Phylogenetic reconstruction: criteria

Bastien Boussau
Bastien.boussau@univ-lyon1.fr
@bastounette

Phylogenetic inference

How to find the best tree given my data?

- **Need for a criterion/score**

- **Need for an algorithm to find/construct the tree**

Phylogenetic inference

How to find the best tree given my data?

- **Need for a criterion/score**
  - Maximum Parsimony
  - Minimum Evolution or least squares (distance methods)
  - Maximum Likelihood ~ P(D|M)
  - Posterior Probability P(M|D)
- **Need for an algorithm to find/construct the tree**

Phylogenetic inference

How to find the best tree given my data?

- **Need for a criterion/score**
  - Maximum Parsimony
  - Minimum Evolution or least squares (distance methods)
  - Maximum Likelihood ~ P(D|M)
  - Posterior Probability P(M|D)
- **Need for an algorithm to find/construct the tree**

Phylogenetic inference

How to find the best tree given my data?

- **Need for a criterion/score**
  - Maximum Parsimony
  - (M) Minimum Evolution or least squares (distance methods)
  - (M) Maximum Likelihood ~ P(D|M)
  - (M) Posterior Probability P(M|D)
- **Need for an algorithm to find/construct the tree**
  - e.g.: try several topologies, (choose some branch lengths,) score the topologies, choose the one that has the best score

Plan: Criteria for evaluating phylogenies

- Criteria for evaluating phylogenetic trees:

  - Parsimony

  - Distance methods

  - Maximum Likelihood

    - Using a Monte Carlo simulation

    - Using Felsenstein's pruning algorithm

  - Posterior probability (Bayesian approach)

- *Conventions:*

  - We're dealing with aligned sequence data

  - gaps are not taken into account

Parsimony

- "The principle that the most acceptable explanation of an occurrence, phenomenon, or event is the simplest, involving the fewest entities, assumptions, or changes. In phylogenetics, for example, the preferred tree showing evolutionary relationships between species, molecules, or other entities is the one that requires the least amount of evolutionary change, that is, maximum parsimony."

Maximum parsimony

- Has been advocated strongly by some against model-based approaches: many controversies (see "The Troubled Growth of Statistical Phylogenetics", Felsenstein 2001)

- Edwards and Cavalli-Sforza (1963): the preferred evolutionary tree involves "the minimum net amount of evolution" = *Maximum parsimony tree*

- → For sequence data: find the phylogeny that involves the minimum number of substitutions

    → We need a way to count the minimum number of substitutions on a phylogeny = compute the **parsimony score** of a phylogenetic tree
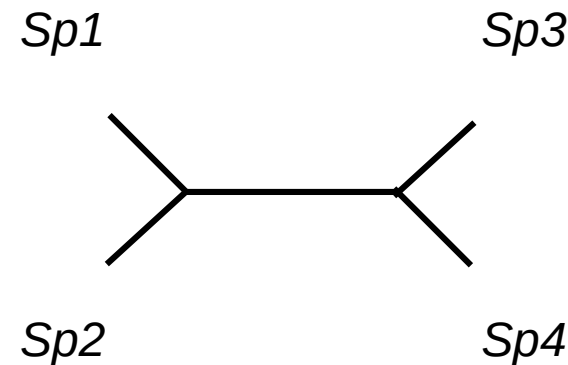
Computing the parsimony score

Sp1 ATGCGCT...

Sp2 AGTCGCA...

Sp3 AGGTGCA...

Sp4 ATGCCCT...

Sp1   Sp3
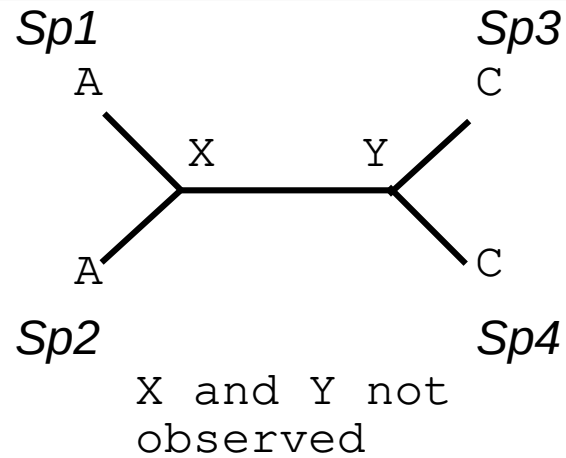
Sp2   Sp4

Parsimony score of a tree given an alignment: sum of the parsimony scores for each site
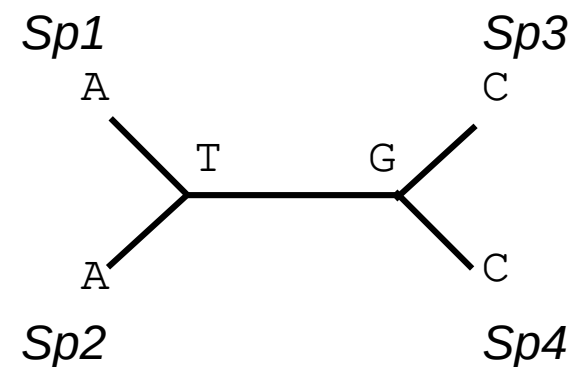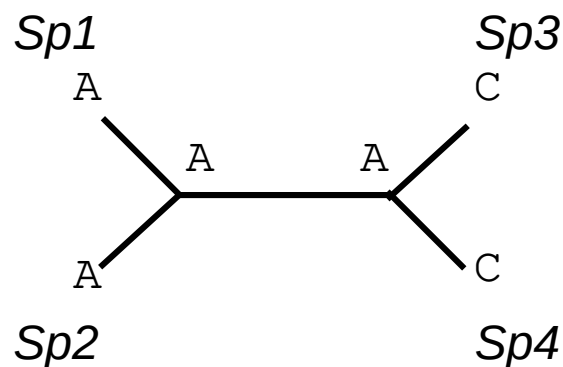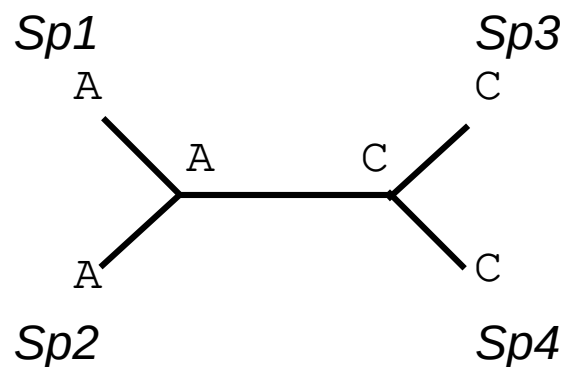
→ We assume that all sites are independent

Computing the parsimony score at one site

Species tree S
Site i

Sp1
A

Sp3
C

X   Y

A

C

Sp2

Sp4
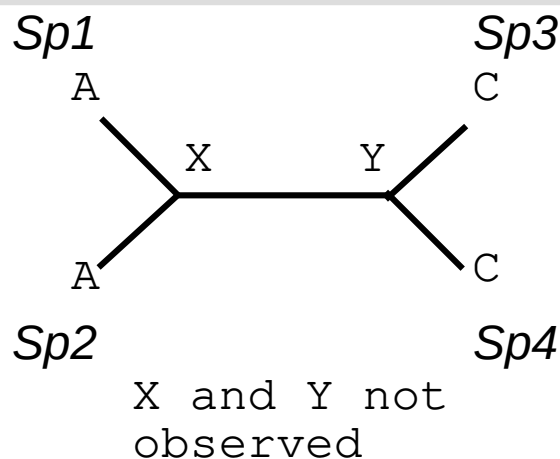
X and Y not
observed

Computing the parsimony score at one site

Species tree S
Site i

Sp1
A

Sp3
C

X

Y

A

C

Sp2

Sp4

X and Y not
observed

Sp1
A

Sp3
C

A

C

A

C

Sp2

Sp4

Sp1
A

Sp3
C

A

A

A

C

Sp2

Sp4

Sp1
A

Sp3
C

T

G

A

C
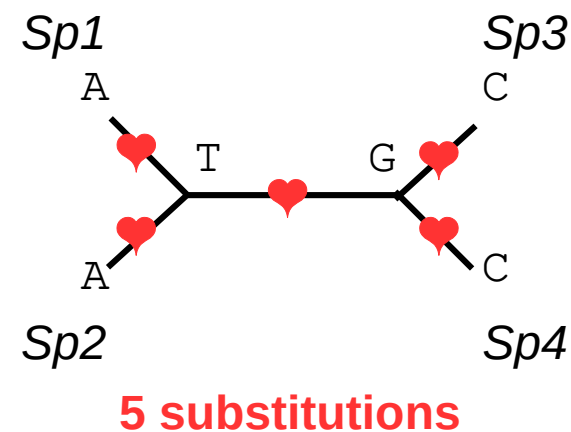
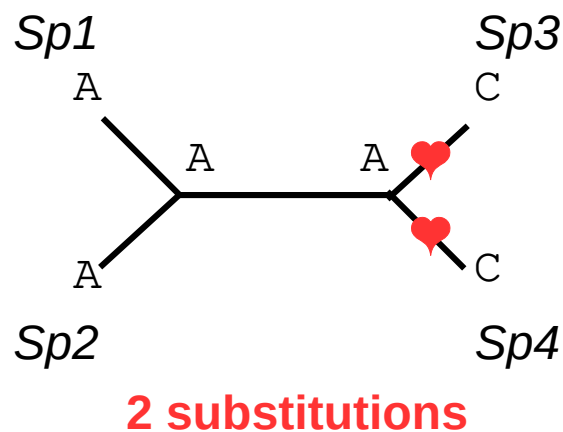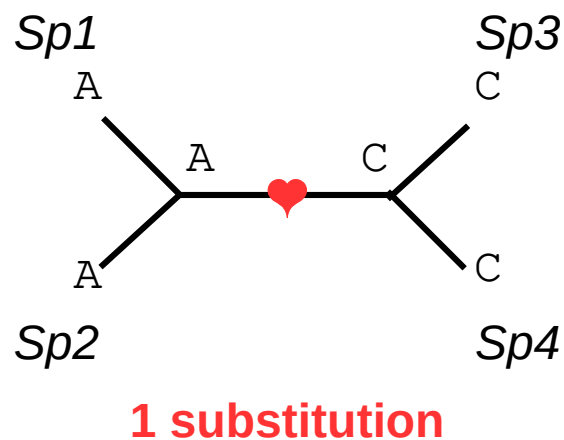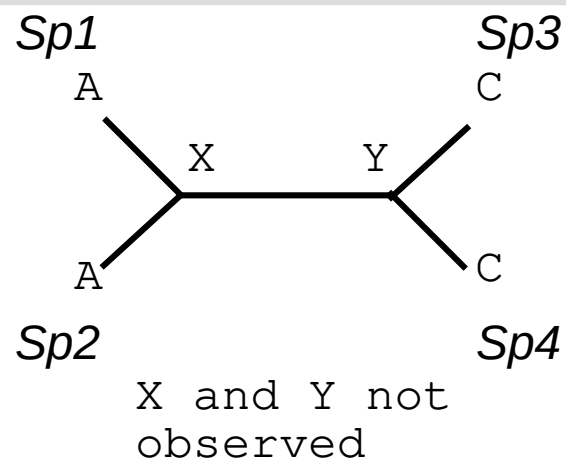Sp2

Sp4

Computing the parsimony score at one site

Computing the parsimony score at one site

Species tree S
Site i
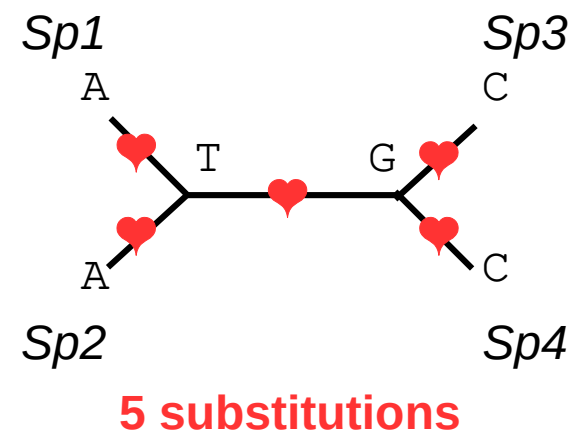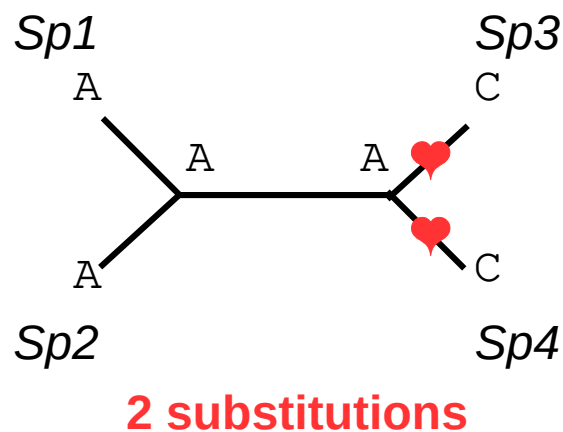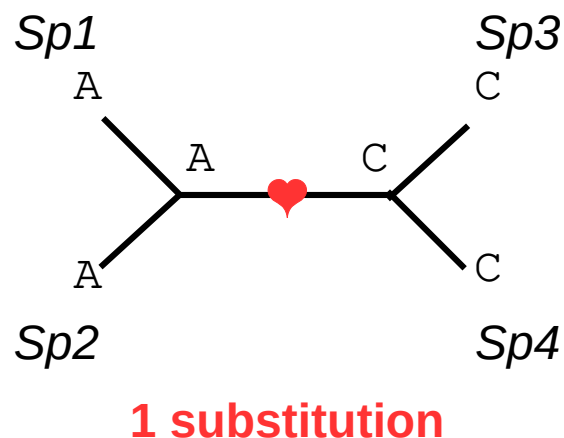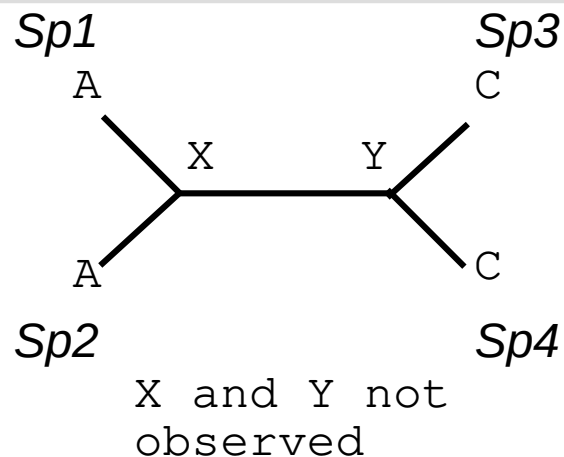

Sp1
A

Sp3
C

X   Y

A
Sp2

C
Sp4

X and Y not
observed

Sp1
A

Sp3
C

A   C

A
Sp2

C
Sp4

1 substitution

Sp1
A

Sp3
C

A   A

A
Sp2

C
Sp4

2 substitutions

Sp1
A

Sp3
C

T   G

A
Sp2

C
Sp4

5 substitutions

Parsimony score of species tree S for site i: 1

Computing the parsimony score of a tree at one site

- Naive brute force approach: test all possible assignments on internal nodes
  - N internal nodes, 4 possibilities {A,C,G,T} per node
  - → 4^N possibilities to try

- Fitch's algorithm (1971):
  - Arbitrarily root the tree
  - Compute, from the tips up, two elements per node:
    - P: The score of the underlying subtree
    - X: The set of states possible at that node, given the score P
  - Complexity: O(4N)

Computing the parsimony score of a tree at one site

- Naive brute force approach: test all possible assignments on internal nodes
  - N internal nodes, 4 possibilities {A,C,G,T} per node
  - → 4^N possibilities to try | **For 2 internal nodes: $4^2 = 16$** |

- Fitch's algorithm (1971):
  - Arbitrarily root the tree
  - Compute, from the tips up, two elements per node:
    - P: The score of the underlying subtree
    - X: The set of states possible at that node, given the score P
  - Complexity: O(4N)

Computing the parsimony score of a tree at one site

- Naive brute force approach: test all possible assignments on internal nodes
  - N internal nodes, 4 possibilities {A,C,G,T} per node
  - → 4^N possibilities to try | For 2 internal nodes: $4^2 = 16$ |

- Fitch's algorithm (1971):
  - Arbitrarily root the tree
  - Compute, from the tips up, two elements per node:
    - P: The score of the underlying subtree
    - X: The set of states possible at that node, given the score P
  - Complexity: O(4N) | For 2 internal nodes: 4*2 =8 |
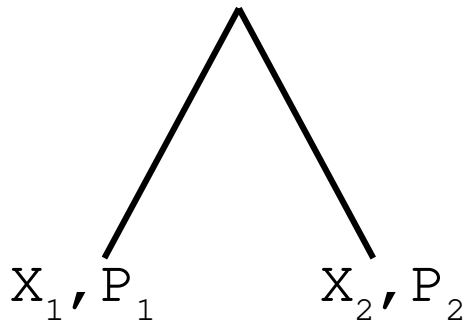
Fitch's algorithm

- P: The score of the underlying subtree

- X: The set of states possible at that node, given the score P

Climbing up the tree: computing P and X for a node given its children

1st case: $X_1 \cap X_2$ not empty

X_1, P_1 X_2, P_2

. . . .

2nd case: $X_1 \cap X_2$ empty

X_1, P_1 X_2, P_2
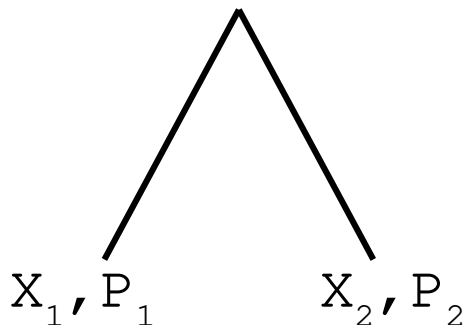
Fitch's algorithm

- P: The score of the underlying subtree

- X: The set of states possible at that node, given the score P

Climbing up the tree: computing P and X for a node given its children

1st case: $X_1 \cap X_2$ not empty

$X_1 \cap X_2$, $P_1 + P_2$

X_1, P_1 X_2, P_2

$\{A\}, 3$

$\{A, T\}, 1$ $\{A, C\}, 2$

. . . .

2nd case: $X_1 \cap X_2$ empty
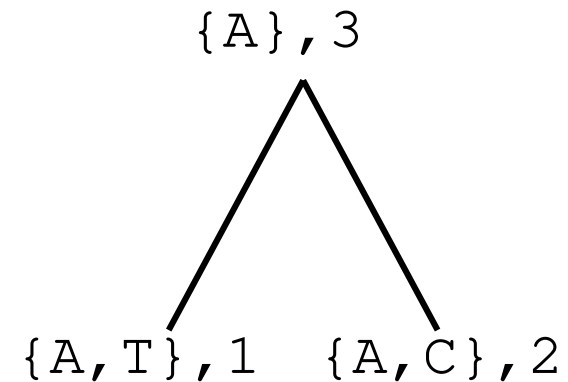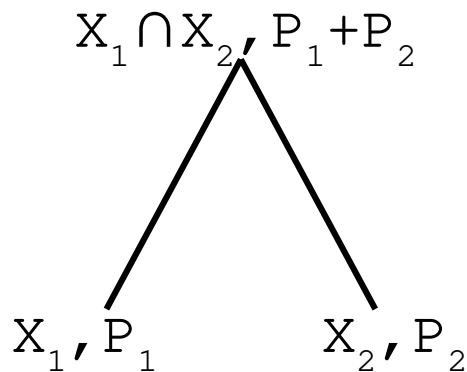
X_1, P_1 X_2, P_2

Fitch's algorithm

- P: The score of the underlying subtree

- X: The set of states possible at that node, given the score P

Climbing up the tree: computing P and X for a node given its children

1st case: $X_1 \cap X_2$ not empty

$X_1 \cap X_2, P_1 + P_2$

X_1, P_1 X_2, P_2

$\{A\}, 3$

$\{A,T\}, 1$ $\{A,C\}, 2$

. . . .

2nd case: $X_1 \cap X_2$ empty

$X_1 \cup X_2, P_1 + P_2 + 1$

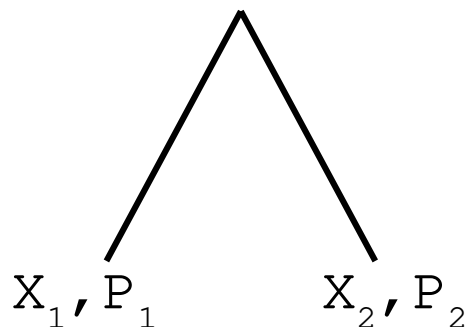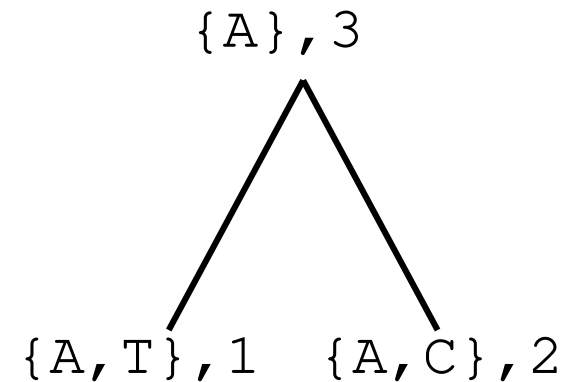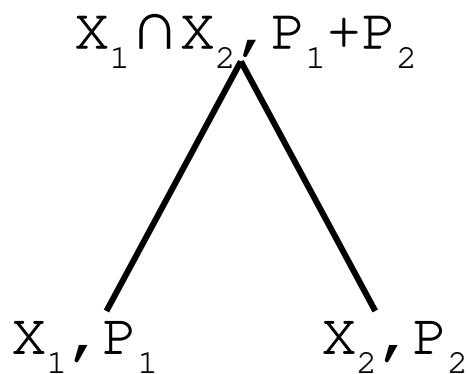X_1, P_1 X_2, P_2
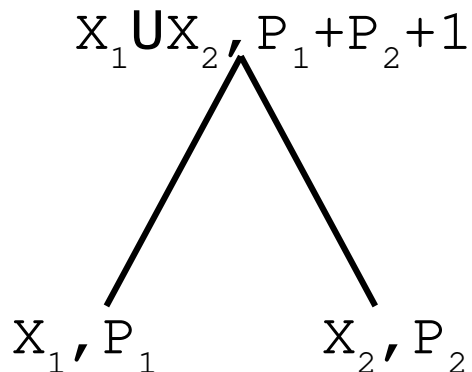
Fitch's algorithm

- P: The score of the underlying subtree

- X: The set of states possible at that node, given the score P

Climbing up the tree: computing P and X for a node given its children

1ˢᵗ case: $X_1 \cap X_2$ not empty

$X_1 \cap X_2, P_1 + P_2$

X_1, P_1 X_2, P_2

$\{A\}, 3$

$\{A, T\}, 1$ $\{A, C\}, 2$

. . . .

2ⁿᵈ case: $X_1 \cap X_2$ empty

$X_1 \cup X_2, P_1 + P_2 + 1$

X_1, P_1 X_2, P_2

$\{A, C, G, T\}, 4$

$\{A, G\}, 4$ $\{C, T\}, 4$

$\{C, T\}, 1$ $\{A, G\}, 2$ $\{C, T\}, 1$ $\{A, G\}, 2$

Example

C   T   G   T   A   A

Example

{C},0   {T},0 {G},0   {T},0   {A},0   {A},0

Example

{C,T},1

{C},0   {T},0 {G},0   {T},0   {A},0   {A},0

Example

Example

Example

Example

{A,T},4

{T},3

{A,G,T},2

{C,T},1

{G,T},1

{C},0   {T},0 {G},0   {T},0   {A},0   {A},0

Several possible scenarios

Conclusion on maximum parsimony

- Involves looking for the tree with the least amount of chang

- Computing the parsimony score of a tree can be done with Fitch's algorithm

- Searching through topology space for the most parsimonious tree can be done with algorithms similar to those used for Maximum Likelihood reconstruction (see Alexis's talk)

Plan: Criteria for evaluating phylogenies

- Criteria for evaluating phylogenetic trees:

  - Parsimony

  - Distance methods

  - Maximum Likelihood

    - Using a Monte Carlo simulation

    - Using Felsenstein's pruning algorithm

  - Posterior probability (Bayesian approach)

- *Conventions:*

  - We're dealing with aligned sequence data

  - gaps are not taken into account

Distance methods

- Distance-based approaches:

  - least squares methods,

  - Minimum evolution method,

  - Neighbor Joining.

Minimum Evolution or least squares: distance methods

- Uses a distance matrix:

Sp1 ATGCGCT...

Sp2 AGTCGCA...

Sp3 AGGTGCA...

Sp4 ATGCCCT...

Minimum Evolution or least squares: distance methods

- Uses a distance matrix:

Sp1 ATGCGCT...

Sp2 AGTCGCA...

Sp3 AGGTGCA...

Sp4 ATGCCCT...

1 →

	Sp1	Sp2	Sp3	Sp4
Sp1	0	0.1	0.2	0.15
Sp2	0.1	0	0.3	0.01
Sp3	0.2	0.3	0	0.6
Sp4	0.15	0.01	0.6	0

Minimum Evolution or least squares: distance methods

- ## Uses a distance matrix:

Minimum Evolution or least squares: distance methods

- Uses a distance matrix:

$Sp1$ ATGCGCT...

$Sp2$ AGTCGCA...

$Sp3$ AGGTGCA...

$Sp4$ ATGCCCT...

①
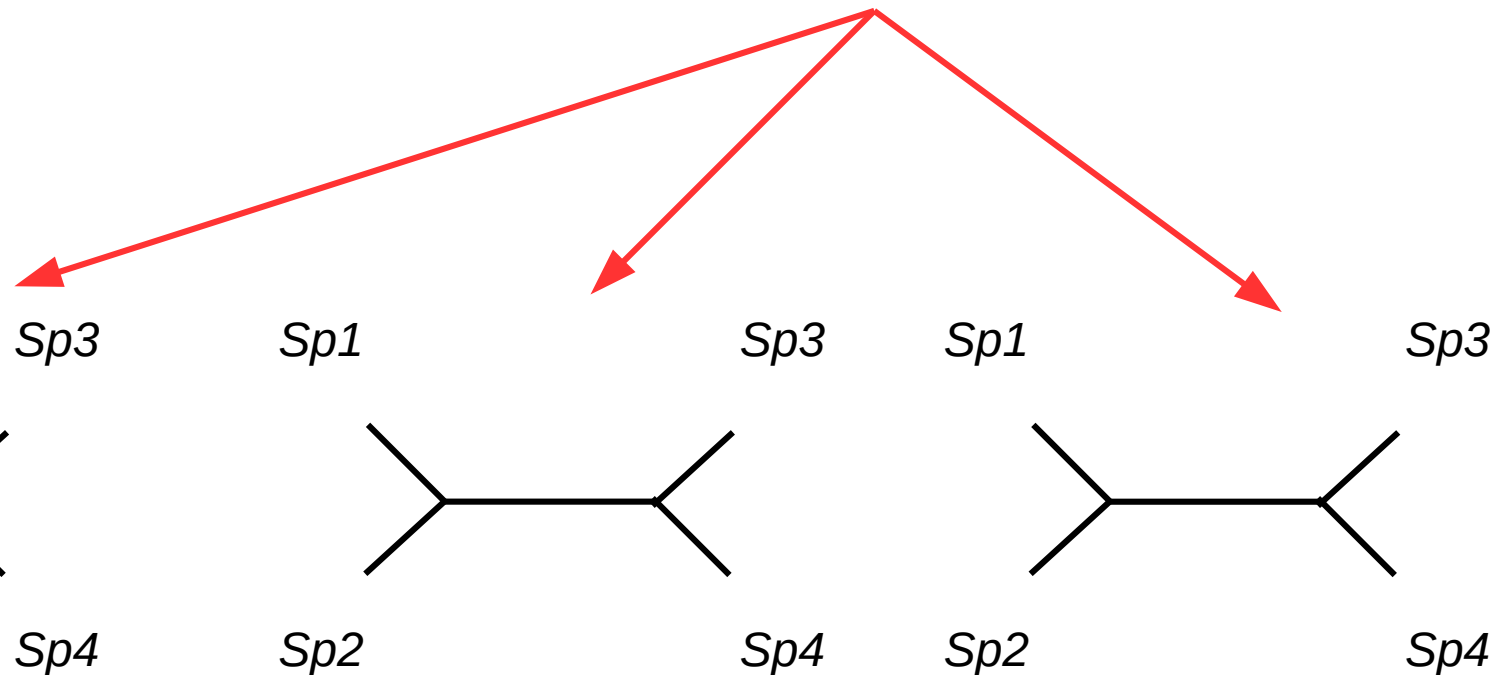
How to compute the distance matrix?

	Sp1	Sp2	Sp3	Sp4
Sp1	0	0.1	0.2	0.15
Sp2	0.1	0	0.3	0.01
Sp3	0.2	0.3	0	0.6
Sp4	0.15	0.01	0.6	0

②

Sp1   *Sp3*   *Sp1*   *Sp3*   *Sp1*   *Sp3*
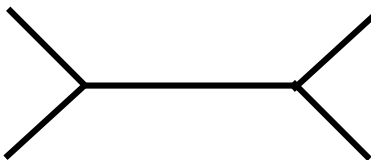
Sp2   *Sp4*   *Sp2*   *Sp4*   *Sp2*   *Sp4*
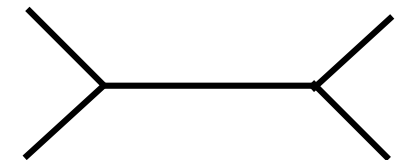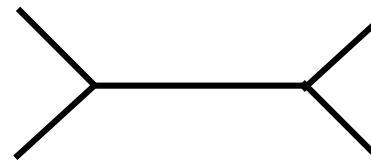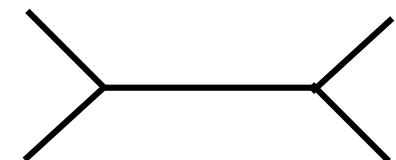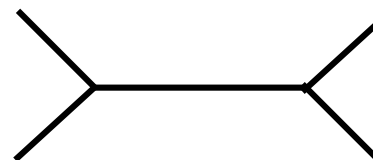
Minimum Evolution or least squares: distance methods

- ## Uses a distance matrix:

Sp1 ATGCGCT...

Sp2 AGTCGCA...

Sp3 AGGTGCA...

Sp4 ATGCCCT...

①

How to compute the distance matrix?

	Sp1	Sp2	Sp3	Sp4
Sp1	0	0.1	0.2	0.15
Sp2	0.1	0	0.3	0.01
Sp3	0.2	0.3	0	0.6
Sp4	0.15	0.01	0.6	0

② ***Which tree fits the distance matrix best?***

1: How to compute distances between sequences?

- Simply count differences (observed divergence)

Sp1 ATGCGCT
Sp2 AGTCGCA ⟶ d(Sp1-Sp2) =

1: How to compute distances between sequences?

- Simply count differences (observed divergence)

Sp1 ATGCGCT
Sp2 AGTCGCA ⟶ d(Sp1-Sp2) =

1: How to compute distances between sequences?

- Simply count differences (observed divergence)

Sp1 ATGCGCT
Sp2 AGTCGCA $\longrightarrow$ d(Sp1-Sp2) = 3/7~0.43

1: How to compute distances between sequences?

- Simply count differences (observed divergence)

$Sp1$ ATGCGCT
$Sp2$ AGTCGCA $\longrightarrow$ d(Sp1-Sp2) = 3/7~0.43
-- -

- Use a model of sequence evolution

  → *cf. talk on models*

  – Advantages:
  - *Hidden substitutions are taken into account*
  - *Parameters of the model of substitution can be estimated in the Maximum Likelihood framework*

2: How to compute the fit between a distance matrix and a tree?

A tree implies distances between tips: compare those **patristic** distances to sequence-based distances

Sp1   Sp3

0.2
      0.35   0.1
0.1   0.15

Sp2   Sp4

2: How to compute the fit between a distance matrix and a tree?
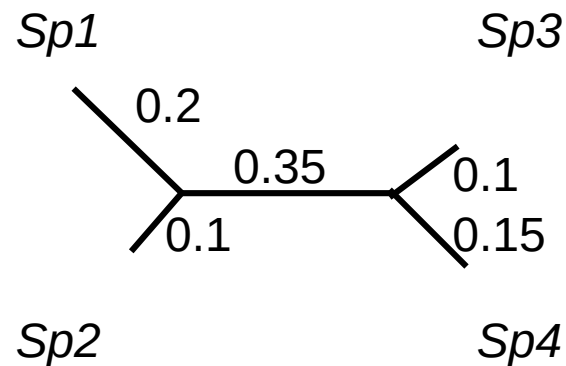
A tree implies distances between tips: compare those **patristic** distances to sequence-based distances
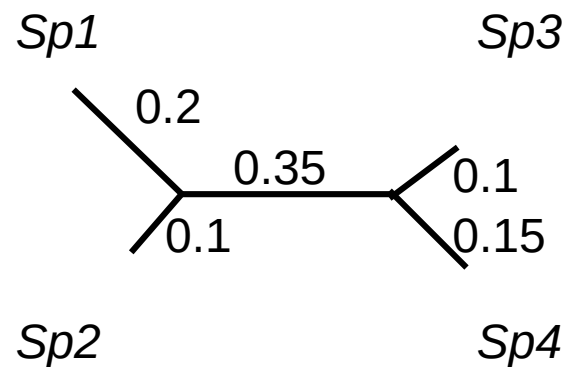
	Sp1	Sp2	Sp3	Sp4
Sp1	0	0.3	0.65	0.7
Sp2	0.3	0	0.55	0.6
Sp3	0.65	0.55	0	0.25
Sp4	0.7	0.6	0.25	0

2: How to compute the fit between a distance matrix and a tree?

A tree implies distances between tips: compare those **patristic** distances to sequence-based distances
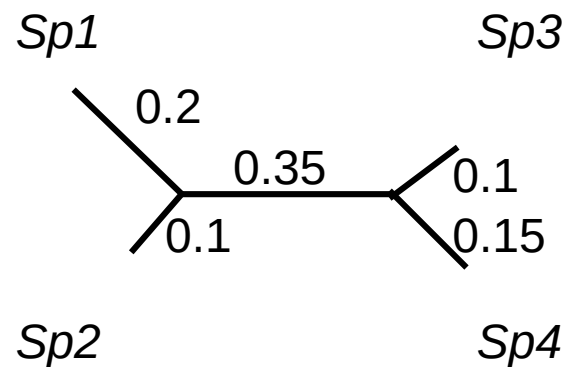
	Sp1	Sp2	Sp3	Sp4
Sp1	0	0.3	0.65	0.7
Sp2	0.3	0	0.55	0.6
Sp3	0.65	0.55	0	0.25
Sp4	0.7	0.6	0.25	0

	Sp1	Sp2	Sp3	Sp4
Sp1	0	0.1	0.2	0.15
Sp2	0.1	0	0.3	0.01
Sp3	0.2	0.3	0	0.6
Sp4	0.15	0.01	0.6	0

2: How to compute the fit between a distance matrix and a tree?

A tree implies distances between tips: compare those **patristic** distances to sequence-based distances

	Sp1	Sp2	Sp3	Sp4
Sp1	0	0.3	0.65	0.7
Sp2	0.3	0	0.55	0.6
Sp3	0.65	0.55	0	0.25
Sp4	0.7	0.6	0.25	0

$\text{score}_{\text{ULS}} = (0-0)^2+(0.3-0.1)^2+(0.65-0.2)^2+\ldots$

With ULS: Unweighted Least Squares
(other criteria have been proposed)

	Sp1	Sp2	Sp3	Sp4
Sp1	0	0.1	0.2	0.15
Sp2	0.1	0	0.3	0.01
Sp3	0.2	0.3	0	0.6
Sp4	0.15	0.01	0.6	0

Computing the optimal distances on a given topology

Using the ULS criterion, we can compute the fit between a sequence-based distance matrix and any tree (topology + branch lengths), thanks to the patristic matrix trick.

Computing the optimal distances on a given topology

Using the ULS criterion, we can compute the fit between a sequence-based distance matrix and any tree (topology + branch lengths), thanks to the patristic matrix trick.

But how can we pick branch lengths on the topology?

Computing the optimal distances on a given topology

Using the ULS criterion, we can compute the fit between a sequence-based distance matrix and any tree (topology + branch lengths), thanks to the patristic matrix trick.

But how can we pick branch lengths on the topology?

ULS provides a mathematical way to find the optimal branch lengths on a given topology! This involves some simple matrix algebra (solving a set of linear equations).

Searching for the best tree using Unweighted Least Squares

- We now know how to compute the ULS score of a tree topology. It involves:

    – Matrix algebra to find the best branch lengths

    – Computing the $score_{ULS}$ for that tree

- Given a set of tree topologies, we can compute the "best" tree topology according to the ULS criterion: it is the one with the lowest $score_{ULS}$

- How to obtain a set of tree topologies to score is tackled later in the course *(see Alexis's talk)*

Searching for the best tree using Unweighted Least Squares

- We now know how to compute the ULS score of a tree topology. It involves:
    - Matrix algebra to find the best branch lengths
    - Computing the $score_{ULS}$ for that tree

- Given a set of tree topologies, we can compute the "best" tree topology according to the ULS criterion: it is the one with the lowest $score_{ULS}$

- How to obtain a set of tree topologies to score is tackled later in the course *(see Alexis's talk)*

Minimum evolution criterion

- Motivation similar to parsimony
- **Hypothesis:** the true tree should be the shortest tree
- → Idea:
  - Given a matrix of pairwise distances and a set of tree topologies to evaluate
  - Match pairwise distances onto each tree topology
  - Sum the branch lengths on each tree
  - *Your best estimate is the tree with the smallest sum of branch lengths*

Minimum evolution criterion

- Motivation similar to parsimony

- **<u>Hypothesis</u>**: the true tree should be the shortest tree

- → Idea:
  - Given a matrix of pairwise distances and a set of tree topologies to evaluate
  - Match pairwise distances onto each tree topology: *Use least-squares fitting!*
  - Sum the branch lengths on each tree
  - ***Your best estimate is the tree with the smallest sum of branch lengths***

Minimum Evolution or least squares: distance methods

- ## Uses a distance matrix:

Sp1 ATGCGCT...

Sp2 AGTCGCA...

Sp3 AGGTGCA...

Sp4 ATGCCCT...

①

How to compute the distance matrix?

	Sp1	Sp2	Sp3	Sp4
Sp1	0	0.1	0.2	0.15
Sp2	0.1	0	0.3	0.01
Sp3	0.2	0.3	0	0.6
Sp4	0.15	0.01	0.6	0

ME: involves the patristic matrix, matrix algebra and summing branch lengths

②
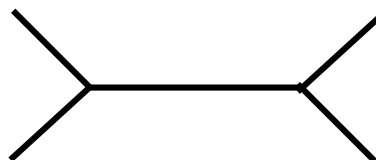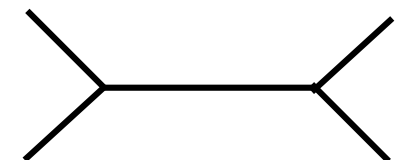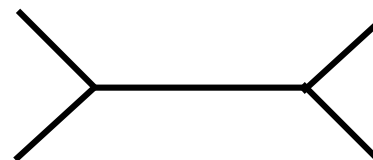
Minimum evolution criterion

- To obtain a Minimum Evolution tree, at some point we have to use Least Squares estimation to assign branch lengths to a tree topology

    $\rightarrow$ hybrid approach where two different criteria are mixed up

- However, Minimum evolution works pretty well in practice

- Neighbor-Joining (Saitou and Nei, 1987) is a famous heuristic algorithm for finding the Minimum Evolution tree (not seen in our course, but has been very widely used); see Gascuel and Steel, 2006 for a clear explanation

Summary on distance methods

- Distance methods are the fastest phylogenetic methods available, notably thanks to Neighbor Joining and other algorithms to build a tree given a distance matrix (e.g. BioNJ, Weighbor, FastME…)

- Can be based on models of sequence evolution to compute pairwise distances

- Better than Maximum Parsimony when sequences are divergent, but less accurate than Maximum Likelihood or Bayesian Inference

- The main reason is that distance methods do not use the entire data matrix together, but look at it pair of sequences by pair of sequences

Plan: Criteria for evaluating phylogenies

- Criteria for evaluating phylogenetic trees:

  - Parsimony

  - Distance methods

  - Maximum Likelihood

    - Using a Monte Carlo simulation

    - Using Felsenstein's pruning algorithm

  - Posterior probability (Bayesian approach)

- *Conventions:*

  - We're dealing with aligned sequence data

  - gaps are not taken into account

An Introduction to the Likelihood Function
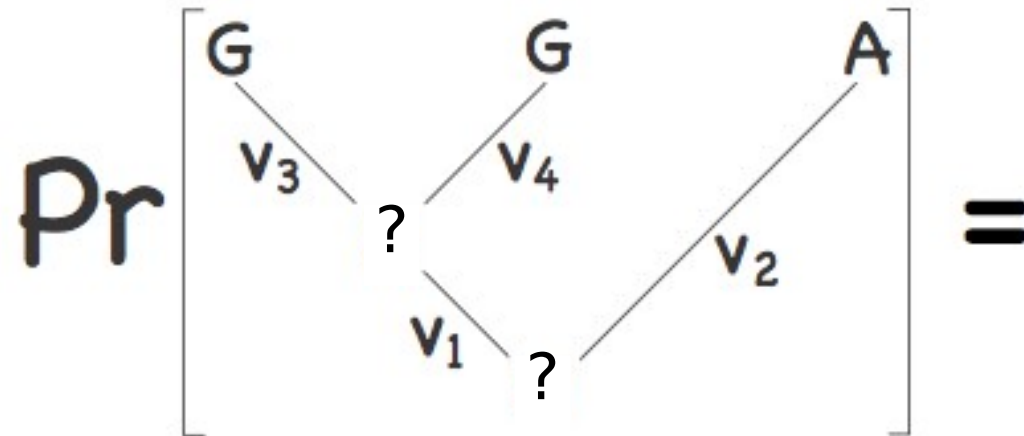
$$\pi_i \times p_{ij}(v_1) \times p_{iA}(v_2) \times p_{jG}(v_3) \times p_{jG}(v_4)$$

π_i Stationary frequencies

$p_{ij}(v)$ Transition probabilities

An Introduction to the Likelihood Function

What is likelihood?

The likelihood is a quantity that is proportional to the probability of observing/realizing the data under a fully specified model/hypothesis.

An Introduction to the Likelihood Function

What is likelihood?

The likelihood is a quantity that is proportional to the probability of observing/realizing the data under a fully specified model/hypothesis.

$$\text{Likelihood}(\text{Parameter}) = \text{Constant} \times \text{Probability}(\text{Data}|\text{Parameter})$$

An Introduction to the Likelihood Function

What is likelihood?

The likelihood is a quantity that is proportional to the probability of observing/realizing the data under a fully specified model/hypothesis.

$$\text{Likelihood}(\text{Parameter}) \propto \text{Probability}(\text{Data}|\text{Parameter})$$

An Introduction to the Likelihood Function

What is likelihood?

The likelihood is a quantity that is proportional to the probability of observing/realizing the data, $\mathbf{X}$, under a fully specified model/hypothesis.

$$\text{Likelihood}(\text{Parameter}) \propto \text{Probability}(\mathbf{X}|\text{Parameter})$$

An Introduction to the Likelihood Function

What is likelihood?

The likelihood is a quantity that is proportional to the probability of observing/realizing the data, $\mathbf{X}$, under a fully specified model/hypothesis, Θ.

$$\text{Likelihood}(\Theta) \propto \text{Probability}(\mathbf{X}|\Theta)$$

An Introduction to the Likelihood Function

What is likelihood?

The likelihood is a quantity that is proportional to the probability of observing/realizing the data, $\mathbf{X}$, under a fully specified model/hypothesis, Θ.

$$L(\Theta) \propto f(\mathbf{X}|\Theta)$$

An Introduction to the Likelihood Function

What is likelihood?

The likelihood is a quantity that is proportional to the probability of observing/realizing the data, $\mathbf{X}$, under a fully specified model/hypothesis, Θ .

$$L(\Theta) \propto f(\mathbf{X}|\Theta)$$

For phylogenetic problems, the likelihood is proportional to the probability of observing the sequence alignment, $\mathbf{X}$, under a fully specified phylogenetic model.

$$L(\tau, \nu, \Phi) \propto f(\mathbf{X} \mid \tau, \nu, \Phi)$$

An Introduction to the Likelihood Function

What is likelihood?

The likelihood is a quantity that is proportional to the probability of observing/realizing the data, $\mathbf{X}$, under a fully specified model/hypothesis, Θ .

$$L(\Theta) \propto f(\mathbf{X}|\Theta)$$

For phylogenetic problems, the likelihood is proportional to the probability of observing the sequence alignment, $\mathbf{X}$, under a fully specified phylogenetic model.

$$L(\tau, \nu, \Phi) \propto f(\mathbf{X} \mid \tau, \nu, \Phi)$$

tree topology

An Introduction to the Likelihood Function

What is likelihood?

The likelihood is a quantity that is proportional to the probability of observing/realizing the data, $\mathbf{X}$, under a fully specified model/hypothesis, Θ.

$$L(\Theta) \propto f(\mathbf{X}|\Theta)$$

For phylogenetic problems, the likelihood is proportional to the probability of observing the sequence alignment, $\mathbf{X}$, under a fully specified phylogenetic model.

$$L(\tau, \nu, \Phi) \propto f(\mathbf{X} \mid \tau, \nu, \Phi)$$

tree topology

branch lengths

An Introduction to the Likelihood Function

What is likelihood?

The likelihood is a quantity that is proportional to the probability of observing/realizing the data, $\mathbf{X}$, under a fully specified model/hypothesis, Θ.

$$L(\Theta) \propto f(\mathbf{X}|\Theta)$$

For phylogenetic problems, the likelihood is proportional to the probability of observing the sequence alignment, $\mathbf{X}$, under a fully specified phylogenetic model.

$$L(\tau, \nu, \Phi) \propto f(\mathbf{X} \mid \tau, \nu, \Phi)$$

tree topology

branch lengths

substitution-model parameters

An Introduction to the Likelihood Function

What is likelihood?

The likelihood is a quantity that is proportional to the probability of observing/realizing the data, $\mathbf{X}$, under a fully specified model/hypothesis, Θ .

$$L(\Theta) \propto f(\mathbf{X}|\Theta)$$

For phylogenetic problems, the likelihood is proportional to the probability of observing the sequence alignment, $\mathbf{X}$, under a fully specified phylogenetic model.

$$L(\tau, \nu, \Phi) \propto f(\mathbf{X} \mid \tau, \nu, \Phi)$$

tree topology

branch lengths

substitution-model parameters

The likelihood is a score that measures the fit of the model to the data, providing a basis for comparing different hypotheses/parameter values on the same data

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

Consider a simple simulation experiment to develop our intuition

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

Consider a simple simulation experiment to develop our intuition

Imagine we have a fully specified phylogenetic model for four species:

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

$$Q = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

OK, so what does likelihood **really** mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

An Introduction to the Likelihood Function

OK, so what does likelihood **really** mean?

The four species have the following observed site pattern: `TTGT`
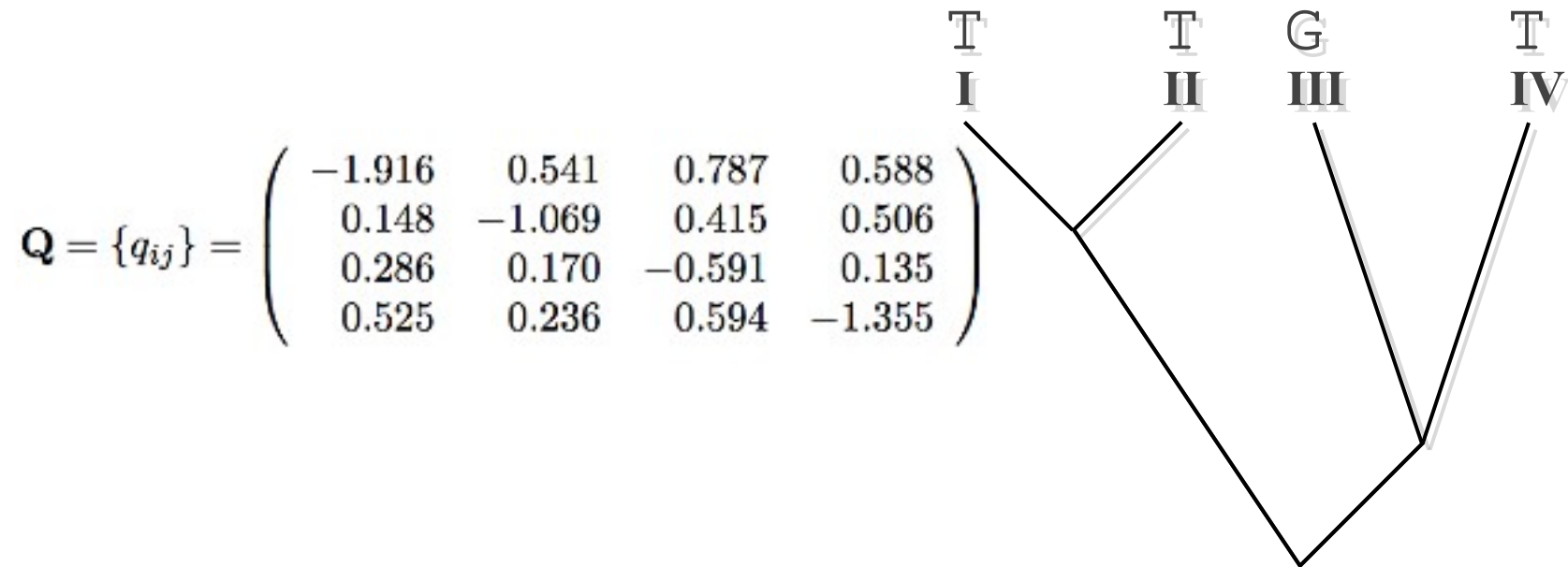
What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:

$$\mathbf{P}(100.0) = \begin{pmatrix} 0.138 & 0.188 & 0.495 & 0.179 \\ 0.138 & 0.188 & 0.495 & 0.179 \\ 0.138 & 0.188 & 0.495 & 0.179 \\ 0.138 & 0.188 & 0.495 & 0.179 \end{pmatrix}$$
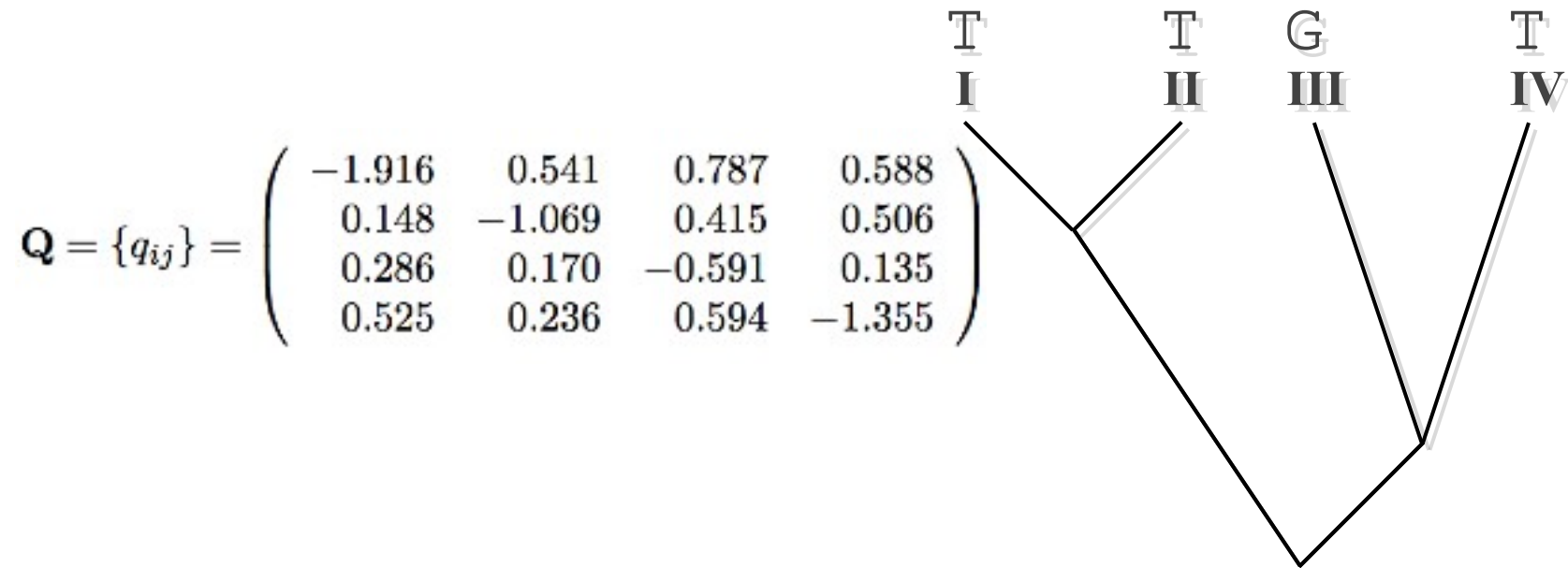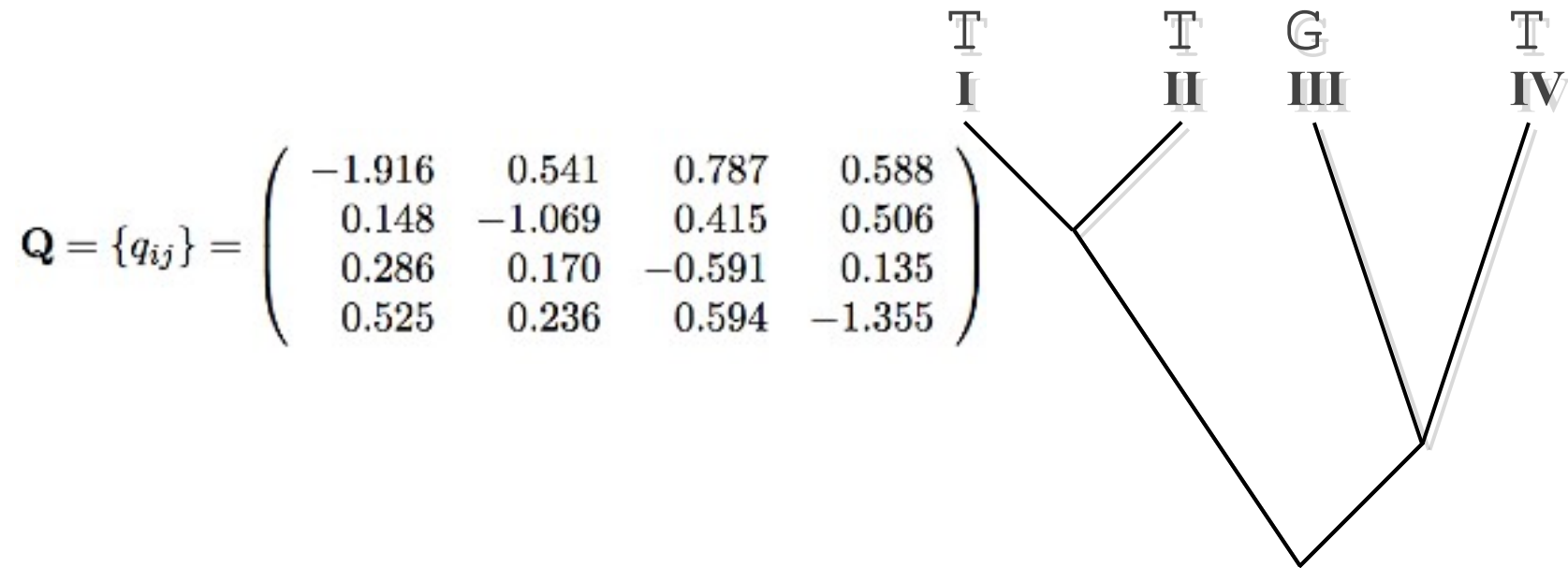
An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$Q = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:
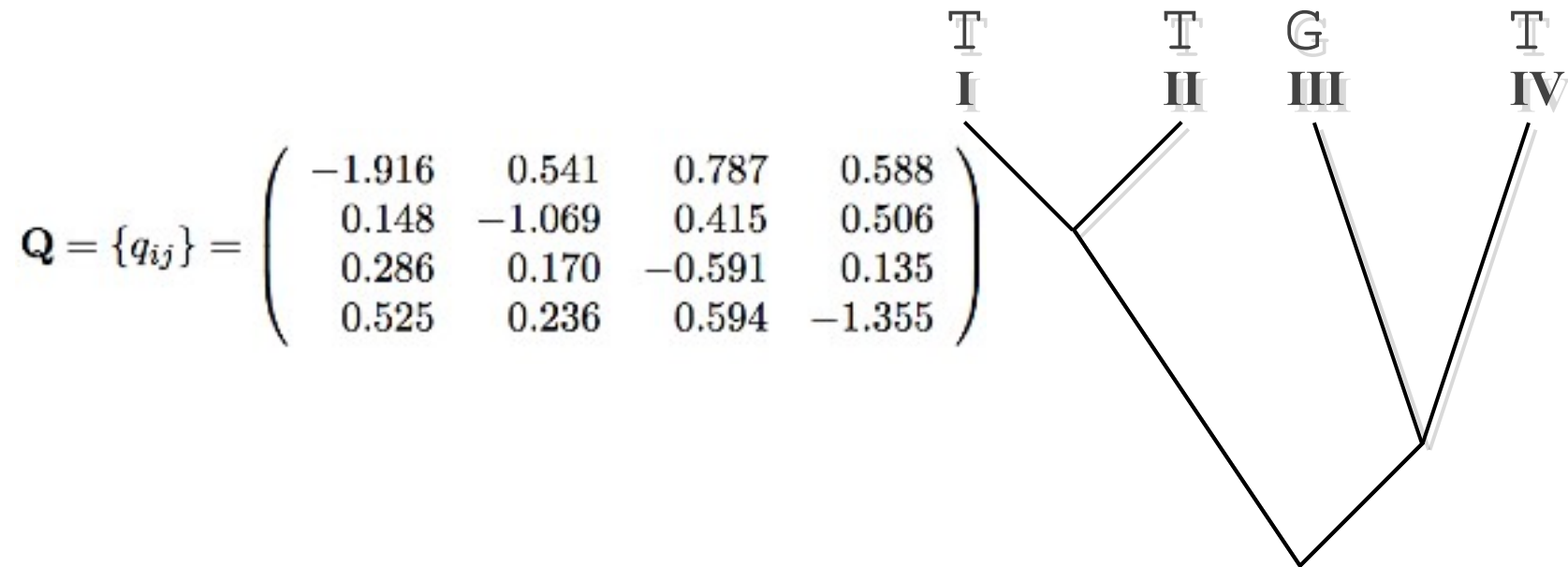
```
A = 0.138
C = 0.188
G = 0.495
T = 0.179
```

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

T (I)   T (II)   G (III)   T (IV)

The stationary frequencies have been estimated from the rate matrix:

```
A = 0.138        0-0.138       = A
C = 0.188        0.138-0.326 = C
G = 0.495        0.326-0.821 = G
T = 0.179        0.821-1       = T
```

An Introduction to the Likelihood Function

OK, so what does likelihood **really** mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$Q = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:

```
A = 0.138        0-0.138       = A
C = 0.188        0.138-0.326 = C
G = 0.495        0.326-0.821 = G
T = 0.179        0.821-1       = T
```
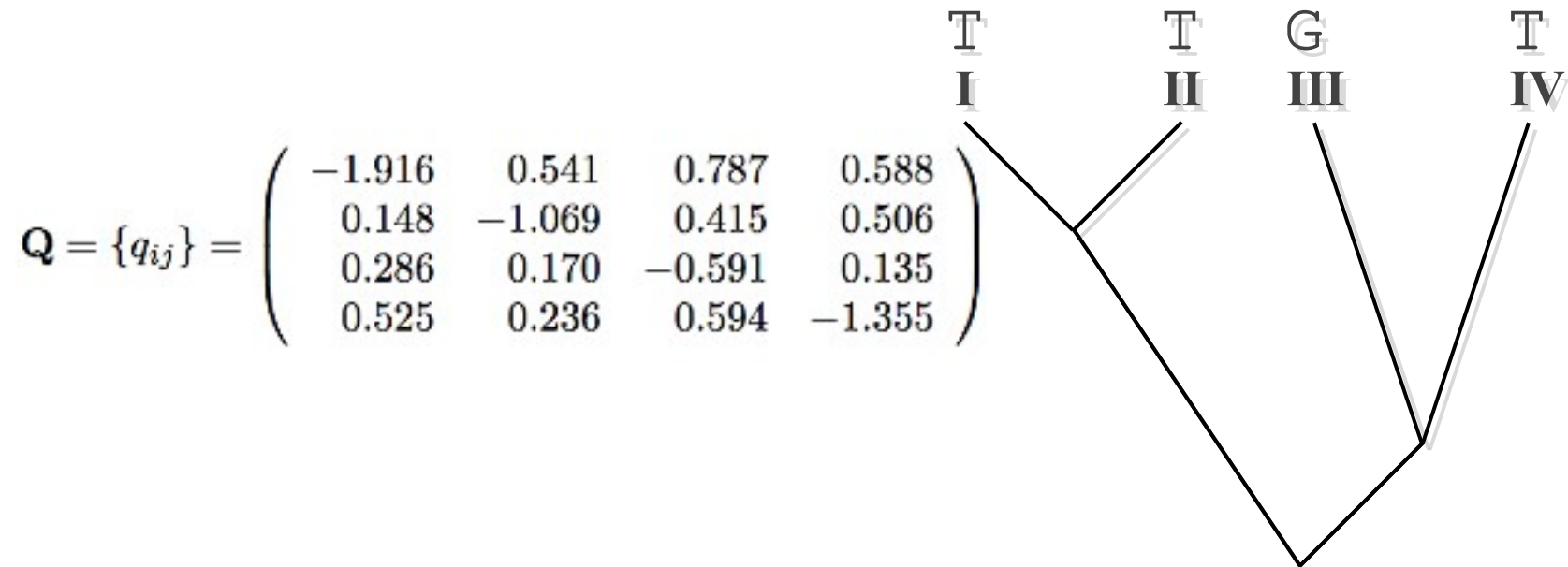
Pick a starting state at the root using `x~dnUniform(0,1)`

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:

```
A = 0.138        0-0.138       = A
C = 0.188        0.138-0.326 = C
G = 0.495        0.326-0.821 = G
T = 0.179        0.821-1       = T
```

Pick a starting state at the root using `x~dnUniform(0,1)`   `x=0.246`
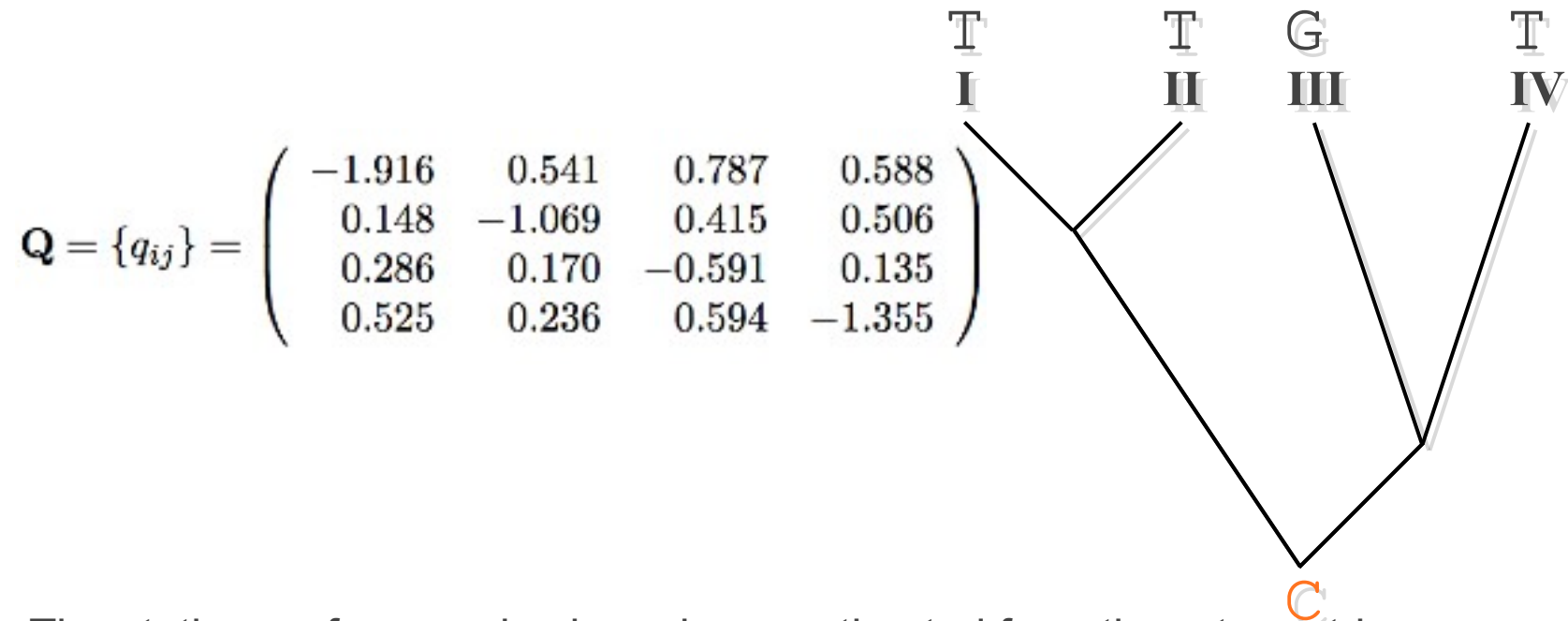
An Introduction to the Likelihood Function

OK, so what does likelihood **really** mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:

```
A = 0.138        0-0.138       = A
C = 0.188        0.138-0.326 = C
G = 0.495        0.326-0.821 = G
T = 0.179        0.821-1       = T
```
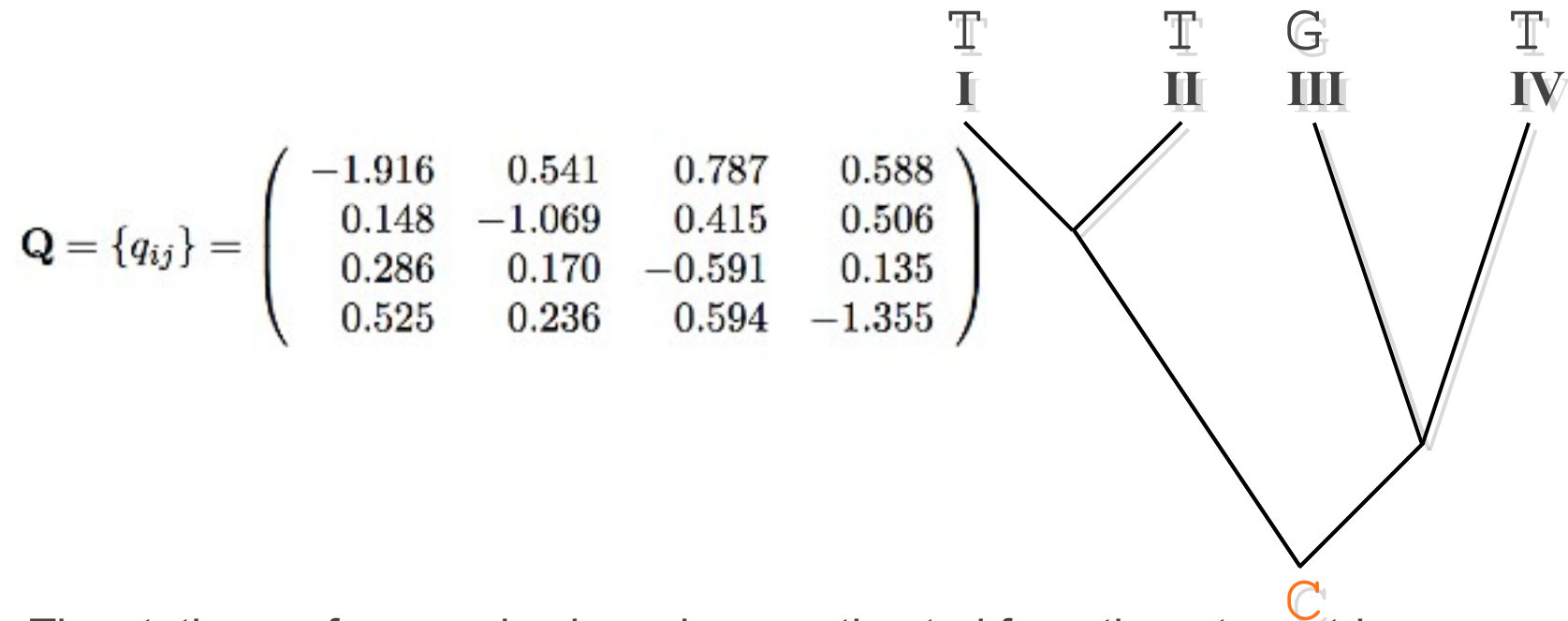
We randomly selected state `C`

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$Q = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:

```
A = 0.138        0-0.138       = A
C = 0.188        0.138-0.326 = C
G = 0.495        0.326-0.821 = G
T = 0.179        0.821-1       = T
```
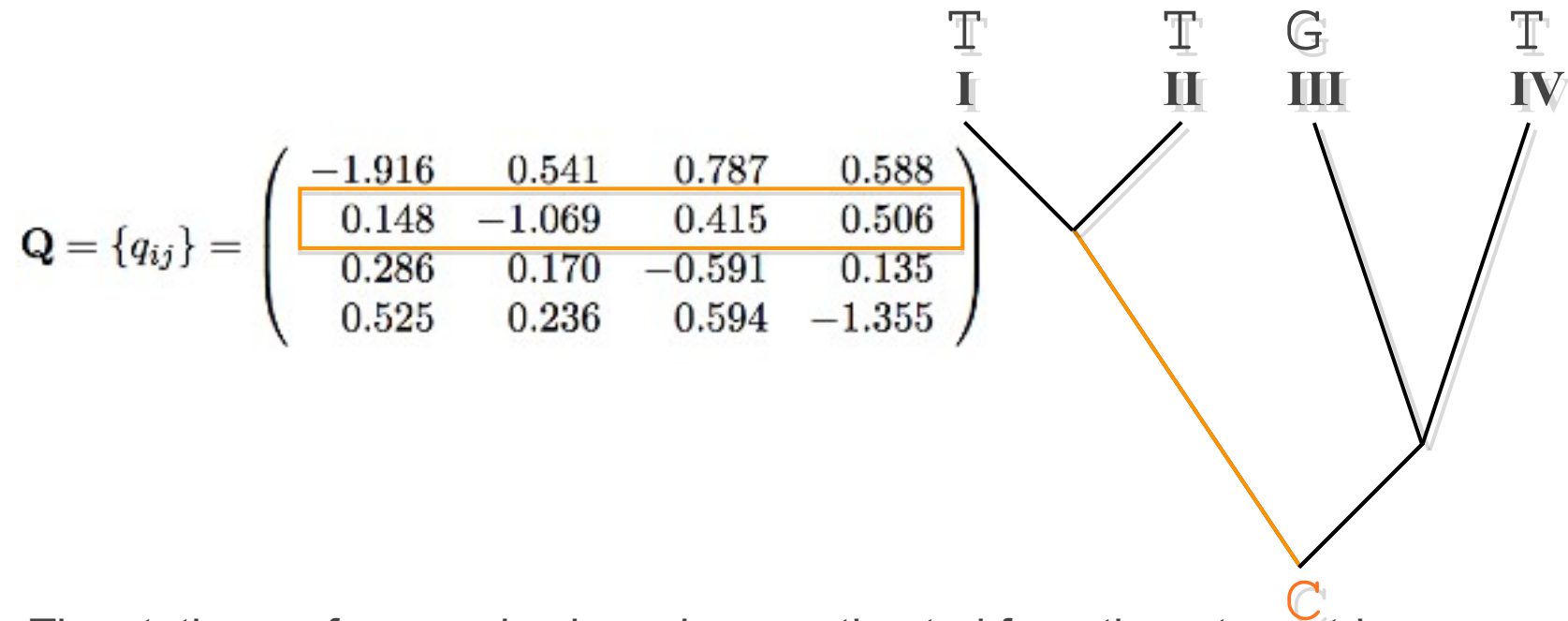
Simulate histories along each branch in a pre-order traversal

An Introduction to the Likelihood Function

OK, so what does likelihood **really** mean?

The four species have the following observed site pattern: TTGT

What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:

```
A = 0.138        0-0.138       = A
C = 0.188        0.138-0.326 = C
G = 0.495        0.326-0.821 = G
T = 0.179        0.821-1       = T
```
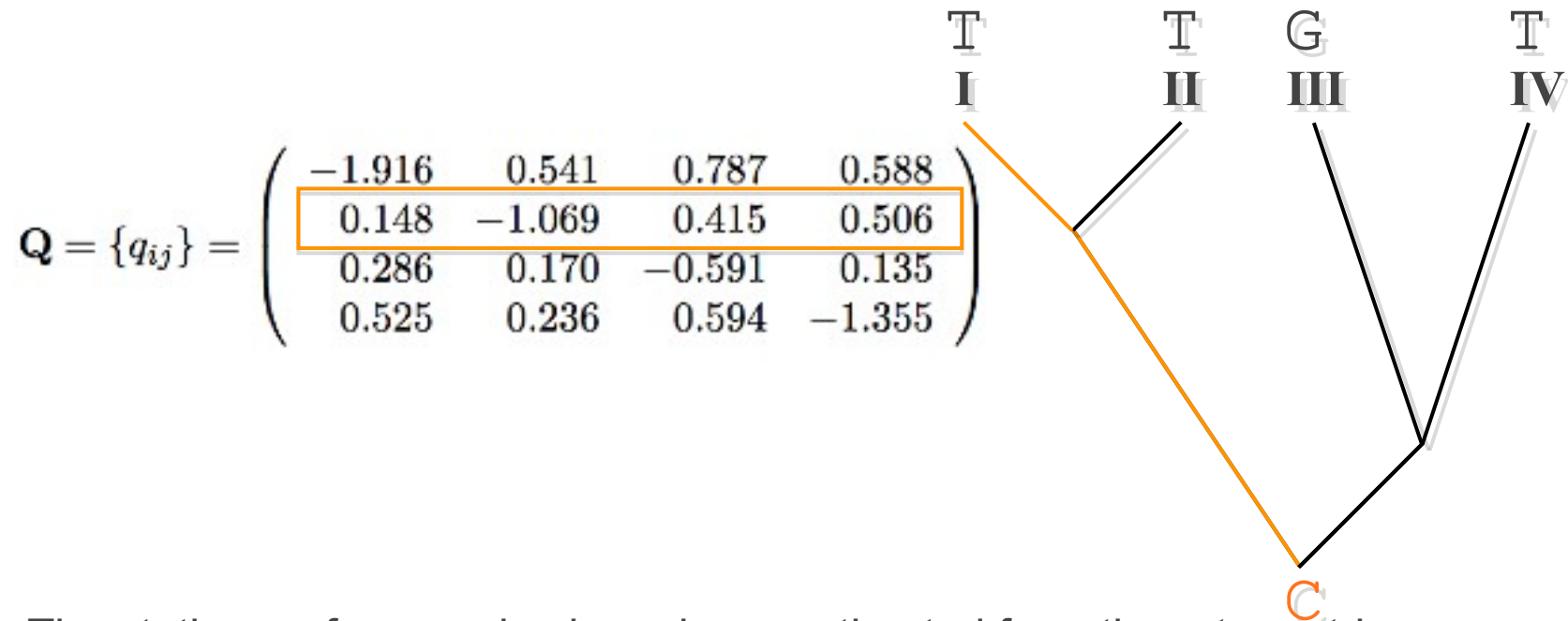
Simulate histories along each branch in a pre-order traversal

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$Q = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:

```
A = 0.138        0-0.138       = A
C = 0.188        0.138-0.326 = C
G = 0.495        0.326-0.821 = G
T = 0.179        0.821-1       = T
```

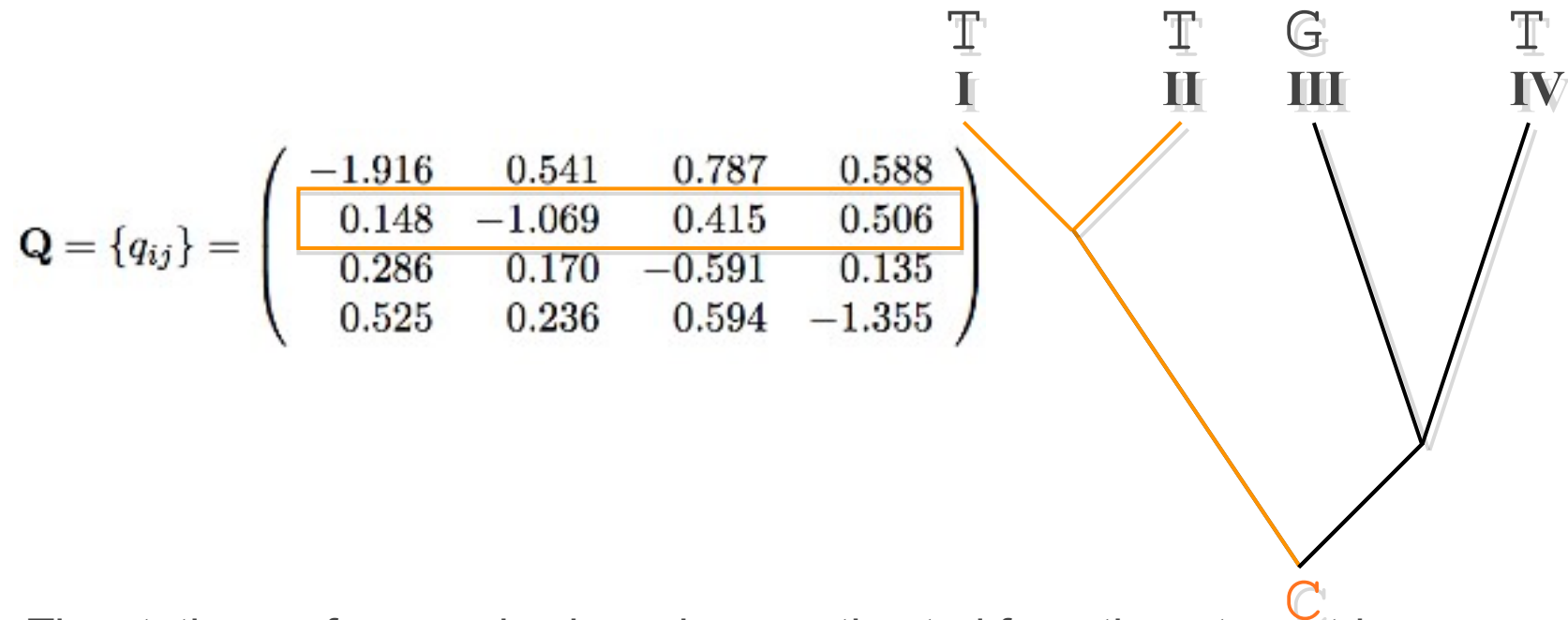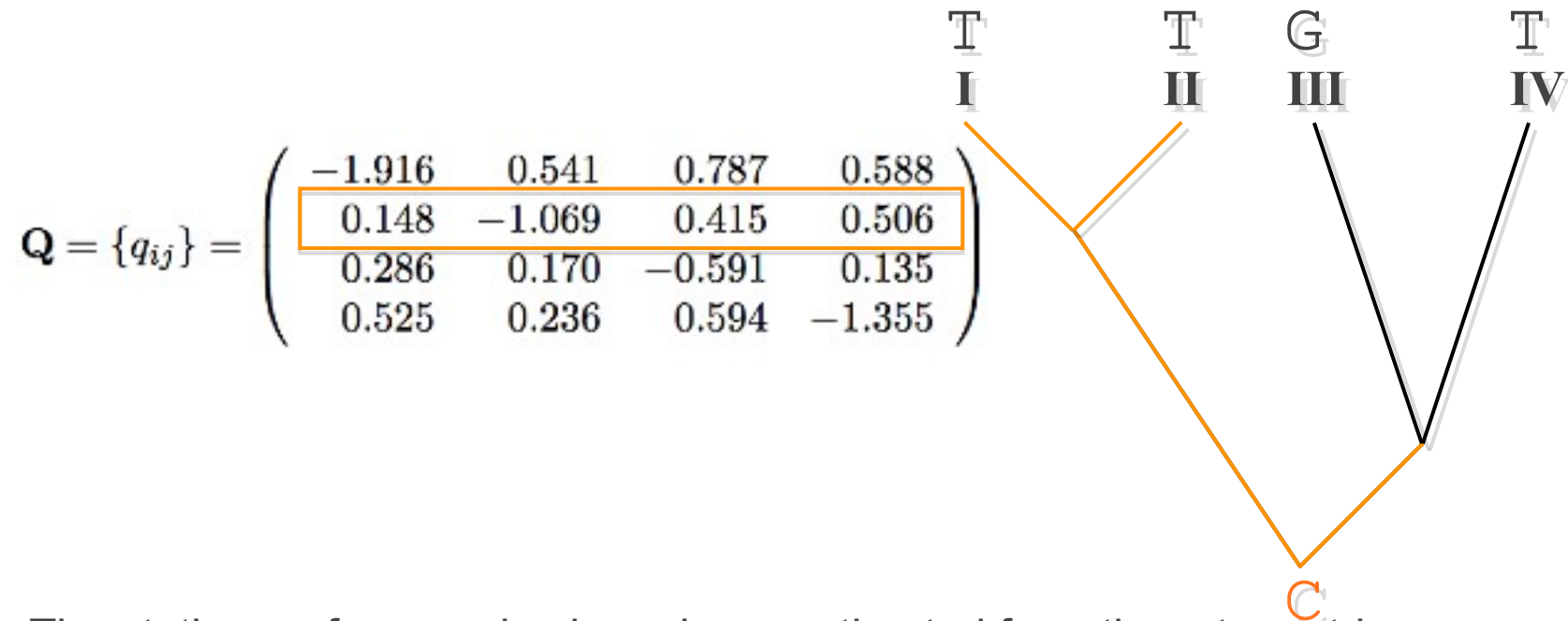Simulate histories along each branch in a pre-order traversal :

`u~dnExponential(1.069)`

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:

```
A = 0.138        0-0.138       = A
C = 0.188        0.138-0.326 = C
G = 0.495        0.326-0.821 = G
T = 0.179        0.821-1       = T
```

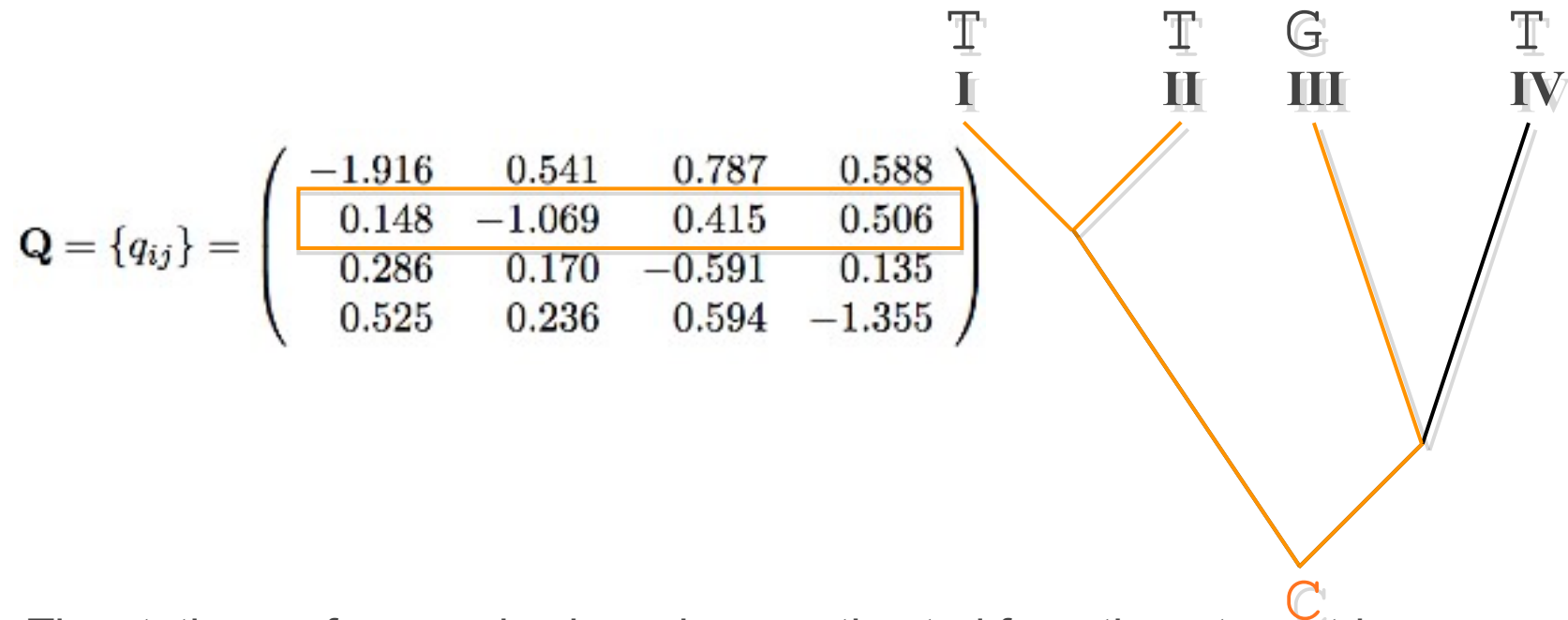Simulate histories along each branch in a pre-order traversal :

`u~dnExponential(1.069)`

An Introduction to the Likelihood Function

OK, so what does likelihood **really** mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$Q = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:

```
A = 0.138        0-0.138       = A
C = 0.188        0.138-0.326 = C
G = 0.495        0.326-0.821 = G
T = 0.179        0.821-1       = T
```

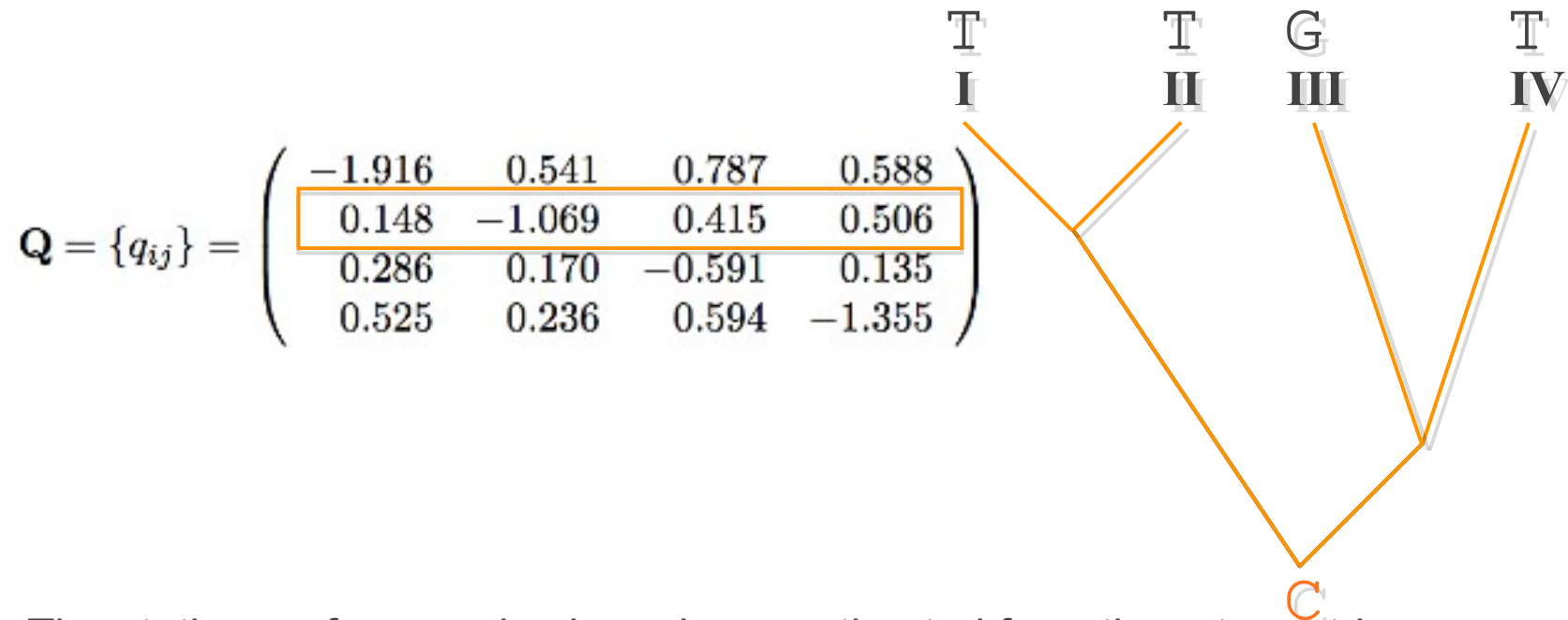Simulate histories along each branch in a pre-order traversal :

`u~dnExponential(1.069)`

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:

```
A = 0.138        0-0.138       = A
C = 0.188        0.138-0.326 = C
G = 0.495        0.326-0.821 = G
T = 0.179        0.821-1       = T
```

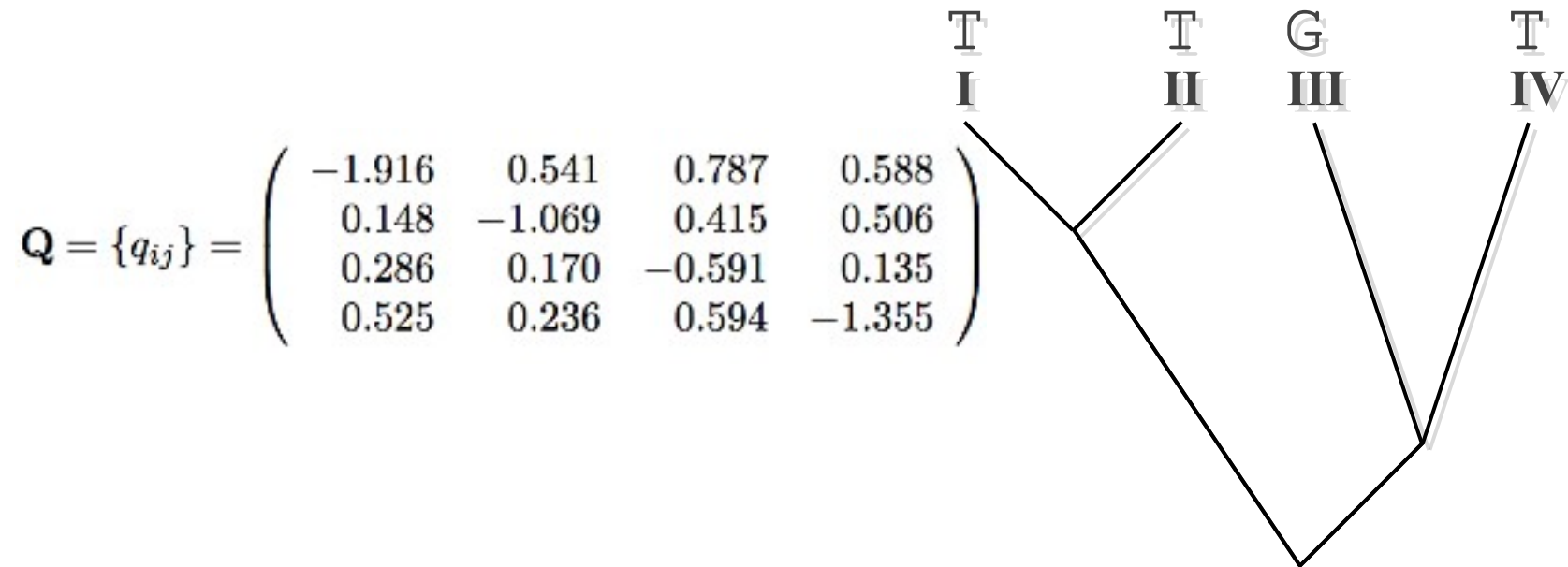Simulate histories along each branch in a pre-order traversal :

`u~dnExponential(1.069)`

An Introduction to the Likelihood Function

OK, so what does likelihood **really** mean?

The four species have the following observed site pattern: `TTGT`

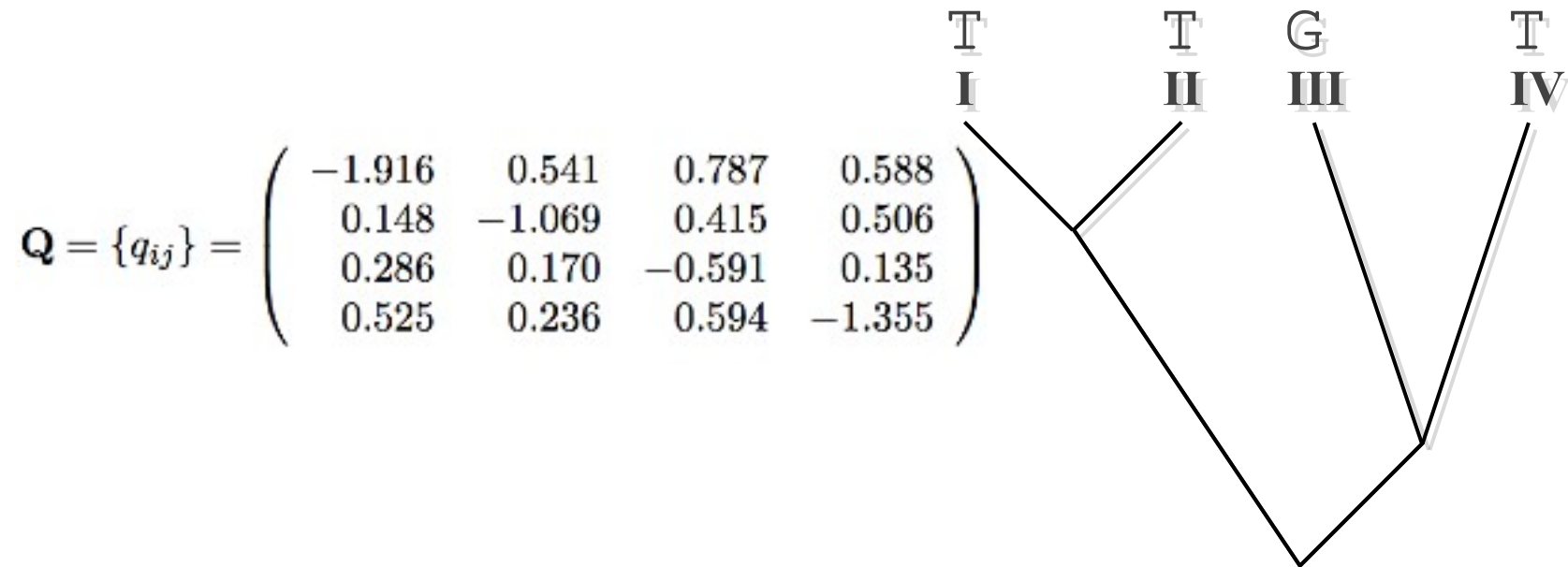What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:

```
A = 0.138        0-0.138       = A
C = 0.188        0.138-0.326 = C
G = 0.495        0.326-0.821 = G
T = 0.179        0.821-1       = T
```

Simulate histories along each branch in a pre-order traversal :

`u~dnExponential(1.069)`

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

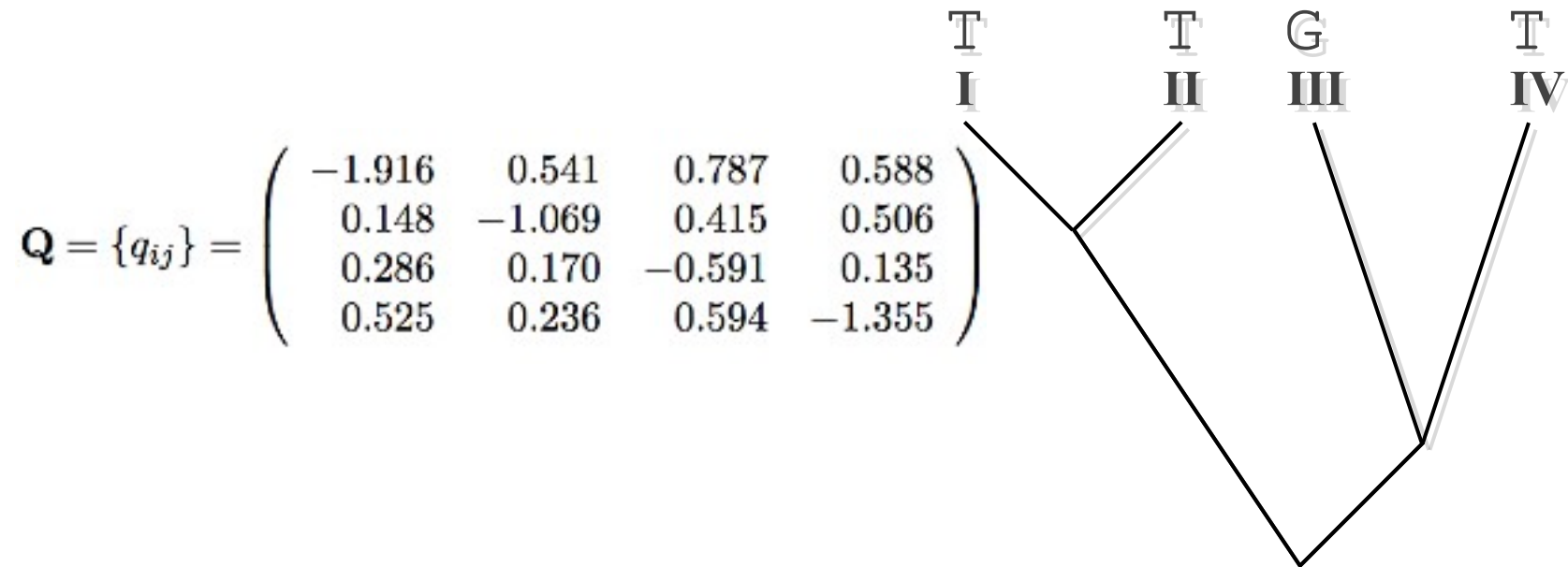What is the likelihood of observing the site pattern under this model?

$$Q = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

The stationary frequencies have been estimated from the rate matrix:

```
A = 0.138        0-0.138        = A
C = 0.188        0.138-0.326 = C
G = 0.495        0.326-0.821 = G
T = 0.179        0.821-1        = T
```

Simulate histories along each branch in a pre-order traversal :

`u~dnExponential(1.069)`

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

We repeat the simulation 100,000,000 times and record the frequency of outcomes that match the observed tip states.

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

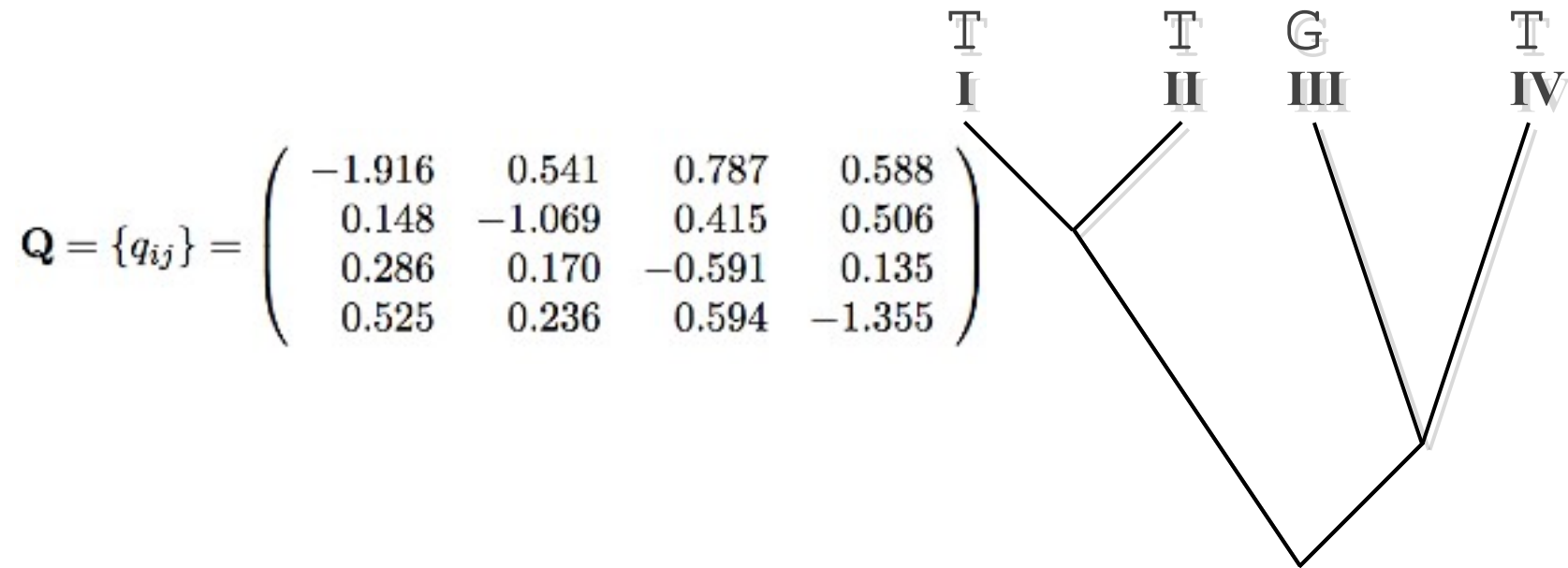What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

We repeat the simulation $100,000,000$ times and record the frequency of outcomes that match the observed tip states.

The observed site pattern, `TTGT`, is one of $4^4 = 256$ possible site patterns.

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$
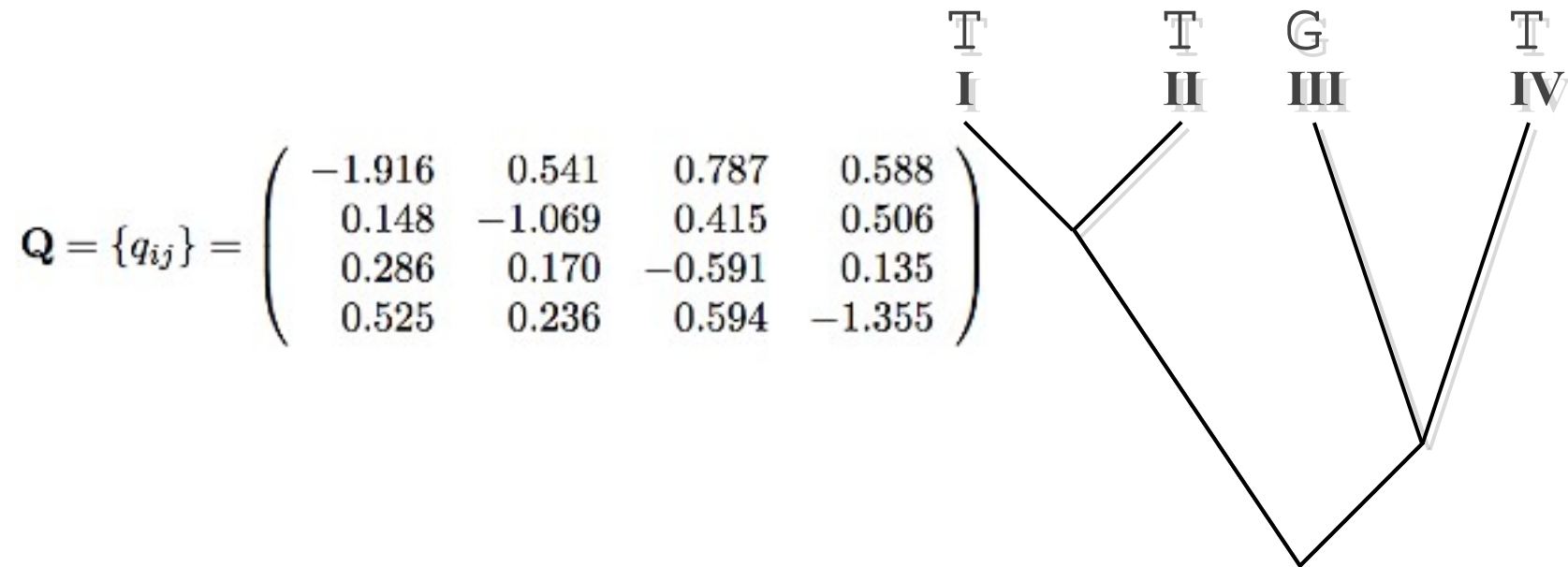
Of the $100{,}000{,}000$ simulations, $850{,}358$ realize the observed data (`TTGT`); the estimated probability of the observed data is therefore 0.00850358.

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

Of the $100{,}000{,}000$ simulations, $850{,}358$ realize the observed data (`TTGT`); the estimated probability of the observed data is therefore 0.00850358.

The probability includes all of the possible histories that can give rise to the data.

Number of Changes

Pattern	0	1	2	3	4	5	6	7
TTGT	0.000000	0.881408	0.075358	0.037817	0.004725	0.000622	0.000062	0.000007

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$
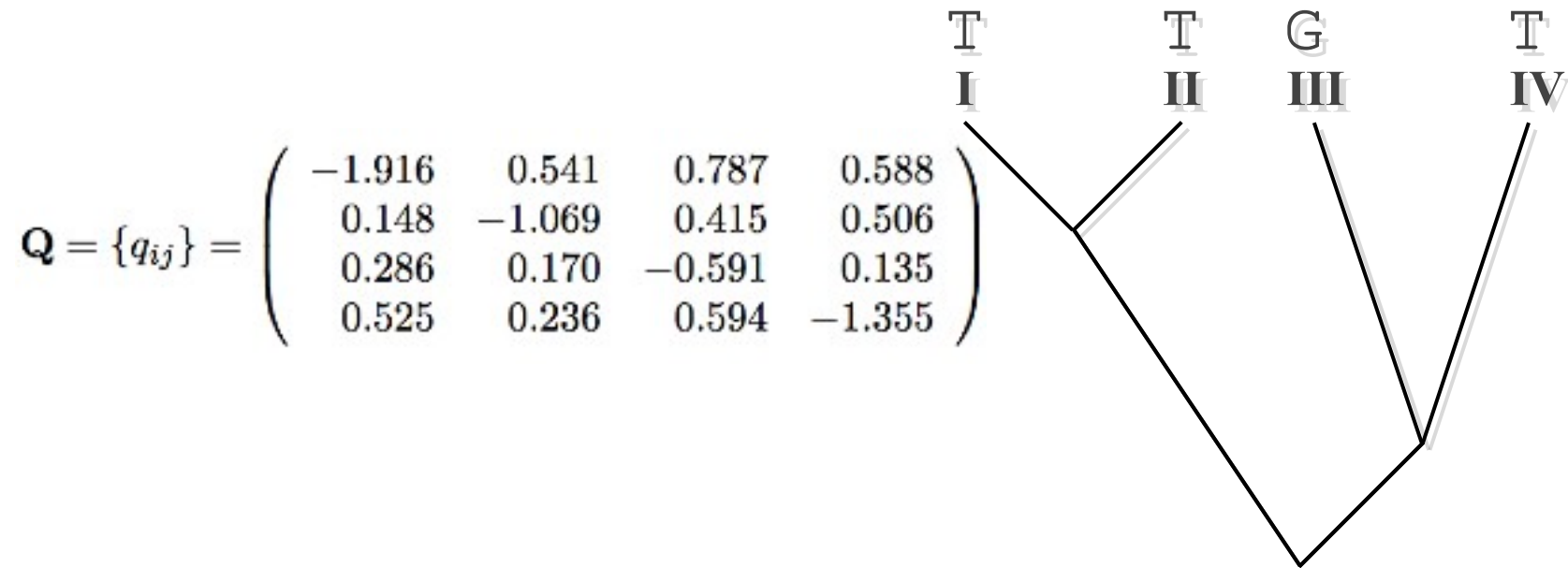
Using Monte Carlo simulation to estimate site likelihood is pedagogical, but too inefficient for the analysis of real data.

An Introduction to the Likelihood Function

OK, so what does likelihood *really* mean?

The four species have the following observed site pattern: `TTGT`

What is the likelihood of observing the site pattern under this model?

$$\mathbf{Q} = \{q_{ij}\} = \begin{pmatrix} -1.916 & 0.541 & 0.787 & 0.588 \\ 0.148 & -1.069 & 0.415 & 0.506 \\ 0.286 & 0.170 & -0.591 & 0.135 \\ 0.525 & 0.236 & 0.594 & -1.355 \end{pmatrix}$$

Using Monte Carlo simulation to estimate site likelihood is pedagogical, but too inefficient for the analysis of real data.

e.g., there are $>1,000,000$ possible site patterns for a tree with 10 species.

Plan: Criteria for evaluating phylogenies

- Criteria for evaluating phylogenetic trees:

  - Parsimony

  - Distance methods

  - Maximum Likelihood

    - Using a Monte Carlo simulation

    - Using Felsenstein's pruning algorithm

  - Posterior probability (Bayesian approach)

- *Conventions:*

  - We're dealing with aligned sequence data
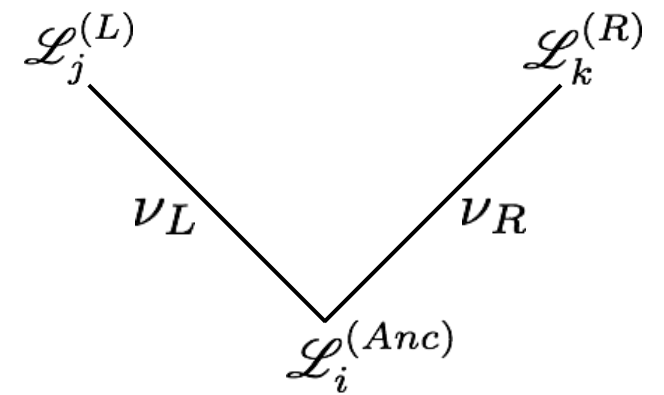
  - gaps are not taken into account

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Make the math mirror the tree to avoid redundant calculations.
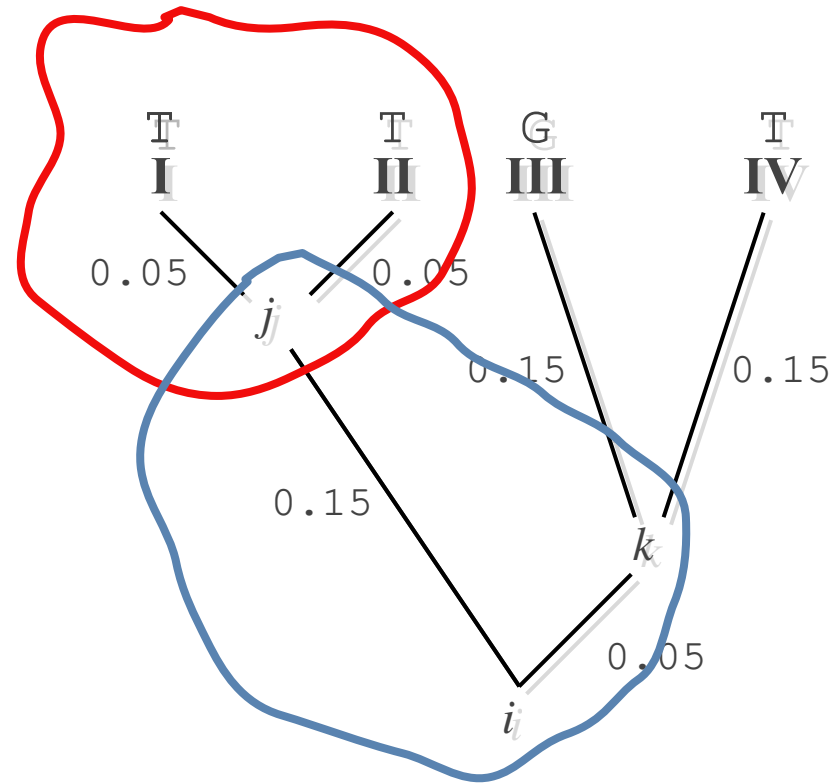

Joe Felsenstein (c.1981)
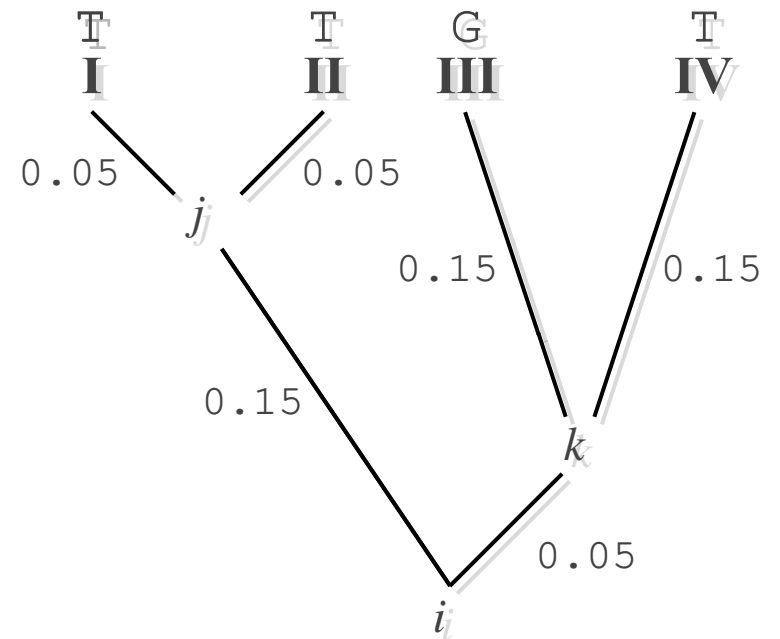
How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Make the math mirror the tree to avoid redundant calculations.

Multiple summations over ancestral-state configurations at internal nodes are moved as far to the right as possible.

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Make the math mirror the tree to avoid redundant calculations.

Multiple summations over ancestral-state configurations at internal nodes are moved as far to the right as possible.

The pruning algorithm reduces this (64 evaluations):

$$\sum_{i\in(A,C,G,T)}\ \sum_{j\in(A,C,G,T)}\ \sum_{k\in(A,C,G,T)} \pi_i \times p_{ij}(0.15) \times p_{jT}(0.05) \times p_{jT}(0.05) \times p_{ik}(0.05) \times p_{kG}(0.15) \times p_{kT}(0.15)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Make the math mirror the tree to avoid redundant calculations.

Multiple summations over ancestral-state configurations at internal nodes are moved as far to the right as possible.

The pruning algorithm reduces this (64 evaluations):

$$\sum_{i \in (A,C,G,T)} \sum_{j \in (A,C,G,T)} \sum_{k \in (A,C,G,T)} \pi_i \times p_{ij}(0.15) \times p_{jT}(0.05) \times p_{jT}(0.05) \times p_{ik}(0.05) \times p_{kG}(0.15) \times p_{kT}(0.15)$$

to this (12 evaluations):

$$\sum_{i \in (A,C,G,T)} \pi_i \left(\sum_{j \in (A,C,G,T)} p_{ij}(0.15) \times p_{jT}(0.05) \times p_{jT}(0.05) \right) \left(\sum_{k \in (A,C,G,T)} p_{ik}(0.05) \times p_{kG}(0.15) \times p_{kT}(0.15) \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

How Do We Calculate Site Likelihoods?

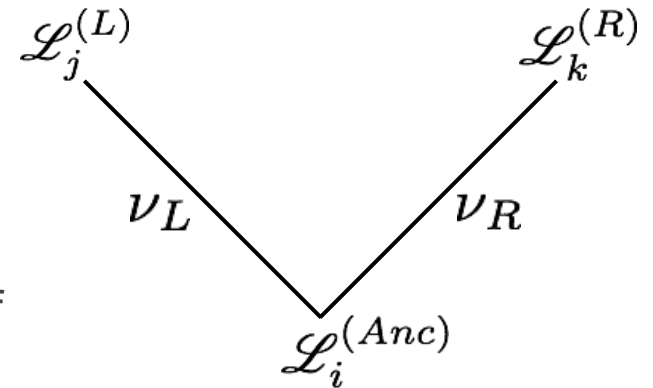The Felsenstein Pruning Algorithm

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Evaluating the likelihood of a site involves the
recursive calculation of *conditional likelihoods*
from the tips of the tree to the root.

$$\mathscr{L}_j^{(L)} \qquad\qquad\qquad \mathscr{L}_k^{(R)}$$

$$\nu_L \qquad\qquad \nu_R$$

$$\mathscr{L}_i^{(Anc)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Evaluating the likelihood of a site involves the recursive calculation of *conditional likelihoods* from the tips of the tree to the root.

The conditional likelihoods $\mathscr{L}_i^{(Anc)}$ are the probabilities of the observations above some point in the tree (a local ancestral node, Anc), conditional on state i at that node.

$$\mathscr{L}_j^{(L)} \qquad\qquad \mathscr{L}_k^{(R)}$$

$$\nu_L \qquad\qquad \nu_R$$

$$\mathscr{L}_i^{(Anc)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Evaluating the likelihood of a site involves the recursive calculation of *conditional likelihoods* from the tips of the tree to the root.

The conditional likelihoods $\mathscr{L}_i^{(Anc)}$ are the probabilities of the observations above some point in the tree (a local ancestral node, Anc), conditional on state i at that node.

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (\text{A,C,G,T})} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (\text{A,C,G,T})} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

where $i \in (\text{A,C,G,T})$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm
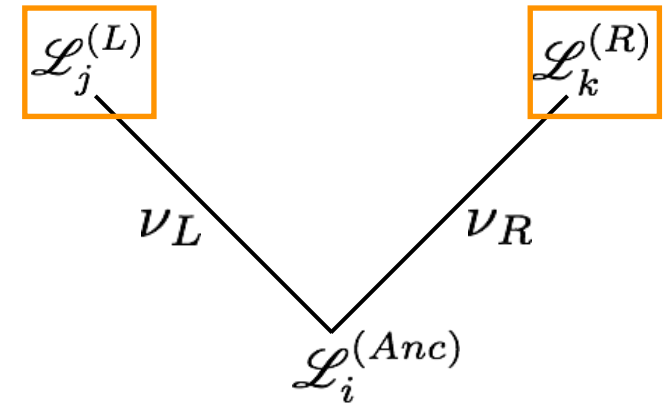
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\,\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\,\mathscr{L}_k^{(R)} \right)$$
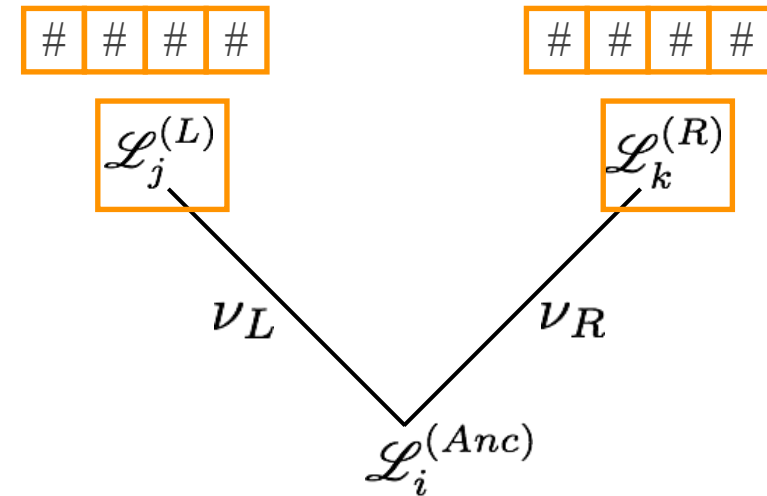
These conditional likelihoods are the likelihoods of
the observations above the end of each branch...

How Do We Calculate Site Likelihoods?
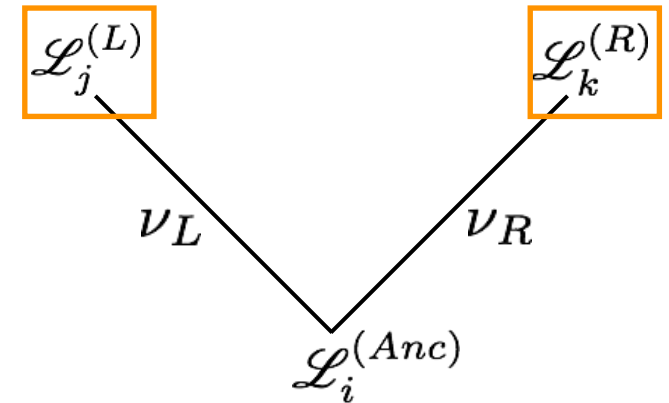
The Felsenstein Pruning Algorithm

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \boxed{\mathscr{L}_j^{(L)}} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \boxed{\mathscr{L}_k^{(R)}} \right)$$

These conditional likelihoods are either known (if tip)
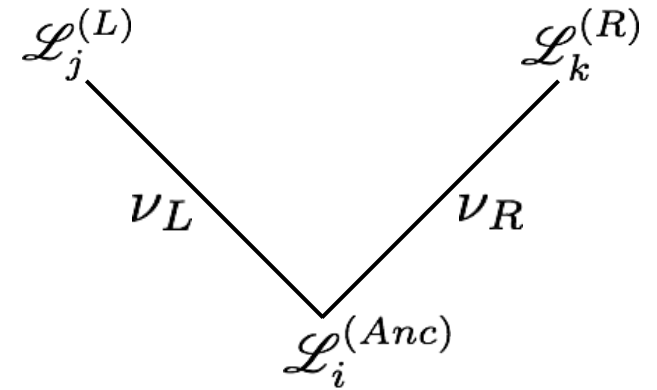or have already been computed (in a previous step).

How Do We Calculate Site Likelihoods?
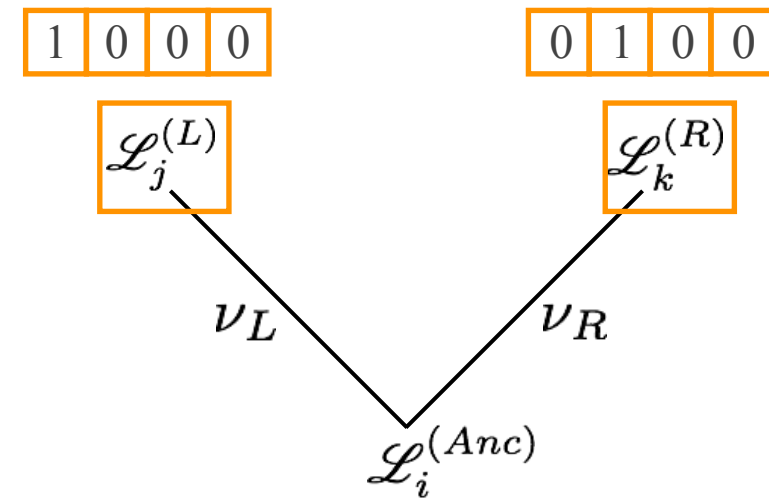
The Felsenstein Pruning Algorithm

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \boxed{\mathscr{L}_j^{(L)}} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \boxed{\mathscr{L}_k^{(R)}} \right)$$

These conditional likelihoods are either known (if tip) or have already been computed (in a previous step).

If the branch is a tip, the probability of the observed state is 1, otherwise it is 0 (*e.g.*, depicted for states A and C in L and R).

How Do We Calculate Site Likelihoods?
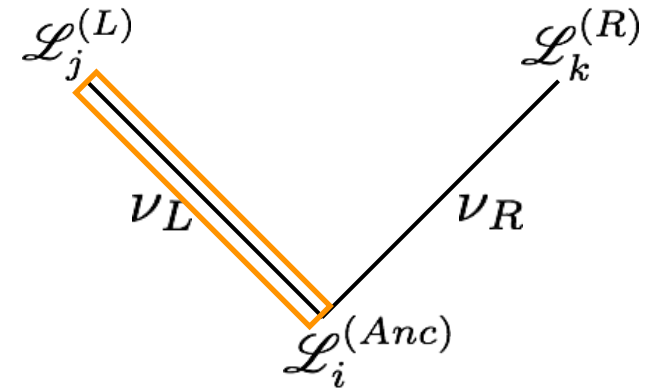
The Felsenstein Pruning Algorithm

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \boxed{\mathscr{L}_j^{(L)}} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \boxed{\mathscr{L}_k^{(R)}} \right)$$

These conditional likelihoods are either known (if tip) or have already been computed (in a previous step).

If the branch is a tip, the probability of the observed state is 1, otherwise it is 0 (*e.g.*, depicted for states A and C in L and R).

If the branch is not a tip, the probability of each possible state has previously been evaluated in a prior step.

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\boxed{\mathscr{L}_j^{(L)}} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\boxed{\mathscr{L}_k^{(R)}} \right)$$

These conditional likelihoods are either known (if tip) or have already been computed (in a previous step).

If the branch is a tip, the probability of the observed state is 1, otherwise it is 0 (*e.g.*, depicted for states A and C in L and R).

If the branch is not a tip, the probability of each possible state has previously been evaluated in a prior step.

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, *how* do we calculate the conditional likelihoods?

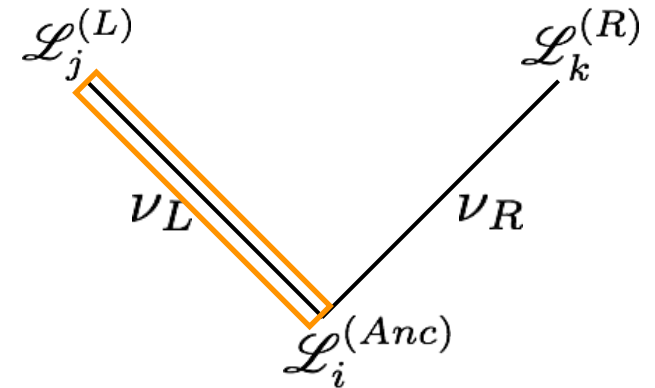$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, *how* do we calculate the conditional likelihoods?

$$\mathscr{L}_j^{(L)} \qquad \mathscr{L}_k^{(R)}$$

$$\nu_L \qquad \nu_R$$

$$\mathscr{L}_i^{(Anc)}$$

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (\text{A,C,G,T})} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right)$$

Let's first focus on the left descendant branch.

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, *how* do we calculate the conditional likelihoods?

$$\mathscr{L}_j^{(L)} \qquad \mathscr{L}_k^{(R)}$$

$$\nu_L \qquad \nu_R$$
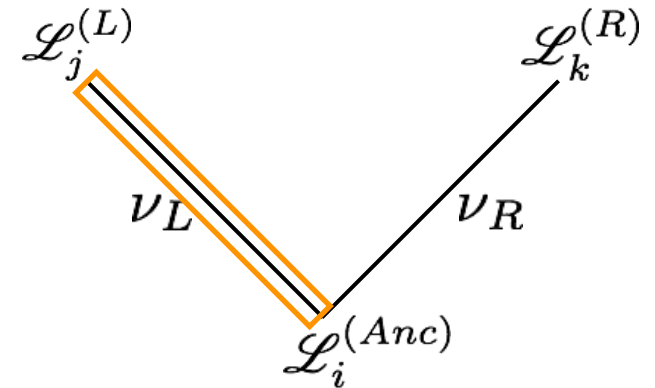
$$\mathscr{L}_i^{(Anc)}$$

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right)$$

for each possible end state sum…

…the product of the transition probabilities of changes from i to j over branch L…

…and the conditional likelihood of each end state, j.

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm
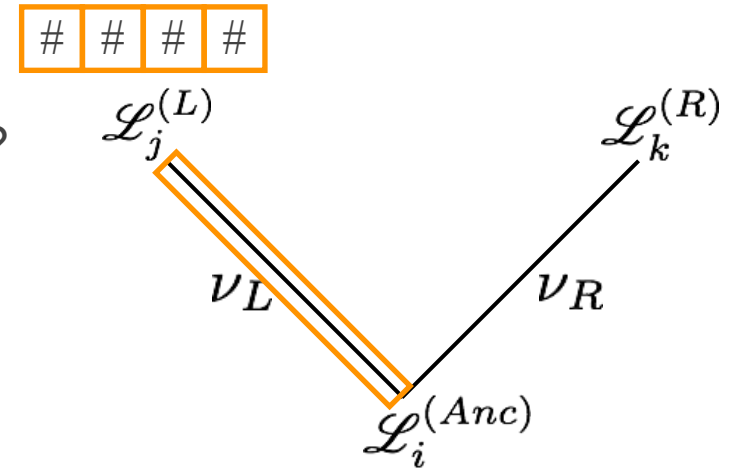
So, *how* do we calculate the conditional likelihoods?
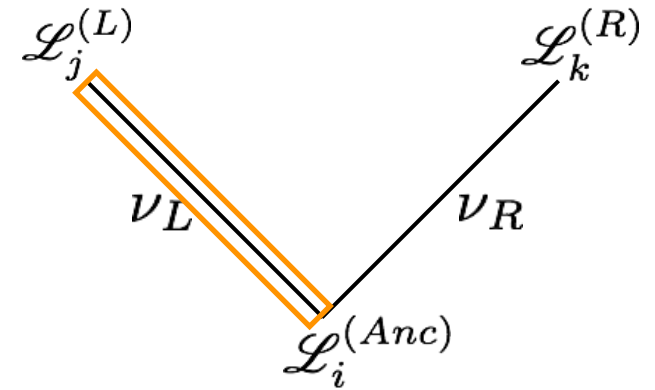
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} \boxed{p_{ij}(\nu_L)\mathscr{L}_j^{(L)}} \right)$$

Recall that the transition probabilities, $\mathbf{P}_{ij}(\nu_L)$, of histories ending in state j that were initiated from state i and run over branch L are either approximated (by Monte Carlo simulation) or solved by exponentiating the product of the instantaneous-rate matrix, $\mathbf{Q}$, and branch length, ν_L

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, *how* do we calculate the conditional likelihoods?

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} \boxed{p_{ij}(\nu_L)} \mathscr{L}_j^{(L)} \right)$$

$$\mathbf{Q} = q_{ij} = \begin{pmatrix} - & \mu a \pi_C & \mu b \pi_G & \mu c \pi_T \\ \mu a \pi_A & - & \mu d \pi_G & \mu e \pi_T \\ \mu b \pi_A & \mu d \pi_C & - & \mu f \pi_T \\ \mu c \pi_A & \mu e \pi_C & \mu f \pi_G & - \end{pmatrix}$$

$$\mathbf{P}_{ij}(\nu) = e^{\mathbf{Q}\nu}$$

Recall that the transition probabilities, $\mathbf{P}_{ij}(\nu_L)$, of histories ending in state j that were initiated from state i and run over branch L are either approximated (by Monte Carlo simulation) or solved by exponentiating the product of the instantaneous-rate matrix, $\mathbf{Q}$, and branch length, ν_L

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, *how* do we calculate the conditional likelihoods?

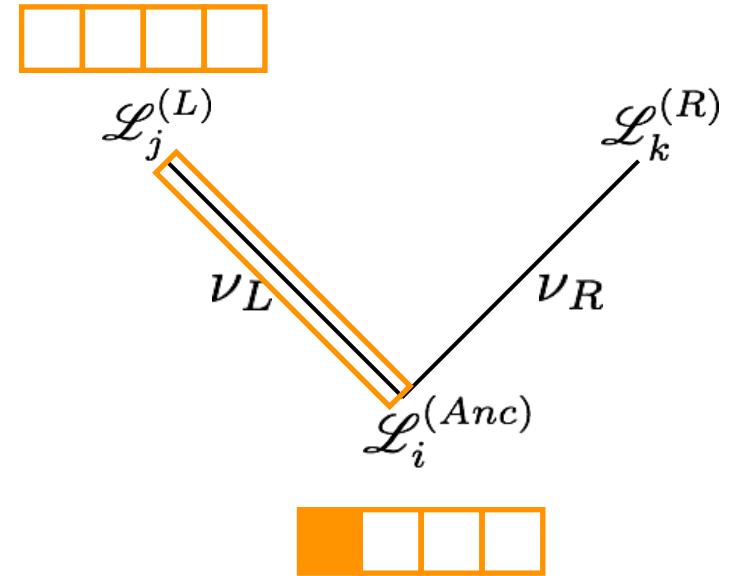$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} \boxed{p_{ij}(\nu_L)} \mathscr{L}_j^{(L)} \right)$$

The probability that the process is in state j at the end of the branch, times…

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, *how* do we calculate the conditional likelihoods?

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right)$$

The probability that the process is in state j at the end of the branch, times…

…the probability of everything 'above' the end of the branch, conditional on state j.

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

#	#	#	#

So, *how* do we calculate the conditional likelihoods?

$$\mathscr{L}_j^{(L)} \qquad\qquad\qquad \mathscr{L}_k^{(R)}$$

$$\nu_L \qquad\qquad \nu_R$$
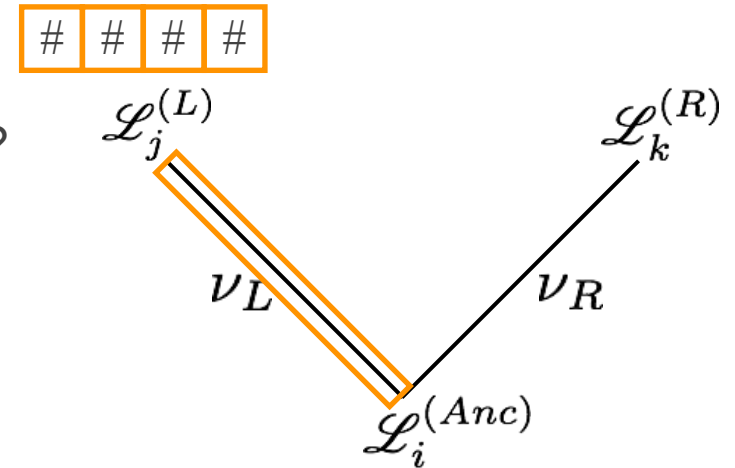
$$\mathscr{L}_i^{(Anc)}$$

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right)$$

We iteratively compute this product for each of the four j end states, A,C,G,T…

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, *how* do we calculate the conditional likelihoods?
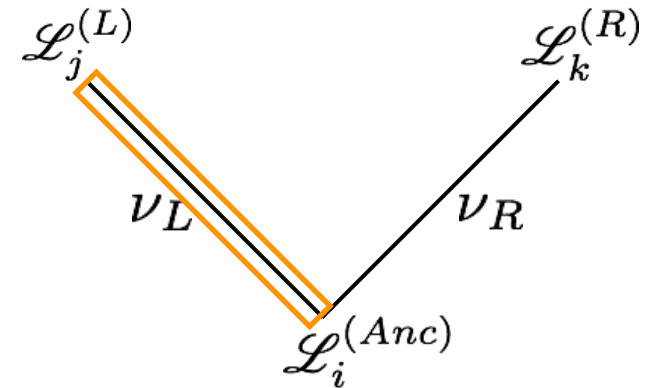
$$\mathcal{L}_i^{(Anc)} = \left(\boxed{\sum_{j \in (A,C,G,T)}} p_{ij}(\nu_L)\mathcal{L}_j^{(L)} \right)$$

We iteratively compute this product for each of the four j end states, A,C,G,T…
…and then we sum these four terms because they are mutually exclusive.

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

First, set the start state, i, to $i = A$

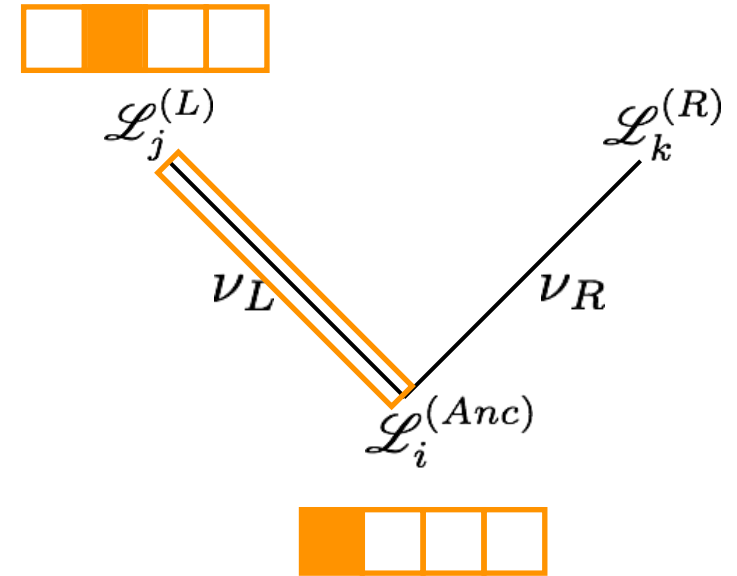$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right)$$

$$\mathscr{L}_A^{(Anc)} \underset{\nu_L}{\rule{5cm}{0.4pt}} \mathscr{L}_A^{(L)}$$

The transition probabilities $p_{AA}(\nu_L)$ over branch ν_L are calculated by exponentiating the product of the instantaneous rate matrix, $\mathbf{Q}$, and branch length ν_L.

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

First, set the start state, i, to $i = A$
and set the end state, j, to $j = A$

$$\mathcal{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathcal{L}_j^{(L)} \right)$$

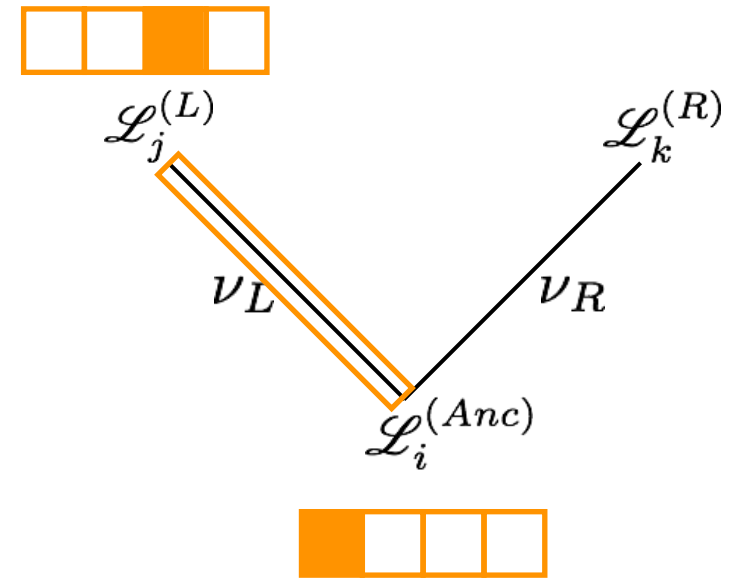$\mathcal{L}_A^{(Anc)}$ ——————————— $\mathcal{L}_A^{(L)}$
$\qquad\qquad\quad \nu_L$

The transition probabilities $p_{AA}(\nu_L)$ over branch ν_L
are calculated by exponentiating the product of the
instantaneous rate matrix, $\mathbf{Q}$, and branch length ν_L.

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm
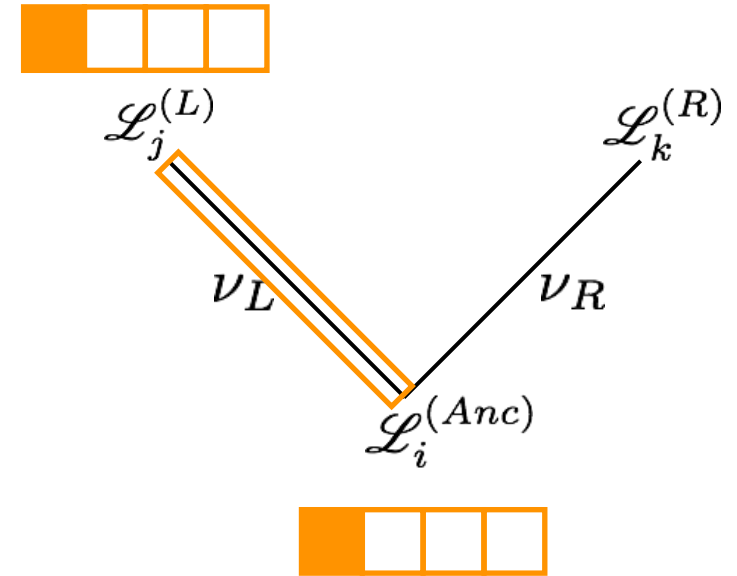
First, set the start state, i, to $i = A$
and set the end state, j, to $j = A$
next set the end state, j, to $j = C\ldots$
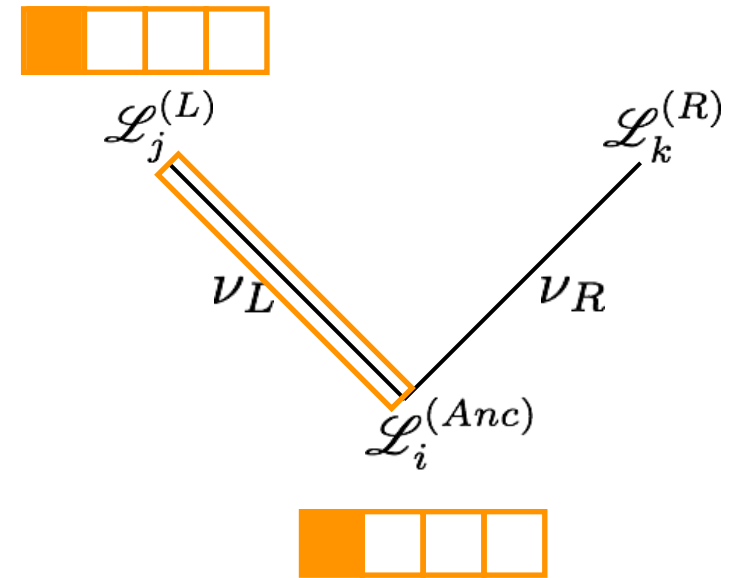
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right)$$

$\mathscr{L}_A^{(Anc)} \underline{\hspace{3cm}}_{\nu_L} \mathscr{L}_C^{(Anc)}$

Then we do the same thing for end state $j = C\ldots$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

First, set the start state, i, to $i = A$
and set the end state, j, to $j = A$
next set the end state, j, to $j = C$
then set the end state, j, to $j = G$…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right)$$

$\mathscr{L}_A^{(Anc)} \underline{\qquad\qquad\qquad} \mathscr{L}_G^{(Anc)}$
ν_L

Then we do the same thing for end state $j = G$…

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

First, set the start state, i, to $i = A$
and set the end state, j, to $j = A$
next set the end state, j, to $j = C$
then set the end state, j, to $j = G$
then set the end state, j, to $j = T$

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right)$$

$$\mathscr{L}_A^{(Anc)} \underbrace{\hspace{4cm}}_{\nu_L} \mathscr{L}_T^{(Anc)}$$

Finally, we do the same thing for end state $j = T$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Next, we sum the fractional likelihoods for each
of the four end states $j = \{A, C, G, T\}...$
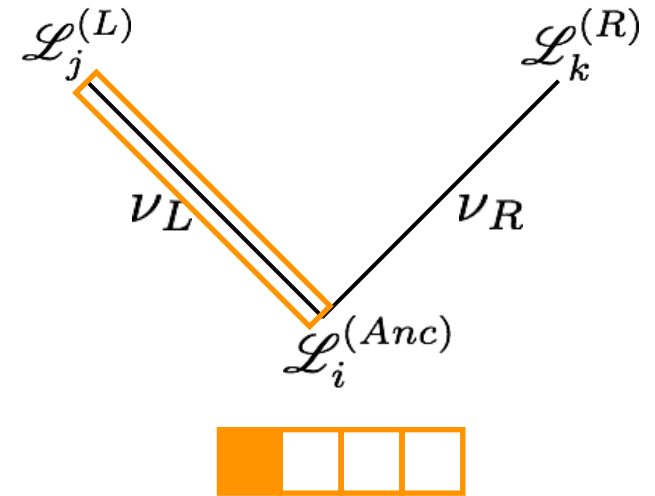
$$\mathcal{L}_i^{(Anc)} = \left(\boxed{\sum_{j \in (A,C,G,T)}} p_{ij}(\nu_L)\mathcal{L}_j^{(L)} \right)$$

Then we sum the fractional likelihoods for each
of the four end states $j = \{A, C, G, T\}...$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Next, we repeat the process for the right branch, ν_R.

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j\in(\text{A,C,G,T})} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k\in(\text{A,C,G,T})} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm
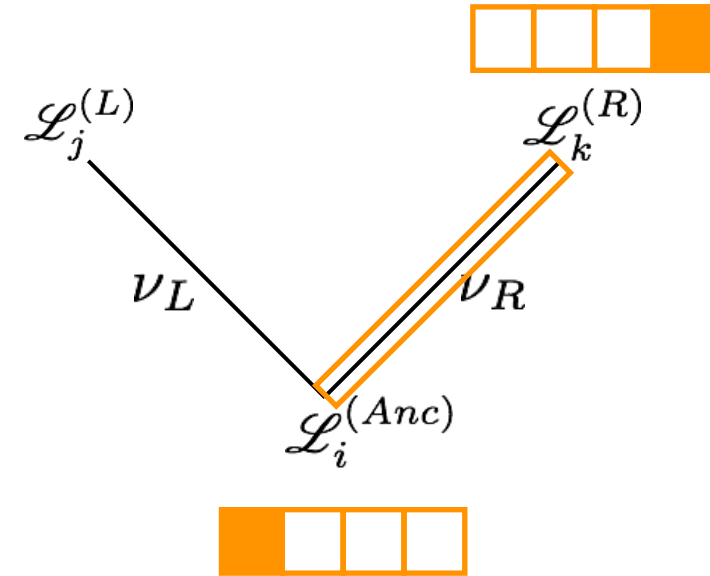
Next, we repeat the process for the right branch, ν_R.

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Next, we repeat the process for the right branch, ν_R .
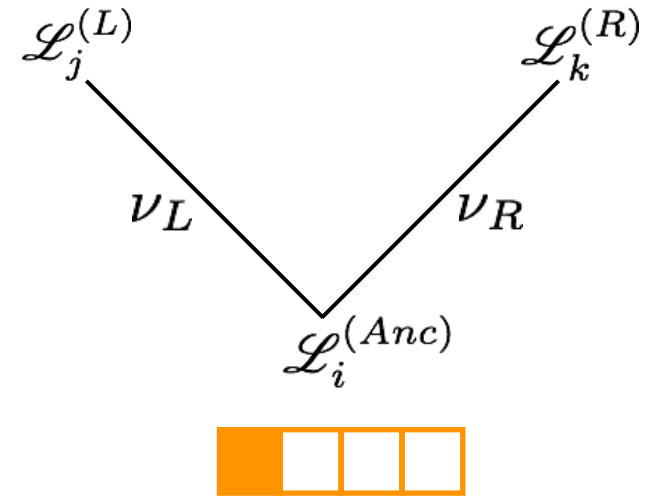

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Next, we repeat the process for the right branch, ν_R.

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm
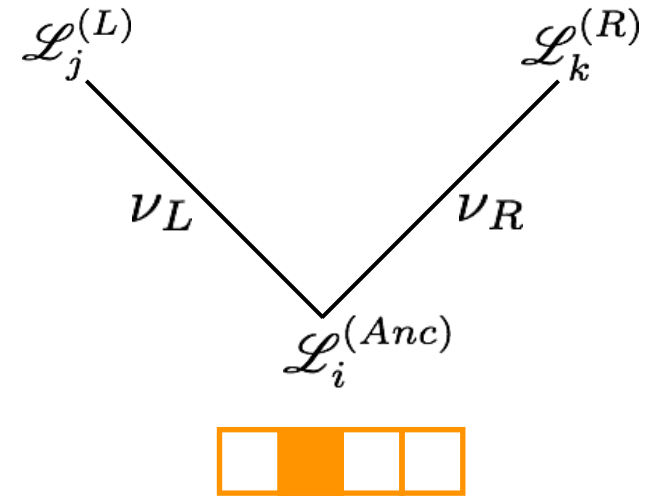
Next, we repeat the process for the right branch, ν_R.

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Finally, we calculate $\mathcal{L}_A^{(Anc)}$ as the product of the conditional likelihoods of the two descendant branches, which makes explicit the assumption that substitutions along these two lineages are independent.
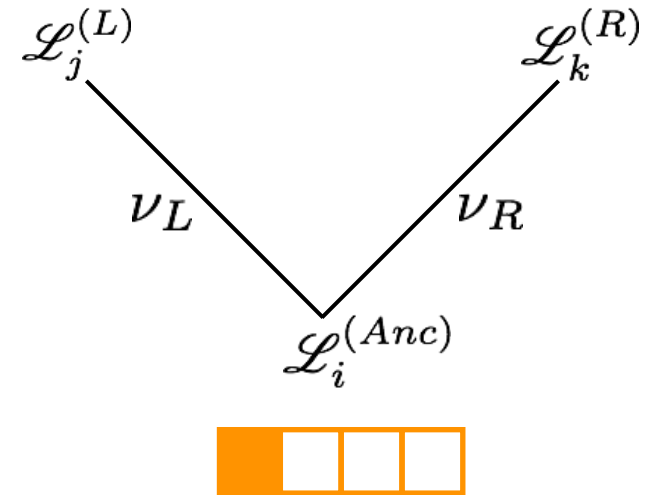
$$\mathcal{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathcal{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathcal{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Now we have computed $\mathcal{L}_A^{(Anc)}$ …

$$\mathcal{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathcal{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathcal{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

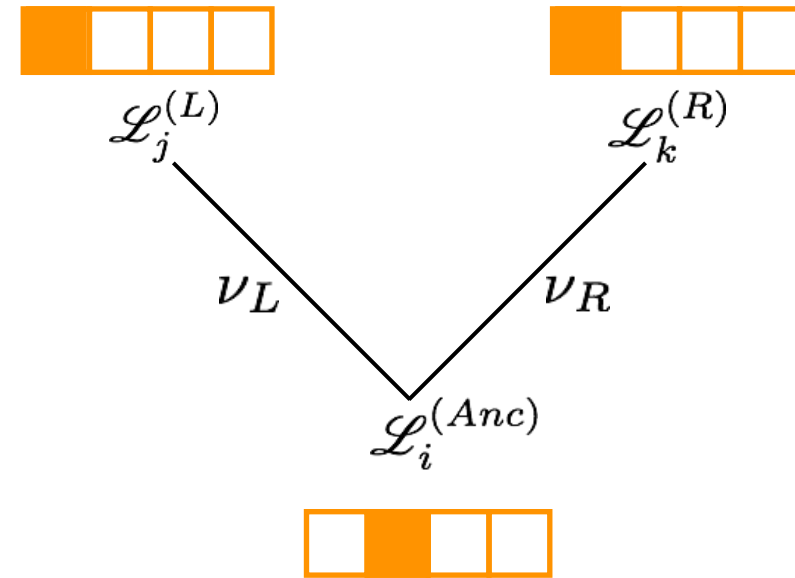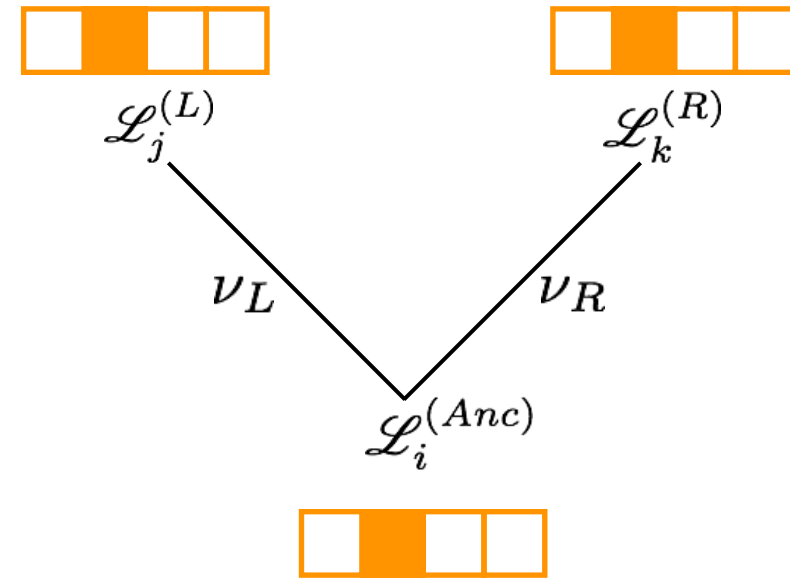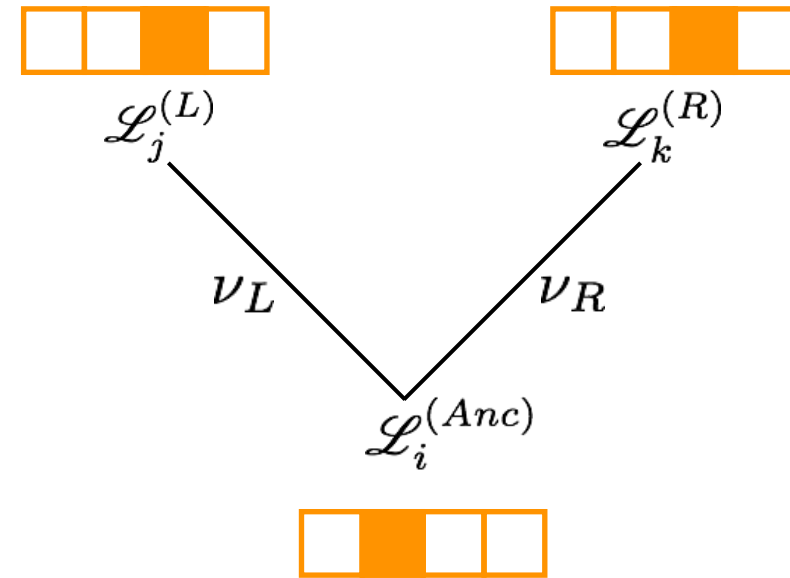So, now we need to repeat the entire process for $i = C$ to compute $\mathscr{L}_C^{(Anc)}$ …

$$\mathscr{L}_j^{(L)} \qquad\qquad \mathscr{L}_k^{(R)}$$

$$\nu_L \qquad\qquad \nu_R$$

$$\mathscr{L}_i^{(Anc)}$$

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, now we need to repeat the entire process for $i = C$ to compute $\mathscr{L}_C^{(Anc)}$ ...

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

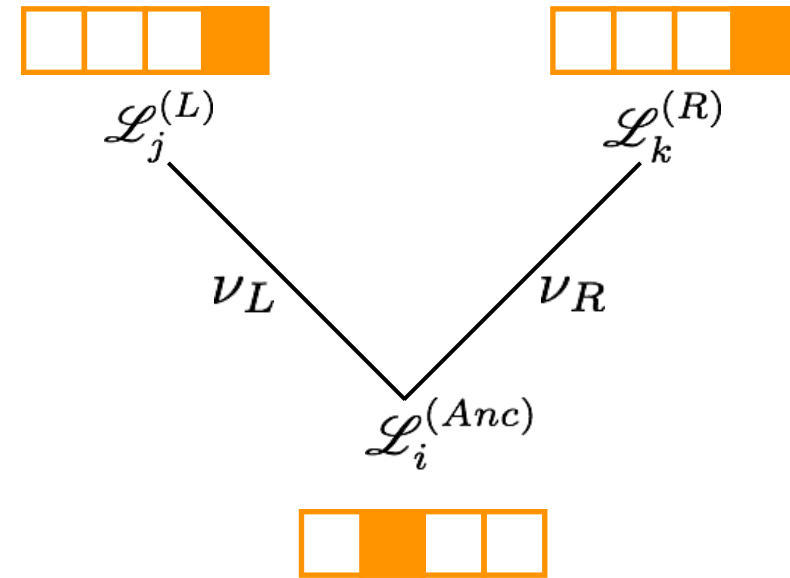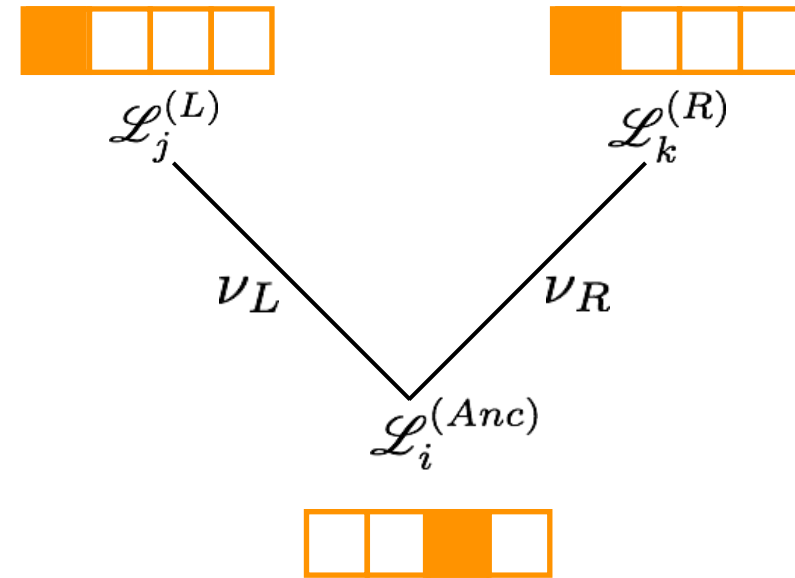So, now we need to repeat the entire process for $i = C$ to compute $\mathscr{L}_C^{(Anc)}$ ...

$$\mathscr{L}_j^{(L)} \qquad \mathscr{L}_k^{(R)}$$

$$\nu_L \qquad \nu_R$$

$$\mathscr{L}_i^{(Anc)}$$

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, now we need to repeat the entire process for $i = \text{C}$ to compute $\mathscr{L}_C^{(Anc)}$ …

$\mathscr{L}_j^{(L)}$

$\mathscr{L}_k^{(R)}$

ν_L

ν_R

$\mathscr{L}_i^{(Anc)}$

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (\text{A,C,G,T})} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (\text{A,C,G,T})} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

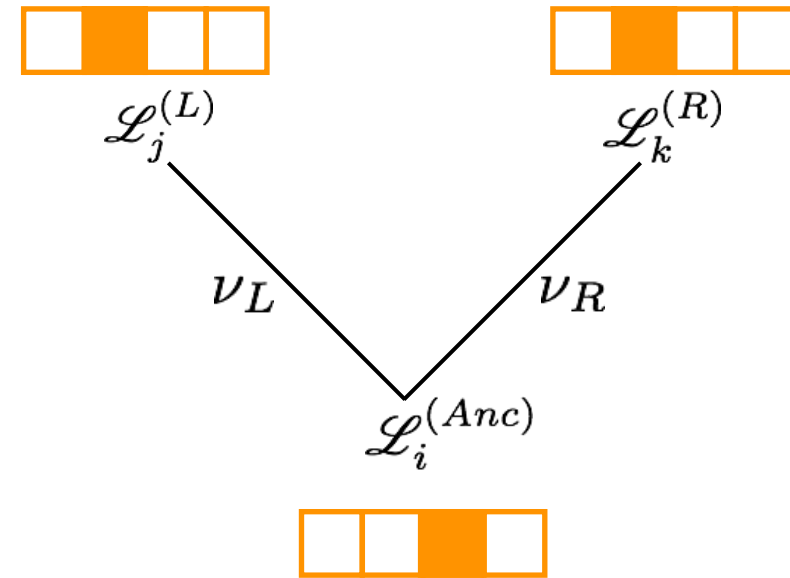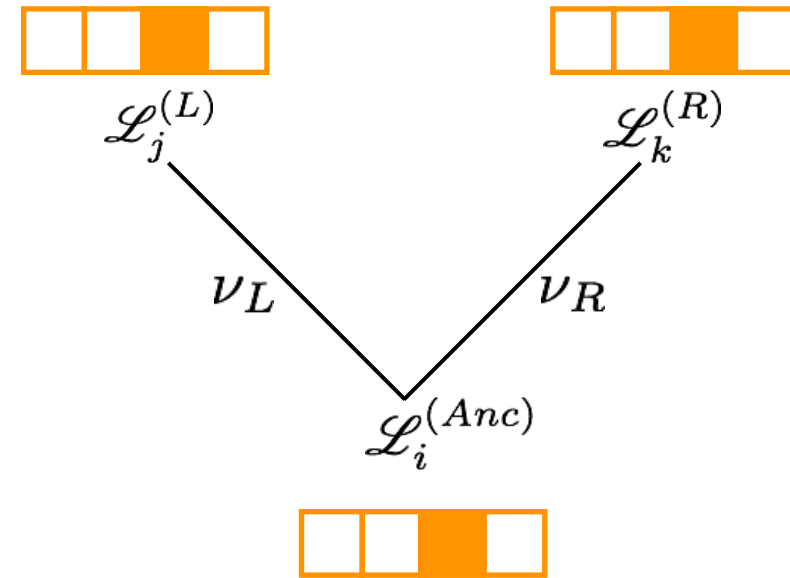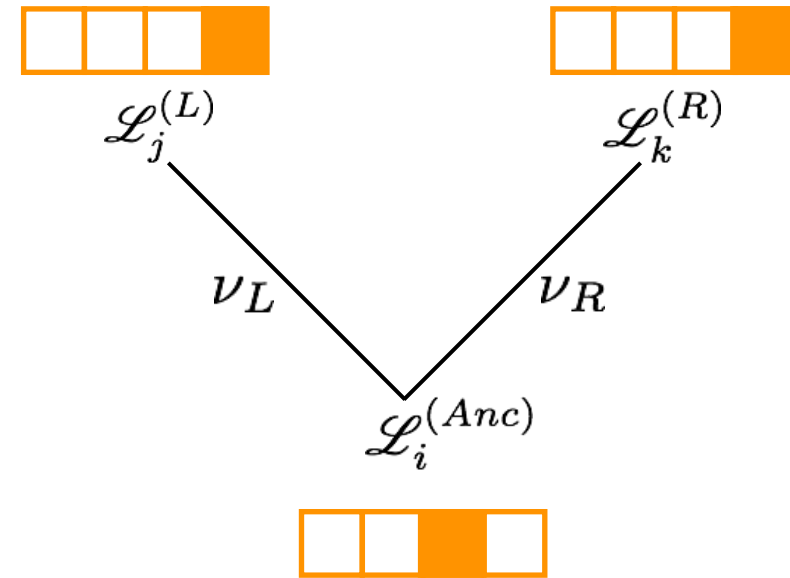So, now we need to repeat the entire process for $i = C$ to compute $\mathscr{L}_C^{(Anc)}$ …

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, now we need to repeat the entire process
for $i = \mathrm{C}$ to compute $\mathscr{L}_C^{(Anc)}$
for $i = \mathrm{G}$ to compute $\mathscr{L}_G^{(Anc)}$ …

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (\mathrm{A,C,G,T})} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (\mathrm{A,C,G,T})} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

$\mathscr{L}_j^{(L)}$ $\mathscr{L}_k^{(R)}$

ν_L ν_R

$\mathscr{L}_i^{(Anc)}$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, now we need to repeat the entire process
for $i = \mathrm{C}$ to compute $\mathscr{L}_C^{(Anc)}$
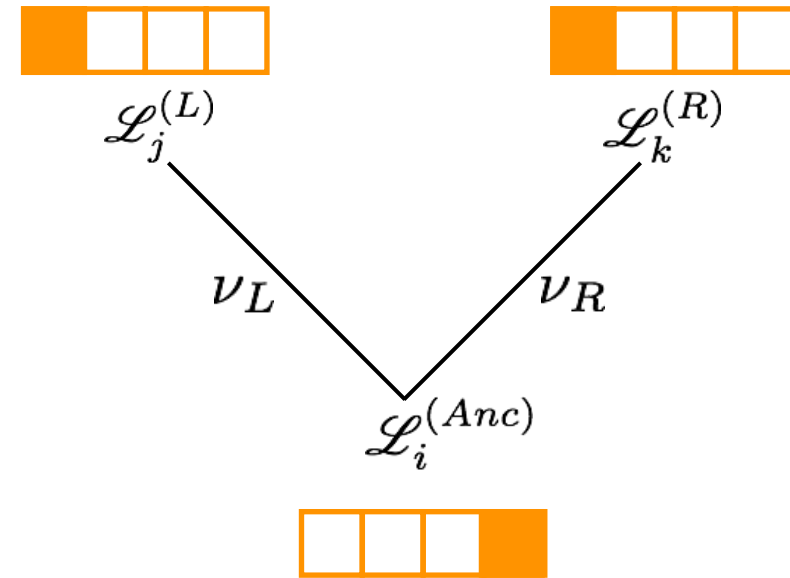for $i = \mathrm{G}$ to compute $\mathscr{L}_G^{(Anc)}$ …

$\mathscr{L}_j^{(L)}$ $\mathscr{L}_k^{(R)}$

ν_L ν_R

$\mathscr{L}_i^{(Anc)}$

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (\mathrm{A,C,G,T})} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (\mathrm{A,C,G,T})} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, now we need to repeat the entire process
for $i = \mathrm{C}$ to compute $\mathscr{L}_C^{(Anc)}$
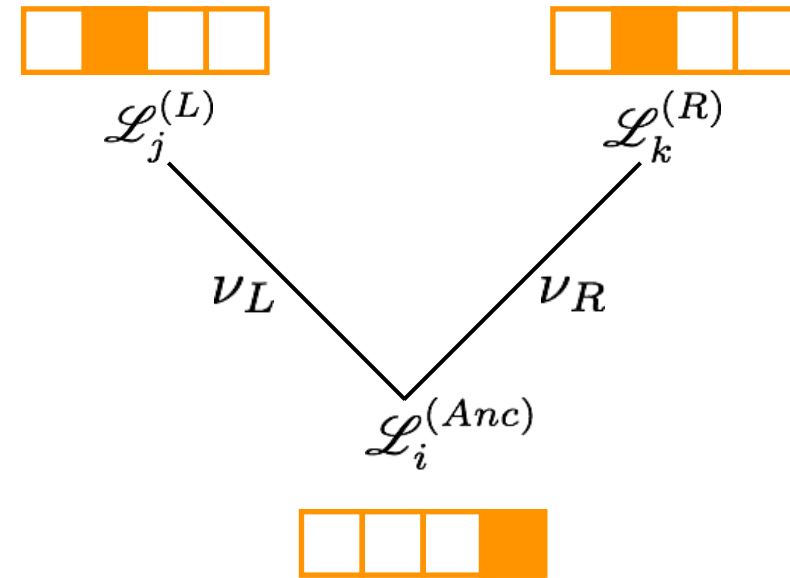for $i = \mathrm{G}$ to compute $\mathscr{L}_G^{(Anc)}$ …

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (\mathrm{A,C,G,T})} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (\mathrm{A,C,G,T})} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, now we need to repeat the entire process
for $i = \text{C}$ to compute $\mathscr{L}_C^{(Anc)}$
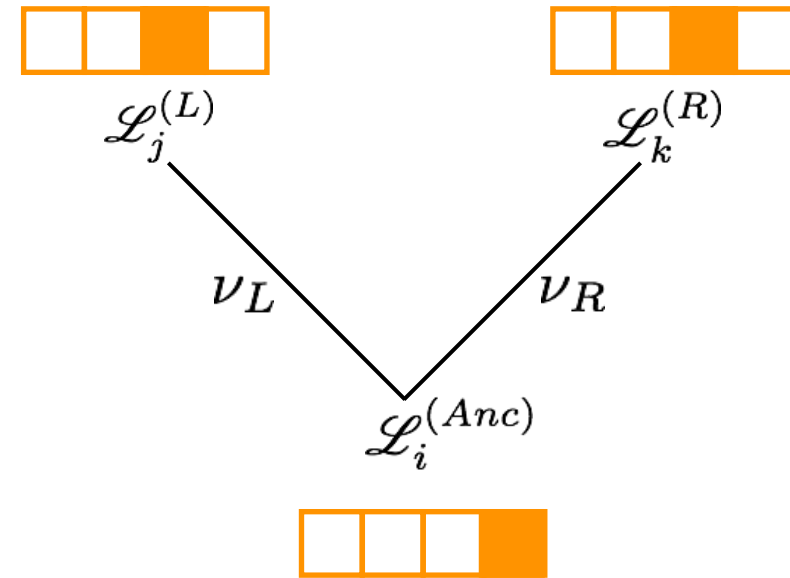for $i = \text{G}$ to compute $\mathscr{L}_G^{(Anc)}$ …

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j\in(\text{A,C,G,T})} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k\in(\text{A,C,G,T})} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, now we need to repeat the entire process
for $i = \mathrm{C}$ to compute $\mathscr{L}_C^{(Anc)}$
for $i = \mathrm{G}$ to compute $\mathscr{L}_G^{(Anc)}$
and finally for $i = \mathrm{T}$ to compute $\mathscr{L}_T^{(Anc)} \ldots$

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (\mathrm{A,C,G,T})} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (\mathrm{A,C,G,T})} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, now we need to repeat the entire process
for $i = \text{C}$ to compute $\mathscr{L}_C^{(Anc)}$
for $i = \text{G}$ to compute $\mathscr{L}_G^{(Anc)}$
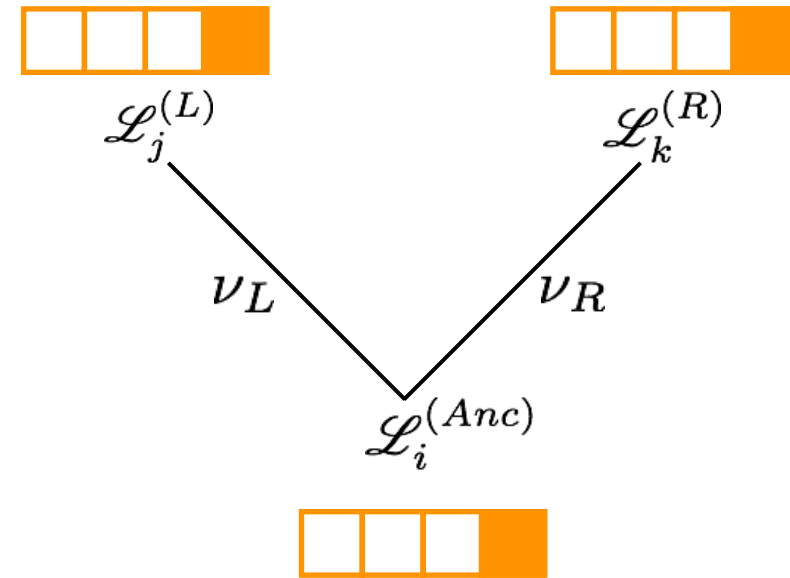and finally for $i = \text{T}$ to compute $\mathscr{L}_T^{(Anc)}$ ...

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (\text{A,C,G,T})} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (\text{A,C,G,T})} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, now we need to repeat the entire process
for $i = \mathrm{C}$ to compute $\mathscr{L}_C^{(Anc)}$
for $i = \mathrm{G}$ to compute $\mathscr{L}_G^{(Anc)}$
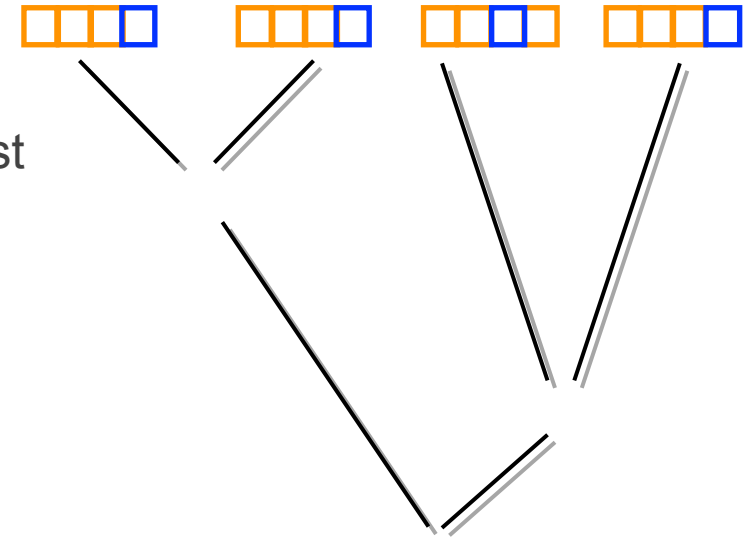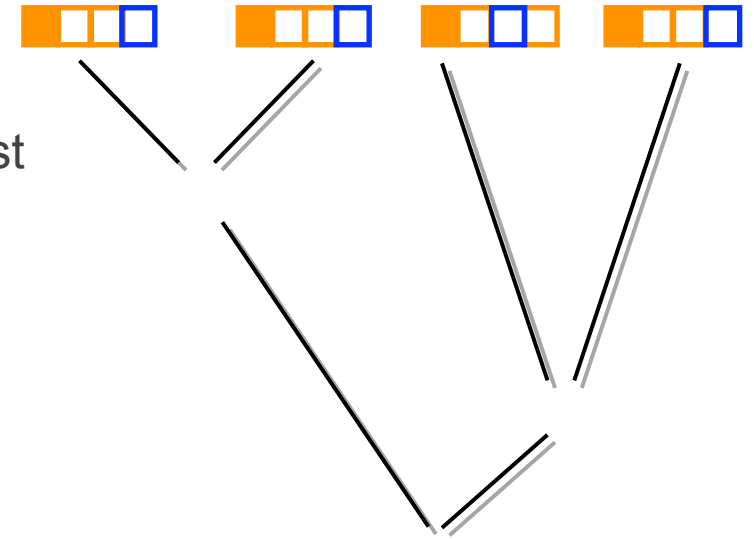and finally for $i = \mathrm{T}$ to compute $\mathscr{L}_T^{(Anc)}$ …

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

So, now we need to repeat the entire process
for $i = C$ to compute $\mathscr{L}_C^{(Anc)}$
for $i = G$ to compute $\mathscr{L}_G^{(Anc)}$
and finally for $i = T$ to compute $\mathscr{L}_T^{(Anc)}$ …

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded...
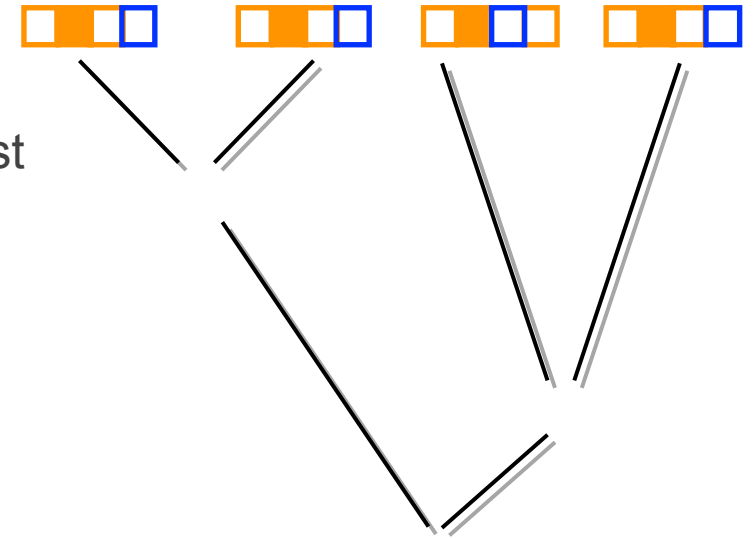
$$\mathcal{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathcal{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathcal{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded...
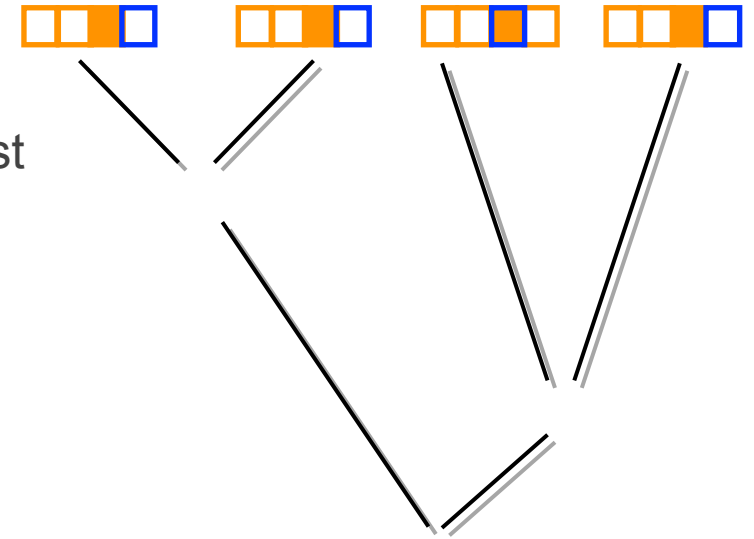
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded...
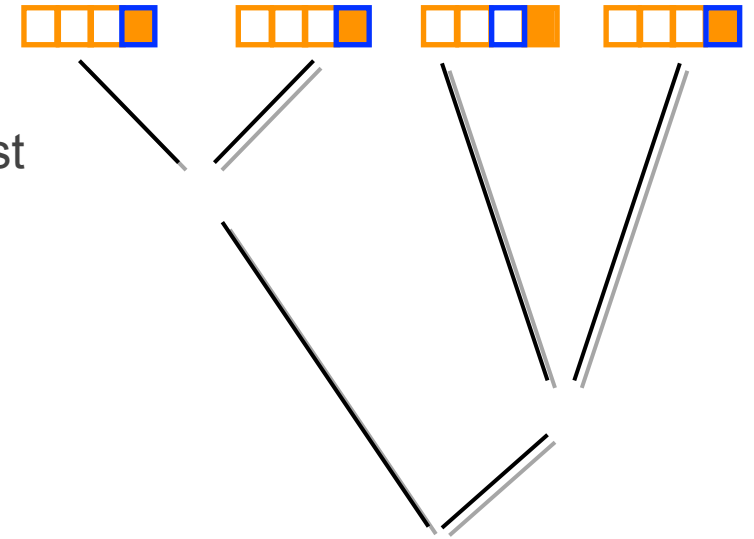

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded...
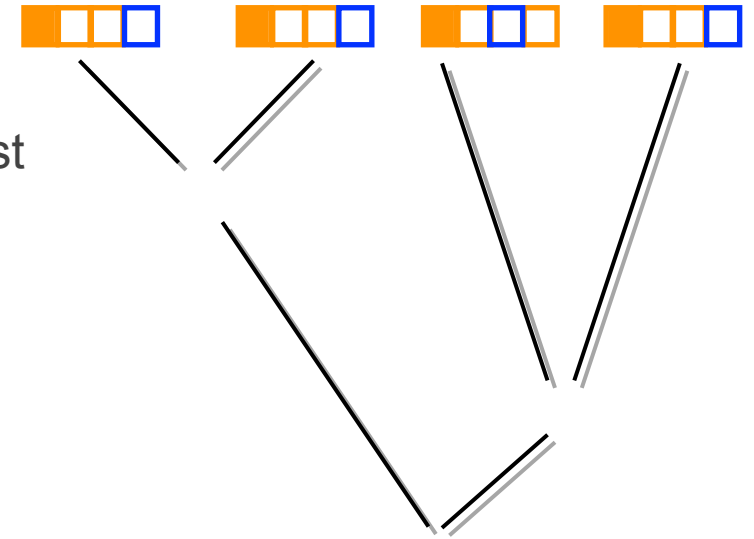

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded...
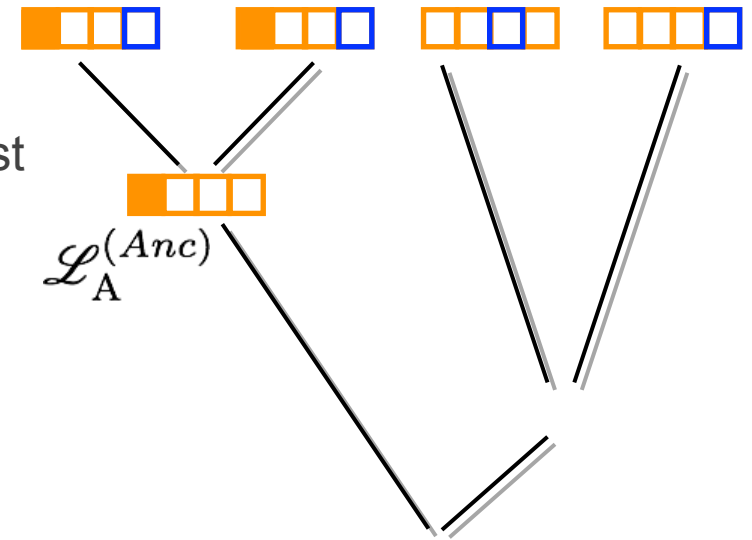

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded...

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
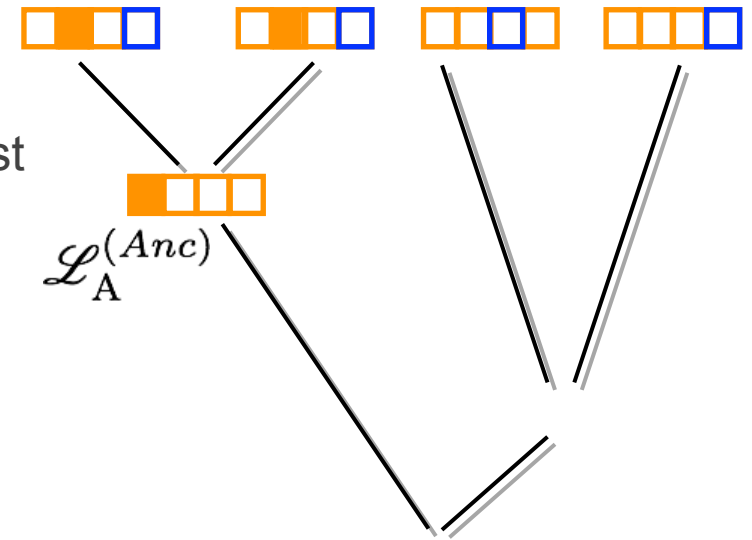

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
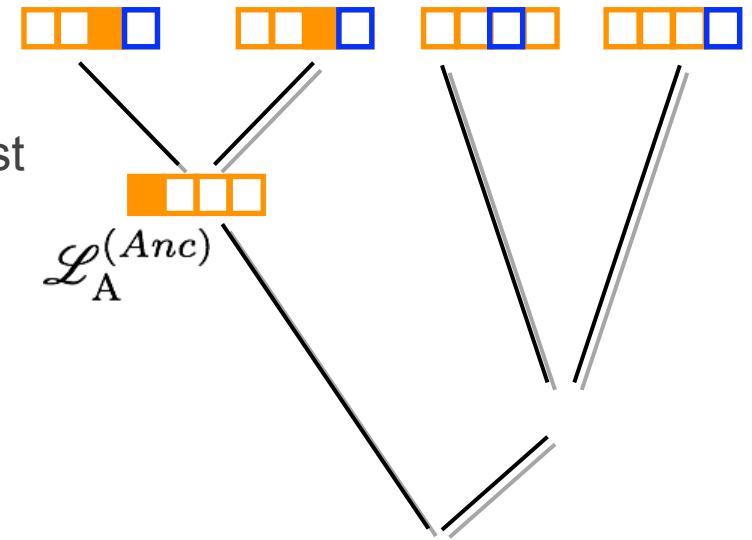
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

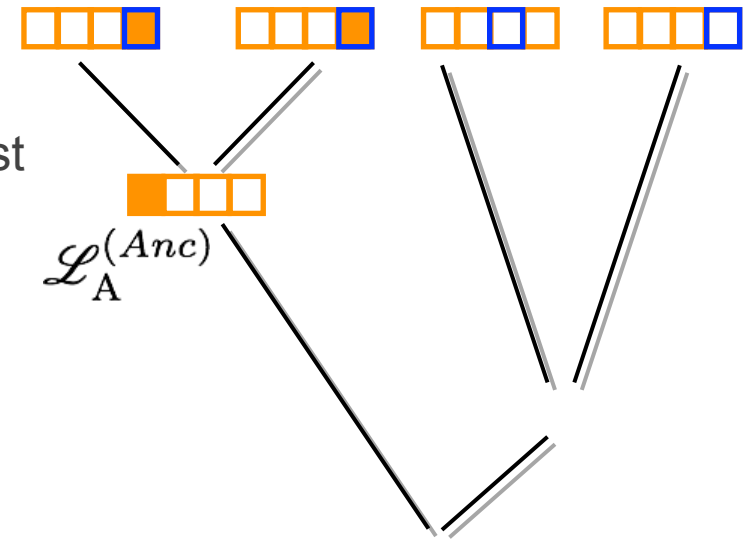recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

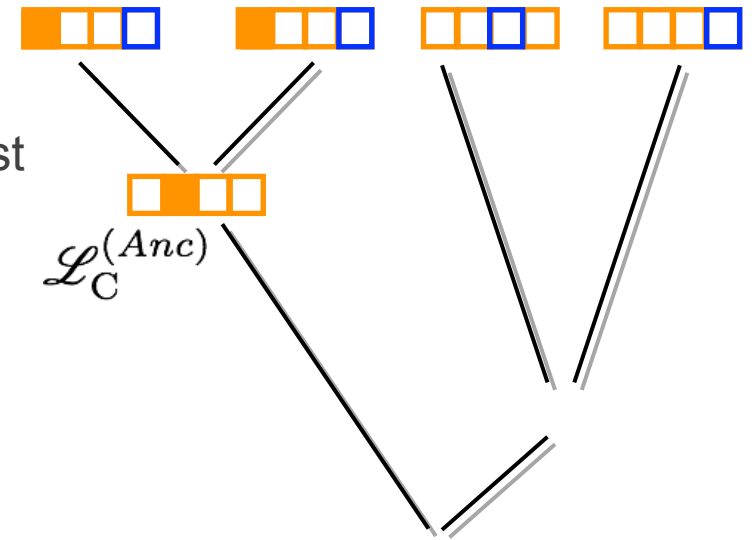recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

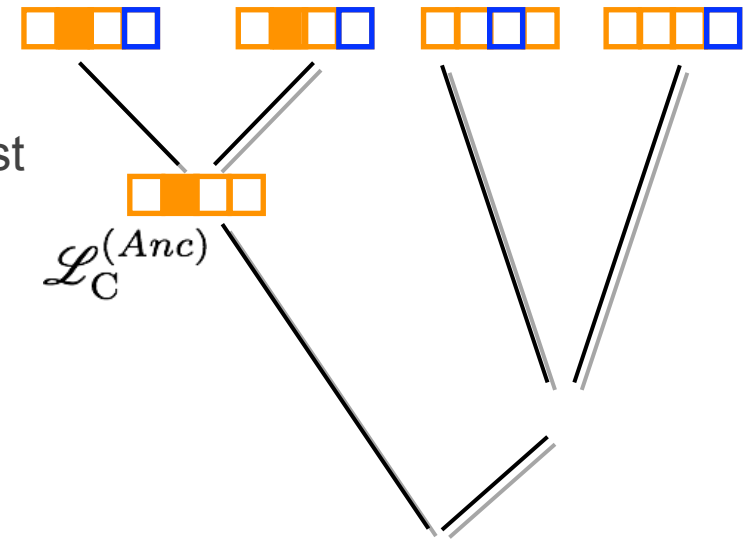recorded, then we loop over the internal nodes in a post-order traversal…

$\mathscr{L}_C^{(Anc)}$

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

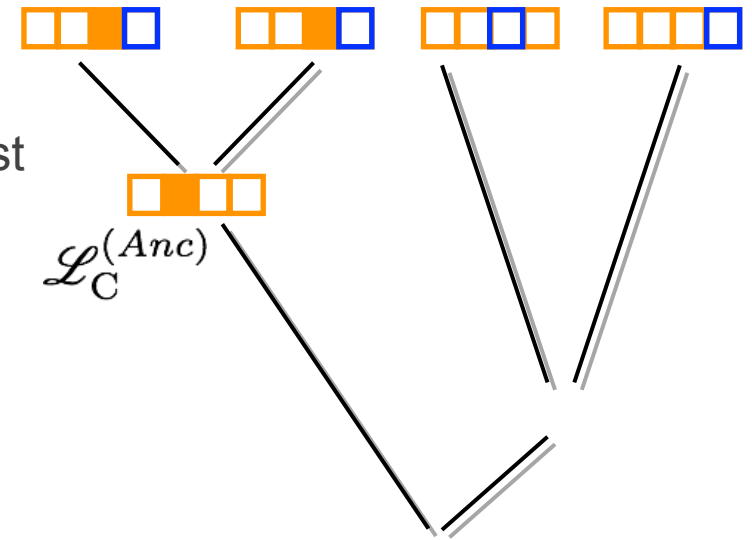recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

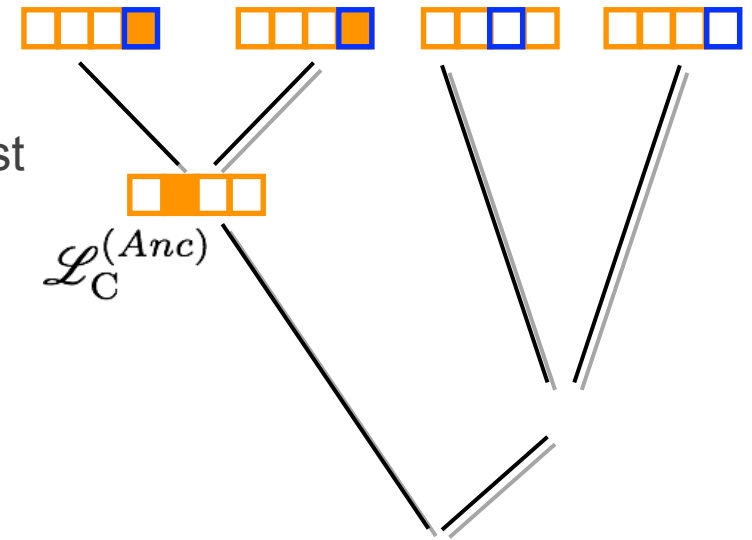recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

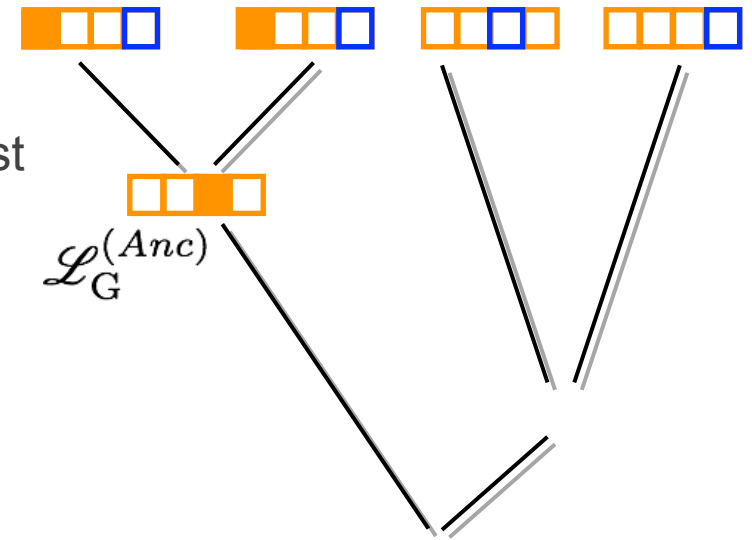recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (\text{A,C,G,T})} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (\text{A,C,G,T})} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

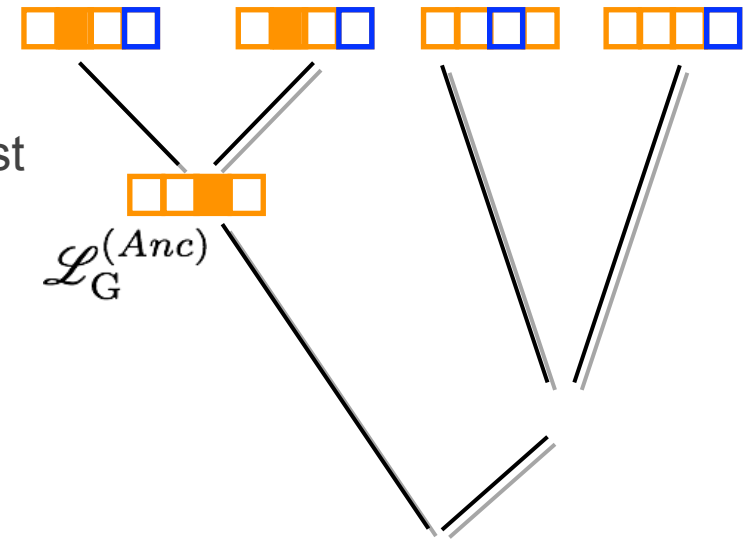recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

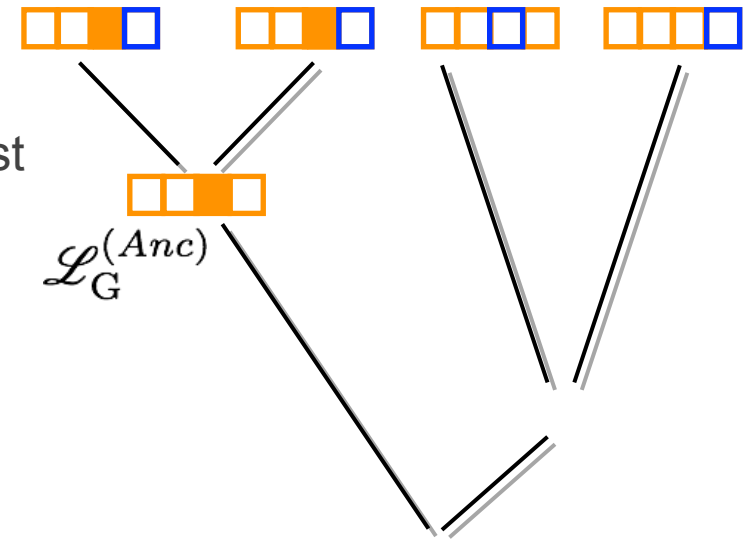recorded, then we loop over the internal nodes in a post-order traversal…

$\mathscr{L}_G^{(Anc)}$

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

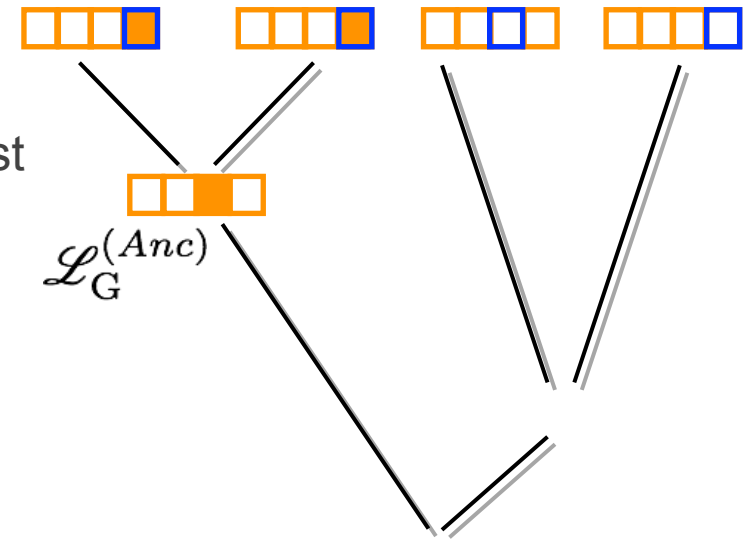recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

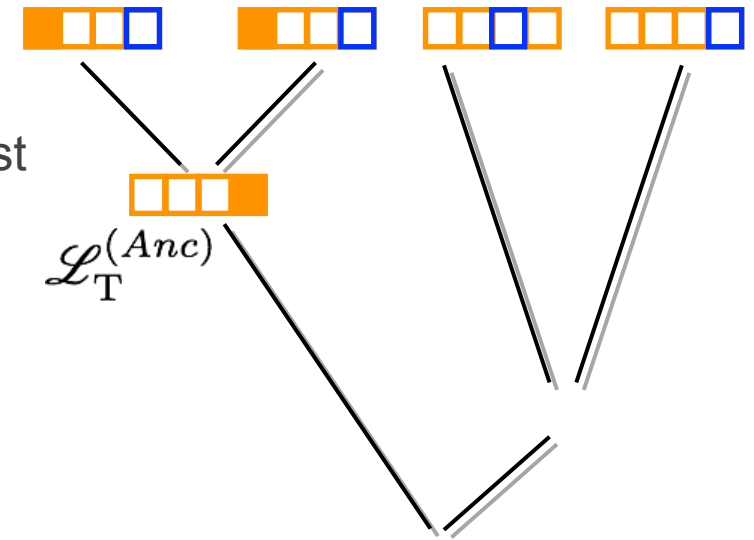recorded, then we loop over the internal nodes in a post-order traversal…

$\mathscr{L}_G^{(Anc)}$

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

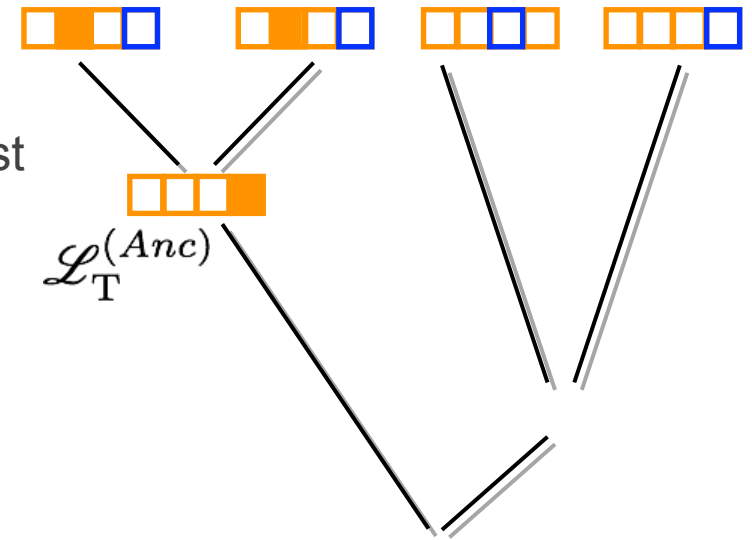recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j\in(A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k\in(A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (\mathrm{A,C,G,T})} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (\mathrm{A,C,G,T})} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

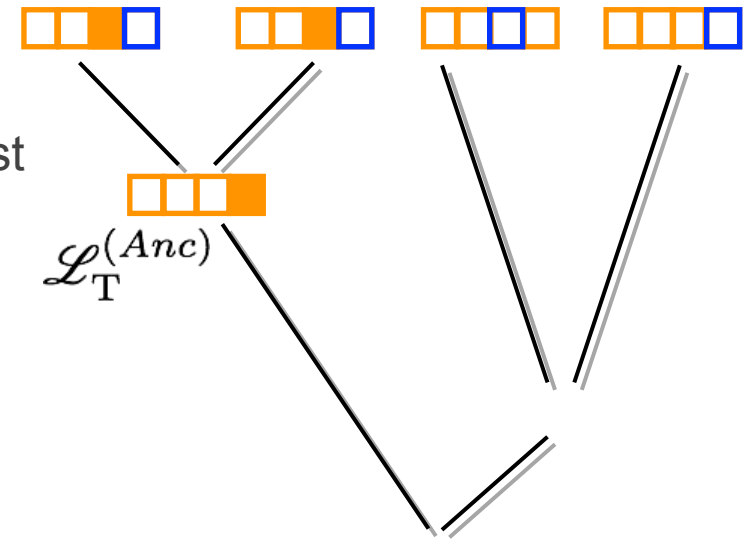recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first

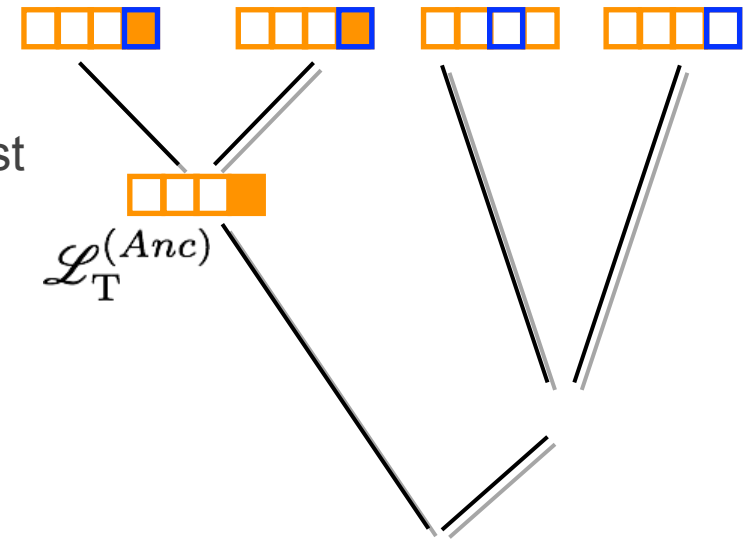recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
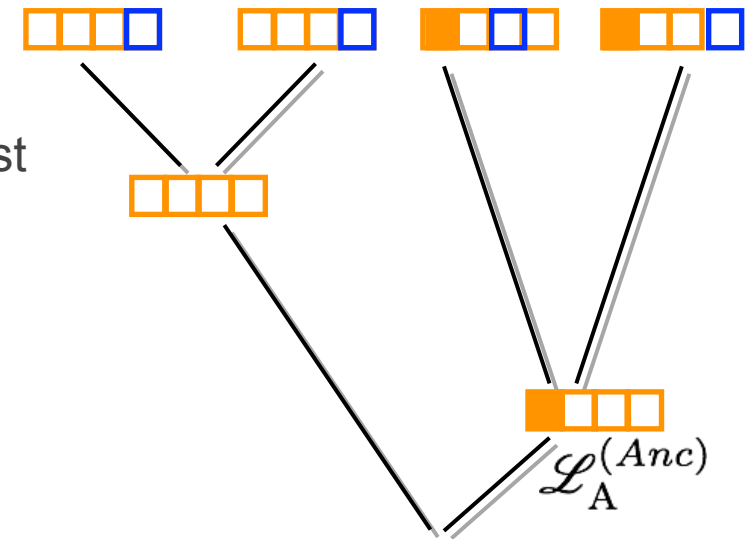

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
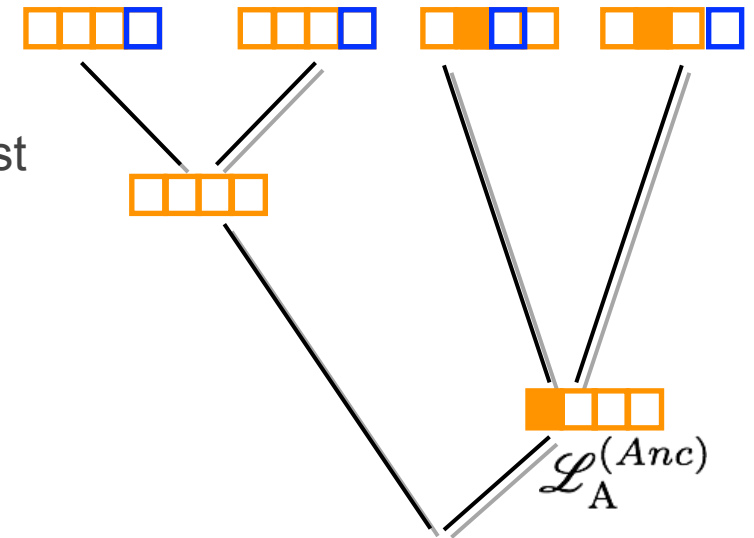
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
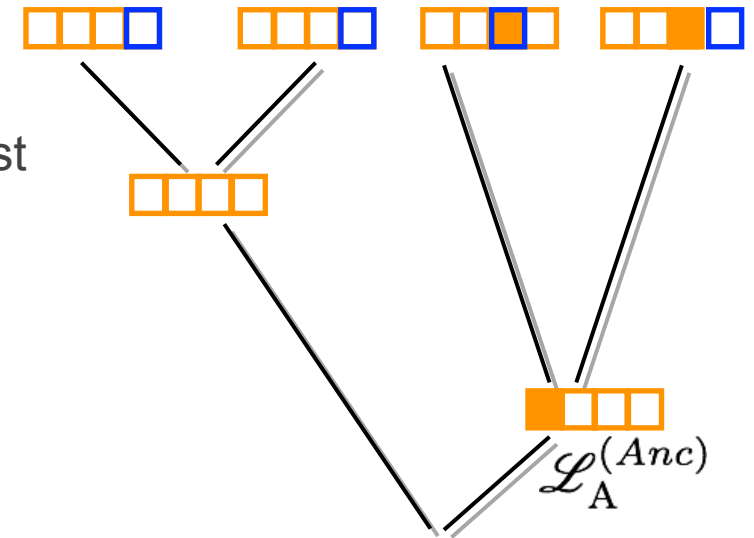
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
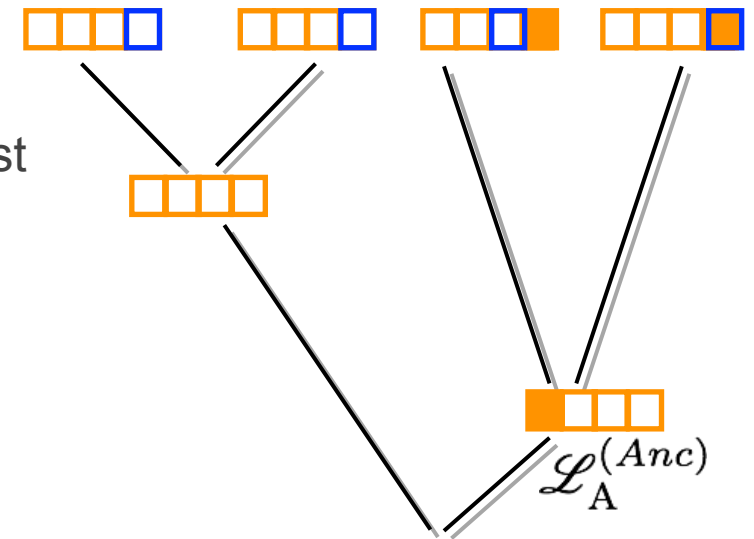
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
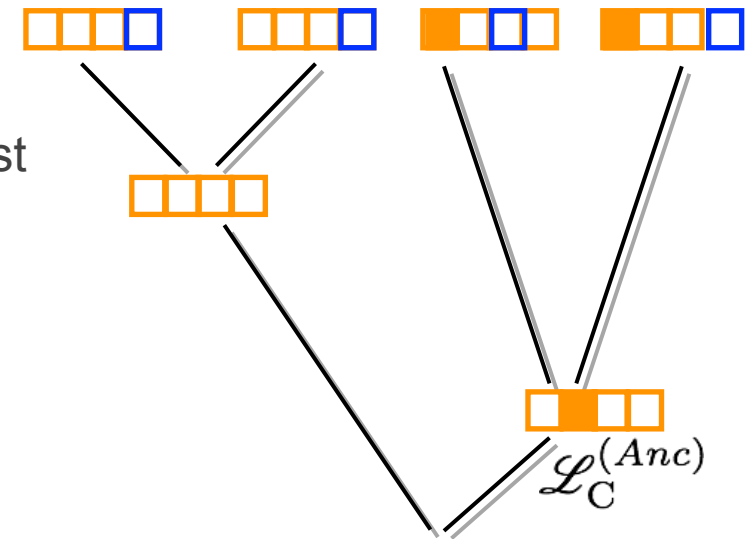
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
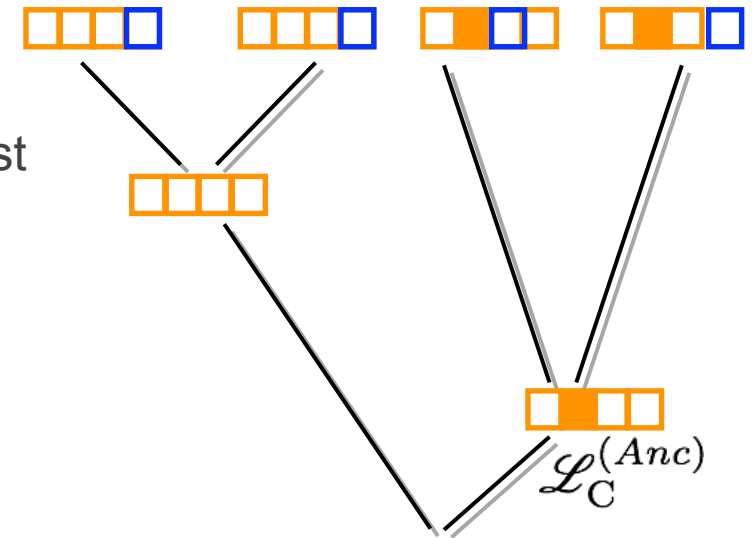
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
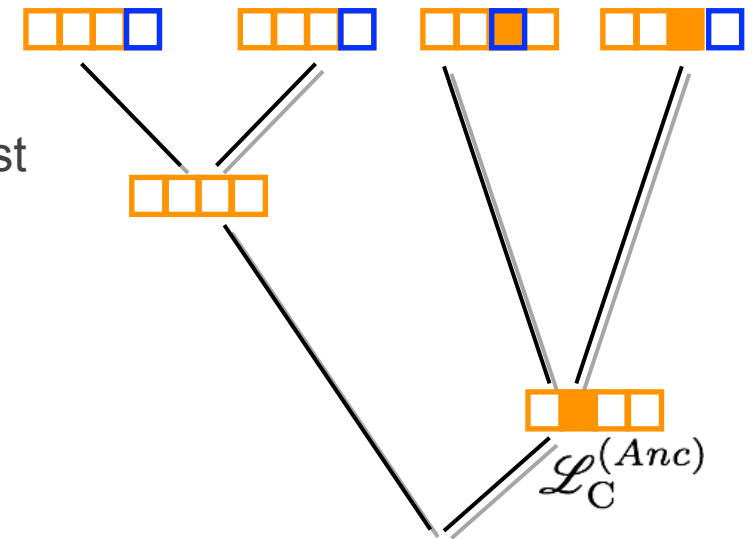
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
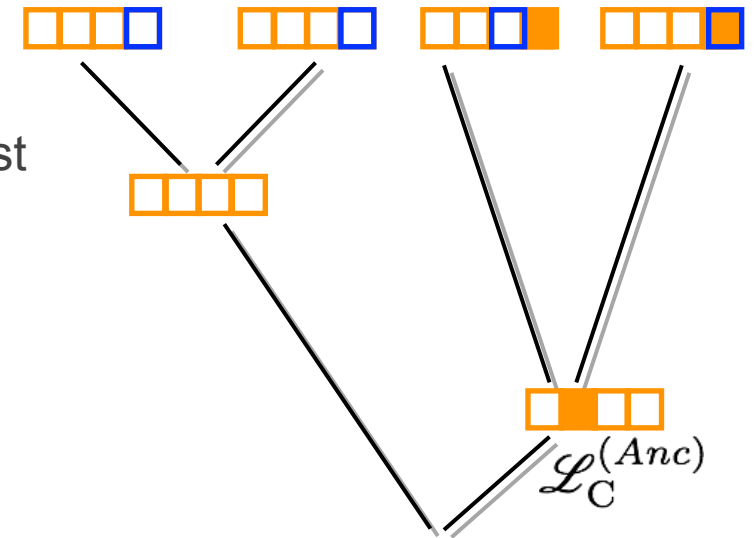

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…

$$
\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)
$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
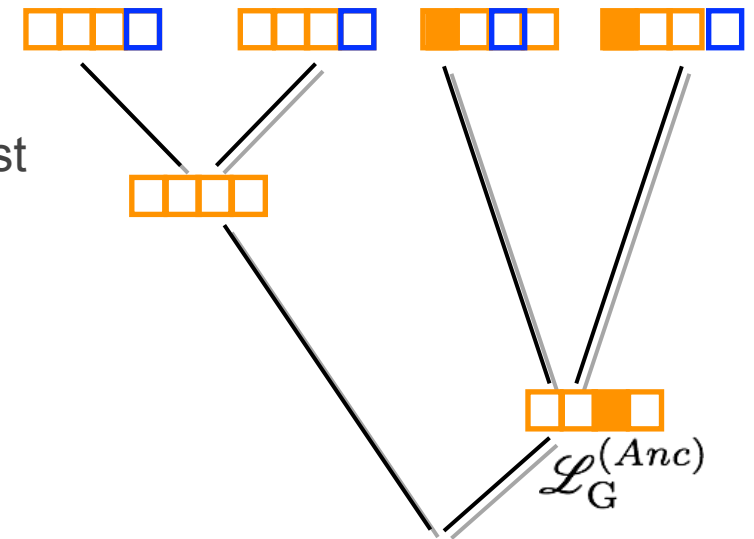
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j\in(A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k\in(A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
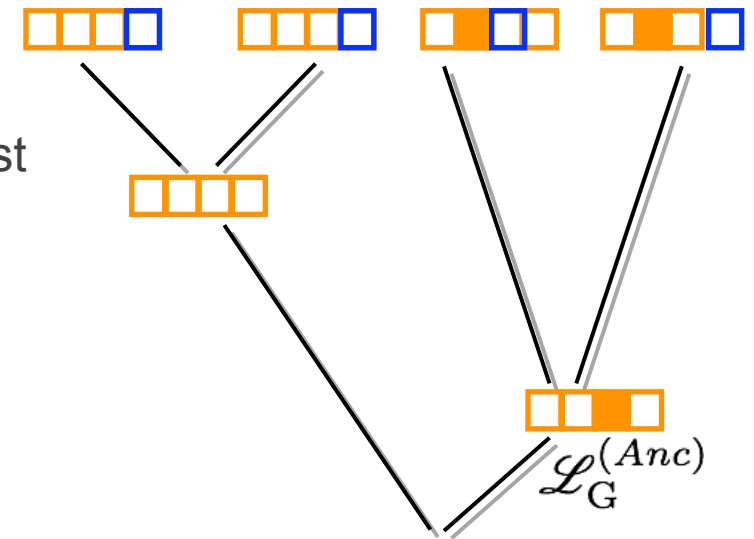

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
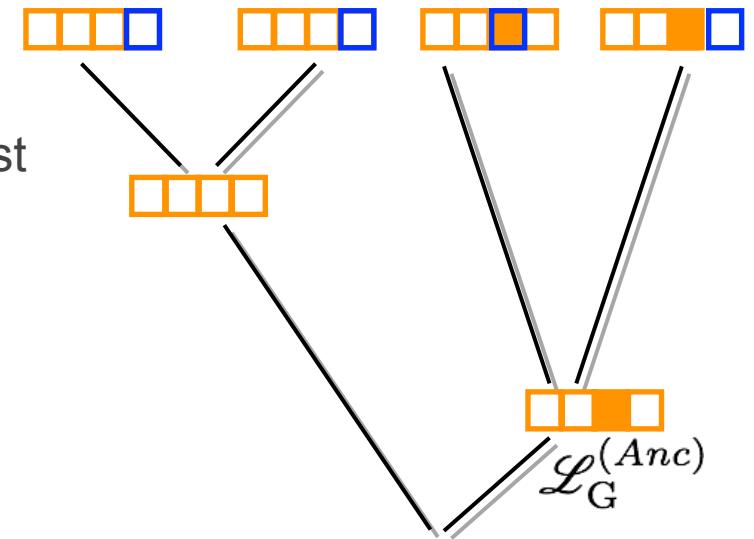

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
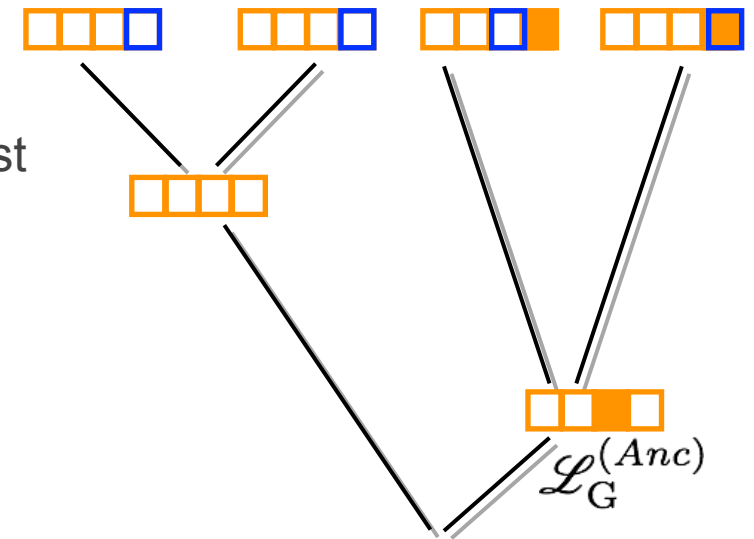

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
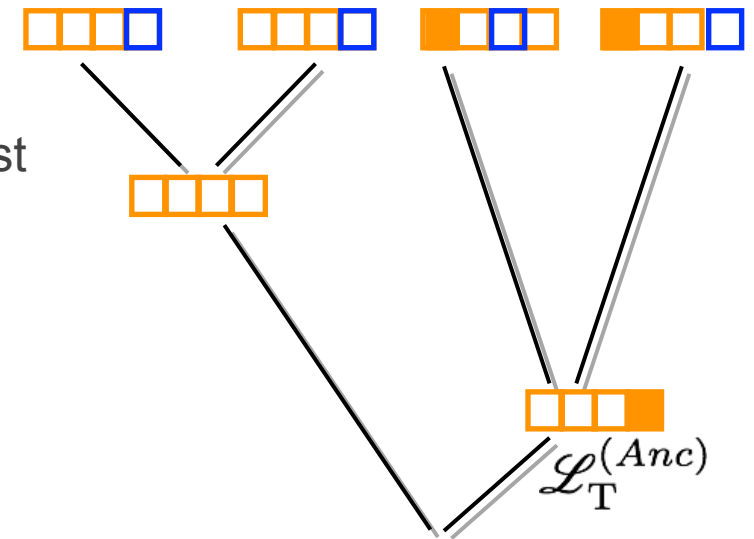

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
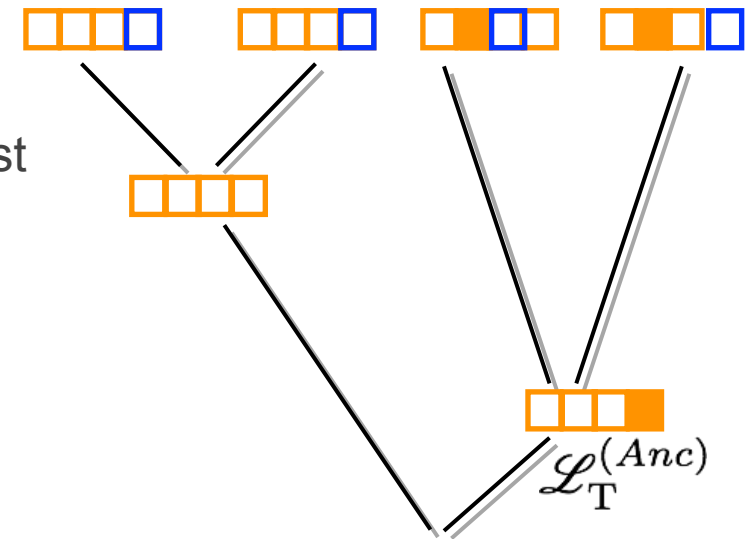
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
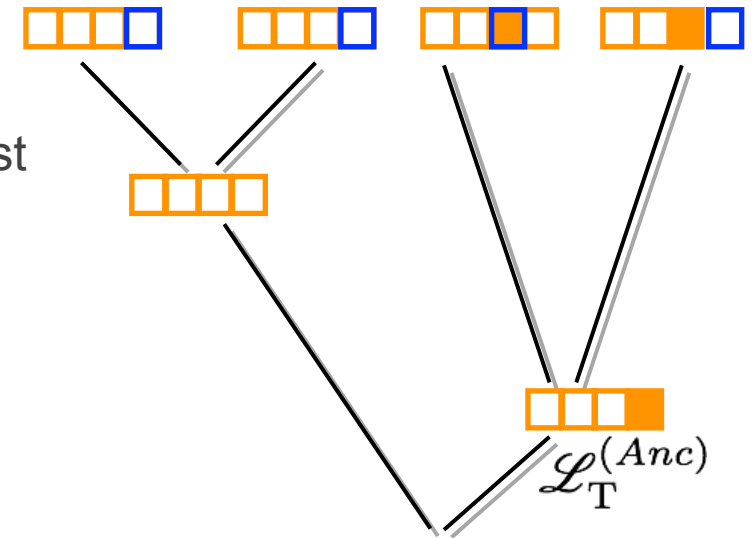

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
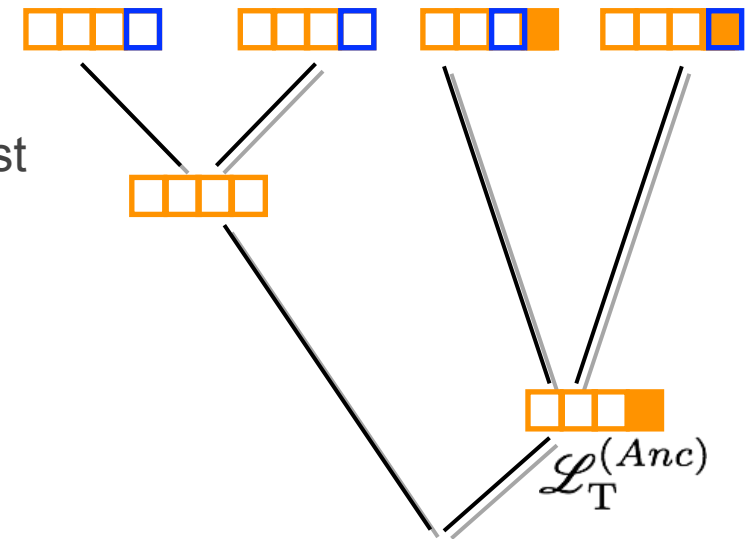
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

$$\mathscr{L}_A^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
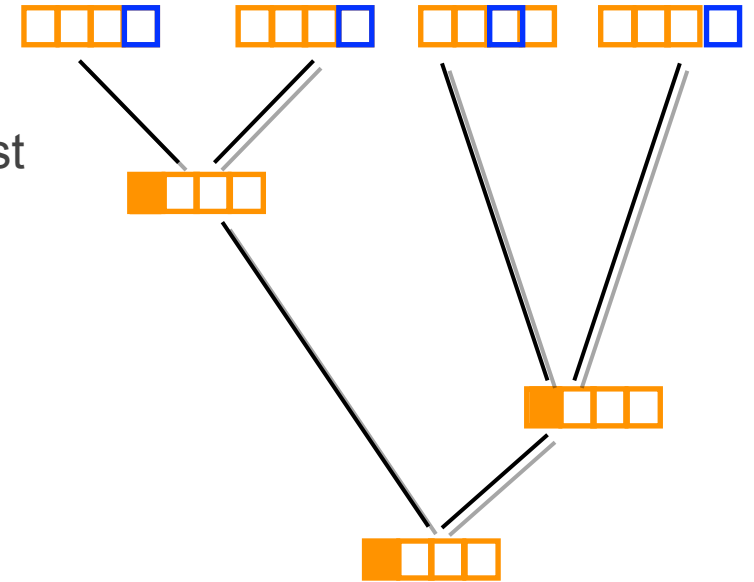
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

$$\mathscr{L}_A^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
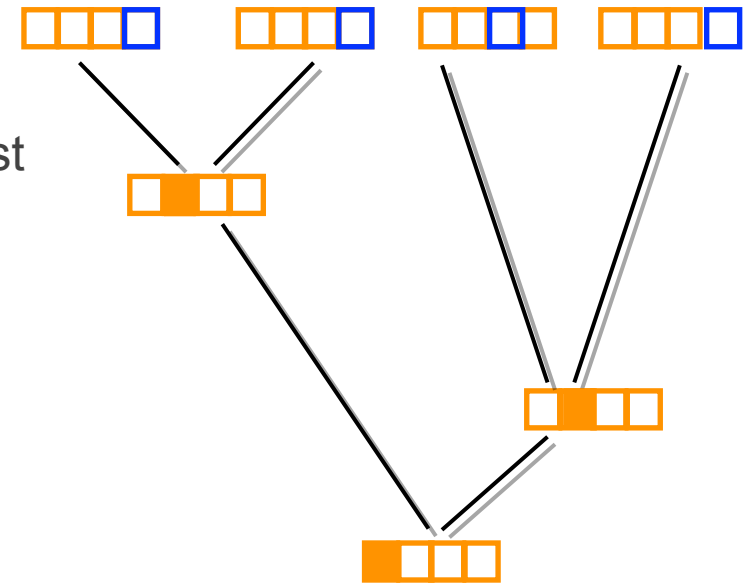
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

$$\mathscr{L}_A^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
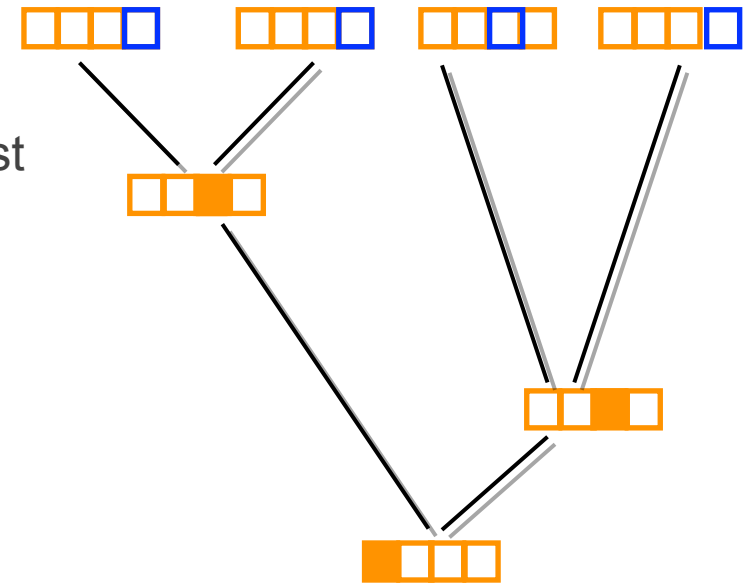

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

$$\mathscr{L}_A^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
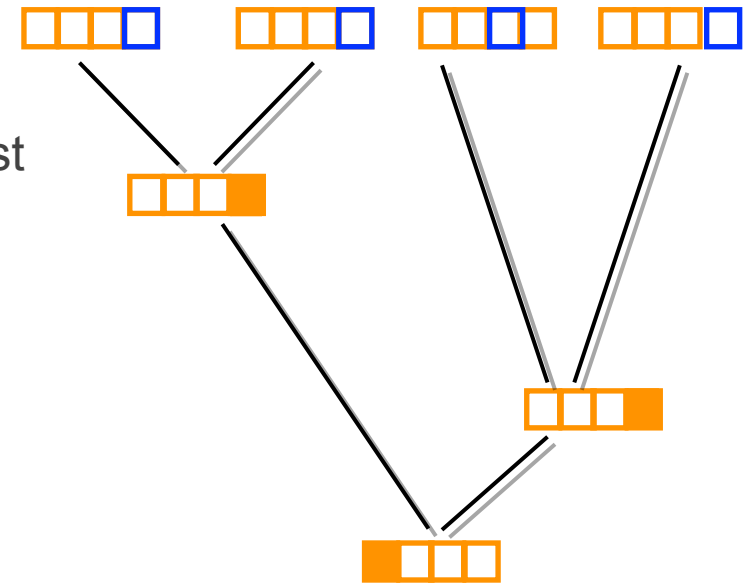
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

$$\mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
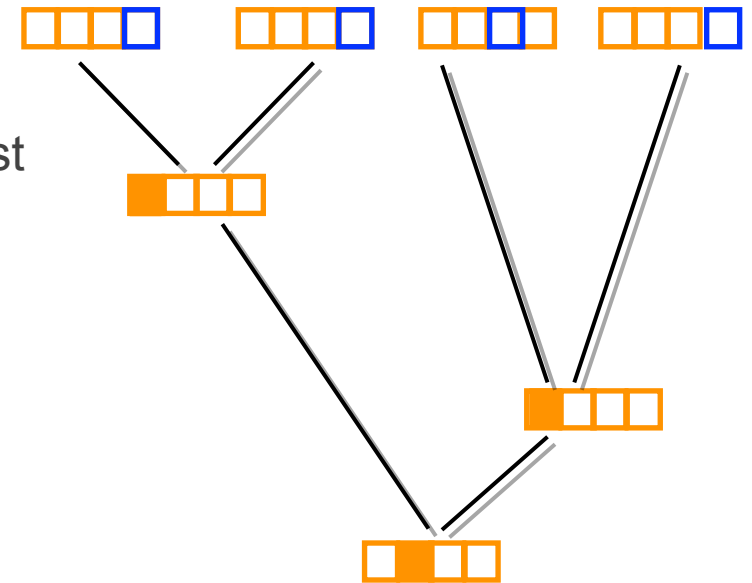

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

$$\mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$
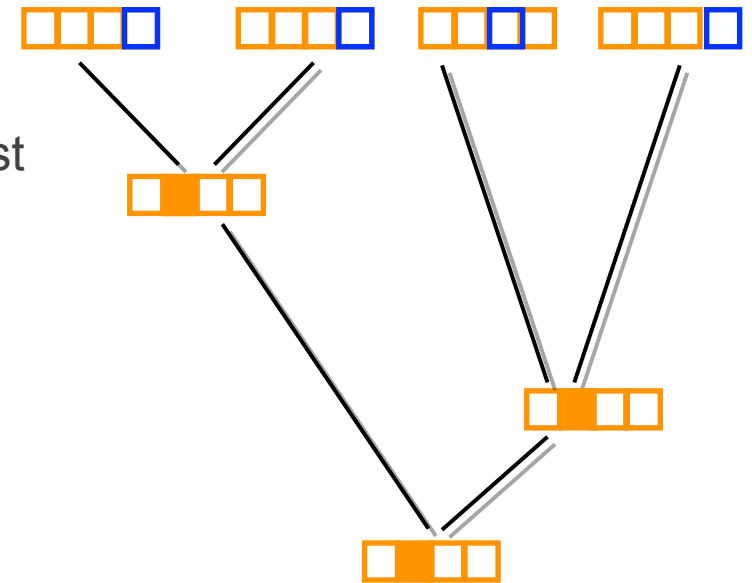
$$\mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
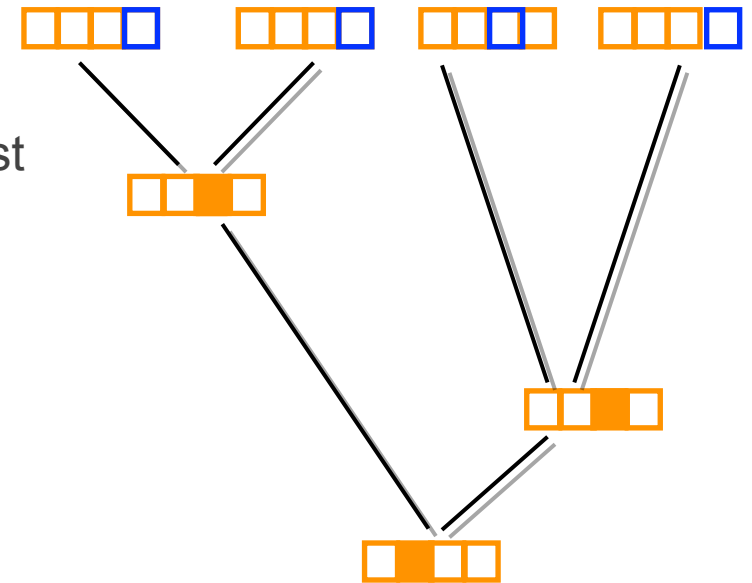
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

$$\mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j\in(A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k\in(A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

$$\mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)} \qquad \mathscr{L}_G^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
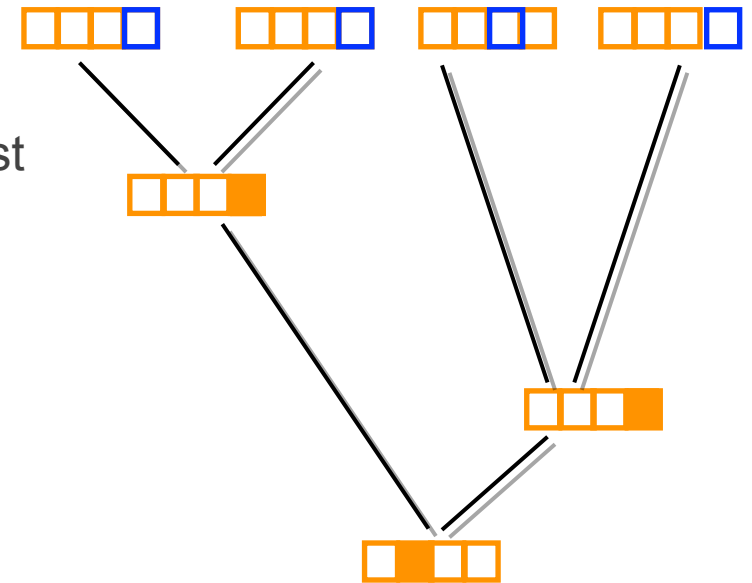

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

$$\mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)} \qquad \mathscr{L}_G^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…
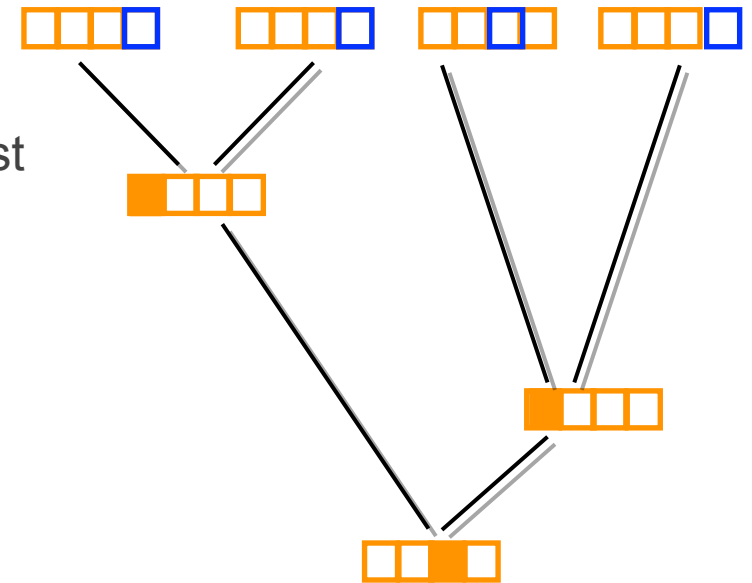
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$
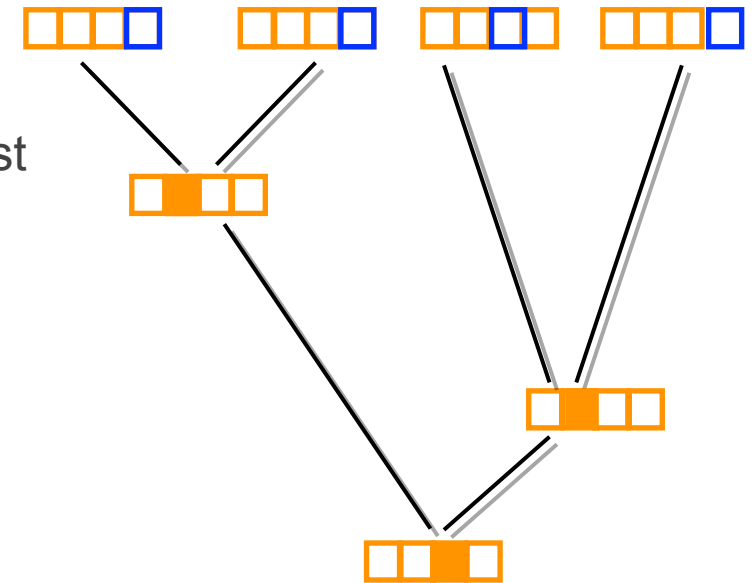
$$\mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)} \qquad \mathscr{L}_G^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$
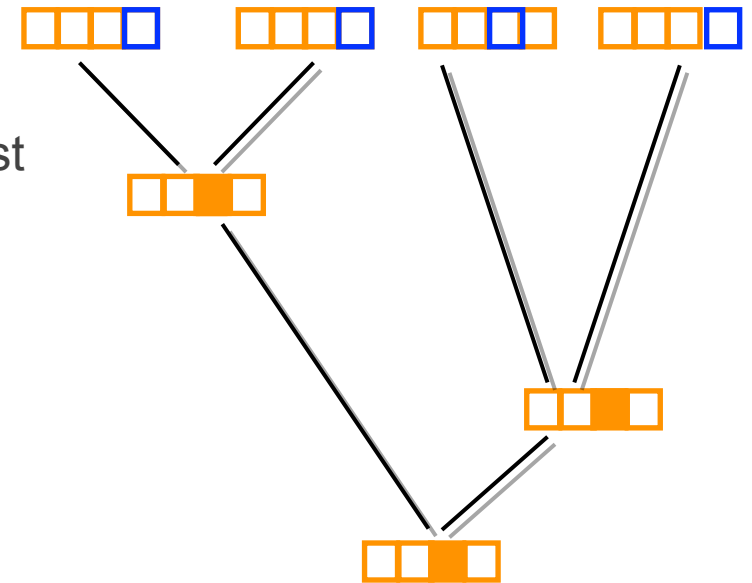
$$\mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)} \qquad \mathscr{L}_G^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$
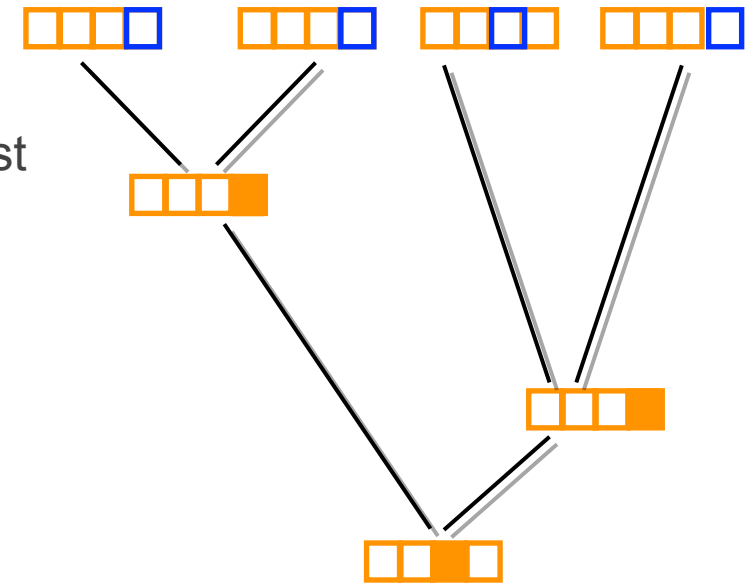
$$\mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)} \qquad \mathscr{L}_G^{(root)} \qquad \mathscr{L}_T^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$
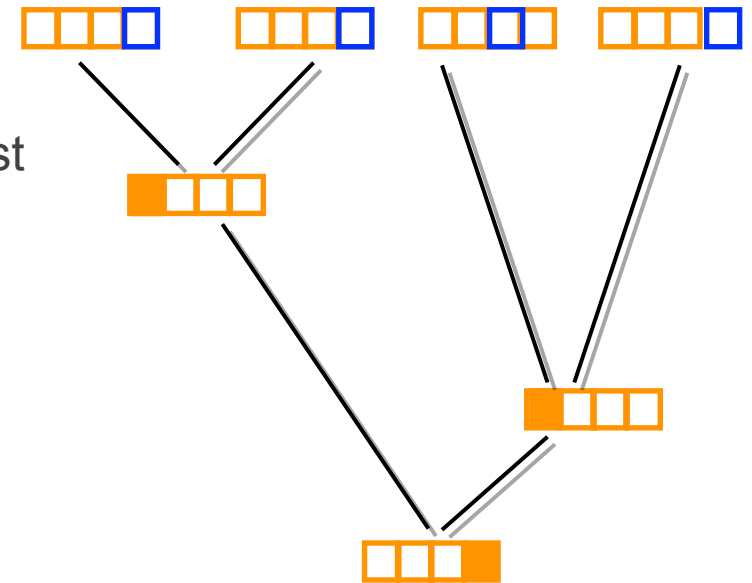
$$\mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)} \qquad \mathscr{L}_G^{(root)} \qquad \mathscr{L}_T^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

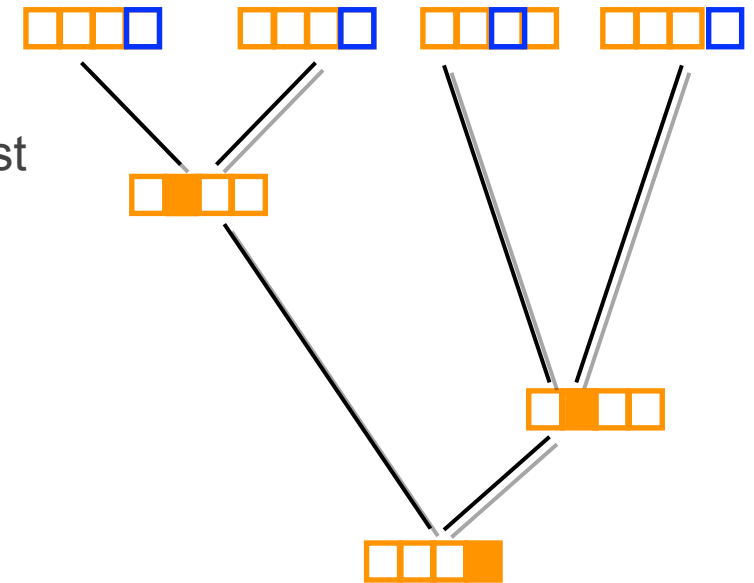$$\mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)} \qquad \mathscr{L}_G^{(root)} \qquad \mathscr{L}_T^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded, then we loop over the internal nodes in a post-order traversal…

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$
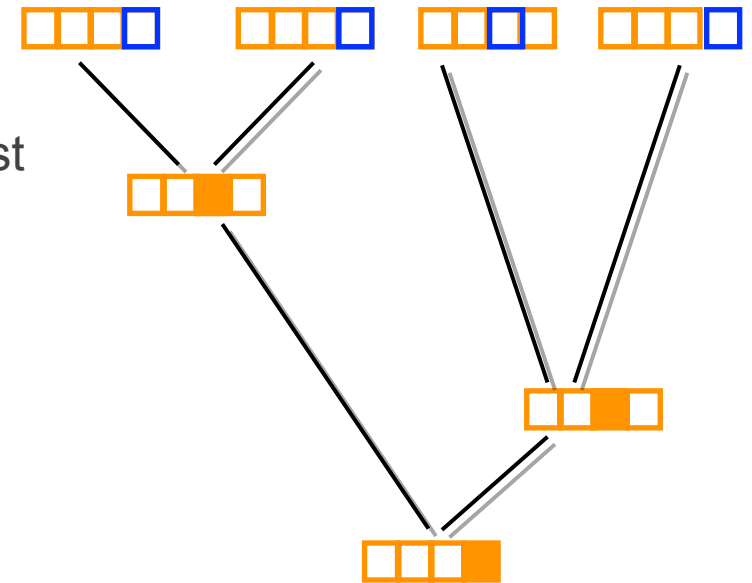
$$\mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)} \qquad \mathscr{L}_G^{(root)} \qquad \mathscr{L}_T^{(root)}$$

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Upon reaching the root of the tree, the conditional likelihood of each state is 'weighted' by the prior probability (stationary frequency) of the corresponding state, π_j, to give the unconditional probability of the jth site, $\mathbf{P}(x_j)$.

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

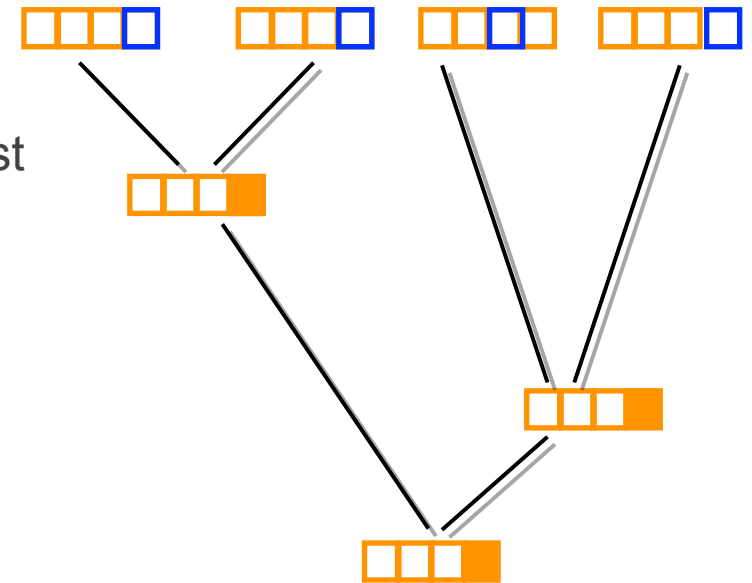$$\boxed{\mathscr{L}_A^{(root)}} \qquad \mathscr{L}_C^{(root)} \qquad \mathscr{L}_G^{(root)} \qquad \mathscr{L}_T^{(root)}$$

The probability of everything in the tree above the root, given that the process is in state A…

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Upon reaching the root of the tree, the conditional likelihood of each state is 'weighted' by the prior probability (stationary frequency) of the corresponding state, π_j, to give the unconditional probability of the jth site, $\mathbf{P}(x_j)$.

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

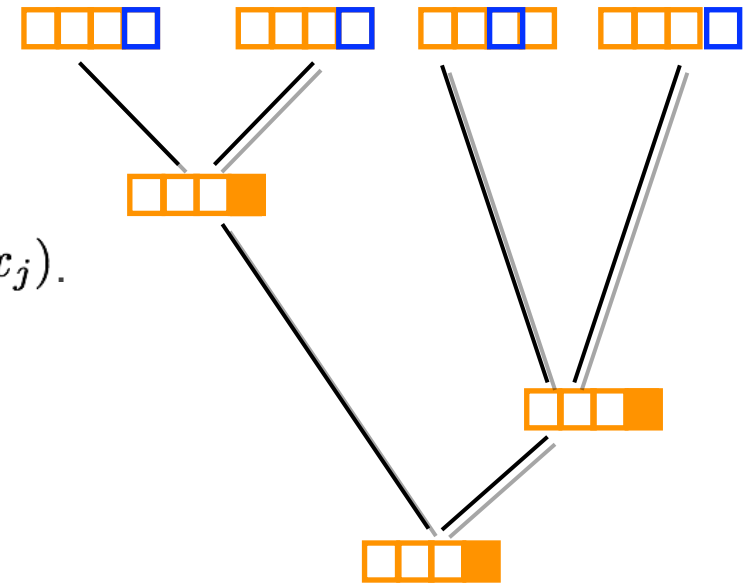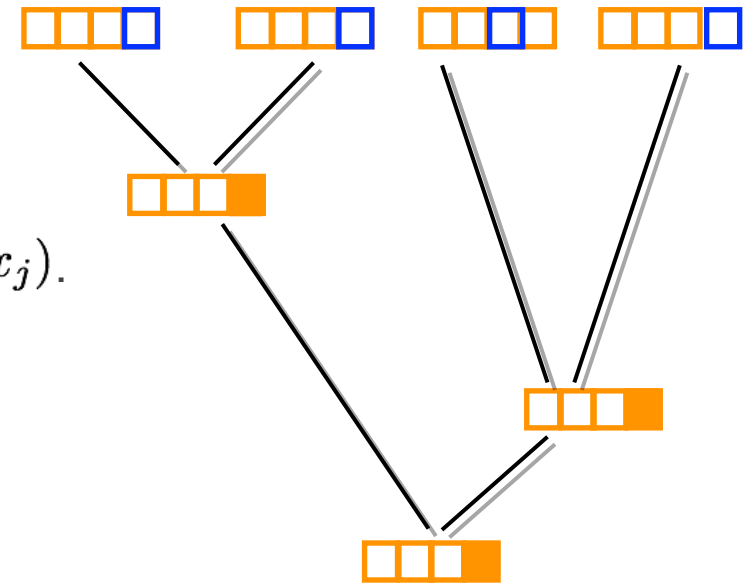$$\pi_A \mathscr{L}_A^{(root)} \qquad \mathscr{L}_C^{(root)} \qquad \mathscr{L}_G^{(root)} \qquad \mathscr{L}_T^{(root)}$$

The probability of everything in the tree above the root, *given* that the process is in state A,…

…times the probability that the process *was* in state A at the root.

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Upon reaching the root of the tree, the conditional likelihood of each state is 'weighted' by the prior probability (stationary frequency) of the corresponding state, π_j, to give the unconditional probability of the jth site, $\mathbf{P}(x_j)$.
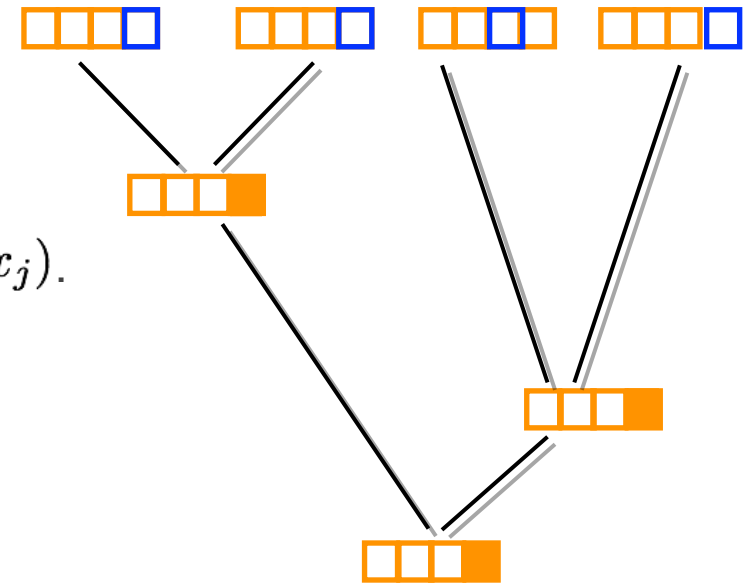
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

$$\pi_A \mathscr{L}_A^{(root)} \qquad \pi_C \mathscr{L}_C^{(root)} \qquad \mathscr{L}_G^{(root)} \qquad \mathscr{L}_T^{(root)}$$

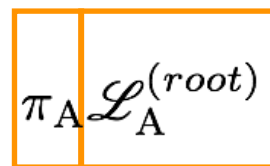We compete a similar product for the other three states, C…

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Upon reaching the root of the tree, the conditional likelihood of each state is 'weighted' by the prior probability (stationary frequency) of the corresponding state, π_j, to give the unconditional probability of the jth site, $\mathbf{P}(x_j)$.

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (\text{A,C,G,T})} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (\text{A,C,G,T})} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

$$\pi_A \mathscr{L}_A^{(root)} \qquad \pi_C \mathscr{L}_C^{(root)} \qquad \pi_G \mathscr{L}_G^{(root)} \qquad \mathscr{L}_T^{(root)}$$
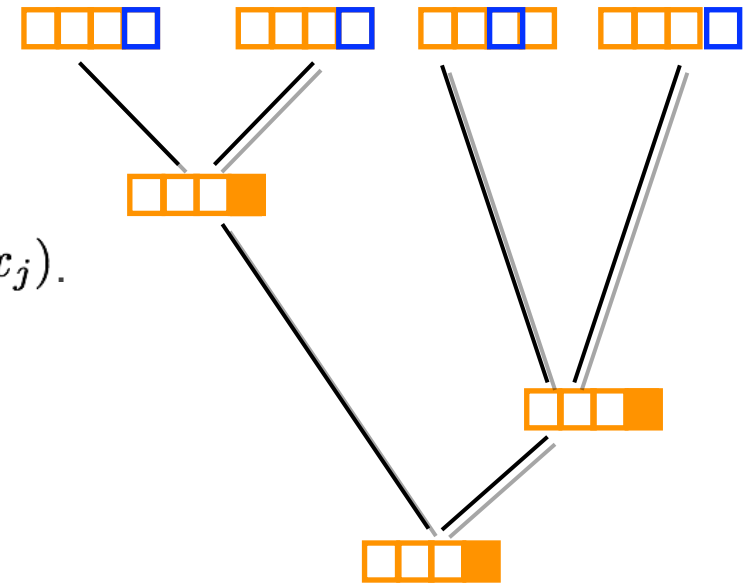
We compete a similar product for the other three states, C, G…

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Upon reaching the root of the tree, the conditional likelihood of each state is 'weighted' by the prior probability (stationary frequency) of the corresponding state, π_j, to give the unconditional probability of the jth site, $\mathbf{P}(x_j)$.

$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$

$$\pi_A \mathscr{L}_A^{(root)} \qquad \pi_C \mathscr{L}_C^{(root)} \qquad \pi_G \mathscr{L}_G^{(root)} \qquad \pi_T \mathscr{L}_T^{(root)}$$

We compete a similar product for the other three states, C, G, and T…

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Upon reaching the root of the tree, the conditional likelihood of each state is 'weighted' by the prior probability (stationary frequency) of the corresponding state, π_j, to give the unconditional probability of the jth site, $\mathbf{P}(x_j)$.
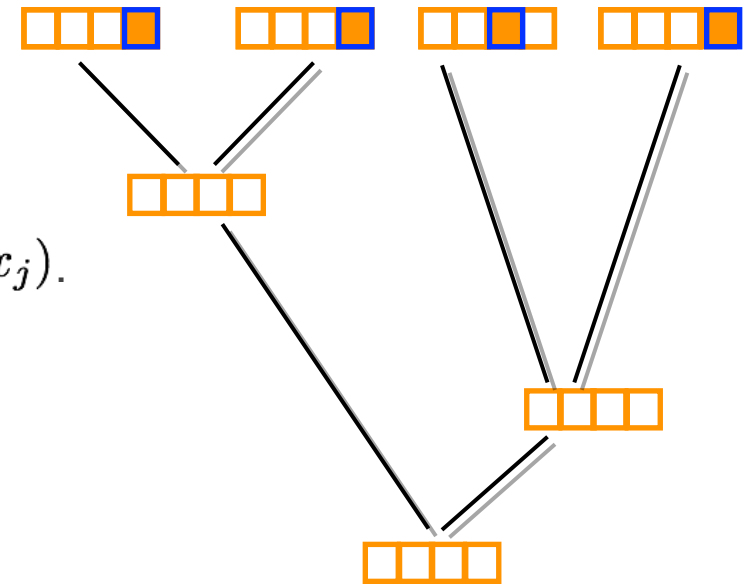
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L)\mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R)\mathscr{L}_k^{(R)} \right)$$
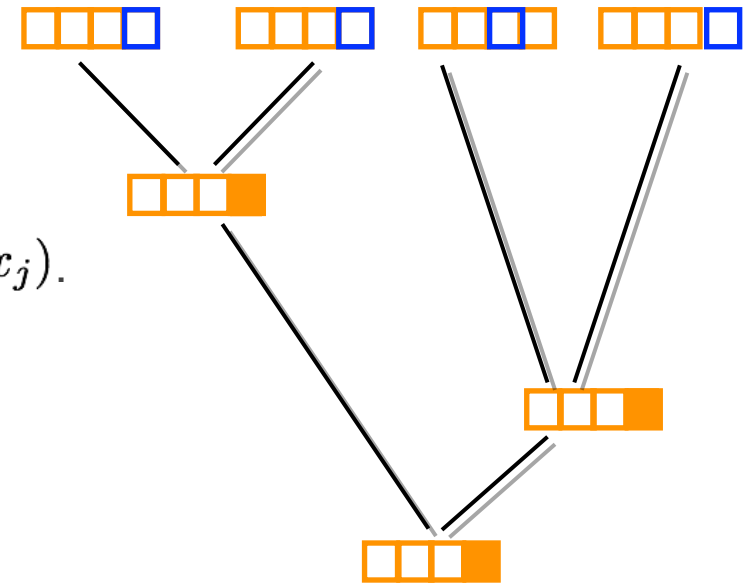
$$\mathbf{P}(TTGT) = \pi_A \mathscr{L}_A^{(root)} + \pi_C \mathscr{L}_C^{(root)} + \pi_G \mathscr{L}_G^{(root)} + \pi_T \mathscr{L}_T^{(root)}$$

…and then sum the four terms because these root states are mutually exclusive.

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Upon reaching the root of the tree, the conditional likelihood of each state is 'weighted' by the prior probability (stationary frequency) of the corresponding state, π_i, to give the unconditional probability of the data, x_i
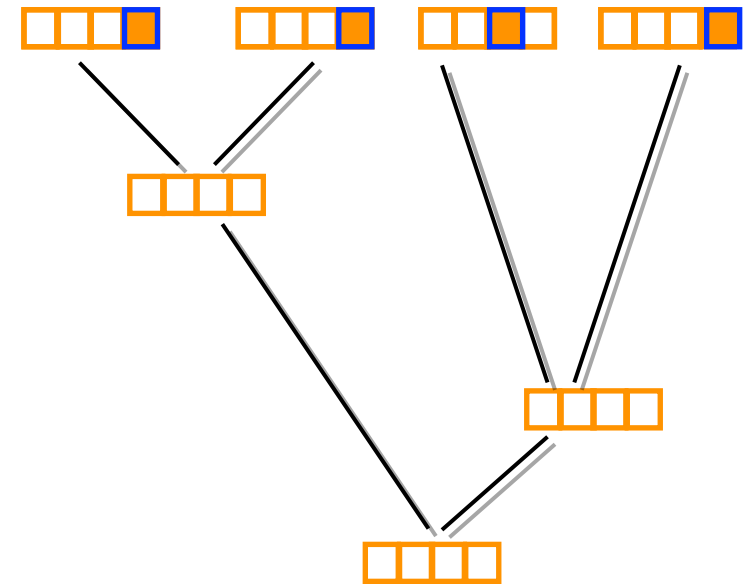
$$\mathscr{L}_i^{(Anc)} = \left(\sum_{j \in (A,C,G,T)} p_{ij}(\nu_L) \mathscr{L}_j^{(L)} \right) \times \left(\sum_{k \in (A,C,G,T)} p_{ik}(\nu_R) \mathscr{L}_k^{(R)} \right)$$

$$\mathbf{P}(TTGT) = \pi_A \mathscr{L}_A^{(root)} + \pi_C \mathscr{L}_C^{(root)} + \pi_G \mathscr{L}_G^{(root)} + \pi_T \mathscr{L}_T^{(root)}$$

→ *We have computed the likelihood of site pattern TTGT using Felsenstein's pruning algorithm!*

Calculating the Likelihood of several sites

Calculating the likelihood of an alignment assumes independence of sites

We generally assume that the substitution process is independent across sites, which allows us to combine their probabilities as the product of the N site likelihoods.

Calculating the Likelihood of several sites

Calculating the likelihood of an alignment assumes independence of sites

We generally assume that the substitution process is independent across sites, which allows us to combine their probabilities as the product of the N site likelihoods.

$$L(\tau, \nu, \Theta) \propto f(\mathbf{X} \mid \tau, \nu, \Theta) = \prod_{i=1}^{N} f(x_i \mid \tau, \nu, \Theta)$$

Calculating the Likelihood of several sites

Calculating the likelihood of an alignment assumes independence of sites

We generally assume that the substitution process is independent across sites, which allows us to combine their probabilities as the product of the N site likelihoods.

$$L(\tau, \nu, \Theta) \propto f(\mathbf{X} \mid \tau, \nu, \Theta) = \prod_{i=1}^{N} f(x_i \mid \tau, \nu, \Theta)$$

That is, we compute the likelihood for each site in the alignment (one site at a time) and then combine them as a product

Calculating the Likelihood of several sites

Calculating the likelihood of an alignment assumes independence of sites

We generally assume that the substitution process is independent across sites, which allows us to combine their probabilities as the product of the N site likelihoods.

$$L(\tau, \nu, \Theta) \propto f(\mathbf{X} \mid \tau, \nu, \Theta) = \prod_{i=1}^{N} f(x_i \mid \tau, \nu, \Theta)$$

That is, we compute the likelihood for each site in the alignment (one site at a time) and then combine them as a product

Although this is a convenient assumption, it is not very biologically realistic.

More complex models can relax it, cf Jeff Thorne's talk.

Plan: Criteria for evaluating phylogenies

- Criteria for evaluating phylogenetic trees:

  - Parsimony

  - Distance methods

  - Maximum Likelihood

    - Using a Monte Carlo simulation

    - Using Felsenstein's pruning algorithm

  - Posterior probability (Bayesian approach)

- *Conventions:*

  - We're dealing with aligned sequence data

  - gaps are not taken into account

Problems when relying on the maximum

So far, we have presented all methods for phylogenetic reconstruction as attempting to find the best tree according to some criterion.

This optimization approach has some drawbacks.

Let's see two examples.

Problems when relying on the maximum

Ex. A: likelihood surface for a simple model with two parameters:
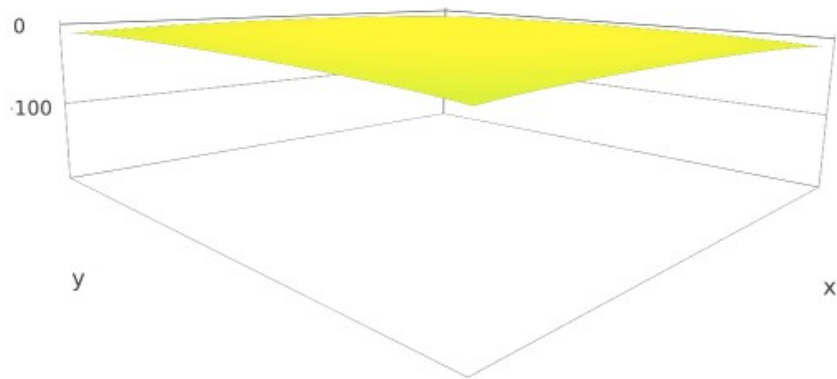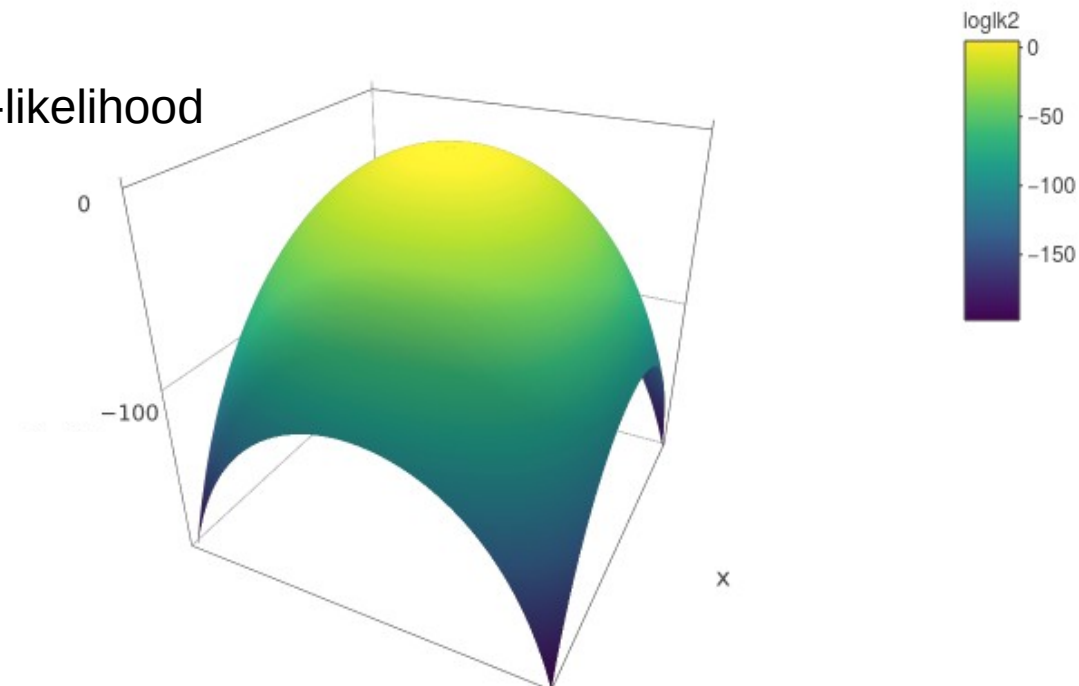
Problems when relying on the maximum

Ex. A: likelihood surface for a simple model with two parameters:
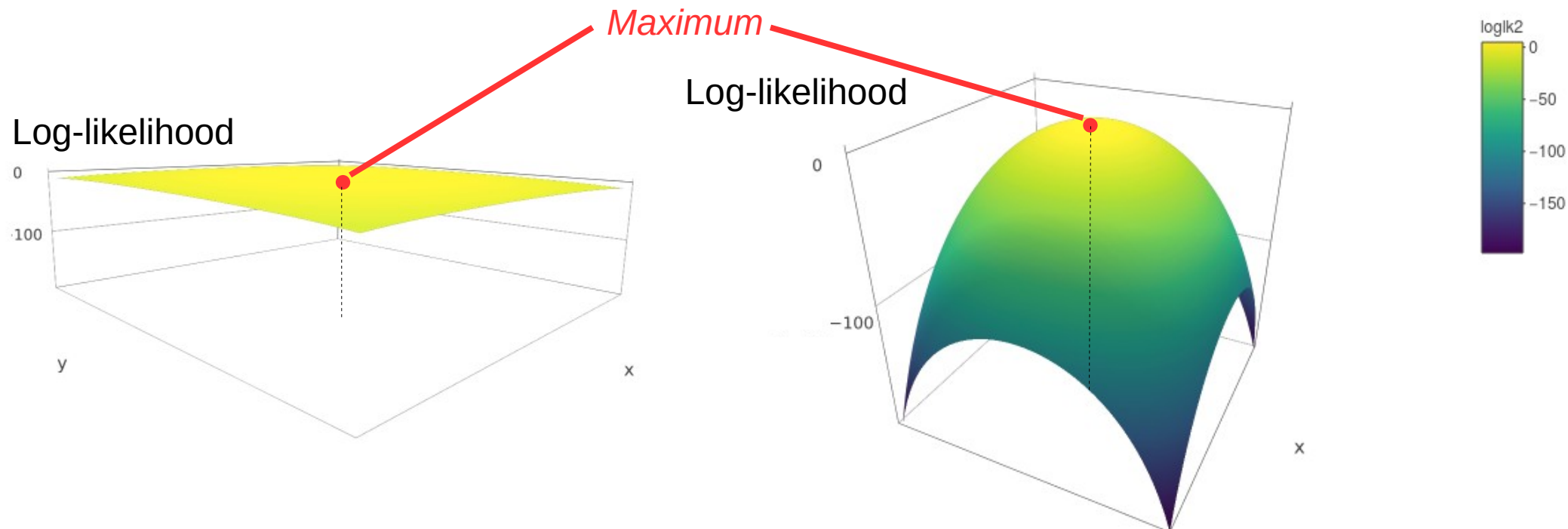
Problems when relying on the maximum

Ex. B : Given an alignment, *9* trees have very similar likelihoods and are much more likely than all the other ones.

Taking the most likely tree only provides knowledge about *1* of those *9* almost equi-likely trees.

→ It would be better to take into account all *9* of them!

Optimizing, integrating, sampling

- **Optimizing**: Looking for the topology and all other parameter values that are most likely

- **Integrating**: visiting all topologies and parameter values according to their probability

  Pros :

  – If we are interested only in the topology, we integrate over all "nuisance" parameters

  – One gets confidence intervals for free

- **Sampling:** like integrating, but we do not attempt to be exhaustive in our exploration of parameter values

Likelihood vs Bayesian approach for sampling

Likelihood : $P(D|M,\theta)$

We want to sample parameter values θ of model M.

How can we know that the 9 sets of parameter values θ we have sampled are much more likely than all other θ, without sampling everything?

$$\sum_\theta P(D|M,\theta) != 1 \qquad \rightarrow \text{NOT a probability distribution.}$$

Likelihood vs Bayesian approach for sampling

Likelihood : $P(D|M,\theta)$

We want to sample parameter values θ of model M.

How can we know that the 9 sets of parameter values θ we have sampled are much more likely than all other θ, without sampling everything?

$$\sum_{\theta} P(D|M,\theta)\, != 1 \qquad \rightarrow \text{NOT a probability distribution.}$$

But :

$$\sum_{\theta} P(\theta|D,M) = 1 \qquad \rightarrow \text{True probability distribution.}$$

Likelihood vs Bayesian approach for sampling

Likelihood : *P(D|M,θ)*

We want to sample parameter values *θ* of model *M*.

How can we know that the *9* sets of parameter values *θ* we have sampled are much more likely than all other *θ*, without sampling everything?

$$\sum_{\theta} P(D|M,\theta)! = 1 \qquad \rightarrow \text{NOT a probability distribution.}$$

But :

$$\sum_{\theta} P(\theta|D,M) = 1 \qquad \rightarrow \text{True probability distribution.}$$

If the added probabilities of *9* sets of parameter values *θ* reach *0.99*, I know I have sampled all the most probable parameter values, without needing to sample more!

Likelihood vs Bayesian approach for sampling

Likelihood : *P(D|M,θ)*

We want to sample parameter values *θ* of model *M*.

How can we know that the *9* sets of parameter values *θ* we have sampled are much more likely than all other *θ*, without sampling everything?

$$\sum_{\theta} P(D|M,\theta)! = 1 \qquad \rightarrow \text{NOT a probability distribution.}$$

But :

$$\sum_{\theta} P(\theta|D,M) = 1 \qquad \rightarrow \text{True probability distribution.}$$

Posterior probability

If the added probabilities of *9* sets of parameter values *θ* reach *0.99*, I know I have sampled all the most probable parameter values, without needing to sample more!

Simplifying the notation

When we write $P(\theta|D,M)$, we mean that we are computing the probability of parameter values θ given model M and given data D.

Here M is used to represent the structure of the model.

For instance:

- The fact that all sites are independent
- The fact that they share a single value of the transition/transversion ratio
- The fact that we have a Birth-death prior on a chronogram
- Etc…

In some cases, we may want to integrate over different model structures M: e.g. a model with 1 value of the transition/transversion ratio, or 2, or 3…

In most cases, we use a single model M, so we will forget about it in most of the following:

$$P(\theta|D,M) \rightarrow P(\theta|D)$$

Bayesian inference

Bayes theorem:

$$P(\theta|D) = \frac{P(\theta \wedge D)}{P(D)} = \frac{P(D \wedge \theta)}{P(D)}$$

$$P(\theta|D) = \frac{P(D|\theta)\,P(\theta)}{P(D)}$$

Bayesian inference

Bayes theorem:

$$P(\theta|D) = \frac{P(\theta \wedge D)}{P(D)} = \frac{P(D \wedge \theta)}{P(D)}$$

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{\int\limits_{\theta} P(D|\theta)d\theta}$$

Bayesian inference

Bayes theorem:

$$P(\theta|D) = \frac{P(\theta \wedge D)}{P(D)} = \frac{P(D \wedge \theta)}{P(D)}$$

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{\int_\theta P(D|\theta)d\theta}$$

Hard to compute. In practice, we can often ignore it.

Bayesian inference

Bayes theorem:

$$P(\theta|D) = \frac{P(\theta \wedge D)}{P(D)} = \frac{P(D \wedge \theta)}{P(D)}$$

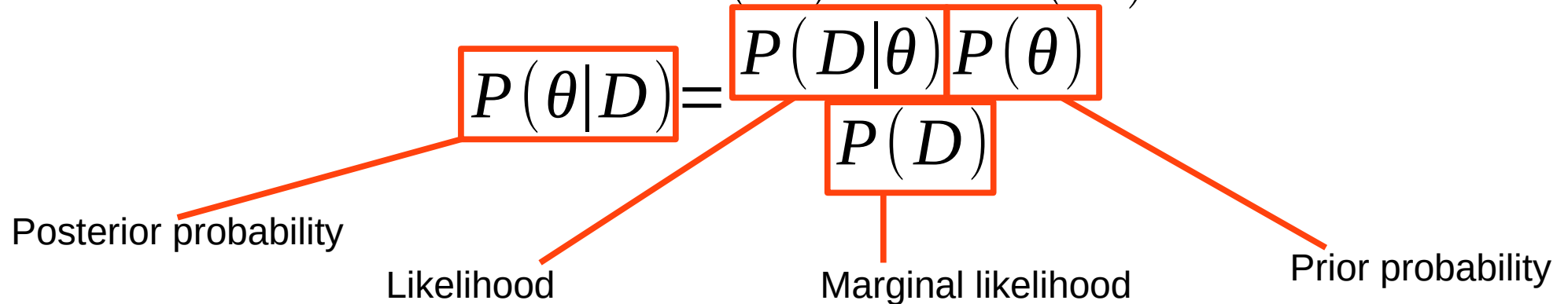$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

Posterior probability

Likelihood

Marginal likelihood

Prior probability

The importance of prior probabilities

The Bayesian approach amounts to considering that all parameters of a model are random variables in a probabilistic world.

Therefore, one needs to assign probability distributions to those parameters: the **priors**.

$$P(\theta|D) = P(D|\theta)\,P(\theta)/P(D)$$

→ PROS: allows incorporating prior information coming from the analysis of other data: my conclusion relies on more than the tiny amount of data I have analyzed in a given experiment.

→ CONS: introduces prior information into the analysis: is my conclusion simply propagating my prior information?

Bayesian inference

Bayes theorem: $P(\theta|D) = \dfrac{P(D|\theta)P(\theta)}{P(D)}$

We are interested in the posterior probability of the parameter values: we want to **sample parameter values according to their posterior probability given the data**.
Therefore we need to get the distribution for *P(θ|D)*.

We have two solutions:
- ***Integrate*** *over all parameter values*
  - Requires some maths
  - Not always possible
- ***Sample*** *from this distribution in some (smart) way*

Naive sampling of *P(θ|D)*

Random sampling

Can work for small problems.

- Inference of the probability of getting heads from a coin

http://rpubs.com/boussau/384012

Naive sampling of *P(θ|D)*

Random sampling

Can work for small problems.

- Inference of the probability of getting heads from a coin:

http://rpubs.com/boussau/384012

<u>*Will not work for complex problems :*</u> in phylogenetics, for *20* sequences, there are already *221. 10^{18}* possible topologies… for which one would want to sample branch lengths and other parameter values.

→ **We cannot reasonably hope to sample trees with a good probability using random sampling**

Naive sampling of *P(θ|D)*

Random sampling

Can work for small problems.

- Inference of the probability of getting heads from a coin:

  http://rpubs.com/boussau/384012

<u>*Will not work for complex problems :*</u> in phylogenetics, for *20* sequences, there are already *221. 10^{18}* possible topologies… for which one would want to sample branch lengths and other parameter values.

→ **We cannot reasonably hope to sample trees with a good probability using random sampling**

Later we will see smarter ways to sample from complex posterior distributions
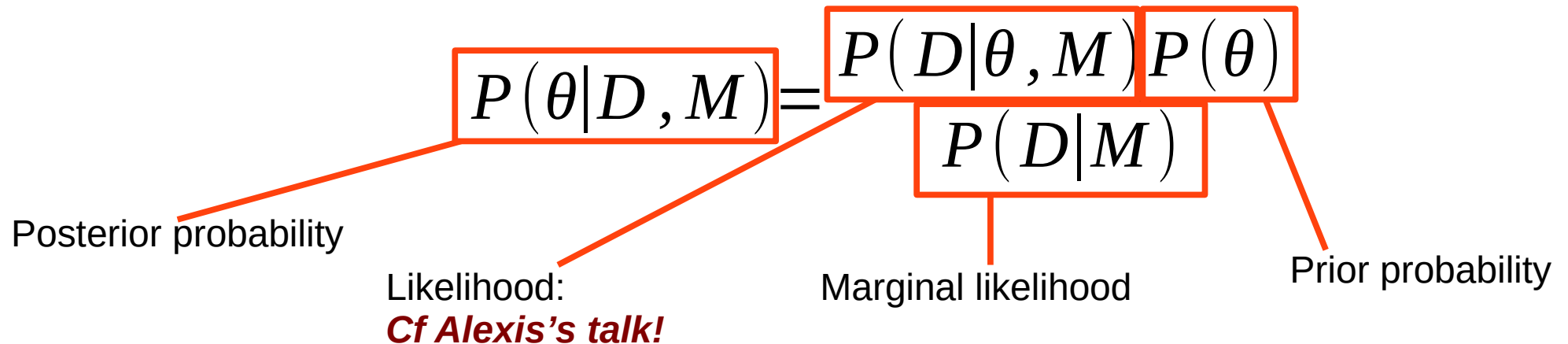
Bayesian phylogenetics

$$P(\theta|D,M) = \frac{P(D|\theta,M)P(\theta)}{P(D|M)}$$

Posterior probability

Likelihood

Marginal likelihood

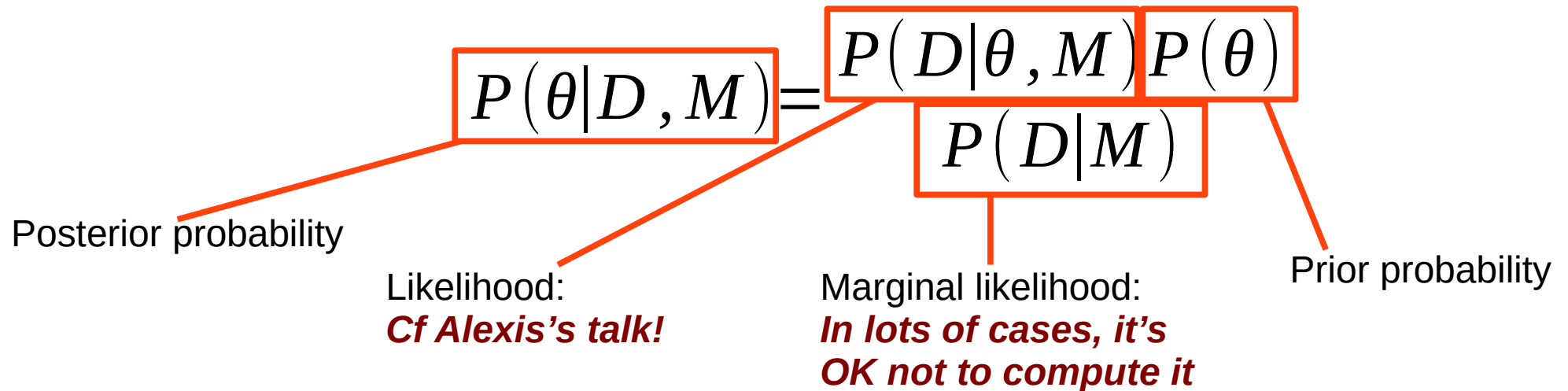Prior probability

- In phylogenetics:
  - D: aligned sequence data
  - M: the model can be very complicated, but usually contains:
    - Topology
    - Branch lengths
    - Rate matrix
    - Etc…
  - θ: the values of the parameters above

Bayesian phylogenetics

$$P(\theta|D,M)=\frac{P(D|\theta,M)\,P(\theta)}{P(D|M)}$$

Posterior probability

Likelihood:
Cf Alexis's talk!

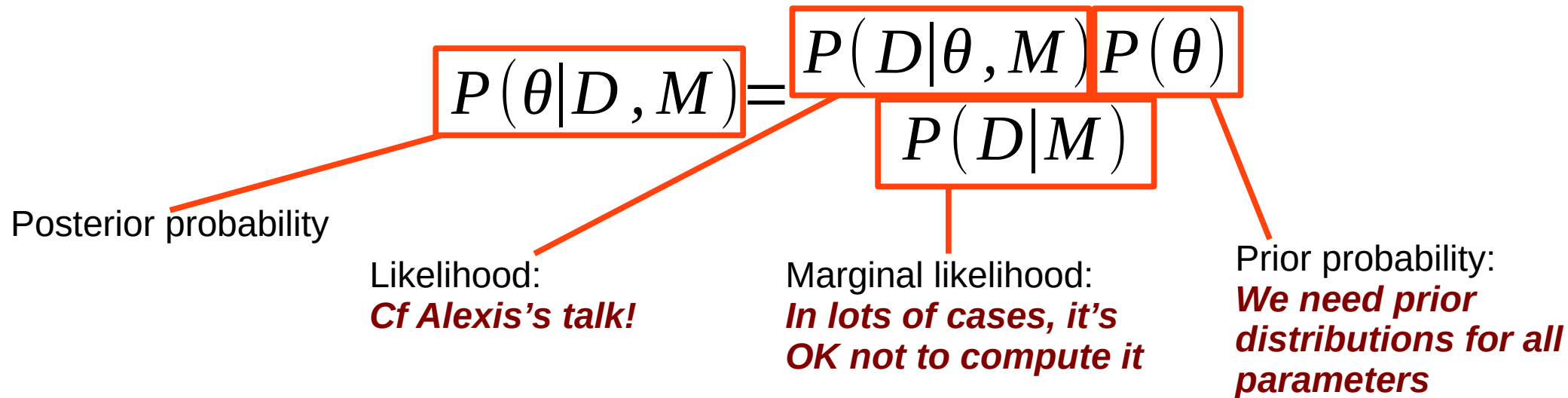Marginal likelihood

Prior probability

- In phylogenetics:
  - D: aligned sequence data
  - M: the model can be very complicated, but usually contains:
    - Topology
    - Branch lengths
    - Rate matrix
    - Etc…
  - θ: the values of the parameters above

Bayesian phylogenetics

$$P(\theta|D,M)=\frac{P(D|\theta,M)P(\theta)}{P(D|M)}$$

Posterior probability

Likelihood:
Cf Alexis's talk!

Marginal likelihood:
In lots of cases, it's OK not to compute it

Prior probability

- In phylogenetics:
  - D: aligned sequence data
  - M: the model can be very complicated, but usually contains:
    - Topology
    - Branch lengths
    - Rate matrix
    - Etc…
  - θ: the values of the parameters above

Bayesian phylogenetics

$$P(\theta|D,M) = \frac{P(D|\theta,M)\,P(\theta)}{P(D|M)}$$

Posterior probability

Likelihood:
Cf Alexis's talk!

Marginal likelihood:
In lots of cases, it's OK not to compute it

Prior probability:
We need prior distributions for all parameters

- In phylogenetics:
  - D: aligned sequence data
  - M: the model can be very complicated, but usually contains:
    - Topology
    - Branch lengths
    - Rate matrix
    - Etc…
  - θ: the values of the parameters above

Bayesian phylogenetics

In phylogenetics, one may want to sample:

- Phylogenetic trees (when we don't care about dates):
  - topologies
  - branch lengths
- Chronograms (when dates are of interest)
- Rates:
  - Rates of exchangeability
  - Site-wise rates of evolution
  - Birth-death rates
  - ...
- Frequencies:
  - ACGT equilibrium frequencies
  - Root frequencies
- Other parameters...

MCMC in phylogenetics

| Parameter	Prior (example)
Topology	Uniform...
Branch lengths	Exponential, Gamma+Dirichlet...
Chronogram	Birth-Death, Coalescent...
Rates of exchangeability	Dirichlet...
Site-wise rates of evolution	(Discretized) Gamma...
Birth-death rates	Lognormal, Exponential...
ACGT equilibrium frequencies	Dirichlet...
Root frequencies	Dirichlet...

Prior for proportions

- Beta prior
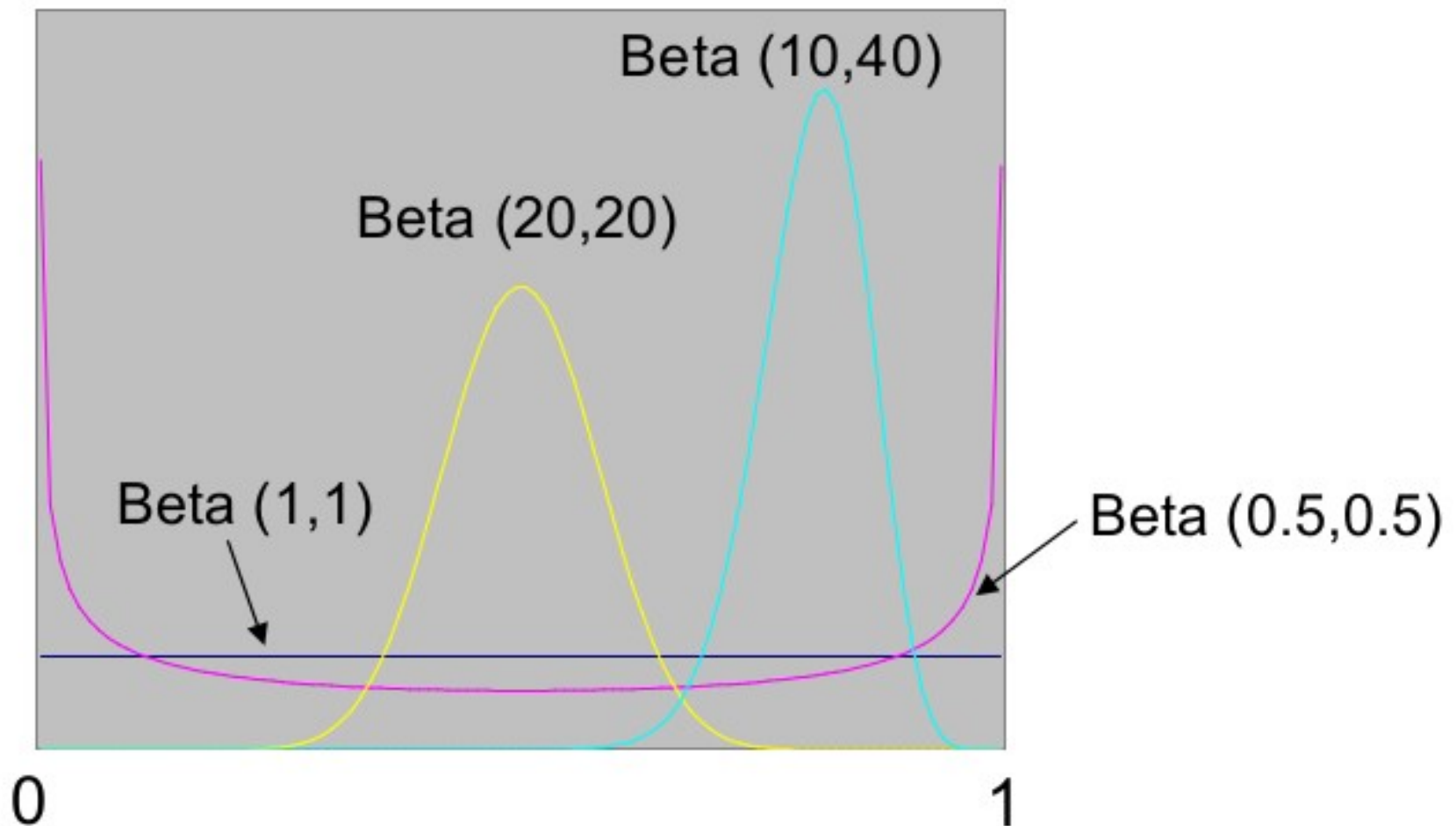
Prior for simplices (ACGT equilibrium frequencies, exchangeability rates…)

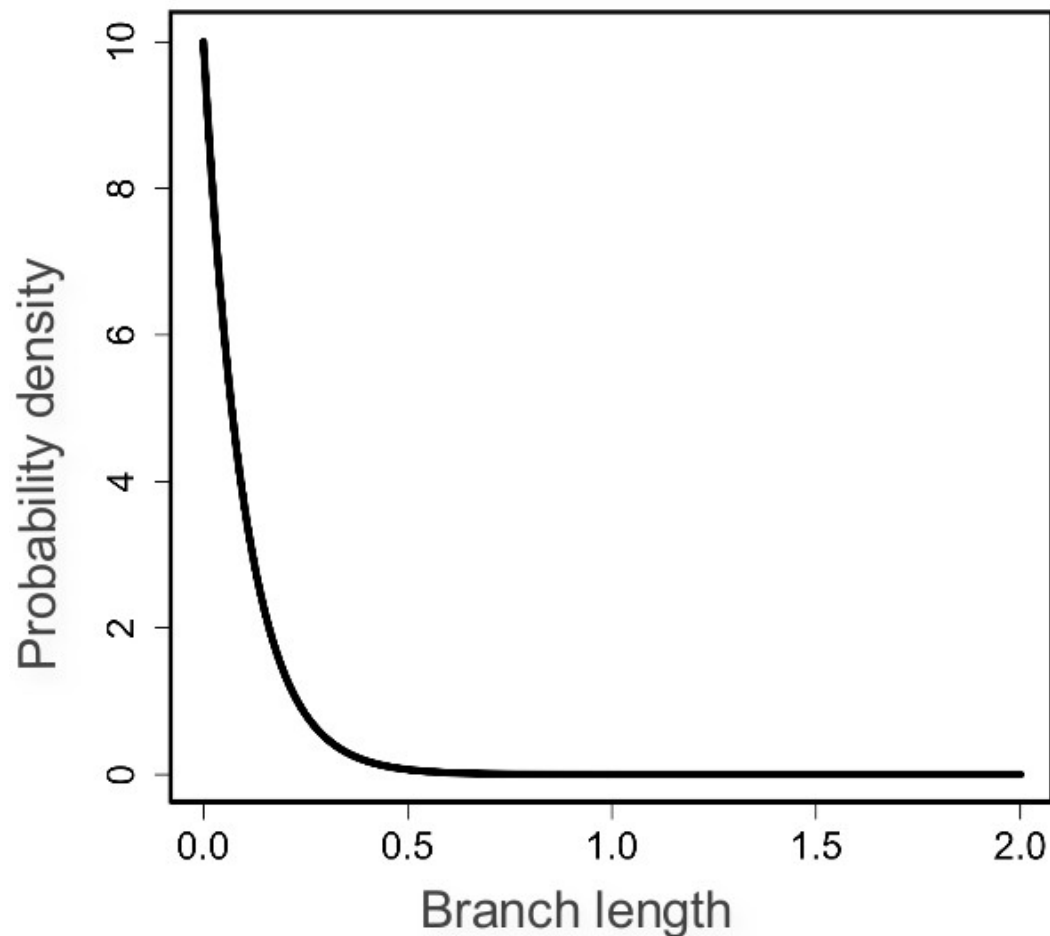- Dirichlet prior (*i.e.* Beta but in more dimensions)
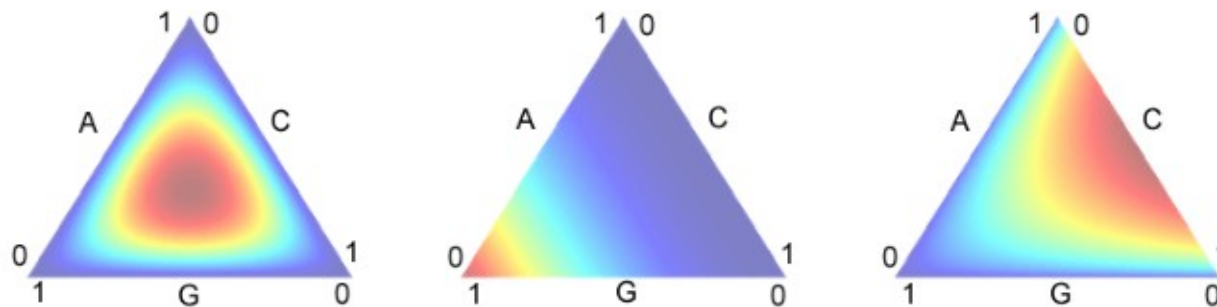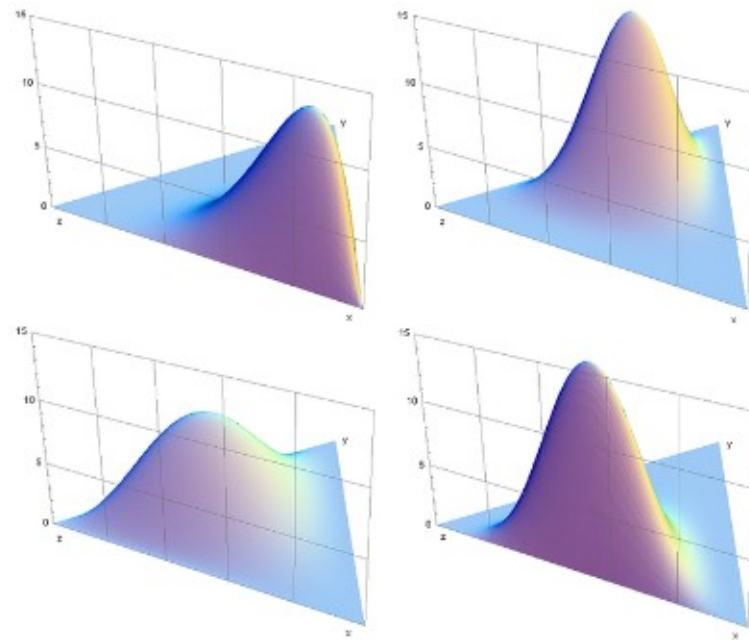
Unrooted topology prior

- Discrete uniform prior
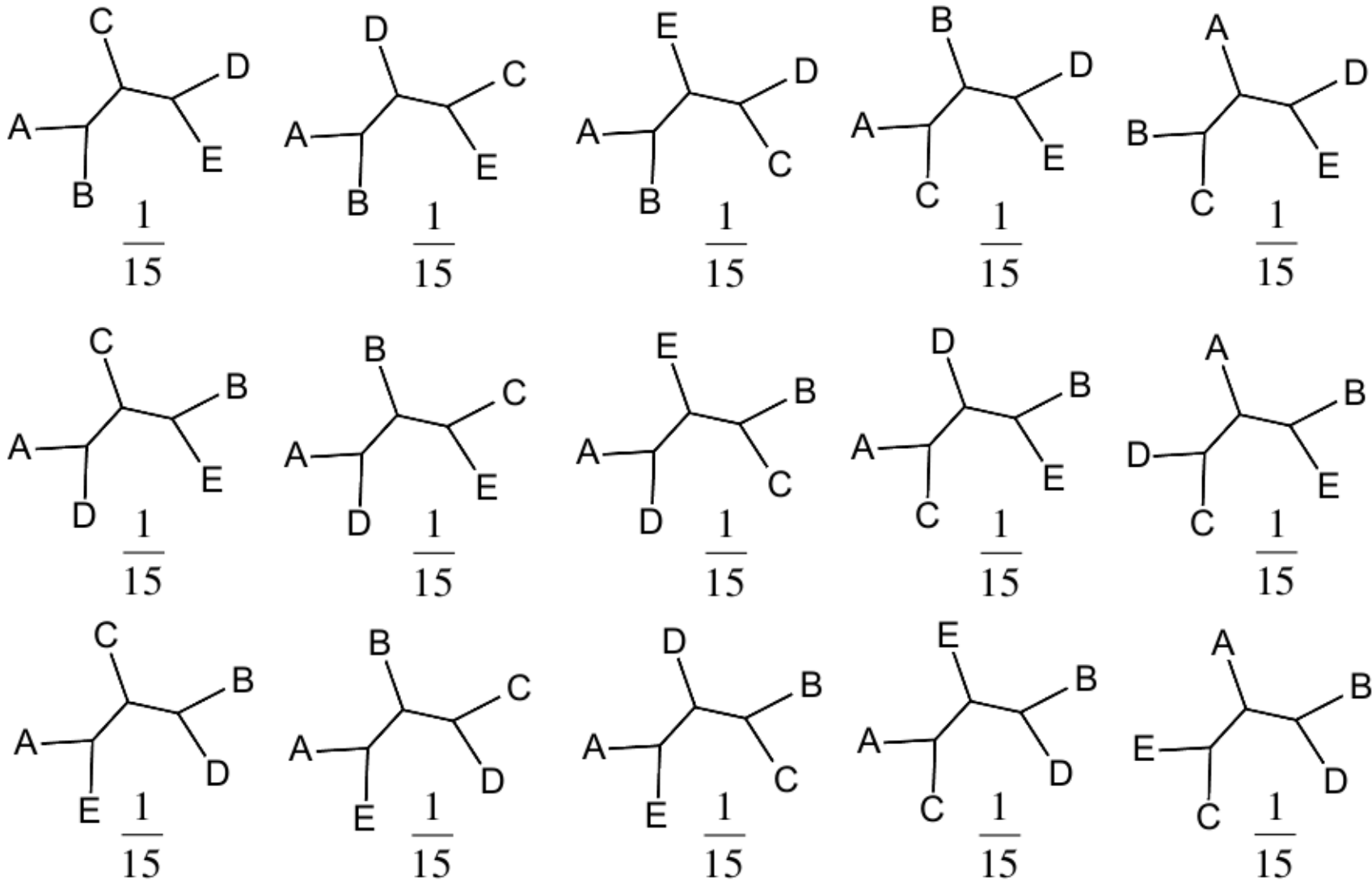
Prior for branch lengths
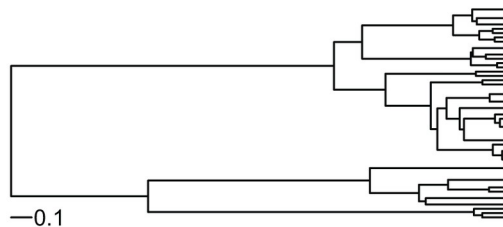
- Exponential prior (λ=10, mean=0.1)

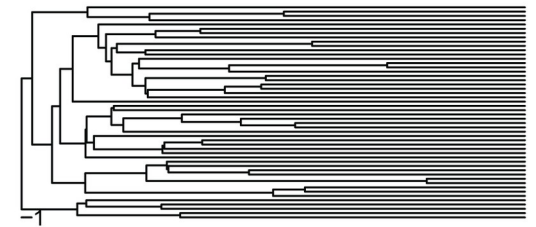- Alternative: Gamma for total tree length+Dirichlet

Chronogram prior: Birth-death process

Phylogenies simulated under a model with saturated diversity and a constant turnover rate (Model 1) have short terminal branches compared to phylogenies simulated under the pure-birth process (Yule model; Model 5). With saturated diversity but decaying turnover rates, terminal branches become longer (Model 2). Compared to the pure-birth process (Model 5), the presence of extinction pushes phylogenetic nodes towards the tips (Model 3), whereas a decay in speciation rate pushes them towards the root (Model 6). In the presence of both extinction and a decay in speciation rate (Model 4), however, these two effects counteract, producing a phylogeny that appears similar to the pure-birth model. All phylogenies were simulated with the same initial speciation rate (six speciation events per time unit). The extinction rate in Models 3 and 4a was identical (three speciation events per time unit). The exponential variation in speciation rate in Models 2, 4a, and 6 was identical (0.25 per time unit). Note the different time scales.
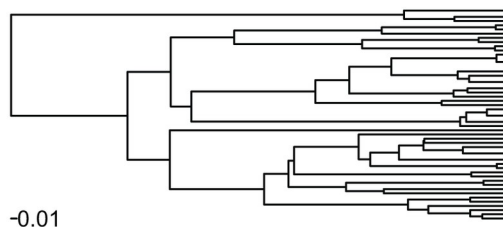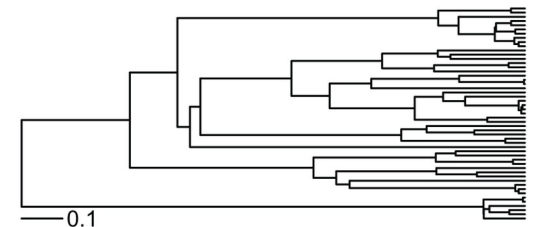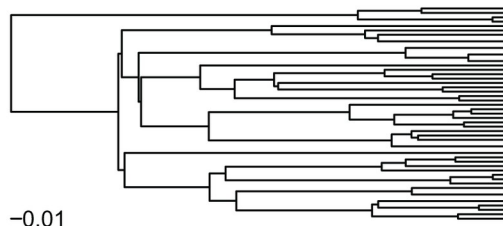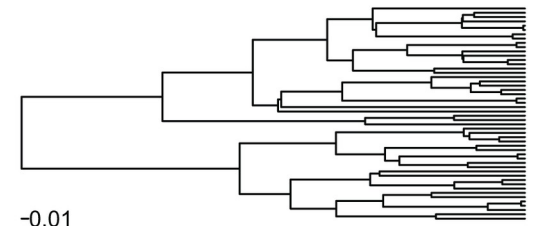
Morlon et al., 2010

Model 1

Model 2

Model 3

Model 4a

Model 5

Model 6

Bayesian phylogenetics

$$P(\theta|D,M) = \frac{P(D|\theta,M)\,P(\theta)}{P(D|M)}$$

Posterior probability

Likelihood:
Cf Alexis's talk!

Marginal likelihood

Prior probability

- In phylogenetics:
  - D: aligned sequence data
  - M: the model can be very complicated, but usually contains:
    - Topology
    - Branch lengths
    - Rate matrix
    - Etc…
  - θ: the values of the parameters above

Bayesian phylogenetics

$$P(\theta|D,M)=\frac{P(D|\theta,M)\,P(\theta)}{P(D|M)}$$

Posterior probability

Likelihood:
Cf Alexis's talk!

Marginal likelihood

Prior probability

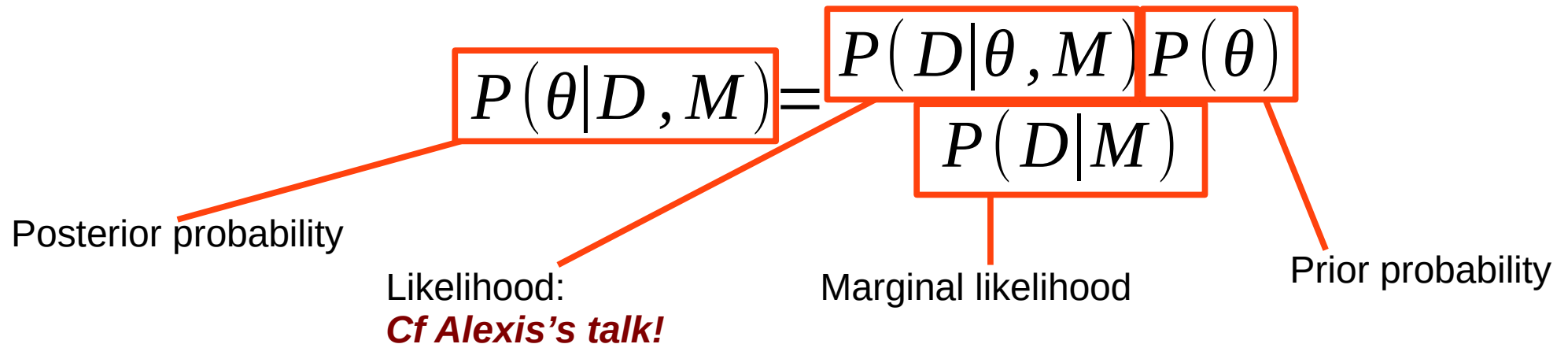$$P(\theta)=P(topology)\,P(branch\,lengths)$$
$$P(exchangeabilities)\,P(equilibrium\,frequencies)\ldots$$
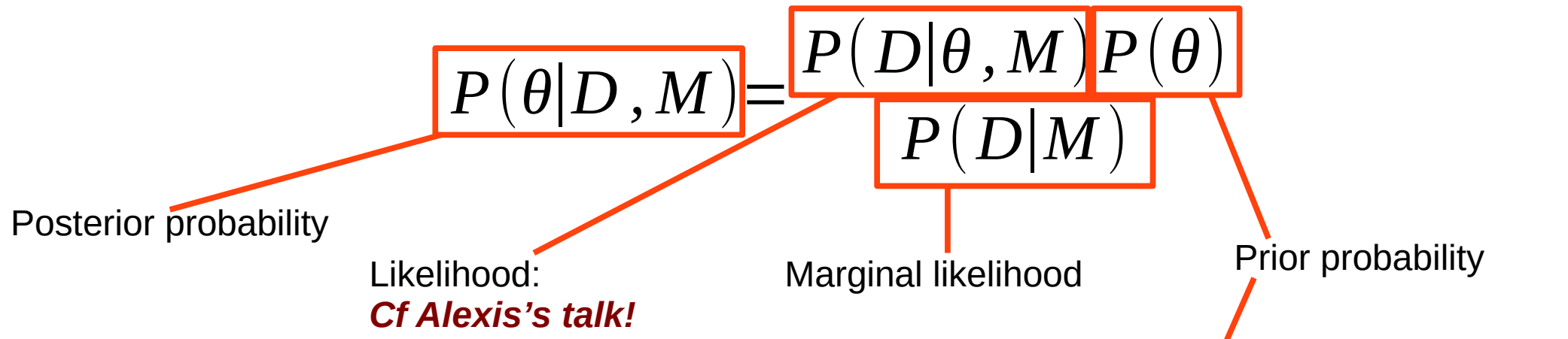
- In phylogenetics:
  - D: aligned sequence data
  - M: the model can be very complicated, but usually contains:
    - Topology
    - Branch lengths
    - Rate matrix
    - Etc…
  - θ: the values of the parameters above

Smart sampling of *P(θ|D)*

- *Smart sampling = no need to sample a huge number of points !*

→ Ideally : Sample trees and parameter values with a frequency equal to their probability.

E.g.: we would sample one of our *9* trees *99%* of the time!

Several approaches:

- *Importance sampling*

- *Markov Chain Monte Carlo (MCMC)*

- Sequential Monte Carlo

- ...

Smart sampling of $P(\theta|D)$

- *Smart sampling = no need to sample a huge number of points !*

→ Ideally : Sample trees and parameter values with a frequency equal to their probability.

E.g.: we would sample one of our *9* trees *99%* of the time!

Several approaches:

- *Importance sampling*
- *Markov Chain Monte Carlo (MCMC)*
- Sequential Monte Carlo
- ...

Next time!

Conclusion

- One can perform inference according to the posterior probability of a probabilistic model

- That's what Bayesian inference is about

- It combines the likelihood with priors on parameter values

- In phylogenetics, the likelihood is typically computed thanks to Felsenstein's pruning algorithm (1981, cf Alexis)

- Then priors need to be defined for:
  - Topologies and branch lengths / chronograms
  - Rates
  - Other parameters…

- When parameters are independent, to compute the prior of all parameter values, one only needs to compute the product over individual parameter priors

Plan: Criteria for evaluating phylogenies

- Criteria for evaluating phylogenetic trees:

  - Parsimony

  - Distance methods

  - Maximum Likelihood

    - Using a Monte Carlo simulation

    - Using Felsenstein's pruning algorithm

  - Posterior probability (Bayesian approach)

- *Conventions:*

  - We're dealing with aligned sequence data

  - gaps are not taken into account

Optimization algorithms (Alexis)
MCMC (Mike)