Introduction to Bioinformatics for Computer Scientists

Lecture 8

Plan for next lectures

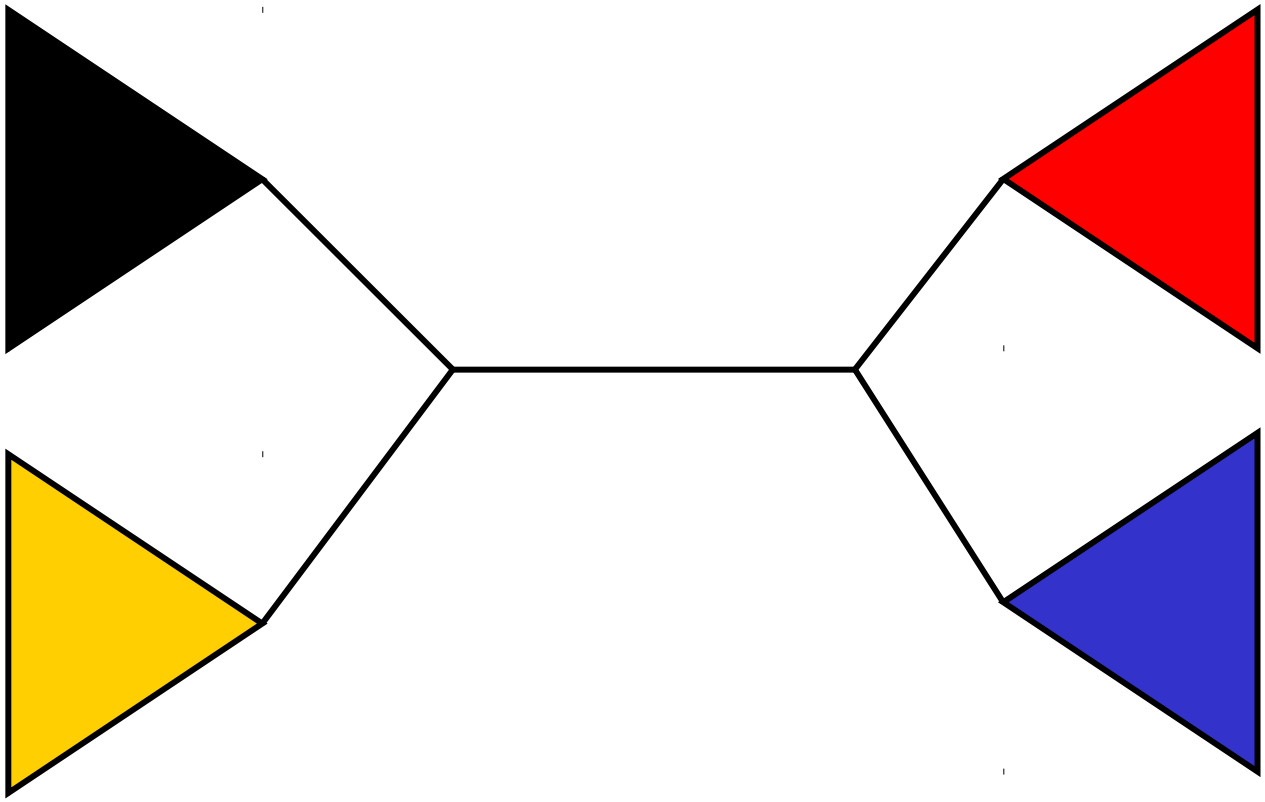
- Today (Alexis):
  - Tree Search Algorithms continued
  - Phylogenetic Maximum Likelihood: The Maths
- Lecture 9 (Alexis) Maximum Likelihood Continued
  - Implementation
  - Parallelization
- Lecture 10 (Lucas) Discrete Operations on Trees

Search Strategies

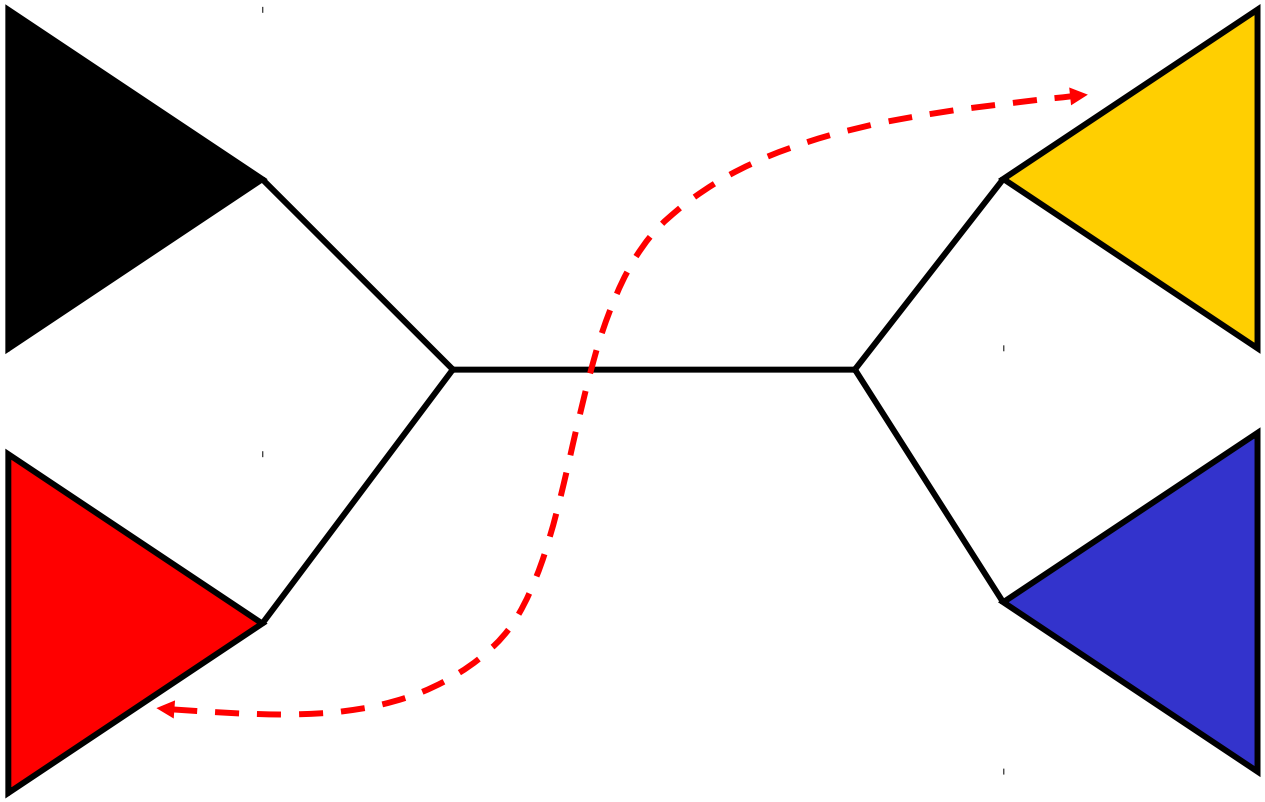
- Given a comprehensive tree
- Apply topological alteration mechanisms in some order to improve the score, for instance, via
  - Hill-climbing
  - Simulated annealing
  - Genetic algorithms
  - Some other technique

    → design of ad hoc heuristics
- The three basic moves are:
  - **NNI:** Nearest Neighbor Interchange
  - **SPR:** Subtree Pruning and Re-Grafting
  - **TBR:** Tree Bisection and Reconnection

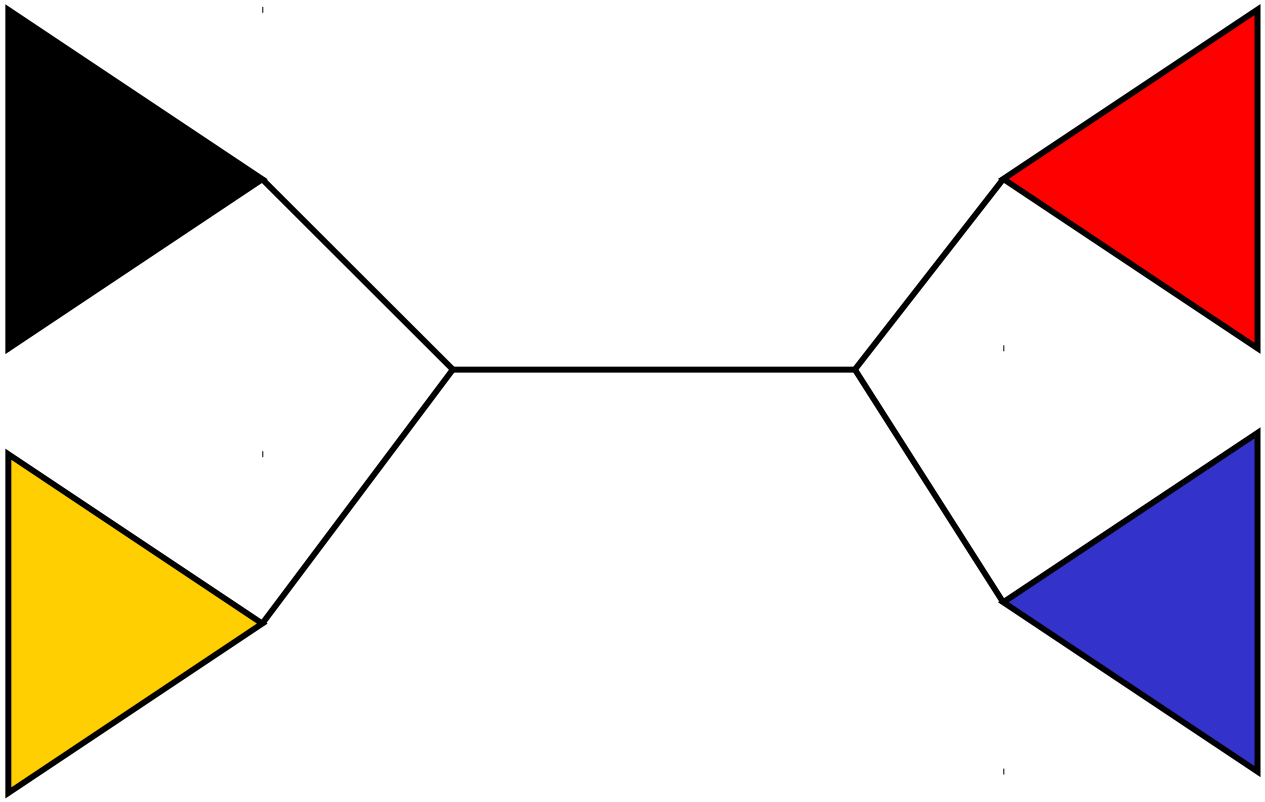
NNI



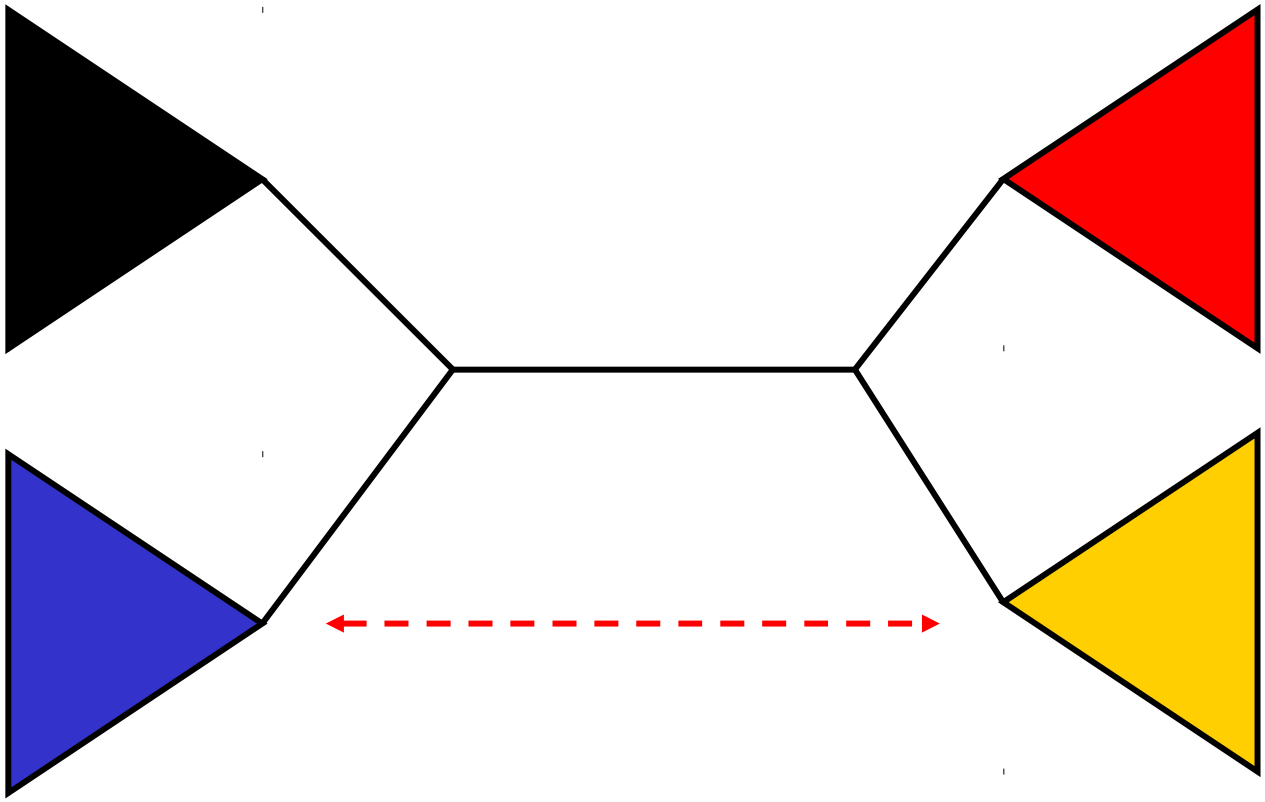
NNI



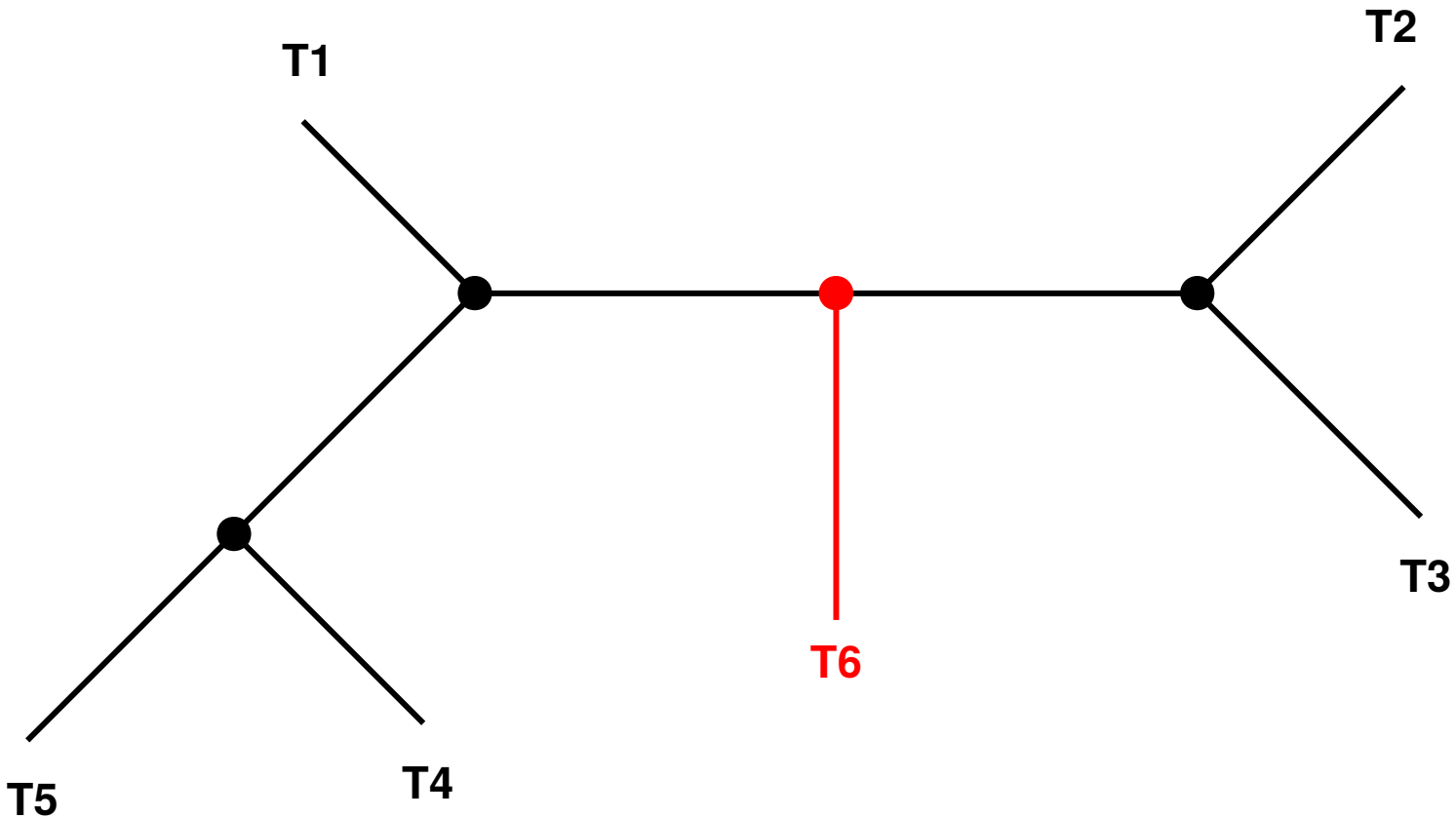
NNI



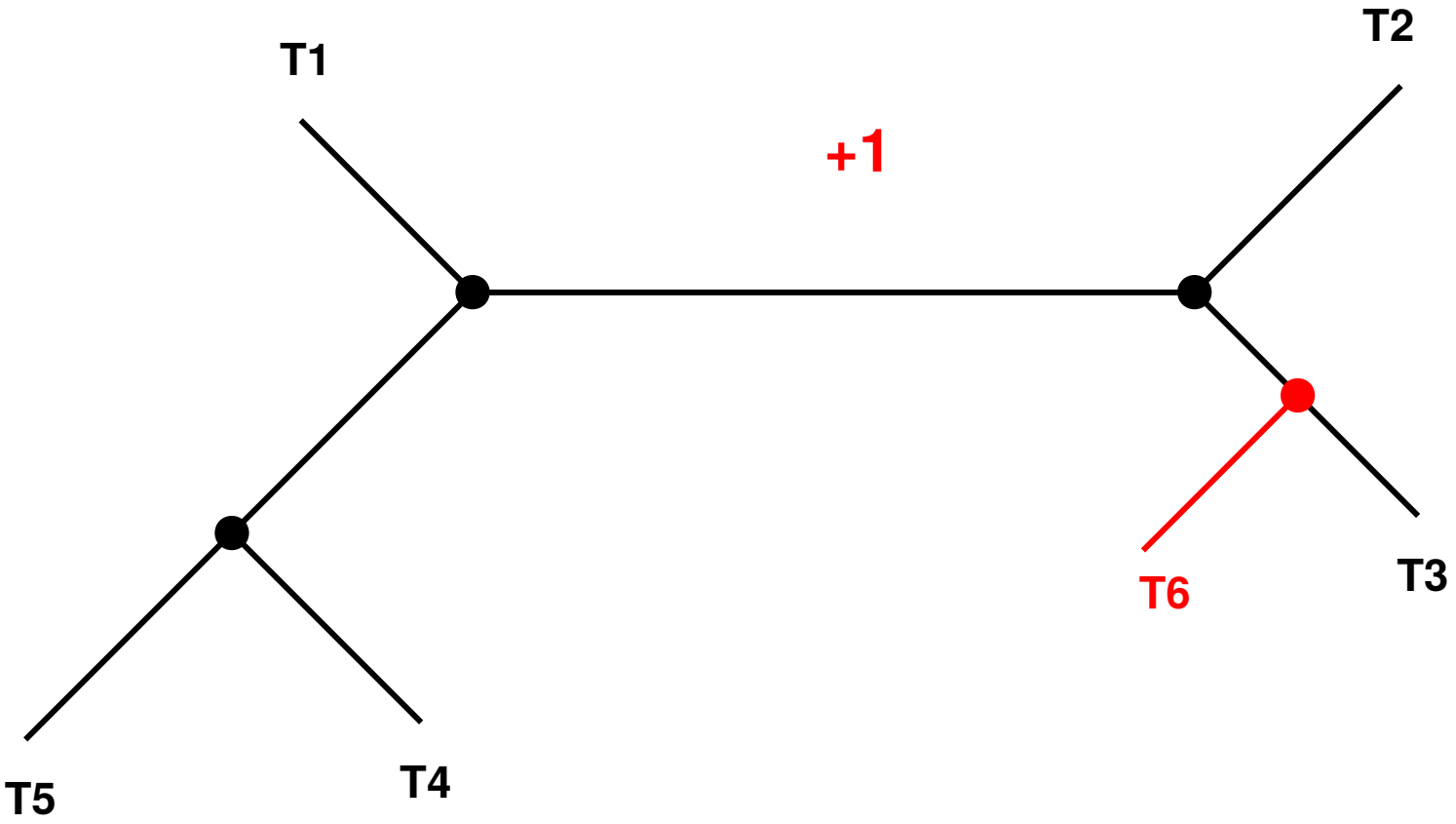
NNI



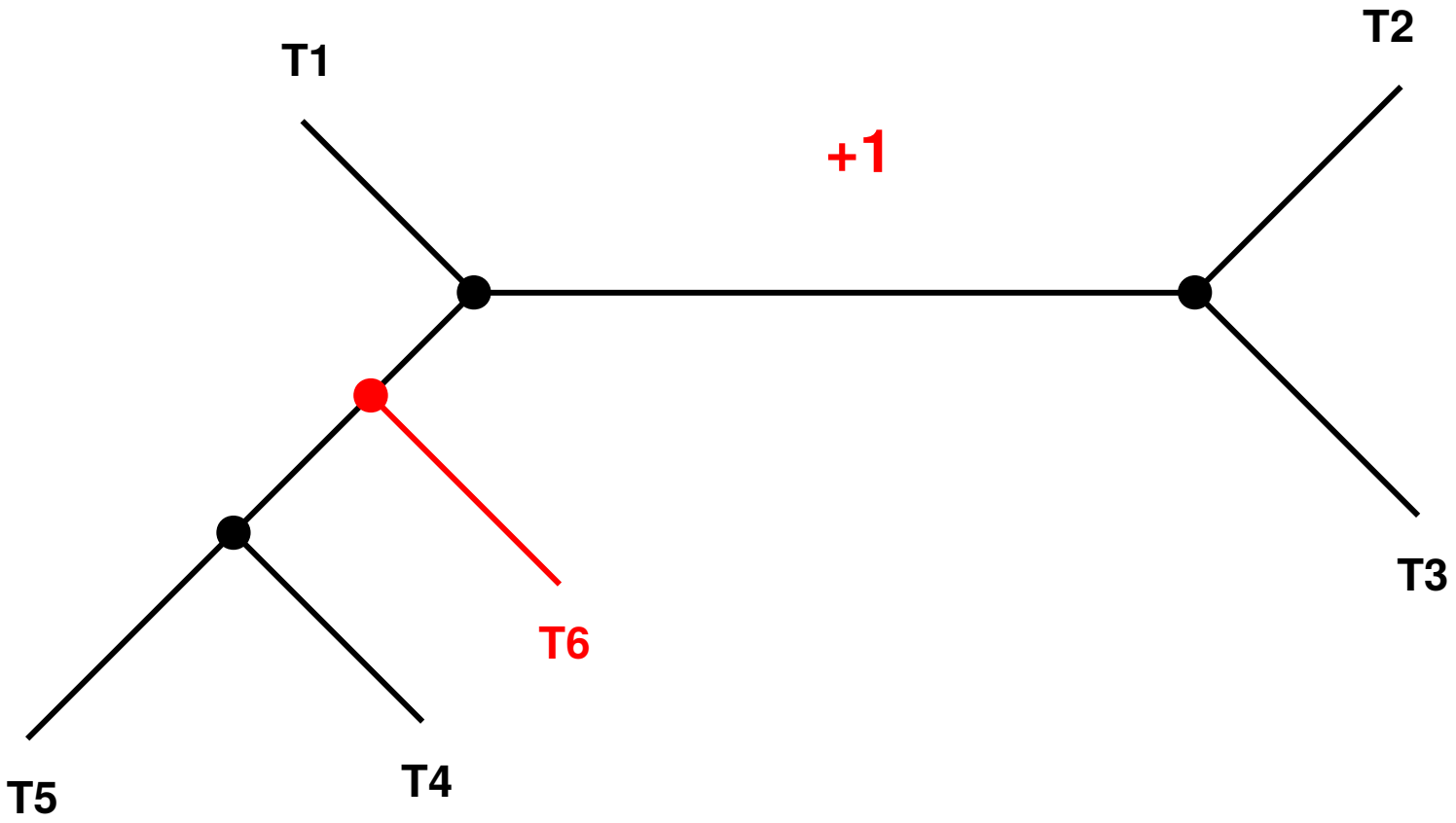
SPR



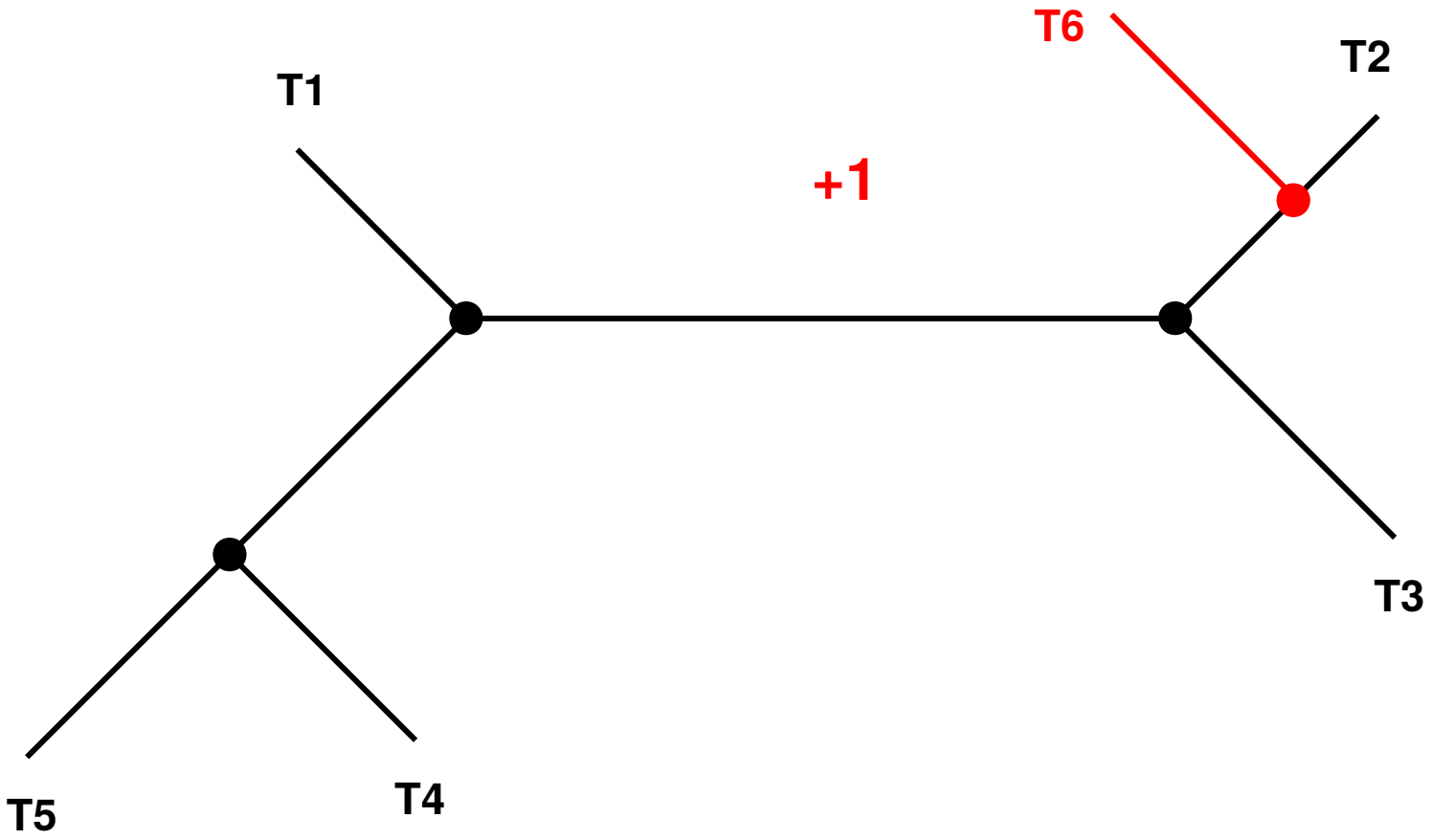
SPR



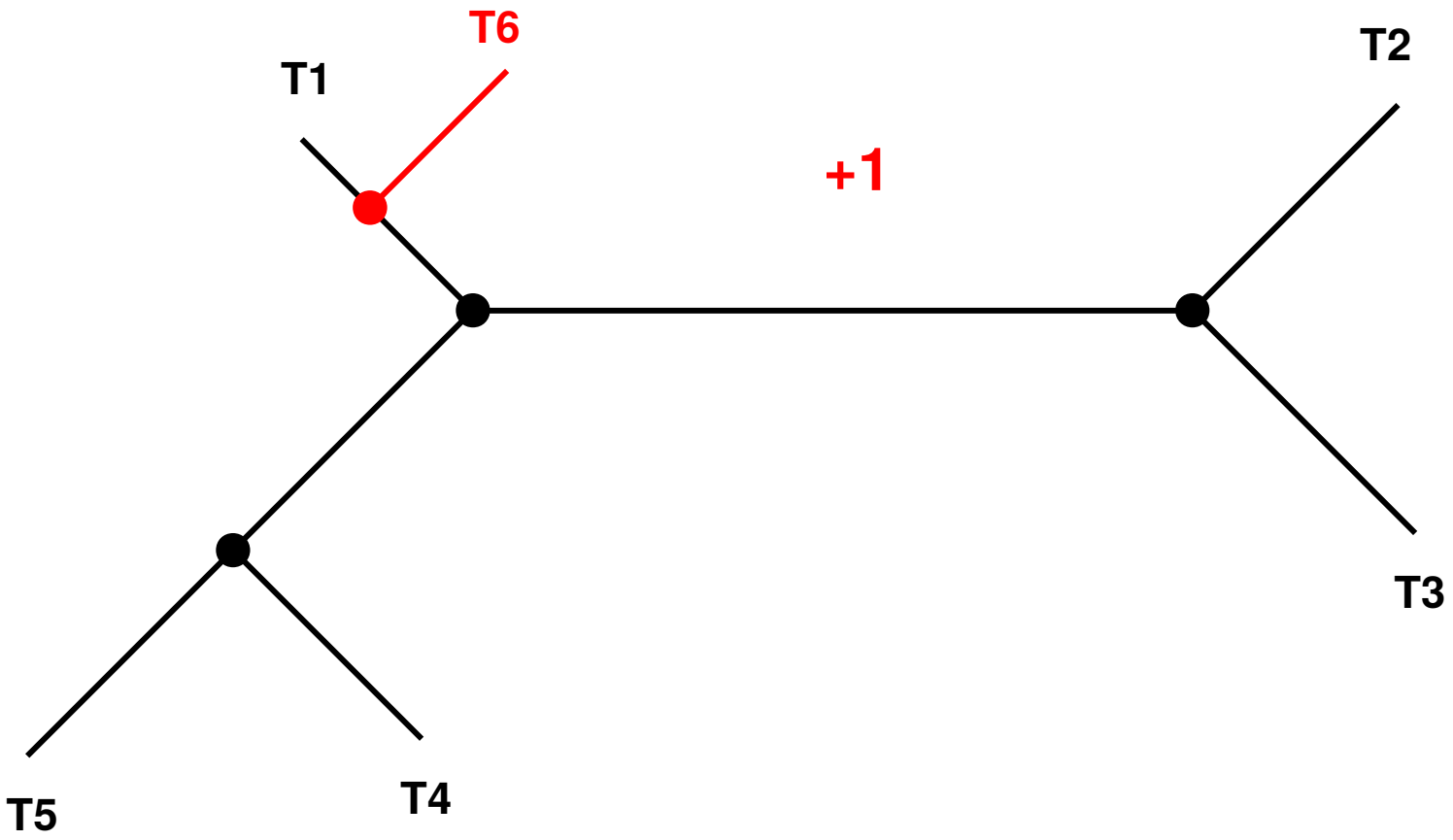
SPR



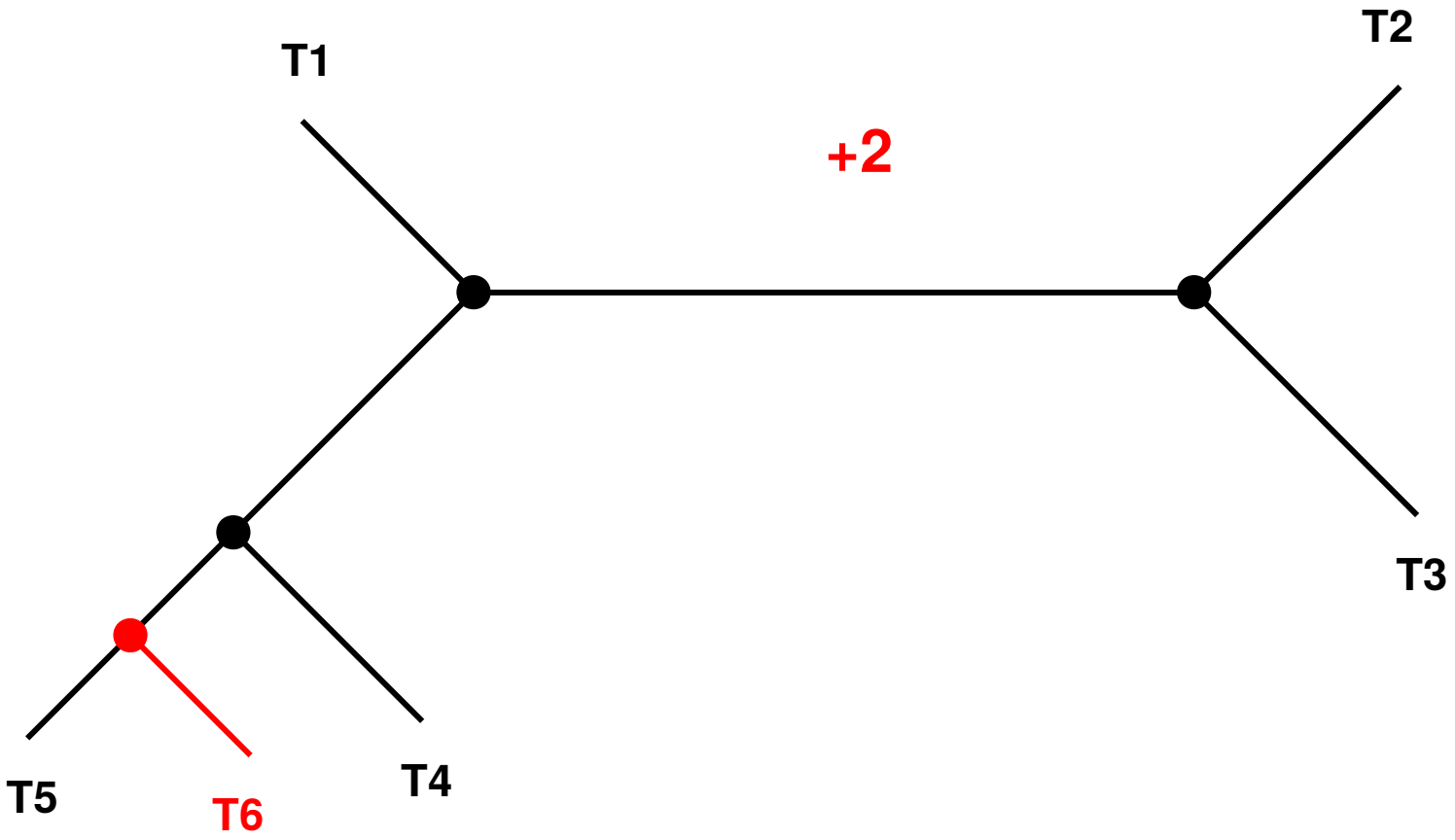
SPR



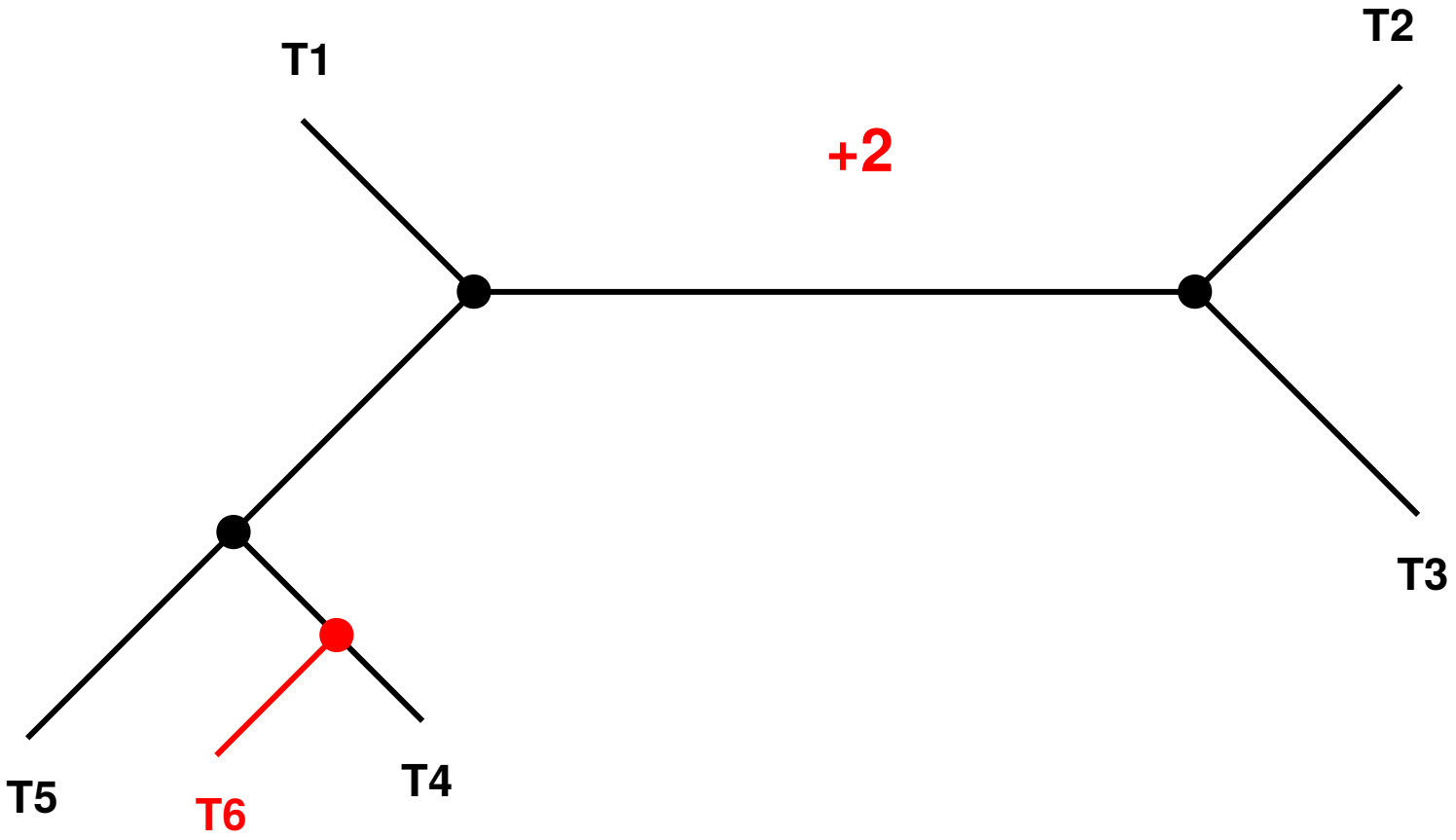
SPR



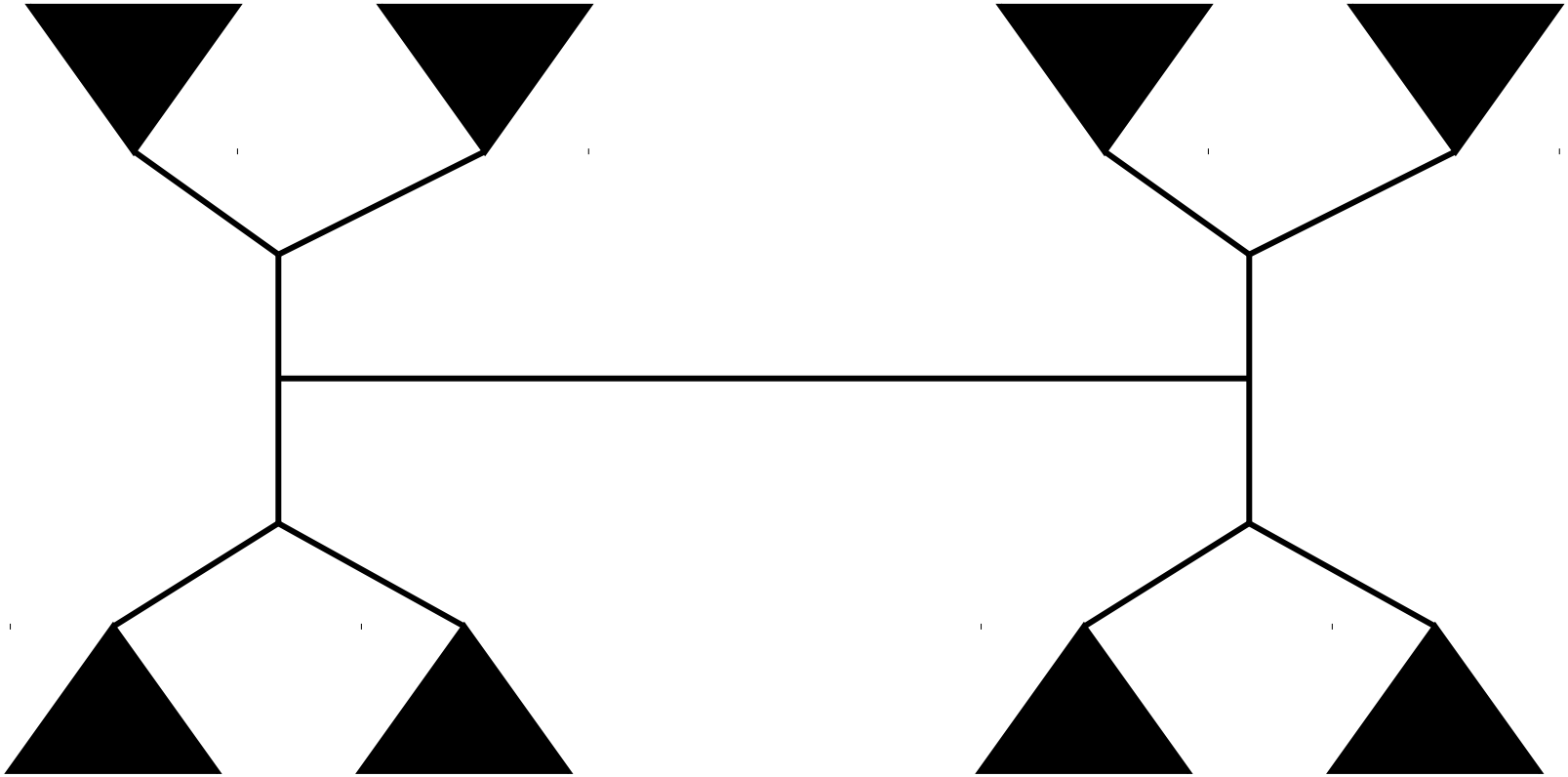
SPR



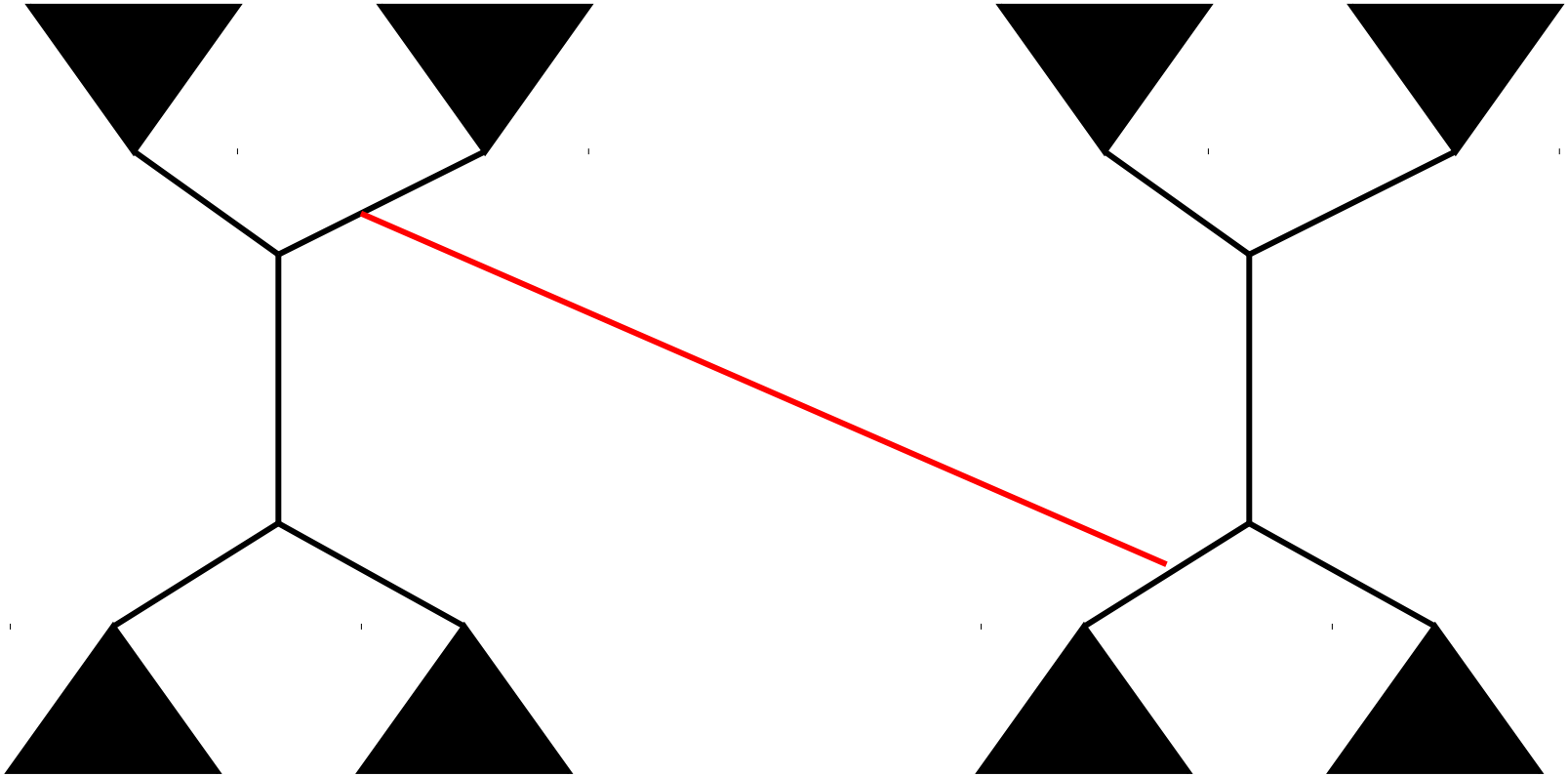
SPR



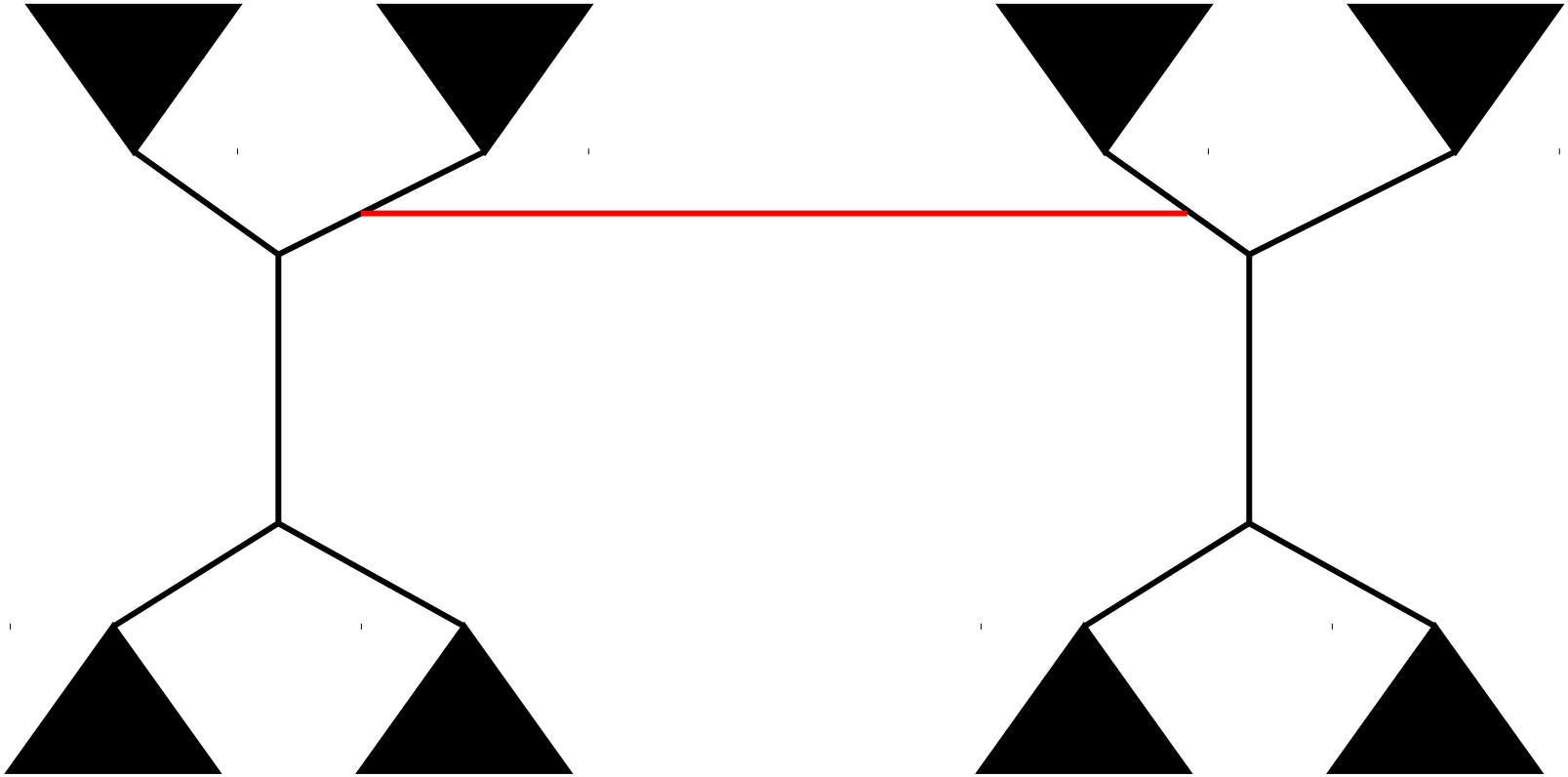
TBR



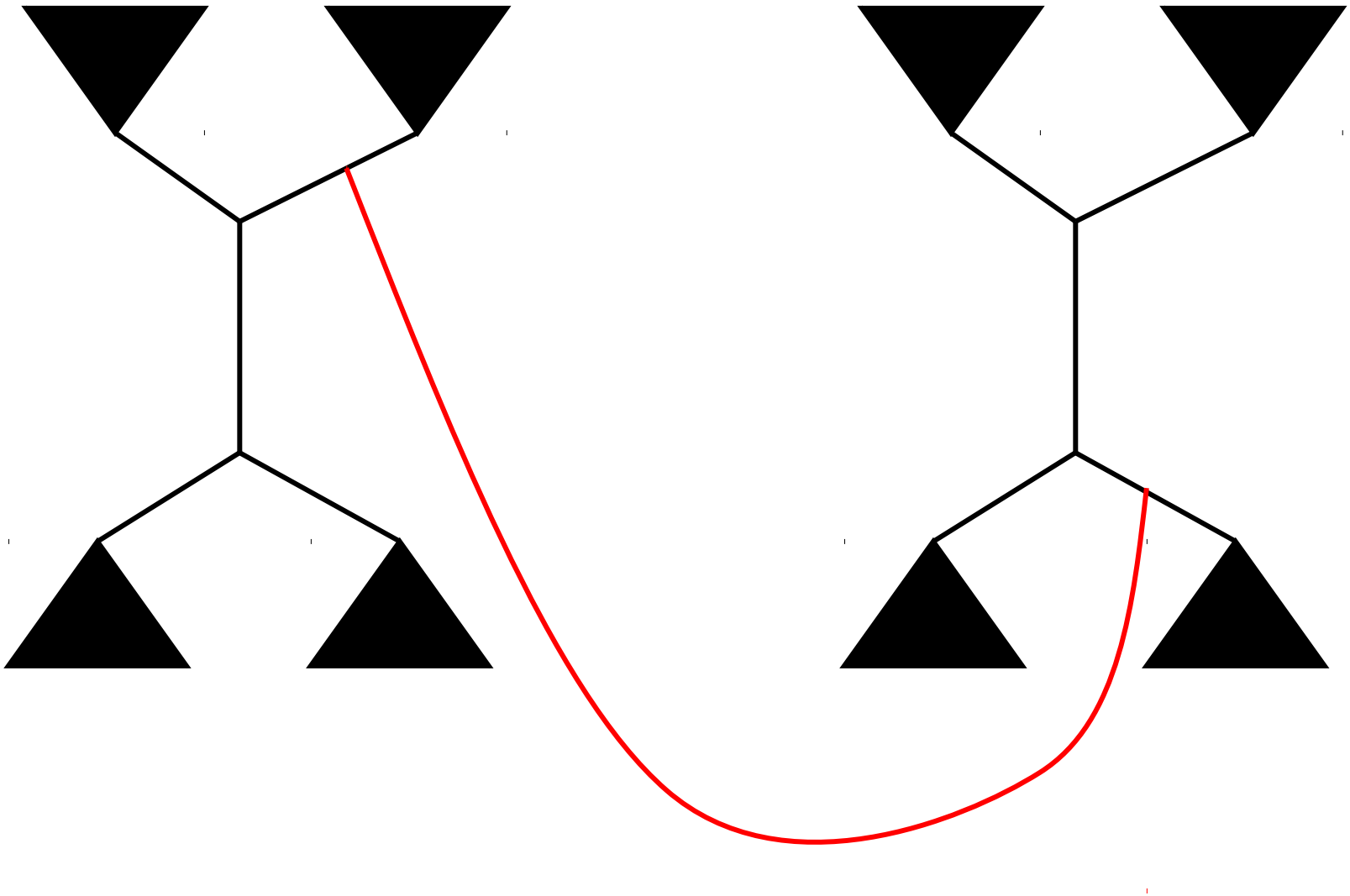
TBR



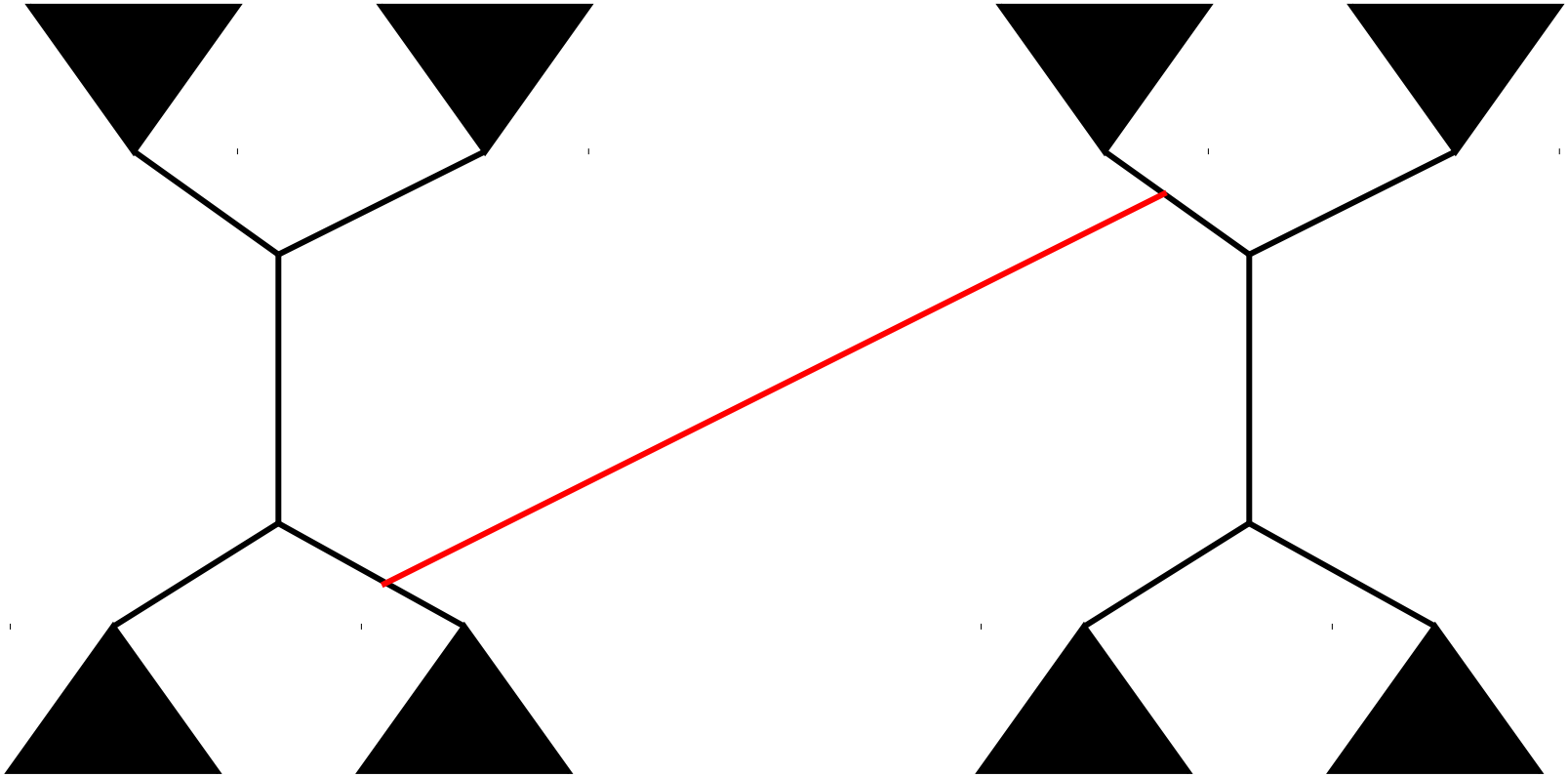
TBR



TBR



TBR



The Parsimonator Algorithm

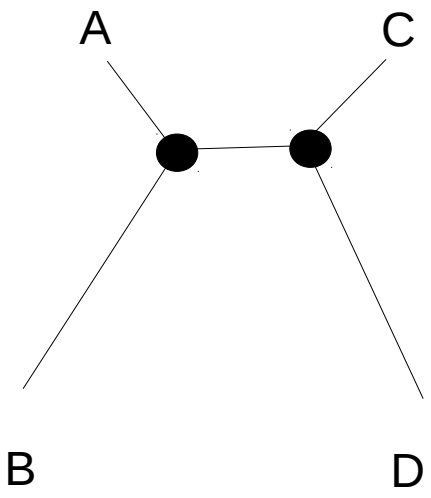
- Build a randomized stepwise addition order parsimony tree
- Apply SPR moves to all subtrees of the current (comprehensive) tree with a rearrangement radius of *20*
- If the rearrangement of a subtree yields an improved parsimony score, keep it immediately
  - → this is somewhat greedy as opposed to a steepest ascent hill climbing algorithm
- Continue applying SPR moves with a radius of *20* to all subtrees until no tree with a better parsimony score can be found
- There are much more sophisticated algorithms available
  - → TNT tool by Pablo Goloboff
- Keep in mind that parsimony returns discrete scores, that is, there may be many equally parsimonious trees among which we can not distinguish!

(Maximum) Likelihood Model on Phylogenies

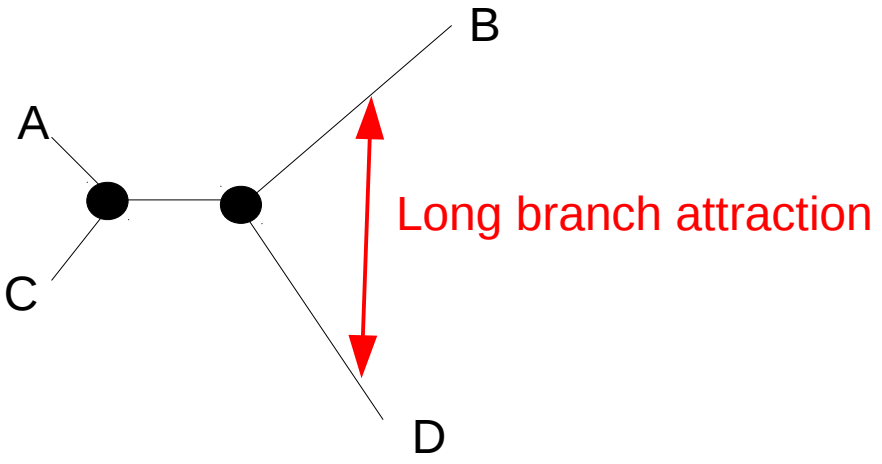
- Why do we need it?
- How do we compute the likelihood of a phylogeny?

Parsimony & Long Branch Attraction

- Because parsimony tries to minimize the number of mutations it faces some problems on trees with long branches



Correct tree

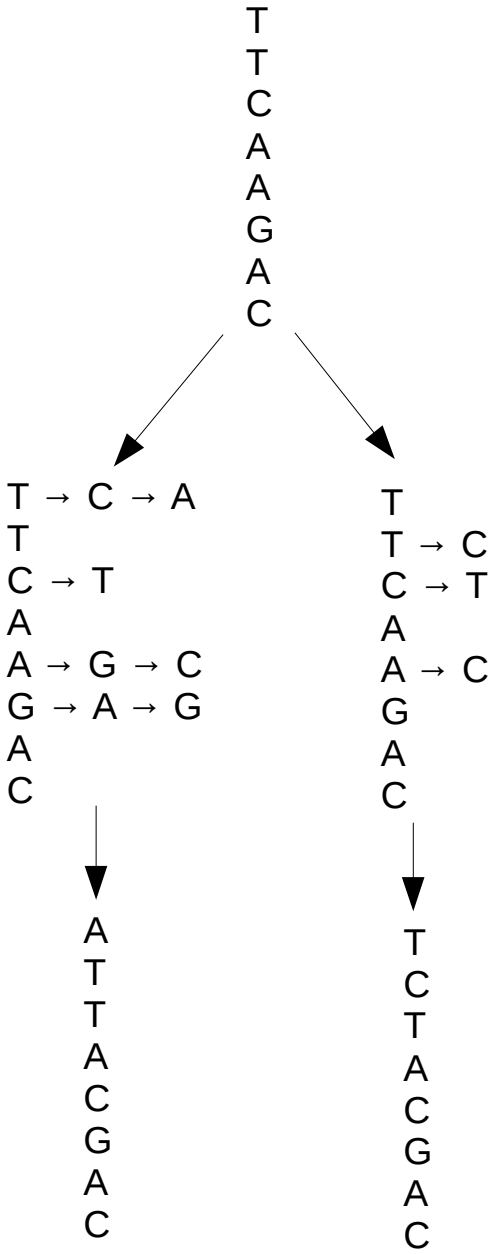


Wrong tree inferred by parsimony

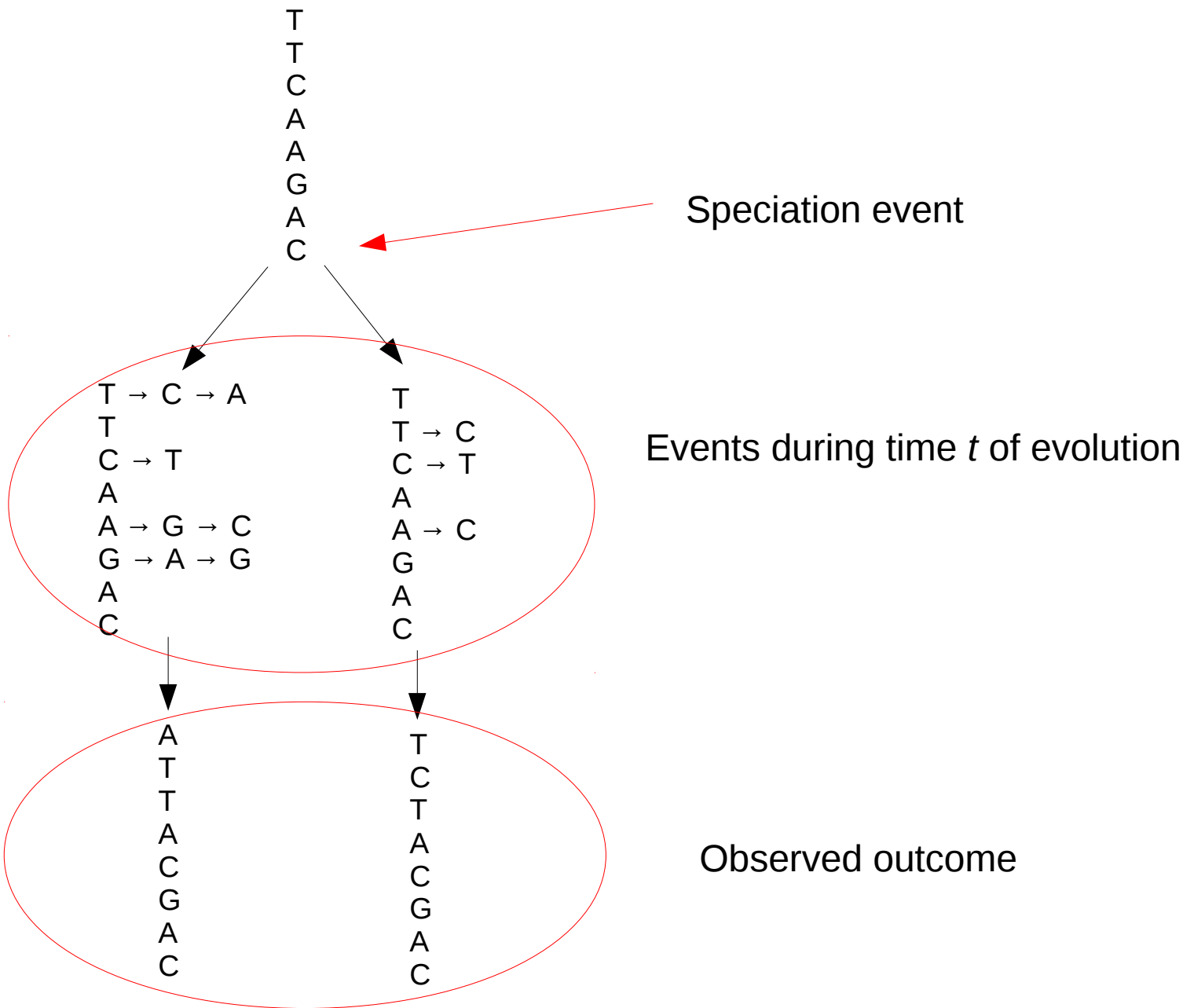
Parsimony & Long Branch Attraction

- Settings under which parsimony recovers the wrong tree are also called “the Felsenstein Zone” after *Joe Felsenstein* who has made numerous very important contributions to the field, e.g.
  - The Maximum Likelihood model
  - The Bootstrapping procedure
- If you are interested in statistics, there are some on-line courses by Joe at http://evolution.gs.washington.edu/courses.html

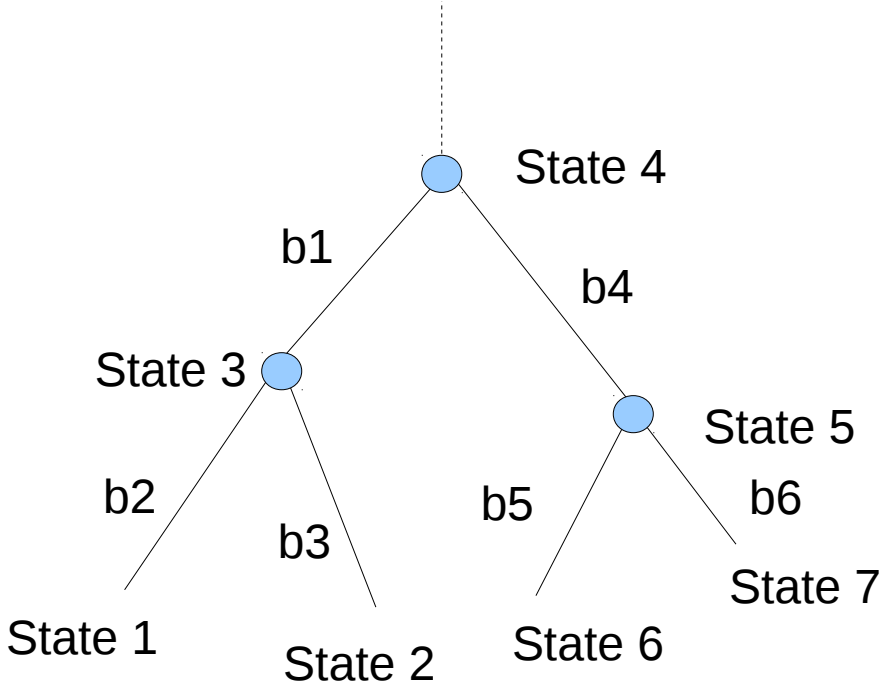
Likelihood tries to fix this Problem



Likelihood tries to fix this Problem



How do we compute the likelihood?



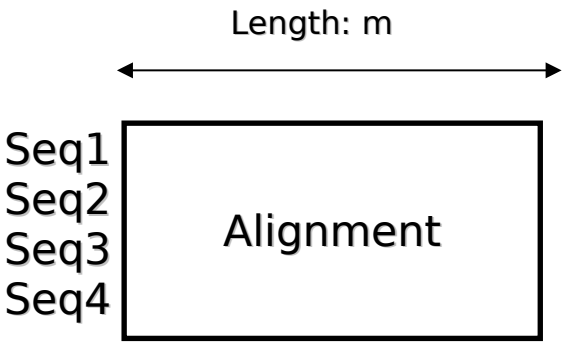
For all the rest, see blackboard!

Back from the blackboard

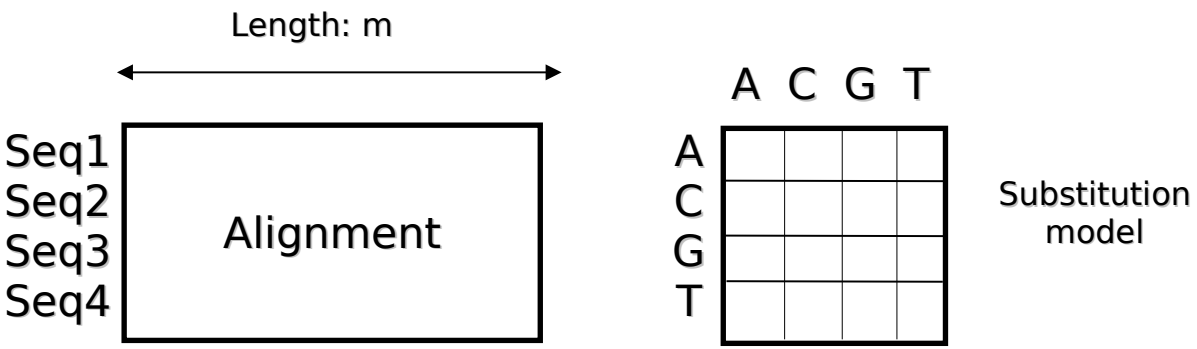
- Knowledge quiz results
  - Especially at the technical level there seem to be some lack of knowledge (NUMA, ILP, etc.)
  - Time permitting I will insert a lecture in the new year
- Exam Registration
  - Did everybody receive the link to the poll with the exam slots?
  - https://terminplaner2.dfn.de/foodle/Bio-Exam-5a2e4

Maximum Likelihood Overview

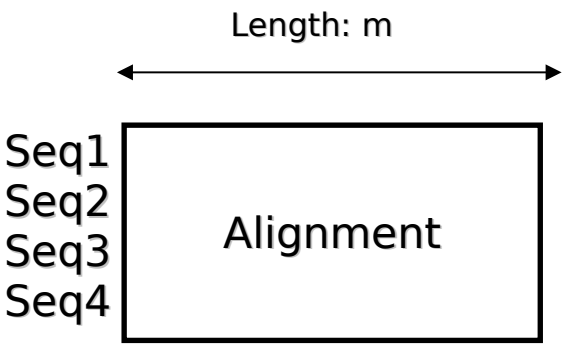
Maximum Likelihood

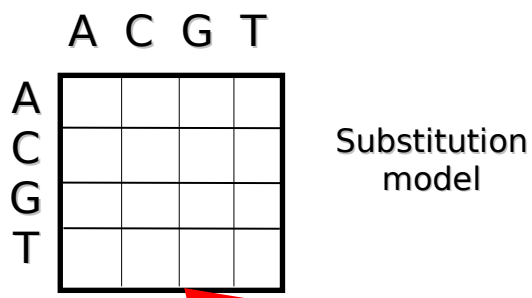


Maximum Likelihood



Maximum Likelihood

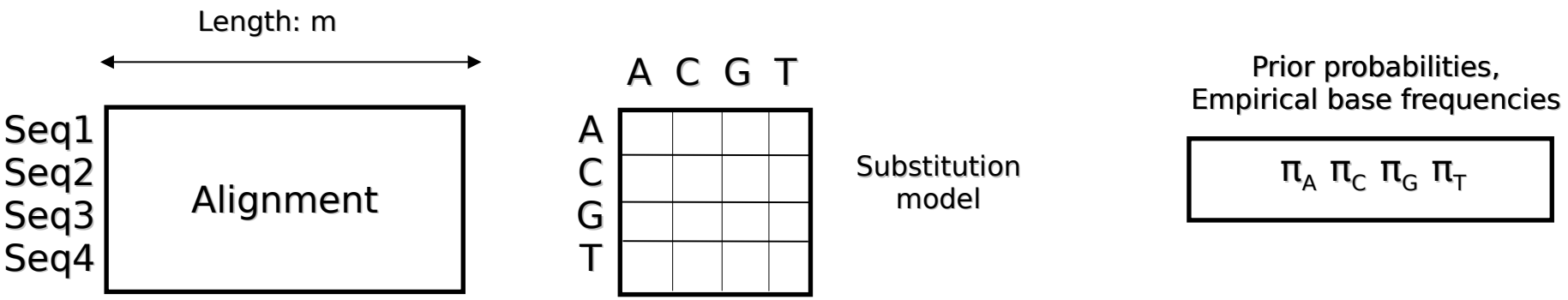




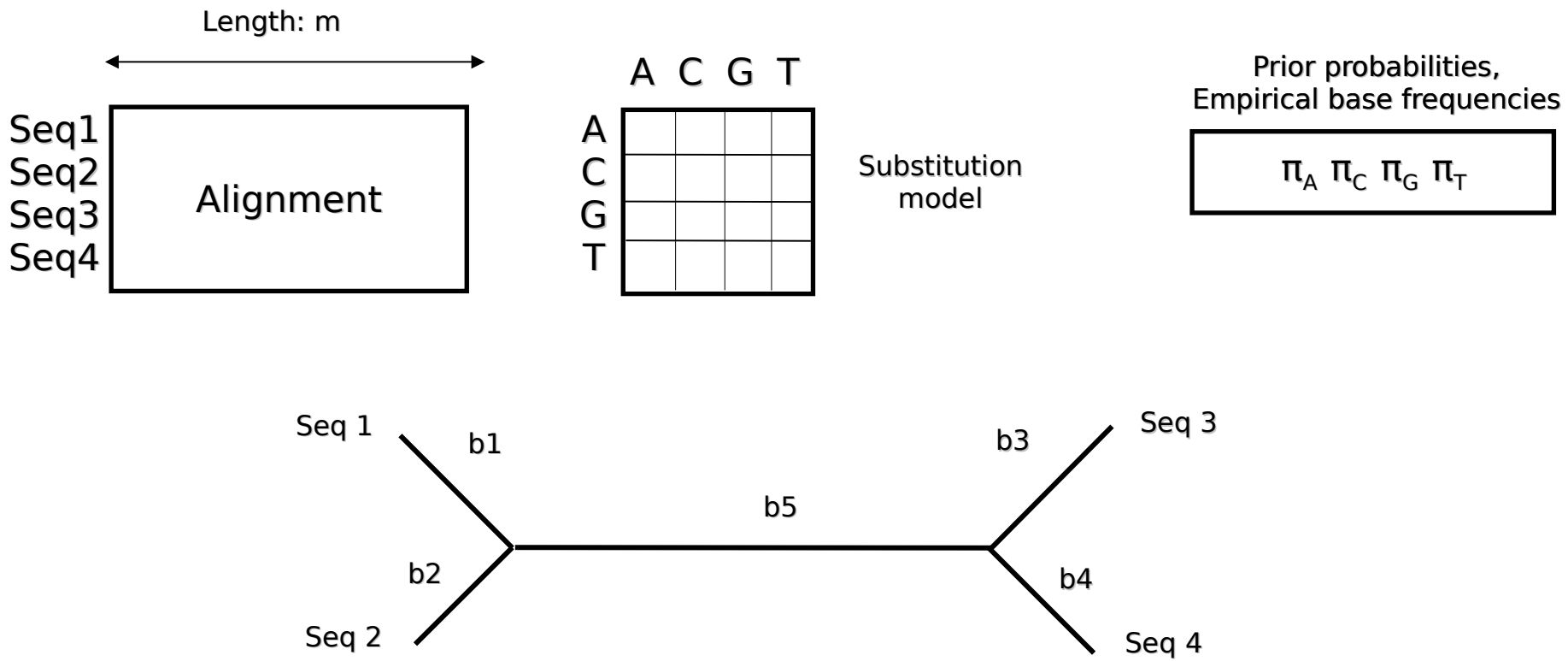
Substitution model

Commonly denoted as *Q* matrix: transition probs for time *dt*, for time *t*: $P(t)=e^{Qt}$

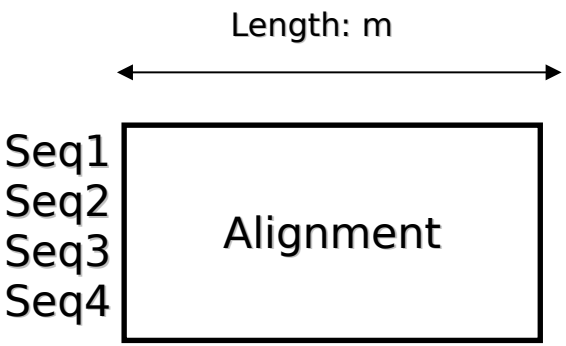
Maximum Likelihood

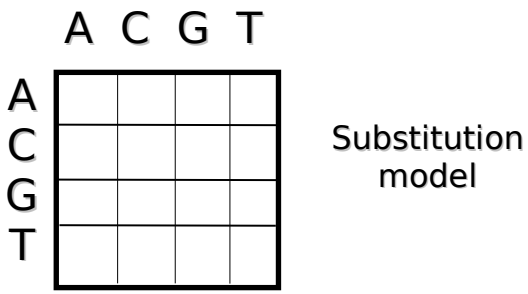


Maximum Likelihood



Maximum Likelihood

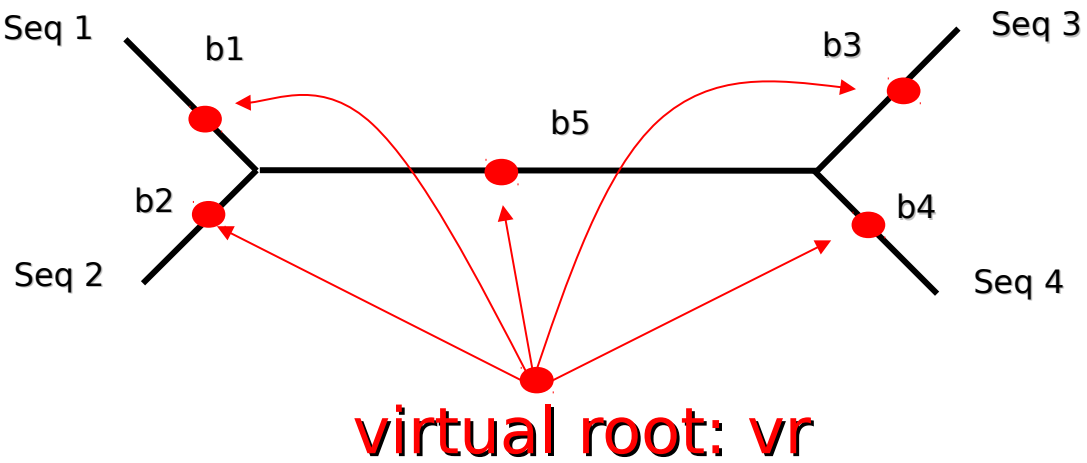




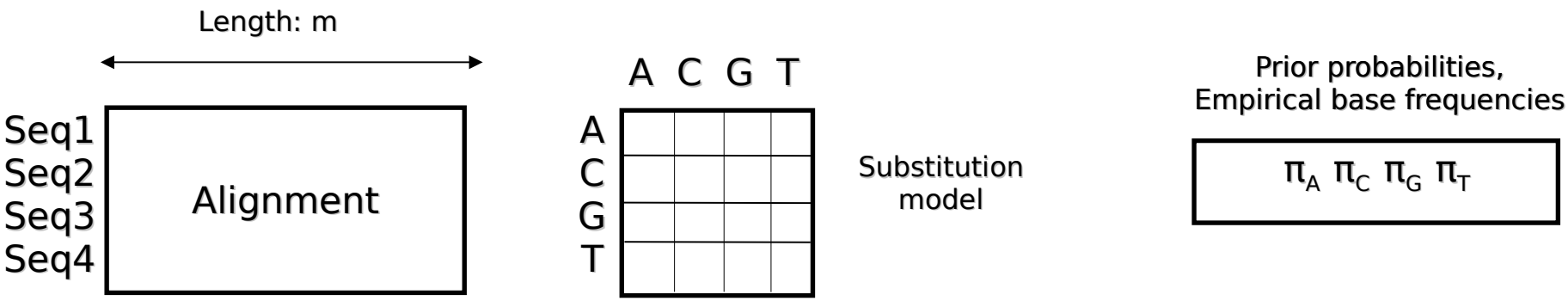
Substitution model

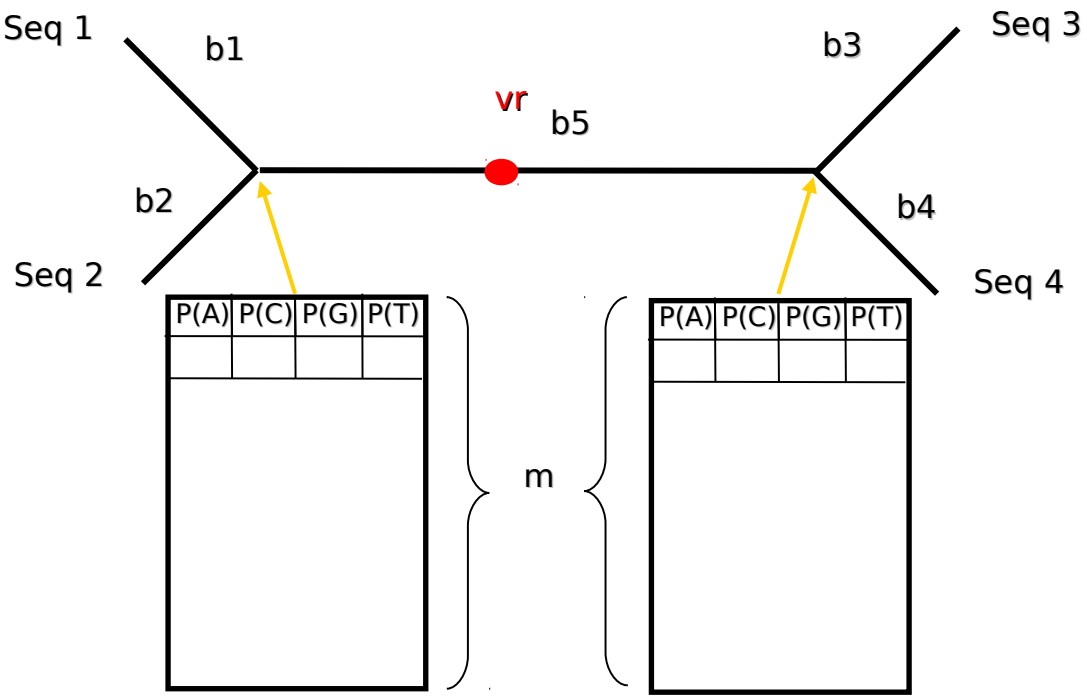
Prior probabilities,
Empirical base frequencies

π_A π_C π_G π_T

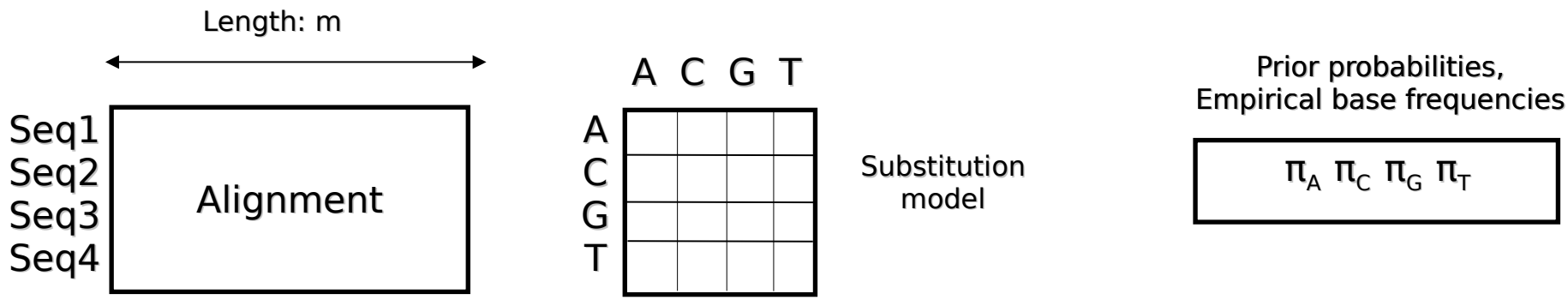


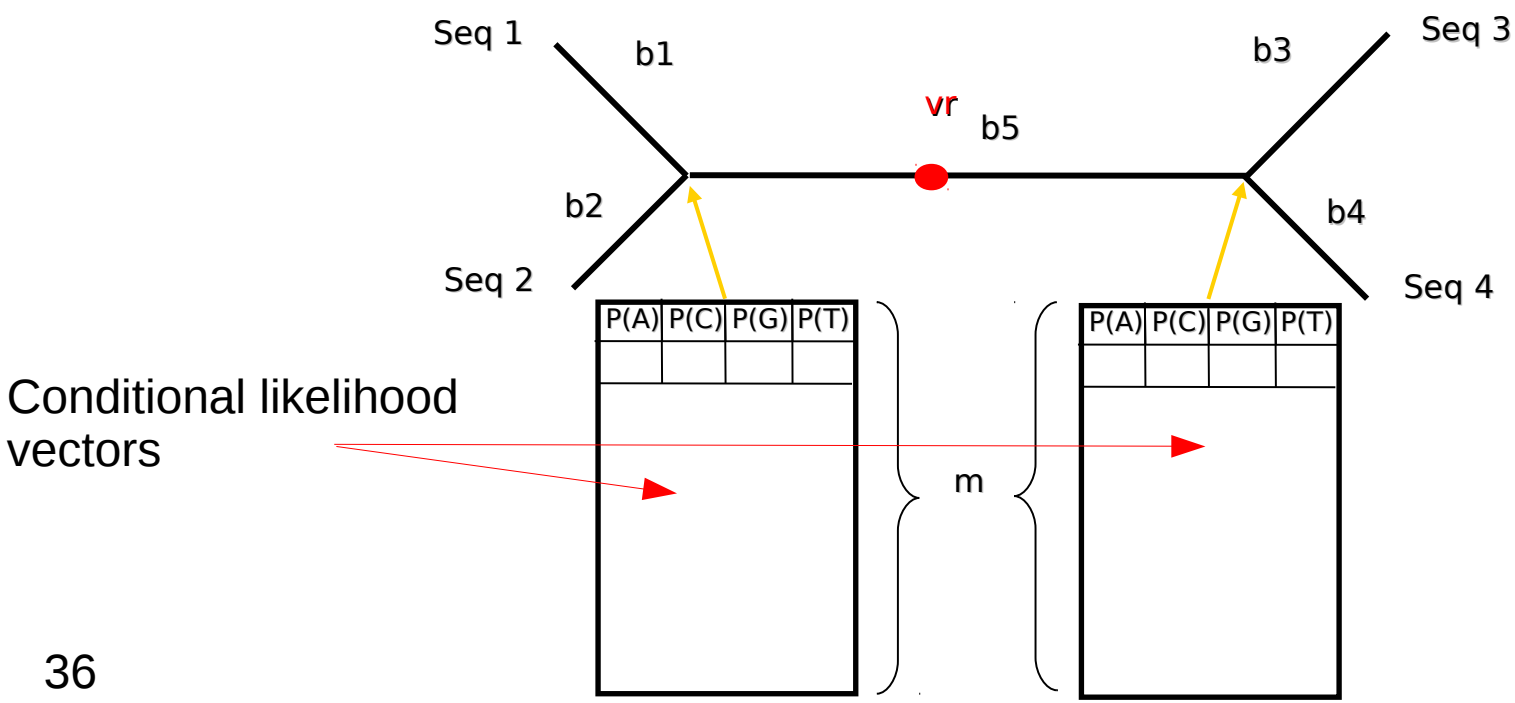
Maximum Likelihood



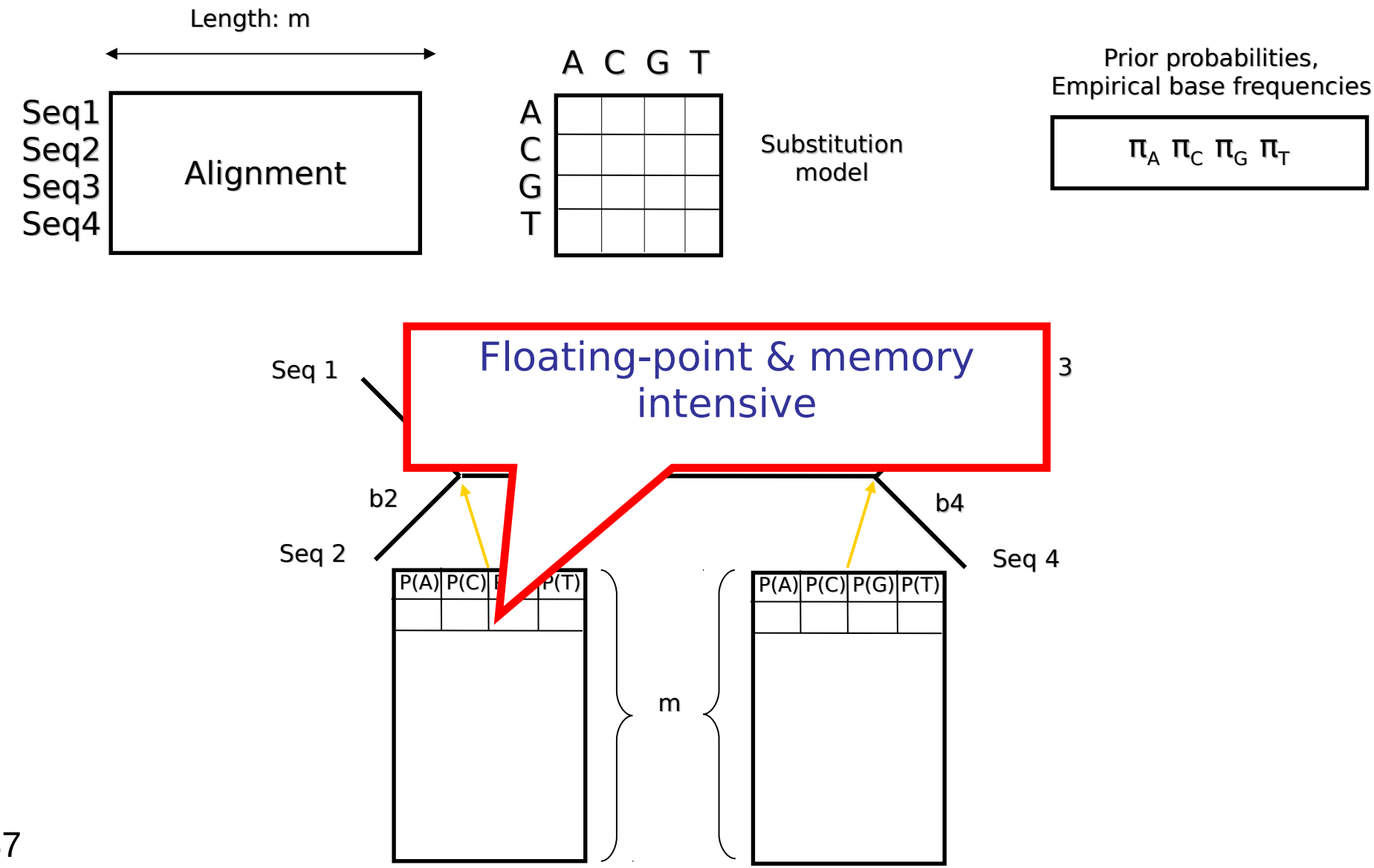


Maximum Likelihood



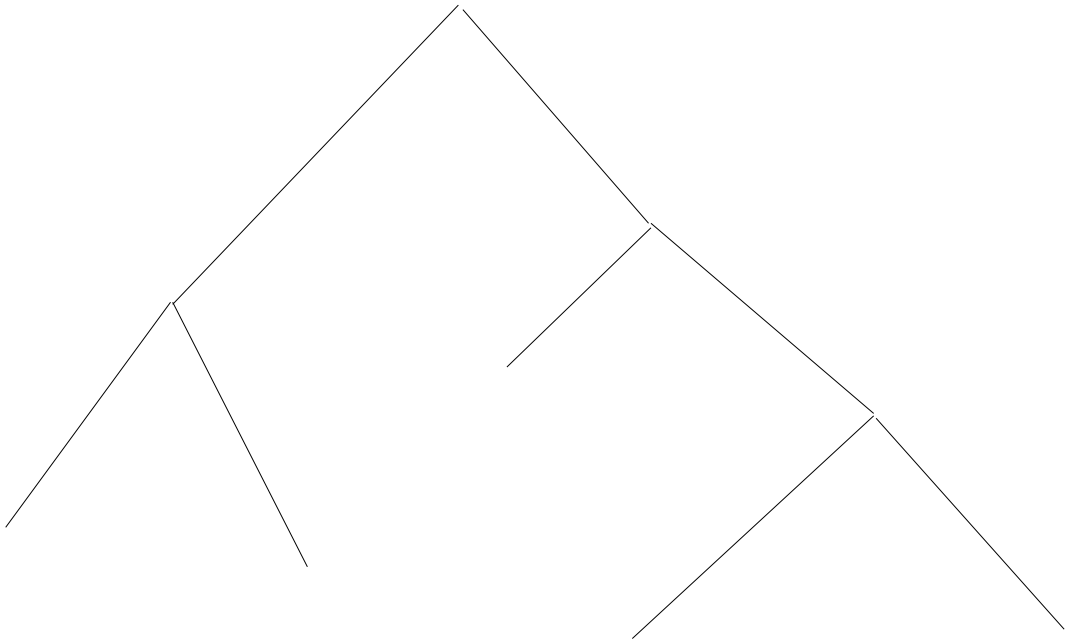


Maximum Likelihood

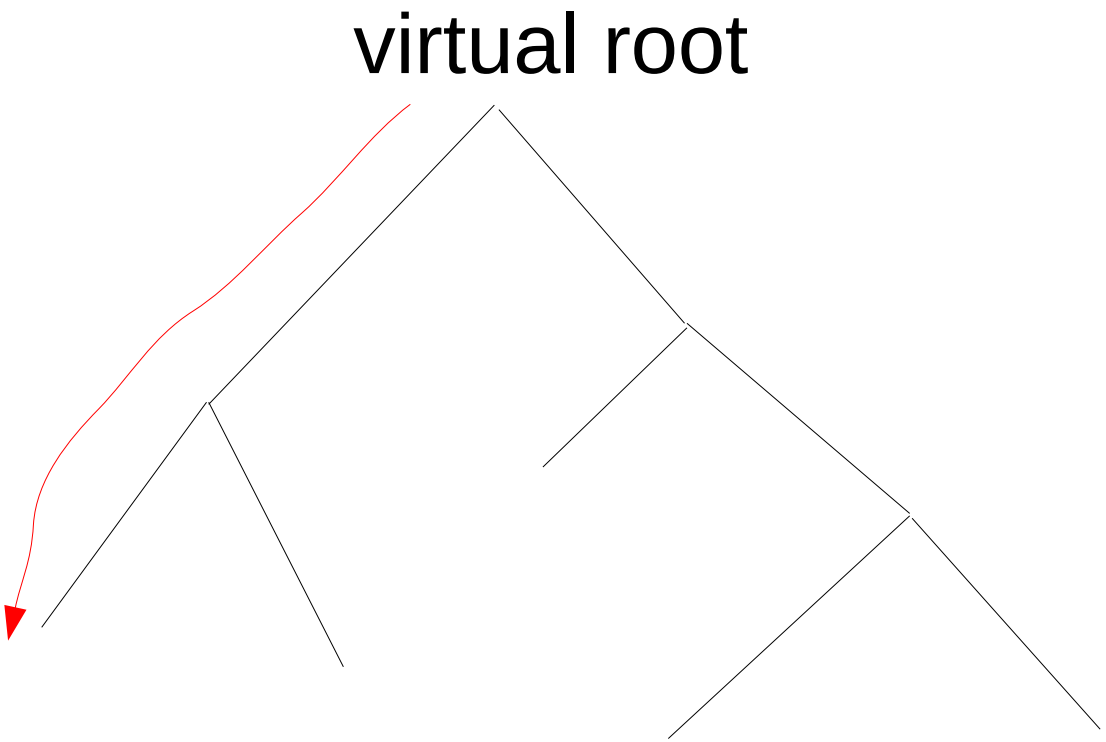


Post-order Traversal

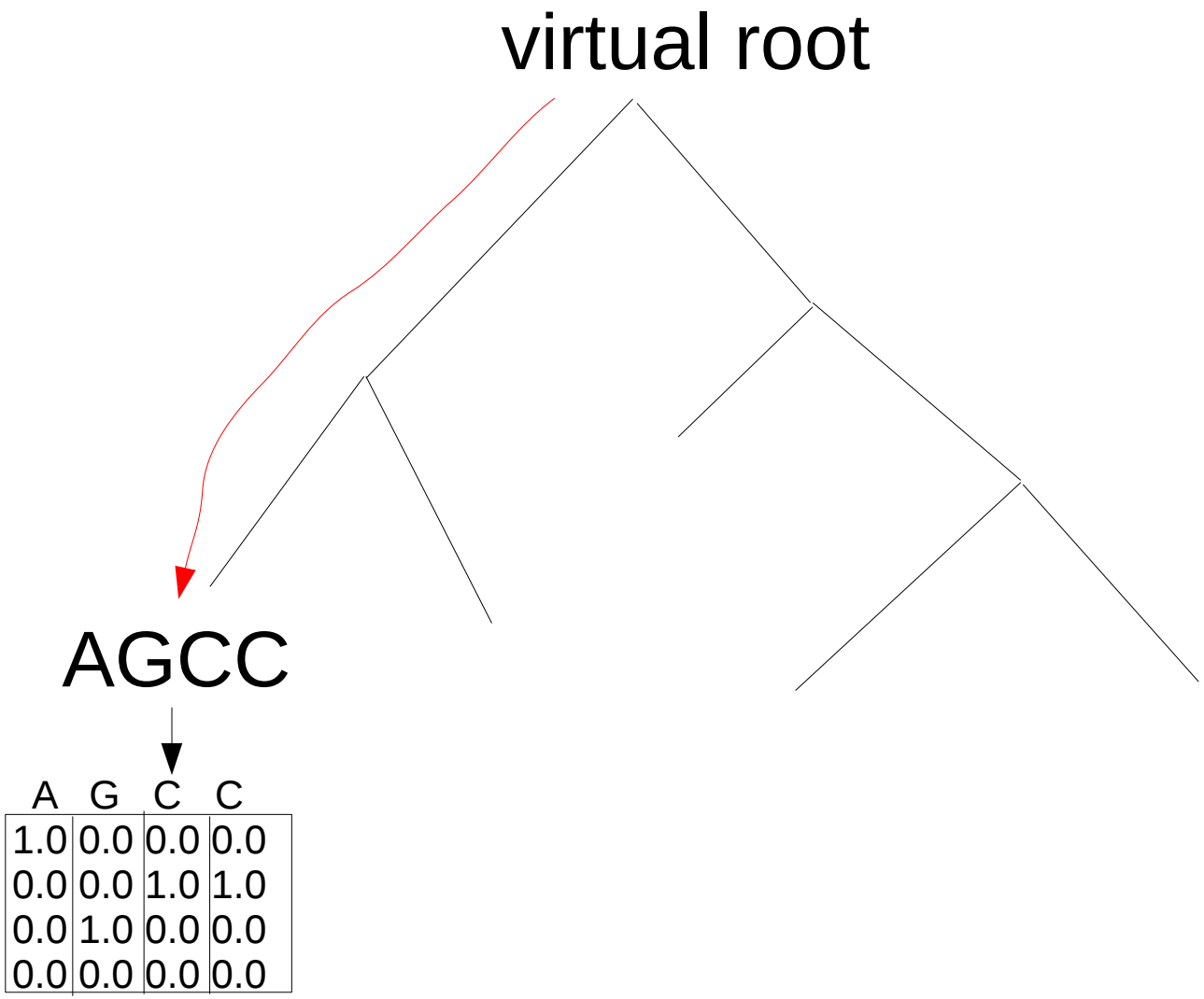
virtual root



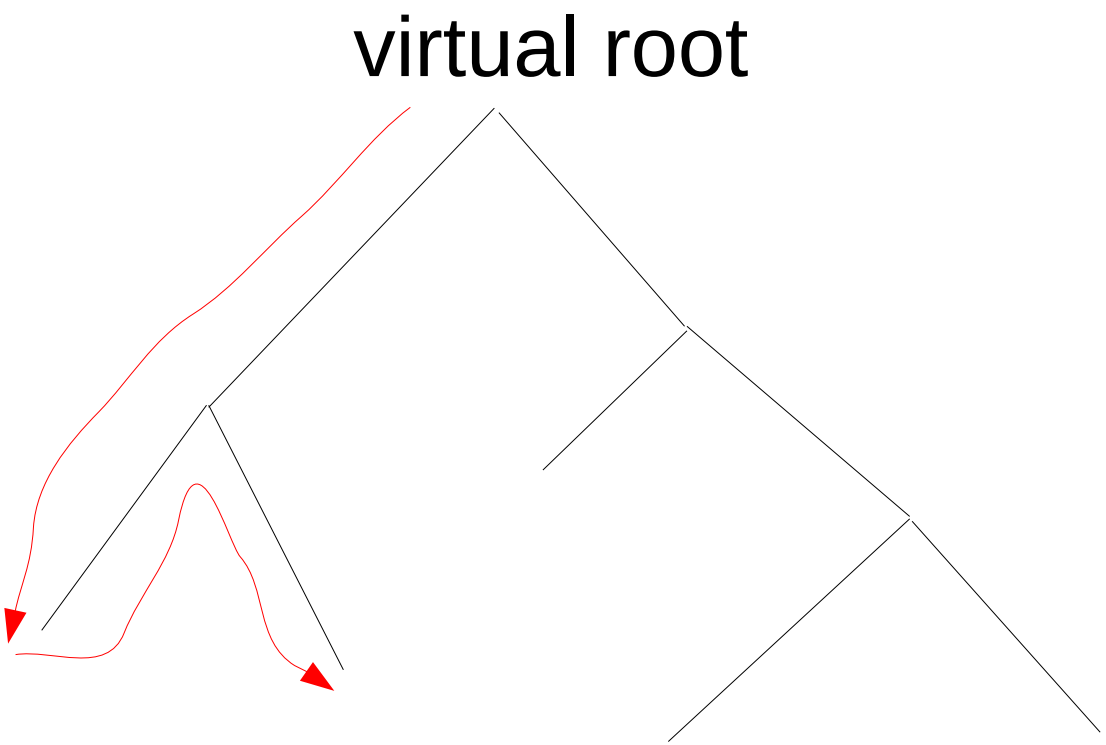
Post-order Traversal



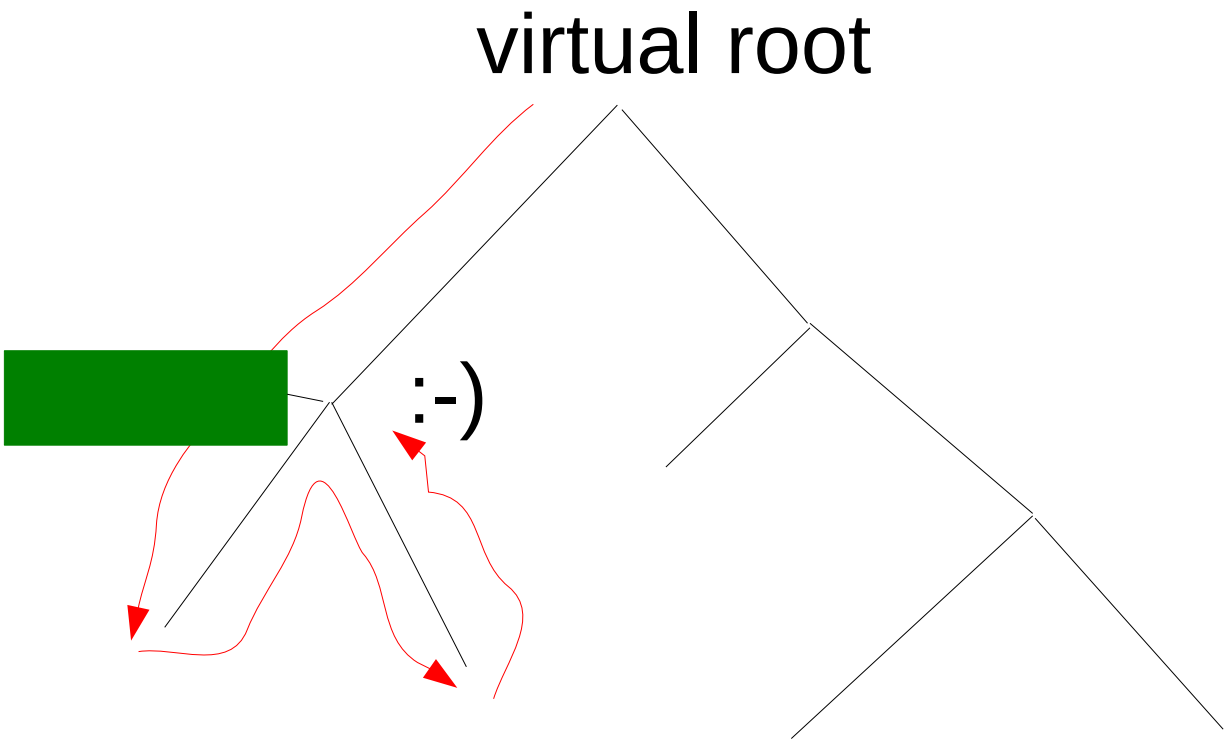
Post-order Traversal



Post-order Traversal

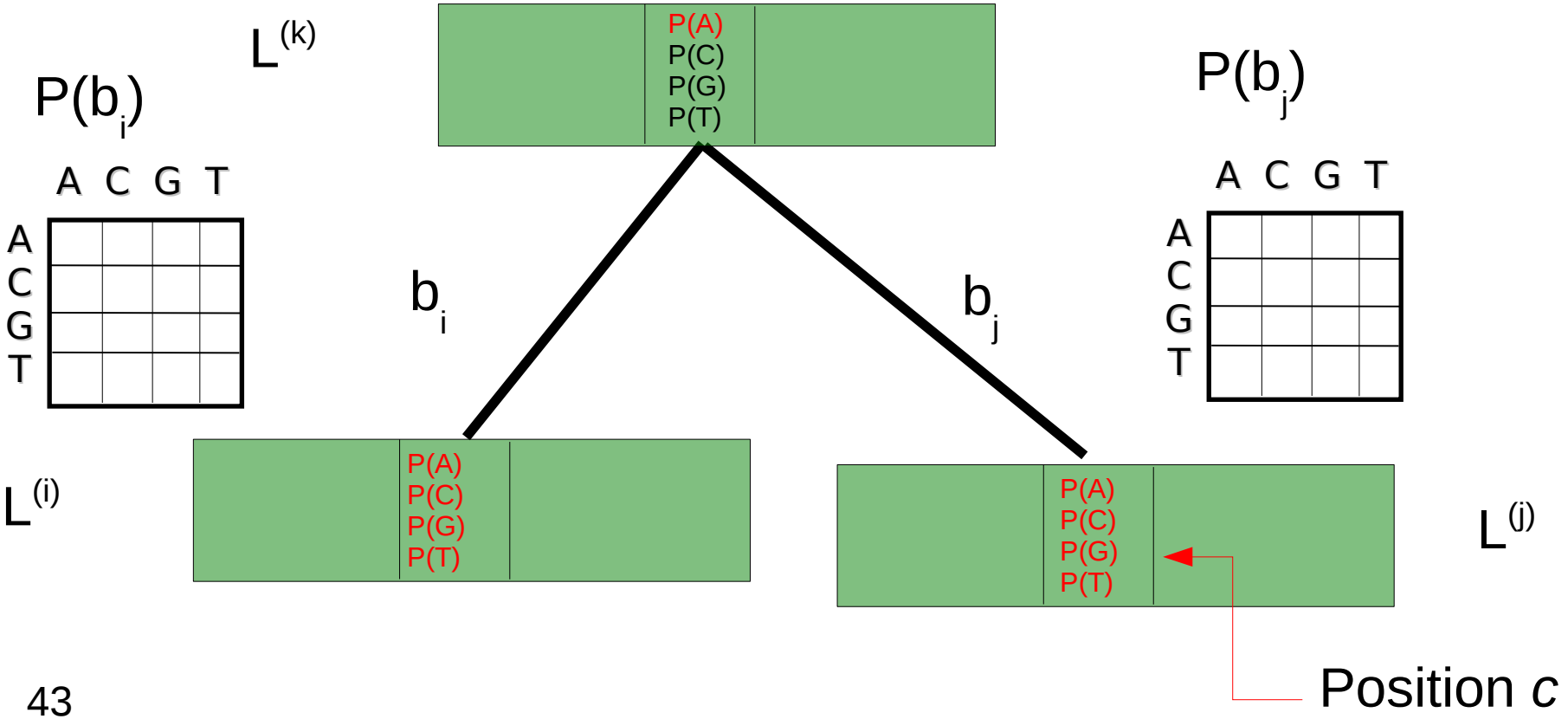


Post-order Traversal

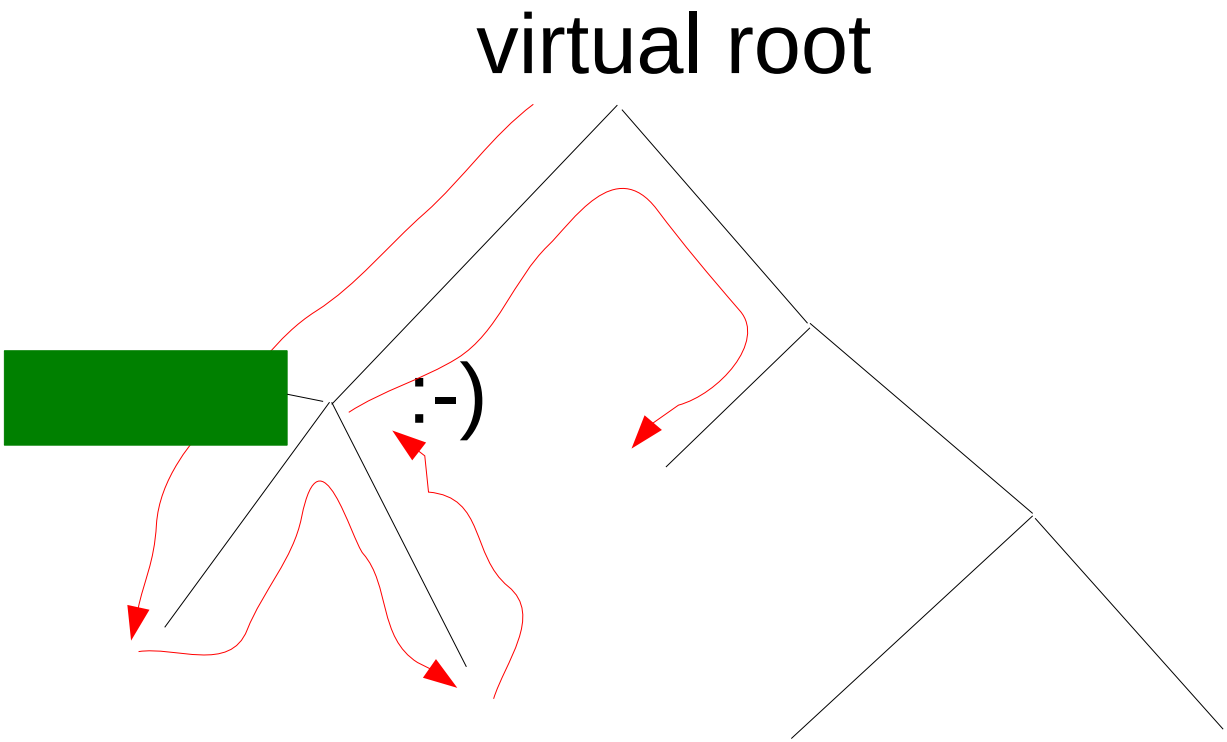


What happens when we compute this inner vector?

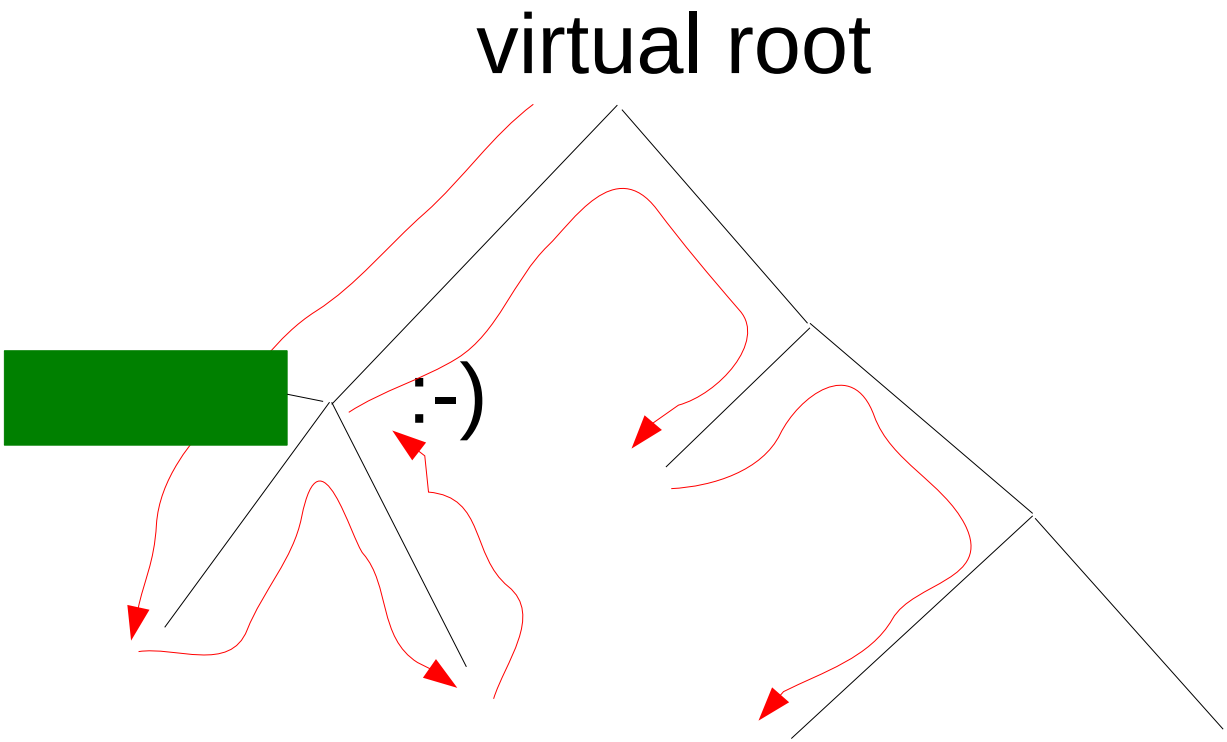
$$\vec{L}_A^{(k)}(c) = \left(\sum_{S=A}^{T} P_{AS}(b_i)\vec{L}_S^{(i)}(c)\right)\left(\sum_{S=A}^{T} P_{AS}(b_j)\vec{L}_S^{(j)}(c)\right)$$



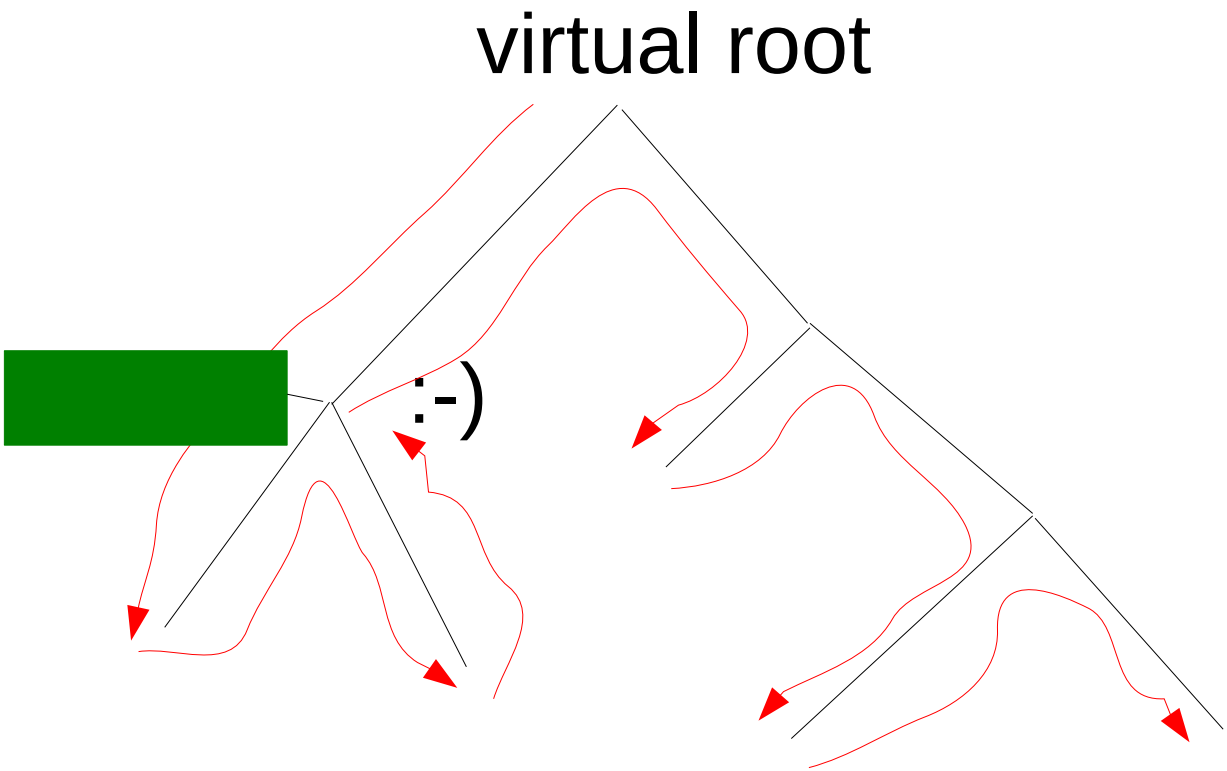
Post-order Traversal



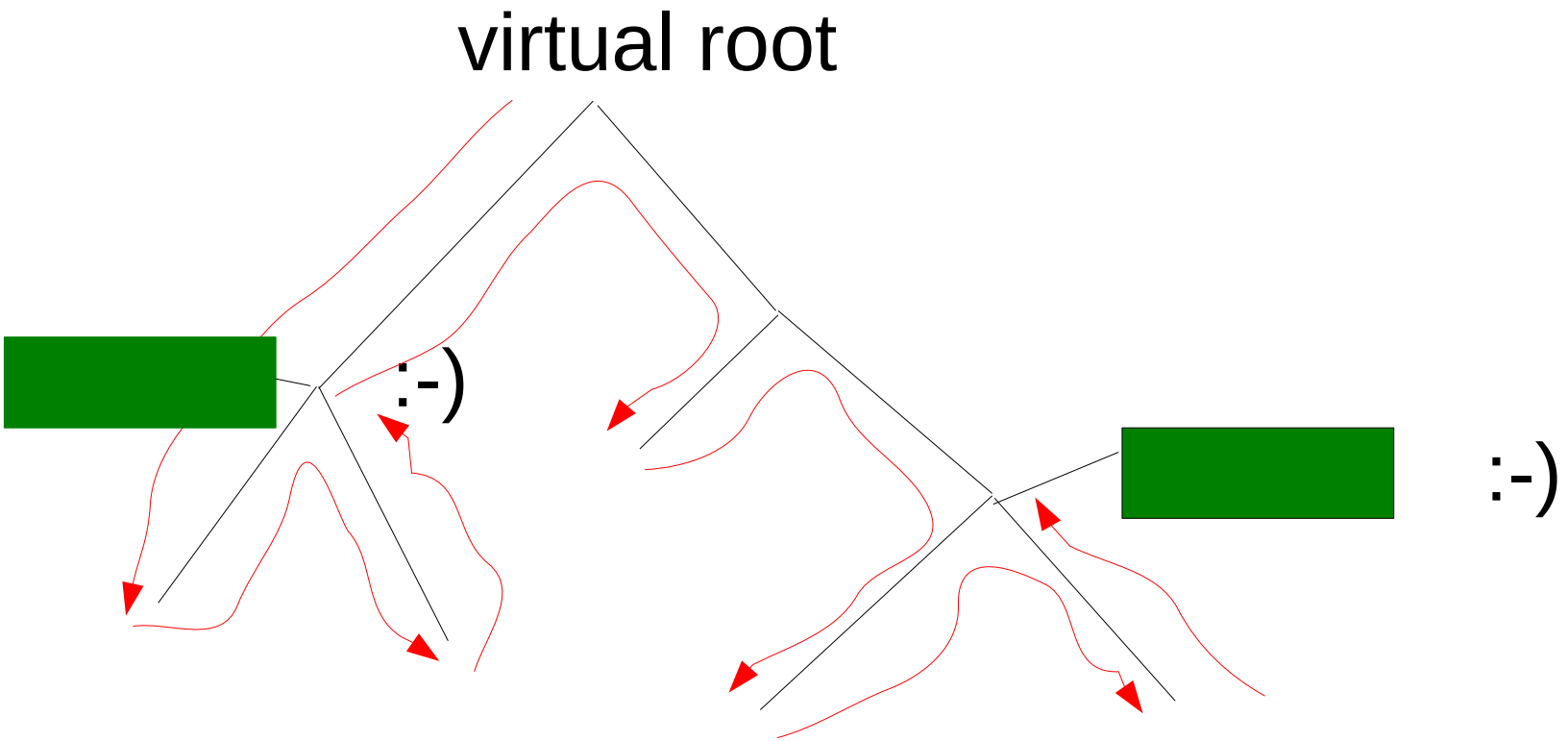
Post-order Traversal



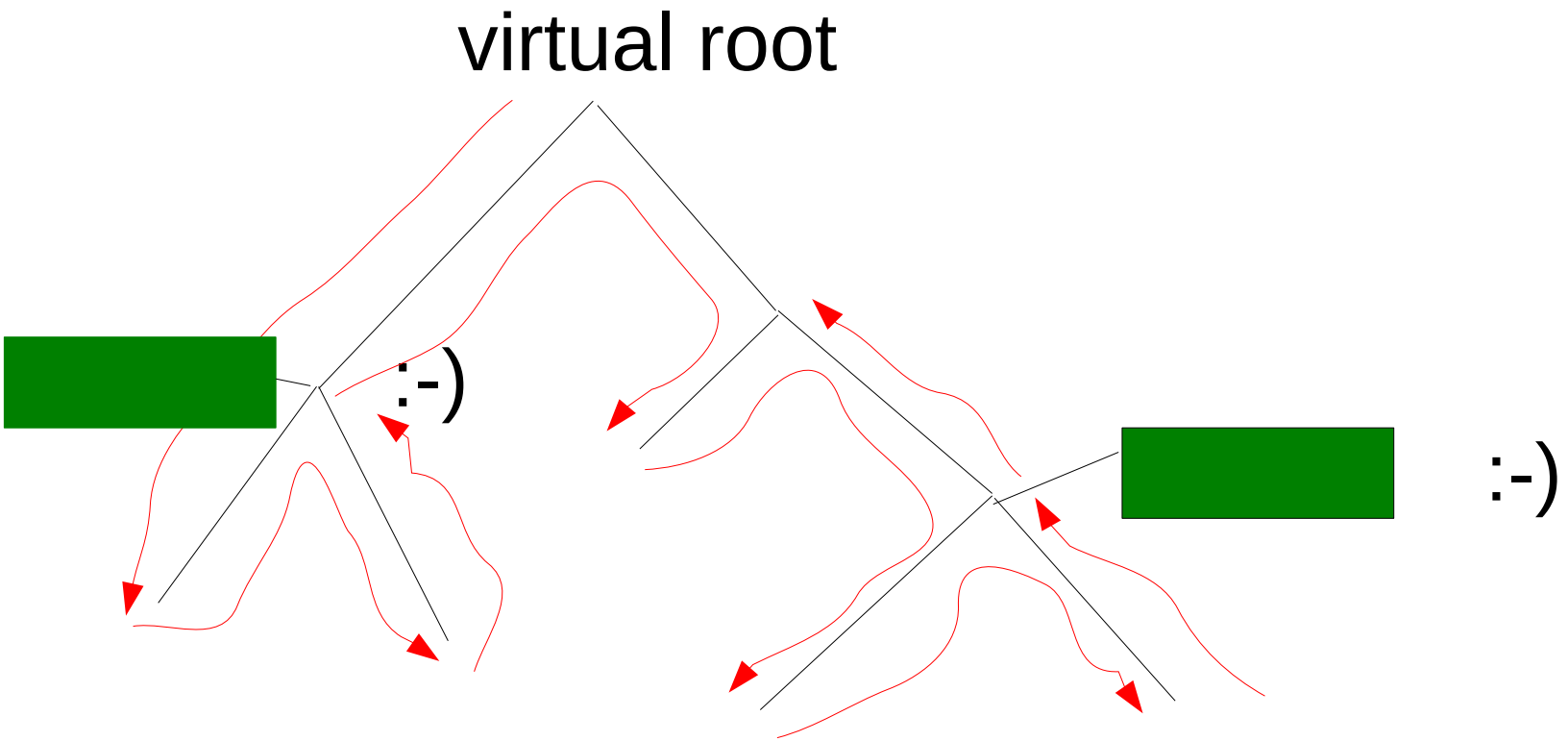
Post-order Traversal



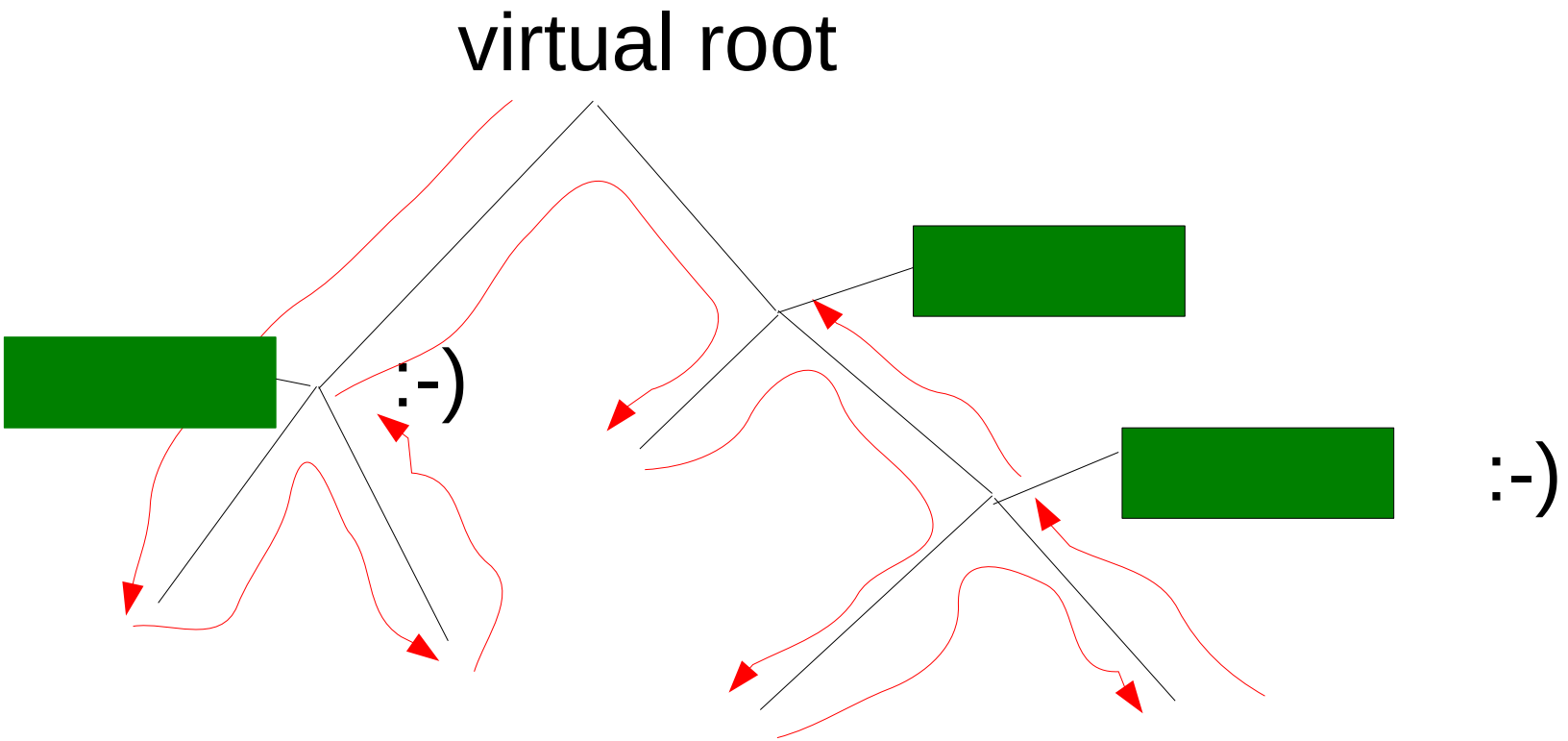
Post-order Traversal



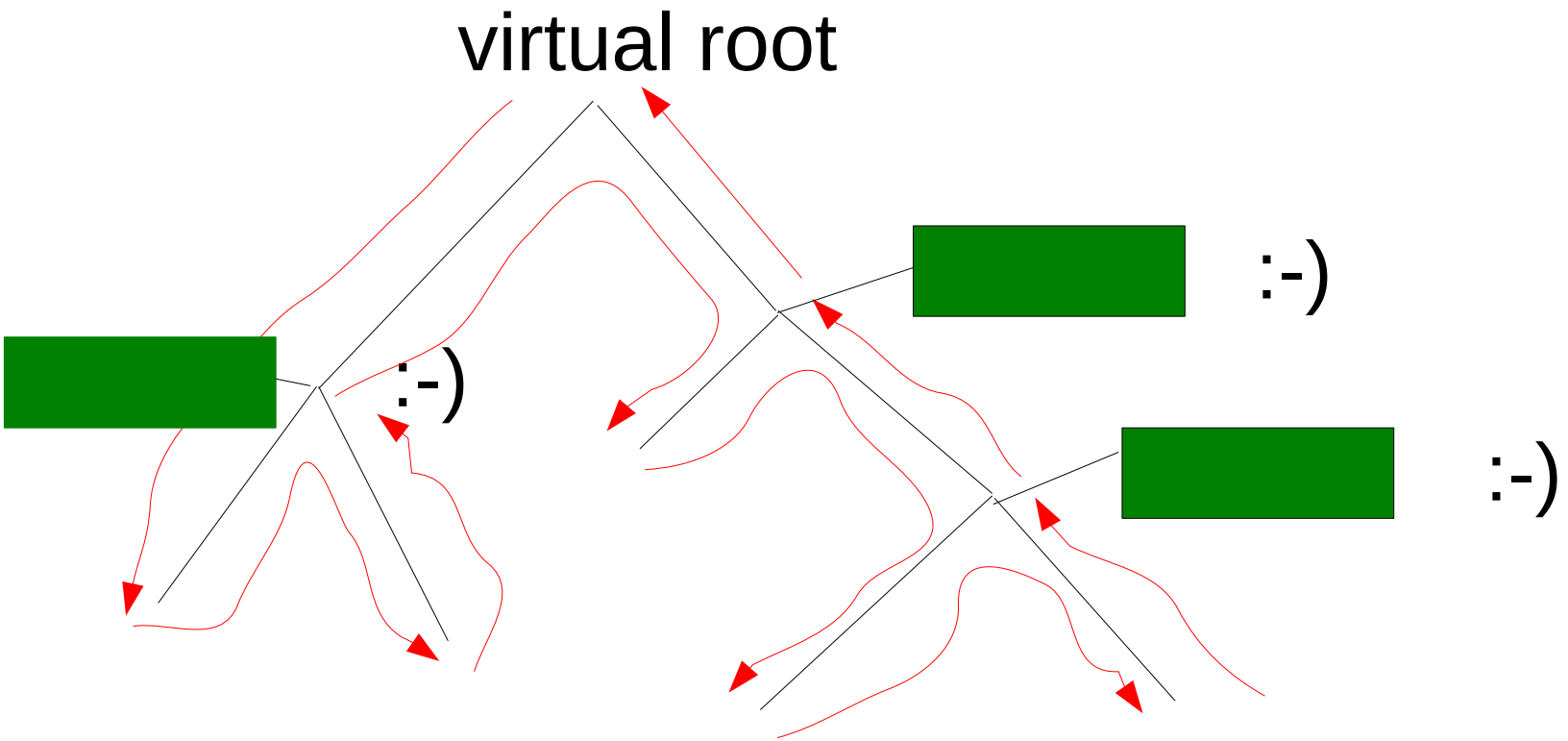
Post-order Traversal



Post-order Traversal

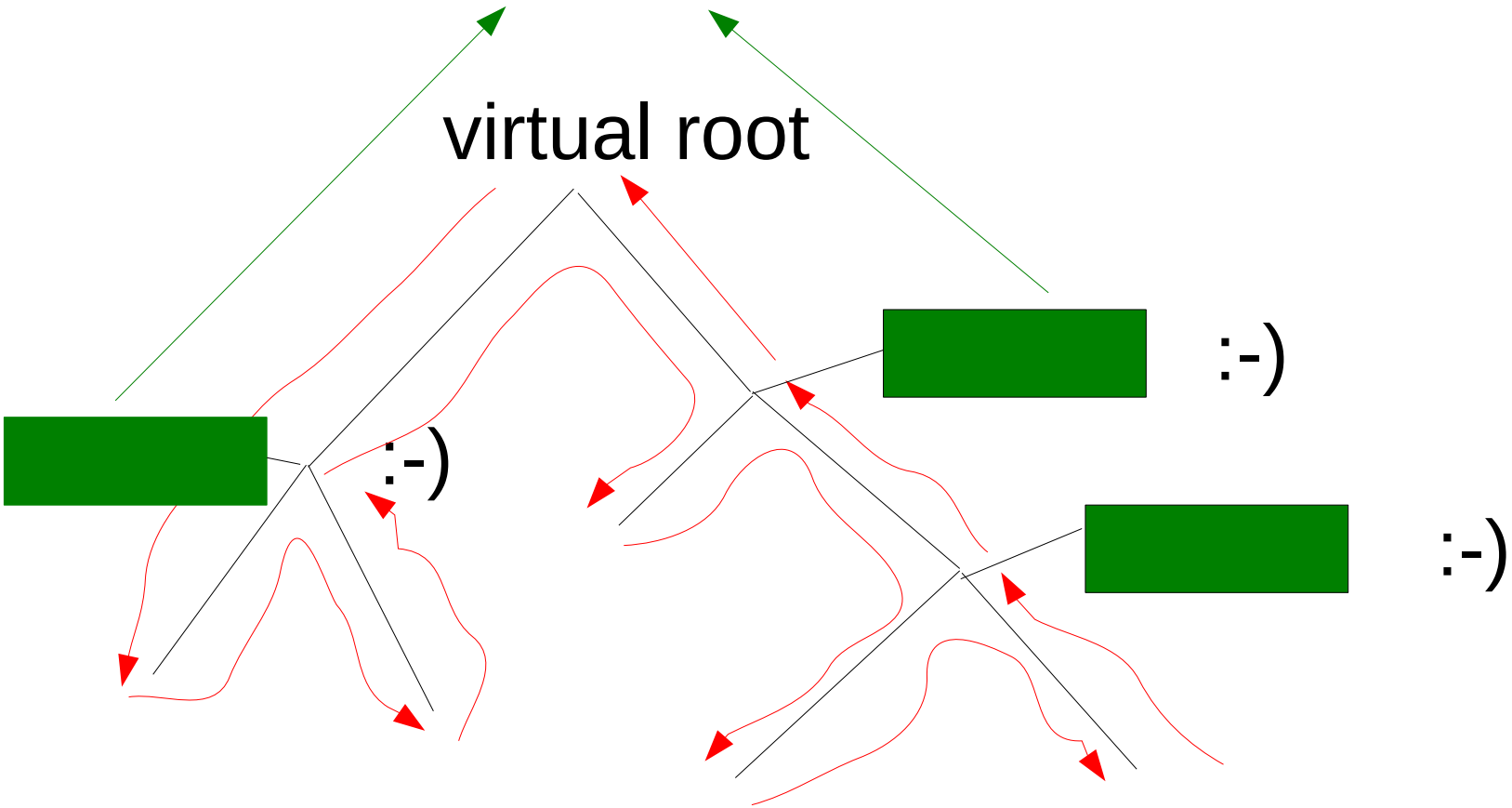


Post-order Traversal



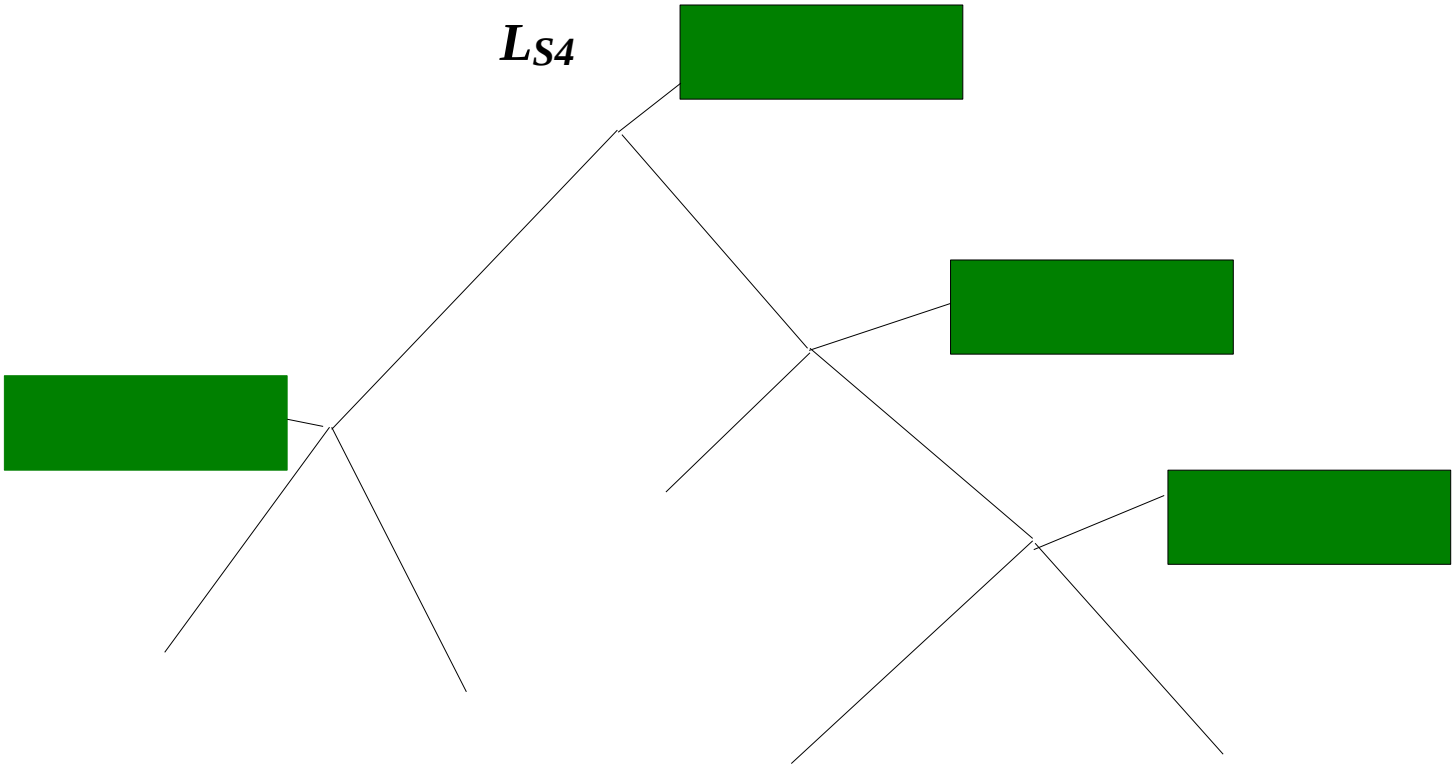
Post-order Traversal

Overall likelihood: sum over logs of per-site likelihoods



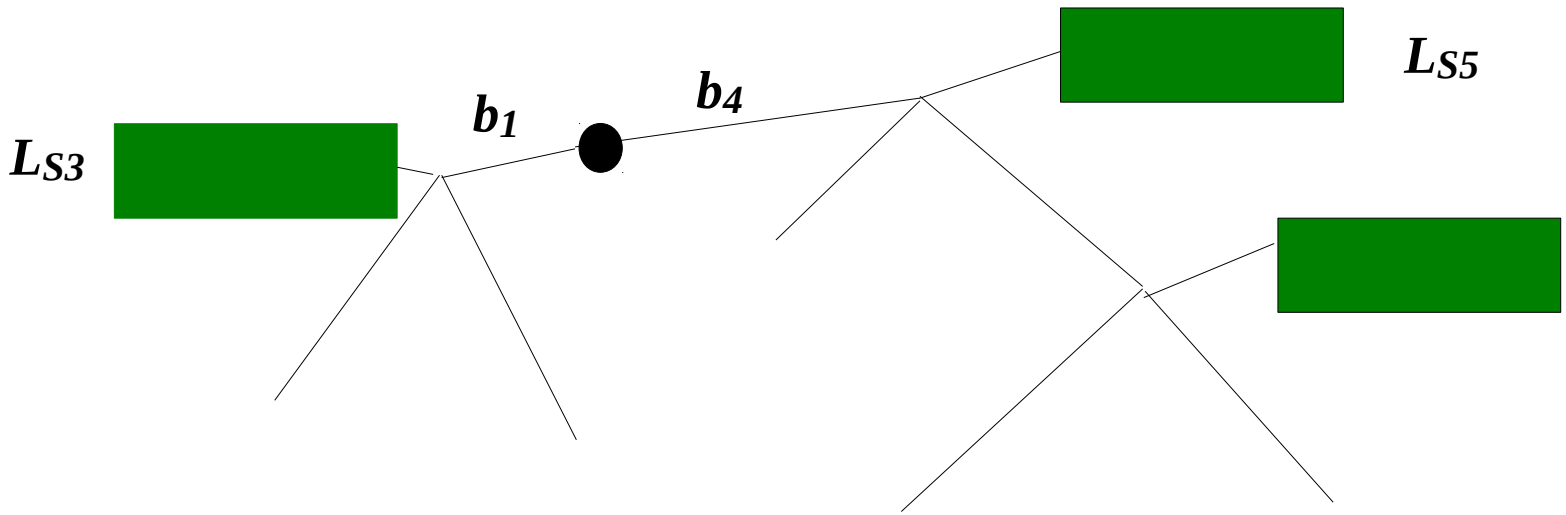
Post-order Traversal

$$L = \sum_{S_4=A}^{T} \pi_{S_4} L_{S_4}$$



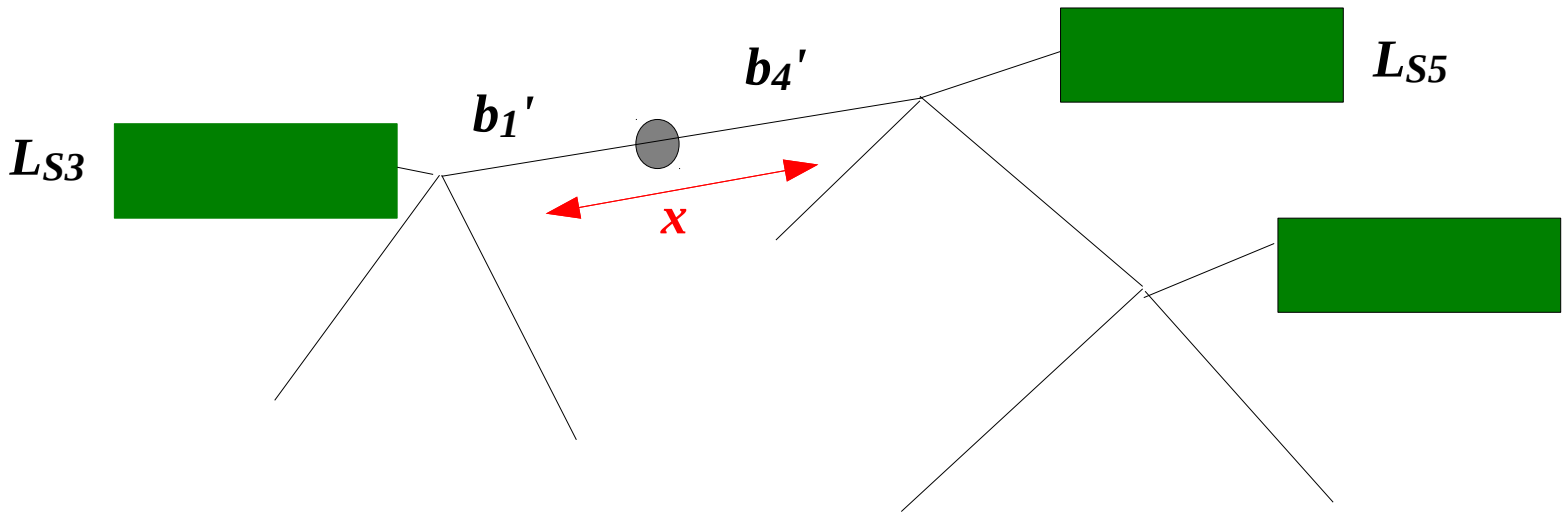
Post-order Traversal

$$L = \sum_{S_4=A}^{T} \pi_{S_4} \left(P_{S_4 S_3}(b_1) L_{S_3}^{(3)} \right) \left(P_{S_4 S_5}(b_4) L_{S_5}^{(5)} \right)$$



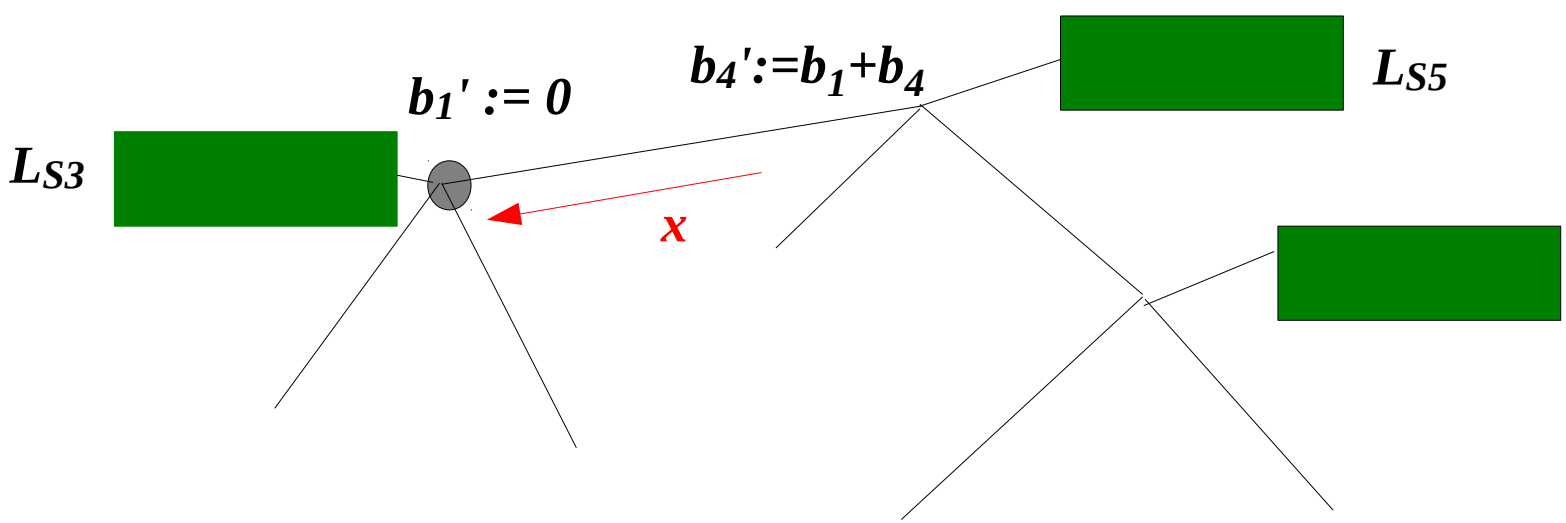
Post-order Traversal

$$L = L' = \sum_{S_4=A}^{T} \pi_{S_4}\left(P_{S_4S_3}(b_1+x)L_{S_3}^{(3)}\right)\left(P_{S_4S_5}(b_4-x)L_{S_5}^{(5)}\right)$$

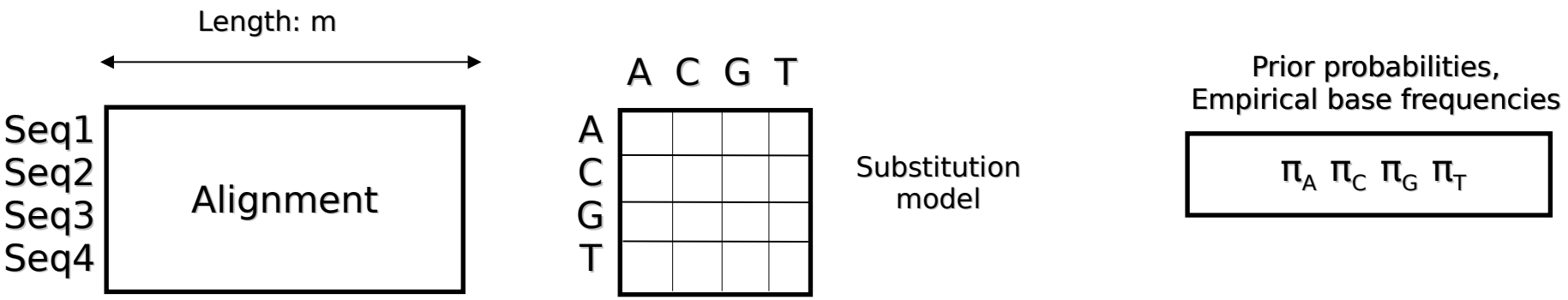


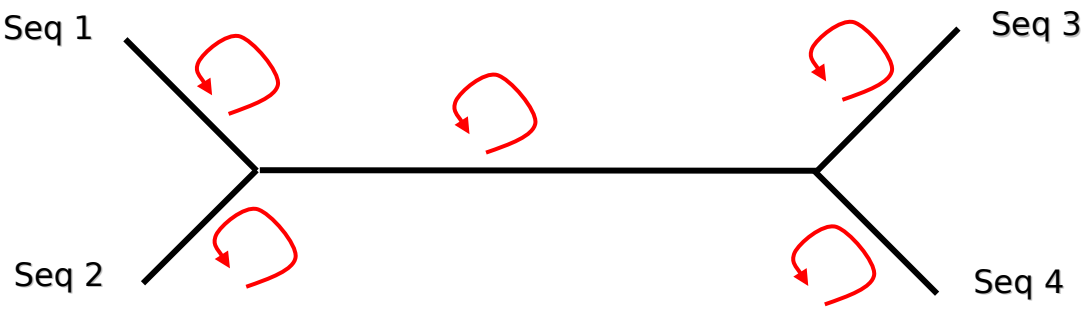
Post-order Traversal

$$L = L' = \sum_{S_4=A}^{T} \pi_{S_4}\left(P_{S_4S_3}(b_1+x)L_{S_3}^{(3)}\right)\left(P_{S_4S_5}(b_4-x)L_{S_5}^{(5)}\right)$$



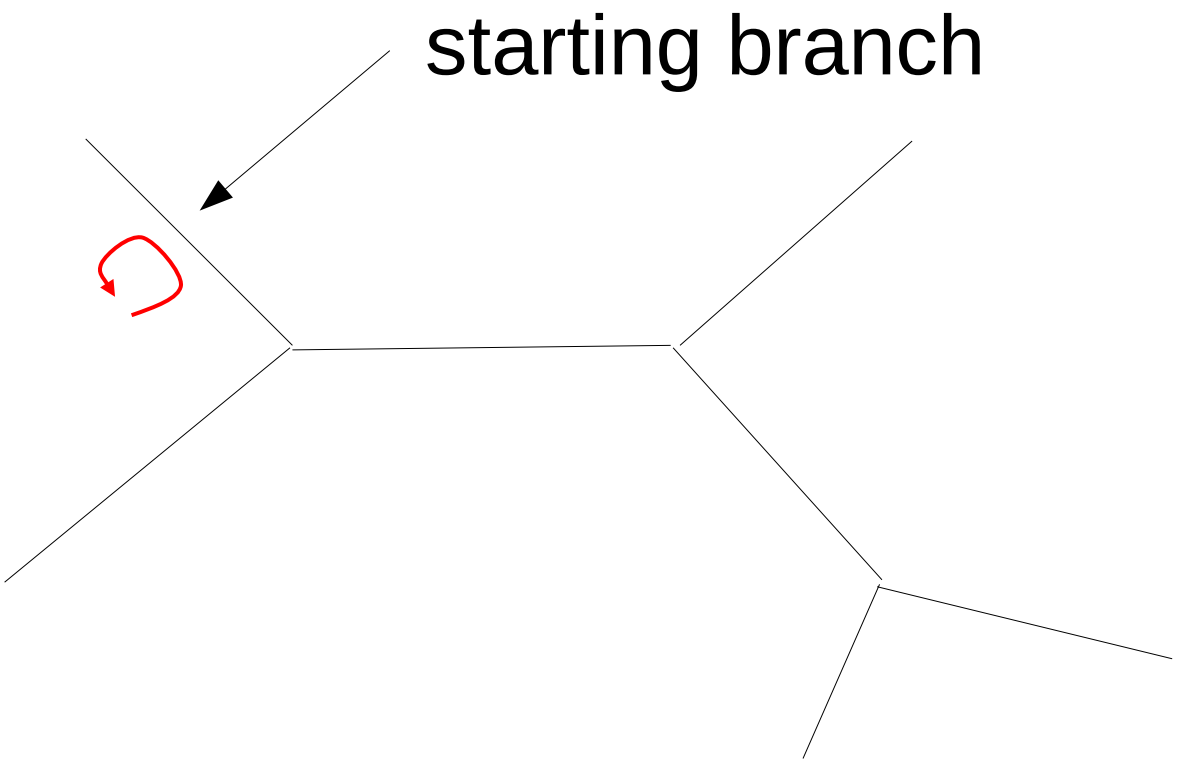
Maximum Likelihood



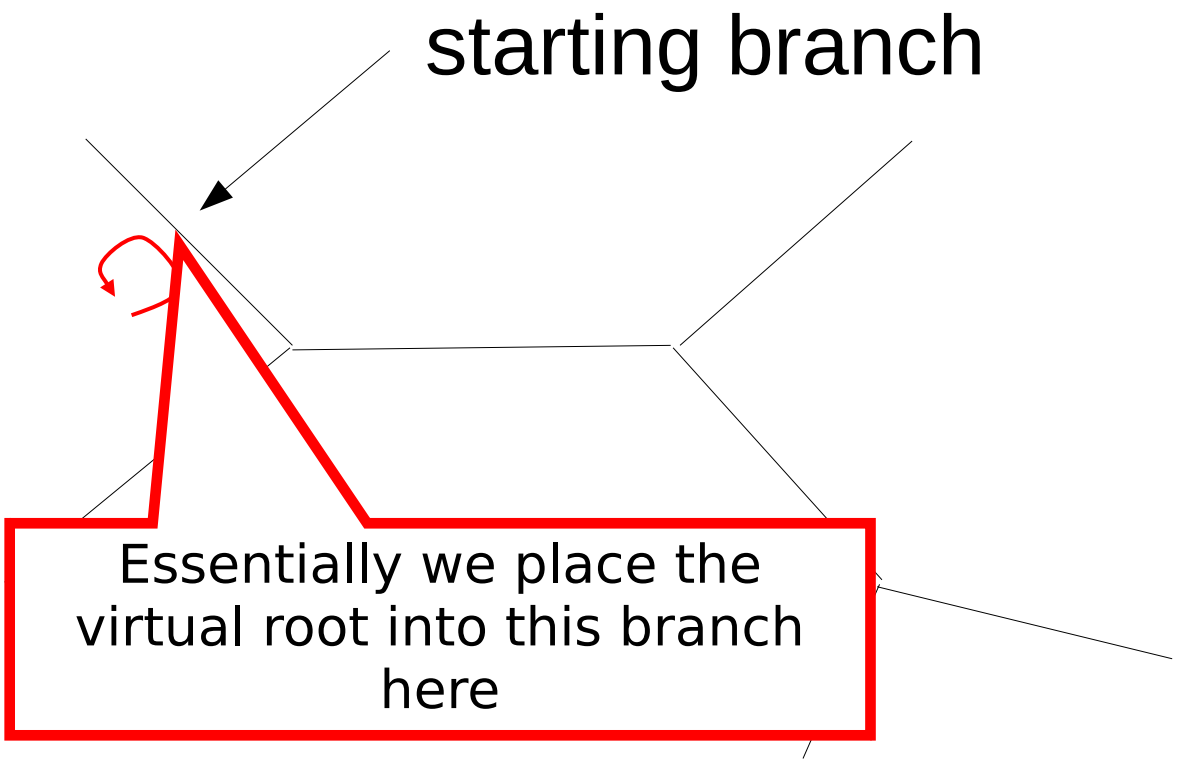


optimize branch lengths

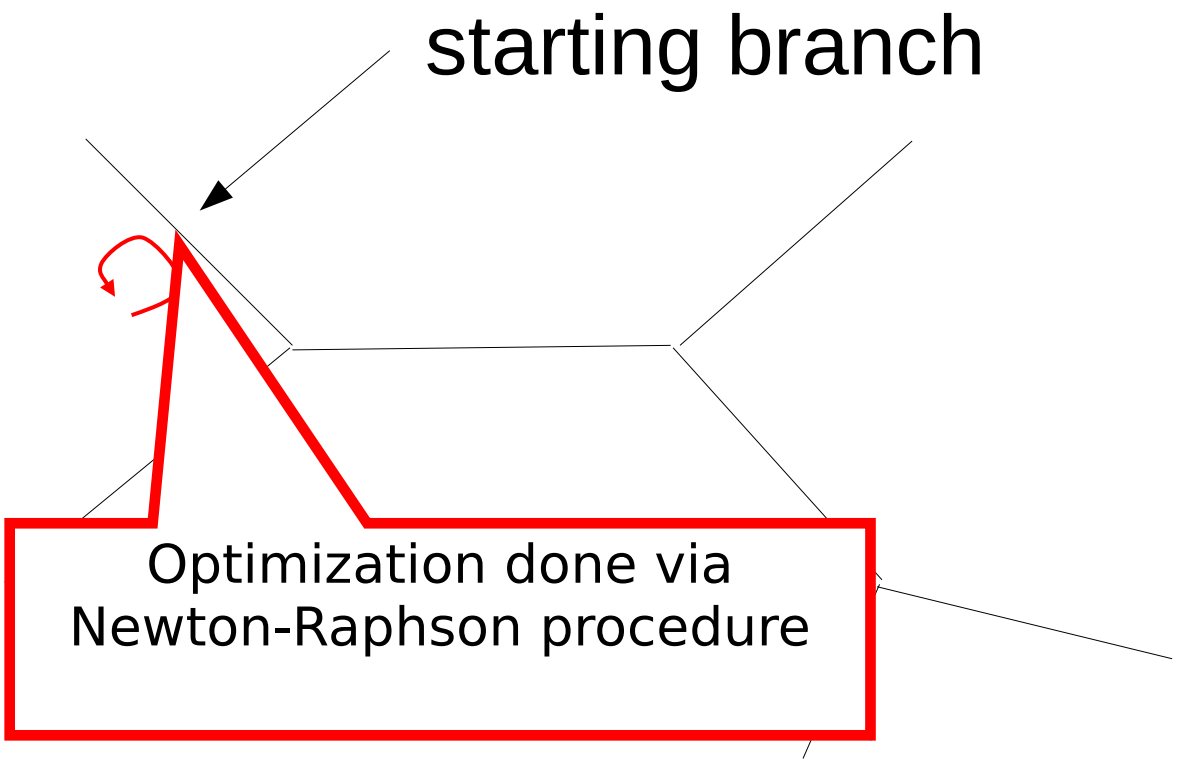
Branch Length Optimization



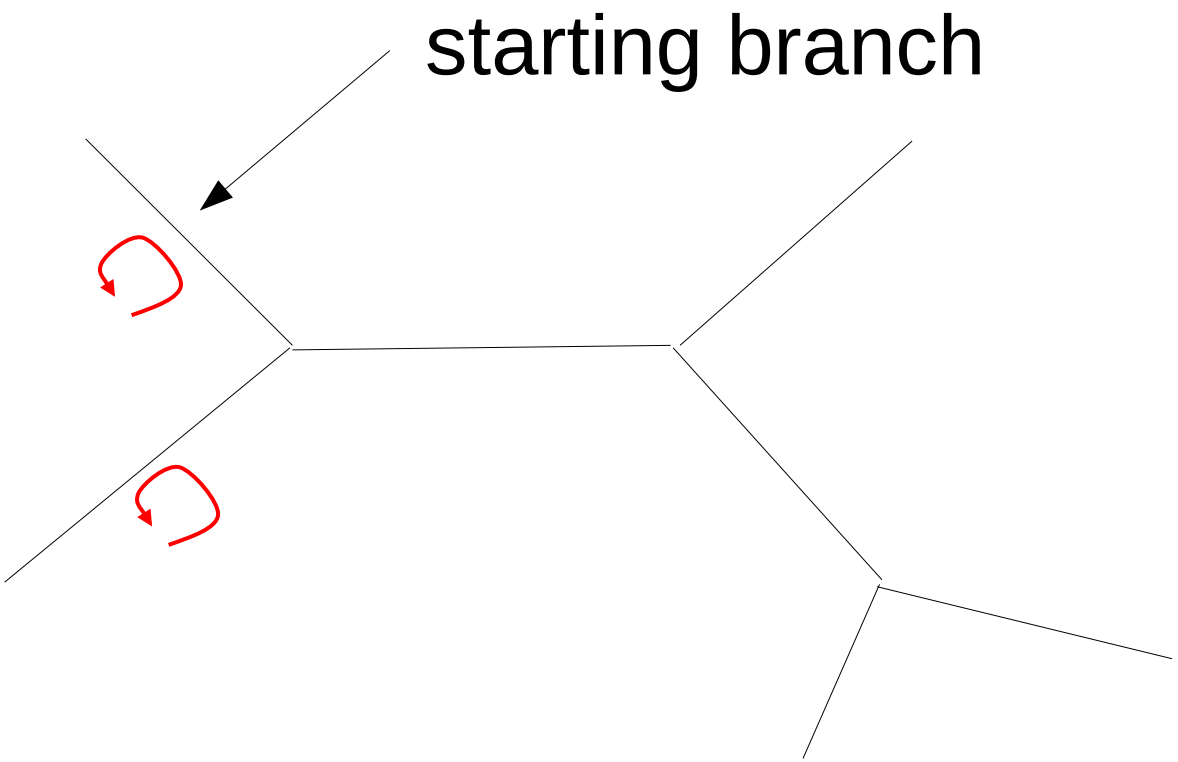
Branch Length Optimization



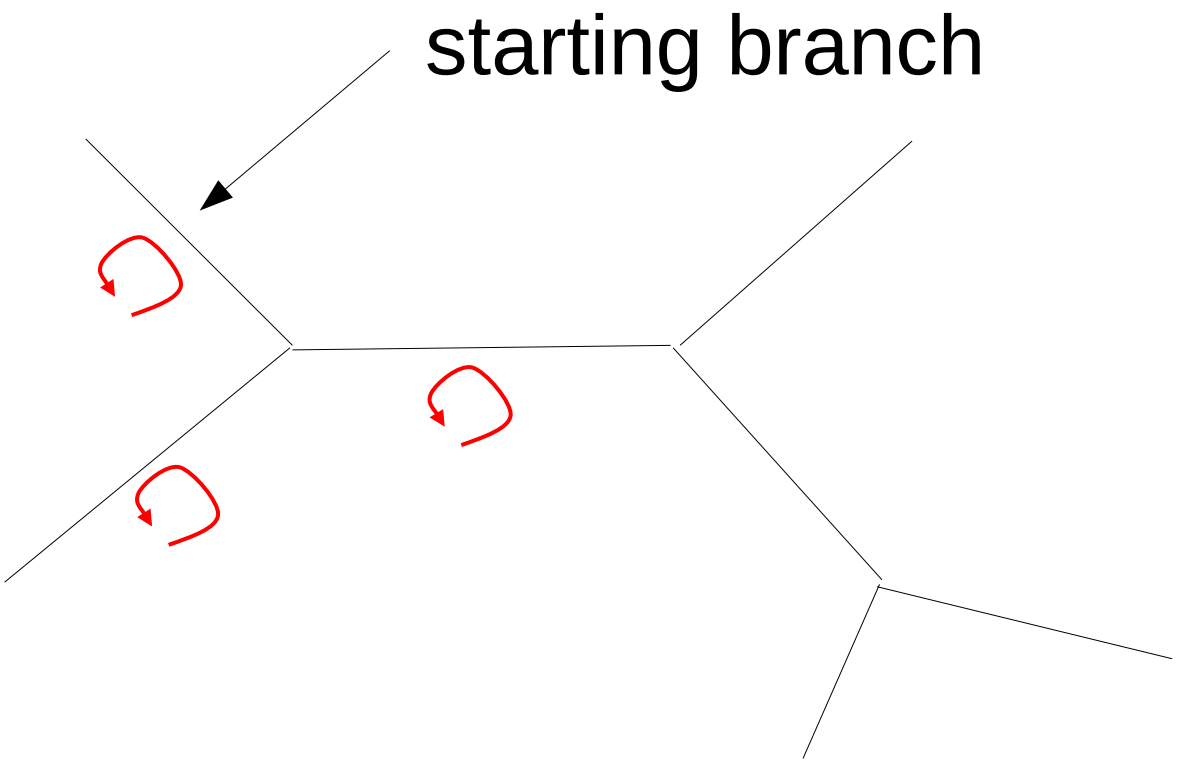
Branch Length Optimization



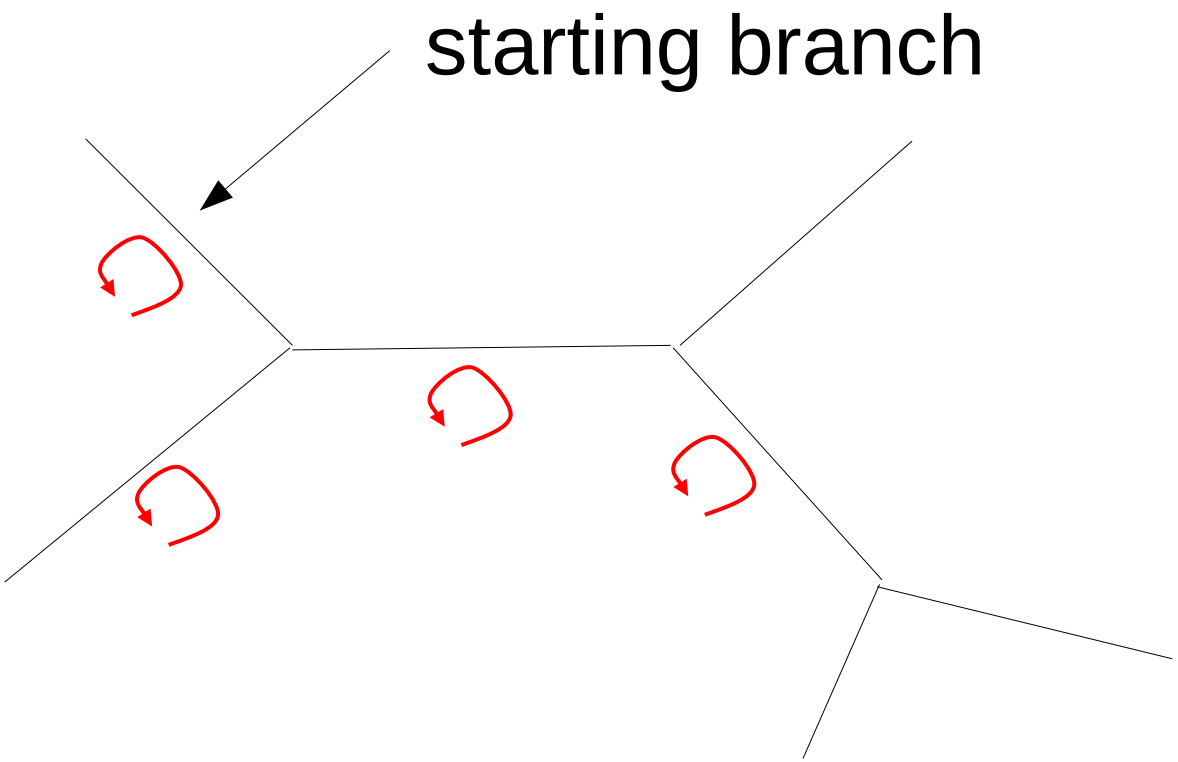
Branch Length Optimization



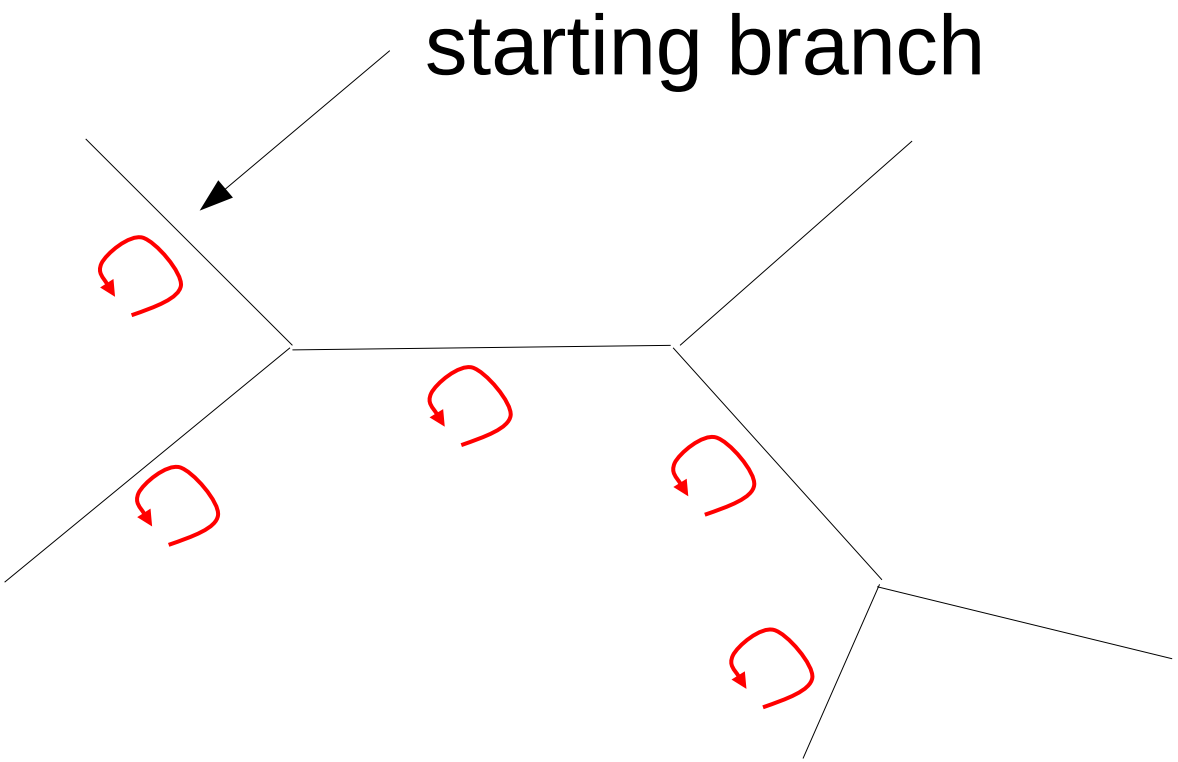
Branch Length Optimization



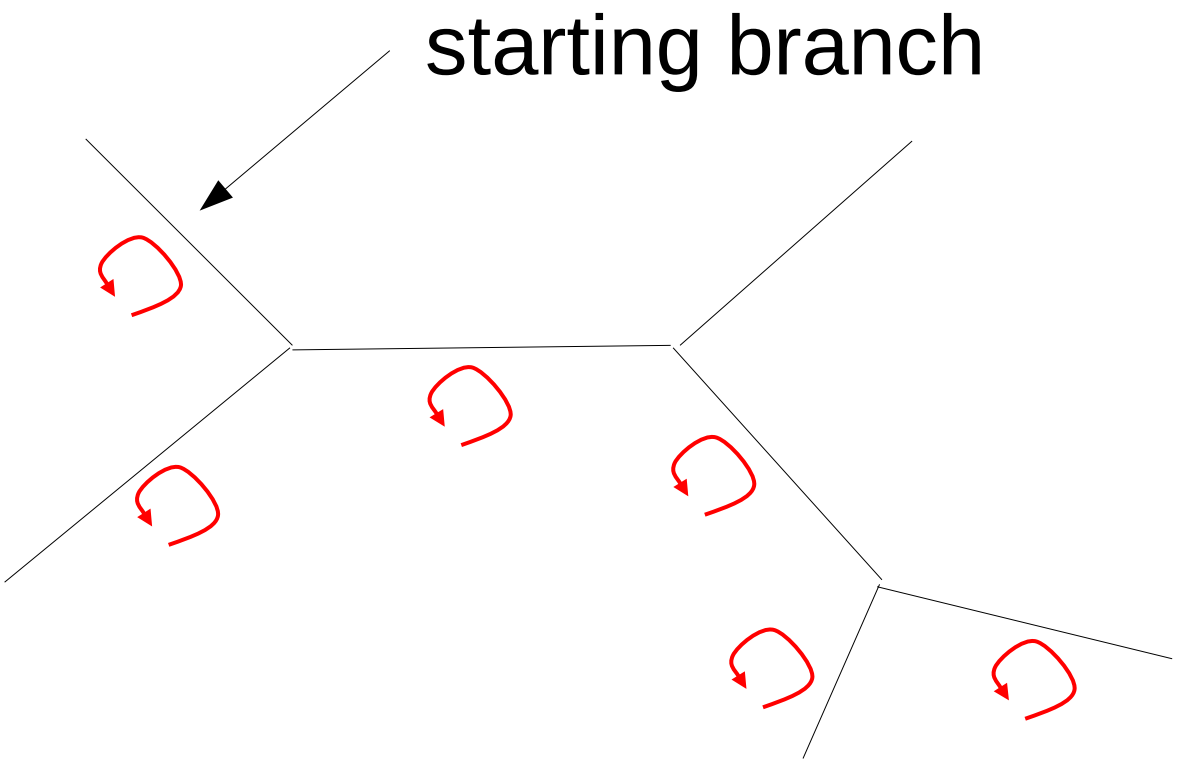
Branch Length Optimization



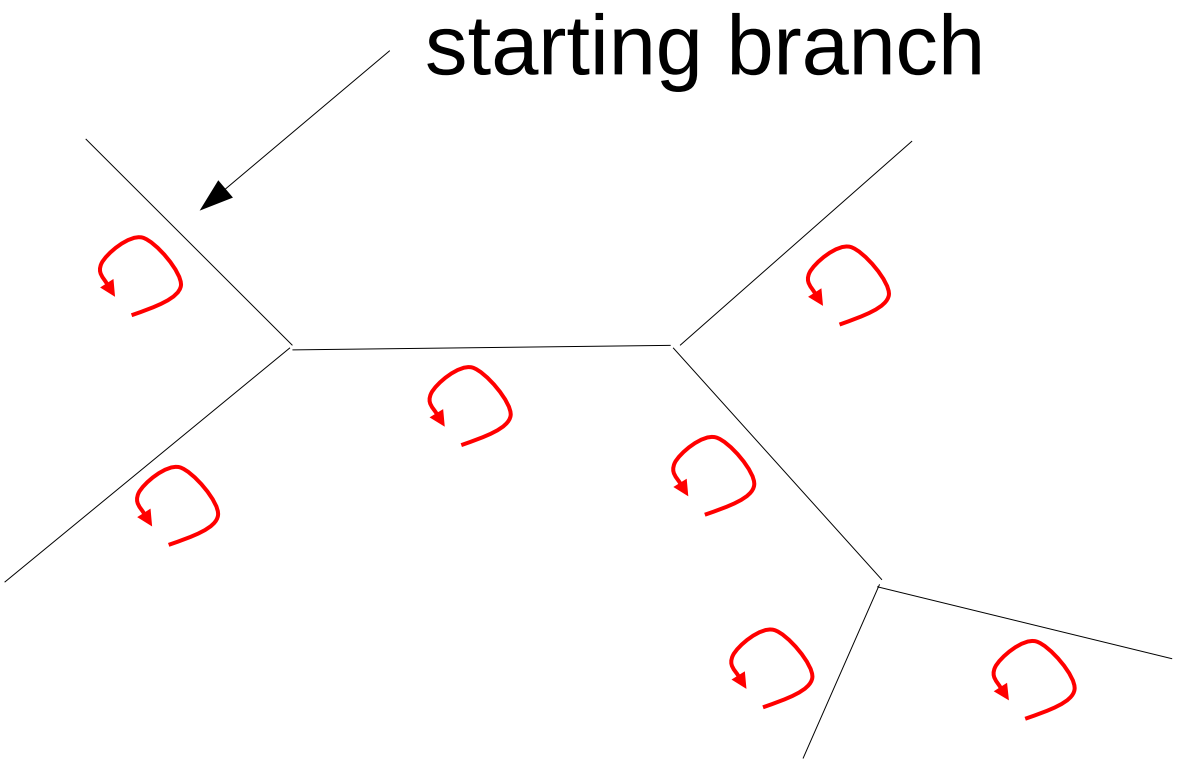
Branch Length Optimization



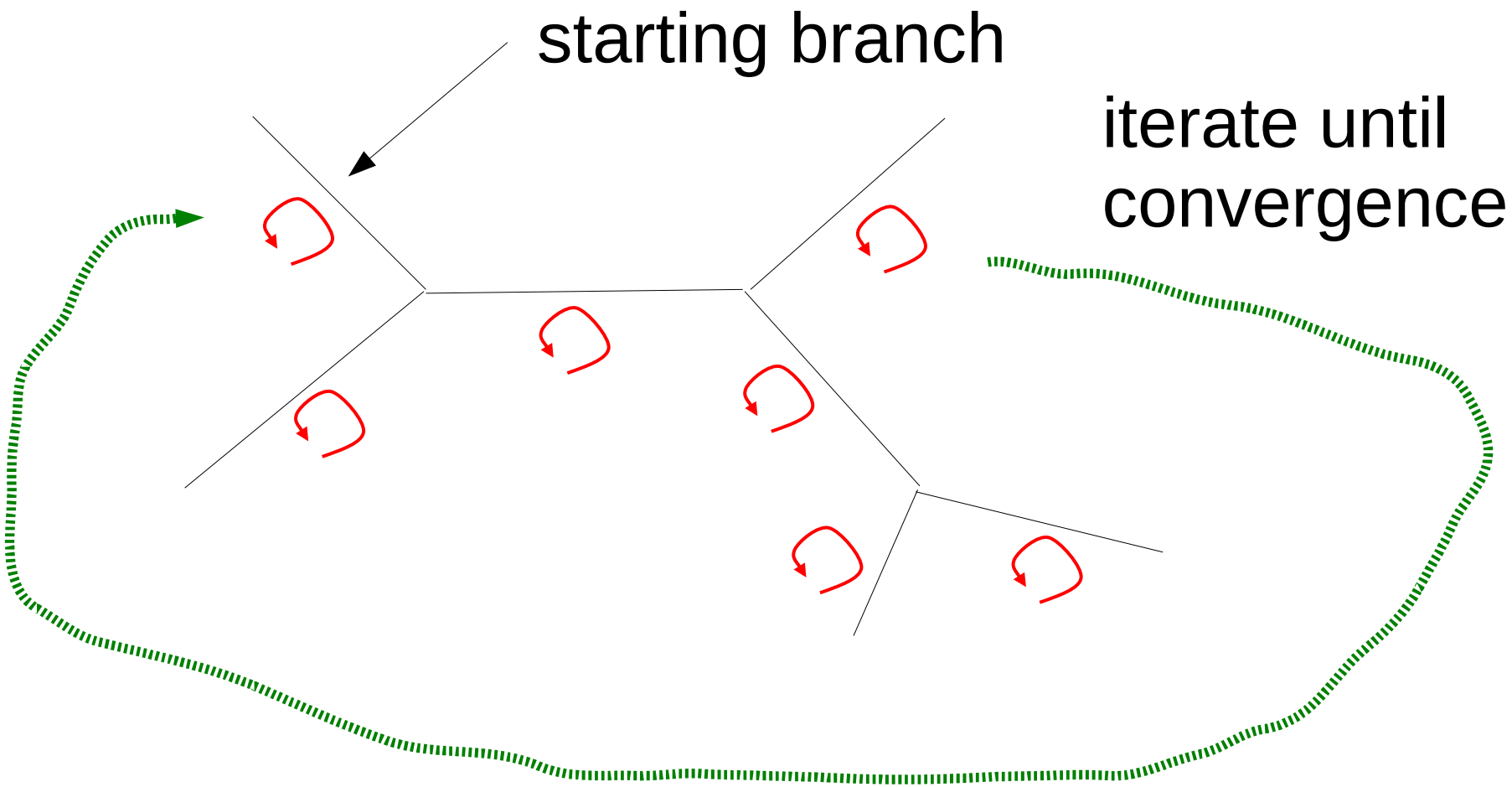
Branch Length Optimization



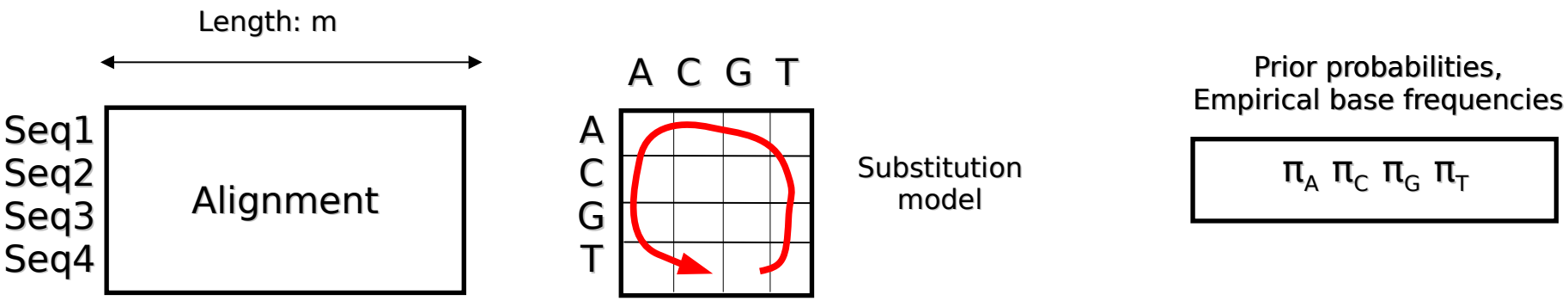
Branch Length Optimization



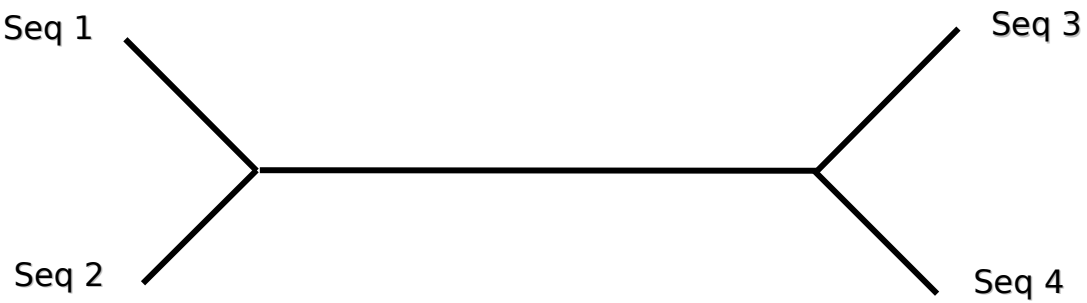
Branch Length Optimization



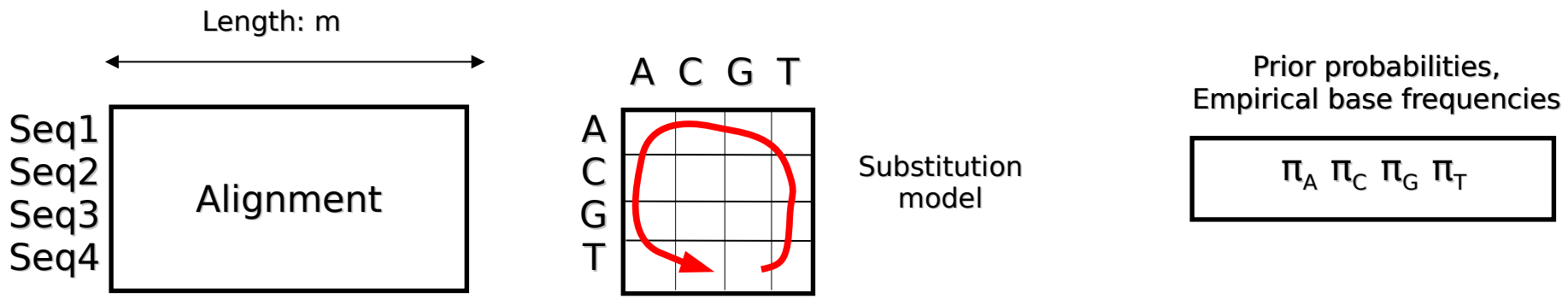
Maximum Likelihood



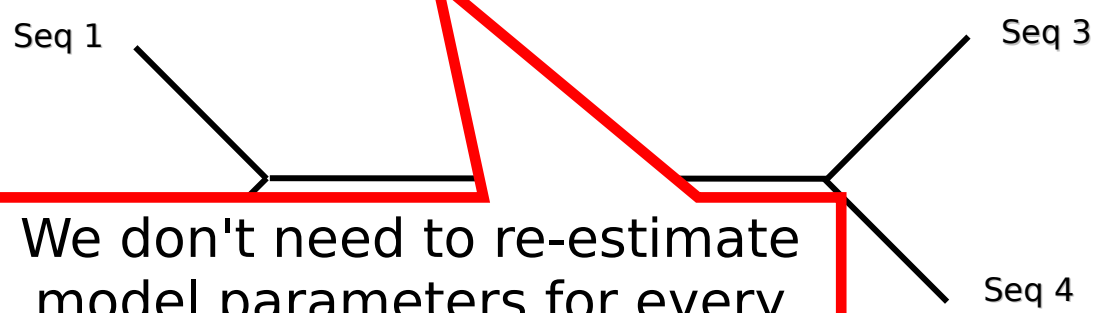
optimize model parameters



Maximum Likelihood

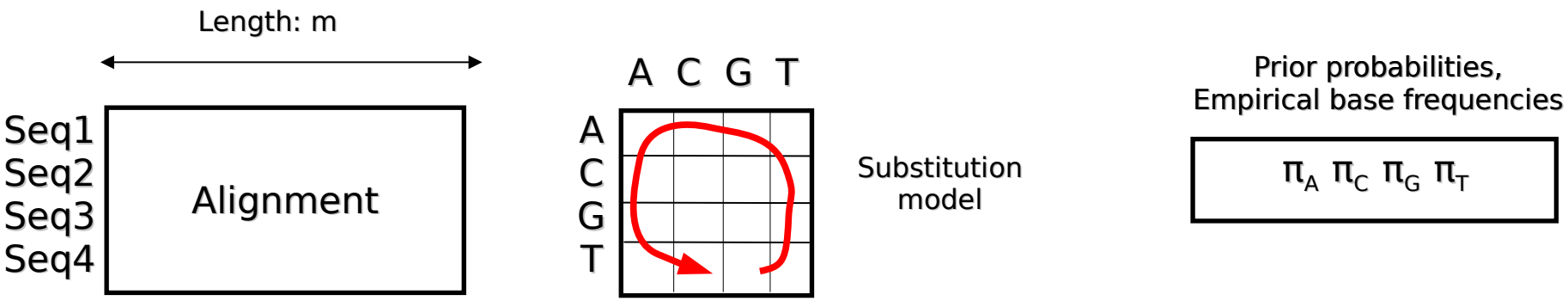


optimize model parameters

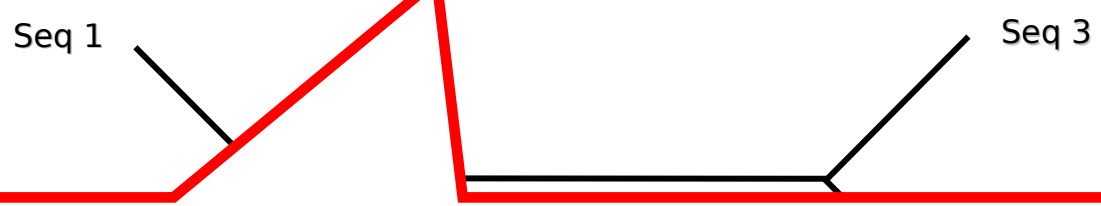


We don't need to re-estimate model parameters for every tree topology as long as the tree is reasonable, i.e., non-random

Maximum Likelihood



optimize model parameters



Methods used for model parameter optimization (other than branch lengths)

1. BFGS
2. Brent's method
3. Expectation maximization approaches

Numerical Optimization Procedures

- See chapters 9 & 10 of: *Numerical Recipes in C – The Art of Scientific Computing*

Basic Operations
Maximum Likelihood

- Compute Conditional Likelihood Vector at an inner node
- Compute Likelihood at Virtual Root
- Optimize a Branch Length for a given Branch
- Optimize all Branch Lengths
- Optimize other Model Parameters

Basic Operations
Maximum Likelihood

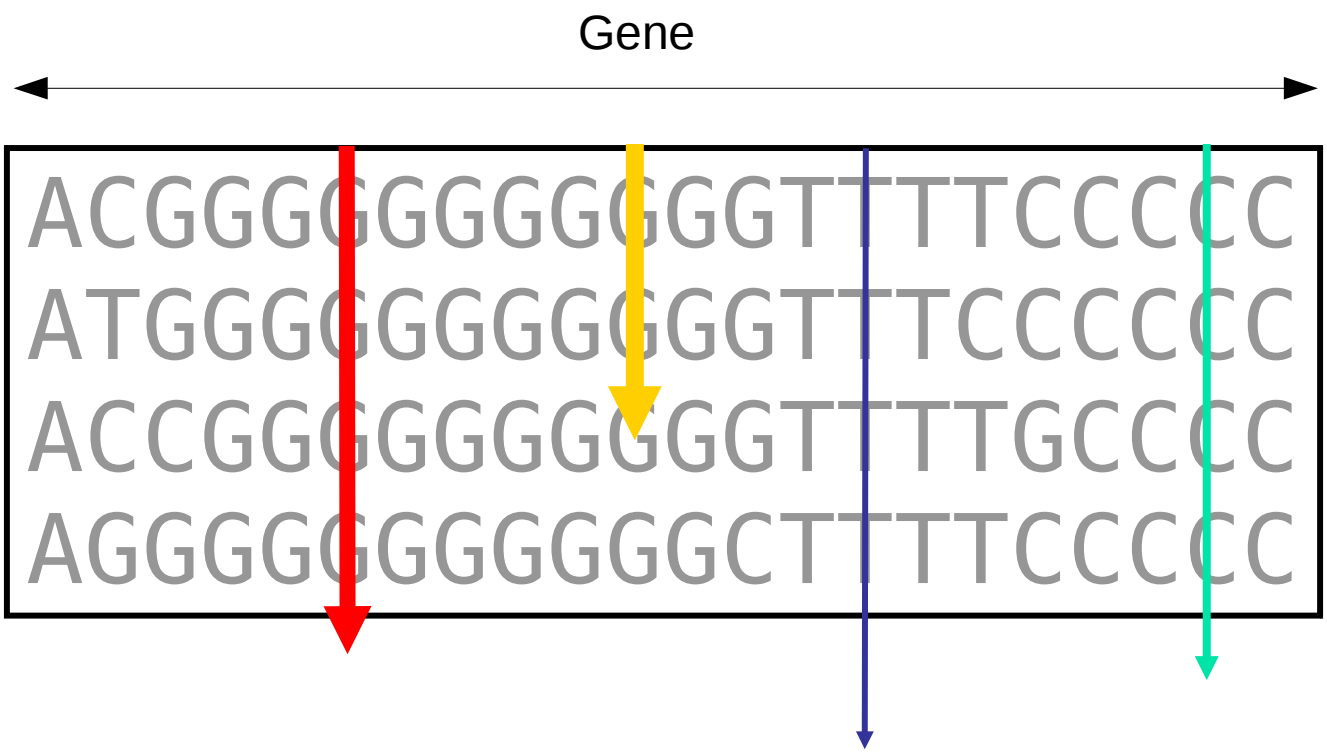
- Compute Conditional Likelihood Vector at an inner node
- Compute Likelihood at Virtual Root
- Optimize a Branch Length for a given Branch
- Optimize all Branch Lengths
- Optimize other Model Parameters

The optimizers are the tricky routines!

Protein Substitution Models

- The GTR *Q* matrix for protein data has 189 free parameters instead of just 5 (DNA)
- Estimating 189 rate parameters is difficult, time-consuming, and may lead to over-parameterizing the model
- Instead, empirical models such as `JTT, LG, WAG, MTMAM`, etc. are used
- The *Q* matrices are obtained by jointly optimizing model parameters on a large collection of reference alignments
- The models differ with respect to:
  - the amount of data used to obtain them
  - the type of data on which the models have been optimized
    - e.g., dedicated models for HIV, FLU, Mammals
  - the numerical optimization methods used
- Examples of general models:
  - **`LG`**: Le & Gascuel: “An Improved General Amino Acid Replacement Matrix”
  - **`WAG`**: Whelan & Goldman: “A General Empirical Model of Protein Evolution Derived from Multiple Protein Families Using a Maximum-Likelihood Approach”

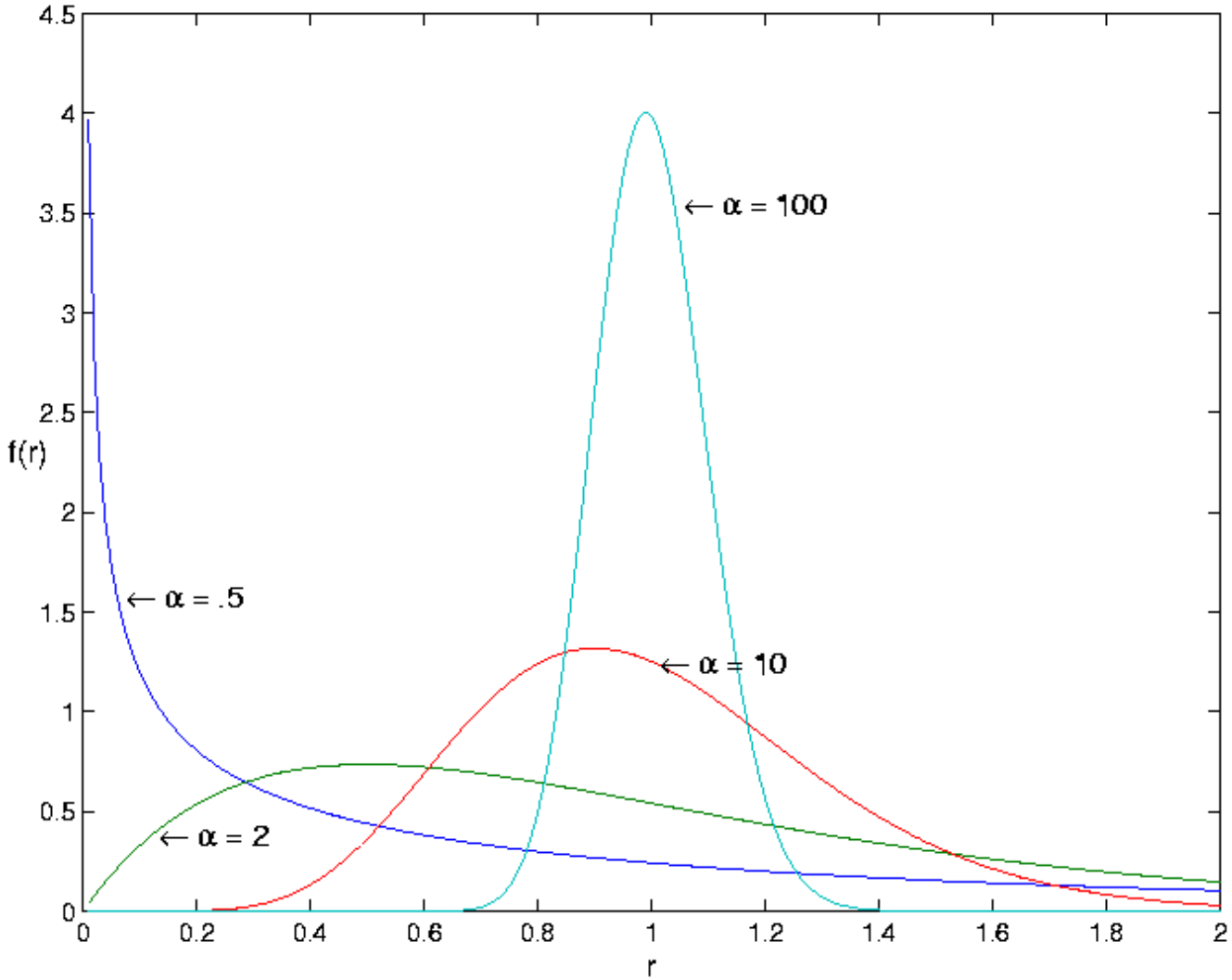
Rate Heterogeneity among Sites



- Among-site rate heterogeneity
  - Biological phenomenon

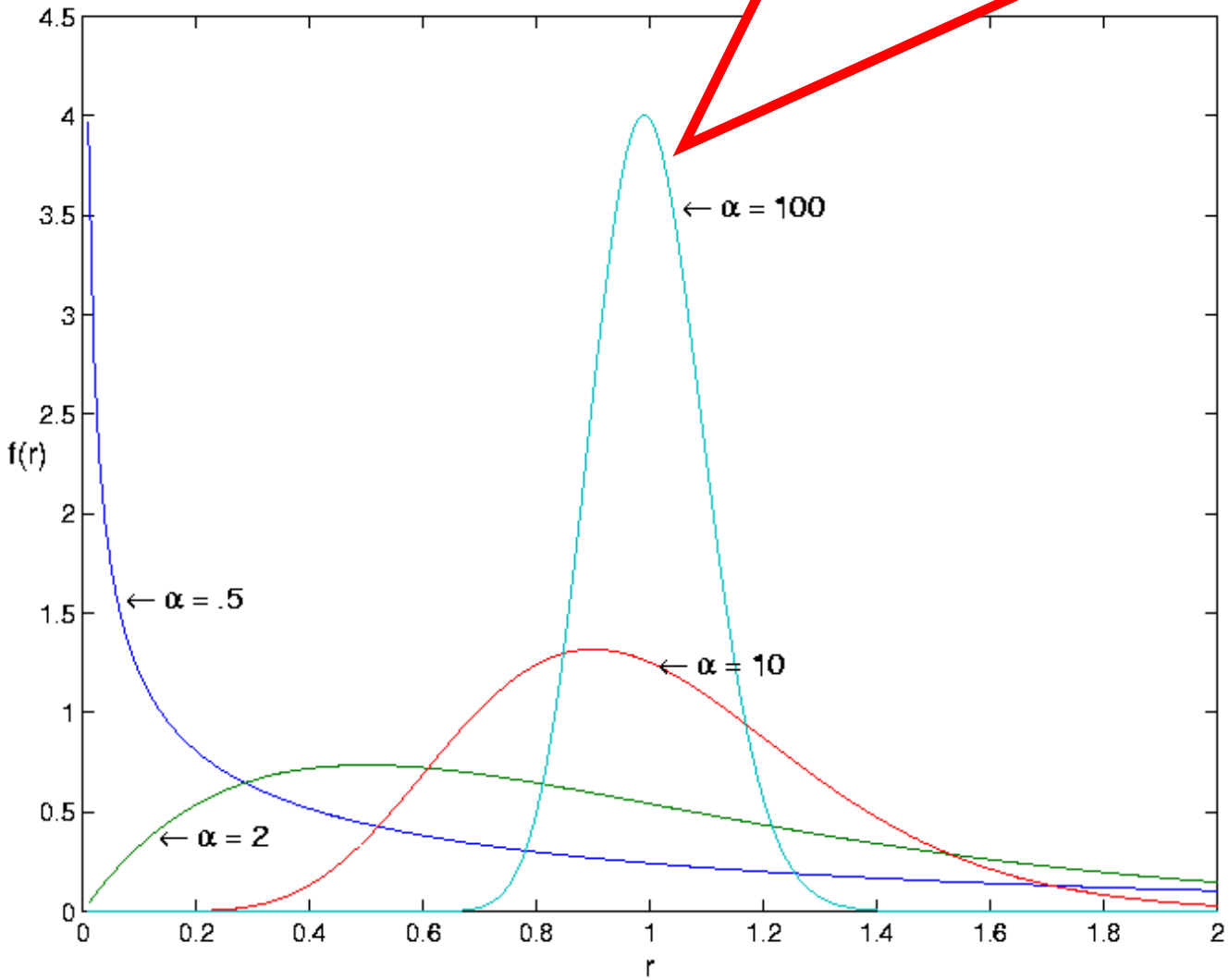
    → different sites/columns evolve at different speeds
  - Need to accommodate this in our models

Γ-Distribution

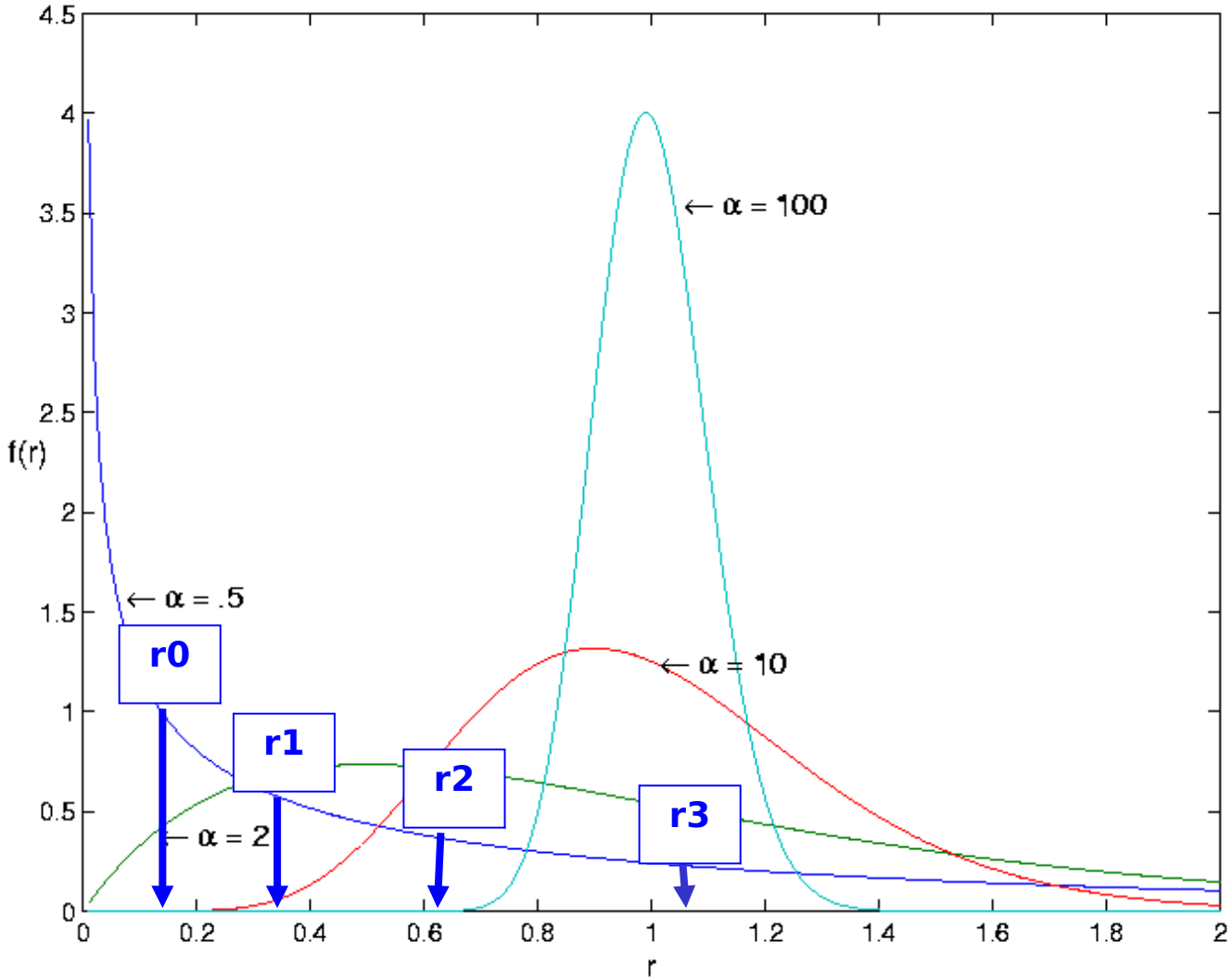


Γ-Distribution

Small α ➔high rate heterogeneity
Large α ➔low rate heterogeneity



Discrete Γ-Distribution



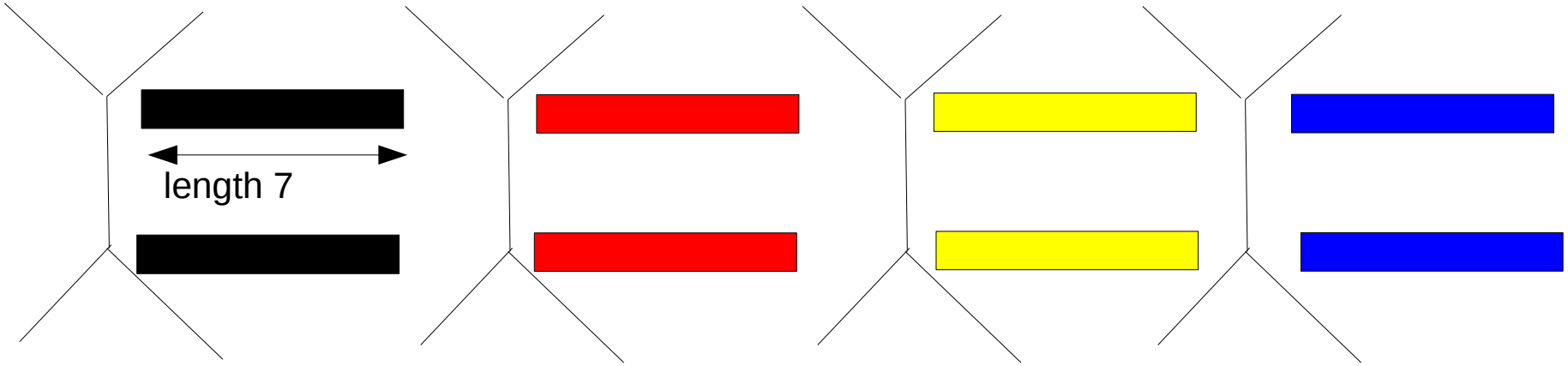
An Abstract View of Γ

rate 0
$P(t) = e^{Qr_0 t}$

rate 1
$P(t) = e^{Qr_1 t}$

rate 2
$P(t) = e^{Qr_2 t}$

rate 3
$P(t) = e^{Qr_3 t}$



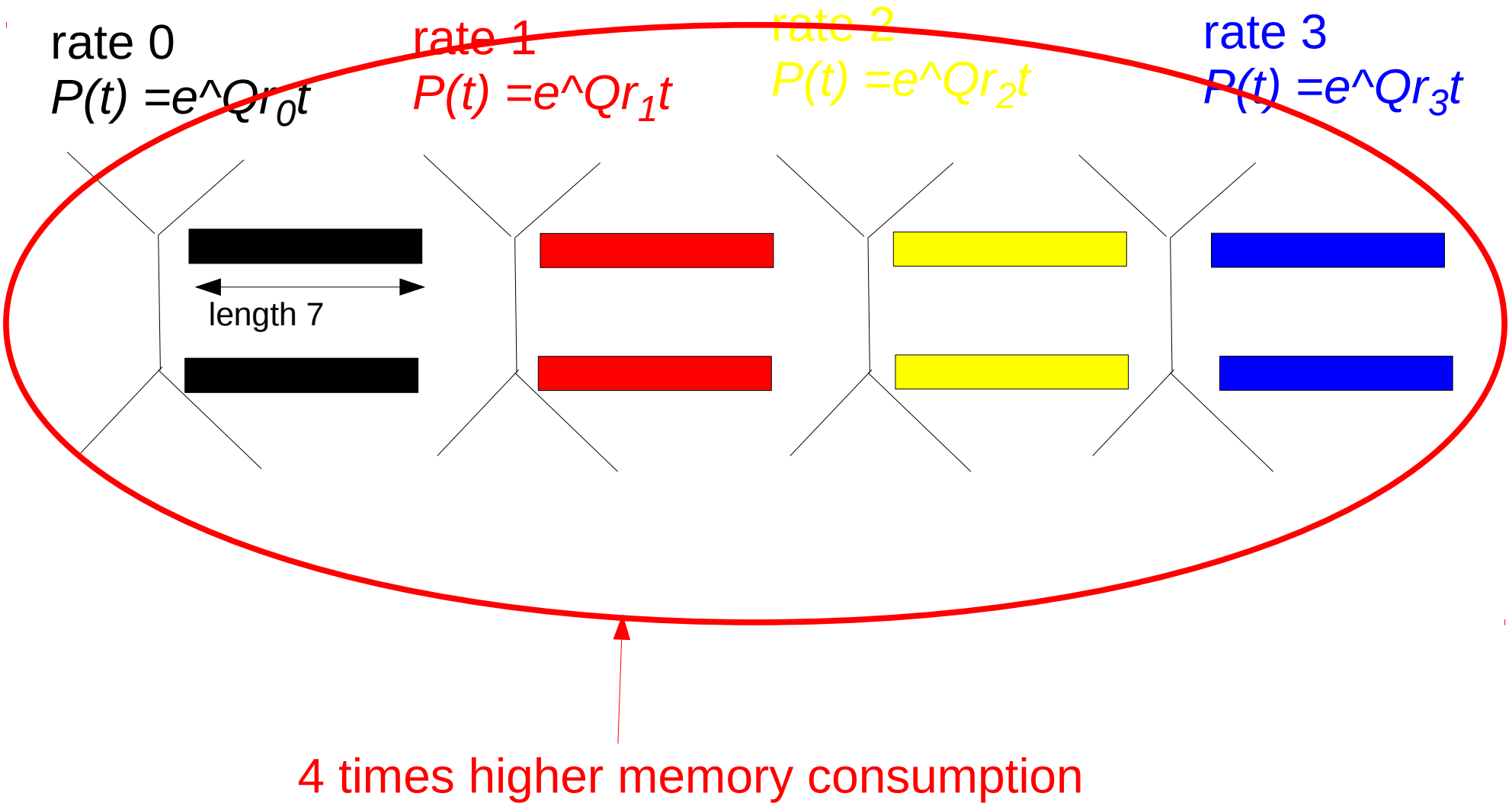
This is the integral of the likelihood we approximate via discretization

$$LnL(i) = \log(\tfrac{1}{4} * (L_0 + L_1 + L_2 + L_3))$$

Log likelihood at site i

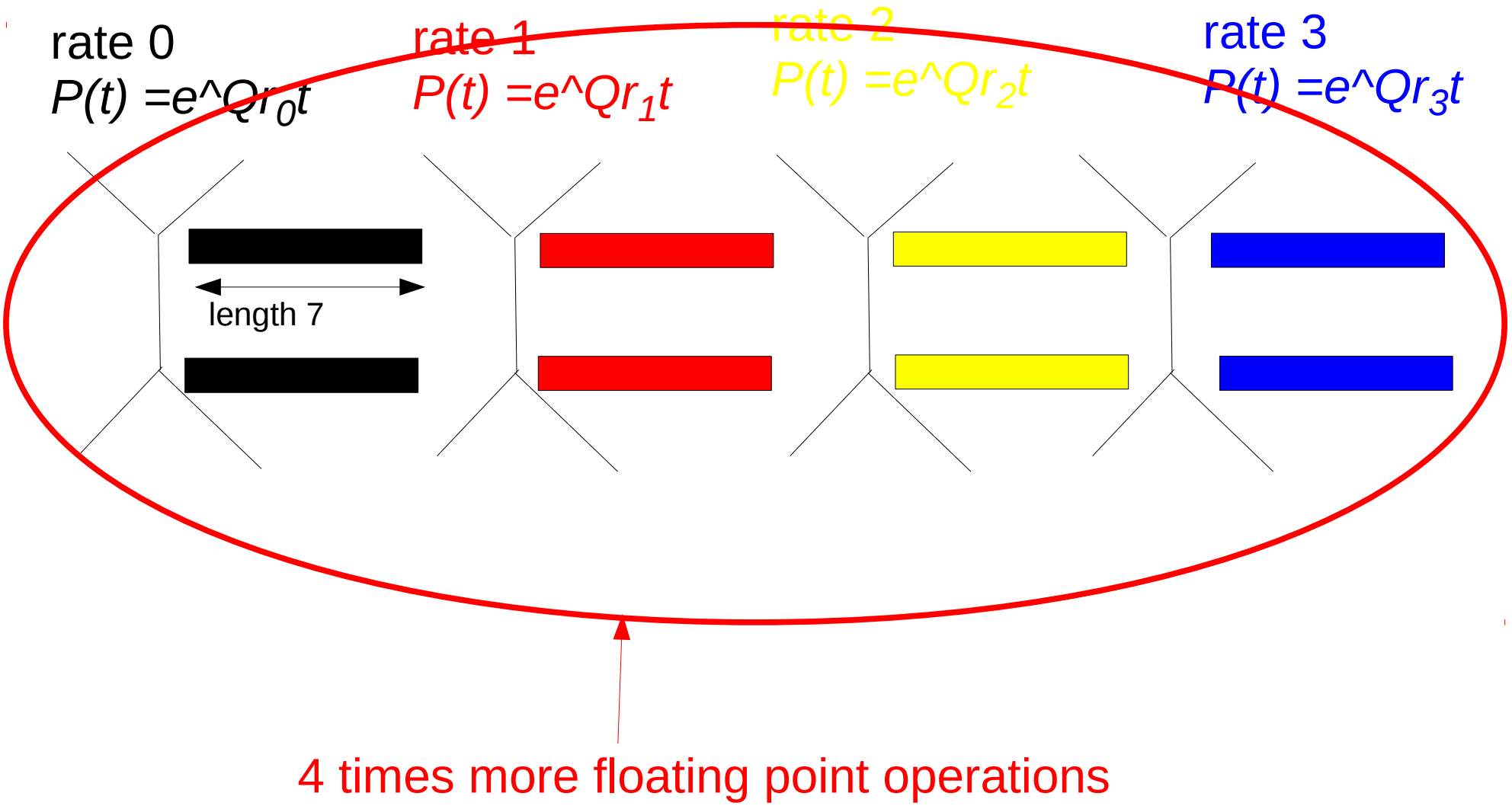
All Γ rates have equal probability

An Abstract View of Γ



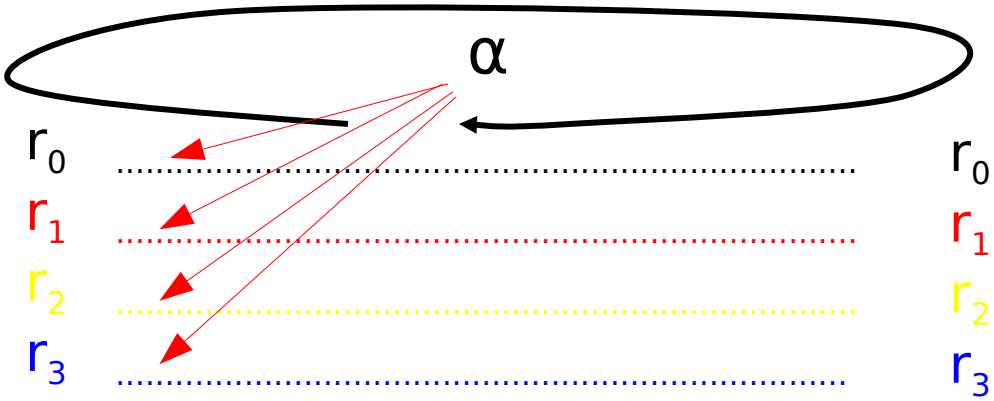
4 times higher memory consumption

An Abstract View of Γ



4 times more floating point operations

Γ Model of Rate Heterogeneity with 4 discrete rates

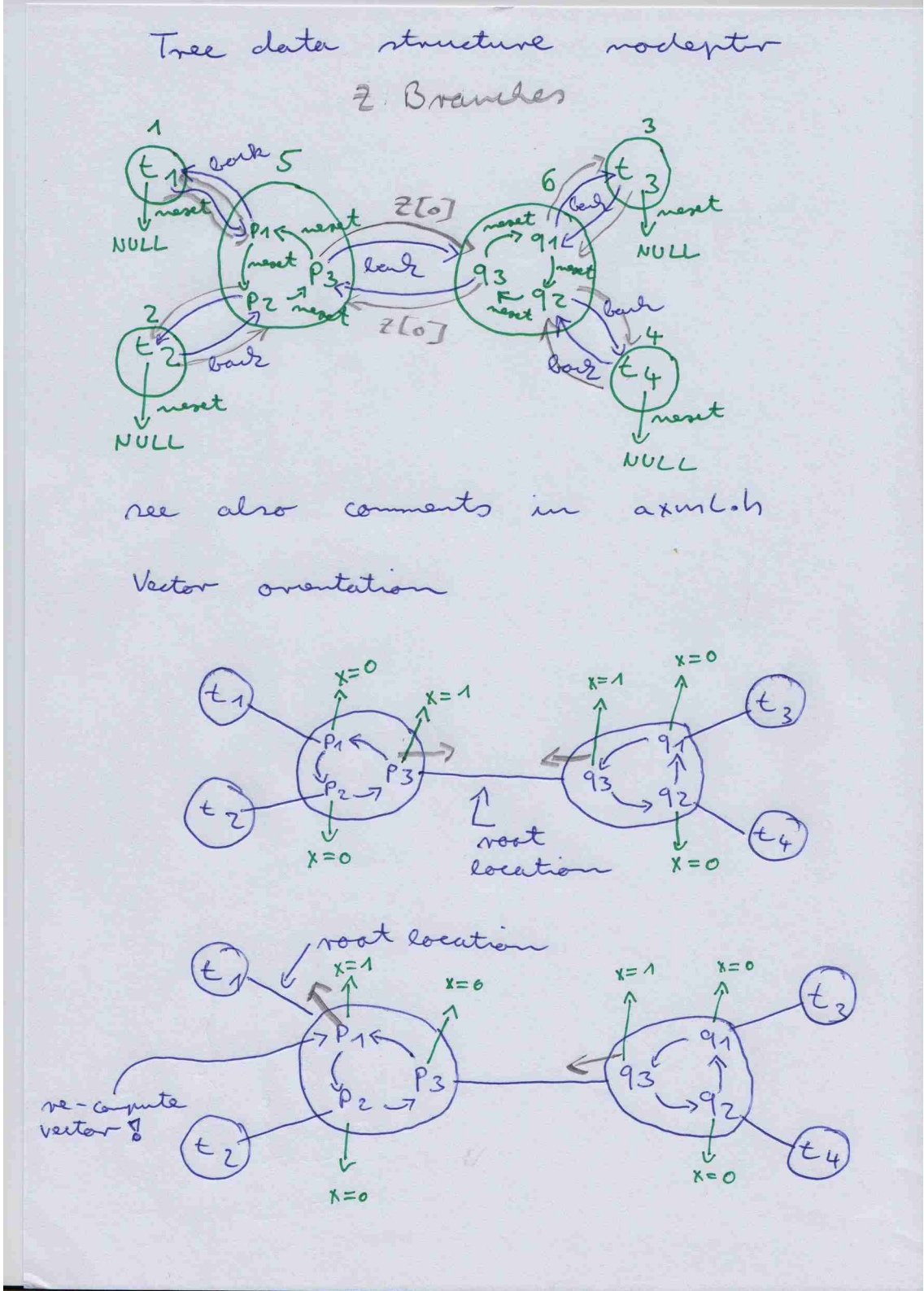



```
Species209   NNNNNNNNNNN NNNNNNNNNNN NNNNNNNNNNN NNNNNNNNNNN
Species159   UAUCUGGUUG AUCCUGCCAG UAGUAUNUGC UUGUCUCAAA
Species109   NNNNNNNNNNN NNNNNNNNNNN NNNNAUAUGC UUGUCUC--A
Species025   NNNNNNNNNNN NNNNNNNNNNN NNNNNNNNNNN NNNNNNNAAA
Species004   NNNNNNNNNNN NNNNNNNNNNN NNNNNNNNNNN NNNNNNNNNN
Species186   UACCUGGUUG AUCCUGCCAG UAGUAUAUGC UUGUCUCAAA
Species034   NNNNNNNNNNN NNNNNNNNNNN NNNNNNNNNNN NNNNNNNAAA
Species003   NNNNNNNNNNN NNNNNNNNNNN NNNNNNNNNNN NNNNNNNAAA
Species192   NNNNNNNNNNN NNNNNNNNNNN NNNNNNNNNNN NNNNNNNCAAA
Species132   NNNNNNNNNNN NNNNNNNNNNN UAGUAUAUGC UUGUCUCAAA
Species172   NNNNNNNNNNN NNNGAAUU-G UAUUANANGC -UGUUUCAAA
```

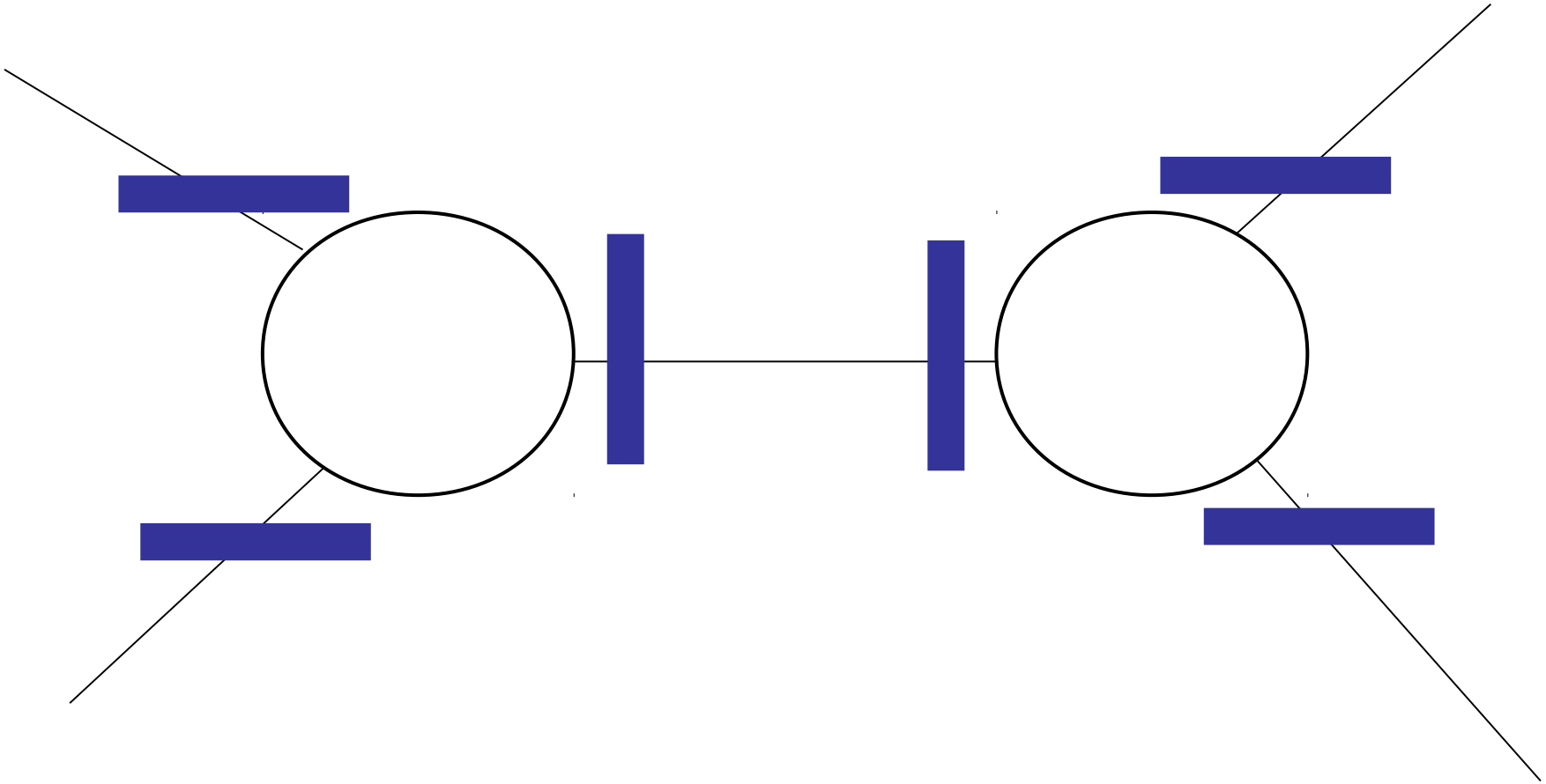
Data Structures for unrooted Trees

- Unrooted trees with dynamically changing virtual roots need a dedicated tree data structure
- Why are the virtual root positions changing dynamically?
- I will outline this on the board

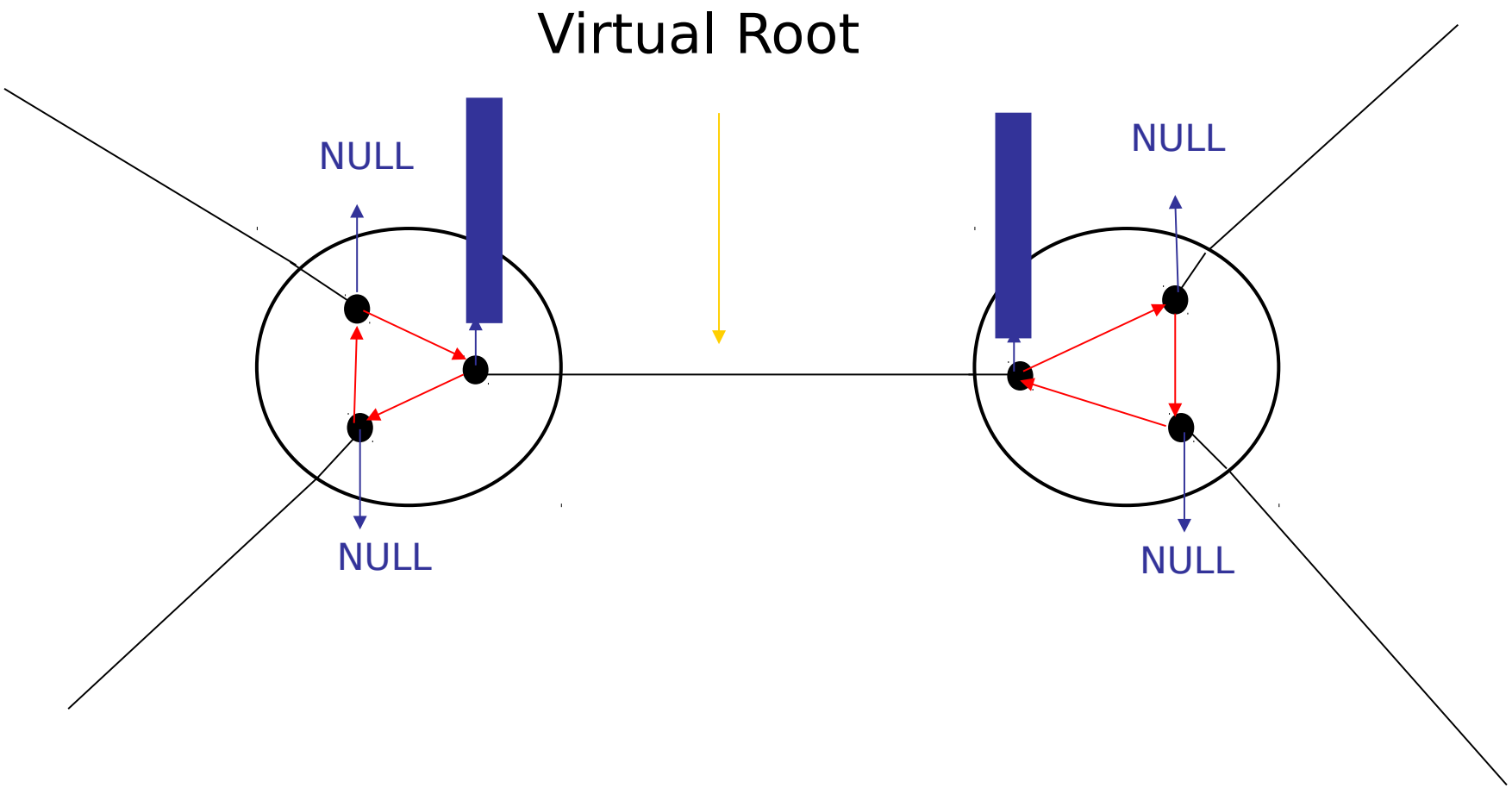
Data Structures for unrooted Trees



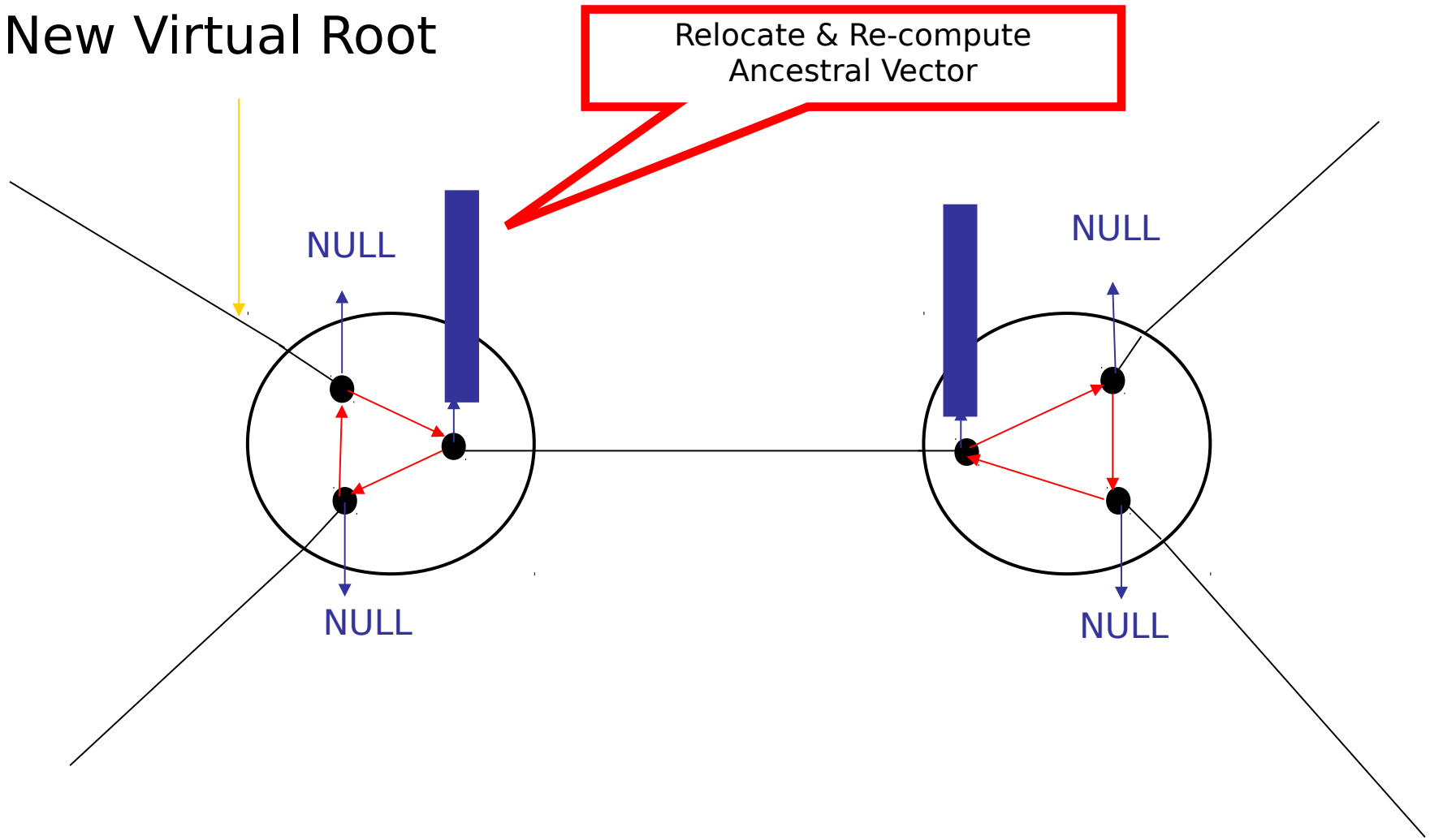
Memory Organization: Conditional Likelihood Vectors with an Unrooted View



Memory Organization: Conditional Likelihood Vectors with a Rooted View

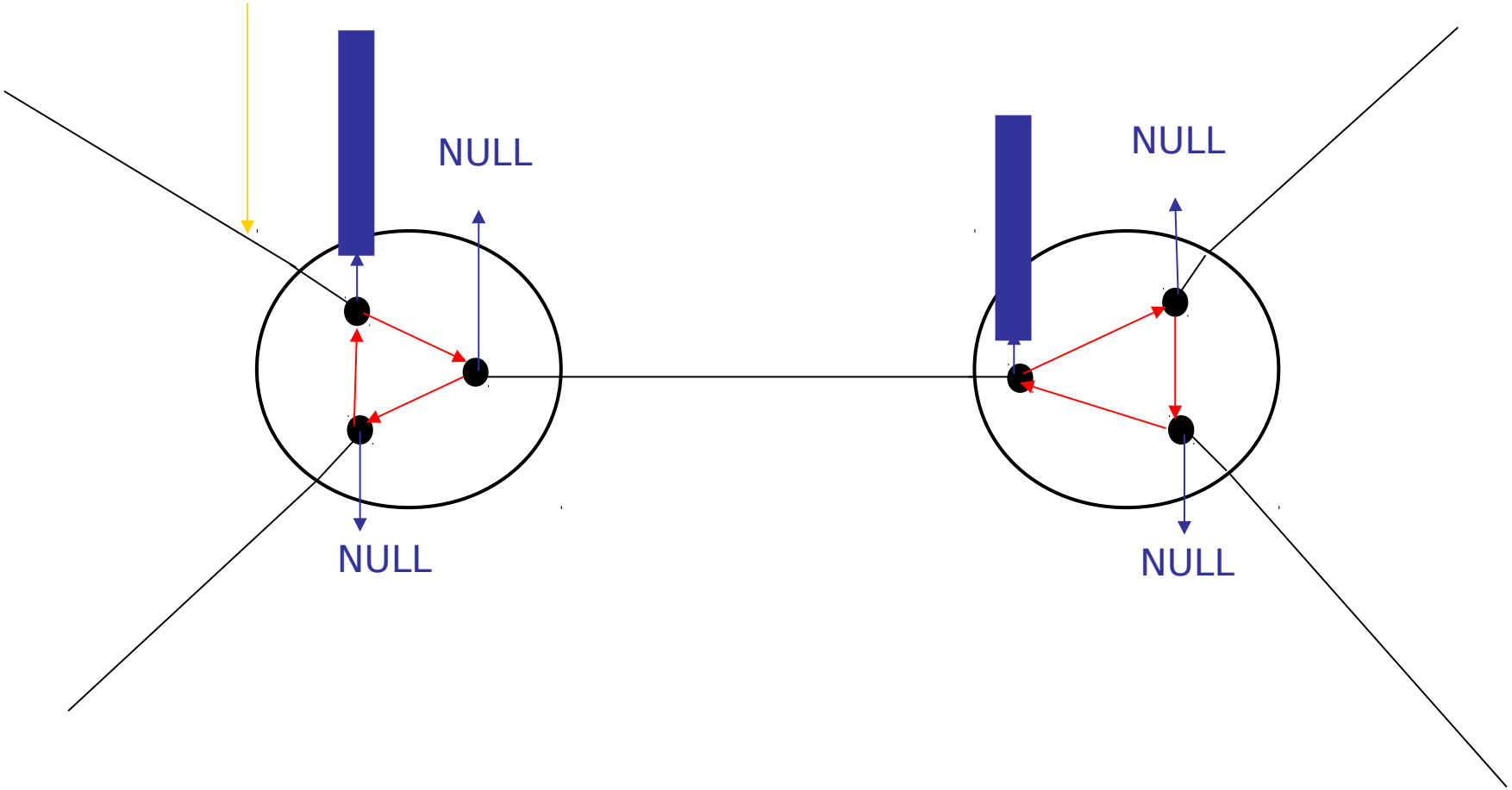


Memory Organization: CLVs with a Rooted View

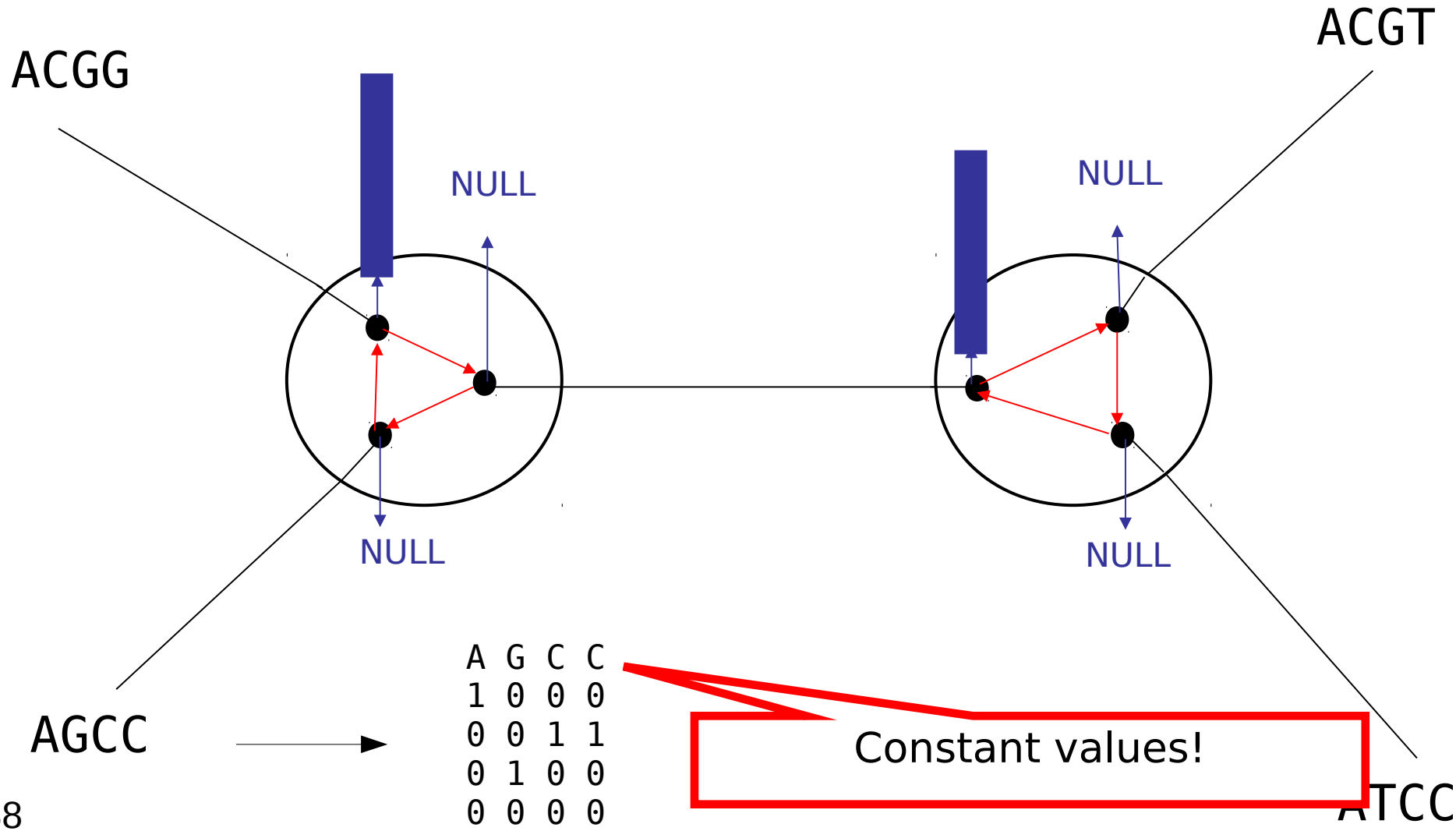


Memory Organization: Ancestral Vectors with a Rooted View

New Virtual Root

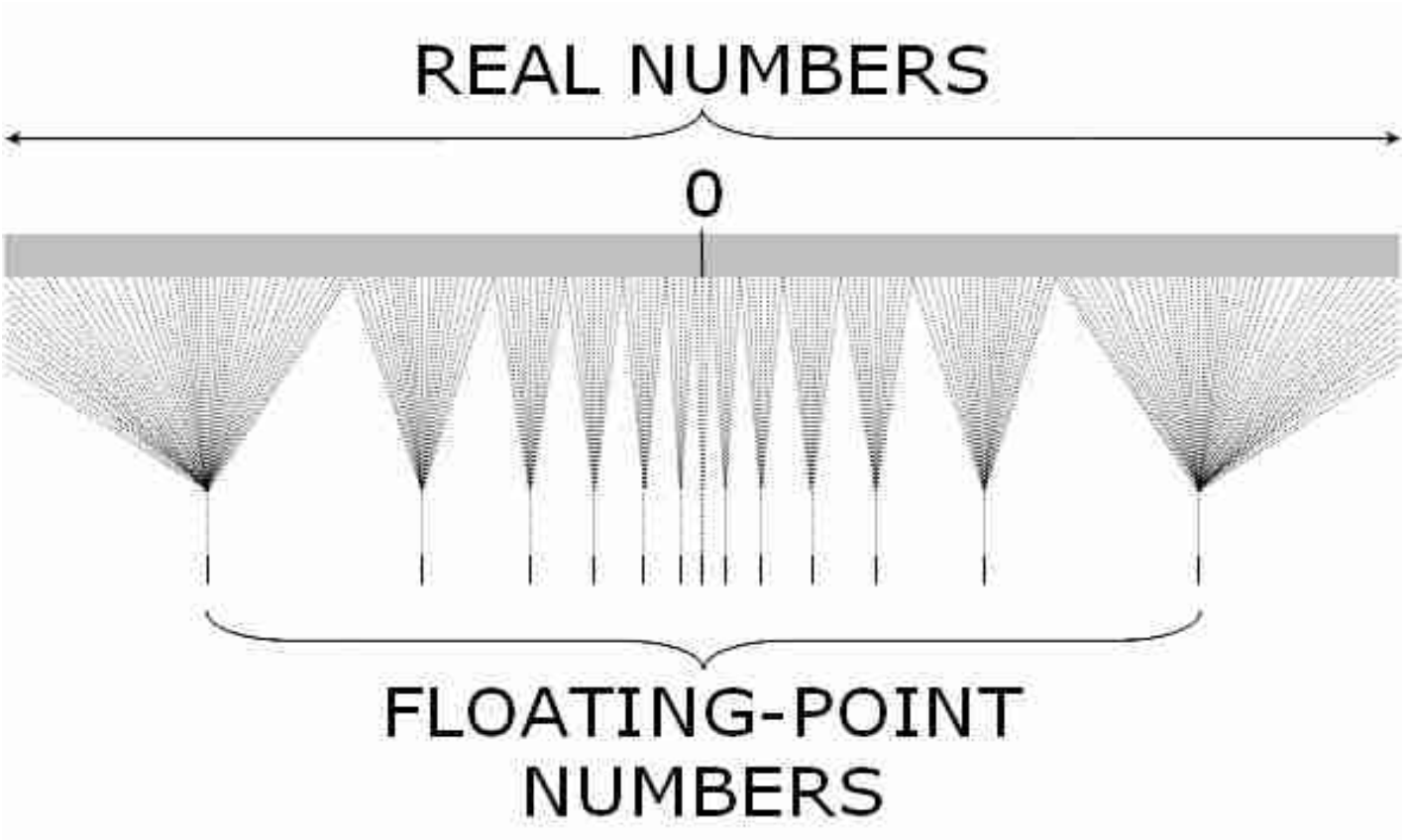


Memory Organization: Tip Vectors



Floating Point Numbers

- Machine numbers are an imperfect mapping of the infinite real numbers to a finite number of machine values!



Floating Point Arithmetics: The Root of All Evil

- Computational science mostly relies on floating-point intensive codes
- How do we verify these codes?
- We stand on shaky grounds
- Scientists using those codes assume that there are no bugs
- Double precision arithmetics required for certain applications
- Who knows what de-normalized floating point numbers are?

→ Please have a look at:

J. Björndalen, O. Anshus: “Trusting floating point benchmarks-are your benchmarks really data-independent?” Applied Parallel Computing. State of the art in Scientific Computing 2010; pp 178-188, Springer.

and at my micro-benchmark at:
https://github.com/stamatak/denormalizedFloatingPointNumbers

Floating Point Arithmetics: The Root of All Evil

- Computational science mostly reli[illegible]e codes
- How do we verify these codes?
- We stand on shaky grounds
- Scientists using those codes assume that there are no bugs
- Double precision arithmetics required for certain applications
- Who knows what de-normalized floating point numbers are?

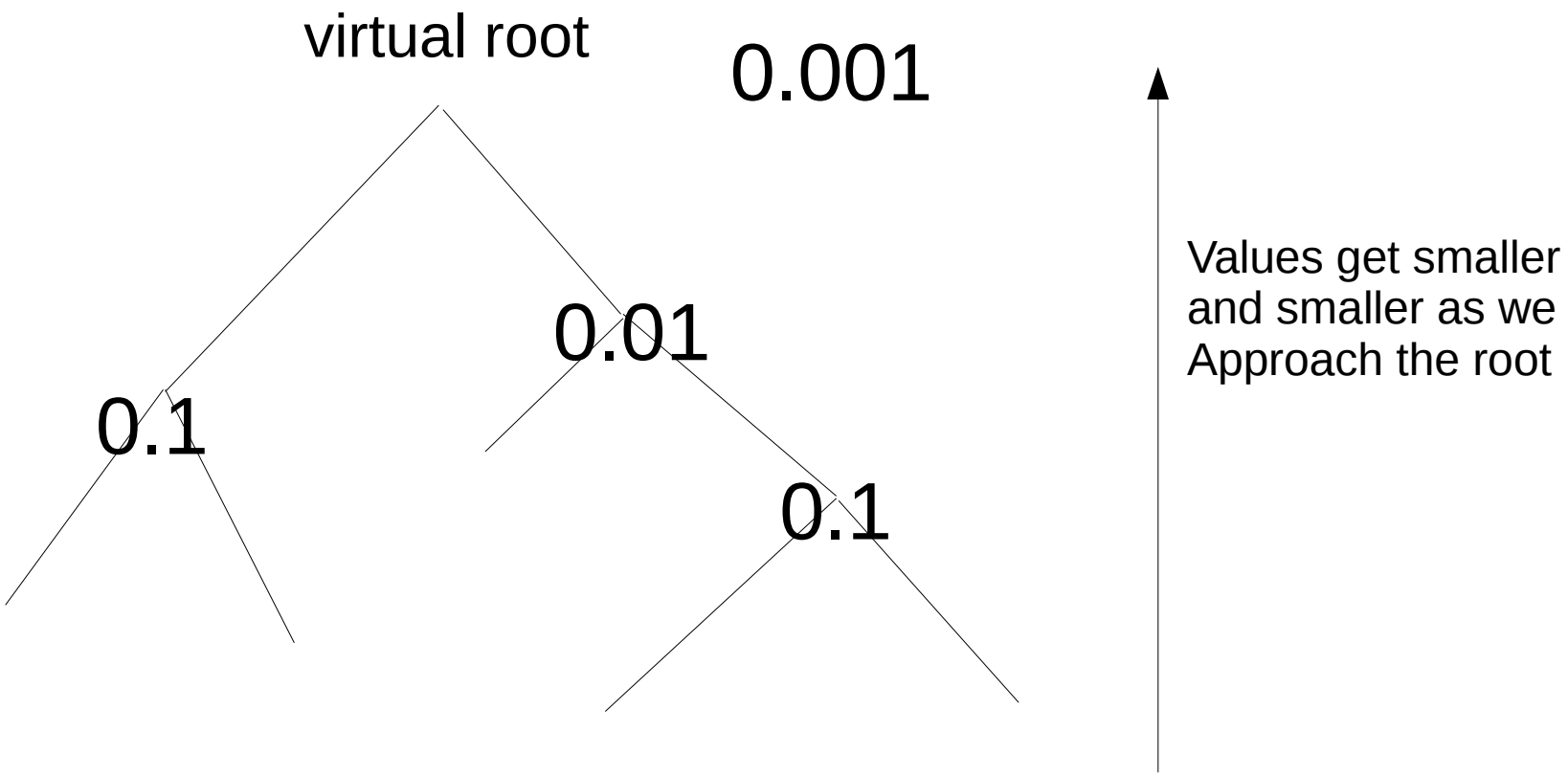
→ Please have a look at:

J. Björndalen, O. Anshus: “Trusting floating point benchmarks-are your benchmarks really data-independent?” Applied Parallel Computing. State of the art in Scientific Computing 2010; pp 178-188, Springer.

and at my micro-benchmark at:
https://github.com/stamatak/denormalizedFloatingPointNumbers

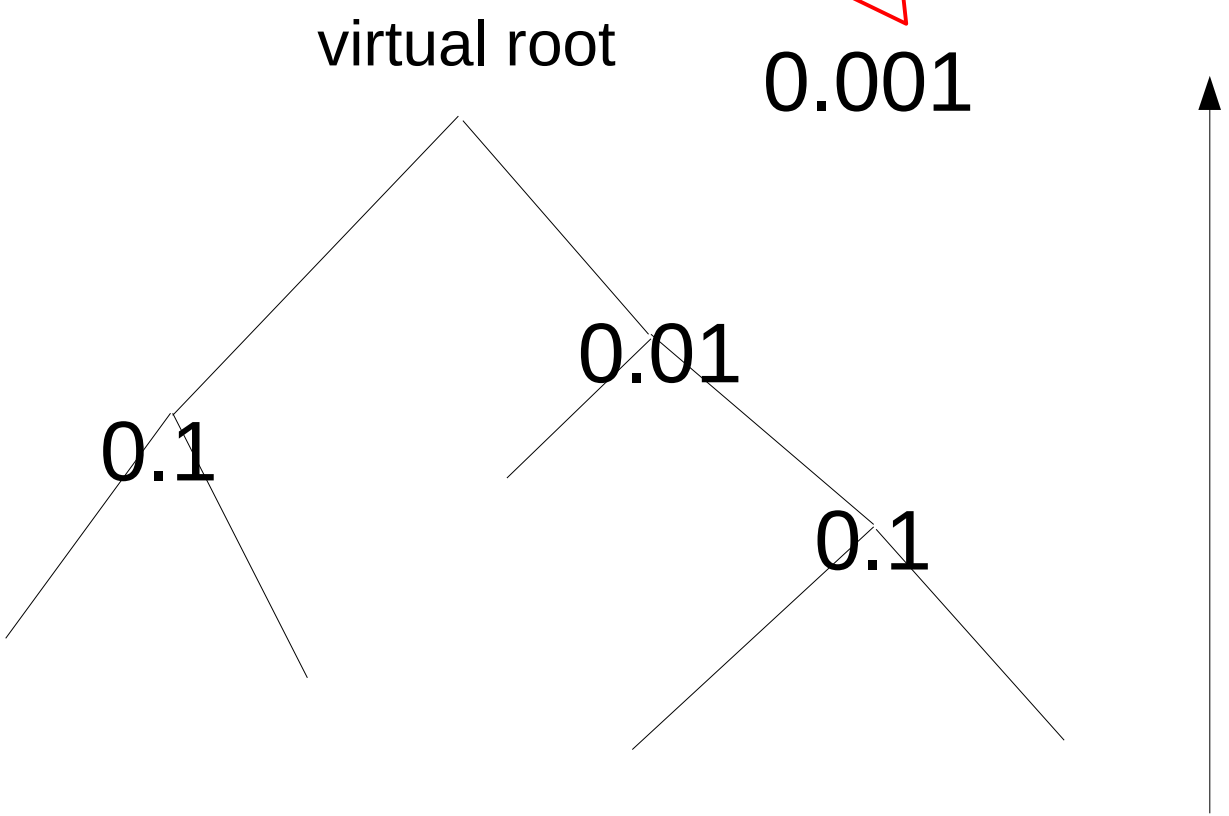
Why is this relevant when Talking about maximum Likelihood?

Post-order Traversal



Post-order Traversal

We need to apply numerical scaling techniques to avoid underflow!



Values get smaller and smaller as we Approach the root

Optimization of Likelihood Calculations

- Use SSE3 & AVX vector intrinsics
- Also: GPUs, FPGAs, the Intel Xeon PHI
- Special implementations for computing CLVs:

- I will spare you the details
- Key goal: avoid redundant computations