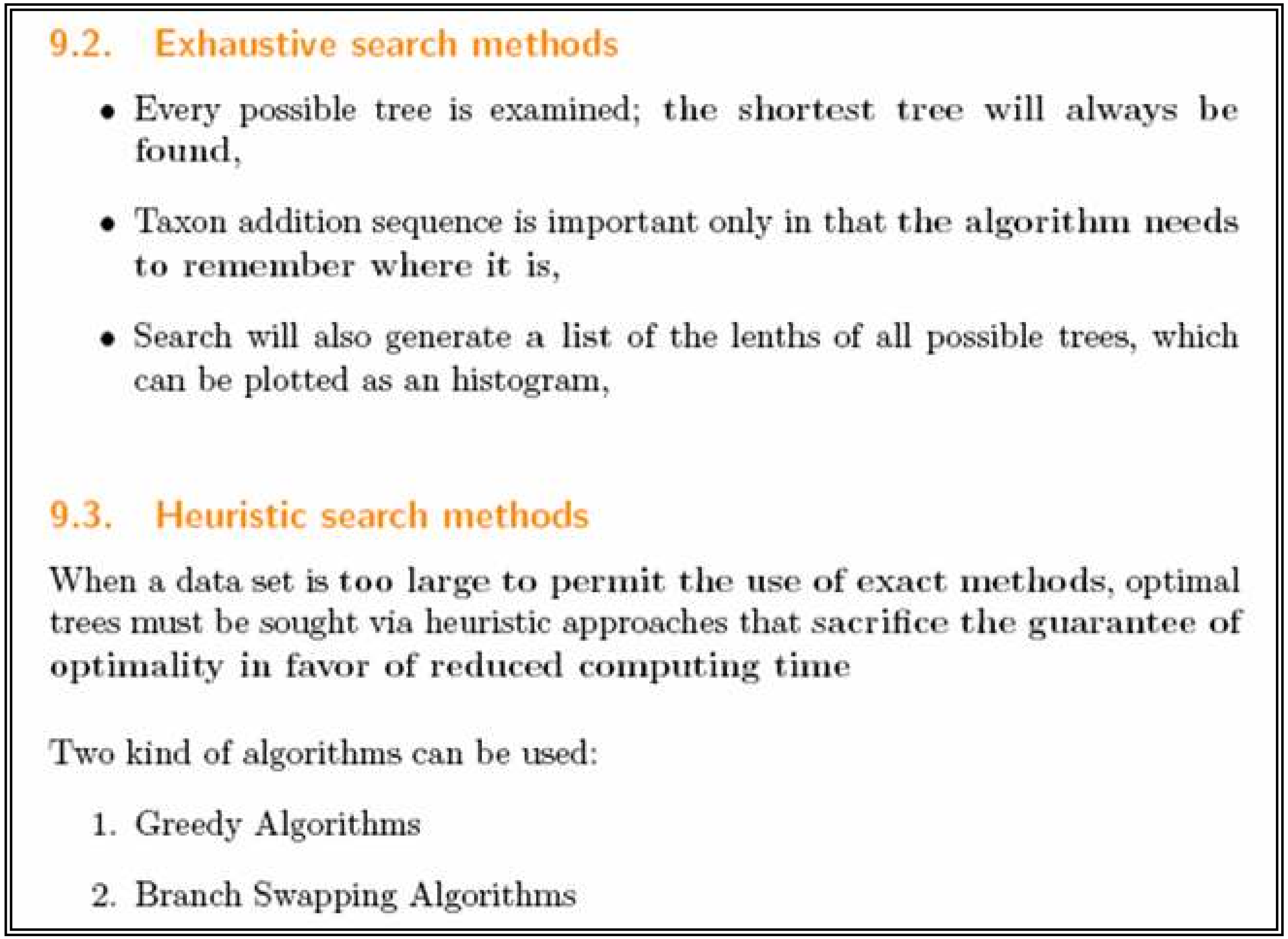
The observable universe has about 8.8x10^{77} atoms

There is not memory neither time to evaluate all the trees!!

For 11 or fewer taxa, a brute-force **exhaustive search** is feasible!!
For more than 11 taxa an **heuristic search** is the best solution!!

Dopazo 2006

Searching Tree Topologies (1)



9.2. Exhaustive search methods

- Every possible tree is examined; the shortest tree will always be found,
- Taxon addition sequence is important only in that the algorithm needs to remember where it is,
- Search will also generate a list of the lenths of all possible trees, which can be plotted as an histogram,

9.3. Heuristic search methods

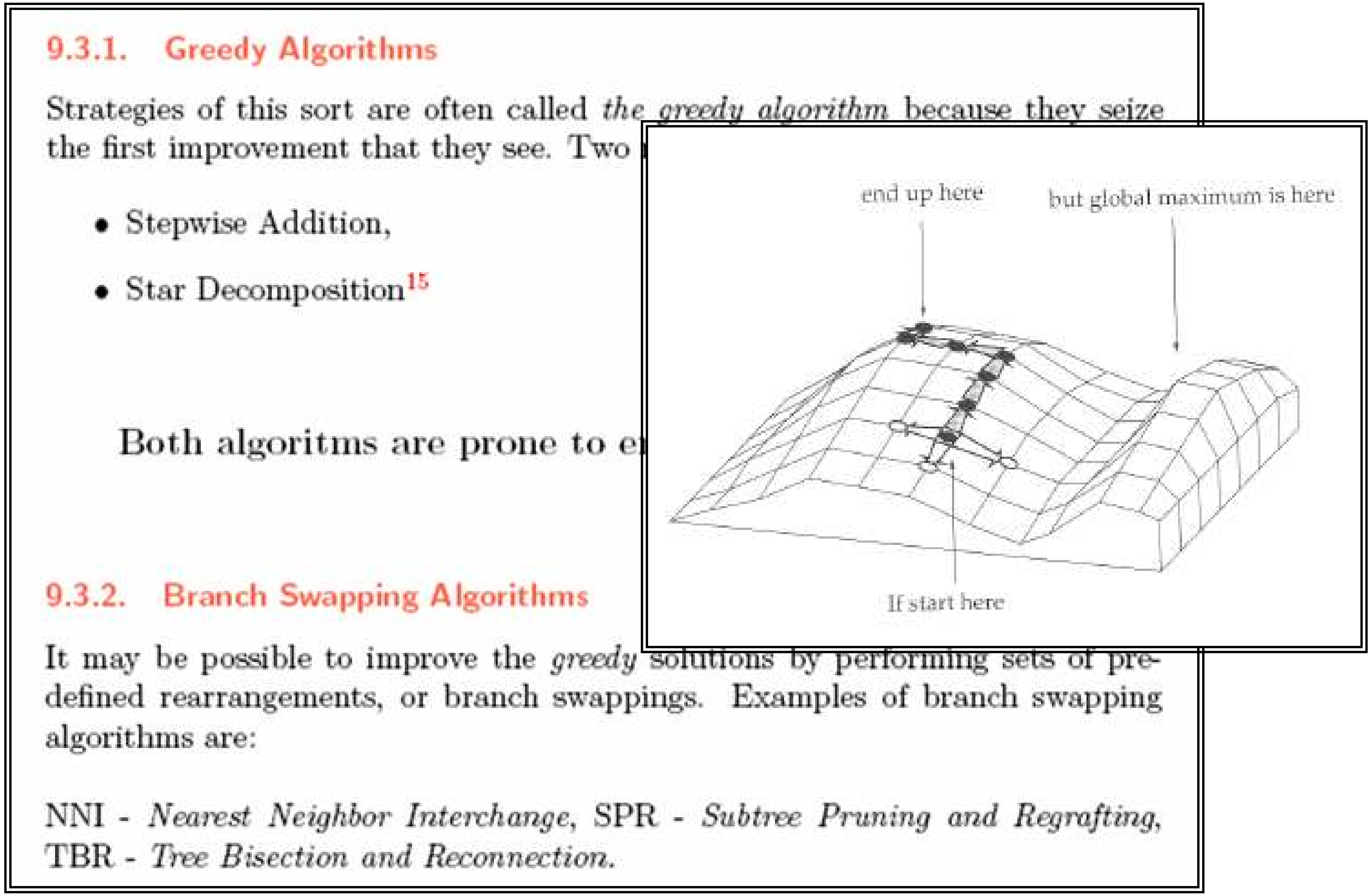
When a data set is too large to permit the use of exact methods, optimal trees must be sought via heuristic approaches that sacrifice the guarantee of optimality in favor of reduced computing time

Two kind of algorithms can be used:

1. Greedy Algorithms
2. Branch Swapping Algorithms

Dopazo 2006

Searching Tree Topologies (2)



9.3.1. Greedy Algorithms

Strategies of this sort are often called *the greedy algorithm* because they seize the first improvement that they see. Two

- Stepwise Addition,
- Star Decomposition[15]

Both algoritms are prone to e

end up here
but global maximum is here
If start here

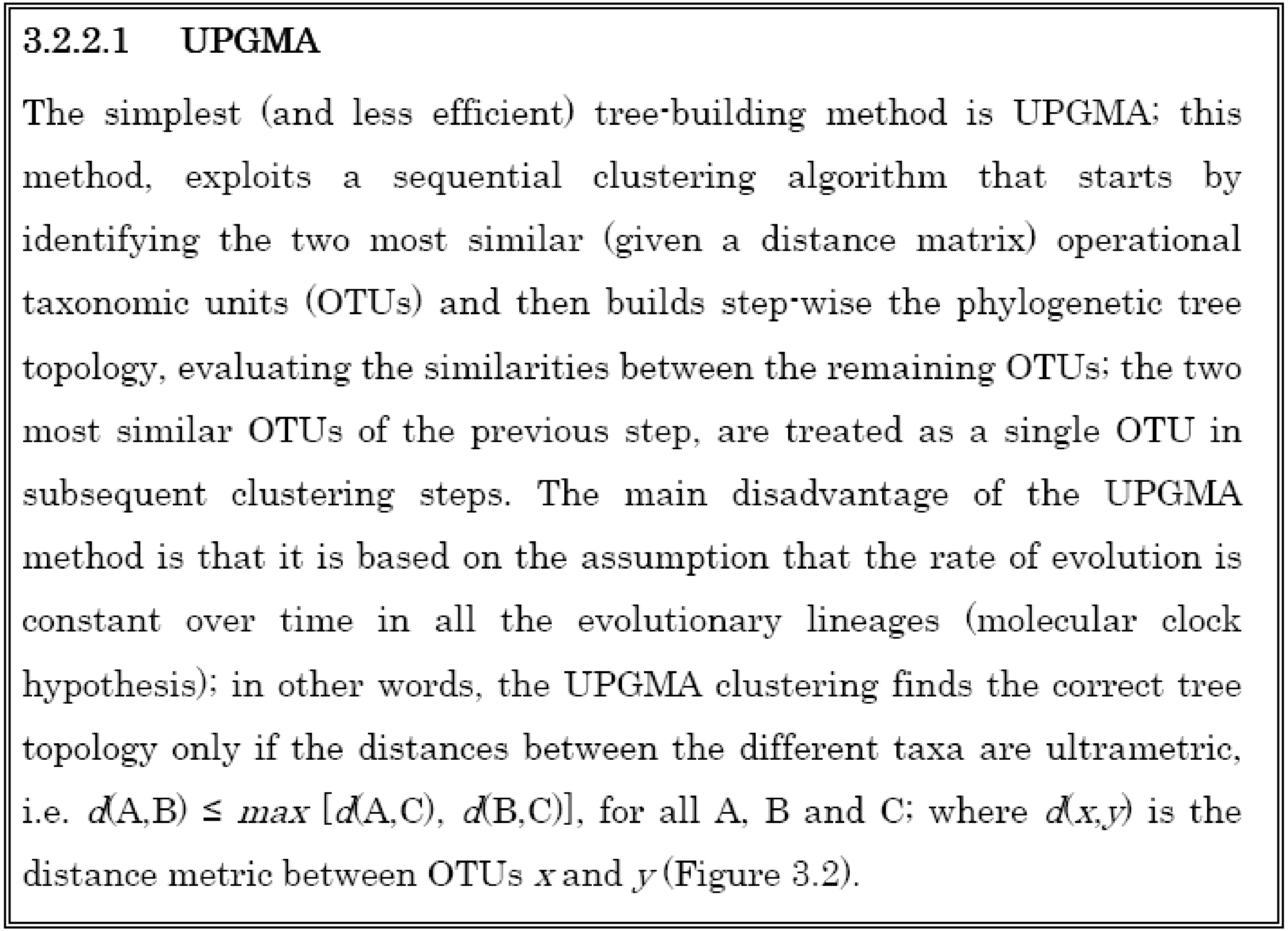
9.3.2. Branch Swapping Algorithms

It may be possible to improve the *greedy* solutions by performing sets of predefined rearrangements, or branch swappings. Examples of branch swapping algorithms are:

NNI - *Nearest Neighbor Interchange*, SPR - *Subtree Pruning and Regrafting*, TBR - *Tree Bisection and Reconnection*.

Dopazo 2006

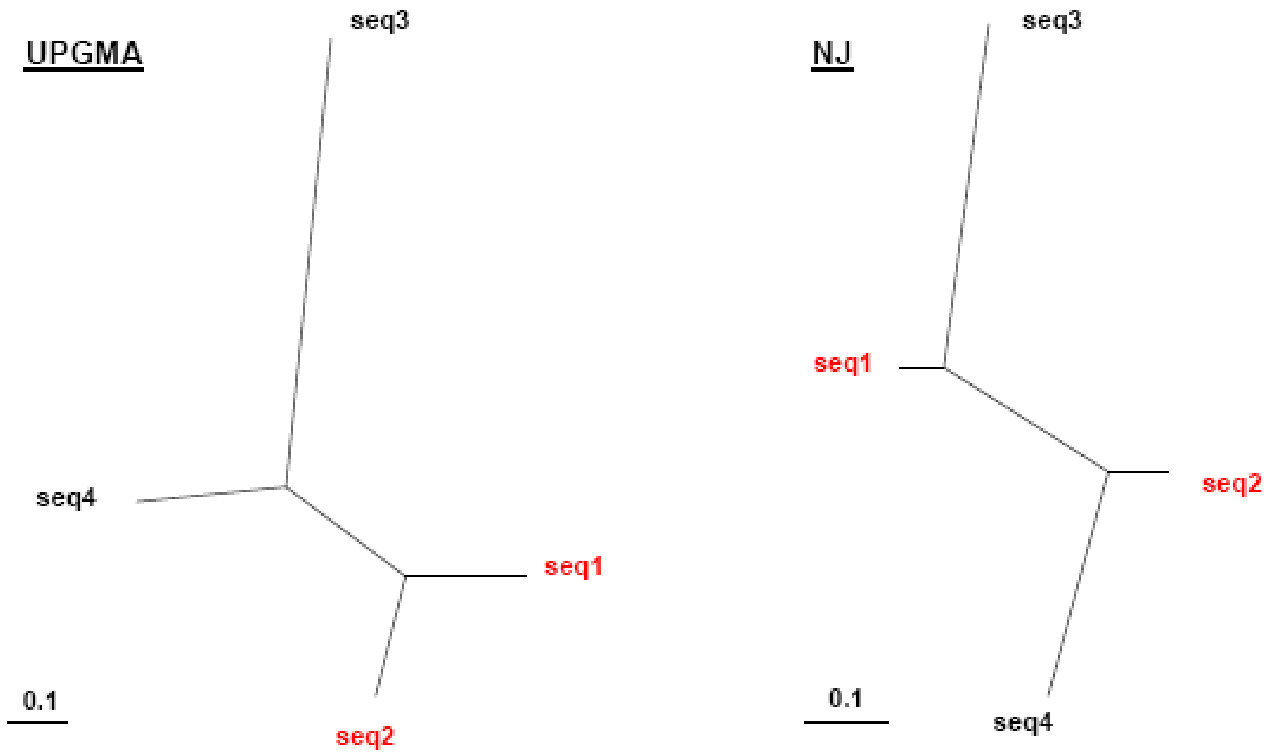
UPGMA: unweighted pair-group method with arithmetic mean



3.2.2.1 UPGMA

The simplest (and less efficient) tree-building method is UPGMA; this method, exploits a sequential clustering algorithm that starts by identifying the two most similar (given a distance matrix) operational taxonomic units (OTUs) and then builds step-wise the phylogenetic tree topology, evaluating the similarities between the remaining OTUs; the two most similar OTUs of the previous step, are treated as a single OTU in subsequent clustering steps. The main disadvantage of the UPGMA method is that it is based on the assumption that the rate of evolution is constant over time in all the evolutionary lineages (molecular clock hypothesis); in other words, the UPGMA clustering finds the correct tree topology only if the distances between the different taxa are ultrametric, i.e. $d(\mathrm{A},\mathrm{B}) \leq \mathit{max}\,[d(\mathrm{A},\mathrm{C}), d(\mathrm{B},\mathrm{C})]$, for all A, B and C; where $d(x,y)$ is the distance metric between OTUs x and y (Figure 3.2).

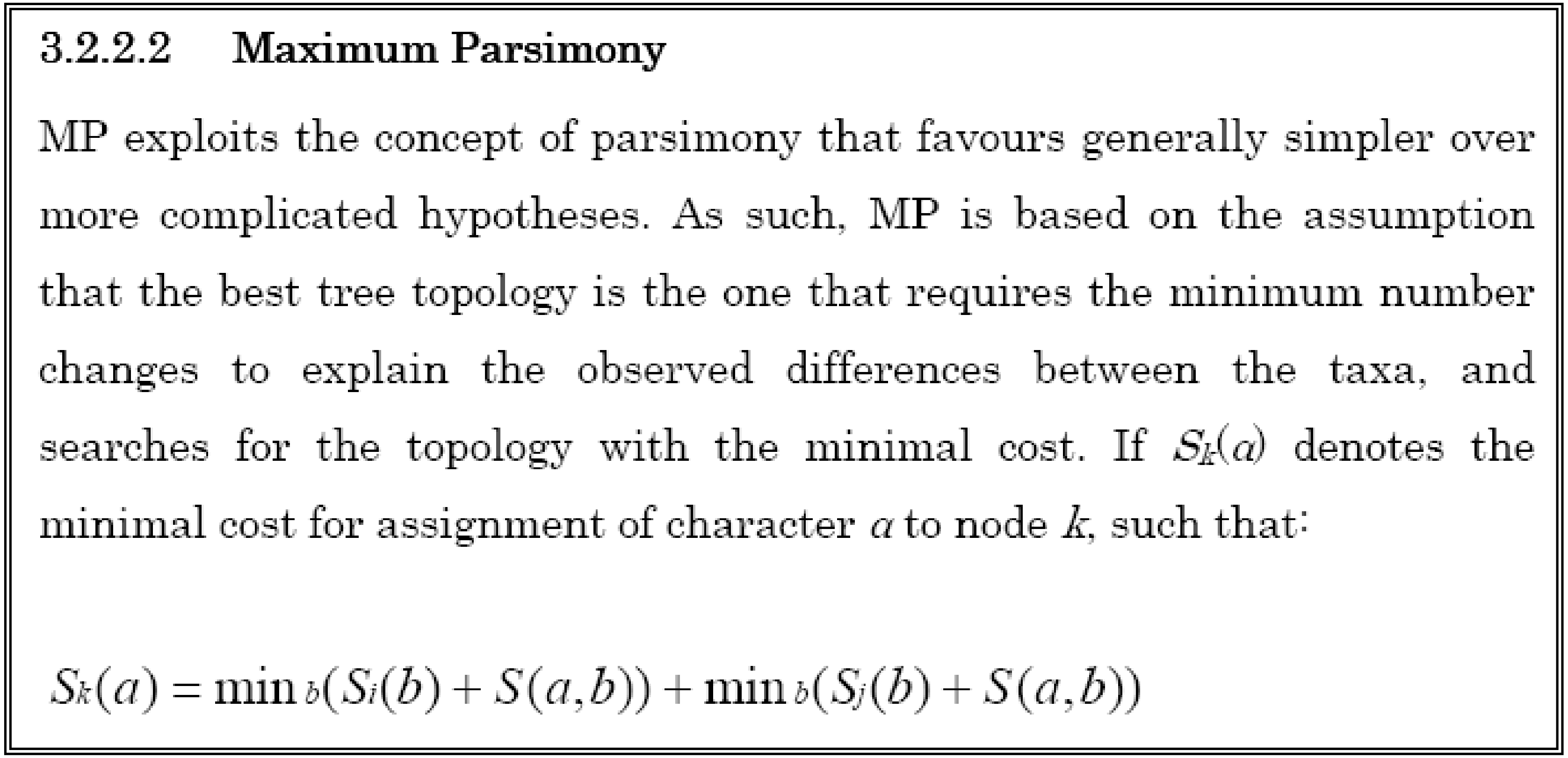
UPGMA: Pitfalls



```
>seq1
AAAAATTTTT
>seq2
GAAAATTAAA
>seq3
TTTTTTTTTT
>seq4
GAAAAGGGGA
```

	seq1	seq2	seq3	seq4
seq1	1.0	0.6	0.5	0.4
seq2	0.6	1.0	0.2	0.6
seq3	0.5	0.2	1.0	0.0
seq4	0.4	0.6	0.0	1.0

Maximum Parsimony (1)

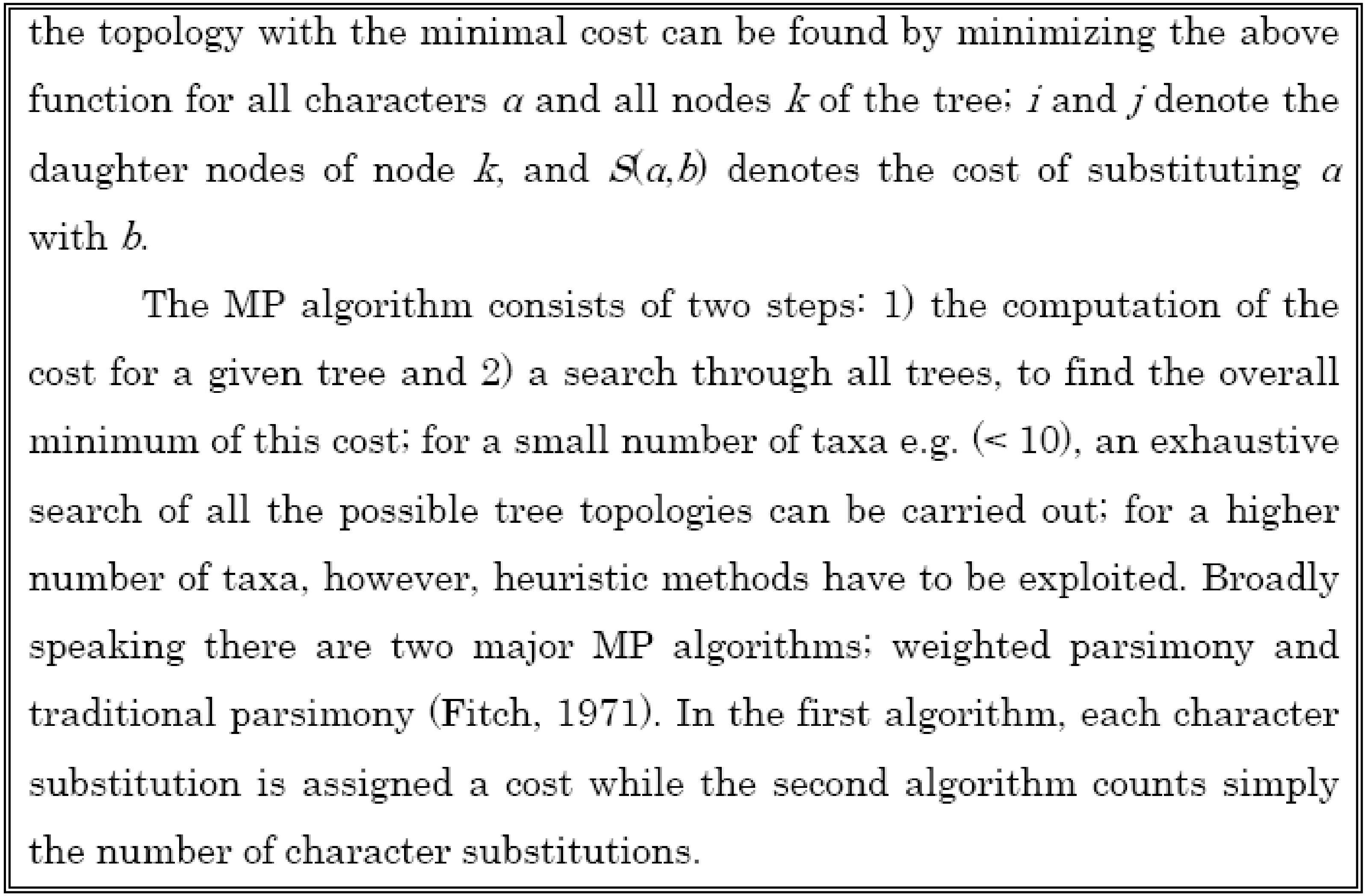


3.2.2.2 Maximum Parsimony

MP exploits the concept of parsimony that favours generally simpler over more complicated hypotheses. As such, MP is based on the assumption that the best tree topology is the one that requires the minimum number changes to explain the observed differences between the taxa, and searches for the topology with the minimal cost. If $S_k(a)$ denotes the minimal cost for assignment of character a to node k, such that:

$$S_k(a) = \min{}_b(S_i(b) + S(a,b)) + \min{}_b(S_j(b) + S(a,b))$$

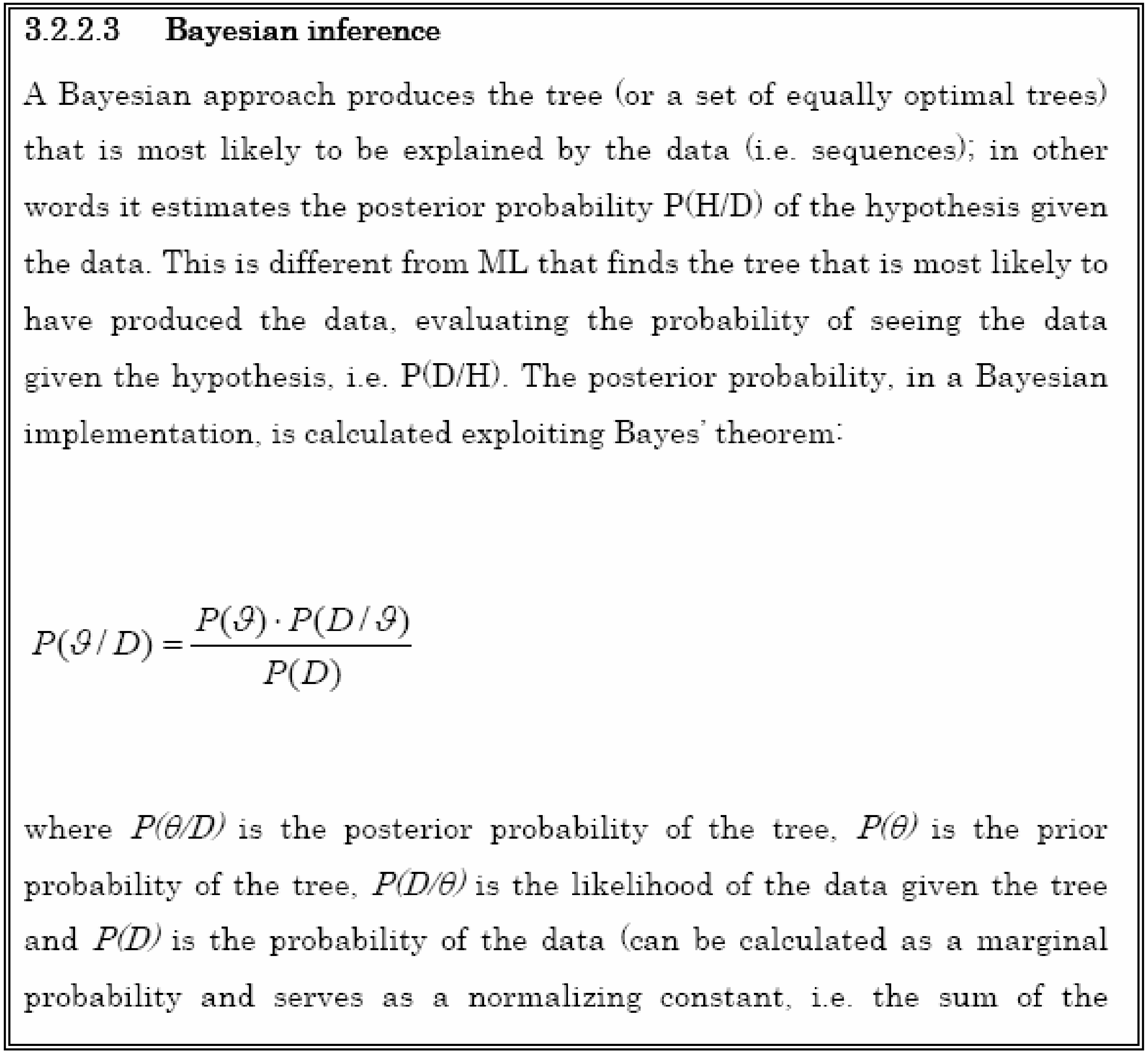
Maximum Parsimony (2)



the topology with the minimal cost can be found by minimizing the above function for all characters a and all nodes k of the tree; i and j denote the daughter nodes of node k, and $S(a,b)$ denotes the cost of substituting a with b.

The MP algorithm consists of two steps: 1) the computation of the cost for a given tree and 2) a search through all trees, to find the overall minimum of this cost; for a small number of taxa e.g. (< 10), an exhaustive search of all the possible tree topologies can be carried out; for a higher number of taxa, however, heuristic methods have to be exploited. Broadly speaking there are two major MP algorithms; weighted parsimony and traditional parsimony (Fitch, 1971). In the first algorithm, each character substitution is assigned a cost while the second algorithm counts simply the number of character substitutions.

Bayesian Inference (1)



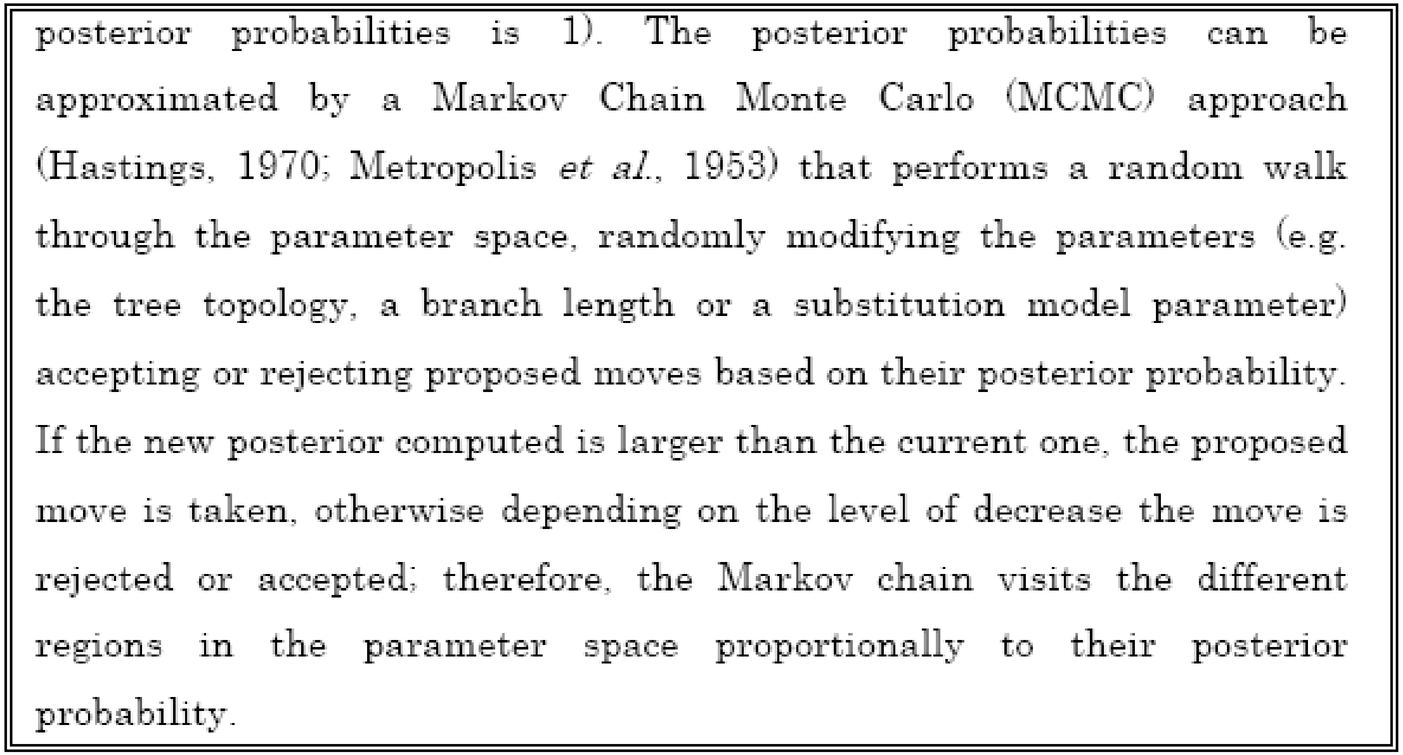
3.2.2.3 Bayesian inference

A Bayesian approach produces the tree (or a set of equally optimal trees) that is most likely to be explained by the data (i.e. sequences); in other words it estimates the posterior probability P(H/D) of the hypothesis given the data. This is different from ML that finds the tree that is most likely to have produced the data, evaluating the probability of seeing the data given the hypothesis, i.e. P(D/H). The posterior probability, in a Bayesian implementation, is calculated exploiting Bayes' theorem:

$$P(\vartheta / D) = \frac{P(\vartheta) \cdot P(D / \vartheta)}{P(D)}$$

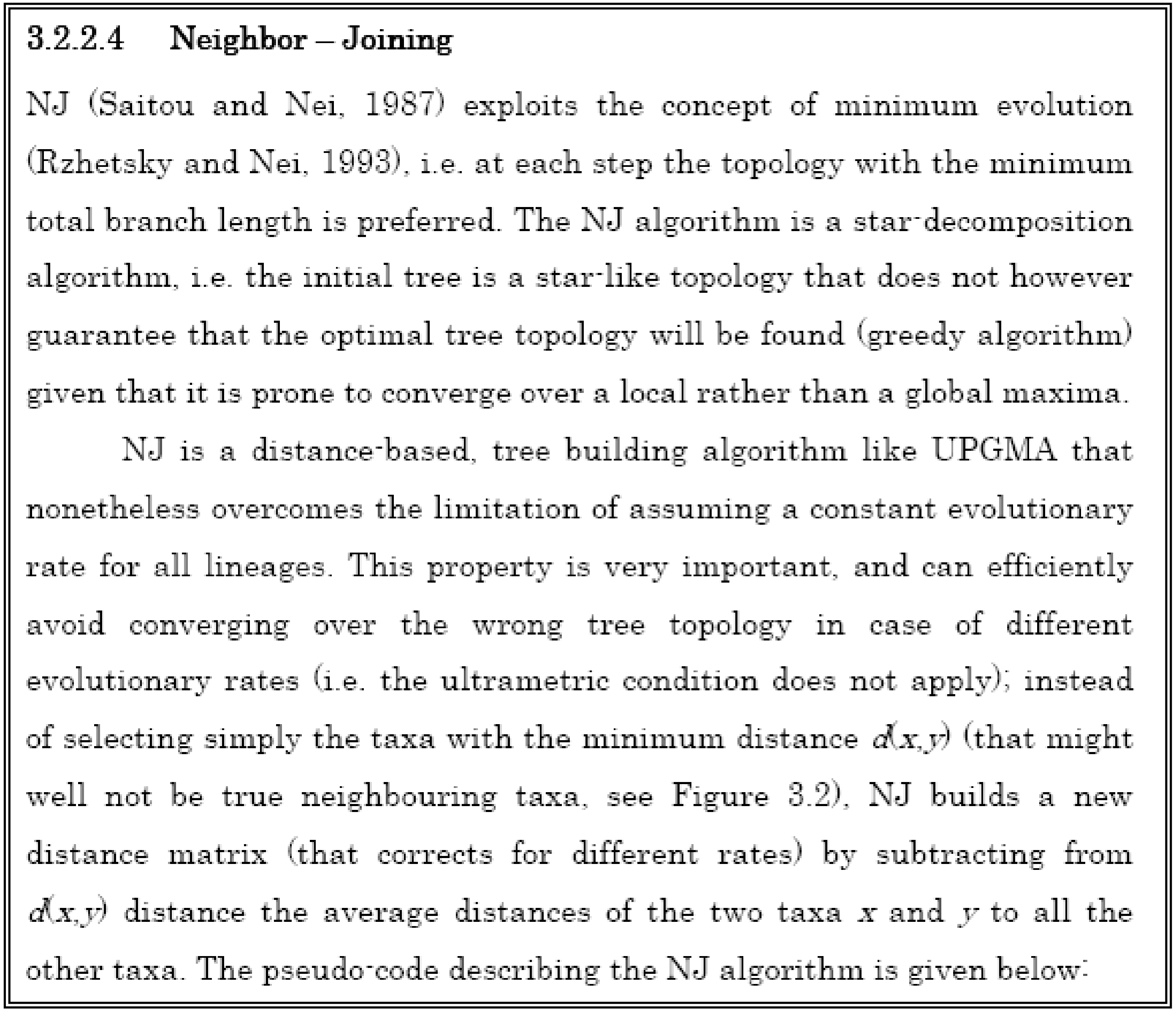
where $P(\theta/D)$ is the posterior probability of the tree, $P(\theta)$ is the prior probability of the tree, $P(D/\theta)$ is the likelihood of the data given the tree and $P(D)$ is the probability of the data (can be calculated as a marginal probability and serves as a normalizing constant, i.e. the sum of the

Bayesian Inference (2)



posterior probabilities is 1). The posterior probabilities can be approximated by a Markov Chain Monte Carlo (MCMC) approach (Hastings, 1970; Metropolis *et al.*, 1953) that performs a random walk through the parameter space, randomly modifying the parameters (e.g. the tree topology, a branch length or a substitution model parameter) accepting or rejecting proposed moves based on their posterior probability. If the new posterior computed is larger than the current one, the proposed move is taken, otherwise depending on the level of decrease the move is rejected or accepted; therefore, the Markov chain visits the different regions in the parameter space proportionally to their posterior probability.

Neighbor Joining (1)

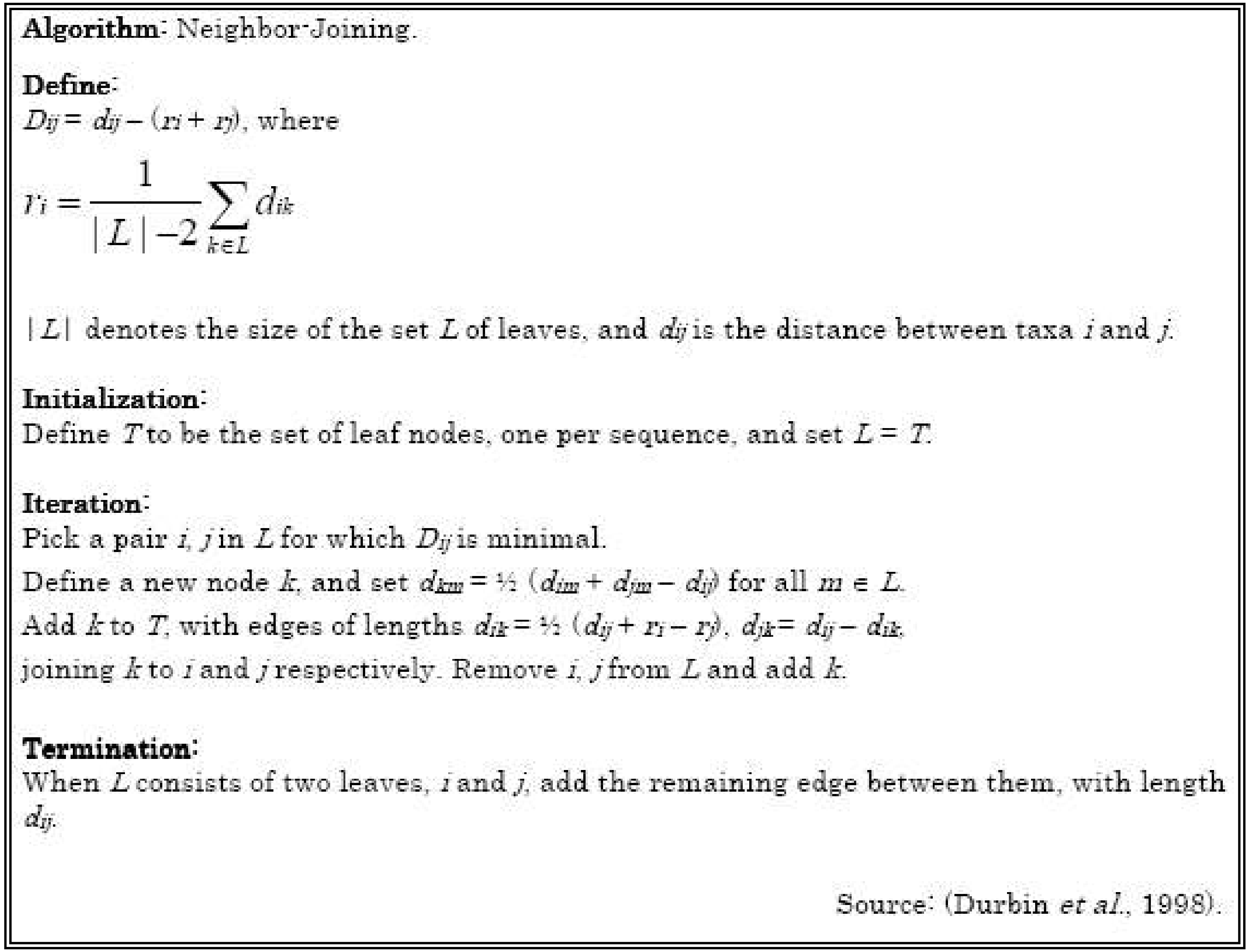


3.2.2.4 Neighbor – Joining

NJ (Saitou and Nei, 1987) exploits the concept of minimum evolution (Rzhetsky and Nei, 1993), i.e. at each step the topology with the minimum total branch length is preferred. The NJ algorithm is a star-decomposition algorithm, i.e. the initial tree is a star-like topology that does not however guarantee that the optimal tree topology will be found (greedy algorithm) given that it is prone to converge over a local rather than a global maxima.

NJ is a distance-based, tree building algorithm like UPGMA that nonetheless overcomes the limitation of assuming a constant evolutionary rate for all lineages. This property is very important, and can efficiently avoid converging over the wrong tree topology in case of different evolutionary rates (i.e. the ultrametric condition does not apply); instead of selecting simply the taxa with the minimum distance $d(x,y)$ (that might well not be true neighbouring taxa, see Figure 3.2), NJ builds a new distance matrix (that corrects for different rates) by subtracting from $d(x,y)$ distance the average distances of the two taxa x and y to all the other taxa. The pseudo-code describing the NJ algorithm is given below:

Neighbor Joining (2)



Algorithm: Neighbor-Joining.

Define:

$D_{ij} = d_{ij} - (r_i + r_j)$, where

$$r_i = \frac{1}{|L|-2} \sum_{k \in L} d_{ik}$$

$|L|$ denotes the size of the set L of leaves, and d_{ij} is the distance between taxa i and j.

Initialization:

Define T to be the set of leaf nodes, one per sequence, and set $L = T$.

Iteration:

Pick a pair i, j in L for which D_{ij} is minimal.

Define a new node k, and set $d_{km} = ½ (d_{im} + d_{jm} - d_{ij})$ for all $m \in L$.

Add k to T, with edges of lengths $d_{ik} = ½ (d_{ij} + r_i - r_j)$, $d_{jk} = d_{ij} - d_{ik}$,

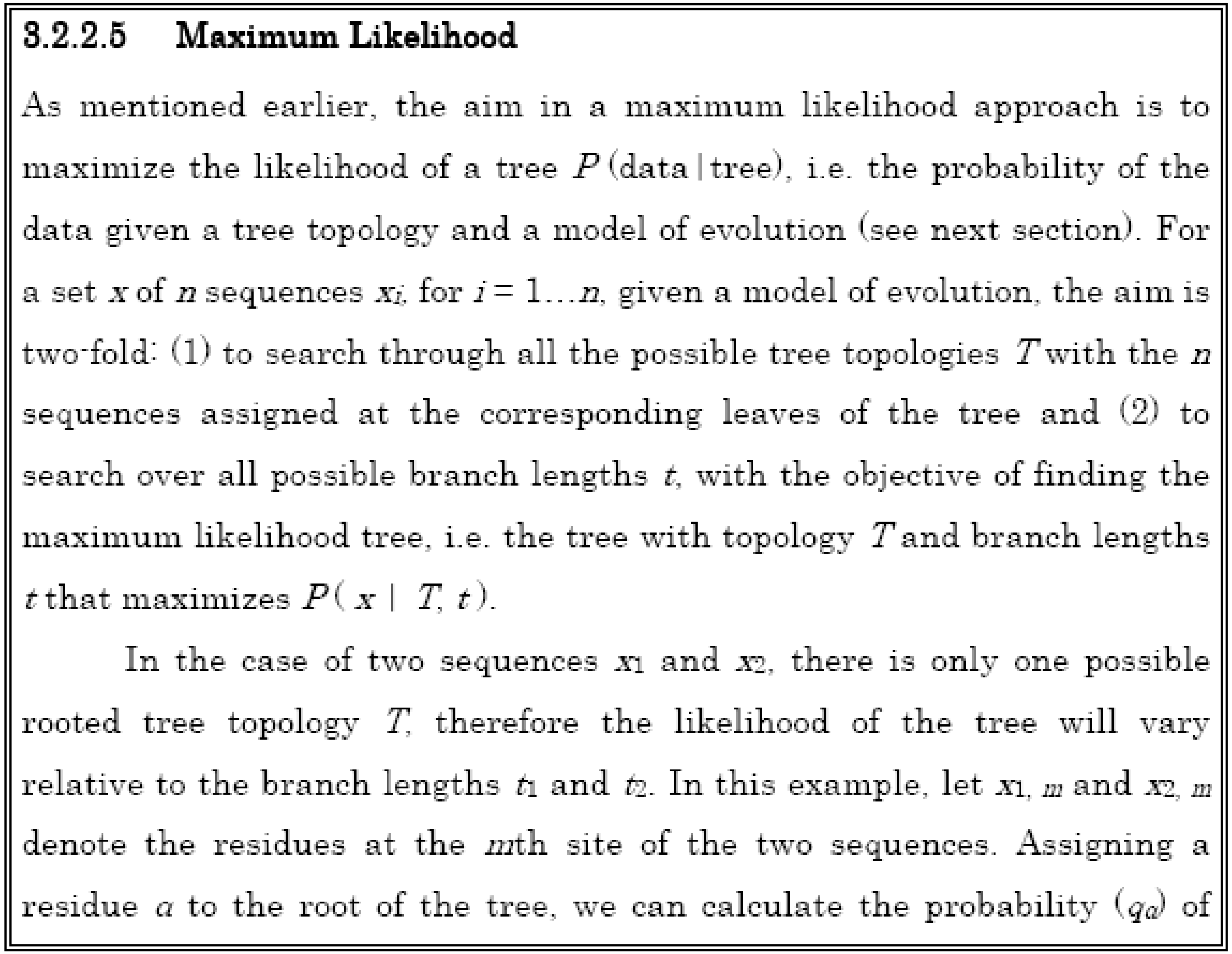
joining k to i and j respectively. Remove i, j from L and add k.

Termination:

When L consists of two leaves, i and j, add the remaining edge between them, with length d_{ij}.

Source: (Durbin *et al.*, 1998).

Maximum Likelihood (1)

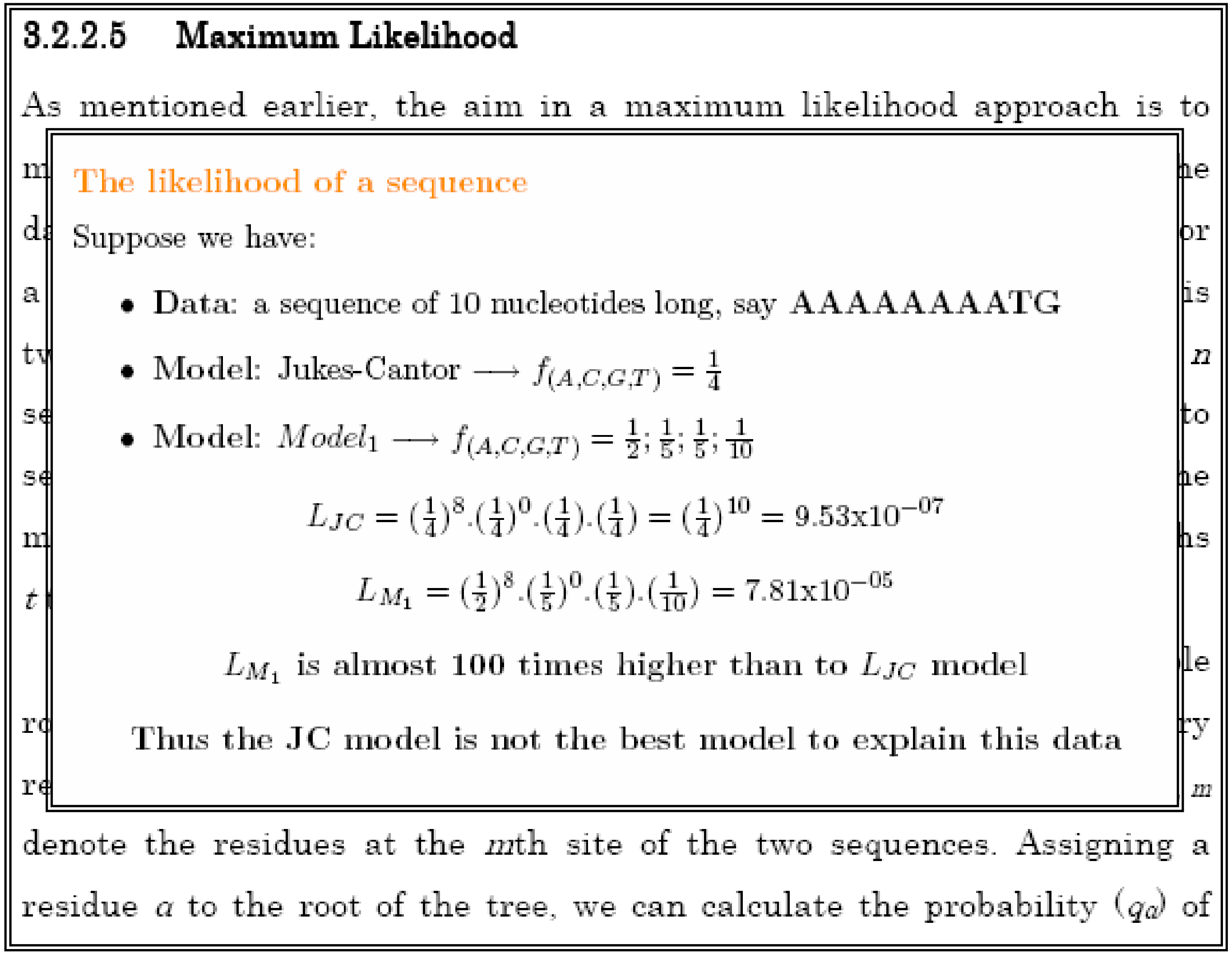


3.2.2.5 Maximum Likelihood

As mentioned earlier, the aim in a maximum likelihood approach is to maximize the likelihood of a tree P(data | tree), i.e. the probability of the data given a tree topology and a model of evolution (see next section). For a set x of n sequences x_i, for $i = 1 \ldots n$, given a model of evolution, the aim is two-fold: (1) to search through all the possible tree topologies T with the n sequences assigned at the corresponding leaves of the tree and (2) to search over all possible branch lengths t, with the objective of finding the maximum likelihood tree, i.e. the tree with topology T and branch lengths t that maximizes $P(x \mid T, t)$.

In the case of two sequences x_1 and x_2, there is only one possible rooted tree topology T, therefore the likelihood of the tree will vary relative to the branch lengths t_1 and t_2. In this example, let $x_{1,m}$ and $x_{2,m}$ denote the residues at the mth site of the two sequences. Assigning a residue a to the root of the tree, we can calculate the probability (q_a) of

Maximum Likelihood (1)



3.2.2.5 Maximum Likelihood

As mentioned earlier, the aim in a maximum likelihood approach is to

The likelihood of a sequence

Suppose we have:

- Data: a sequence of 10 nucleotides long, say **AAAAAAAATG**
- Model: Jukes-Cantor $\longrightarrow f_{(A,C,G,T)} = \frac{1}{4}$
- Model: $Model_1 \longrightarrow f_{(A,C,G,T)} = \frac{1}{2}; \frac{1}{5}; \frac{1}{5}; \frac{1}{10}$

$$L_{JC} = (\tfrac{1}{4})^8.(\tfrac{1}{4})^0.(\tfrac{1}{4}).(\tfrac{1}{4}) = (\tfrac{1}{4})^{10} = 9.53\text{x}10^{-07}$$

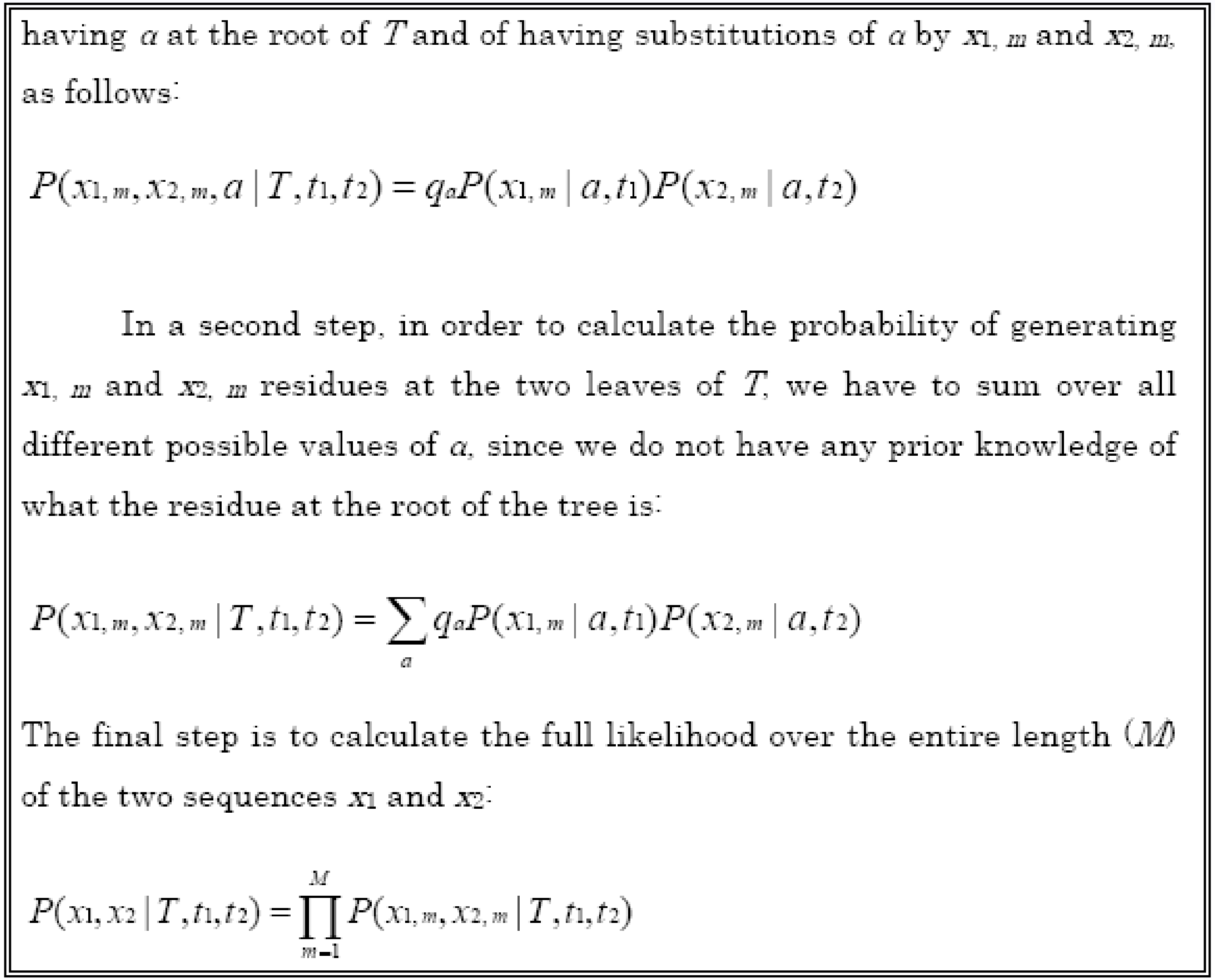
$$L_{M_1} = (\tfrac{1}{2})^8.(\tfrac{1}{5})^0.(\tfrac{1}{5}).(\tfrac{1}{10}) = 7.81\text{x}10^{-05}$$

L_{M_1} is almost 100 times higher than to L_{JC} model

Thus the JC model is not the best model to explain this data

denote the residues at the mth site of the two sequences. Assigning a residue a to the root of the tree, we can calculate the probability (q_a) of

Maximum Likelihood (2)



having a at the root of T and of having substitutions of a by $x_{1,\,m}$ and $x_{2,\,m}$, as follows:

$$P(x_{1,m}, x_{2,m}, a \mid T, t_1, t_2) = q_a P(x_{1,m} \mid a, t_1) P(x_{2,m} \mid a, t_2)$$

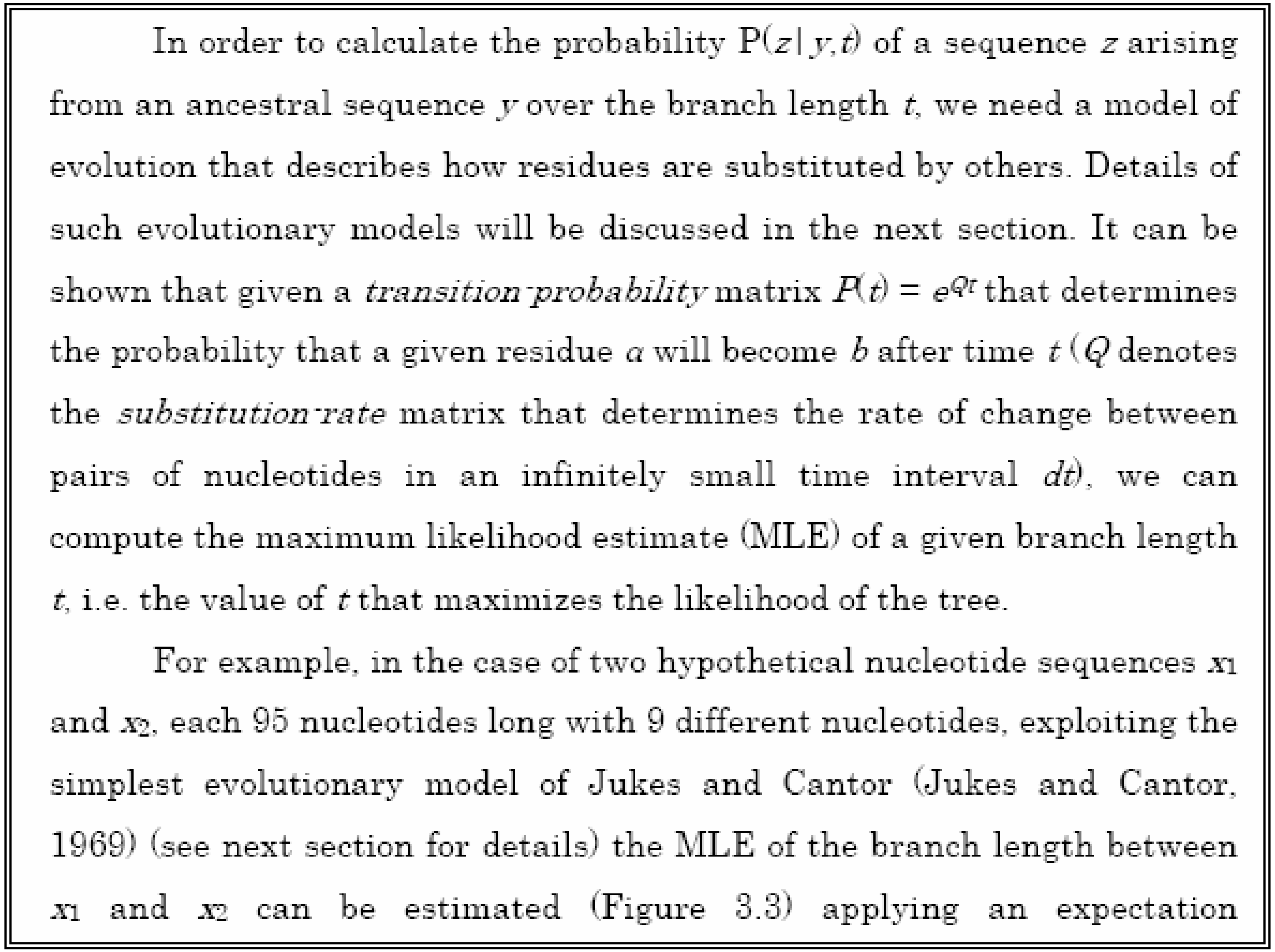
In a second step, in order to calculate the probability of generating $x_{1,\,m}$ and $x_{2,\,m}$ residues at the two leaves of T, we have to sum over all different possible values of a, since we do not have any prior knowledge of what the residue at the root of the tree is:

$$P(x_{1,m}, x_{2,m} \mid T, t_1, t_2) = \sum_a q_a P(x_{1,m} \mid a, t_1) P(x_{2,m} \mid a, t_2)$$

The final step is to calculate the full likelihood over the entire length (M) of the two sequences x_1 and x_2:

$$P(x_1, x_2 \mid T, t_1, t_2) = \prod_{m=1}^{M} P(x_{1,m}, x_{2,m} \mid T, t_1, t_2)$$

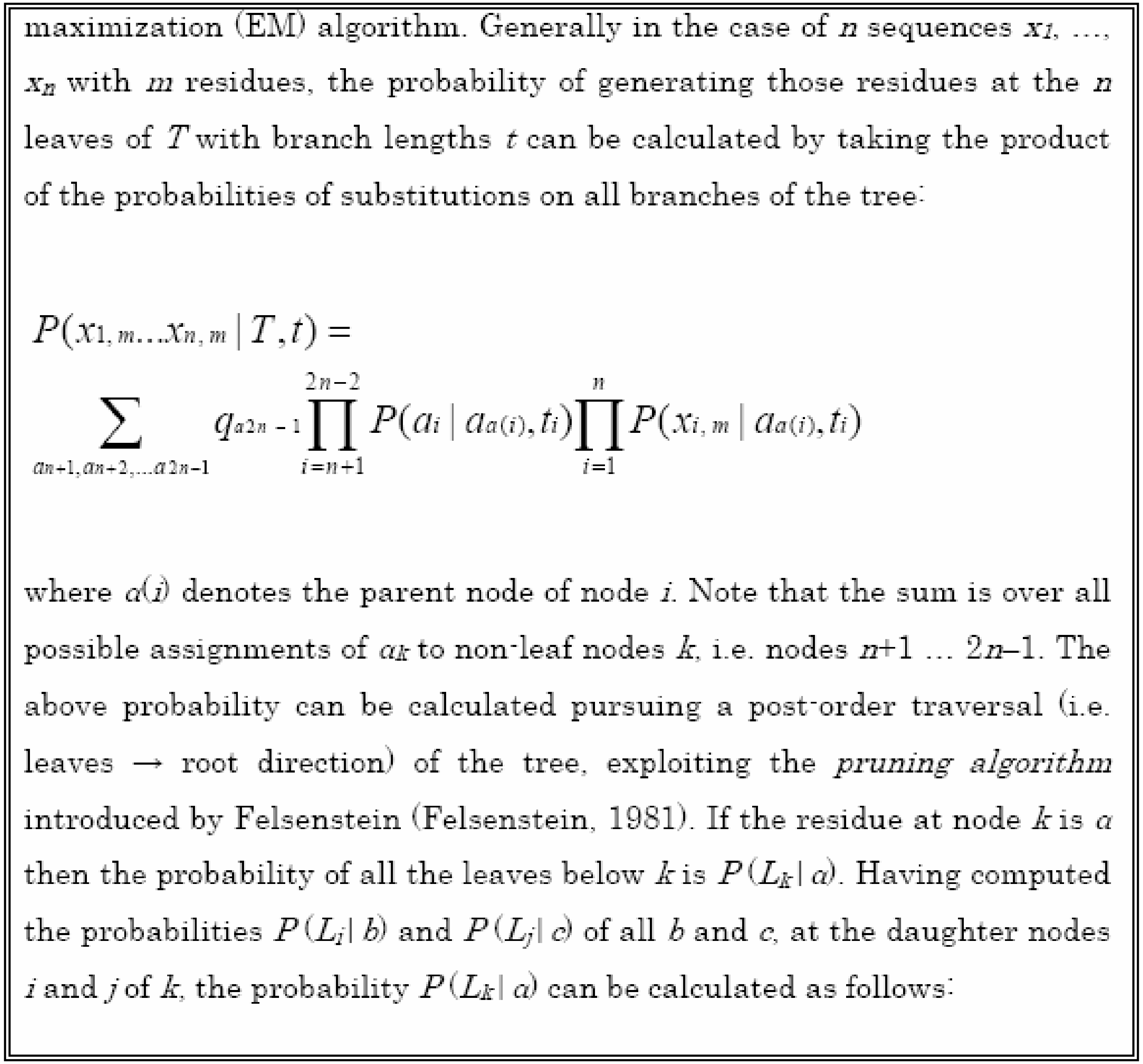
Maximum Likelihood (3)



In order to calculate the probability $P(z \mid y,t)$ of a sequence z arising from an ancestral sequence y over the branch length t, we need a model of evolution that describes how residues are substituted by others. Details of such evolutionary models will be discussed in the next section. It can be shown that given a *transition-probability* matrix $P(t) = e^{Qt}$ that determines the probability that a given residue a will become b after time t (Q denotes the *substitution-rate* matrix that determines the rate of change between pairs of nucleotides in an infinitely small time interval dt), we can compute the maximum likelihood estimate (MLE) of a given branch length t, i.e. the value of t that maximizes the likelihood of the tree.

For example, in the case of two hypothetical nucleotide sequences x_1 and x_2, each 95 nucleotides long with 9 different nucleotides, exploiting the simplest evolutionary model of Jukes and Cantor (Jukes and Cantor, 1969) (see next section for details) the MLE of the branch length between x_1 and x_2 can be estimated (Figure 3.3) applying an expectation

Maximum Likelihood (4)

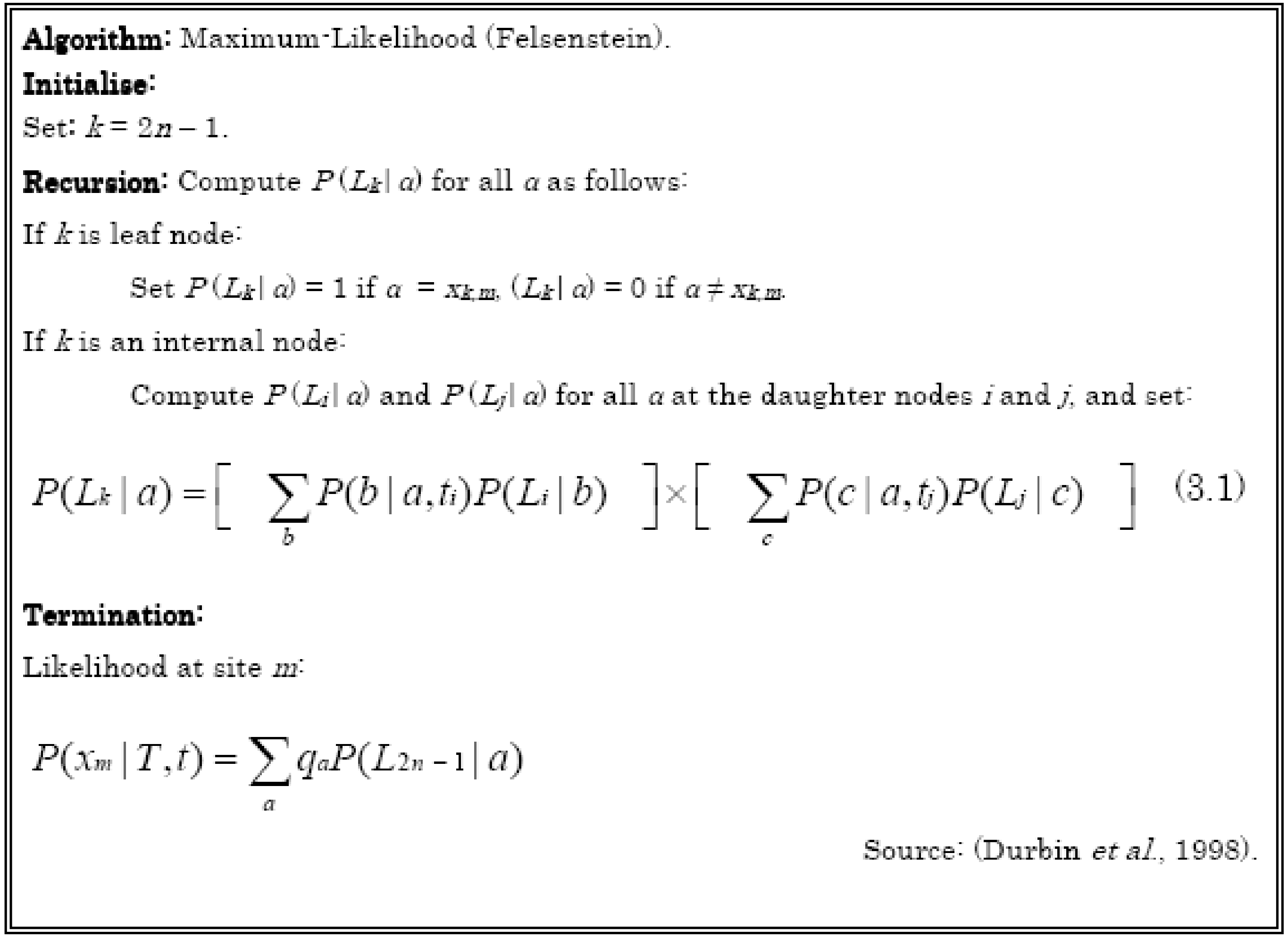


maximization (EM) algorithm. Generally in the case of *n* sequences x_1, …, x_n with *m* residues, the probability of generating those residues at the *n* leaves of *T* with branch lengths *t* can be calculated by taking the product of the probabilities of substitutions on all branches of the tree:

$$P(x_{1,m} \ldots x_{n,m} \mid T, t) = \sum_{a_{n+1}, a_{n+2}, \ldots a_{2n-1}} q_{a_{2n-1}} \prod_{i=n+1}^{2n-2} P(a_i \mid a_{a(i)}, t_i) \prod_{i=1}^{n} P(x_{i,m} \mid a_{a(i)}, t_i)$$

where *a*(*i*) denotes the parent node of node *i*. Note that the sum is over all possible assignments of a_k to non-leaf nodes *k*, i.e. nodes *n*+1 … 2*n*–1. The above probability can be calculated pursuing a post-order traversal (i.e. leaves → root direction) of the tree, exploiting the *pruning algorithm* introduced by Felsenstein (Felsenstein, 1981). If the residue at node *k* is *a* then the probability of all the leaves below *k* is $P(L_k| a)$. Having computed the probabilities $P(L_i| b)$ and $P(L_j| c)$ of all *b* and *c*, at the daughter nodes *i* and *j* of *k*, the probability $P(L_k| a)$ can be calculated as follows:

Maximum Likelihood (5)



```
Algorithm: Maximum-Likelihood (Felsenstein).
Initialise:
Set: $k = 2n - 1$.
Recursion: Compute $P(L_k| a)$ for all $a$ as follows:
If $k$ is leaf node:
        Set $P(L_k| a) = 1$ if $a = x_{k,m}$, $(L_k| a) = 0$ if $a \neq x_{k,m}$.
If $k$ is an internal node:
        Compute $P(L_i| a)$ and $P(L_j| a)$ for all $a$ at the daughter nodes $i$ and $j$, and set:
```

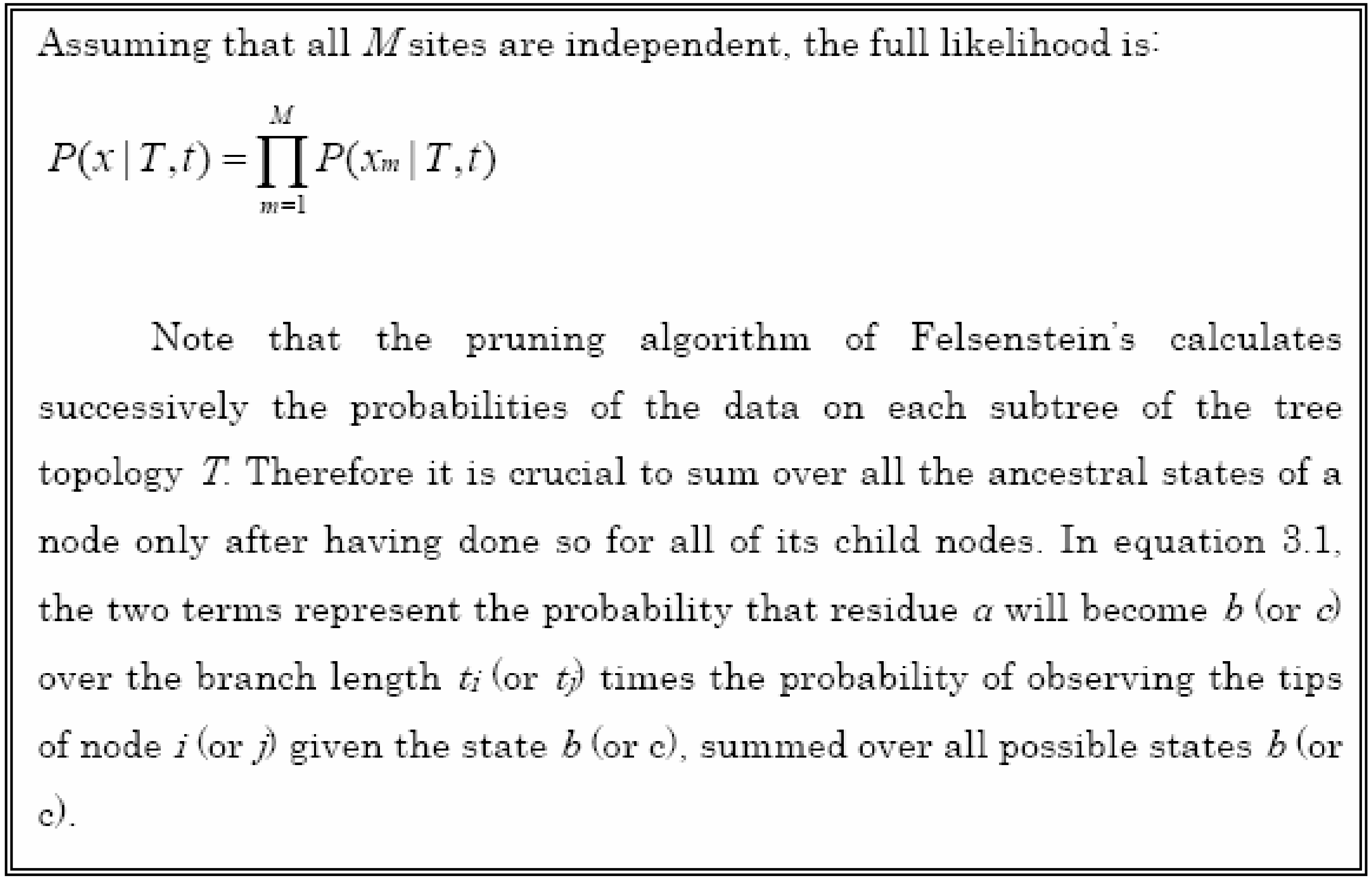
$$P(L_k \mid a) = \left[\sum_b P(b \mid a, t_i) P(L_i \mid b) \right] \times \left[\sum_c P(c \mid a, t_j) P(L_j \mid c) \right] \quad (3.1)$$

```
Termination:
Likelihood at site $m$:
```

$$P(x_m \mid T, t) = \sum_a q_a P(L_{2n-1} \mid a)$$

Source: (Durbin *et al.*, 1998).

Maximum Likelihood (6)



Assuming that all M sites are independent, the full likelihood is:

$$P(x \mid T,t) = \prod_{m=1}^{M} P(x_m \mid T,t)$$

Note that the pruning algorithm of Felsenstein's calculates successively the probabilities of the data on each subtree of the tree topology T. Therefore it is crucial to sum over all the ancestral states of a node only after having done so for all of its child nodes. In equation 3.1, the two terms represent the probability that residue a will become b (or c) over the branch length t_i (or t_j) times the probability of observing the tips of node i (or j) given the state b (or c), summed over all possible states b (or c).

Maximum Likelihood Estimation of Branch Lengths

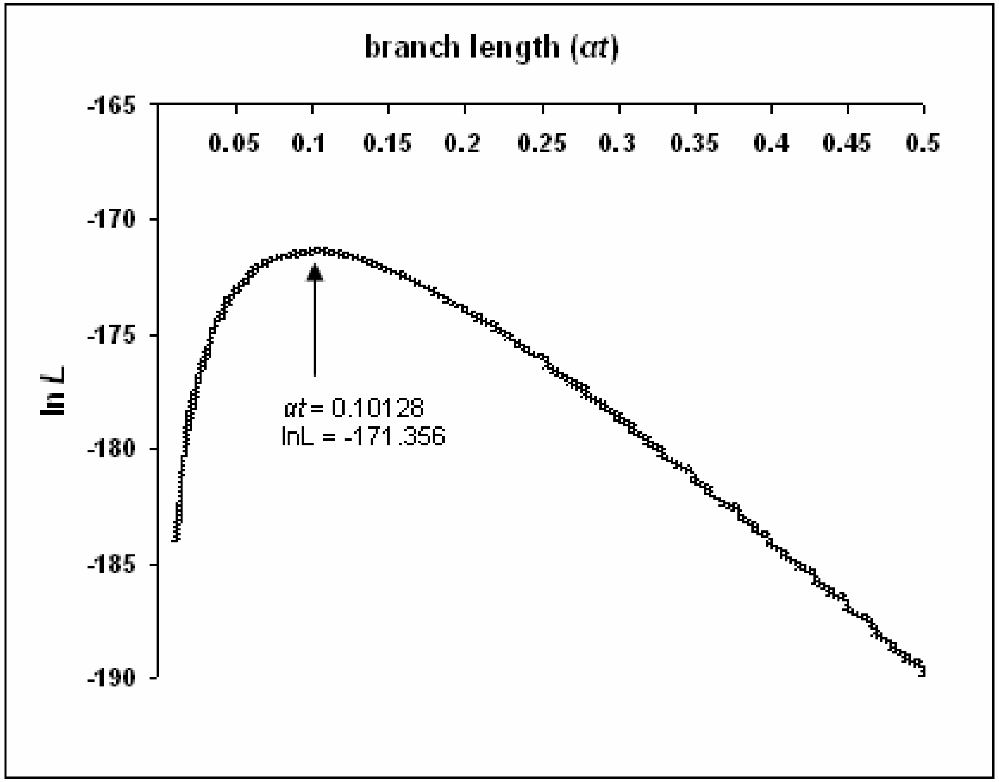
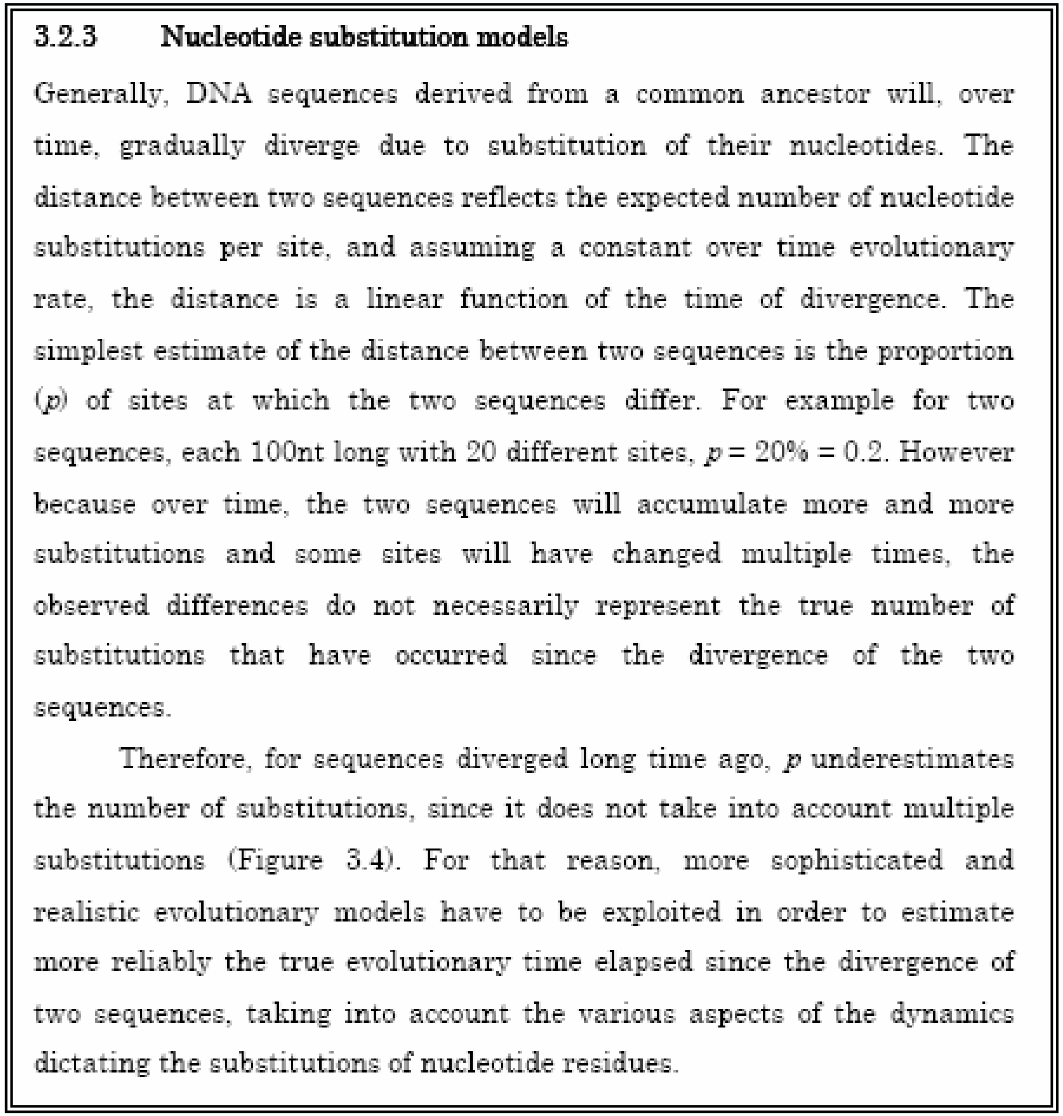


Figure 3.3: The *log* likelihood P(x_1,x_2 | T,t) for two sequences x_1, x_2 with 9 different nucleotides (nt) and a total length of 95nt, exploiting the Jukes and Cantor model. The MLE (0.10128) of the branch length is shown.

Nucleotide Substitution Models



3.2.3 Nucleotide substitution models

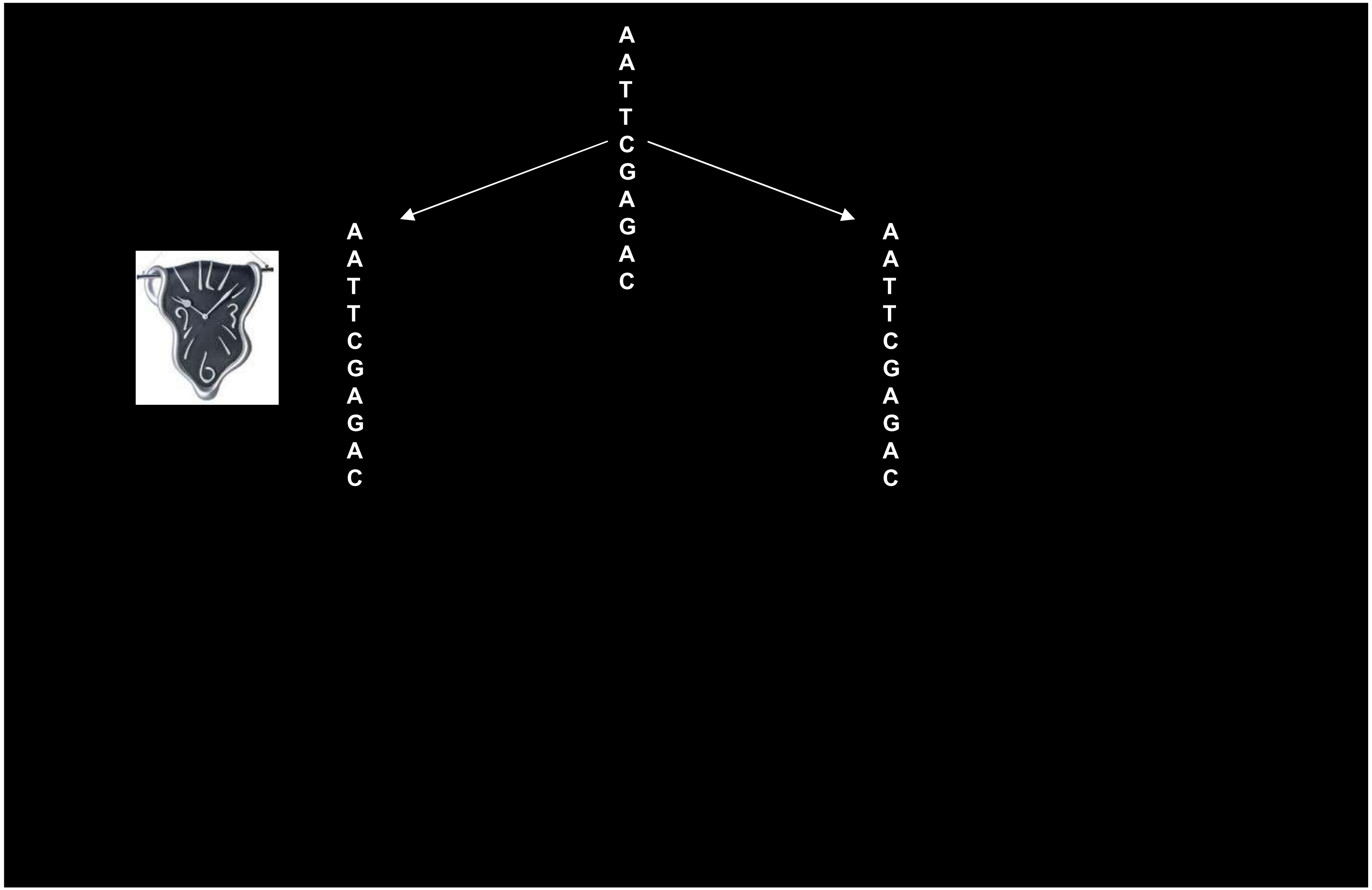
Generally, DNA sequences derived from a common ancestor will, over time, gradually diverge due to substitution of their nucleotides. The distance between two sequences reflects the expected number of nucleotide substitutions per site, and assuming a constant over time evolutionary rate, the distance is a linear function of the time of divergence. The simplest estimate of the distance between two sequences is the proportion (p) of sites at which the two sequences differ. For example for two sequences, each 100nt long with 20 different sites, $p = 20\% = 0.2$. However because over time, the two sequences will accumulate more and more substitutions and some sites will have changed multiple times, the observed differences do not necessarily represent the true number of substitutions that have occurred since the divergence of the two sequences.

Therefore, for sequences diverged long time ago, p underestimates the number of substitutions, since it does not take into account multiple substitutions (Figure 3.4). For that reason, more sophisticated and realistic evolutionary models have to be exploited in order to estimate more reliably the true evolutionary time elapsed since the divergence of two sequences, taking into account the various aspects of the dynamics dictating the substitutions of nucleotide residues.

Sequences are not what we see …

A
A
T
T
C
G
A
G
A
C

Sequences are not what we see …

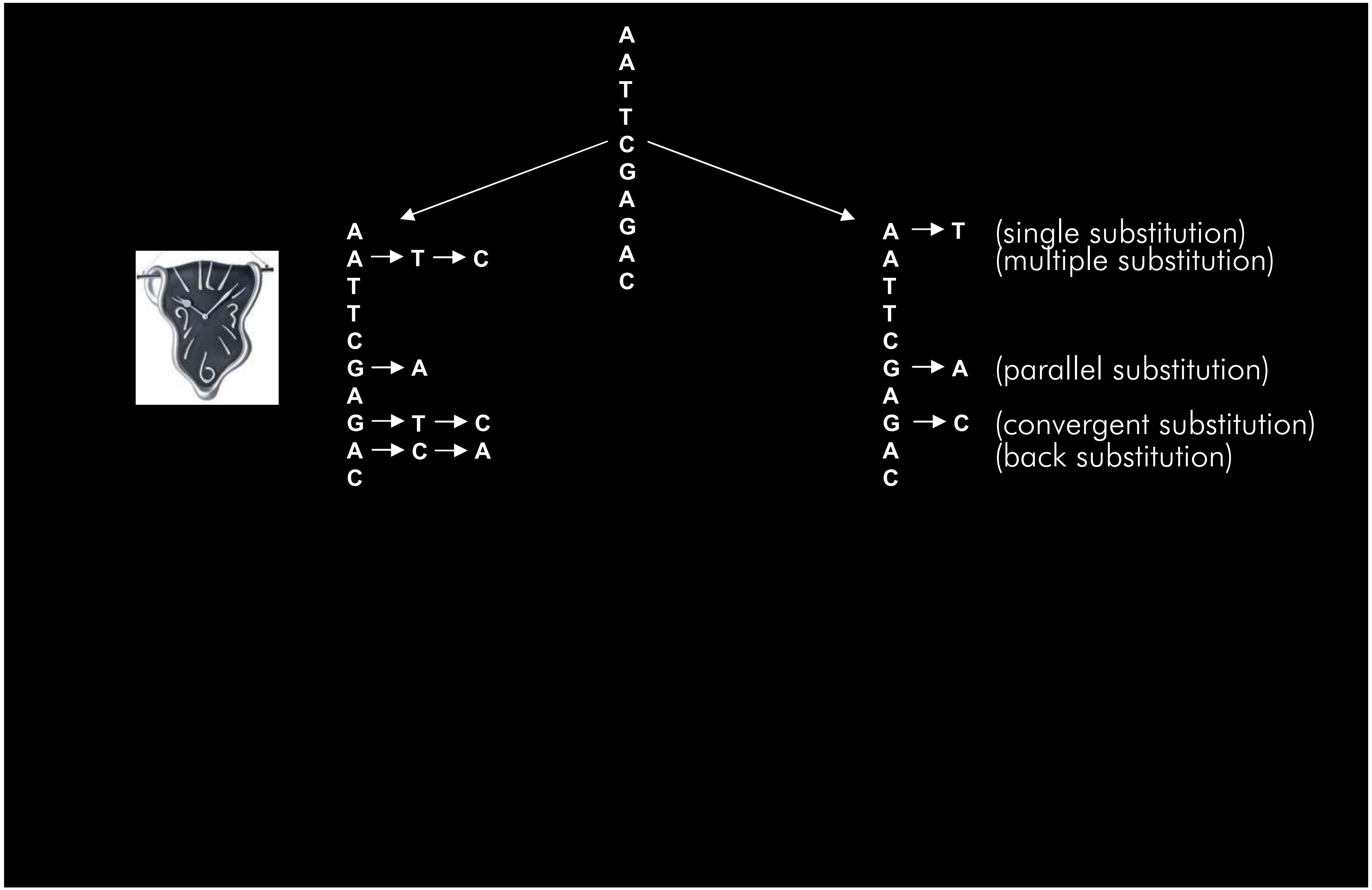


AATTCGAGAC

AATTCGAGAC

AATTCGAGAC

Sequences are not what we see …

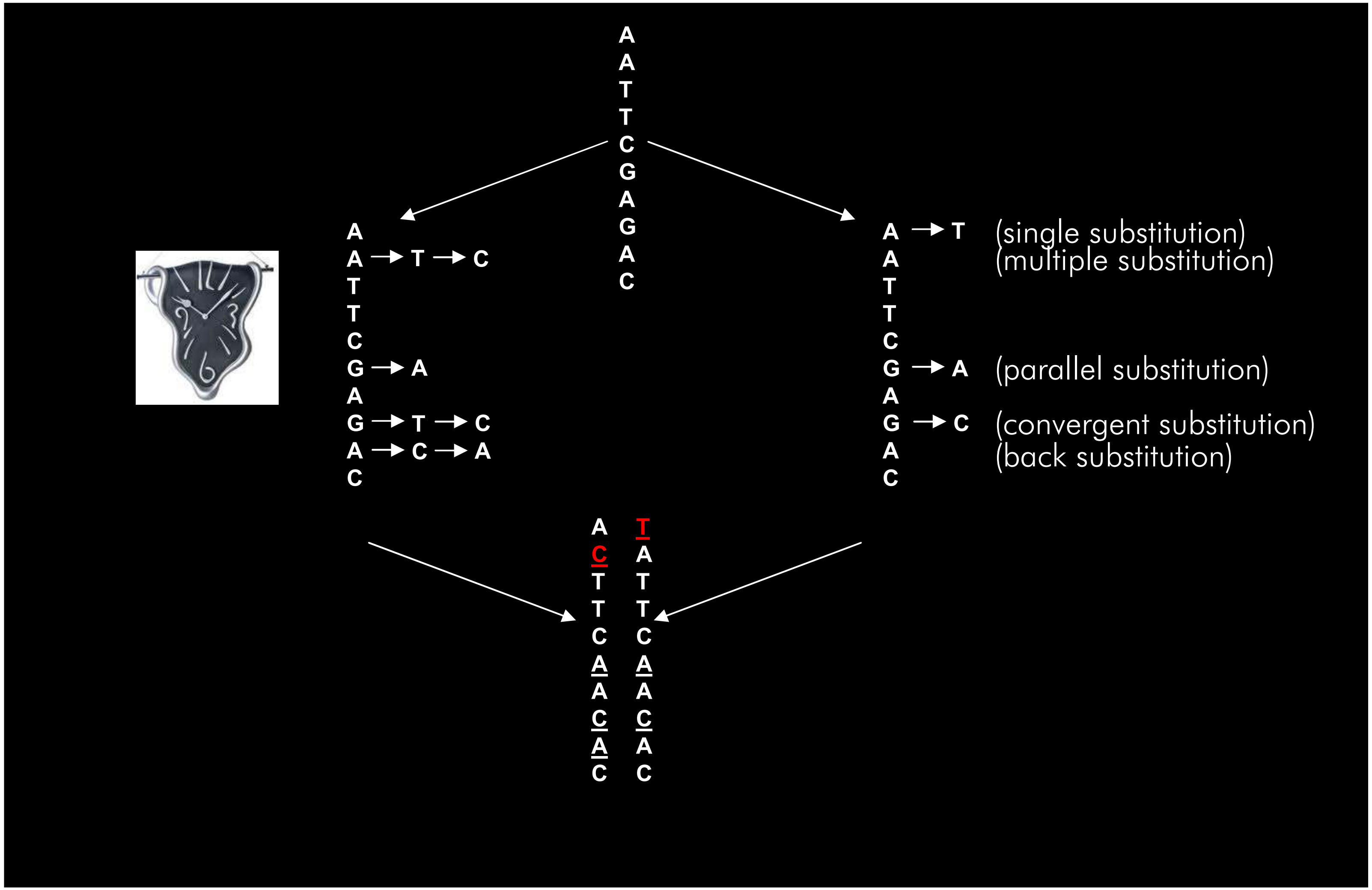


A
A
T
T
C
G
A
G
A
C

A
A → T → C
T
T
C
G → A
A
G → T → C
A → C → A
C

A → T (single substitution)
A (multiple substitution)
T
T
C
G → A (parallel substitution)
A
G → C (convergent substitution)
A (back substitution)
C

Sequences are not what we see …



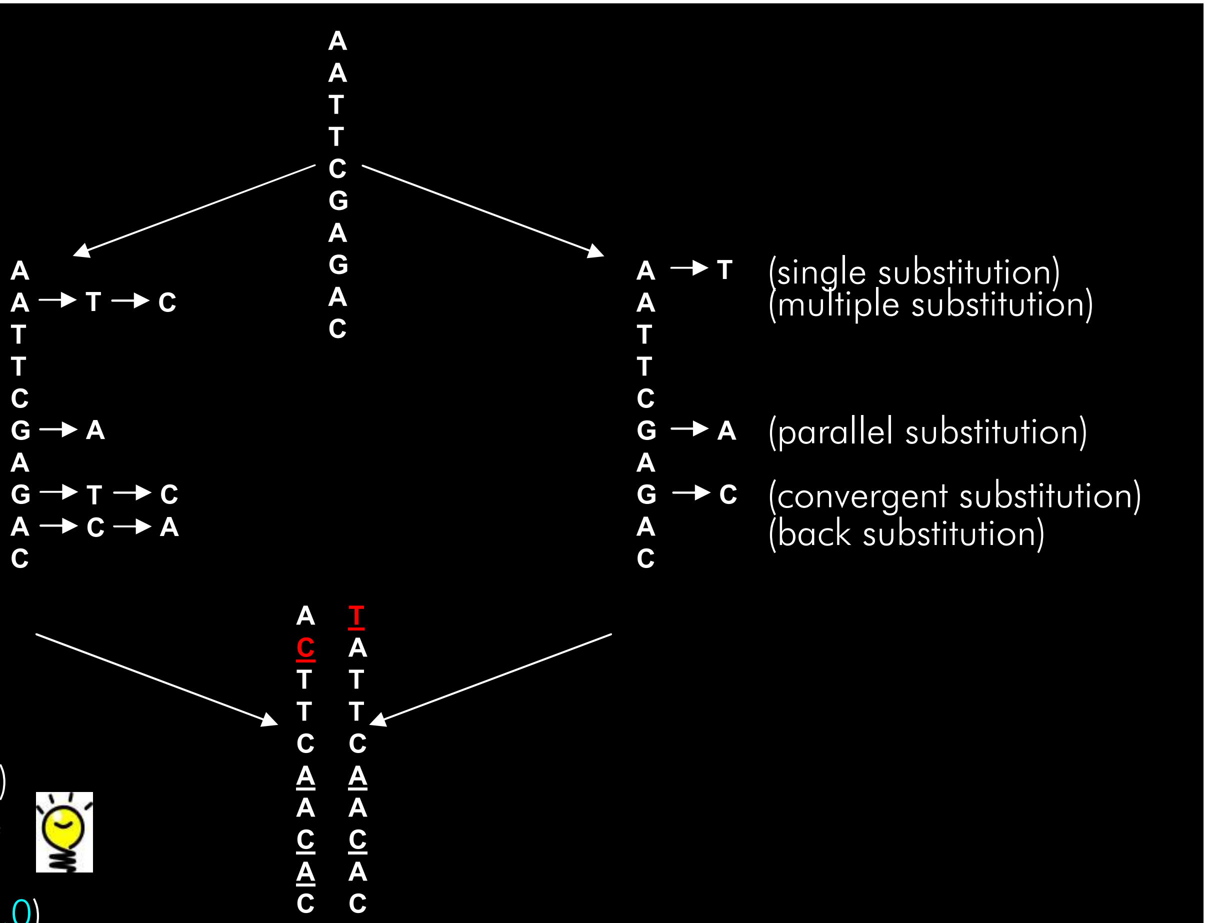
A
A
T
T
C
G
A
G
A
C

A
A → T → C
T
T
C
G → A
A
G → T → C
A → C → A
C

A → T (single substitution)
A (multiple substitution)
T
T
C
G → A (parallel substitution)
A
G → C (convergent substitution)
A (back substitution)
C

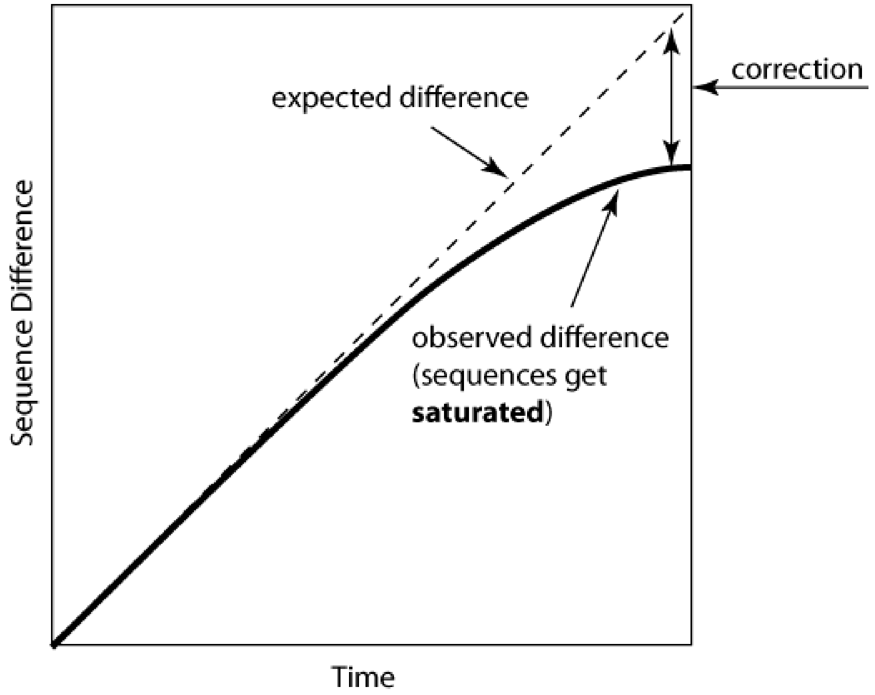
A T
C A
T T
T T
C C
A A
A A
C C
A A
C C

Sequences are not what we see …



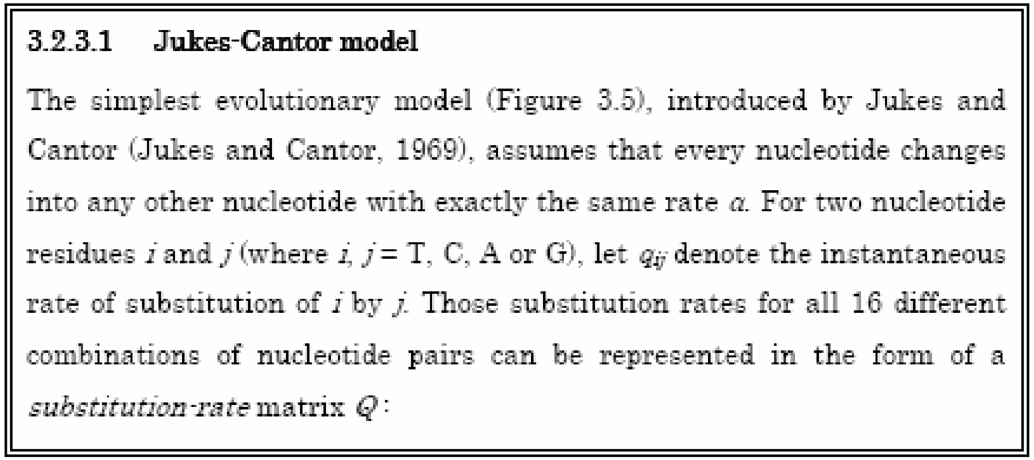
Only two observed substitutions (p = 0.2) are inferred, while the true number of substitutions is (p = 1.0)

Observed vs Expected



Observed vs. Expected number of DNA substitutions. As time since divergence increases, multiple substitutions start to occur, making number of visible substitutions smaller than the number of actual ones. Eventually, after long-long time there will be substitutions at every site. Two random sequences with equal frequencies of base pairs will differ on average in 3/4 of sites. Correction is required to compensate for the difference in observed and expected number of substitutions.

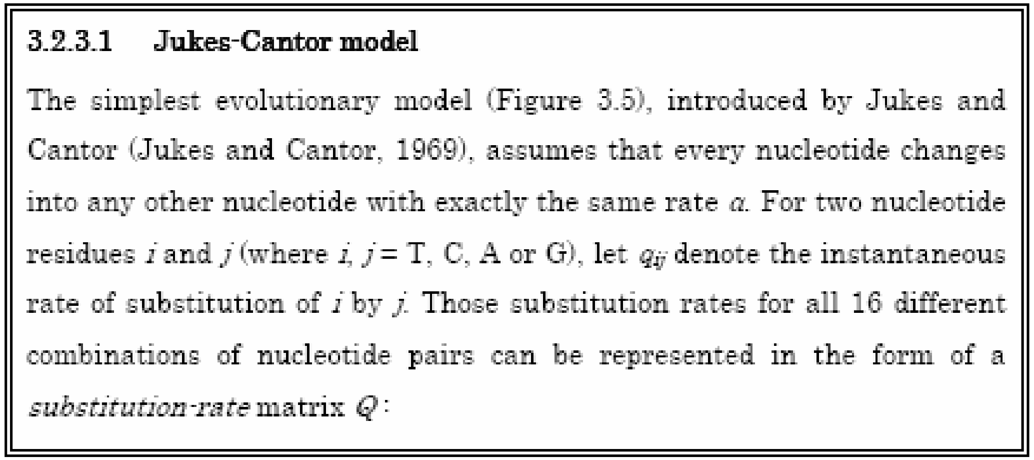
Jukes Cantor (1)



3.2.3.1 Jukes-Cantor model

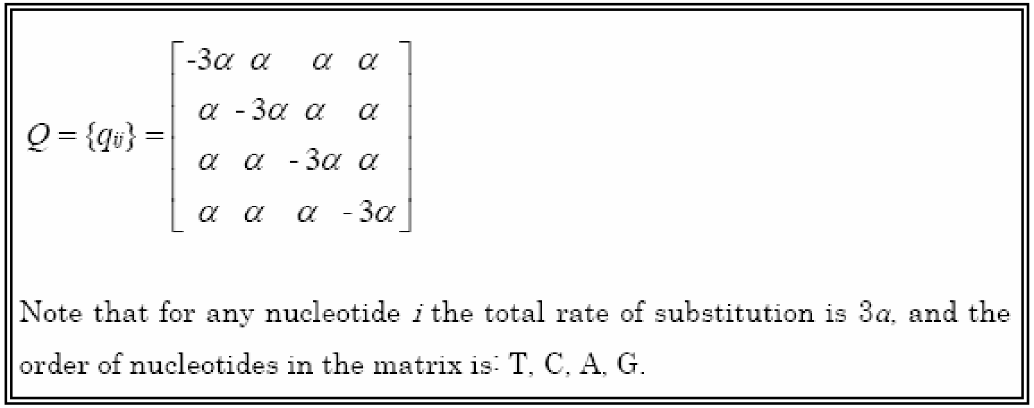
The simplest evolutionary model (Figure 3.5), introduced by Jukes and Cantor (Jukes and Cantor, 1969), assumes that every nucleotide changes into any other nucleotide with exactly the same rate α. For two nucleotide residues i and j (where i, j = T, C, A or G), let q_{ij} denote the instantaneous rate of substitution of i by j. Those substitution rates for all 16 different combinations of nucleotide pairs can be represented in the form of a *substitution-rate* matrix Q :

Jukes Cantor (1)



3.2.3.1 Jukes-Cantor model

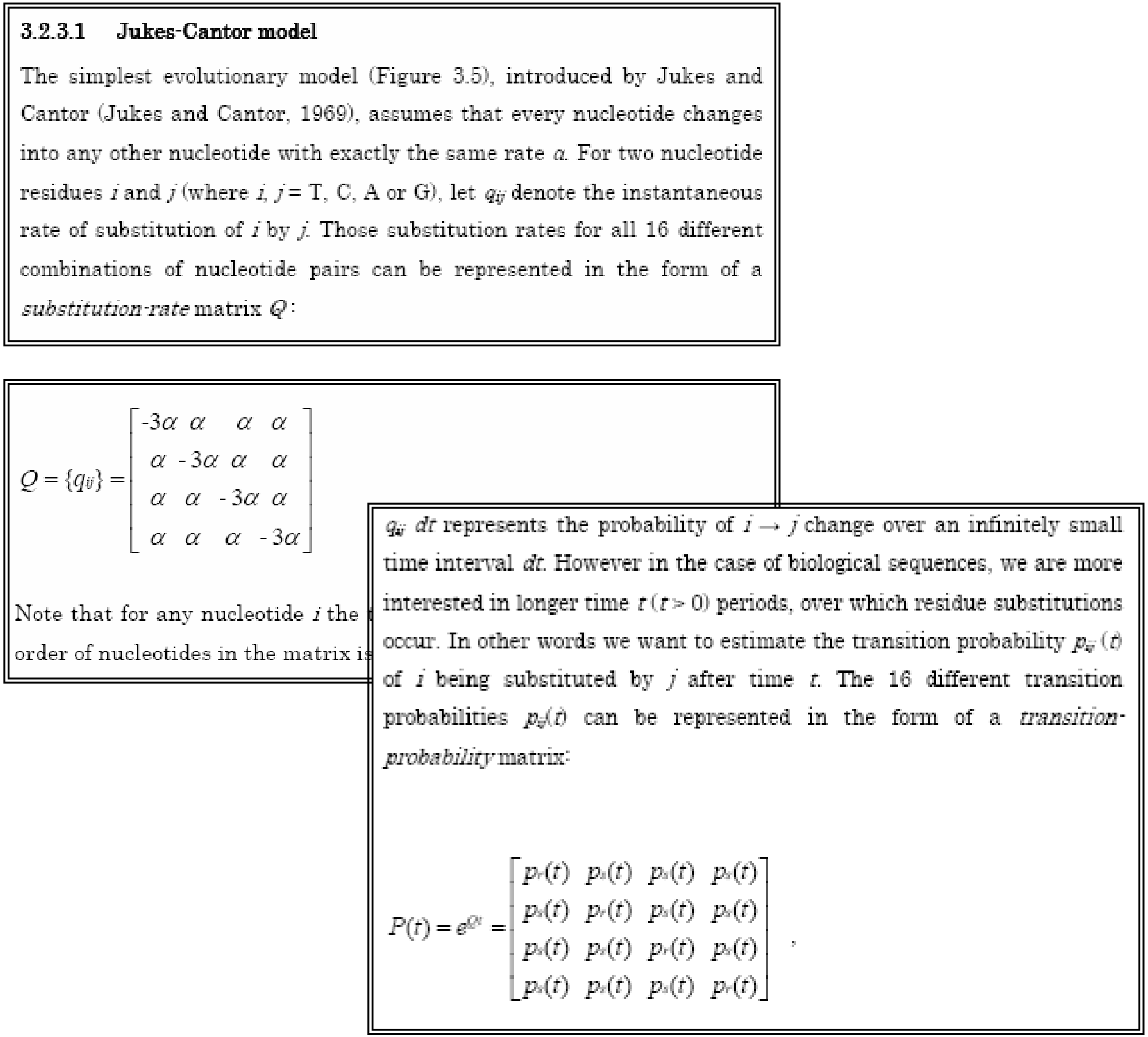
The simplest evolutionary model (Figure 3.5), introduced by Jukes and Cantor (Jukes and Cantor, 1969), assumes that every nucleotide changes into any other nucleotide with exactly the same rate α. For two nucleotide residues i and j (where i, j = T, C, A or G), let q_{ij} denote the instantaneous rate of substitution of i by j. Those substitution rates for all 16 different combinations of nucleotide pairs can be represented in the form of a *substitution-rate* matrix Q:



$$Q = \{q_{ij}\} = \begin{bmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{bmatrix}$$

Note that for any nucleotide i the total rate of substitution is 3α, and the order of nucleotides in the matrix is: T, C, A, G.

Jukes Cantor (1)



3.2.3.1 Jukes-Cantor model

The simplest evolutionary model (Figure 3.5), introduced by Jukes and Cantor (Jukes and Cantor, 1969), assumes that every nucleotide changes into any other nucleotide with exactly the same rate α. For two nucleotide residues i and j (where i, j = T, C, A or G), let q_{ij} denote the instantaneous rate of substitution of i by j. Those substitution rates for all 16 different combinations of nucleotide pairs can be represented in the form of a *substitution-rate* matrix Q:

$$Q = \{q_{ij}\} = \begin{bmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{bmatrix}$$

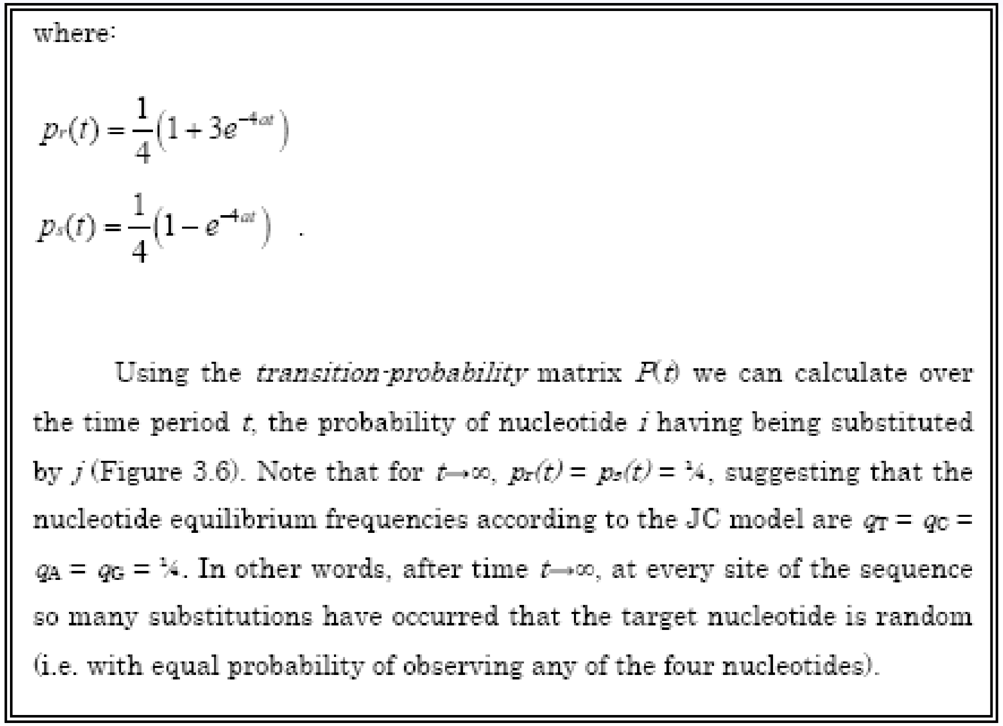
Note that for any nucleotide i the

order of nucleotides in the matrix is

$q_{ij}\ dt$ represents the probability of $i \rightarrow j$ change over an infinitely small time interval dt. However in the case of biological sequences, we are more interested in longer time t ($t > 0$) periods, over which residue substitutions occur. In other words we want to estimate the transition probability $p_{ij}(t)$ of i being substituted by j after time t. The 16 different transition probabilities $p_{ij}(t)$ can be represented in the form of a *transition-probability* matrix:

$$P(t) = e^{Qt} = \begin{bmatrix} p_r(t) & p_s(t) & p_s(t) & p_s(t) \\ p_s(t) & p_r(t) & p_s(t) & p_s(t) \\ p_s(t) & p_s(t) & p_r(t) & p_s(t) \\ p_s(t) & p_s(t) & p_s(t) & p_r(t) \end{bmatrix} ,$$

Jukes Cantor (2)



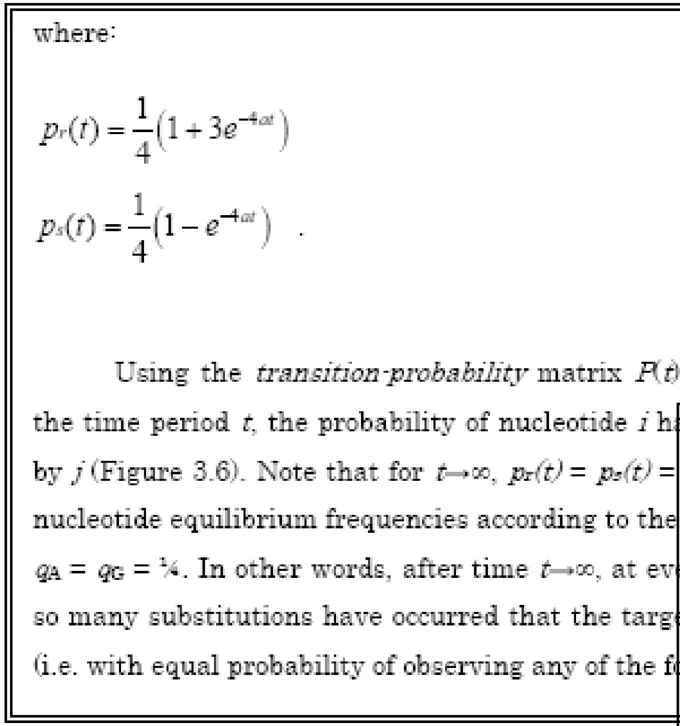
where:

$$p_r(t) = \frac{1}{4}\left(1 + 3e^{-4\alpha t}\right)$$

$$p_s(t) = \frac{1}{4}\left(1 - e^{-4\alpha t}\right) \quad .$$

Using the *transition-probability* matrix *P(t)* we can calculate over the time period *t*, the probability of nucleotide *i* having being substituted by *j* (Figure 3.6). Note that for *t*→∞, $p_r(t)$ = $p_s(t)$ = ¼, suggesting that the nucleotide equilibrium frequencies according to the JC model are q_T = q_C = q_A = q_G = ¼. In other words, after time *t*→∞, at every site of the sequence so many substitutions have occurred that the target nucleotide is random (i.e. with equal probability of observing any of the four nucleotides).

Jukes Cantor (2)



where:

$$p_r(t) = \frac{1}{4}\left(1 + 3e^{-4\alpha t}\right)$$

$$p_s(t) = \frac{1}{4}\left(1 - e^{-4\alpha t}\right) \quad .$$

Using the *transition-probability* matrix *P*(*t*) we can calculate over the time period *t*, the probability of nucleotide *i* h
by *j* (Figure 3.6). Note that for $t \rightarrow \infty$, $p_r(t) = p_s(t) =$
nucleotide equilibrium frequencies according to the
$q_A = q_G = ¼$. In other words, after time $t \rightarrow \infty$, at eve
so many substitutions have occurred that the targe
(i.e. with equal probability of observing any of the f

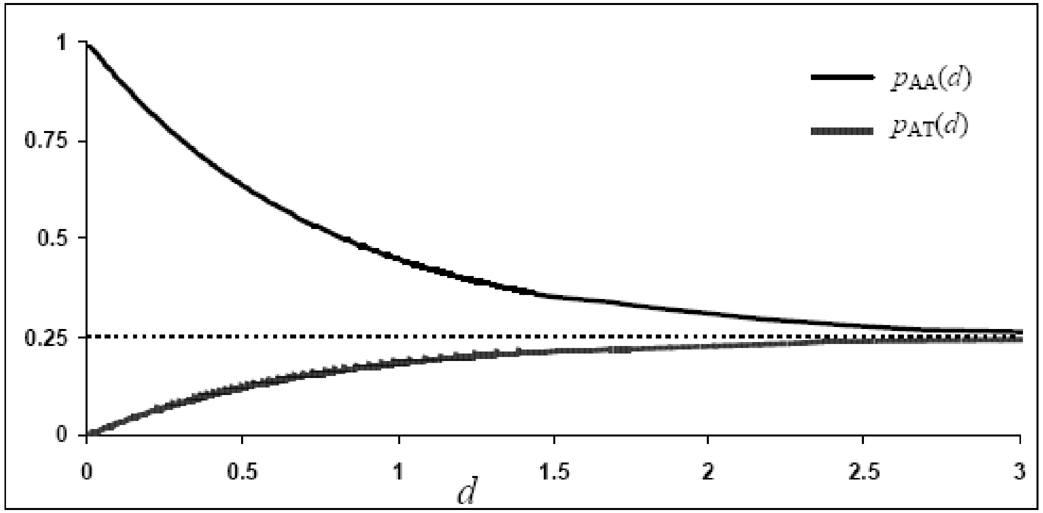
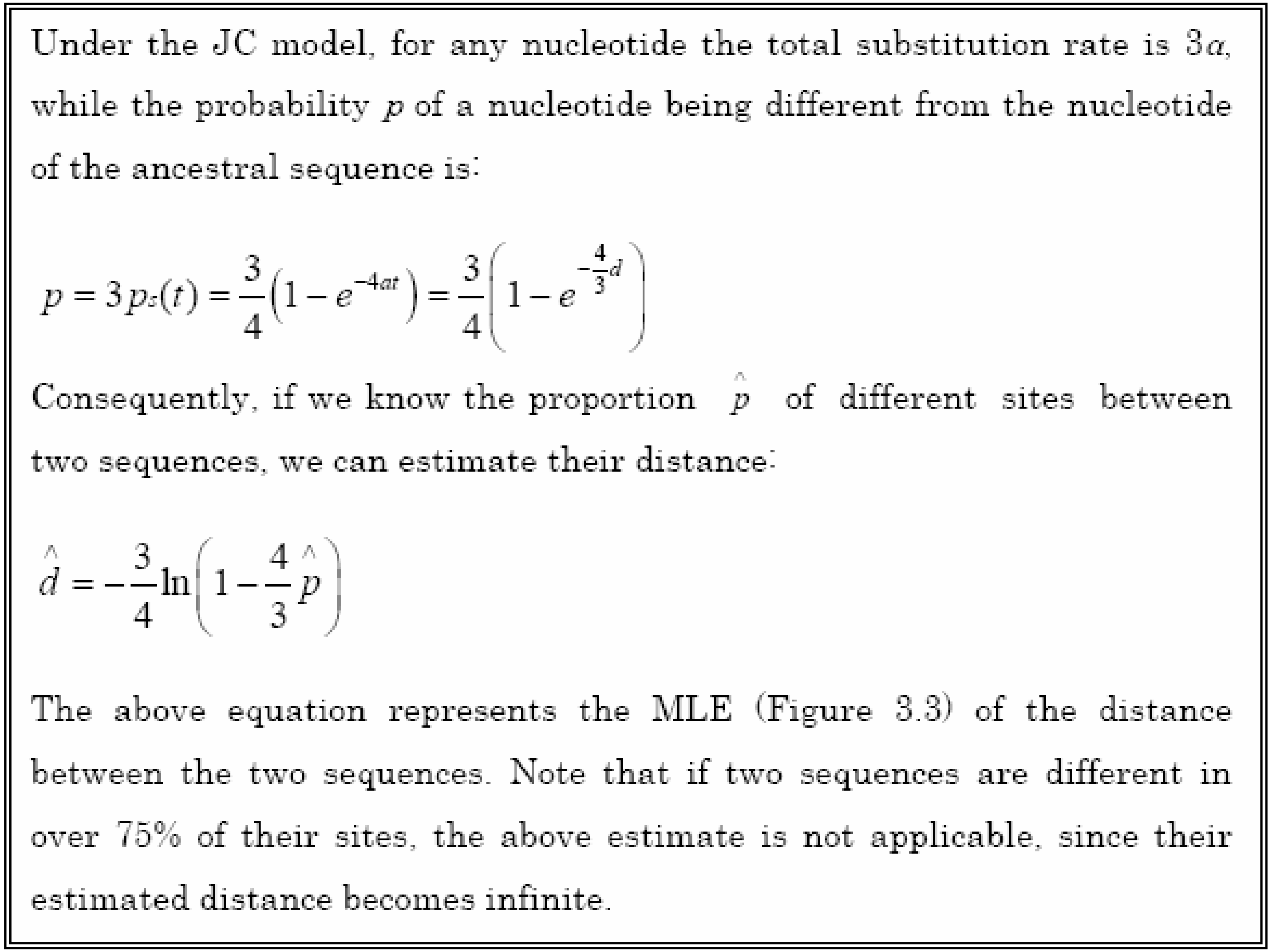


Figure 3.6: Jukes and Cantor model: transition probabilities $p_r(t)$ and $p_s(t)$ plotted against distance d ($=3\alpha t$); d is expressed as the expected number of substitutions per site. Assuming that for any nucleotide, the total substitution rate is 3α (see *substitution-rate* matrix Q), if two hypothetical sequences are separated by time t (i.e. diverged from their common ancestor $t/2$ ago) the distance d between them is $3\alpha t$.

Jukes Cantor (3)



Under the JC model, for any nucleotide the total substitution rate is 3α, while the probability p of a nucleotide being different from the nucleotide of the ancestral sequence is:

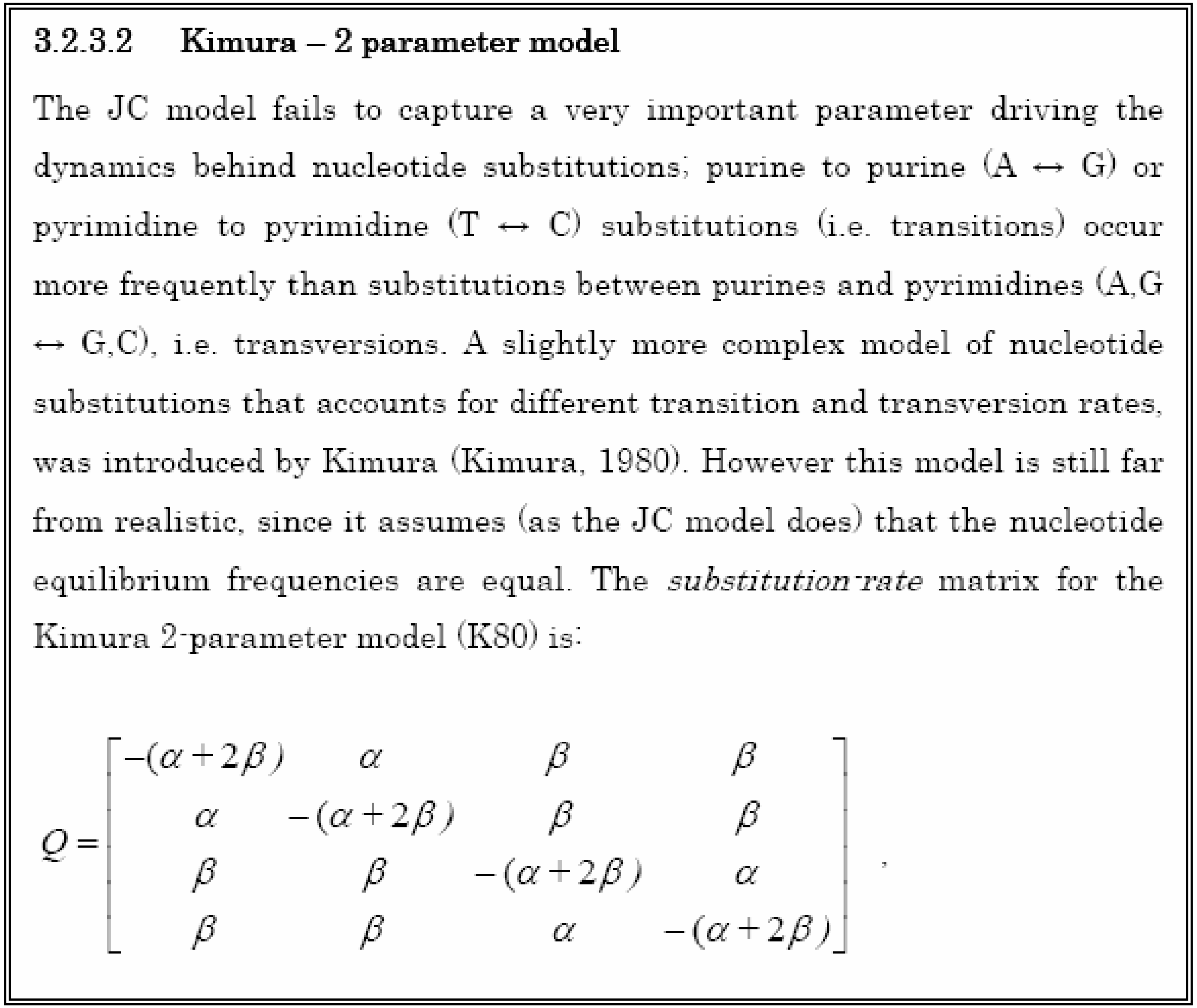
$$p = 3p_{s}(t) = \frac{3}{4}\left(1 - e^{-4\alpha t}\right) = \frac{3}{4}\left(1 - e^{-\frac{4}{3}d}\right)$$

Consequently, if we know the proportion $\hat{p}$ of different sites between two sequences, we can estimate their distance:

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

The above equation represents the MLE (Figure 3.3) of the distance between the two sequences. Note that if two sequences are different in over 75% of their sites, the above estimate is not applicable, since their estimated distance becomes infinite.

Kimura (1)

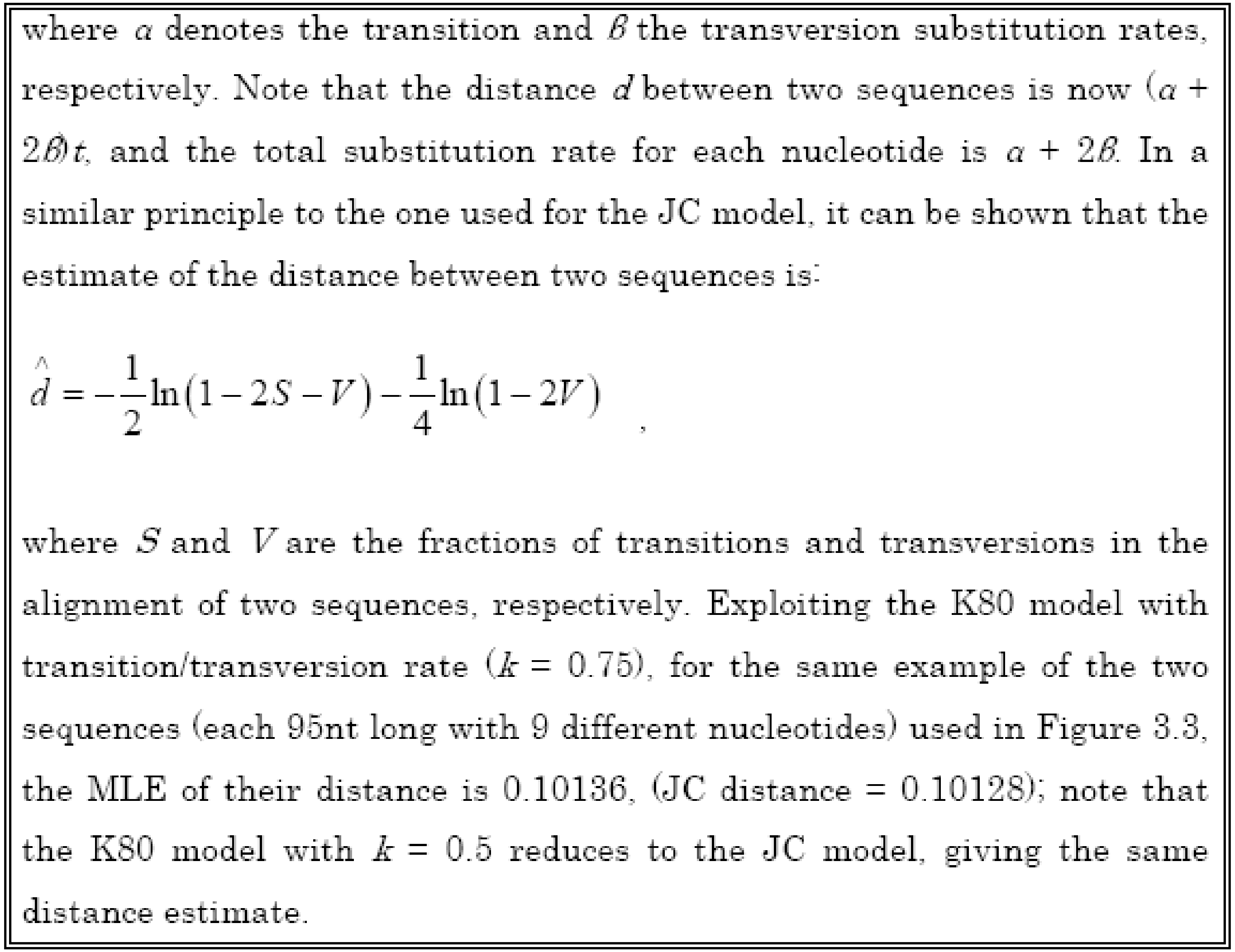


3.2.3.2 Kimura – 2 parameter model

The JC model fails to capture a very important parameter driving the dynamics behind nucleotide substitutions; purine to purine (A ↔ G) or pyrimidine to pyrimidine (T ↔ C) substitutions (i.e. transitions) occur more frequently than substitutions between purines and pyrimidines (A,G ↔ G,C), i.e. transversions. A slightly more complex model of nucleotide substitutions that accounts for different transition and transversion rates, was introduced by Kimura (Kimura, 1980). However this model is still far from realistic, since it assumes (as the JC model does) that the nucleotide equilibrium frequencies are equal. The *substitution-rate* matrix for the Kimura 2-parameter model (K80) is:

$$Q = \begin{bmatrix} -(\alpha+2\beta) & \alpha & \beta & \beta \\ \alpha & -(\alpha+2\beta) & \beta & \beta \\ \beta & \beta & -(\alpha+2\beta) & \alpha \\ \beta & \beta & \alpha & -(\alpha+2\beta) \end{bmatrix},$$

Kimura (2)

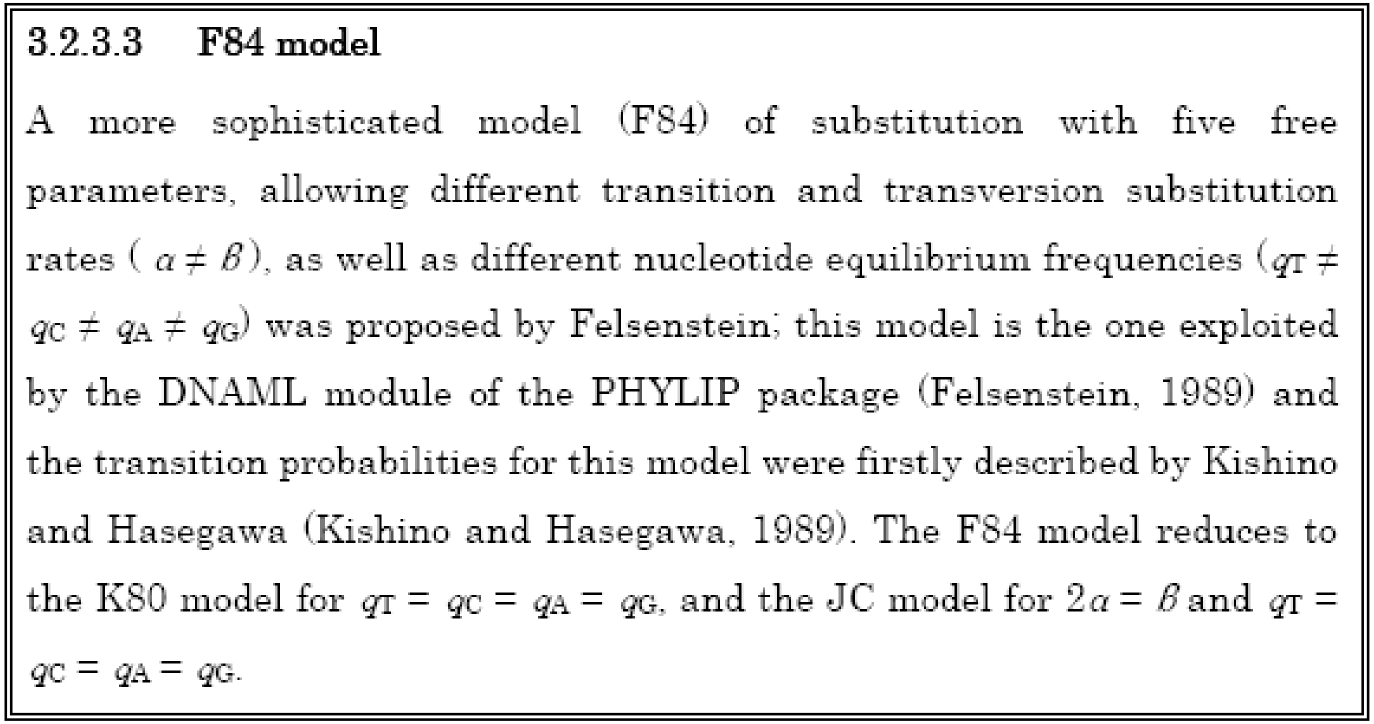


where α denotes the transition and β the transversion substitution rates, respectively. Note that the distance d between two sequences is now $(\alpha + 2\beta)t$, and the total substitution rate for each nucleotide is $\alpha + 2\beta$. In a similar principle to the one used for the JC model, it can be shown that the estimate of the distance between two sequences is:

$$\hat{d} = -\frac{1}{2}\ln(1-2S-V) - \frac{1}{4}\ln(1-2V) \quad ,$$

where S and V are the fractions of transitions and transversions in the alignment of two sequences, respectively. Exploiting the K80 model with transition/transversion rate (k = 0.75), for the same example of the two sequences (each 95nt long with 9 different nucleotides) used in Figure 3.3, the MLE of their distance is 0.10136, (JC distance = 0.10128); note that the K80 model with k = 0.5 reduces to the JC model, giving the same distance estimate.

F84



3.2.3.3 F84 model

A more sophisticated model (F84) of substitution with five free parameters, allowing different transition and transversion substitution rates ($\alpha \neq \beta$), as well as different nucleotide equilibrium frequencies ($q_T \neq q_C \neq q_A \neq q_G$) was proposed by Felsenstein; this model is the one exploited by the DNAML module of the PHYLIP package (Felsenstein, 1989) and the transition probabilities for this model were firstly described by Kishino and Hasegawa (Kishino and Hasegawa, 1989). The F84 model reduces to the K80 model for $q_T = q_C = q_A = q_G$, and the JC model for $2\alpha = \beta$ and $q_T = q_C = q_A = q_G$.

JC, K80, F84

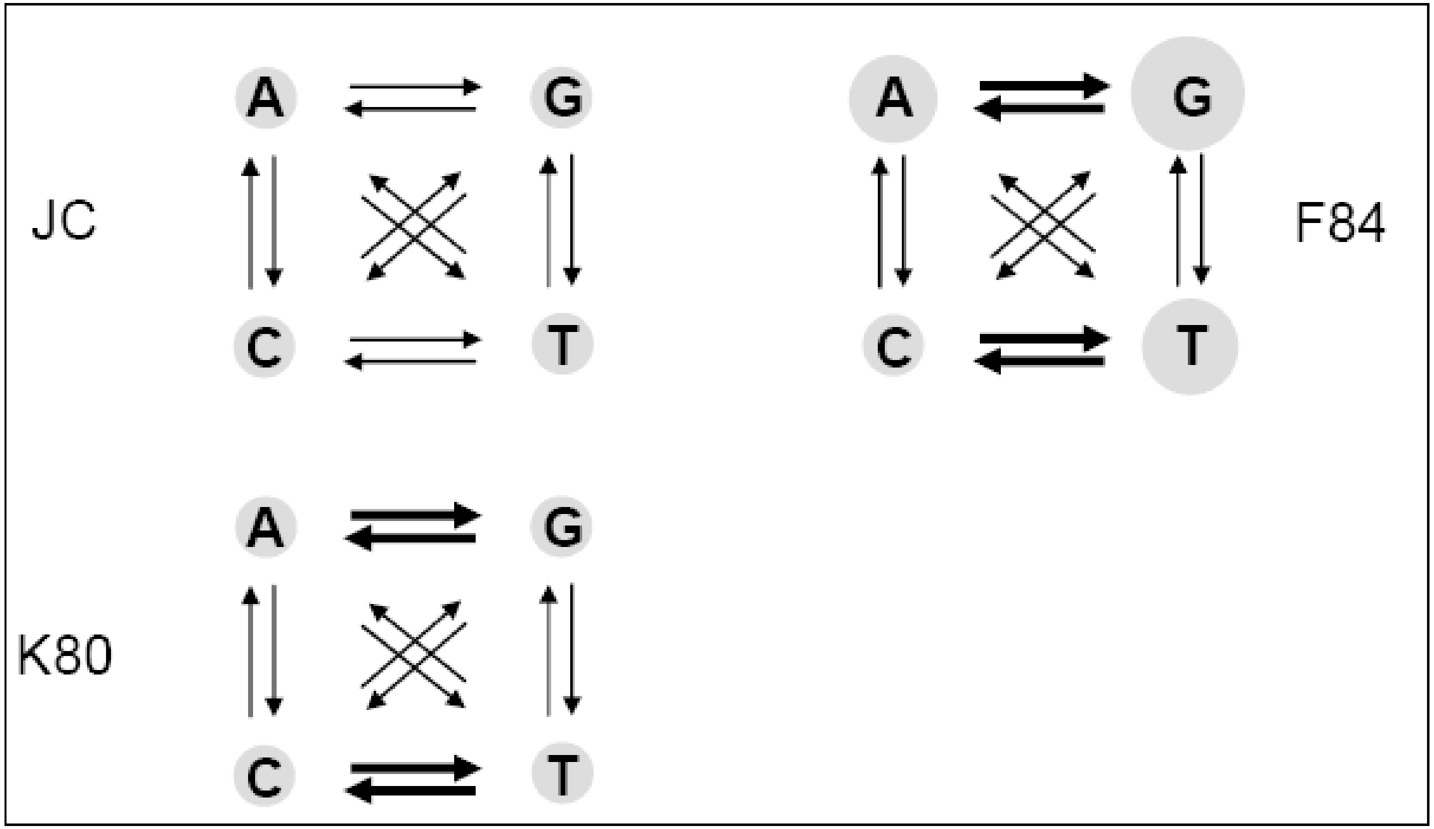
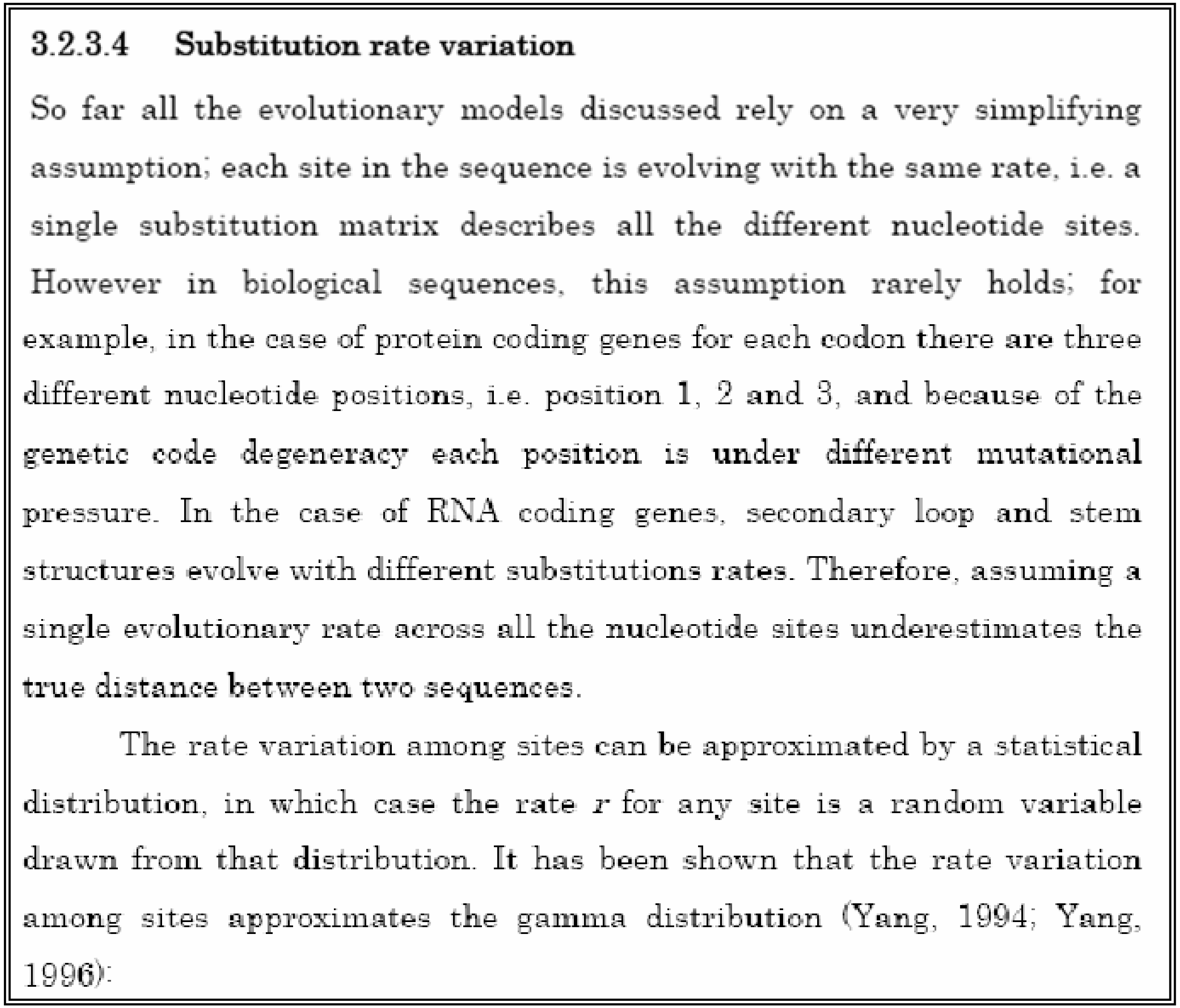


Figure 3.5: Three models of nucleotide substitution; JC (Jukes and Cantor, 1969), K80 (Kimura, 1980) and F84 (Kishino and Hasegawa, 1989).Arrows of different thickness represent different substitution rates and circles of different size the different nucleotide equilibrium frequencies.

Substitution Rate Variation (1)

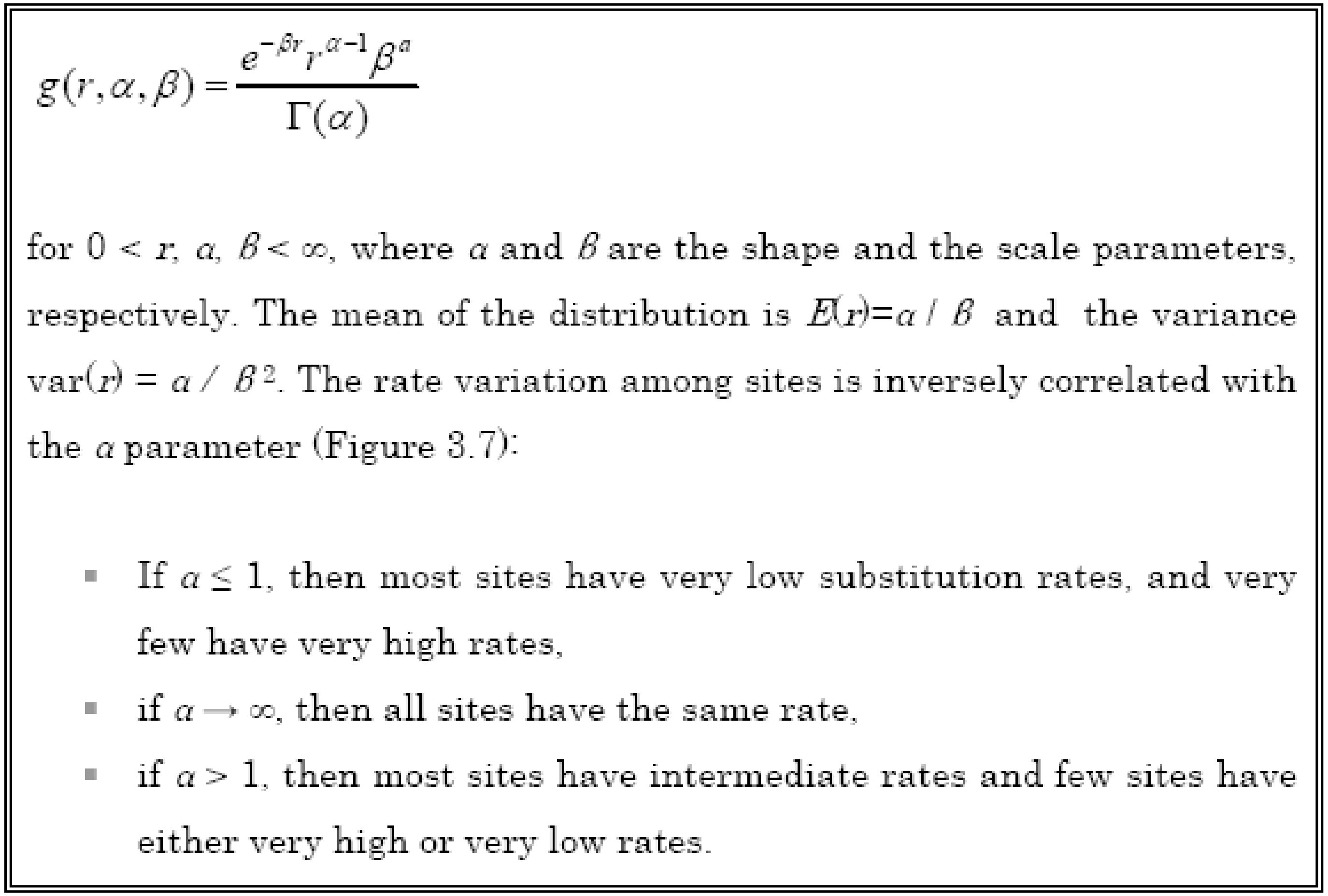


3.2.3.4 Substitution rate variation

So far all the evolutionary models discussed rely on a very simplifying assumption; each site in the sequence is evolving with the same rate, i.e. a single substitution matrix describes all the different nucleotide sites. However in biological sequences, this assumption rarely holds; for example, in the case of protein coding genes for each codon there are three different nucleotide positions, i.e. position 1, 2 and 3, and because of the genetic code degeneracy each position is under different mutational pressure. In the case of RNA coding genes, secondary loop and stem structures evolve with different substitutions rates. Therefore, assuming a single evolutionary rate across all the nucleotide sites underestimates the true distance between two sequences.

The rate variation among sites can be approximated by a statistical distribution, in which case the rate r for any site is a random variable drawn from that distribution. It has been shown that the rate variation among sites approximates the gamma distribution (Yang, 1994; Yang, 1996):

Substitution Rate Variation (2)



$$g(r,\alpha,\beta)=\frac{e^{-\beta r}r^{\alpha-1}\beta^{a}}{\Gamma(\alpha)}$$

for $0 < r,\ \alpha,\ \beta < \infty$, where α and β are the shape and the scale parameters, respectively. The mean of the distribution is $E(r)=\alpha\ /\ \beta$ and the variance $\mathrm{var}(r) = \alpha\ /\ \beta^{2}$. The rate variation among sites is inversely correlated with the α parameter (Figure 3.7):

- If $\alpha \leq 1$, then most sites have very low substitution rates, and very few have very high rates,
- if $\alpha \rightarrow \infty$, then all sites have the same rate,
- if $\alpha > 1$, then most sites have intermediate rates and few sites have either very high or very low rates.

Substitution Rate Variation (3)

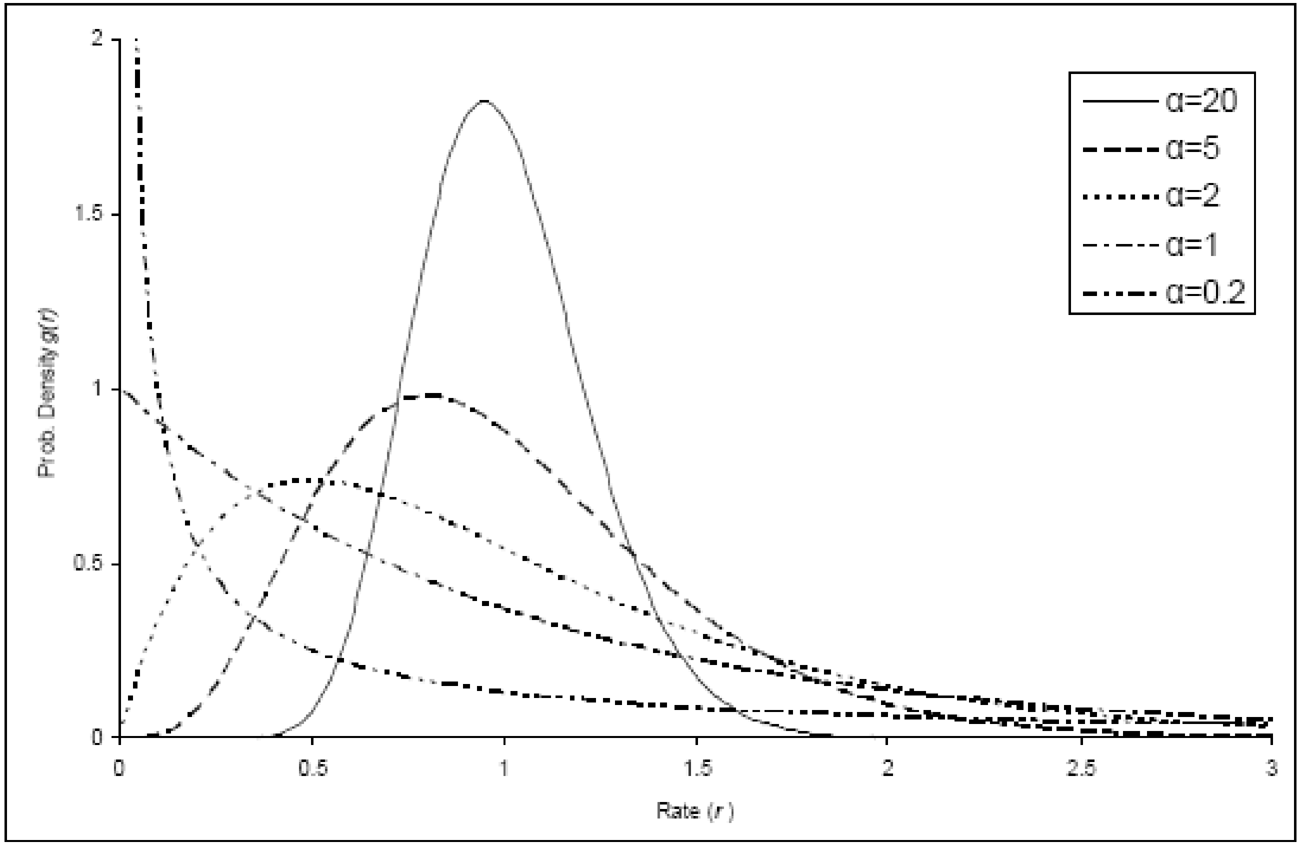
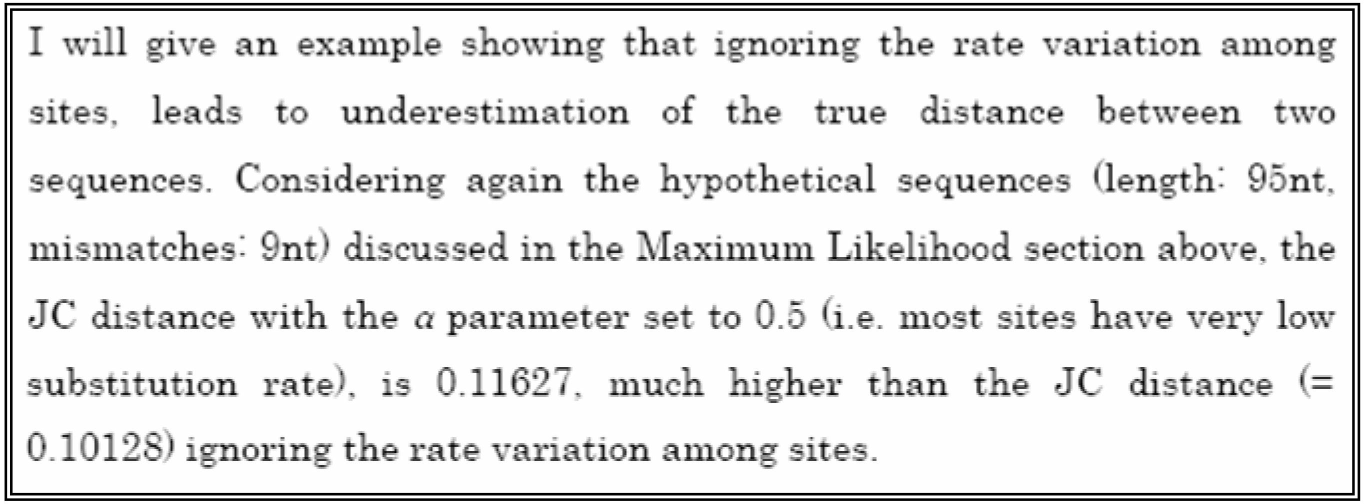


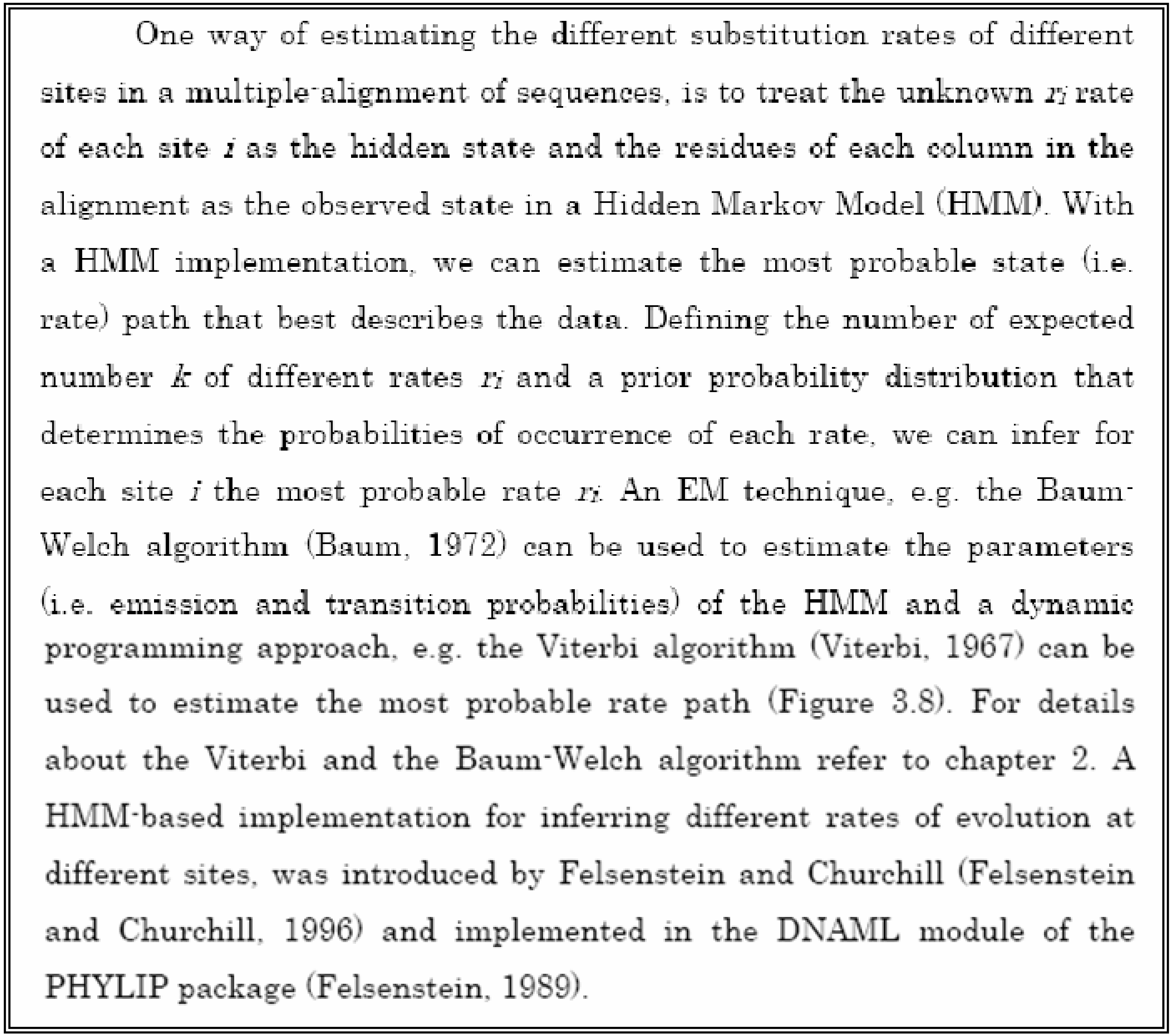
Figure 3.7: *Gamma* distribution $g\,(r,\ \alpha,\ \beta)$; probability densities for different values of the α parameter. In this example, $\alpha = \beta$. The mean of the distribution is $E(r)=\alpha\ /\ \beta = 1$ and the variance $\mathrm{var}(r) = \alpha\ /\ \beta^2 = 1/\alpha$.

Substitution Rate Variation (4)



I will give an example showing that ignoring the rate variation among sites, leads to underestimation of the true distance between two sequences. Considering again the hypothetical sequences (length: 95nt, mismatches: 9nt) discussed in the Maximum Likelihood section above, the JC distance with the α parameter set to 0.5 (i.e. most sites have very low substitution rate), is 0.11627, much higher than the JC distance (= 0.10128) ignoring the rate variation among sites.

Substitution Rate Variation (5)



One way of estimating the different substitution rates of different sites in a multiple-alignment of sequences, is to treat the unknown r_i rate of each site i as the hidden state and the residues of each column in the alignment as the observed state in a Hidden Markov Model (HMM). With a HMM implementation, we can estimate the most probable state (i.e. rate) path that best describes the data. Defining the number of expected number k of different rates r_i and a prior probability distribution that determines the probabilities of occurrence of each rate, we can infer for each site i the most probable rate r_i. An EM technique, e.g. the Baum-Welch algorithm (Baum, 1972) can be used to estimate the parameters (i.e. emission and transition probabilities) of the HMM and a dynamic programming approach, e.g. the Viterbi algorithm (Viterbi, 1967) can be used to estimate the most probable rate path (Figure 3.8). For details about the Viterbi and the Baum-Welch algorithm refer to chapter 2. A HMM-based implementation for inferring different rates of evolution at different sites, was introduced by Felsenstein and Churchill (Felsenstein and Churchill, 1996) and implemented in the DNAML module of the PHYLIP package (Felsenstein, 1989).

Substitution Rate Variation (6)

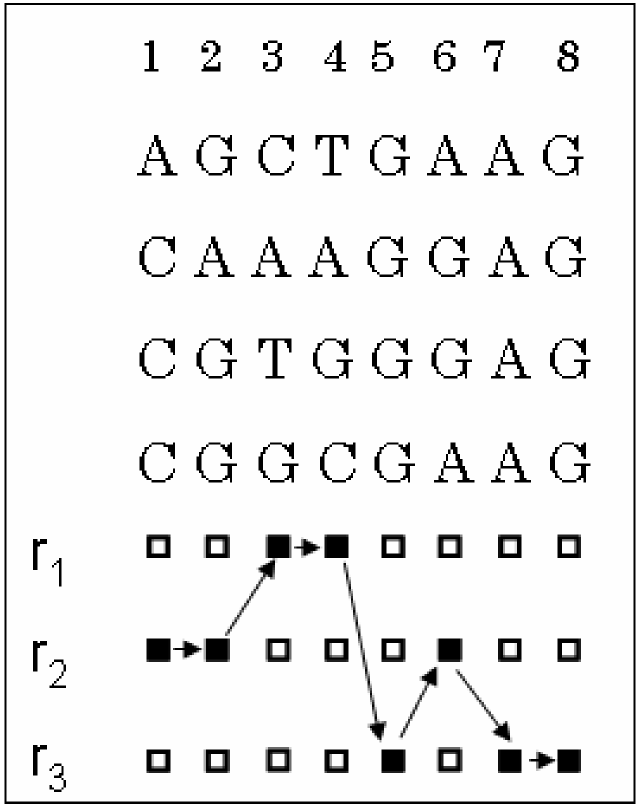
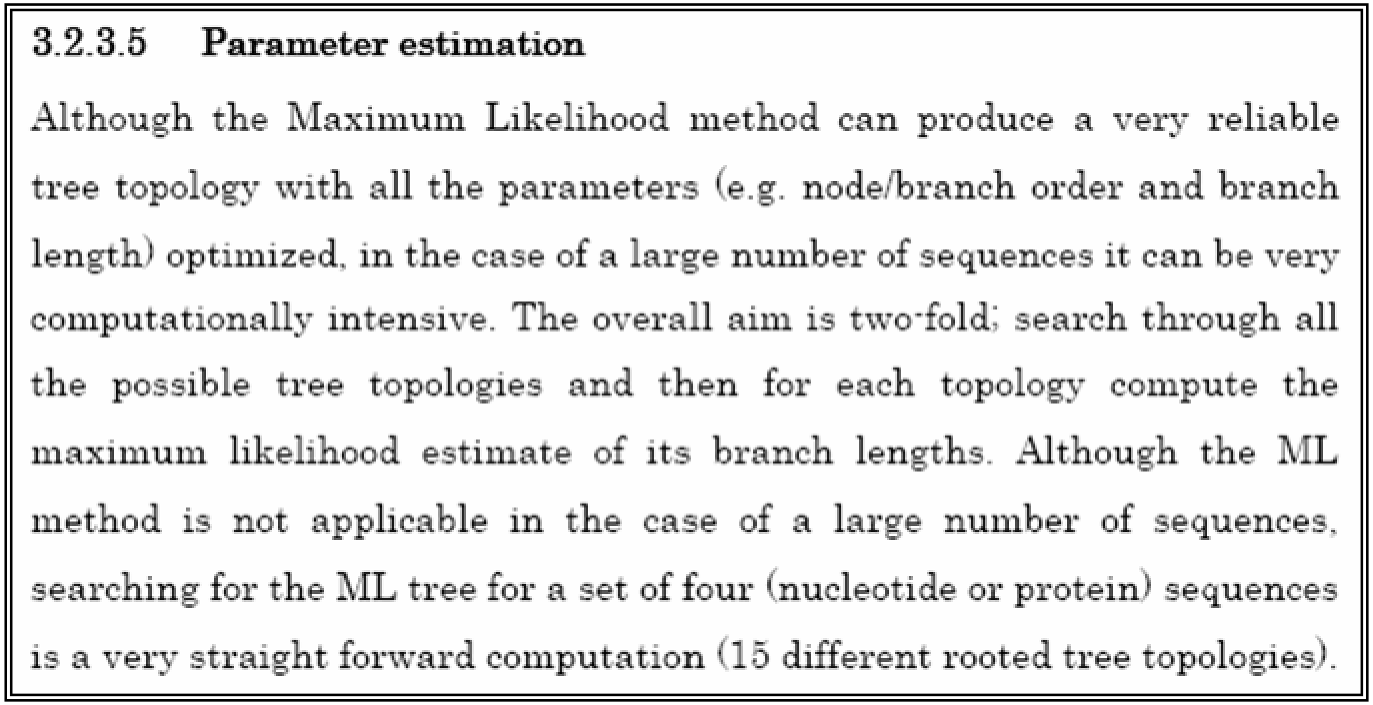


Figure 3.8: An example of four hypothetical sequences, each 8nt long. Each nucleotide site evolves under a different substitution rate ($r_1 > r_2 > r_3$). Assuming that there are k (=3) different substitution rates, implementing a Hidden Markov Model (HMM) approach, we can infer the most likely rate r_f for each site.

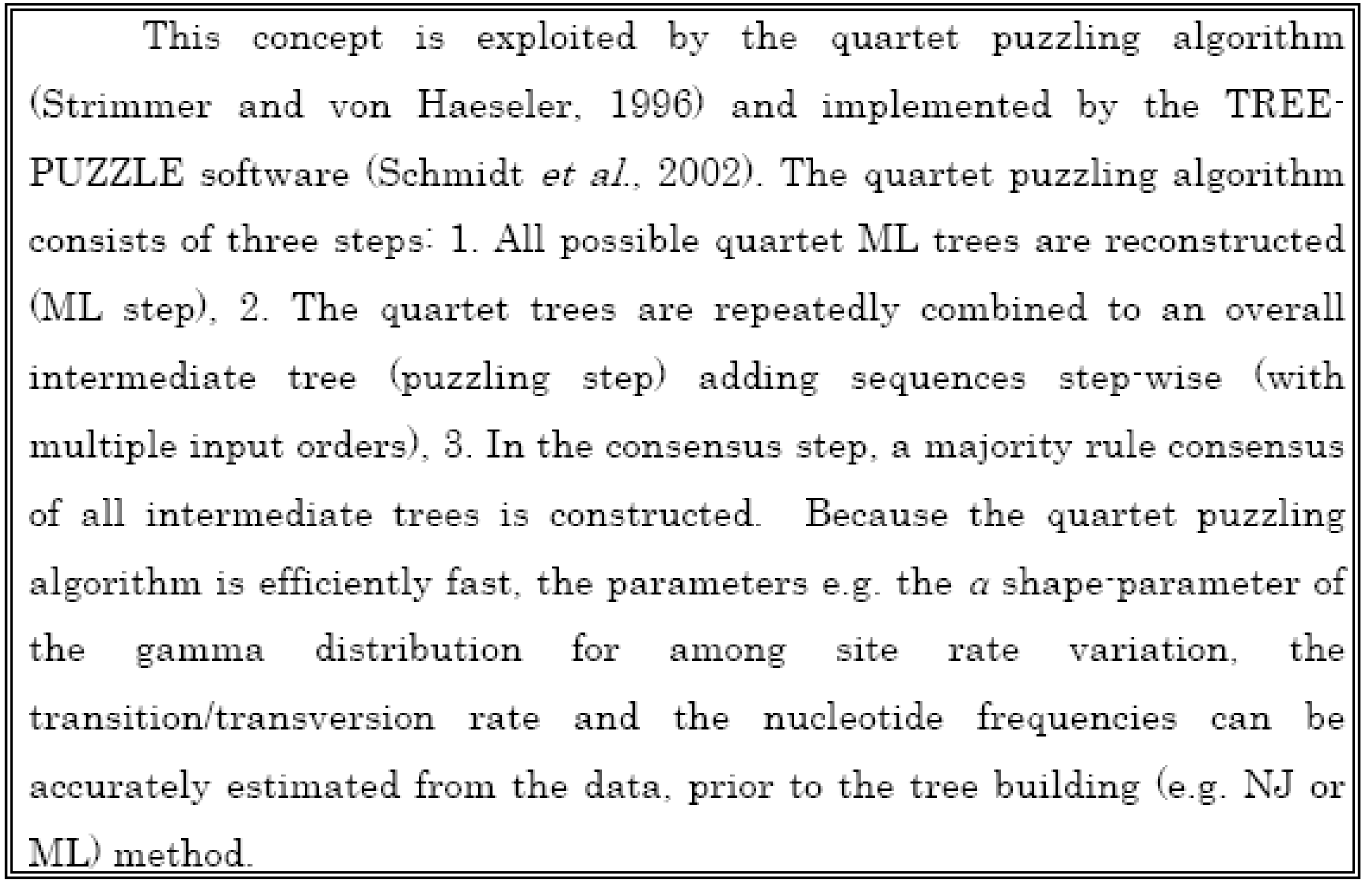
Parameter Estimation (1)



3.2.3.5 Parameter estimation

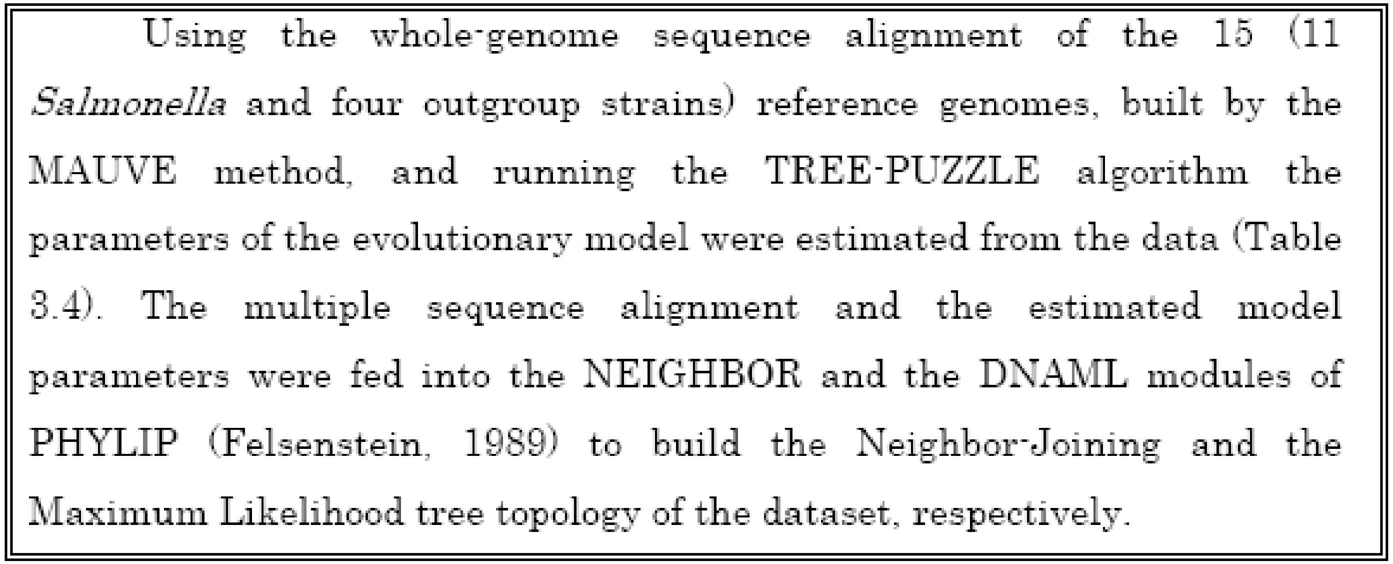
Although the Maximum Likelihood method can produce a very reliable tree topology with all the parameters (e.g. node/branch order and branch length) optimized, in the case of a large number of sequences it can be very computationally intensive. The overall aim is two-fold; search through all the possible tree topologies and then for each topology compute the maximum likelihood estimate of its branch lengths. Although the ML method is not applicable in the case of a large number of sequences, searching for the ML tree for a set of four (nucleotide or protein) sequences is a very straight forward computation (15 different rooted tree topologies).

Parameter Estimation (2)



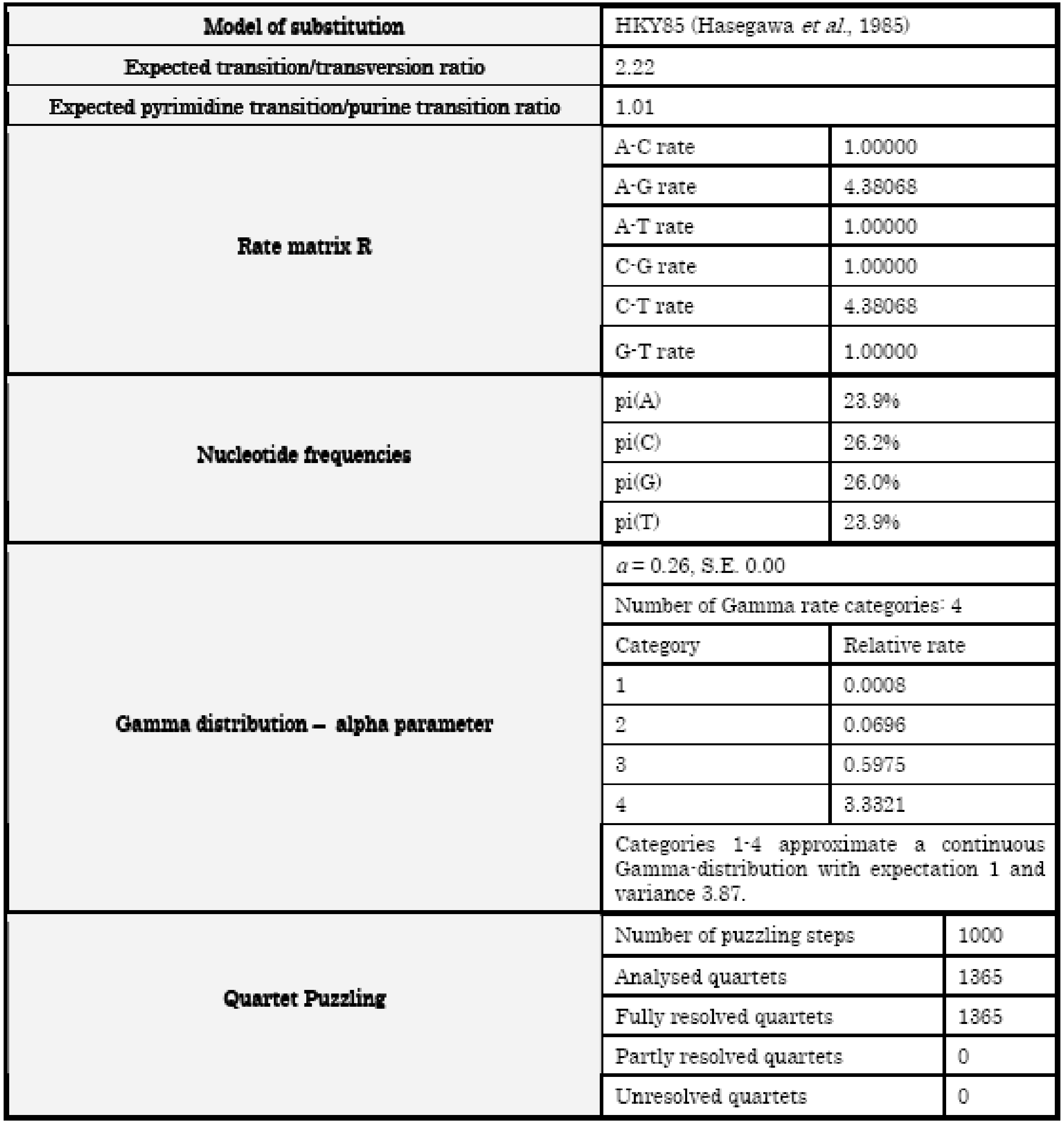
This concept is exploited by the quartet puzzling algorithm (Strimmer and von Haeseler, 1996) and implemented by the TREE-PUZZLE software (Schmidt *et al.*, 2002). The quartet puzzling algorithm consists of three steps: 1. All possible quartet ML trees are reconstructed (ML step), 2. The quartet trees are repeatedly combined to an overall intermediate tree (puzzling step) adding sequences step-wise (with multiple input orders), 3. In the consensus step, a majority rule consensus of all intermediate trees is constructed. Because the quartet puzzling algorithm is efficiently fast, the parameters e.g. the α shape-parameter of the gamma distribution for among site rate variation, the transition/transversion rate and the nucleotide frequencies can be accurately estimated from the data, prior to the tree building (e.g. NJ or ML) method.

Parameter Estimation (3)



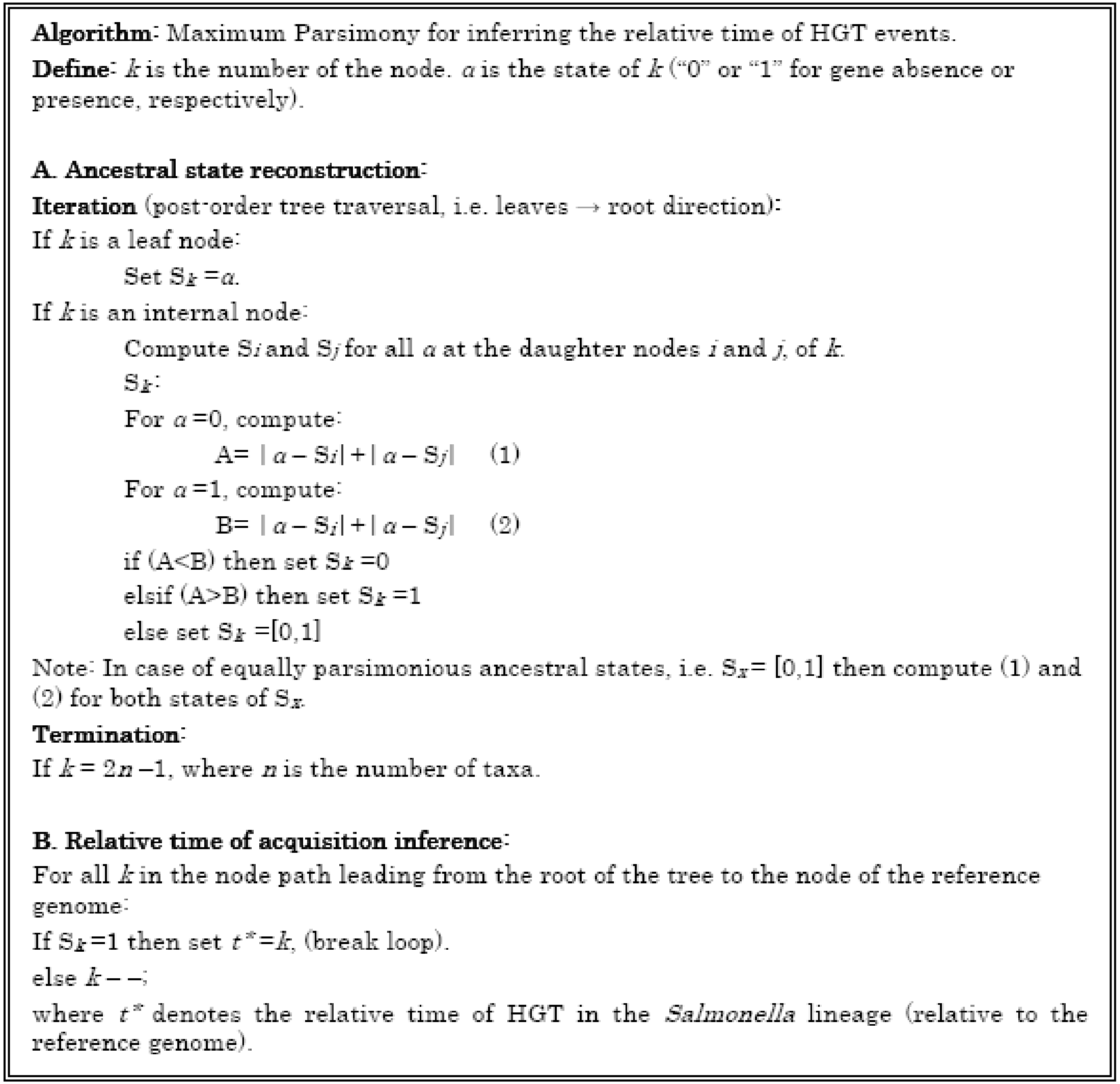
Using the whole-genome sequence alignment of the 15 (11 *Salmonella* and four outgroup strains) reference genomes, built by the MAUVE method, and running the TREE-PUZZLE algorithm the parameters of the evolutionary model were estimated from the data (Table 3.4). The multiple sequence alignment and the estimated model parameters were fed into the NEIGHBOR and the DNAML modules of PHYLIP (Felsenstein, 1989) to build the Neighbor-Joining and the Maximum Likelihood tree topology of the dataset, respectively.

Parameter Estimation (4)



Model of substitution	HKY85 (Hasegawa *et al.*, 1985)	
Expected transition/transversion ratio	2.22	
Expected pyrimidine transition/purine transition ratio	1.01	
Rate matrix R	A-C rate	1.00000
	A-G rate	4.38068
	A-T rate	1.00000
	C-G rate	1.00000
	C-T rate	4.38068
	G-T rate	1.00000
Nucleotide frequencies	pi(A)	23.9%
	pi(C)	26.2%
	pi(G)	26.0%
	pi(T)	23.9%
Gamma distribution – alpha parameter	α = 0.26, S.E. 0.00	
	Number of Gamma rate categories: 4	
	Category	Relative rate
	1	0.0008
	2	0.0696
	3	0.5975
	4	3.3321
	Categories 1-4 approximate a continuous Gamma-distribution with expectation 1 and variance 3.87.	
Quartet Puzzling	Number of puzzling steps	1000
	Analysed quartets	1365
	Fully resolved quartets	1365
	Partly resolved quartets	0
	Unresolved quartets	0

Maximum Parsimony Algorithm



Algorithm: Maximum Parsimony for inferring the relative time of HGT events.

Define: *k* is the number of the node. *α* is the state of *k* ("0" or "1" for gene absence or presence, respectively).

A. Ancestral state reconstruction:

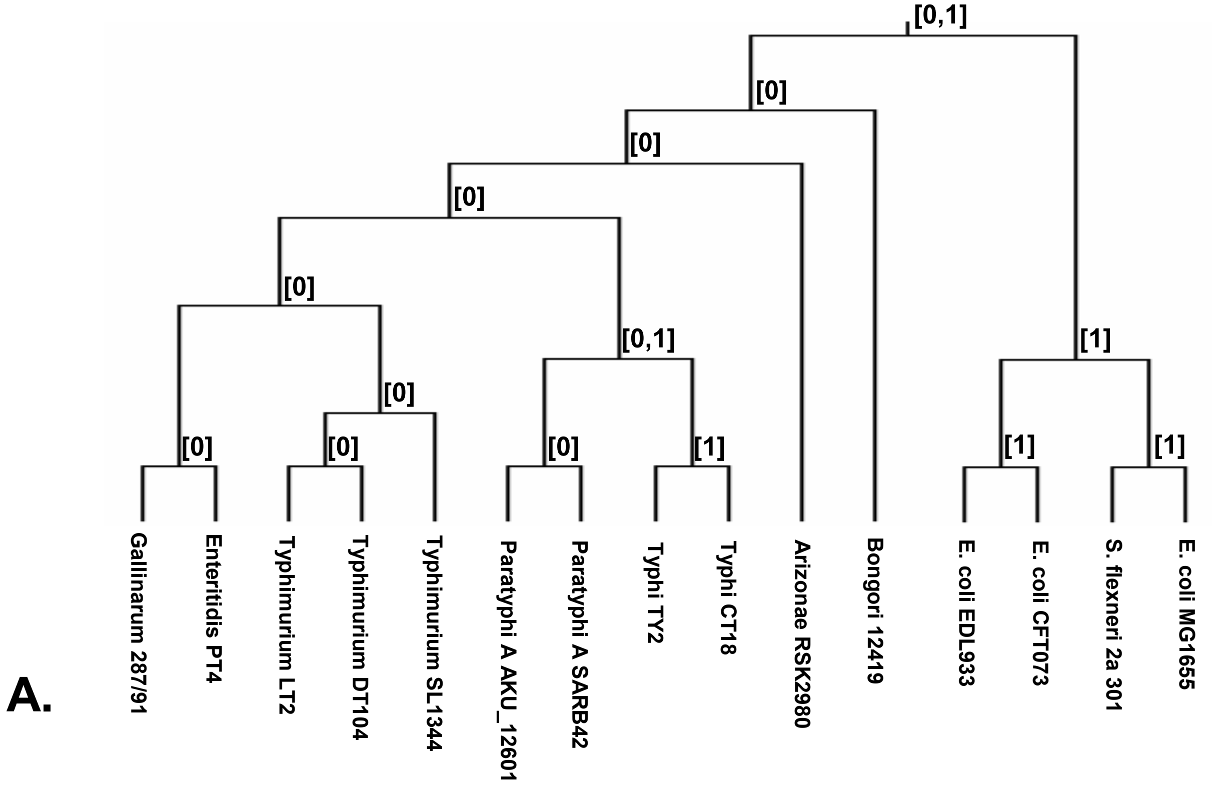
```
Iteration (post-order tree traversal, i.e. leaves → root direction):
If $k$ is a leaf node:
        Set $S_k = α$.
If $k$ is an internal node:
        Compute $S_i$ and $S_j$ for all $α$ at the daughter nodes $i$ and $j$, of $k$.
        $S_k$:
        For $α = 0$, compute:
                $A = |α - S_i| + |α - S_j|$     (1)
        For $α = 1$, compute:
                $B = |α - S_i| + |α - S_j|$     (2)
        if (A<B) then set $S_k = 0$
        elsif (A>B) then set $S_k = 1$
        else set $S_k = [0,1]$
Note: In case of equally parsimonious ancestral states, i.e. $S_x = [0,1]$ then compute (1) and (2) for both states of $S_x$.
Termination:
If $k = 2n - 1$, where $n$ is the number of taxa.
```

B. Relative time of acquisition inference:

```
For all $k$ in the node path leading from the root of the tree to the node of the reference genome:
If $S_k = 1$ then set $t^* = k$, (break loop).
else $k--$;
```

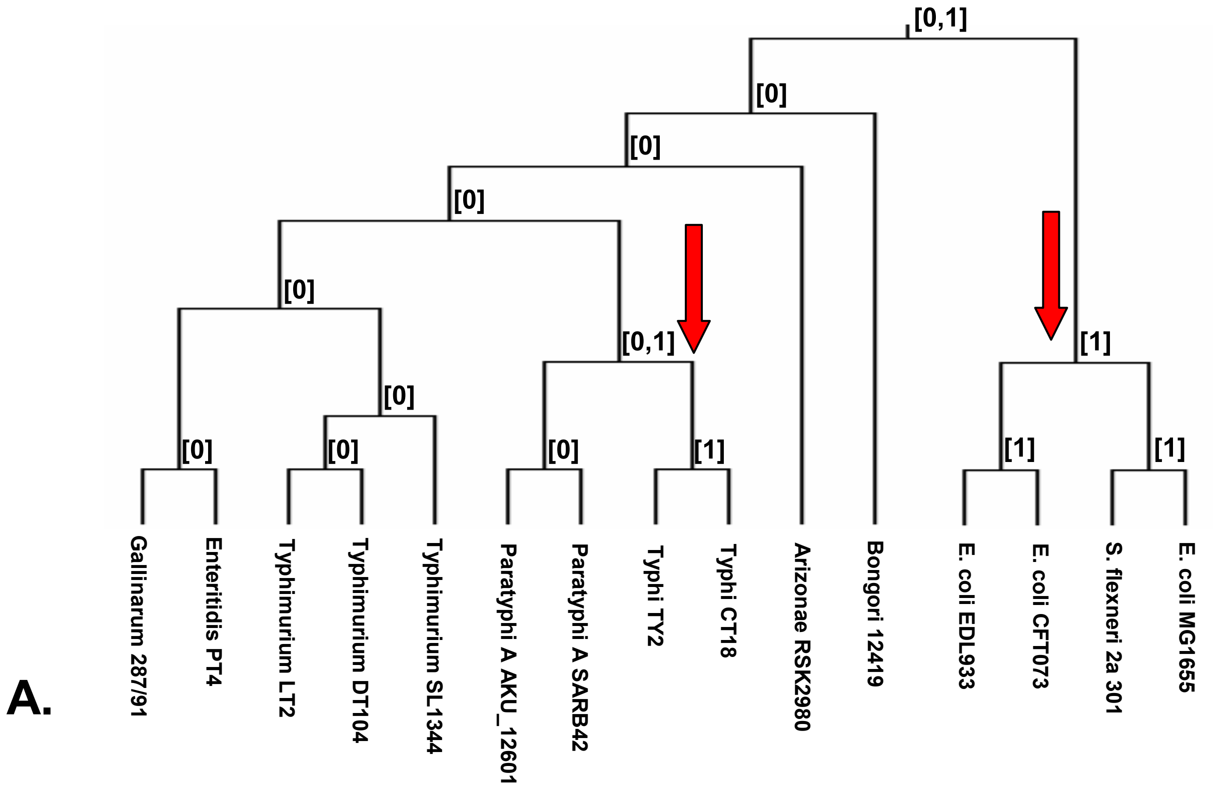
where t^* denotes the relative time of HGT in the *Salmonella* lineage (relative to the reference genome).

Maximum Parsimony Example (1)



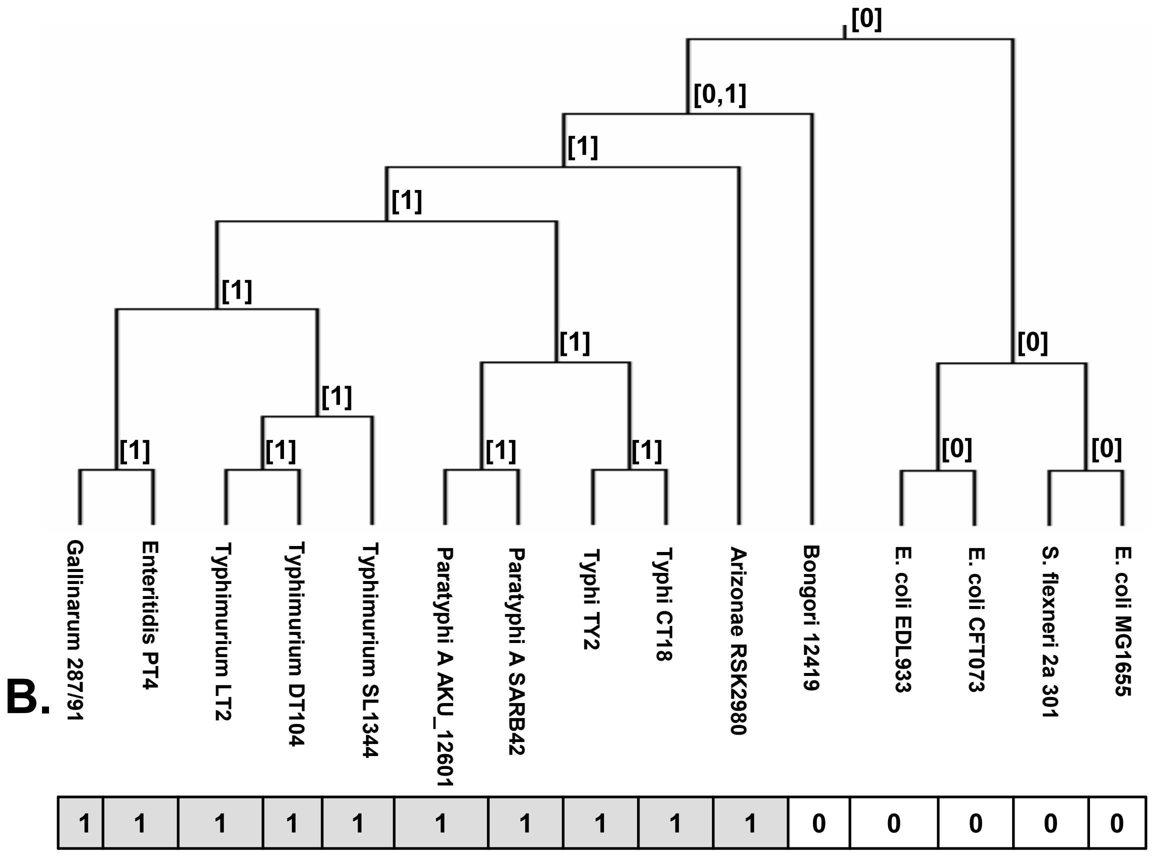
0	0	0	0	0	0	0	1	1	0	0	1	1	1	1

Maximum Parsimony Example (1)



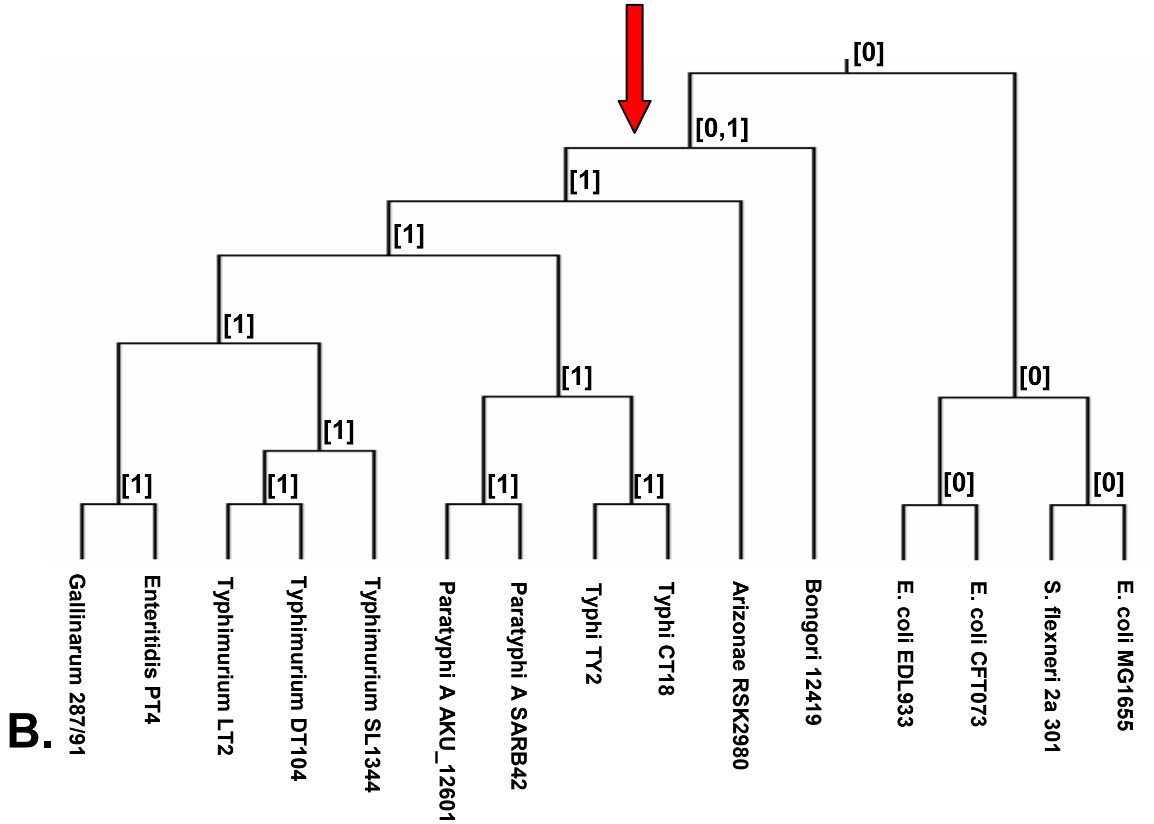
Gallinarum 287/91	Enteritidis PT4	Typhimurium LT2	Typhimurium DT104	Typhimurium SL1344	Paratyphi A AKU_12601	Paratyphi A SARB42	Typhi TY2	Typhi CT18	Arizonae RSK2980	Bongori 12419	E. coli EDL933	E. coli CFT073	S. flexneri 2a 301	E. coli MG1655
0	0	0	0	0	0	0	1	1	0	0	1	1	1	1

Maximum Parsimony Example (2)



1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

Maximum Parsimony Example (2)



1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

Putative Horizontally Acquired Genes - Summary

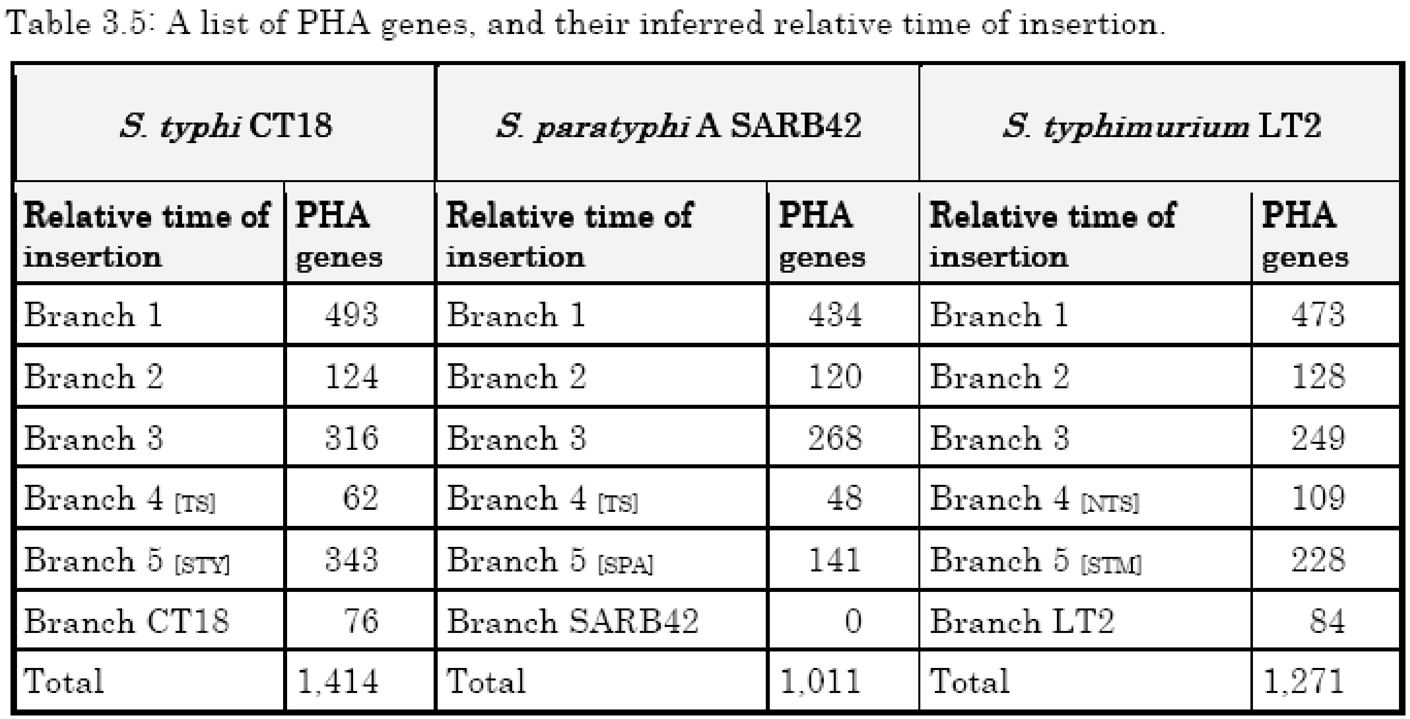
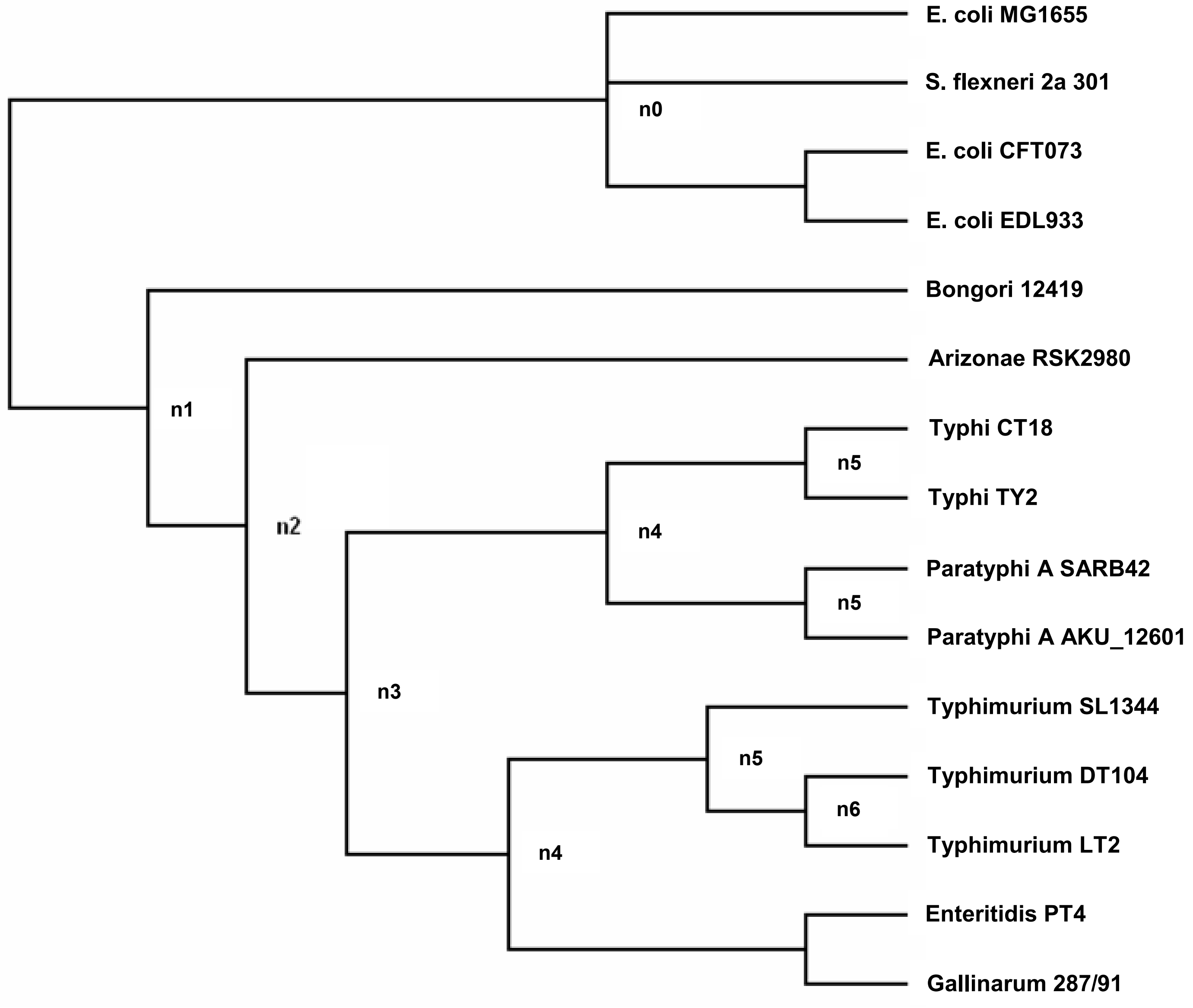


Table 3.5: A list of PHA genes, and their inferred relative time of insertion.

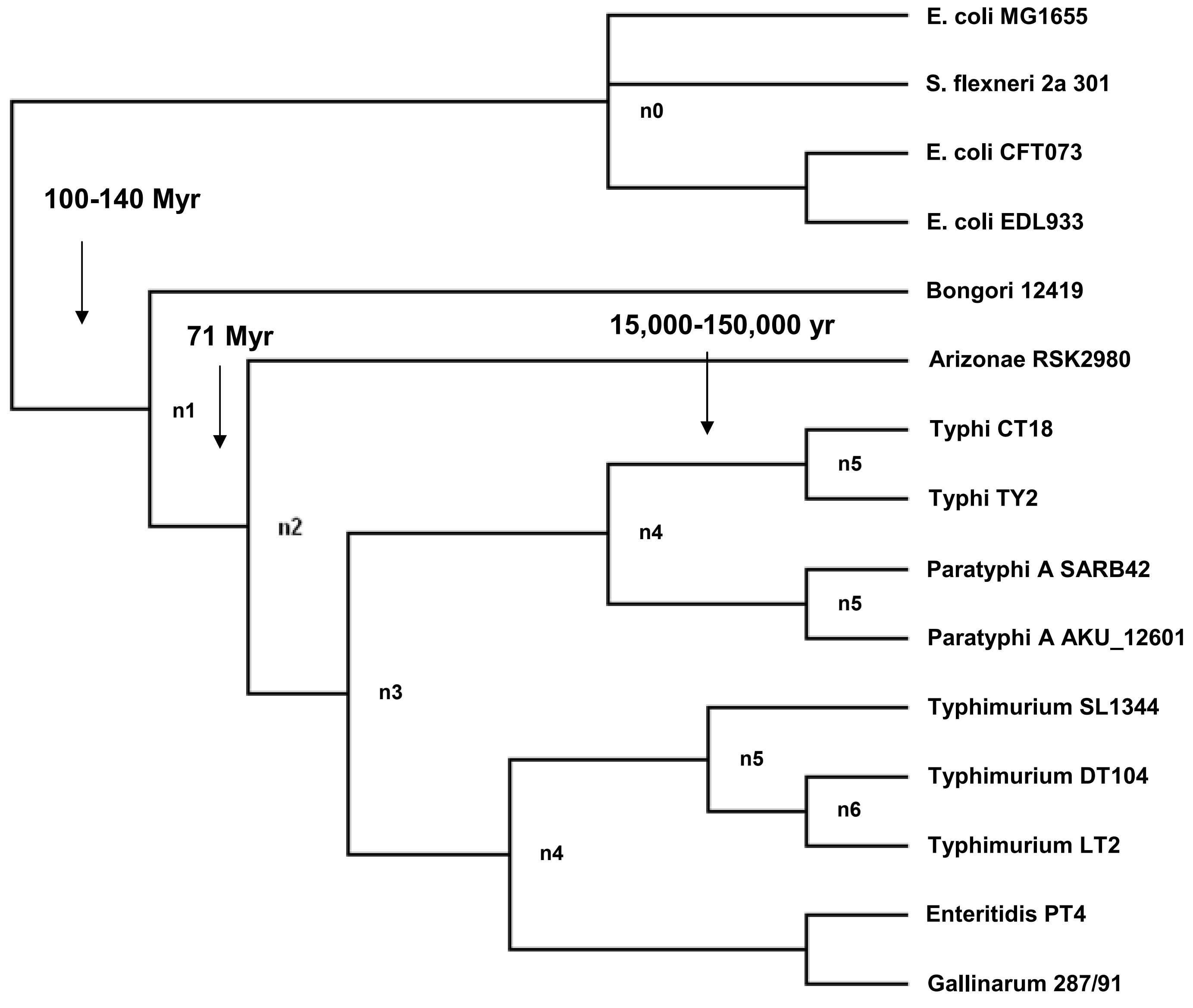
S. typhi CT18		*S. paratyphi* A SARB42		*S. typhimurium* LT2	
Relative time of insertion	**PHA genes**	**Relative time of insertion**	**PHA genes**	**Relative time of insertion**	**PHA genes**
Branch 1	493	Branch 1	434	Branch 1	473
Branch 2	124	Branch 2	120	Branch 2	128
Branch 3	316	Branch 3	268	Branch 3	249
Branch 4 [TS]	62	Branch 4 [TS]	48	Branch 4 [NTS]	109
Branch 5 [STY]	343	Branch 5 [SPA]	141	Branch 5 [STM]	228
Branch CT18	76	Branch SARB42	0	Branch LT2	84
Total	1,414	Total	1,011	Total	1,271

PHAs: Relative Time Distribution



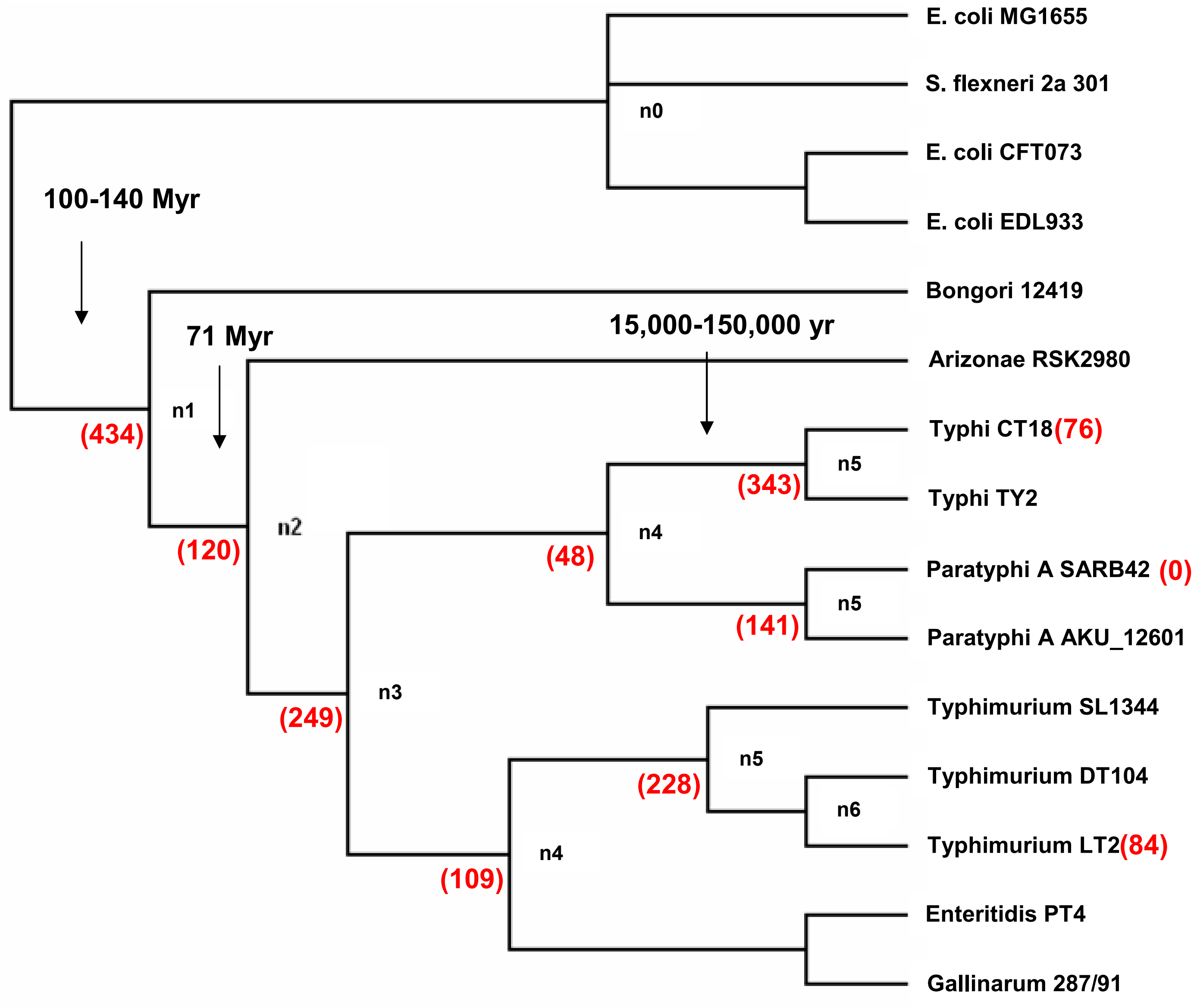
Vernikos GS et al., Genome Biol 2007

PHAs: Relative Time Distribution



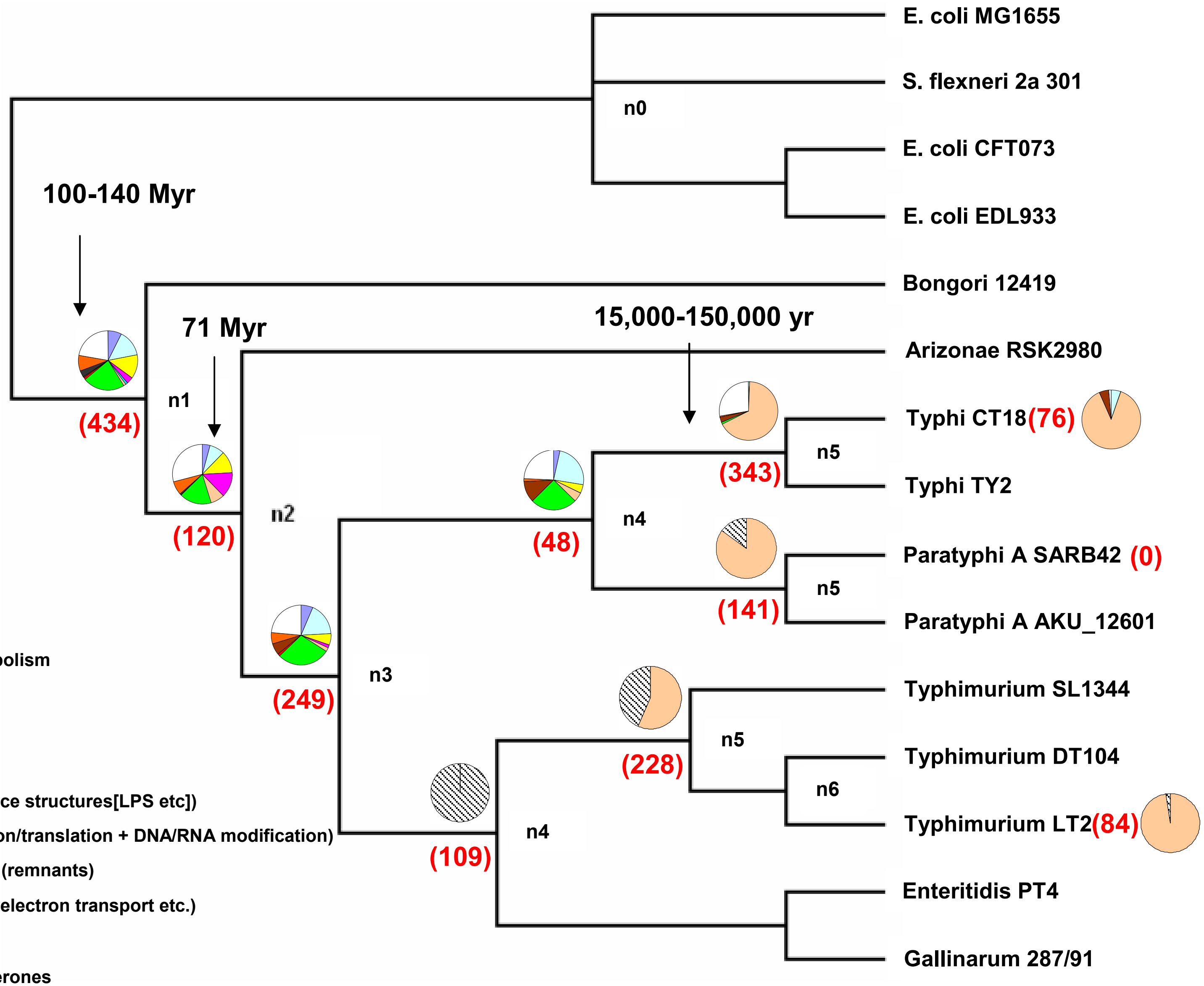
Vernikos GS et al., Genome Biol 2007

PHAs: Relative Time Distribution



Vernikos GS et al., Genome Biol 2007

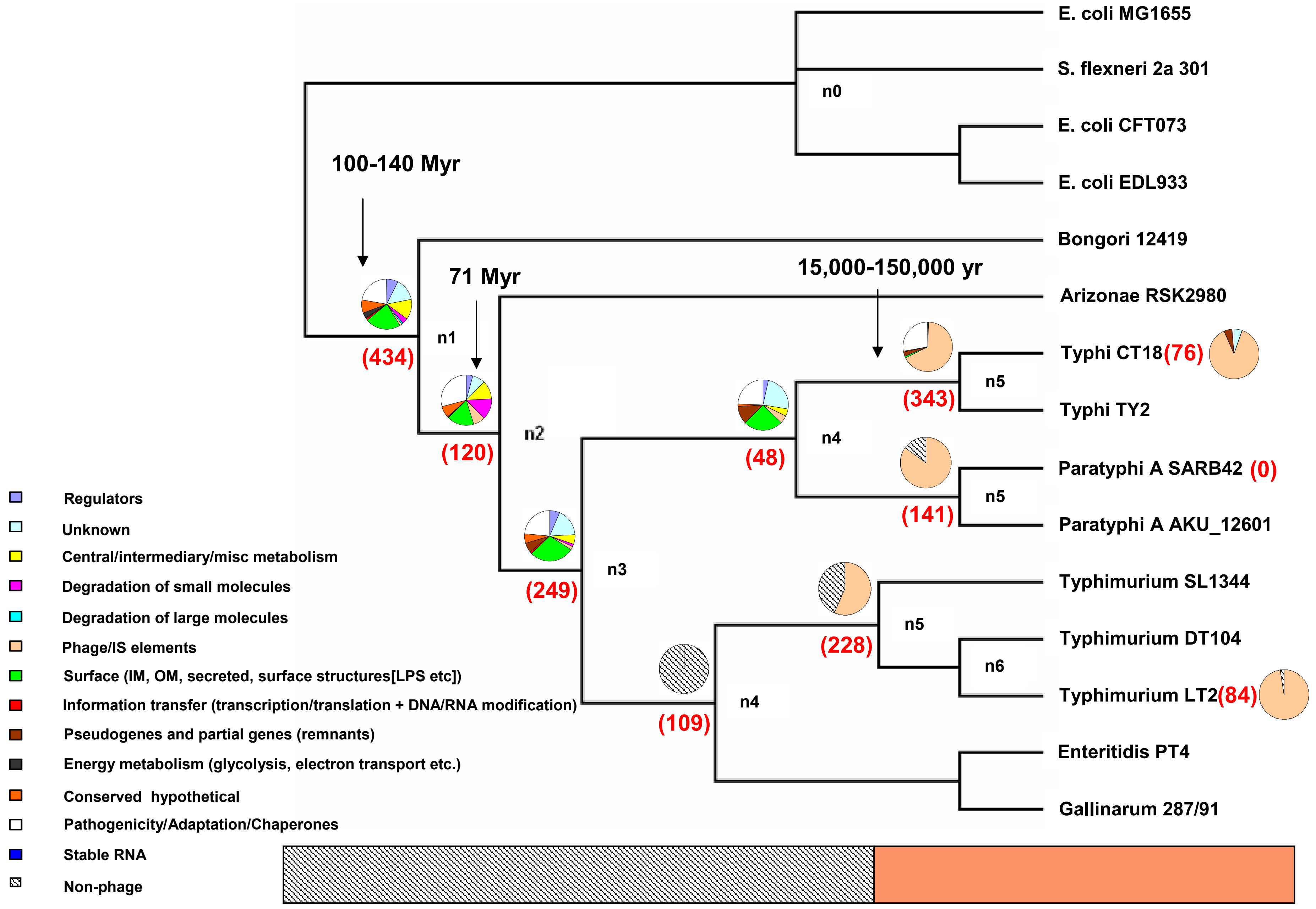
PHAs: Relative Time Distribution



- 🟪 Regulators
- ⬜ Unknown
- 🟨 Central/intermediary/misc metabolism
- 🟪 Degradation of small molecules
- 🟦 Degradation of large molecules
- 🟧 Phage/IS elements
- 🟩 Surface (IM, OM, secreted, surface structures[LPS etc])
- 🟥 Information transfer (transcription/translation + DNA/RNA modification)
- 🟫 Pseudogenes and partial genes (remnants)
- ⬛ Energy metabolism (glycolysis, electron transport etc.)
- 🟧 Conserved hypothetical
- ⬜ Pathogenicity/Adaptation/Chaperones
- 🟦 Stable RNA
- ▨ Non-phage

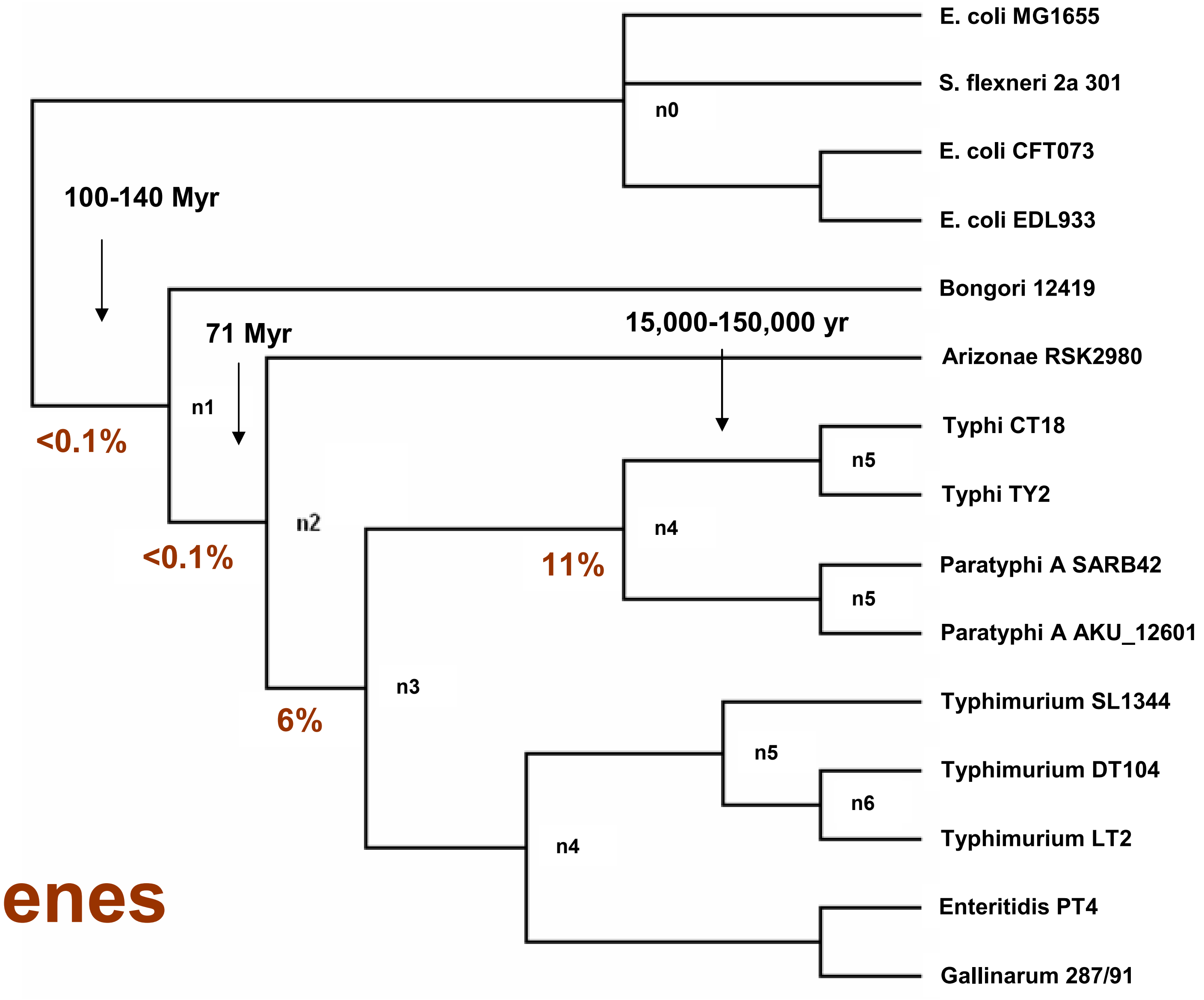
Vernikos GS et al., Genome Biol 2007

PHAs: Relative Time Distribution



- 🟪 Regulators
- ⬜ Unknown
- 🟨 Central/intermediary/misc metabolism
- 🟪 Degradation of small molecules
- 🟦 Degradation of large molecules
- 🟧 Phage/IS elements
- 🟩 Surface (IM, OM, secreted, surface structures[LPS etc])
- 🟥 Information transfer (transcription/translation + DNA/RNA modification)
- 🟫 Pseudogenes and partial genes (remnants)
- ⬛ Energy metabolism (glycolysis, electron transport etc.)
- 🟧 Conserved hypothetical
- □ Pathogenicity/Adaptation/Chaperones
- 🟦 Stable RNA
- ▨ Non-phage

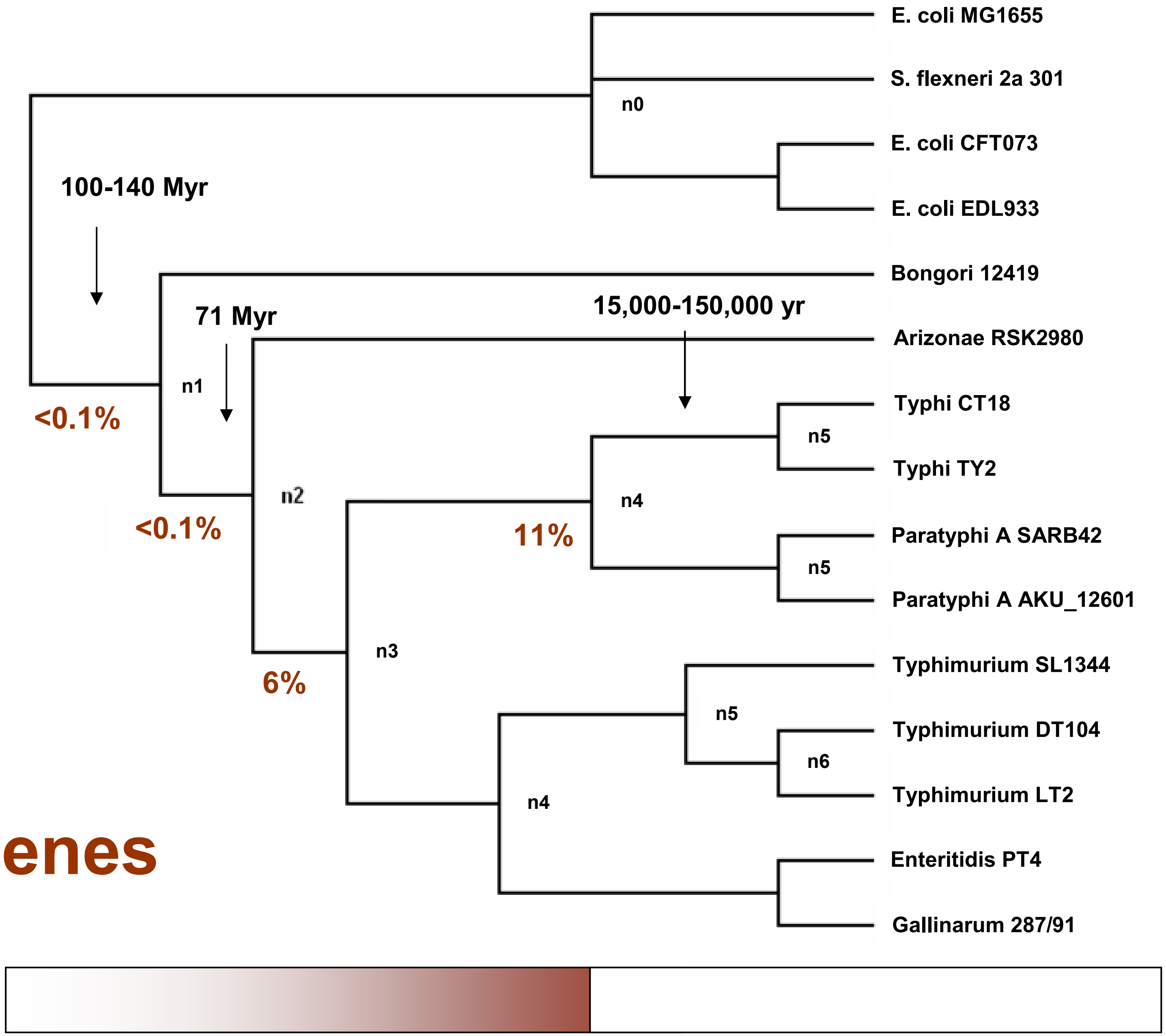
PHAs: Relative Time Distribution



Pseudogenes

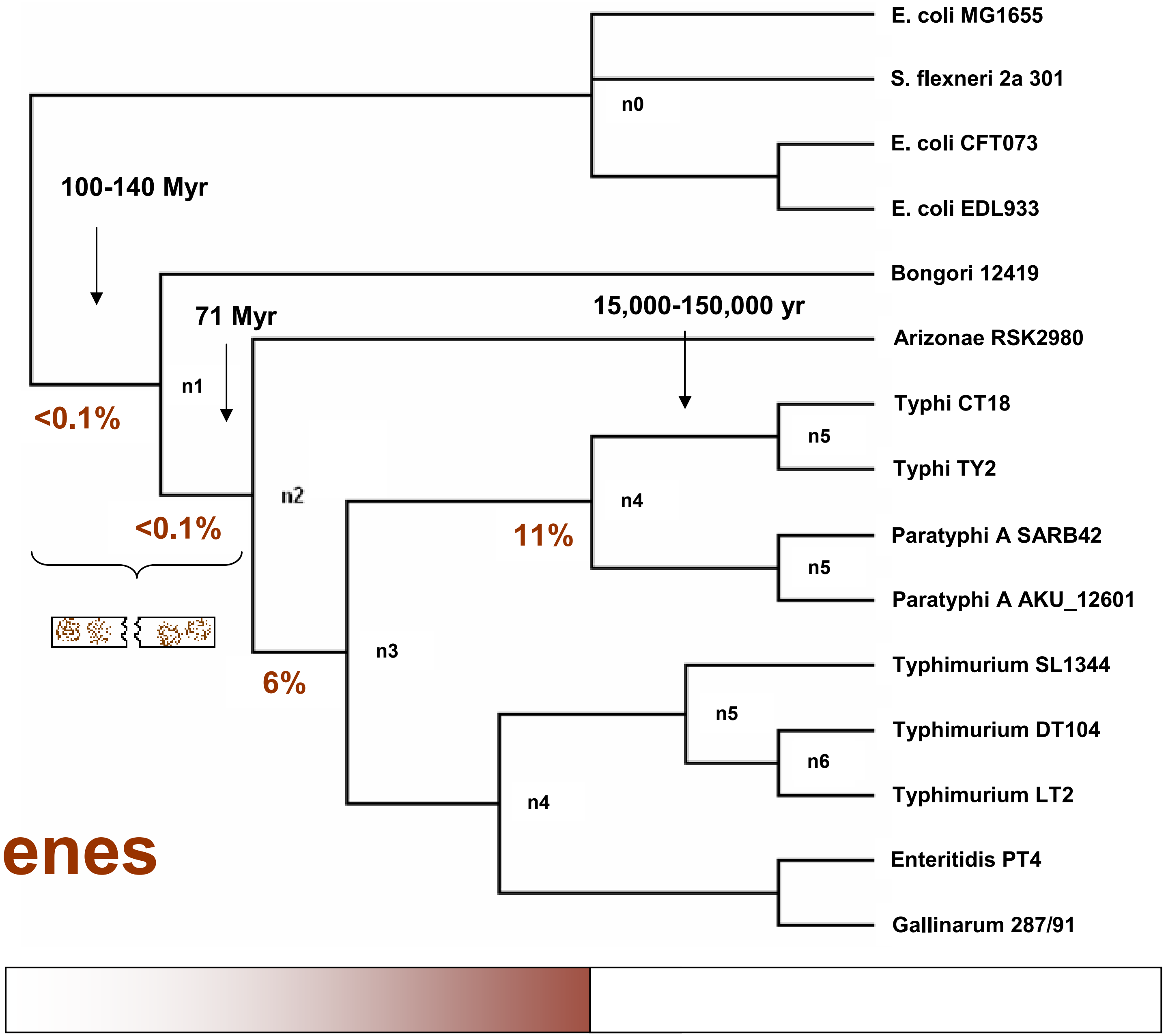
Vernikos GS et al., Genome Biol 2007

PHAs: Relative Time Distribution



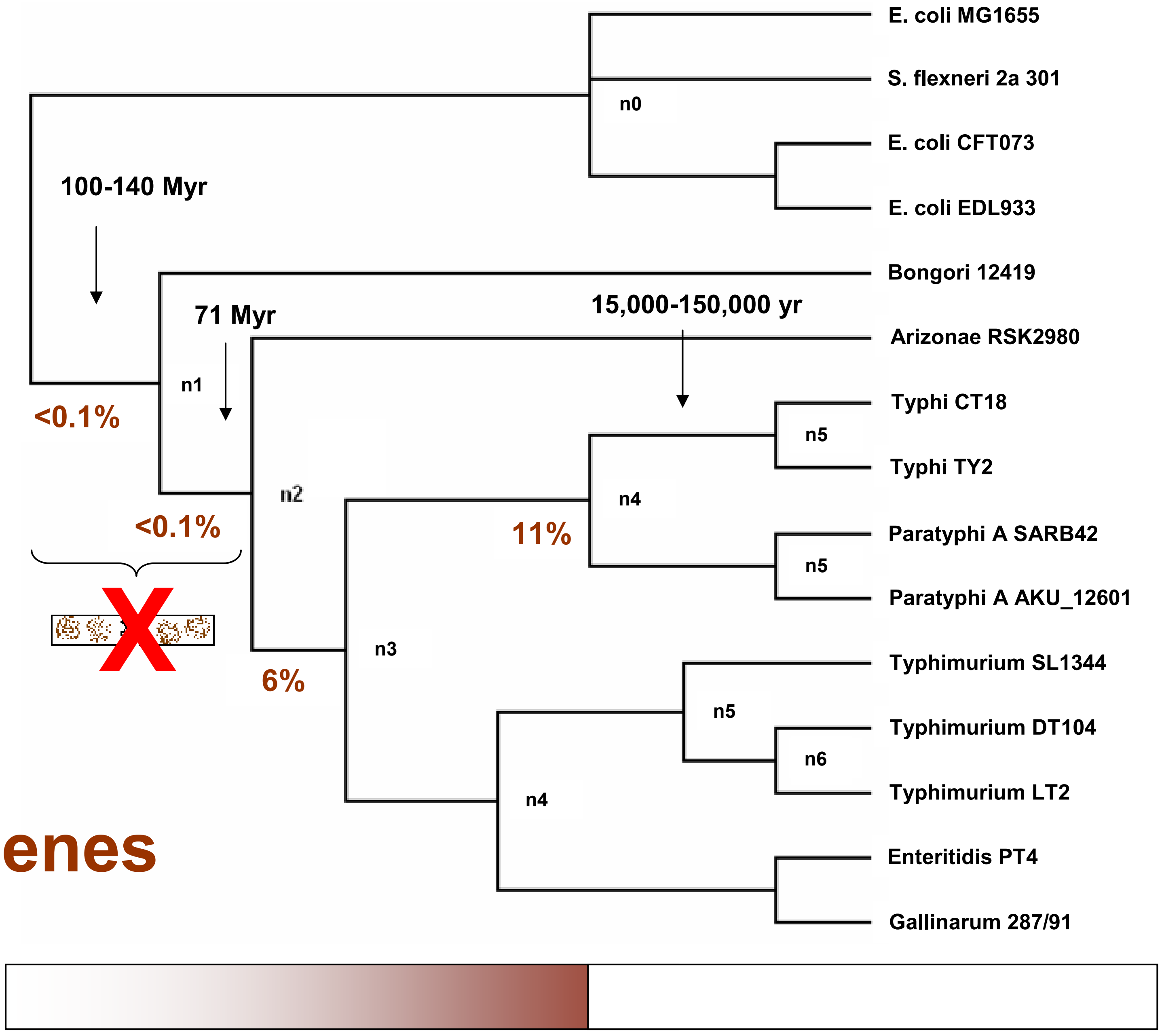
Pseudogenes

PHAs: Relative Time Distribution



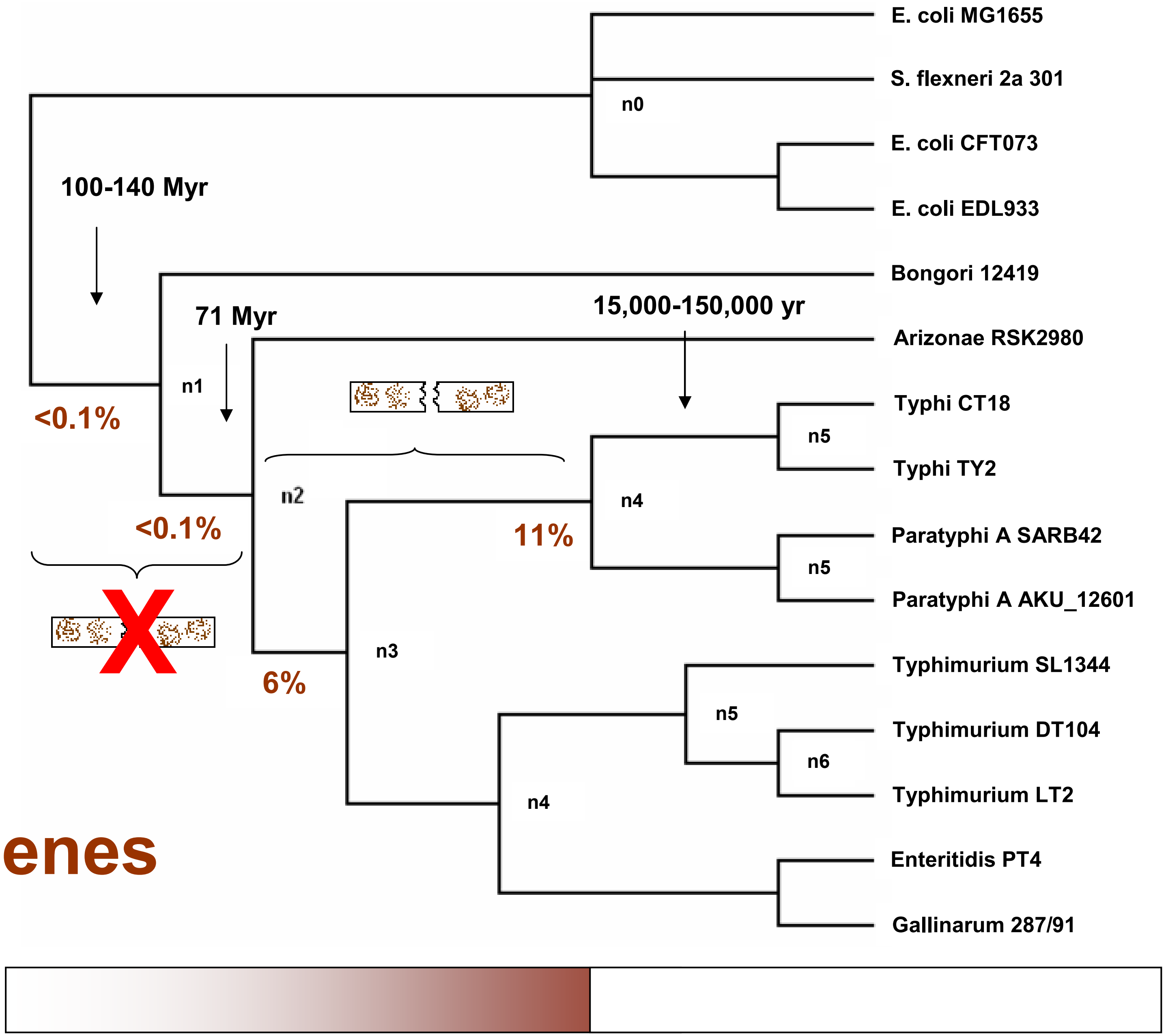
Pseudogenes

PHAs: Relative Time Distribution



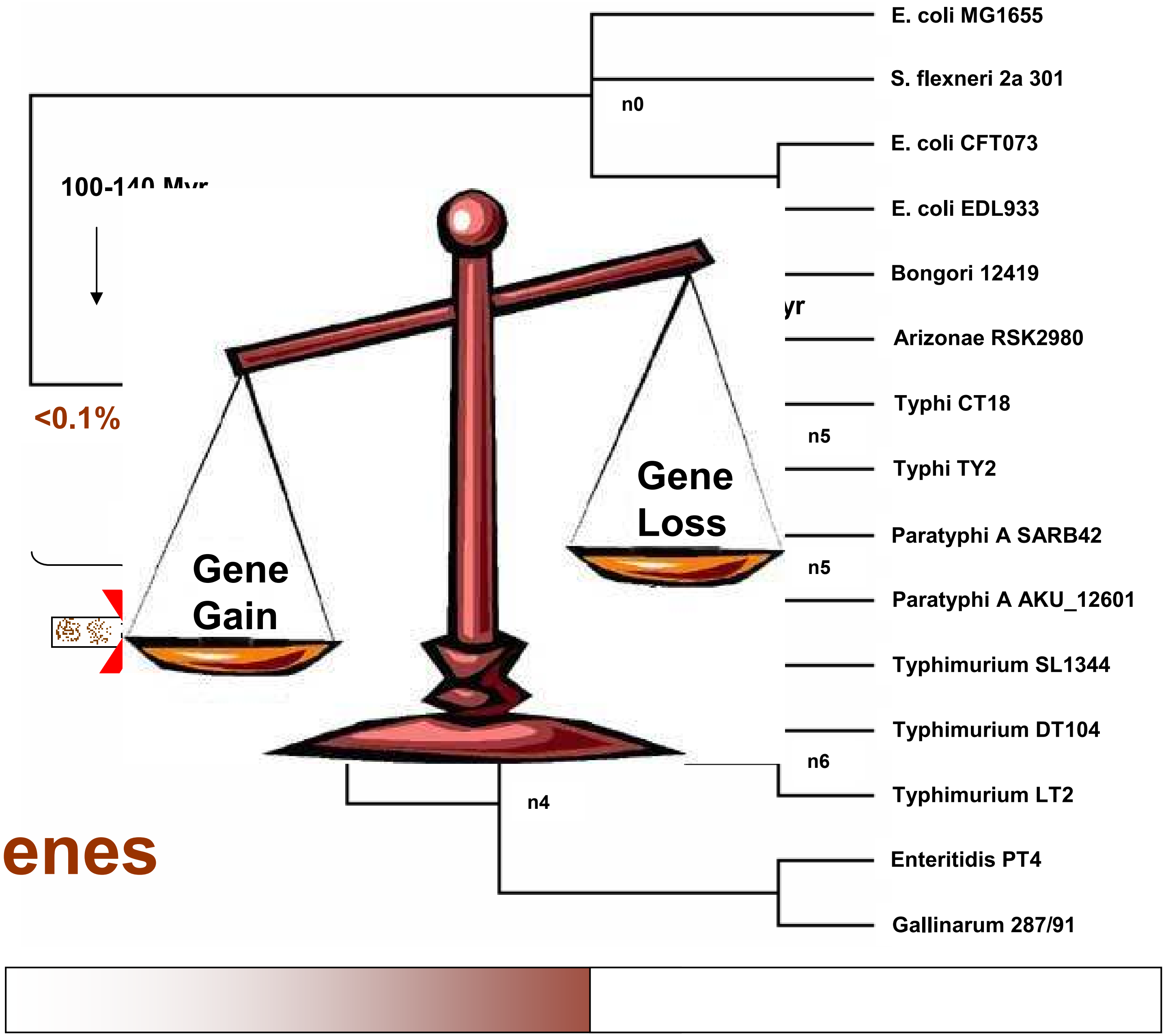
Pseudogenes

PHAs: Relative Time Distribution



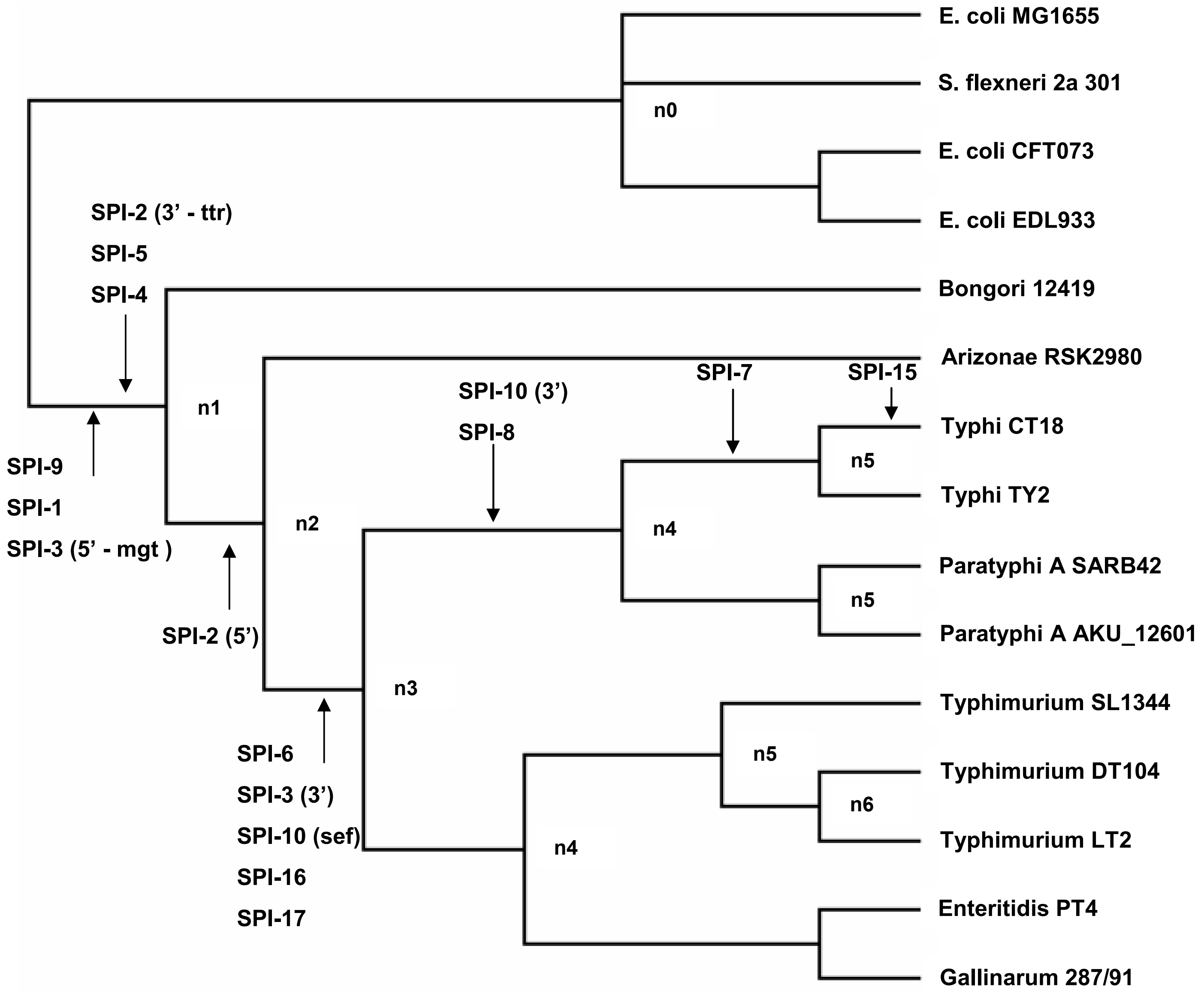
Pseudogenes

PHAs: Relative Time Distribution

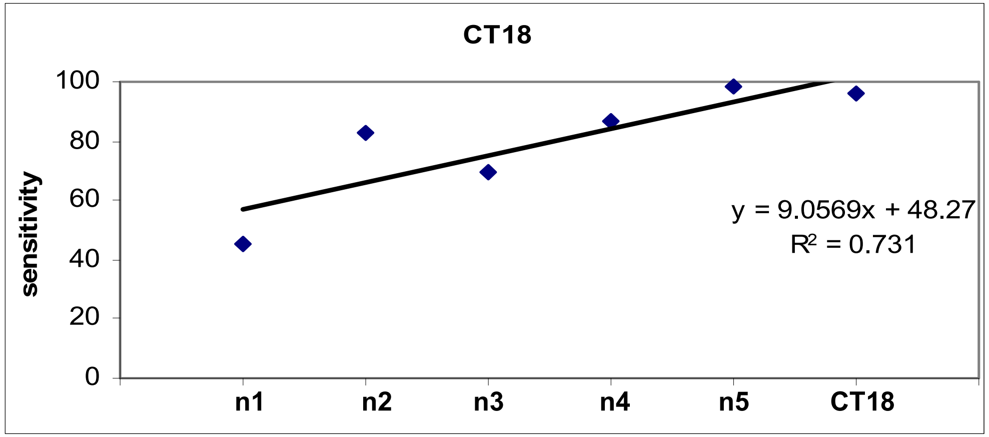


Pseudogenes

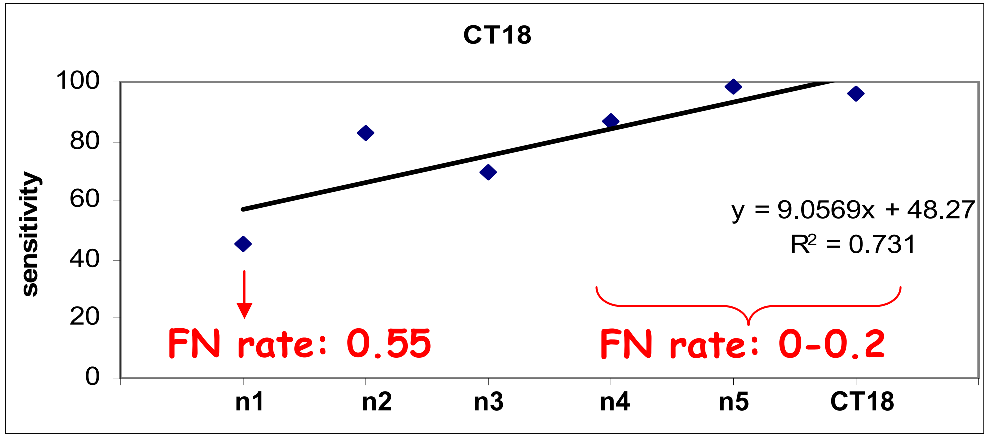
SPIs: Relative Time Distribution



Sensitivity vs Time



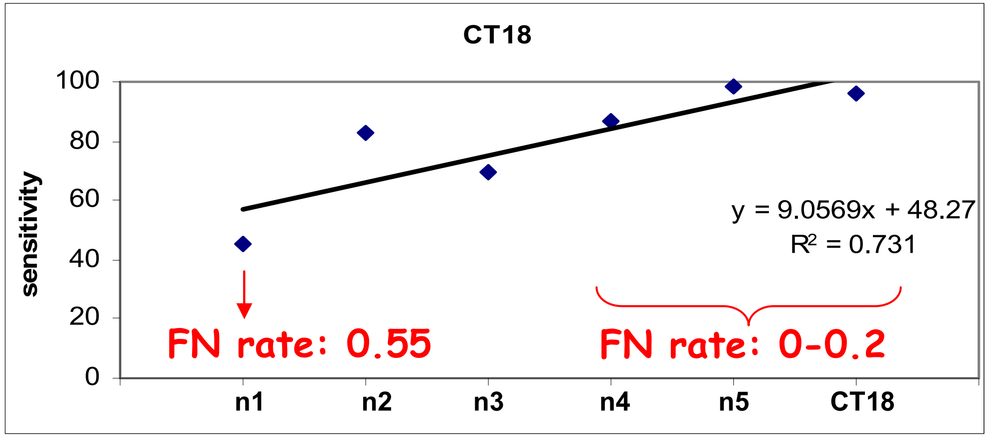
Sensitivity vs Time

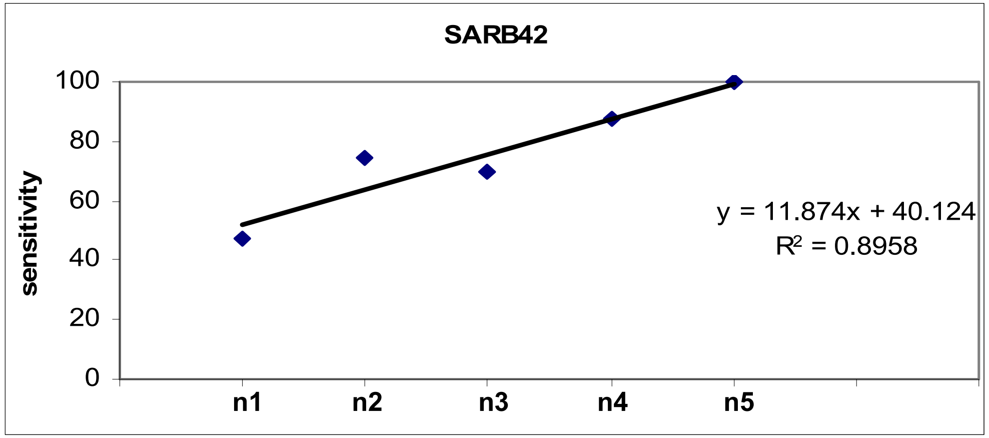


FN rate: 0.55

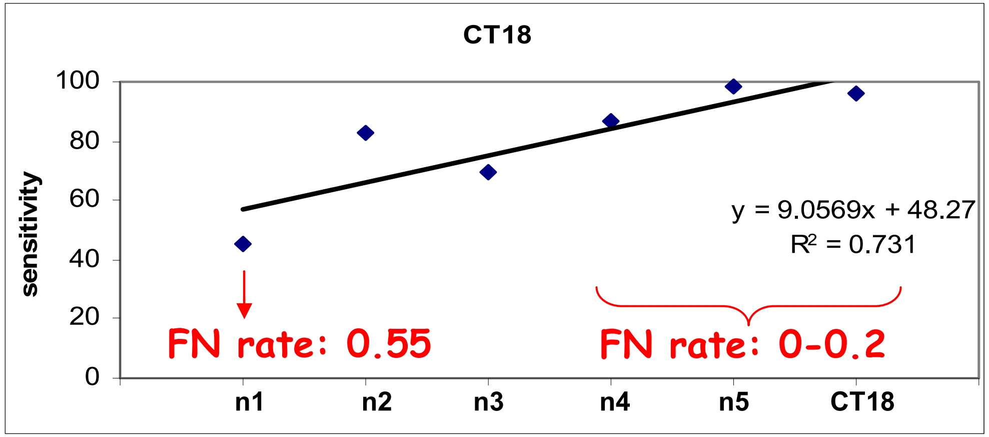
FN rate: 0-0.2

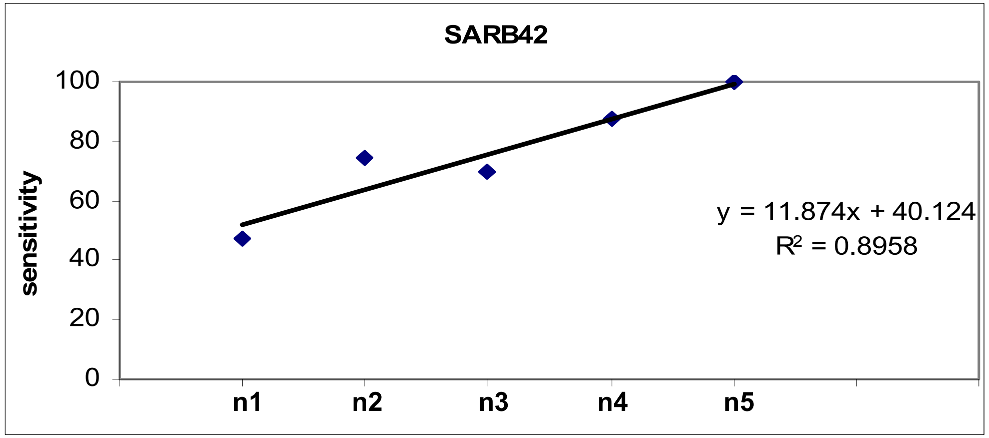
Sensitivity vs Time

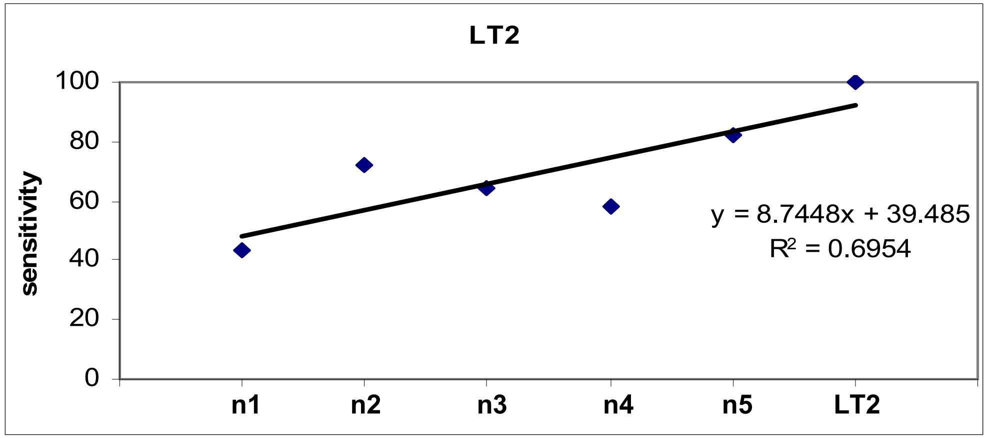




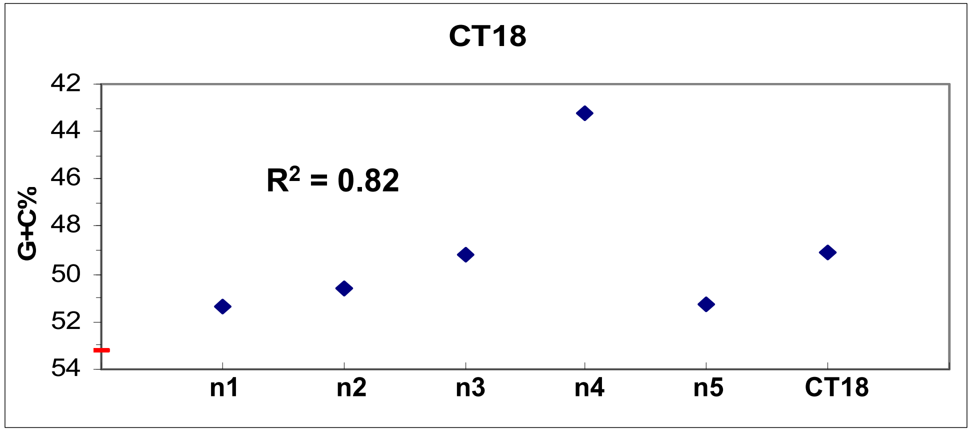
Sensitivity vs Time

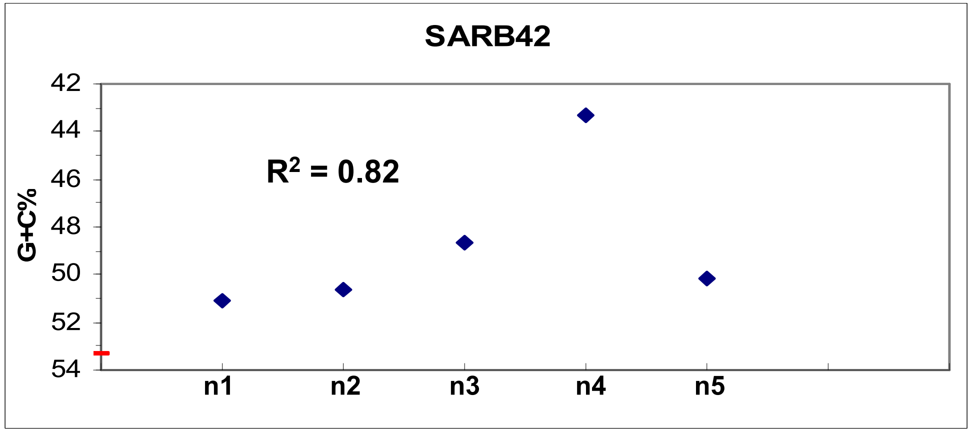


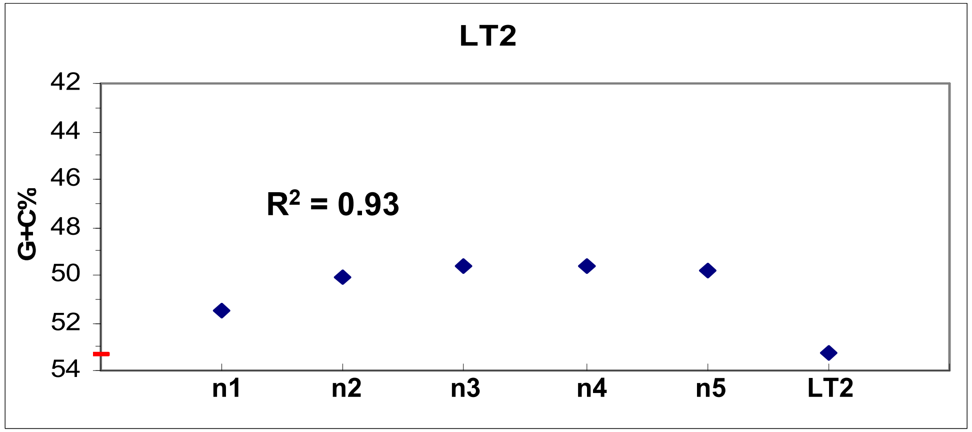




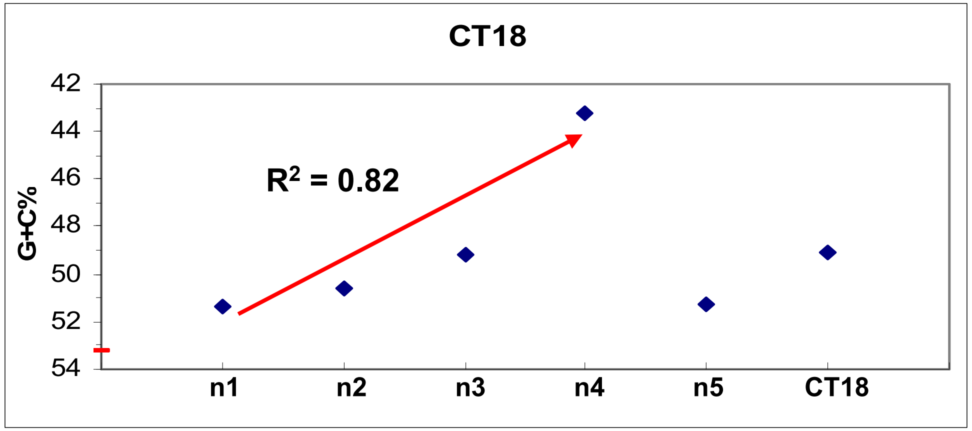
G+C vs Time

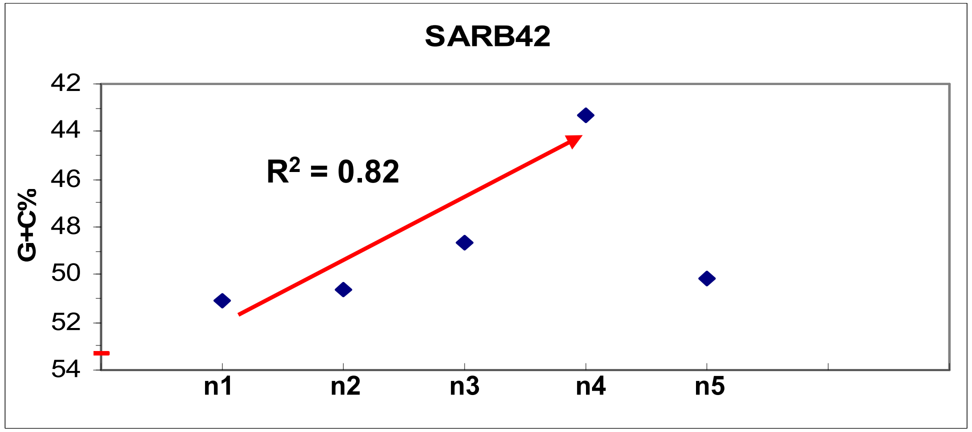

CT18

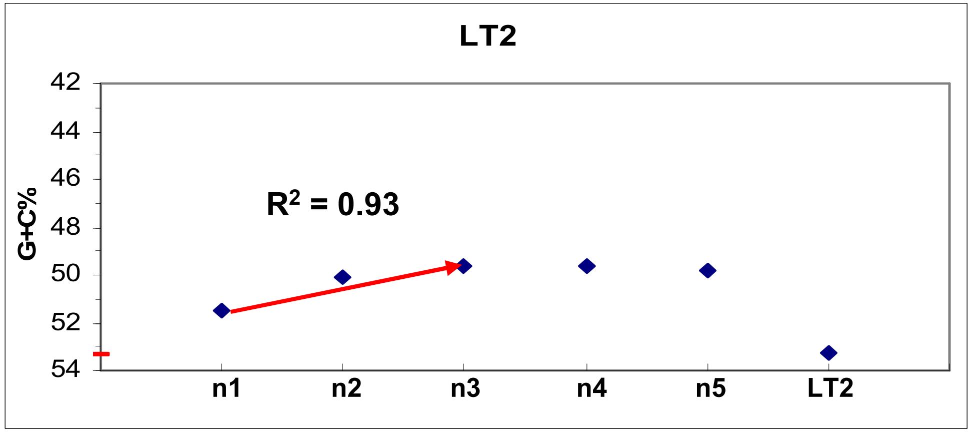

SARB42


LT2

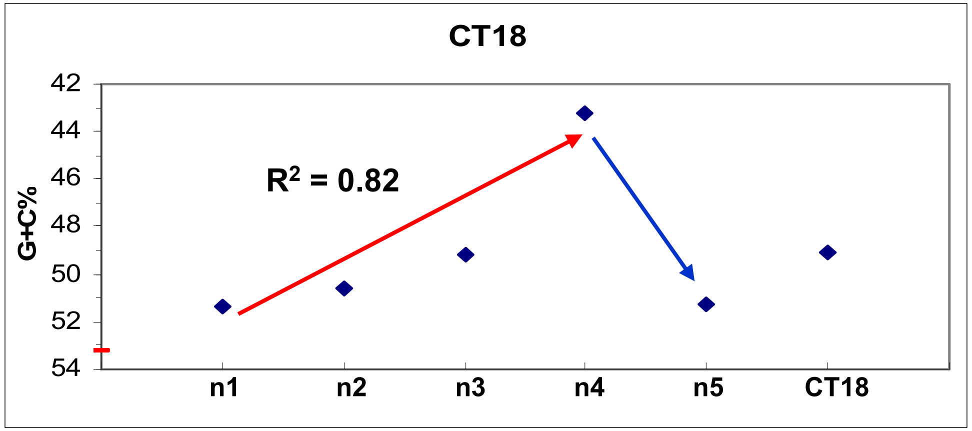
G+C vs Time

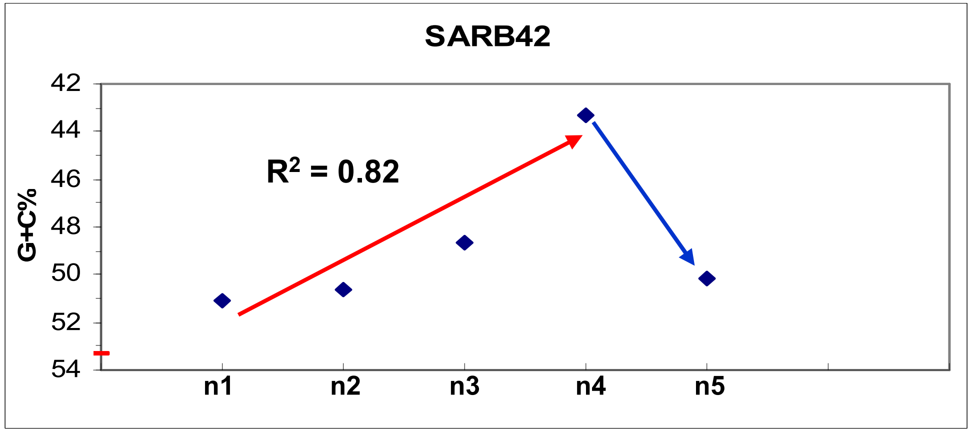

CT18
G+C%
42
44
46
48
50
52
54
R² = 0.82
n1
n2
n3
n4
n5
CT18

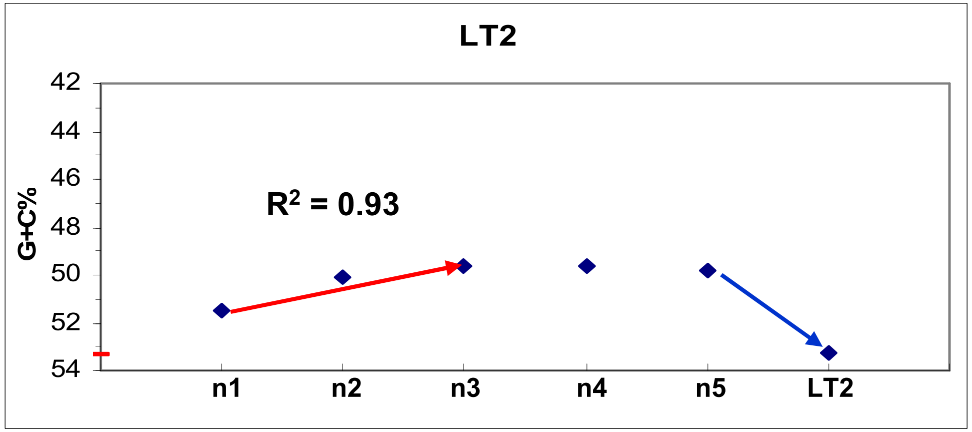

SARB42
G+C%
42
44
46
48
50
52
54
R² = 0.82
n1
n2
n3
n4
n5


LT2
G+C%
42
44
46
48
50
52
54
R² = 0.93
n1
n2
n3
n4
n5
LT2

G+C vs Time


CT18
G+C%
42
44
46
48
50
52
54
R2 = 0.82
n1
n2
n3
n4
n5
CT18


SARB42
G+C%
42
44
46
48
50
52
54
R2 = 0.82
n1
n2
n3
n4
n5


LT2
G+C%
42
44
46
48
50
52
54
R2 = 0.93
n1
n2
n3
n4
n5
LT2