Introduction to Bioinformatics for Computer Scientists

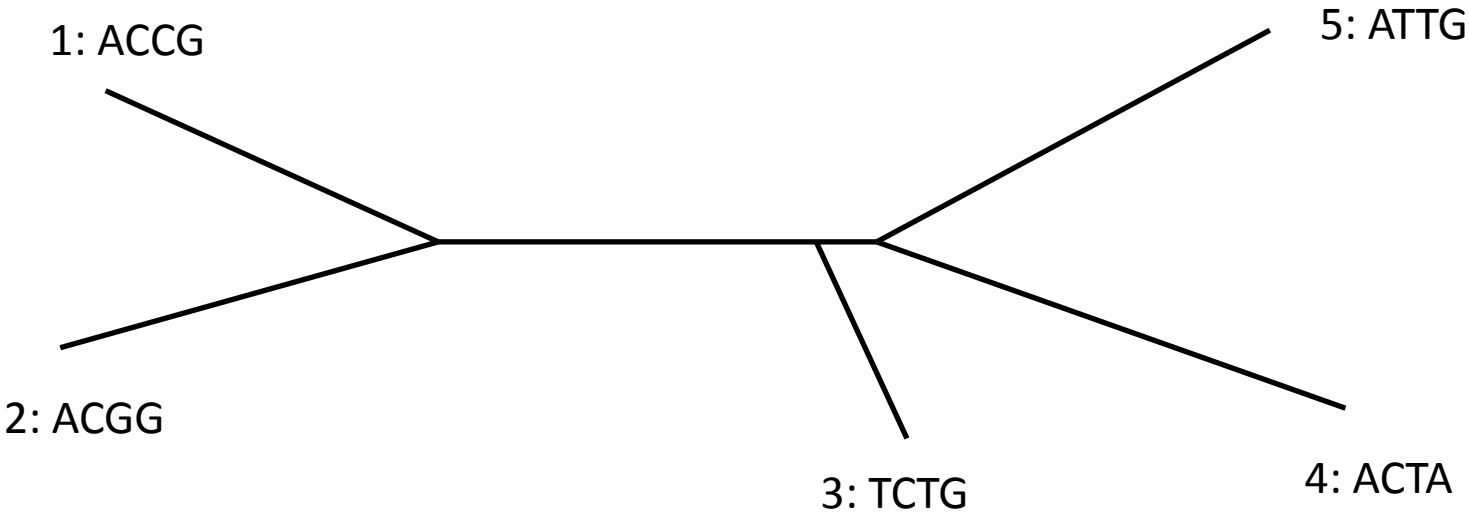
Lecture 8

Outline for today

Maximum Likelihood:

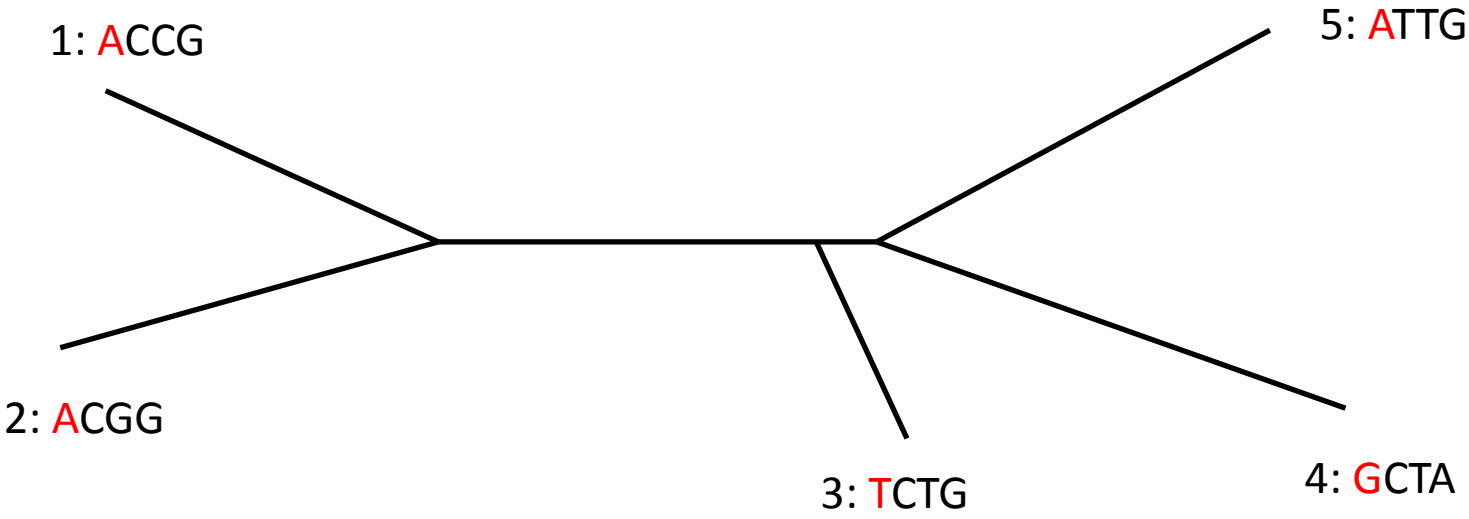
- Likelihood Function
- Models of Evolution
- Efficient Likelihood Evaluation
- Pairwise Distances
- Branch Lengths

Likelihood



$$LH(T|D) = P(D|T)$$

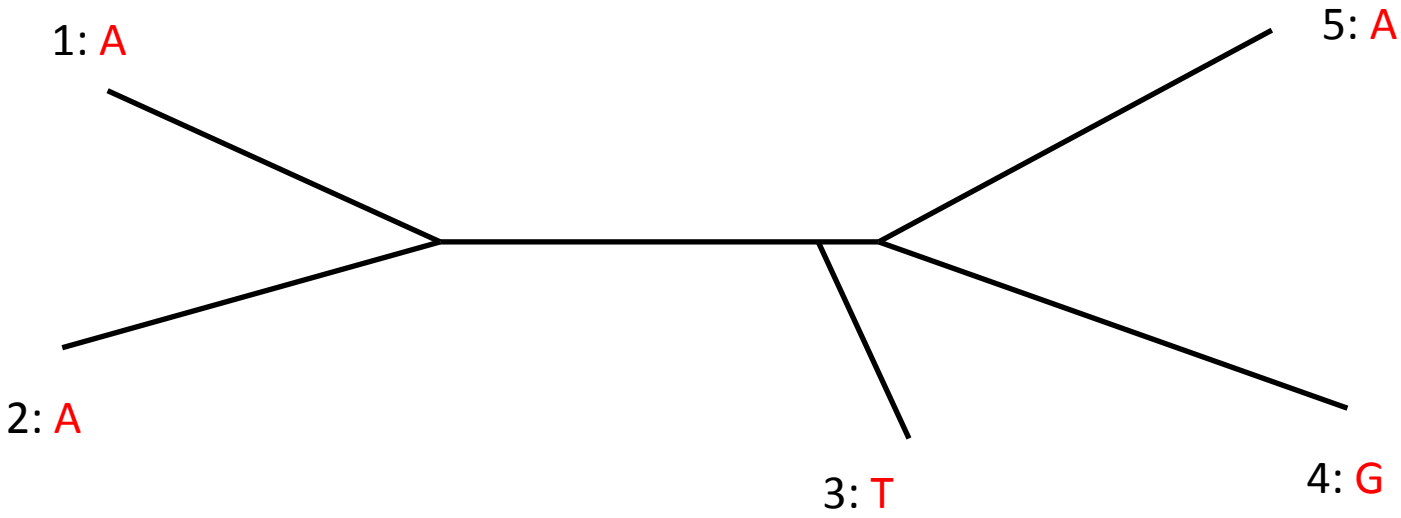
Likelihood



$$LH(T|D) = P(D|T)$$

$$LH(T|D) = \prod_{s_i \ sites} P(s_i|T)$$

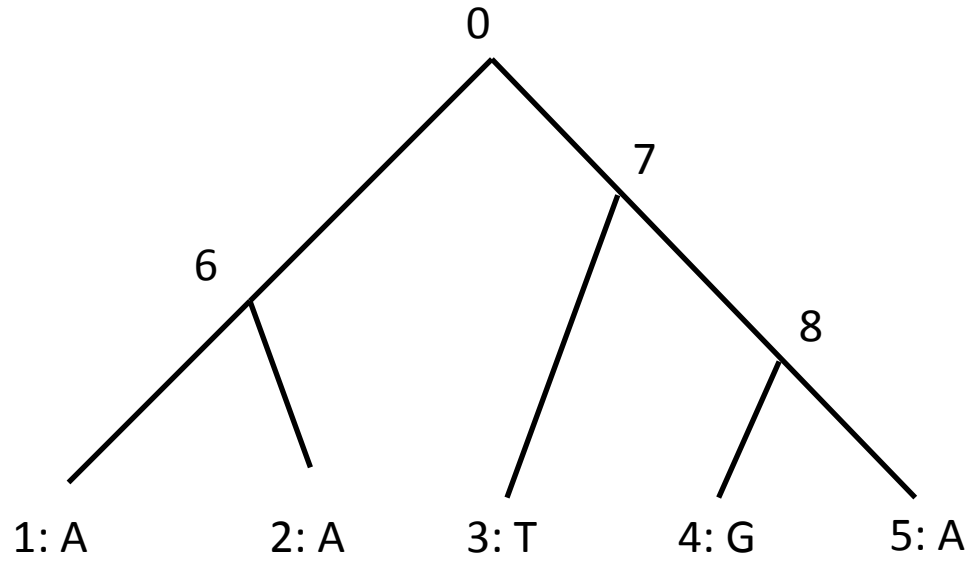
Likelihood



$$log(LH(T|D)) = \sum_{s_i \ sites} log(P(s_i|T))$$

BB

Likelihood



$$LH = \sum_{x_0} \sum_{x_6} \sum_{x_7} \sum_{x_8} (\prod_{x_0} \cdot p_{x_0,x_6} \cdot p_{x_0,x_7} \cdot p_{x_6,A} \cdot p_{x_6,A} \cdot p_{x_7,T} \cdot p_{x_7,x_8} \cdot p_{x_8,G} \cdot p_{x_8,G})$$

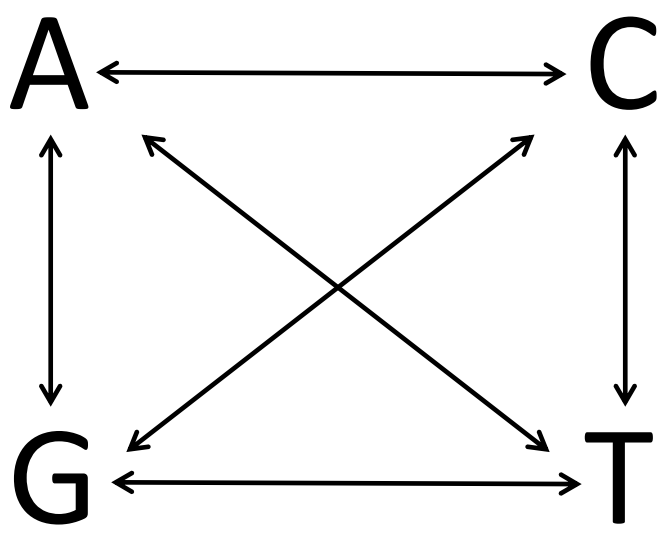
Likelihood:

How to get: p_{x_i,x_j}

Stochastic Process:

Seq 1 AGGGAG

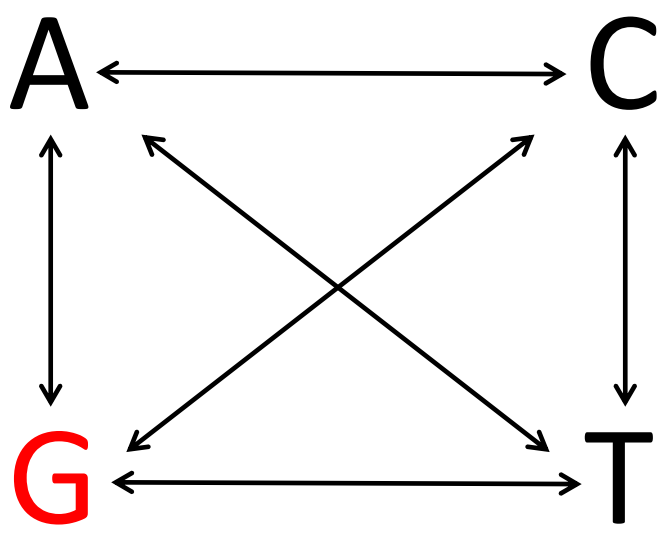
Seq 2 ACGGAA



Stochastic Process:

Seq 1 AGGGAG

Seq 2 ACGGAA

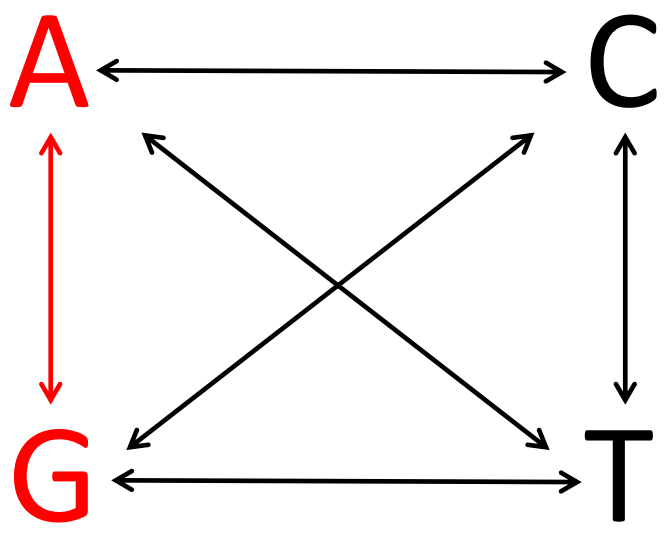


Seq 1: G

Stochastic Process:

Seq 1 AGGGAG

Seq 2 ACGGAA

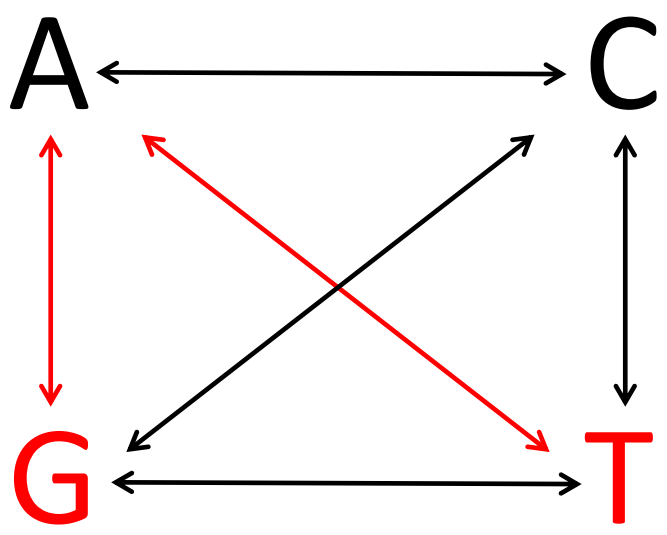


Seq 1: G ⟶ A

Stochastic Process:

Seq 1 AGGGAG

Seq 2 ACGGAA

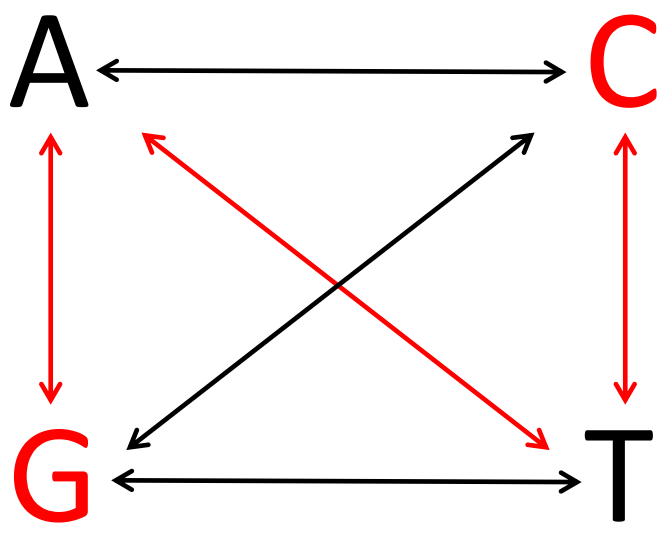


Seq 1: G ⟶ A ⟶ T

Stochastic Process:

Seq 1 AGGGAG

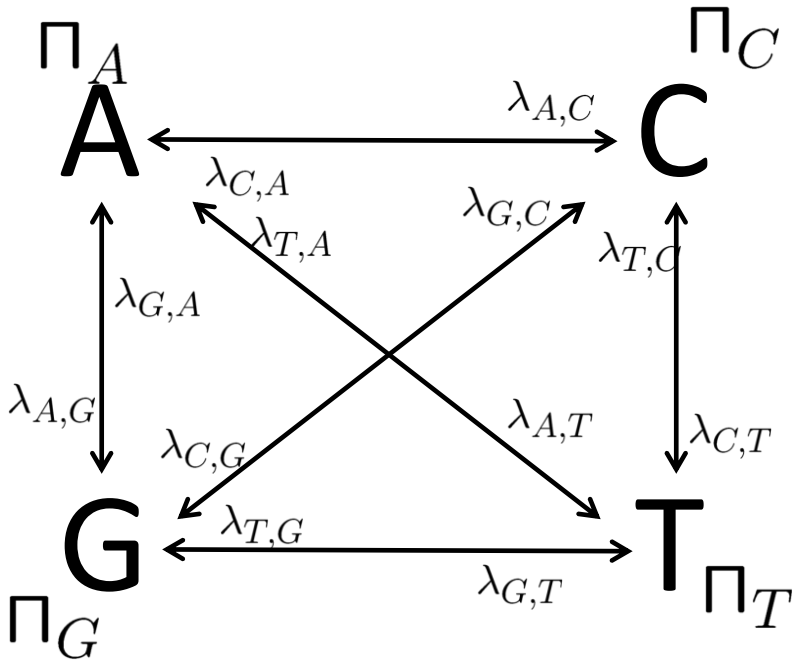
Seq 2 ACGGAA



Seq 1: G ⟶ A ⟶ T ⟶ C : Seq 2

BB

Stochastic Process:



Jukes-Cantor (JC):

$\lambda_{i,j} = \lambda_{j,i}, \Pi_i = \Pi_j$

•

•

•

General Time Reversible (GTR):

$\Pi_i \cdot \lambda_{i,j} = \lambda_{j,i} \cdot \Pi_j$

From Π and λ we construct the Q-matrix Q

Stochastic Process:

We want:

$P(S_t = Y | S_0 = X) := P_{X,Y}(t)$

i.e., The probability to end up in state Y after time t, when starting in state X

Note that our models are time reversible.

(i.e., $\Pi_i \cdot \lambda_{i,j} = \lambda_{j,i} \cdot \Pi_j$)

$\Rightarrow \Pi_X \cdot P_{X,Y}(t) = \Pi_Y \cdot P_{Y,X}(t)$

Stochastic Process:

Jukes-Cantor

$$P_{i,j}(t) = \frac{1}{4}(1 - e^{-\frac{4}{3}t})$$

BB

Stochastic Process:

In General:

$$P(t) = e^{Q \cdot t}$$

Stochastic Process:

Spectral Decomposition:

$$P(t) = e^{Q \cdot t} \ , \quad Q = U \Lambda U^{-1}$$

$$\Rightarrow P(t) = U e^{\Lambda \cdot t} U^{-1}$$

$$= U diag(e^{\Lambda_i \cdot t}) U^{-1}$$

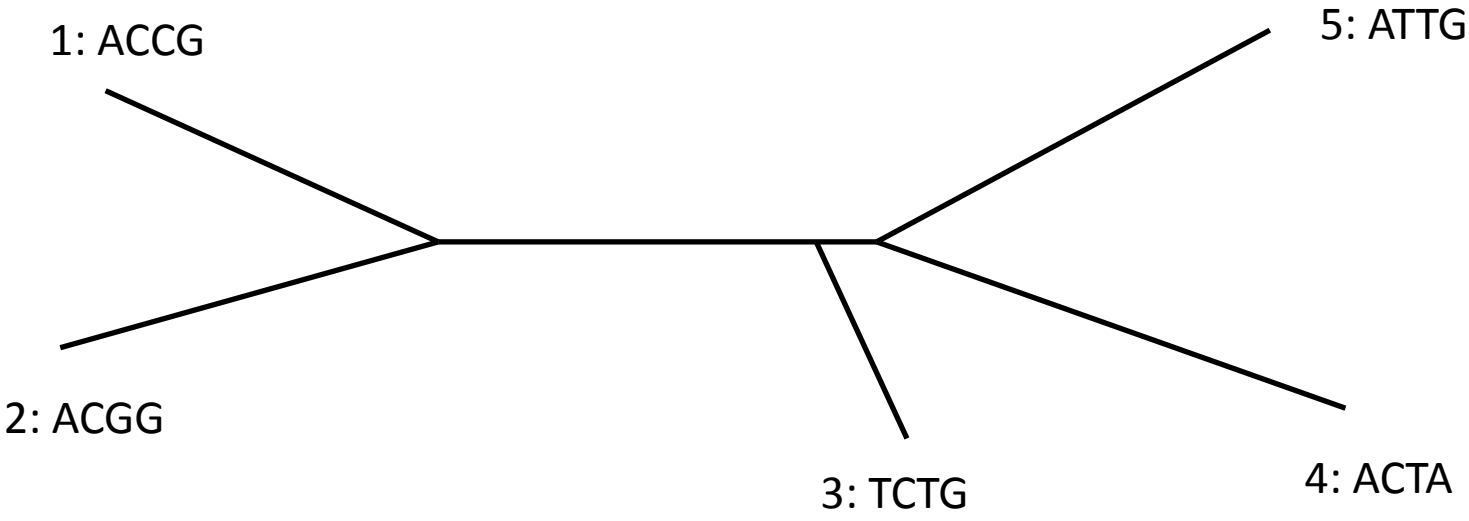
Easy to compute

BB

Efficient Likelihood Computation

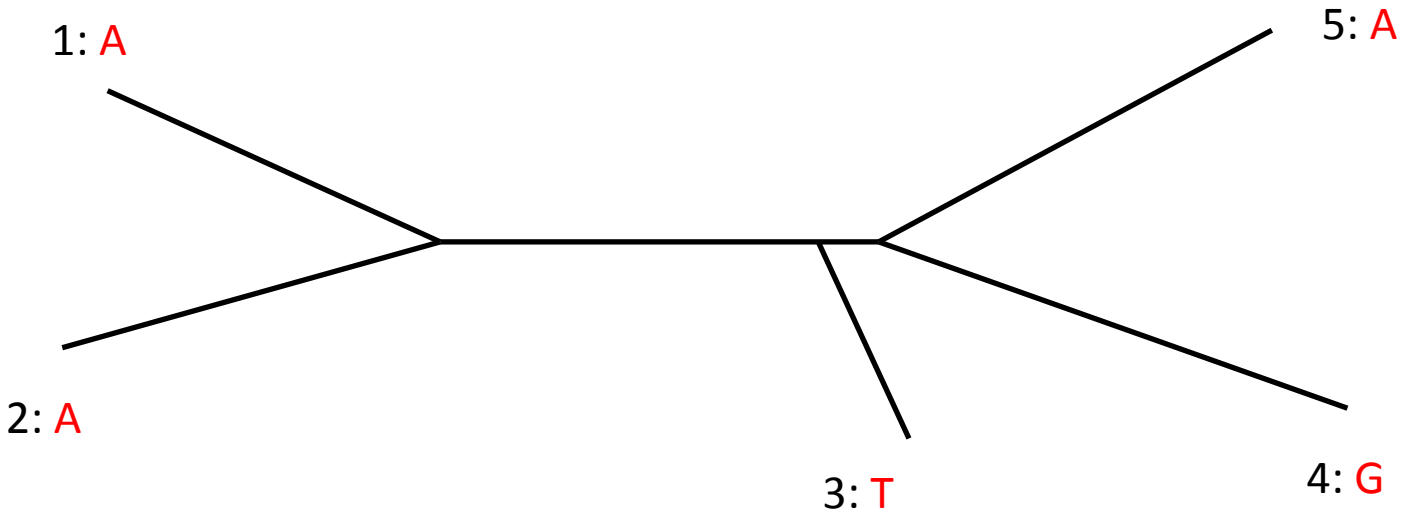
- Dynamic Programming similar to Parsimony
- Felsenstein-Pruning Algorithm

Likelihood



$$LH(T|D) = P(D|T)$$

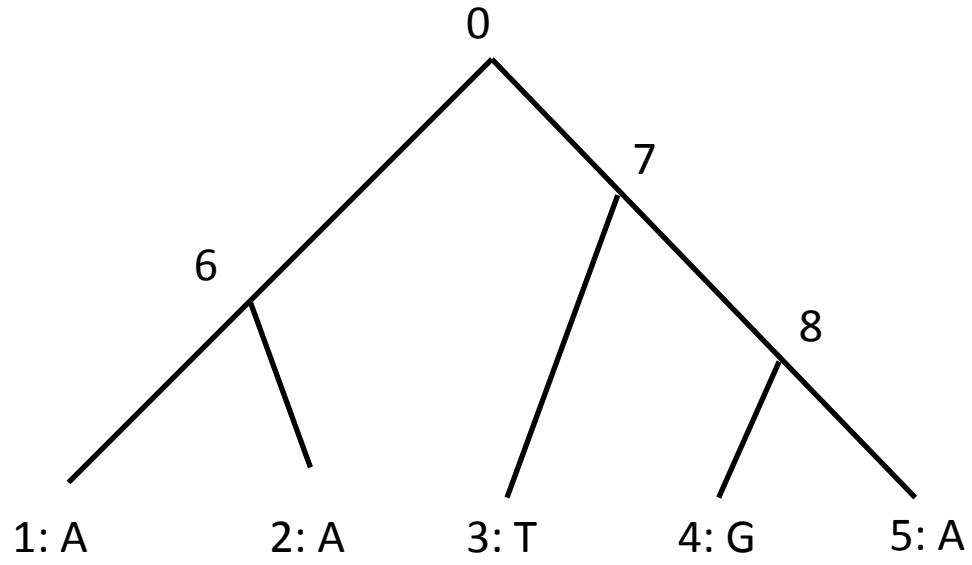
Likelihood



$$log(LH(T|D)) = \sum_{s_i \ sites} log(P(s_i|T))$$

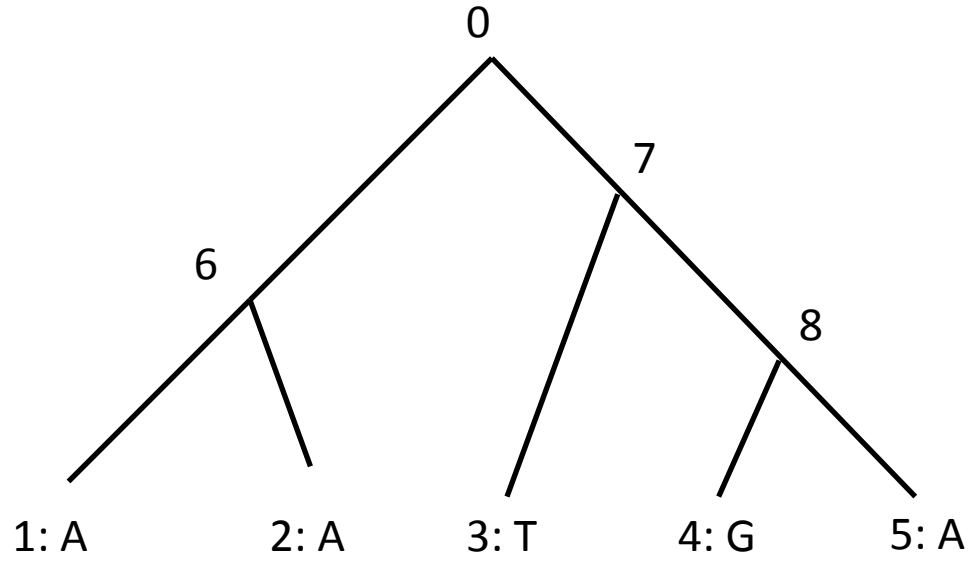
BB

Likelihood



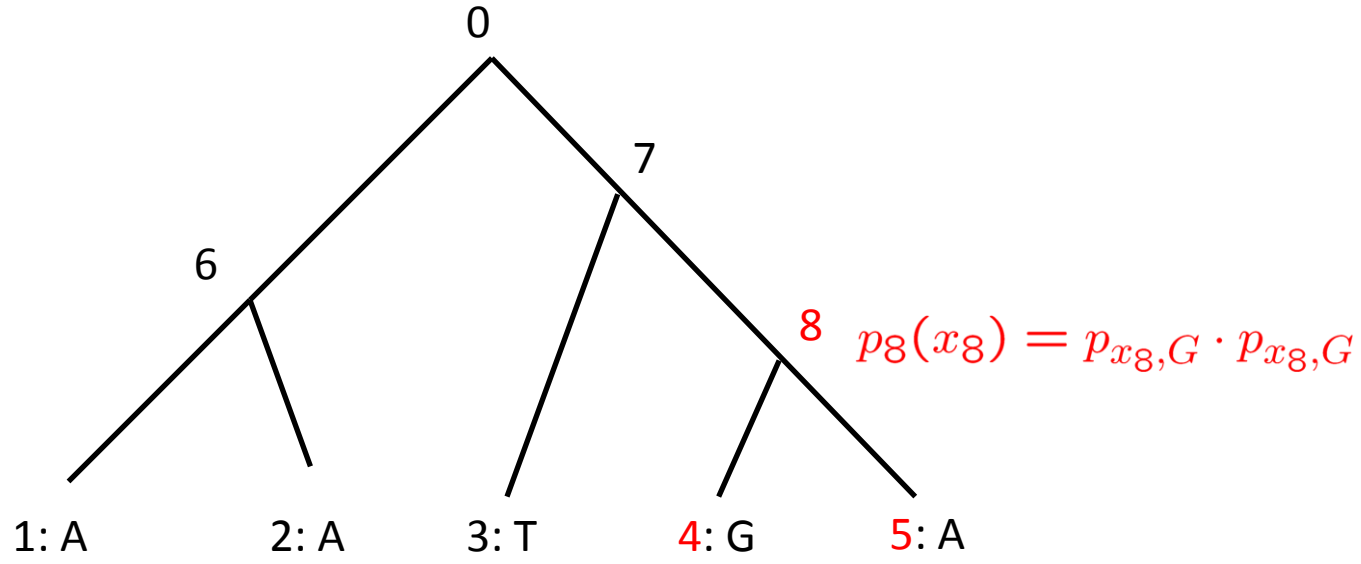
$$LH = \sum_{x_0} \sum_{x_6} \sum_{x_7} \sum_{x_8} (\prod_{x_0} \cdot p_{x_0,x_6} \cdot p_{x_0,x_7} \cdot p_{x_6,A} \cdot p_{x_6,A} \cdot p_{x_7,T} \cdot p_{x_7,x_8} \cdot p_{x_8,G} \cdot p_{x_8,G})$$

Likelihood: Pruning Algorithm



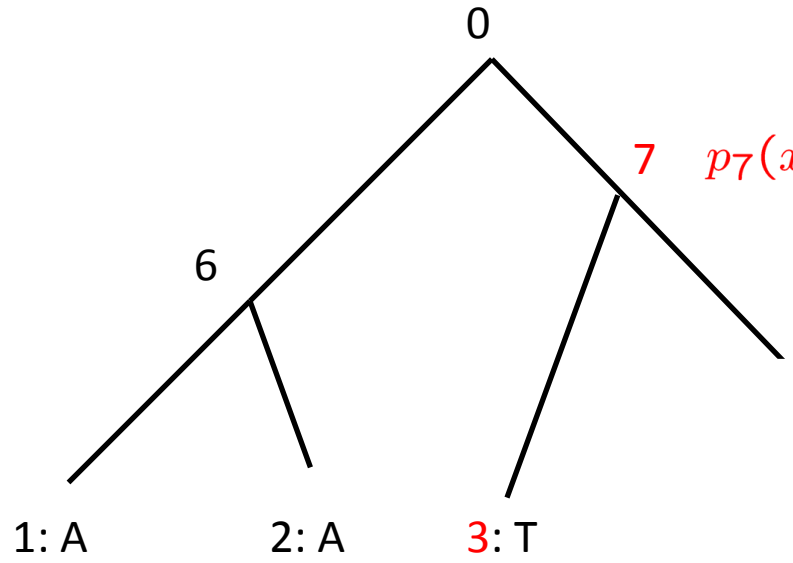
$$LH = \sum_{x_0}(\sqcap_{x_0} \cdot \sum_{x_6}(p_{x_0,x_6} \cdot p_{x_6,A} \cdot p_{x_6,A})$$
$$\cdot \sum_{x_7}(p_{x_0,x_7} p_{x_7,T} \cdot \sum_{x_8}(p_{x_7,x_8} \cdot p_{x_8,G} \cdot p_{x_8,G})))$$

Likelihood: Pruning Algorithm



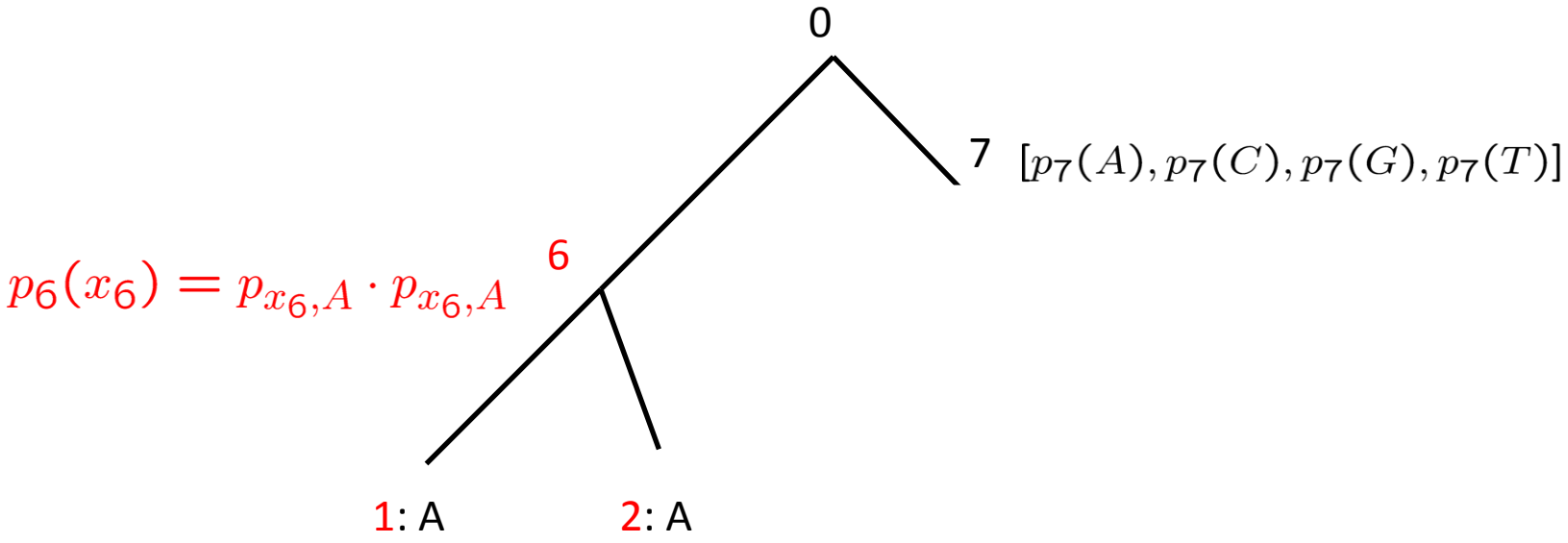
$$LH = \sum_{x_0} (\Pi_{x_0} \cdot \sum_{x_6} (p_{x_0,x_6} \cdot p_{x_6,A} \cdot p_{x_6,A})$$
$$\cdot \sum_{x_7} (p_{x_0,x_7} p_{x_7,T} \cdot \sum_{x_8} (p_{x_7,x_8} \cdot p_{x_8,G} \cdot p_{x_8,G})))$$

Likelihood: Pruning Algorithm



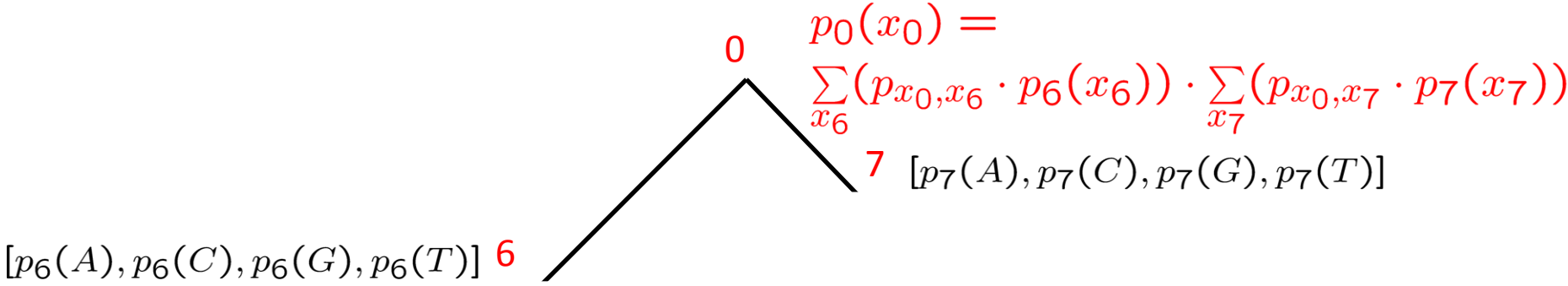
$$LH = \sum_{x_0}(\Pi_{x_0} \cdot \sum_{x_6}(p_{x_0,x_6} \cdot p_{x_6,A} \cdot p_{x_6,A})$$
$$\cdot \sum_{x_7}(p_{x_0,x_7} p_{x_7,T} \cdot \sum_{x_8}(p_{x_7,x_8} \cdot p_8(x_8))))$$

Likelihood: Pruning Algorithm



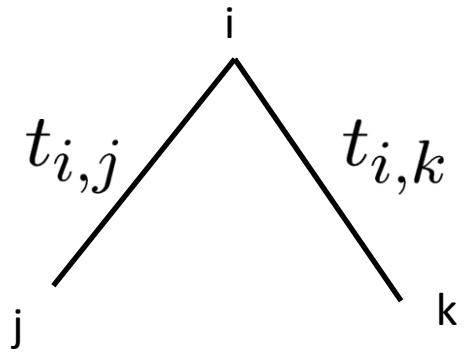
$$LH = \sum_{x_0} (\Pi_{x_0} \cdot \sum_{x_6} (p_{x_0,x_6} \cdot p_{x_6,A} \cdot p_{x_6,A}) \cdot \sum_{x_7} (p_{x_0,x_7} p_7(x_7)))$$

Likelihood: Pruning Algorithm



$$LH = \sum_{x_0}(\Pi_{x_0} \cdot \sum_{x_6}(p_{x_0,x_6} \cdot p_6(x_6)) \cdot \sum_{x_7}(p_{x_0,x_7} p_7(x_7)))$$

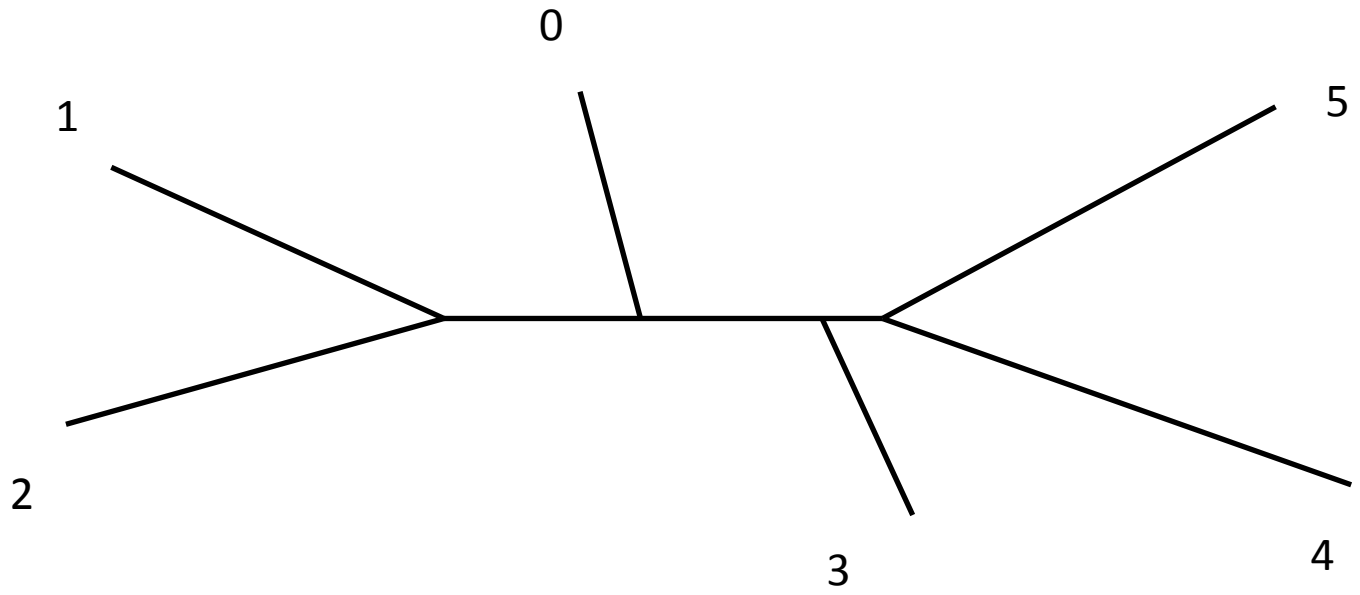
Likelihood: Pruning Algorithm



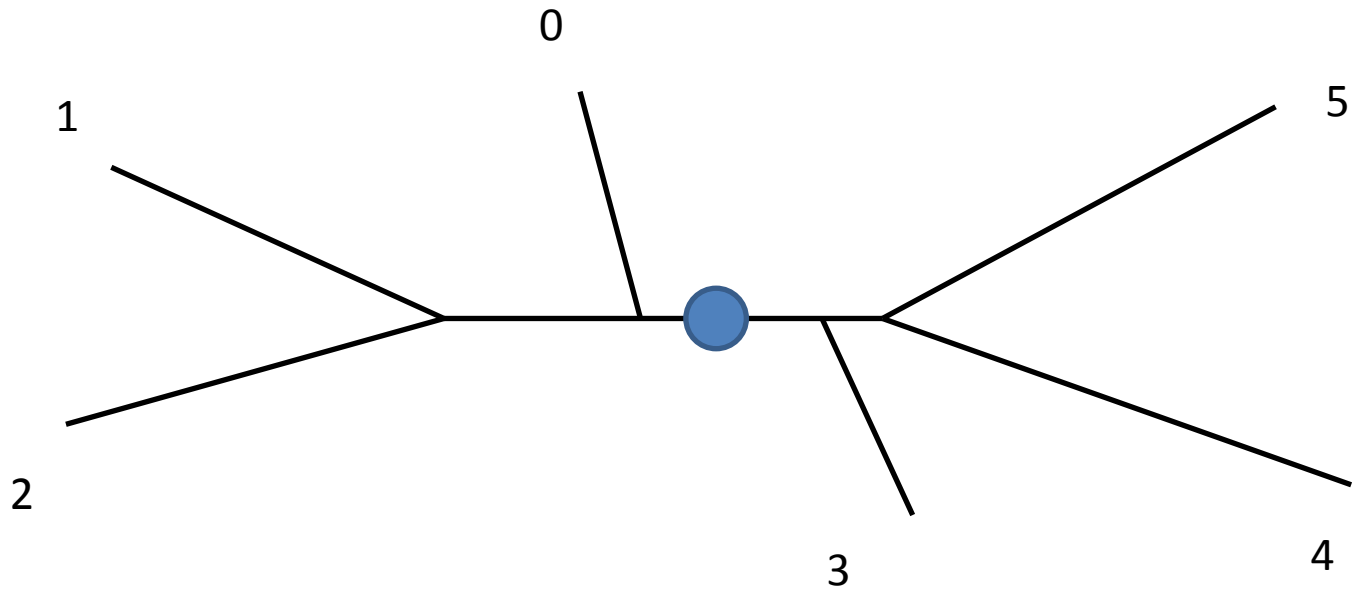
$$L_i(x_i) =$$

$$\sum_{x_j}(p_{x_i,x_j}(t_{i,j})L_j(x_j)) \cdot \sum_{x_k}(p_{x_i,x_k}(t_{i,k})L_k(x_k))$$

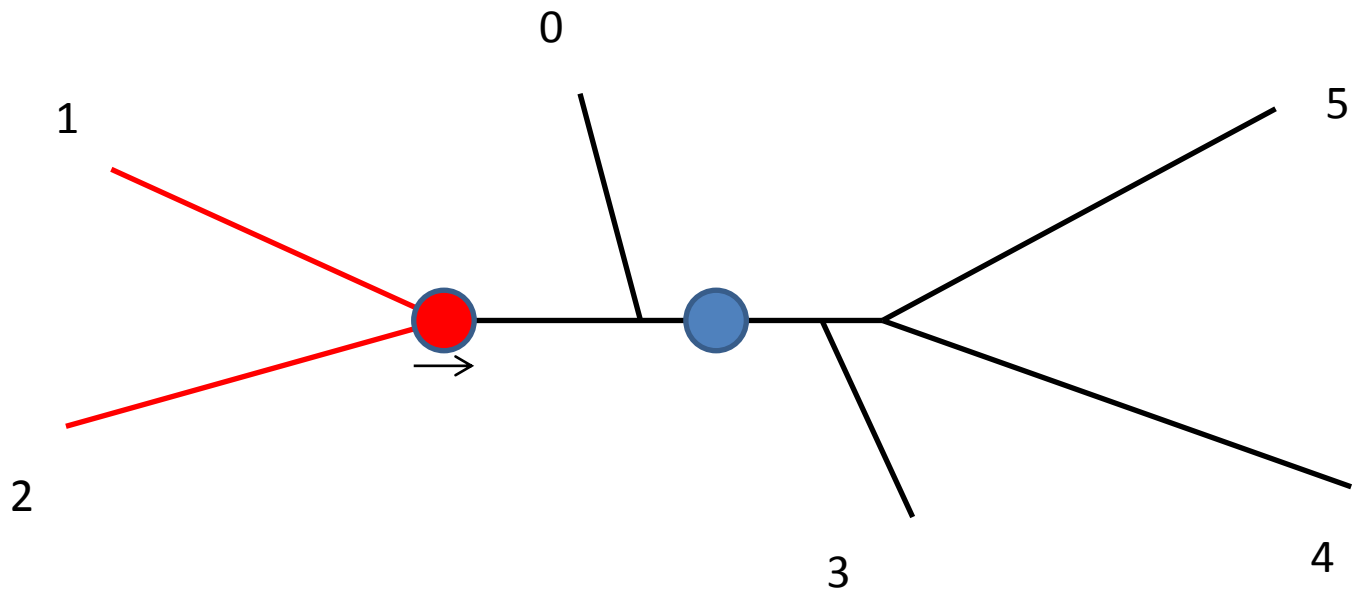
Likelihood: Tree Rearrangement



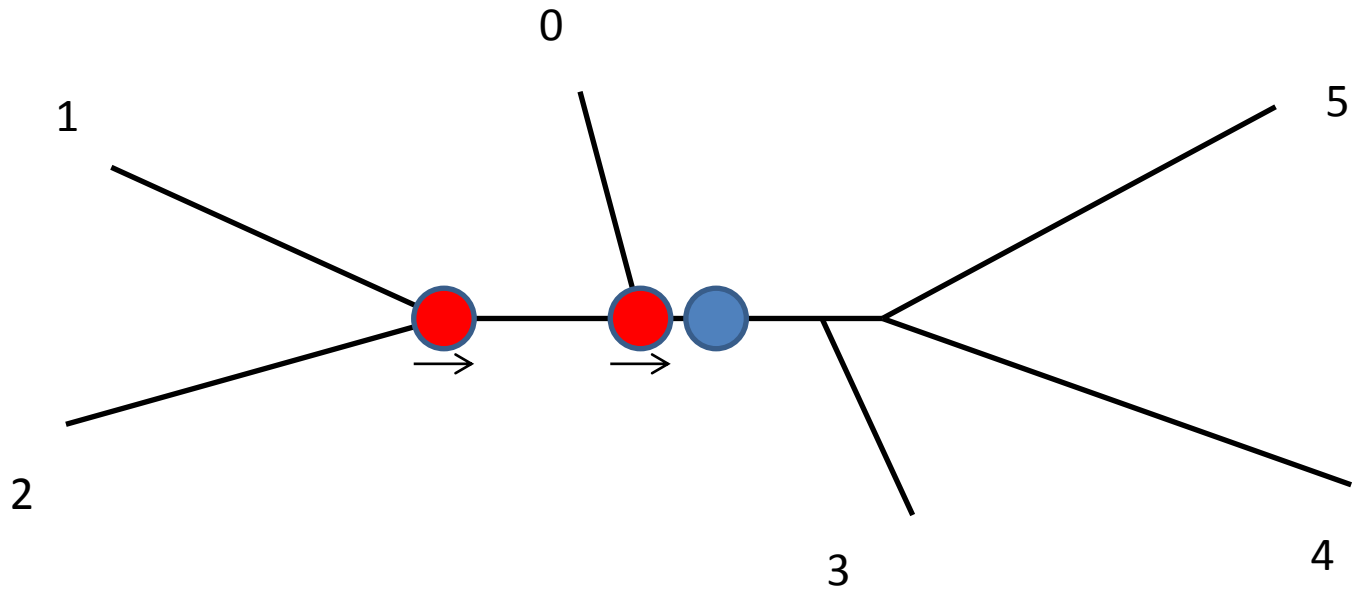
Likelihood: Tree Rearrangement



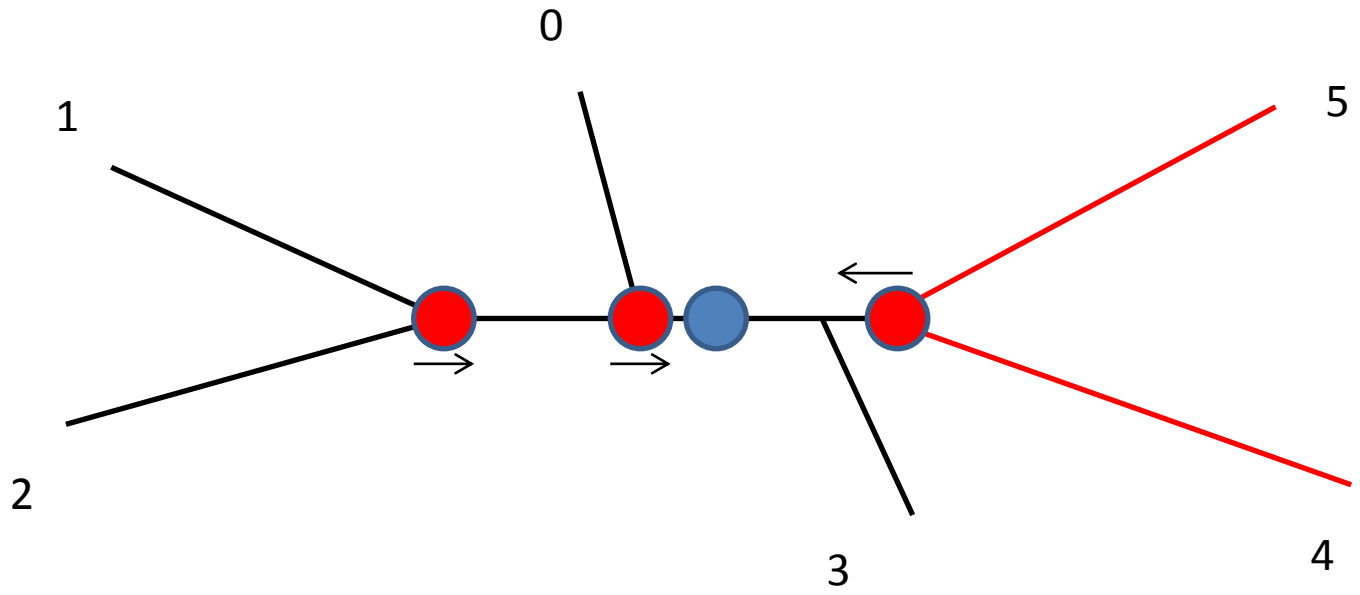
Likelihood: Tree Rearrangement



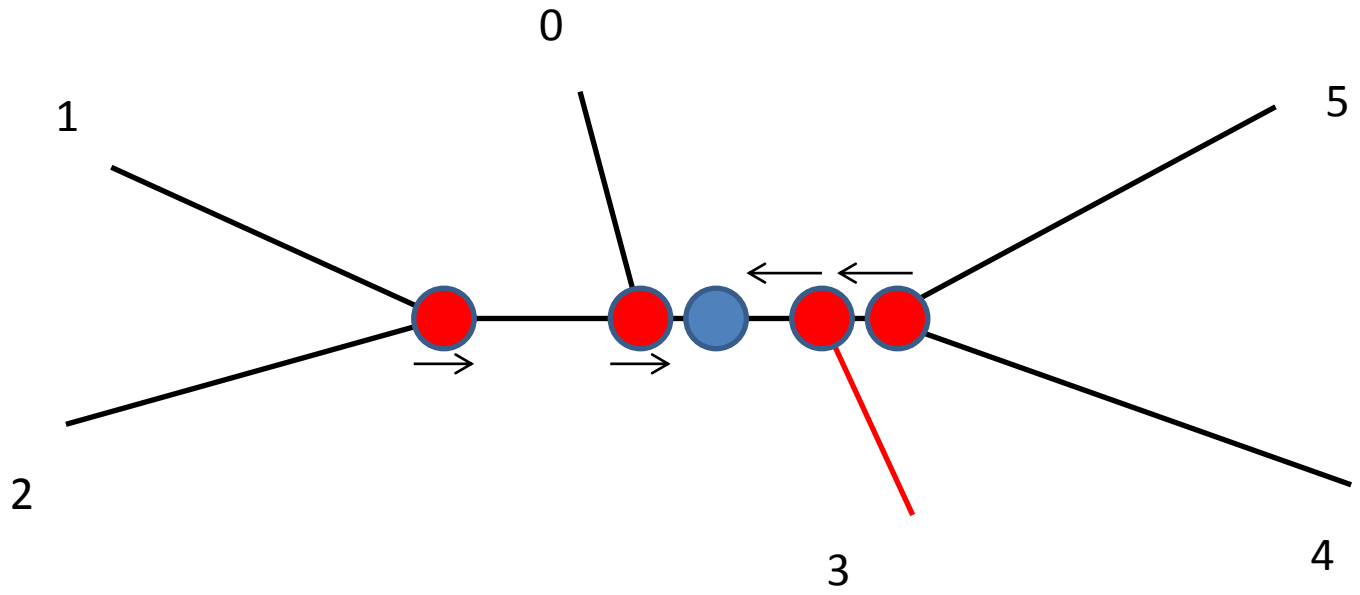
Likelihood: Tree Rearrangement



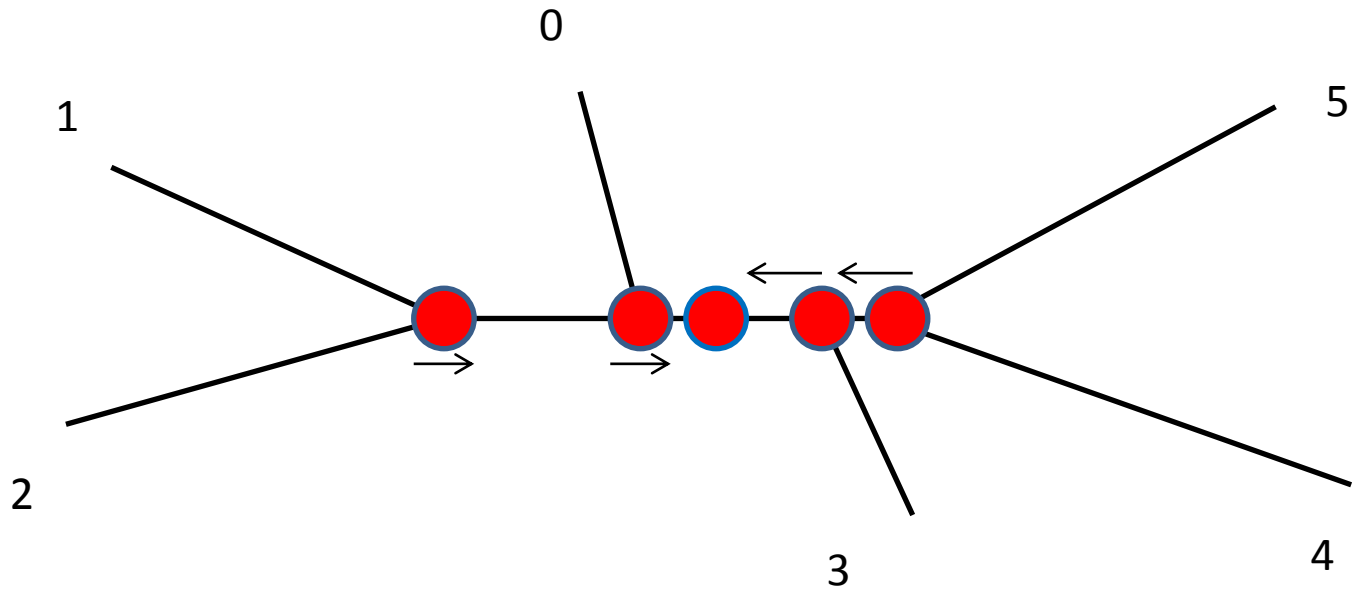
Likelihood: Tree Rearrangement



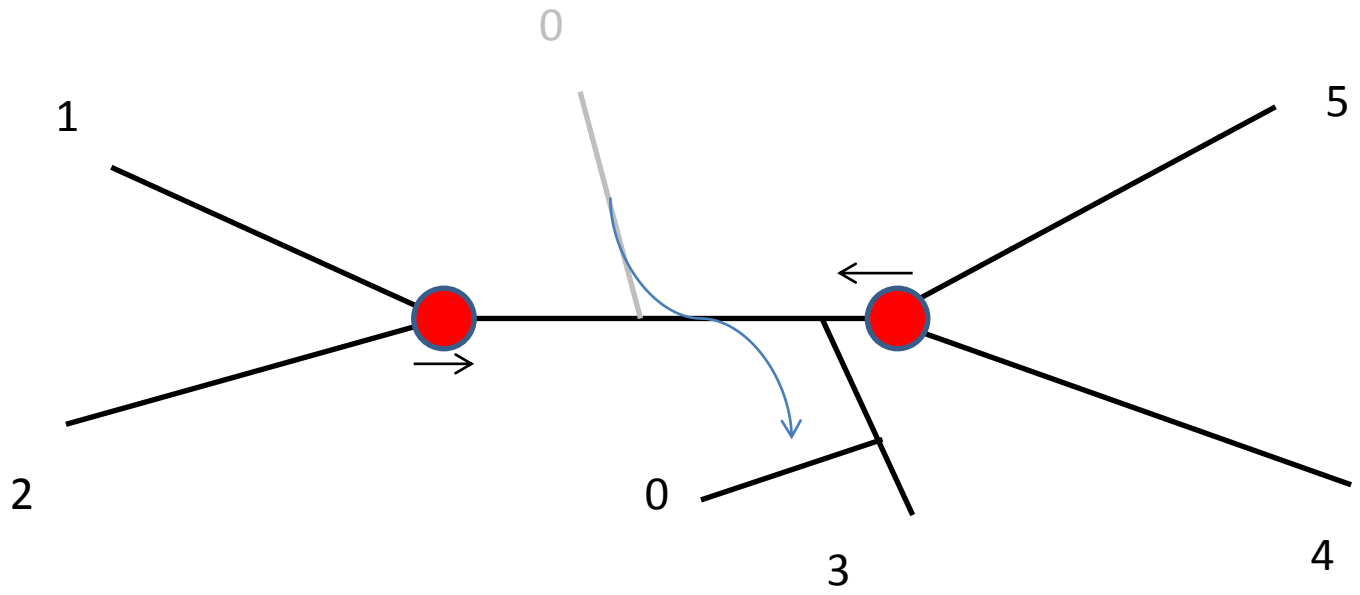
Likelihood: Tree Rearrangement



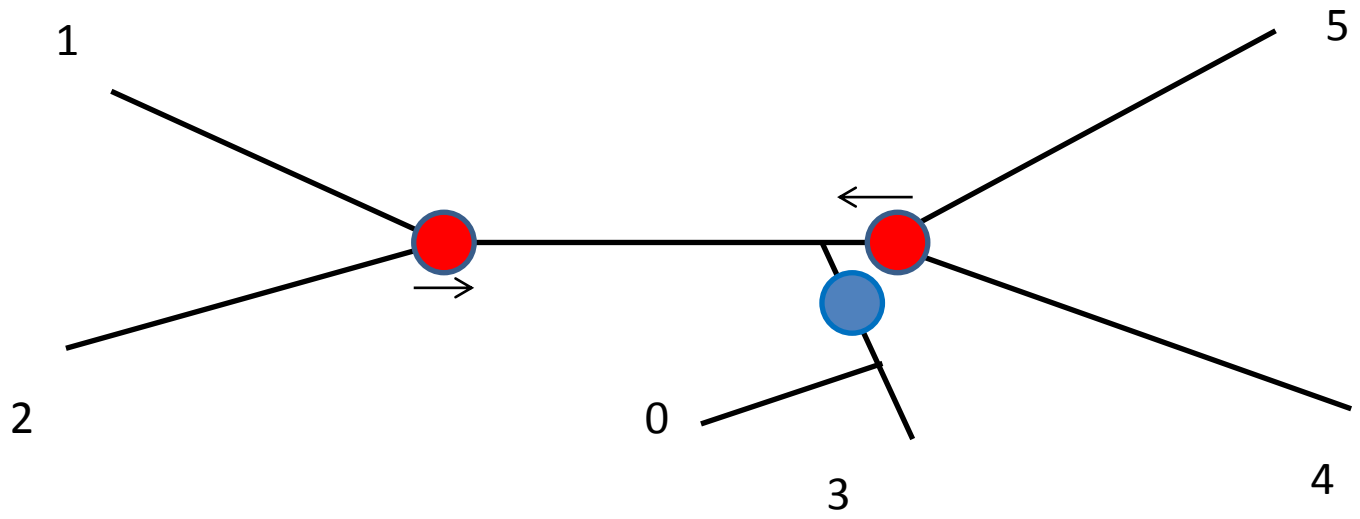
Likelihood: Tree Rearrangement



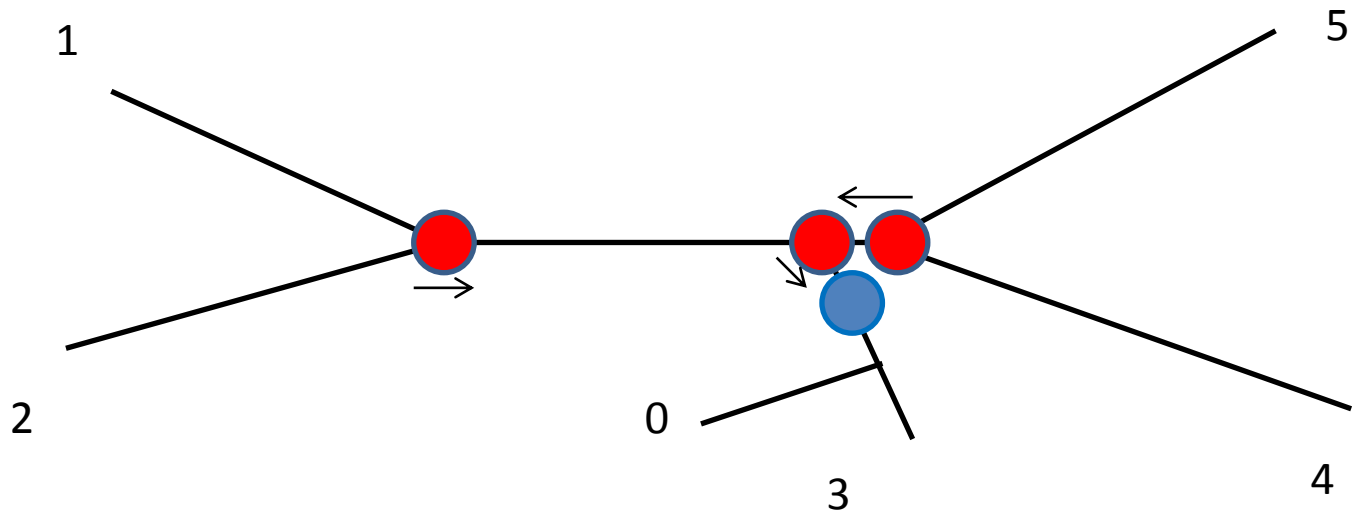
Likelihood: Tree Rearrangement



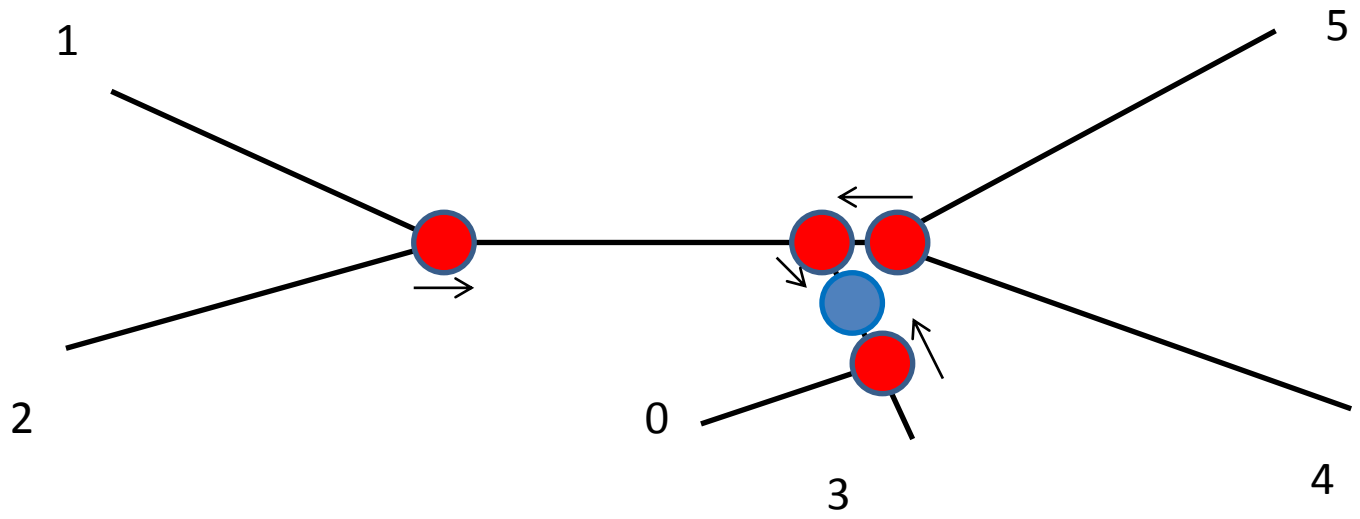
Likelihood: Tree Rearrangement



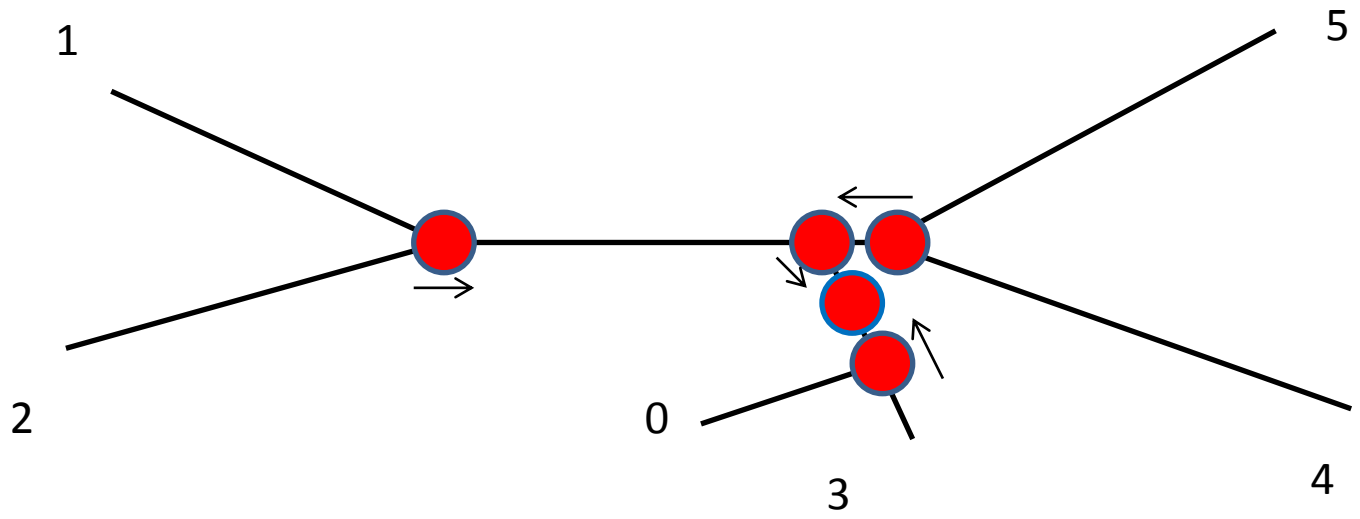
Likelihood: Tree Rearrangement



Likelihood: Tree Rearrangement



Likelihood: Tree Rearrangement



Rate Heterogeniety

Branch length

rate

$$P(t) = e^{Q \cdot b} = e^{Q \cdot ut}$$

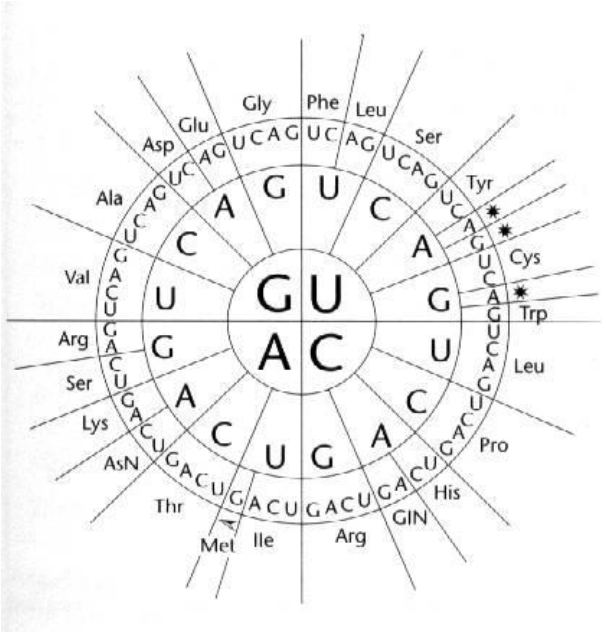
time

Rate Heterogeniety

Rates may differ among site:

$$P(t) = e^{Q \cdot u_i t}$$

One reason is the codon position:



Rate Heterogeniety

Per Site Rates:

- Optimize rates for each site
- Pick 20 best rates
- Assign all sites to one of these rates

$$LH(T|D) = \prod_{s_i \ sites} P(s_i|T, u_i)$$

Rate Heterogeniety

GAMMA Rates:

- Optimize one parameter for gamma function
- 4 rates from gamma distribution
- Average over these 4 rates for each site

$$LH(T|D) = \prod_{s_i} \frac{1}{4} \sum_{j=1}^{4} P(s_i|T, u_j)$$

Protein Substitution Models:

DNA Models: 4 states, 5 + 3 free parameters

Protein Models: 20 states, 189 +19 free parameters

Problem of overparametrization!

Solution: Use precomputed empirical models.

Empirical models: Optimize GTR matrix on large/huge alignments

BB

Pairwise Distances:

Seq 1 AGGGAG

Seq 2 ACGGAA

Distance between seq 1 and 2?

i.e., How many changes?

Pairwise Distances:

Seq 1 AGGGAG

Seq 2 ACGGAA

Distance between seq 1 and 2?

i.e., How many changes?

2 changes visible…

Pairwise Distances:

Seq 1 AGGGAG

Seq 2 ACGGAA

Distance between seq 1 and 2?

i.e., How many changes?

2 changes visible…

Seq 1: G ⟶ A ⟶ T ⟶ C : Seq 2 , 3 changes

Pairwise Distances:

Seq 1 AGGGAG

Seq 2 ACGGAA

Distance between seq 1 and 2?

i.e., How many changes?

2 changes visible…

Seq 1: G ⟶ A ⟶ T ⟶ C : Seq 2 , 3 changes

Seq 1: A ⟶ C ⟶ G ⟶ A : Seq 2 , 3 changes

Pairwise Distances:

Seq 1 AGGGAG

Seq 2 ACGGAA

Distance between seq 1 and 2?

i.e., How many changes?

$$d(seq_1, seq_2) = -\frac{3}{4} ln(1 - \frac{4}{3}\hat{p})$$

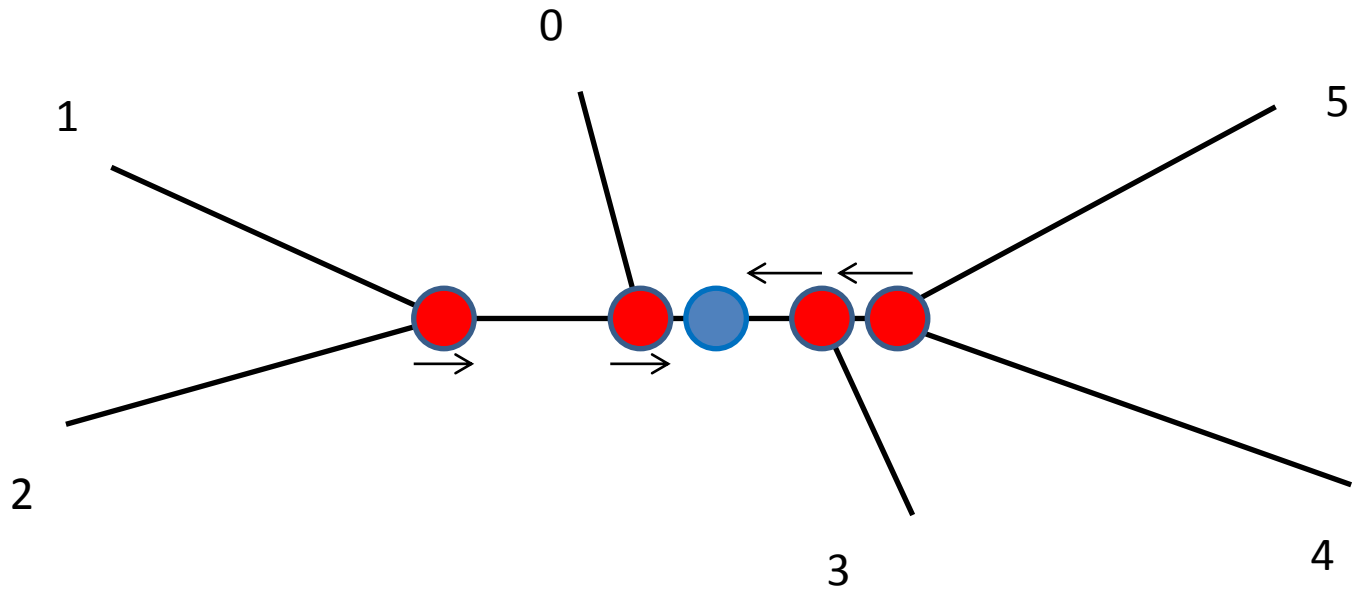
under the most simplistic model (Jukes-Cantor*)
i.e., equal rates and equal frequencies

$\hat{p}$ is the observed fractional difference between the sequences

*Jukes and Cantor, 1969, „Evolution of protein molecules“

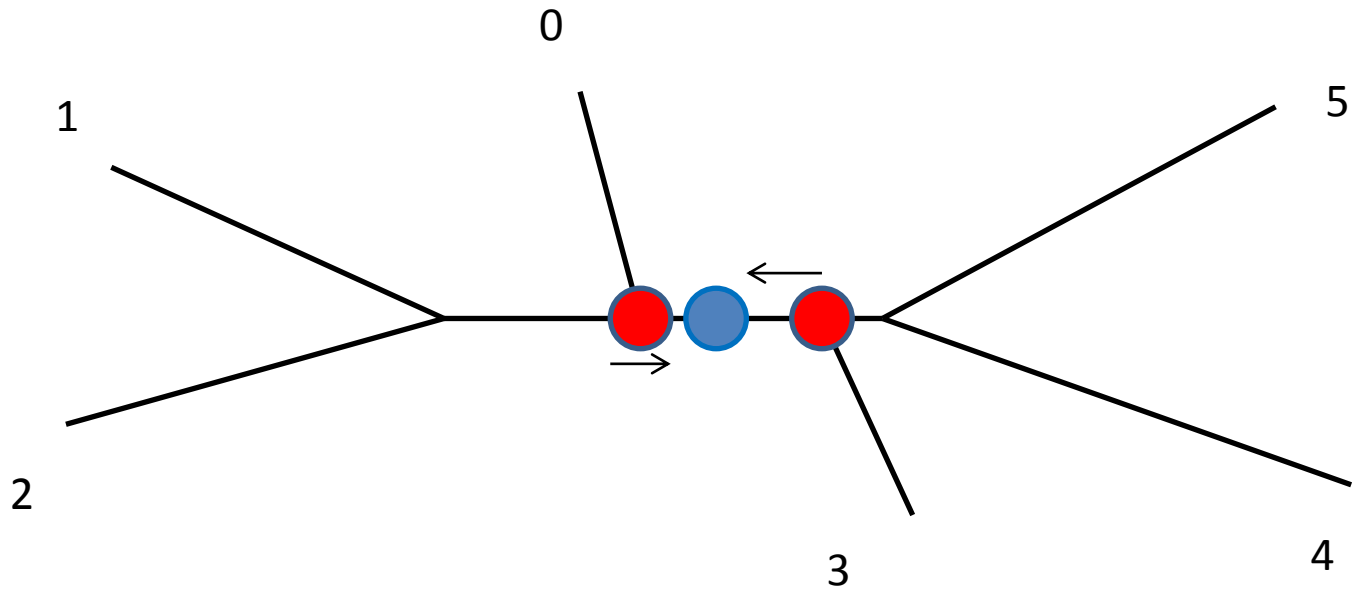
Likelihood:

Branch Lenght Optimization



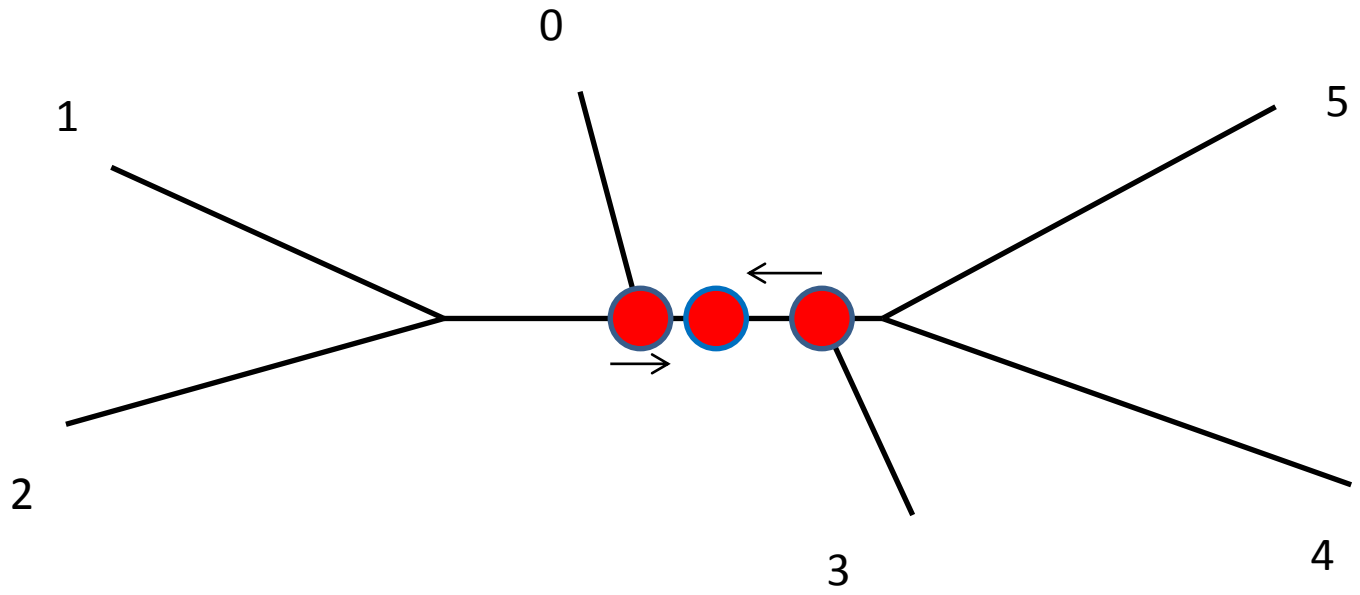
Likelihood:

Branch Lenght Optimization



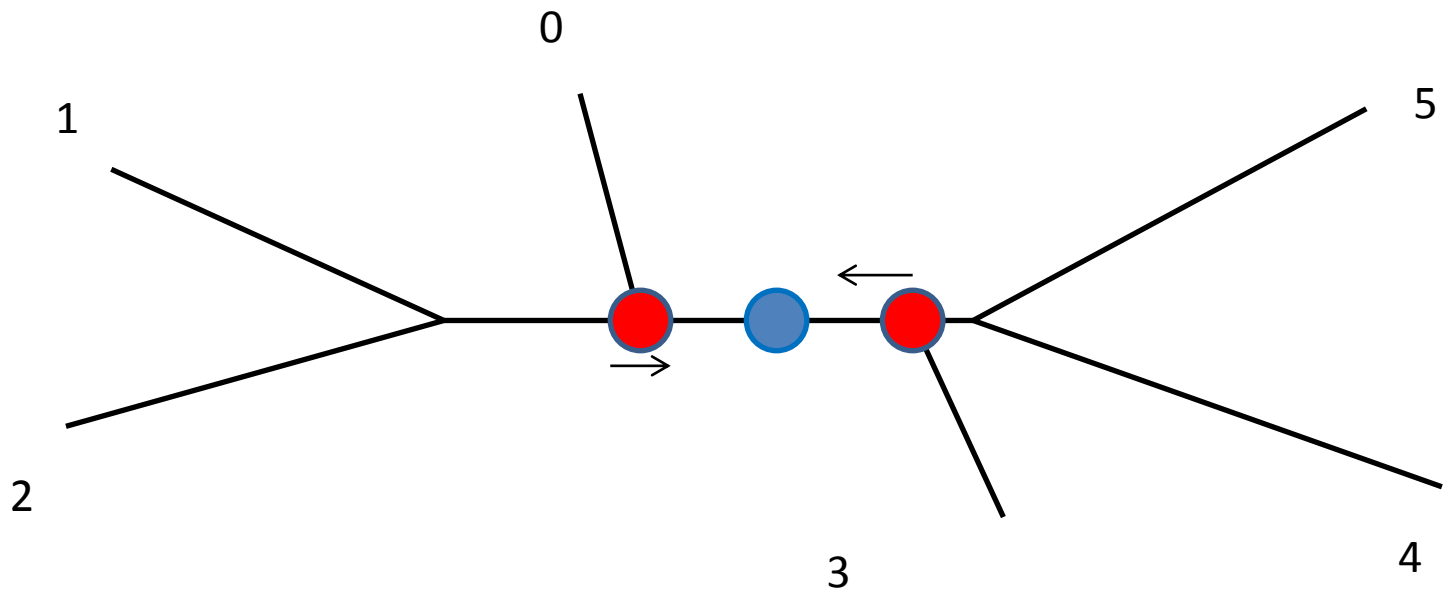
Likelihood:

Branch Lenght Optimization



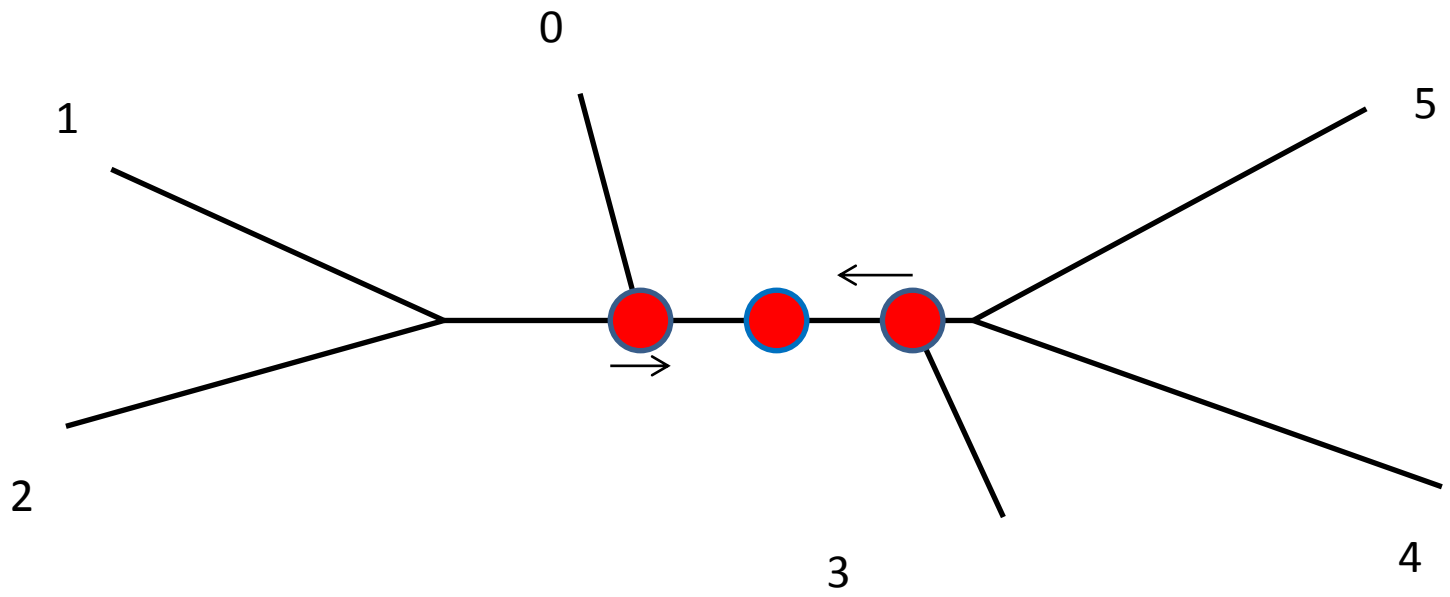
Likelihood:

Branch Lenght Optimization



Likelihood:

Branch Lenght Optimization



Likelihood:
Branch Lenght Optimization

Use Newton-Raphson

BB

Likelihood:
Branch Lenght Optimization

Recall: $P(t) = e^{Q \cdot t} = U e^{\Lambda \cdot t} U^{-1}$

$$\Rightarrow (P(t))' = U \Lambda e^{\Lambda \cdot t} U^{-1}$$
$$(P(t))'' = U \Lambda^2 e^{\Lambda \cdot t} U^{-1}$$

Thus we can apply standard optimization methods, e.g., Newton Raphson

Likelihood:

Newton´s Method

$$t_{next} = t - \frac{F'(t)}{F''(t)}$$

Maximum Likelihood

- Again NP-Hard
- Same Tree Search Heuristics as before
- Added dificulties:
  - Branch Lengths
  - Model Selection/Parameters
- One evaluation O(nm)