Tutorial 6.
Molecular Phylogenetics

The Chinese University of Hong Kong

BMEG3102 Bioinformatics

TA: Yizhen Chen

Agenda

- Phylogenetic tree reconstruction:
  - Given a set of DNA/protein sequences
  - Construct a tree that most likely refers to historical evolutionary events
    - Distance-based methods:
      - Unweighted Pair Group Method with Arithmetic mean (UPGMA)
      - Neighbor Joining (NJ)
    - Sequence-based methods:
      - Maximum parsimony
      - Maximum likelihood

Solving the problem: Ideas

- What do you do when you encounter a computationally hard problem?
  - Define an easier version of the problem
    - By making certain assumptions
  - Design smart algorithms/data structures to avoid redundant calculations
  - Use heuristics to solve it, not necessarily getting the optimal solution

UPGMA

- Algorithm:
  - Compute the distance between each pair of sequences (distance matrix)
  - Treat each sequence as a cluster
  - Merge the two closest clusters and update the distance matrix.
    - The distance between two clusters is the average distance between all their sequences:

$$\mathrm{d}(\mathrm{C_i}, \mathrm{C_j}) = \frac{1}{|\mathrm{C_i}||\mathrm{C_j}|} \sum_{r \in \mathrm{C_i}, s \in \mathrm{C_j}} \mathrm{d}(r, s)$$

  - Repeat previous steps until only one cluster is left

Exercise: UPGMA

- Given the distance matrix, construct the phylogenetic tree using UPGMA algorithm. Calculate the branch lengths (the tree is additive) and represent the tree in Newick format.
- Note: we will specify if we want the branch length in the question.

	A	B	C	D
A	0	11	4	11
B	11	0	13	4
C	4	13	0	13
D	11	4	13	0

Answer: UPGMA

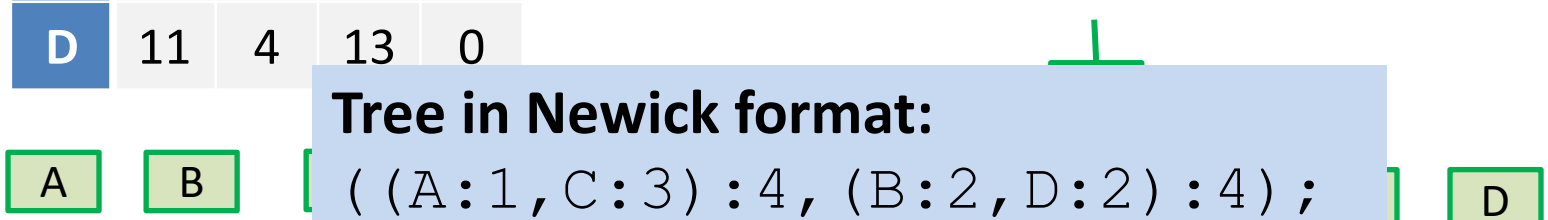
	A	B	C	D
A	0	11	4	11
B	11	0	13	4
C	4	13	0	13
D	11	4	13	0

{A},{C}

	AC	B	D
AC	0	12	12
B	12	0	4
D	12	4	0

Tree in Newick format:

```
((A:1,C:3):4,(B:2,D:2):4);
```

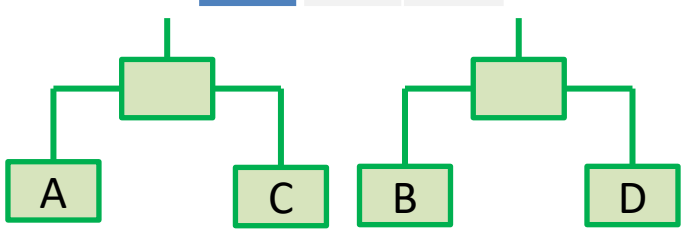


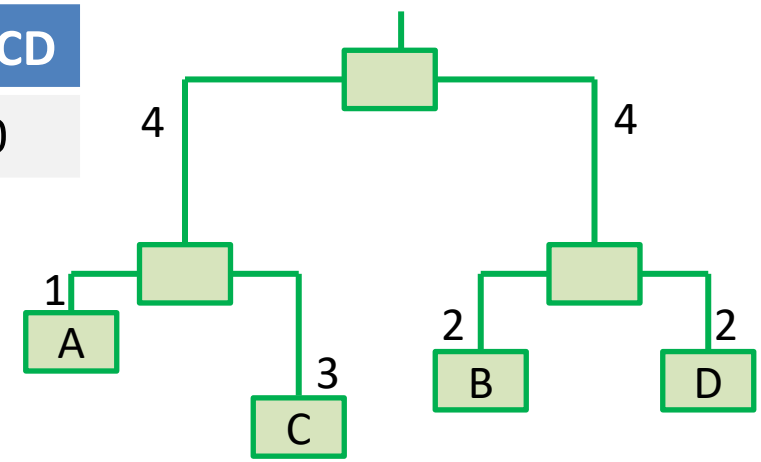
{B},{D}

	AC	BD
AC	0	12
BD	12	0

{AC},{BD}

	ABCD
ABCD	0

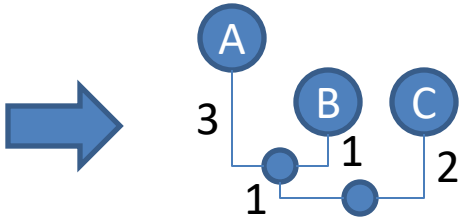




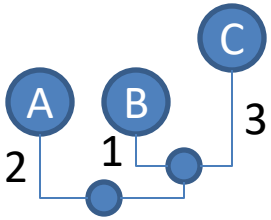
Introduction to Neighbour Joining

- Problems with UPGMA
  - Hard to assign branch length
  - Not always unique

	A	B	C
A	0	4	6
B	4	0	4
C	6	4	0



OR



- Idea of Neighbour Joining
  - Merge some species that are relatively close to each other and distant from the other species

Neighbor Joining

- Repeat the following steps until all branch lengths are assigned
  - Find two sets connected to hub with minimum *Q*, say set C_i and C_j (given r = the current number of clusters)

$$u(C_x) = \sum_y d(C_x, C_y) \qquad Q(i,j) = (r-2)d(C_i, C_j) - u(C_i) - u(C_j)$$

  - Insert an internal node C_k connecting C_i, C_j and the hub
  - Compute the distances with following equations:

$$d(C_i, C_k) = \frac{d(C_i, C_j)}{2} + \frac{u(C_i) - u(C_j)}{2(r-2)} \qquad d(C_j, C_k) = \frac{d(C_i, C_j)}{2} + \frac{u(C_j) - u(C_i)}{2(r-2)}$$

$$d(C_l, C_k) = \frac{d(C_i, C_l) + d(C_j, C_l) - d(C_i, C_j)}{2}$$

Exercise 4: Neighbor Joining

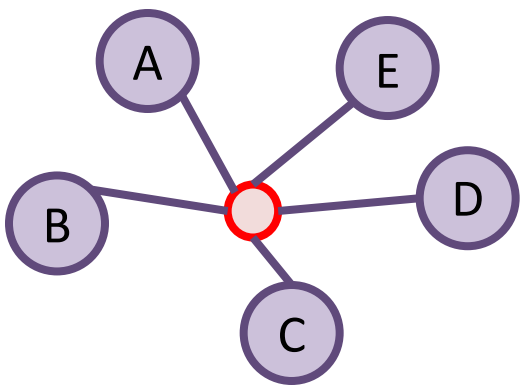
- Construct the phylogenetic tree with the following distance matrix using neighbor joining algorithm.

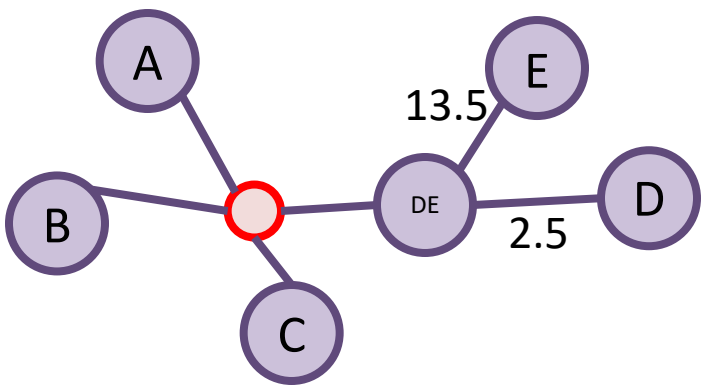
d	A	B	C	D	E
A	0	1	2	6	19
B	1	0	3	7	18
C	2	3	0	8	17
D	6	7	8	0	16
E	19	18	17	16	0

Answer: Neighbor Joining

d	A	B	C	D	E	u
A	0	1	2	6	19	28
B	1	0	3	7	18	29
C	2	3	0	8	17	30
D	6	7	8	0	16	37
E	19	18	17	16	0	70

Q	A	B	C	D	E
A		-54	-52	-47	-41
B			-50	-45	-45
C				-43	-49
D					-59
E					





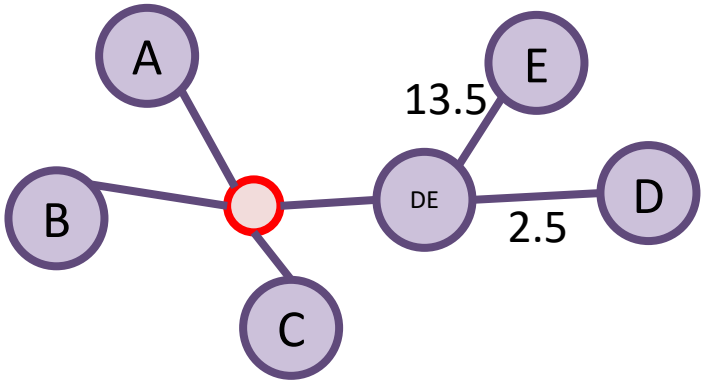
$$C_D C_{DE} = \frac{16}{2} + \frac{37-70}{2(5-2)} = 2.5,$$

$$C_{DE} C_E = \frac{16}{2} + \frac{70-37}{2(5-2)} = 13.5$$

Answer: Neighbor Joining

d	A	B	C	D	E	u
A	0	1	2	6	19	28
B	1	0	3	7	18	29
C	2	3	0	8	17	30
D	6	7	8	0	16	37
E	19	18	17	16	0	70

d	A	B	C	DE
A	0	1	2	4.5
B	1	0	3	4.5
C	2	3	0	4.5
DE	4.5	4.5	4.5	0



$$d(C_A, C_{DE})$$

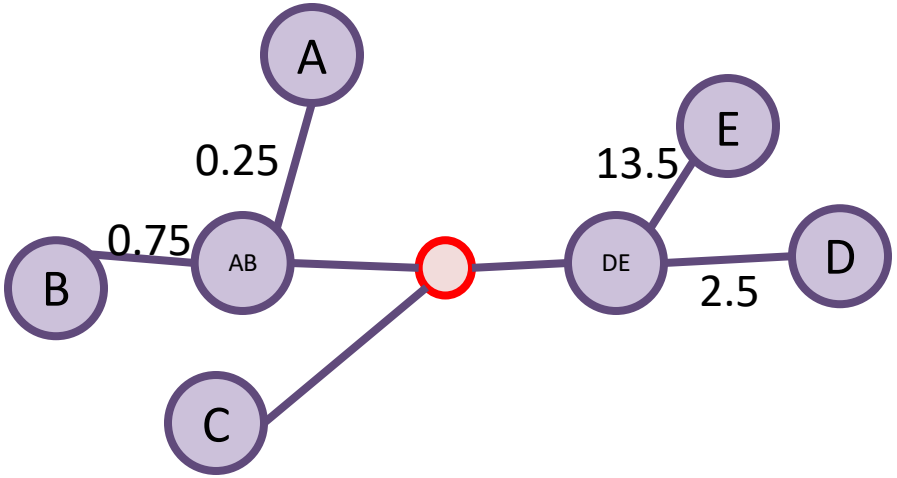
$$= \frac{d(C_D,C_A)+d(C_E,C_A)-d(C_D,C_E)}{2}$$

$$= \frac{6+19-16}{2} = 4.5$$

Answer: Neighbor Joining

d	A	B	C	DE	u
A	0	1	2	4.5	7.5
B	1	0	3	4.5	8.5
C	2	3	0	4.5	9.5
DE	4.5	4.5	4.5	0	13.5

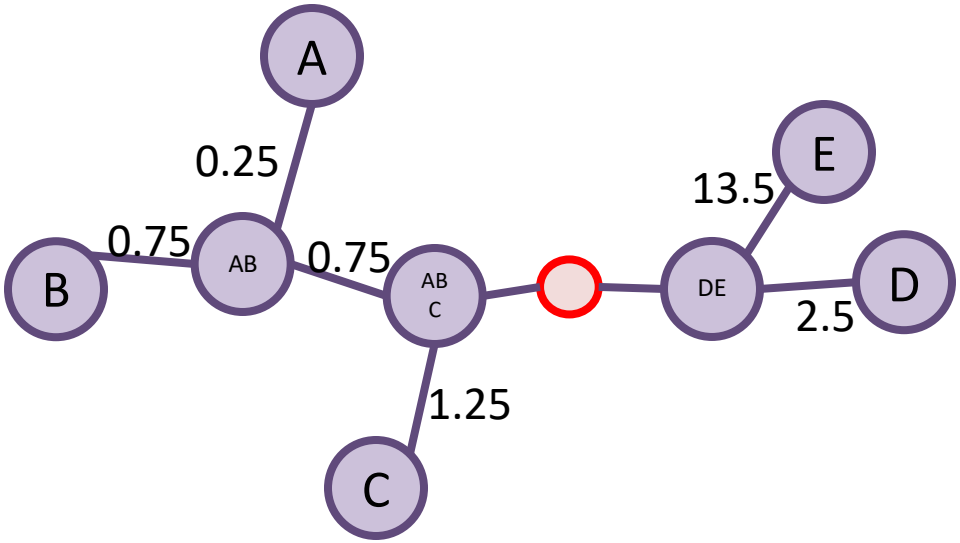
Q	A	B	C	DE
A		-14	-13	-12
B			-12	-13
C				-14
DE				



Answer: Neighbor Joining

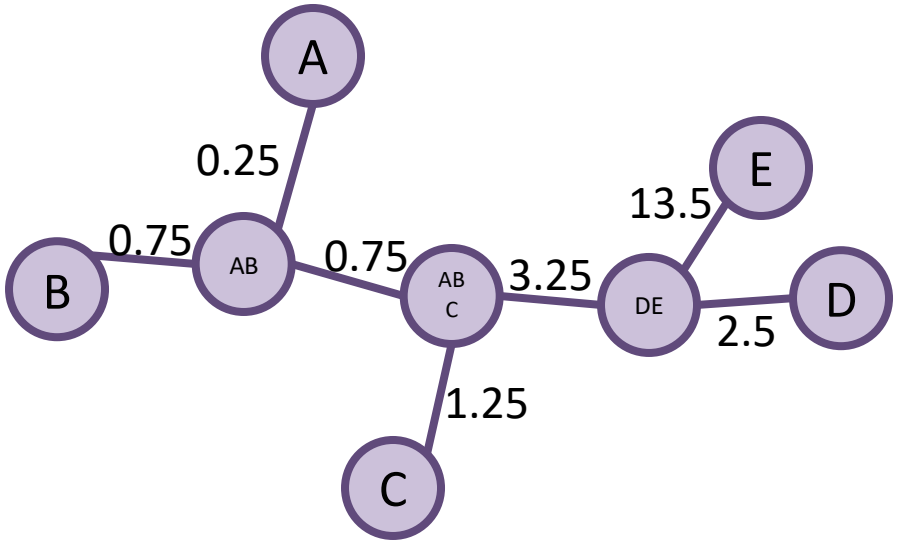
d	AB	C	DE	u
AB	0	2	4	6
C	2	0	4.5	6.5
DE	4	4.5	0	8.5

Q	AB	C	DE
AB		-10.5	-10.5
C			-10.5
DE			



Answer: Neighbor Joining

d	ABC	DE
ABC	0	3.25
DE	3.25	0



Demonstration: Neighbor Joining

- Find a web tool to perform Neighbor Joining using the distance matrix in the previous exercise.
- Compare the tree constructed manually in the previous exercise with the one produced by the web tool.

Step 1: find the tool

neighbor joining web tool - Goo

google.com/search?q=neighbor+joining+web+tool&oq=neighbor+joining+web+tool&aqs=chrom...

Google

neighbor joining web tool

All | Images | News | Videos | Shopping | More | Settings | Tools

About 50,700,000 results (0.60 seconds)

www.ebi.ac.uk › Tools › phylogeny › simple_phylogeny

Simple Phylogenetic Tree < Phylogeny < EMBL-EBI

Simple Phylogeny. Input form · **Web** services · Help & Documentation · Bioinformatics **Tools** FAQ · Feedback; Share ...

molbiol-tools.ca › Phylogeny

Phylogeny - Online Analysis Tools

T-REX includes several popular bioinformatics applications such as MUSCLE, MAFFT, **Neighbor Joining**, NINJA, BioNJ, PhyML, RAxML, random phylogenetic ...

en.wikipedia.org › wiki › List_of_phylogenetics_software

List of phylogenetics software - Wikipedia

Such **tools** are commonly used in comparative genomics, cladistics, and bioinformatics. Methods for estimating phylogenies include **neighbor-joining**, maximum parsimony (also simply referred to as parsimony), **UPGMA**, Bayesian phylogenetic inference, maximum likelihood and distance matrix methods.

www.phylogeny.fr

Phylogeny.fr: Home

Phylogeny.fr is a free, simple to use **web** service dedicated to reconstructing and analysing phylogenetic relationships between molecular ... Explore your sequence **neighbors** ... Direct access to the individual **tools** available on this server.

Step 2: select the tool

neighbor joining web to | Online Analysis Tools - Ph

Secure | https://molbiol-tools.ca/Phylogeny.htm

PHYLOGENY

- T-Rex (Tree and reticulogram REConstruction) - is dedicated to the reconstruction of phylogenetic trees, reticulation networks and to the inference of horizontal gene transfer (HGT) events. T-REX includes several popular bioinformatics applications such as MUSCLE, MAFFT, Neighbor Joining, NINJA, BioNJ, PhyML, RAxML, random phylogenetic tree generator and some well-known sequence-to-distance transformation models. It also comprises fast and effective methods for inferring phylogenetic trees from complete and incomplete distance matrices as well as for reconstructing reticulograms and HGT networks (Reference: Alix, C. et al. 2012. Nucl. Acids Res. **40** (W1): W573-W579).

- Phylogeny.fr - is a simple to use web service dedicated to reconstructing and analysing phylogenetic relationships between molecular sequences.It includes multiple alignment (MUSCLE, T-Coffee, ClustalW, ProbCons), phylogeny (PhyML, MrBayes, TNT, BioNJ), tree viewer (Drawgram, Drawtree, ATV) and utility programs (e.g. Gblocks to eliminate poorly aligned positions and divergent regions) (Reference: A. Dereeper et al,. 2008. Nucl. Acids Res. **36** (Web Server Issue):W465-9). Also available here.

- FastME provides distance algorithms to infer phylogenies. FastME is based on balanced minimum evolution, which is the very principle of NJ. FastME improves over NJ by performing topological moves using fast, sophisticated algorithms. The first version of FastME only included Nearest Neighbor Interchange (NNI). The new 2.0 version also includes Subtree Pruning and Regrafting (SPR), while remaining as fast as NJ and providing a number of facilities: distance estimation for DNA and proteins with various models and options, bootstrapping, and parallel computations. (Reference: Lefort V. et al. Molecular Biology & Evolution **32(10):** 2798-800, 2015).

- PhyML - has been widely used because of its simplicity and a fair compromise between accuracy and speed. In the meantime research on PhyML has continued, and new algorithms and methods have been implemented in the program. (Reference: V. Lefort et al. Molecular Biology and Evolution, msx149, 2017).

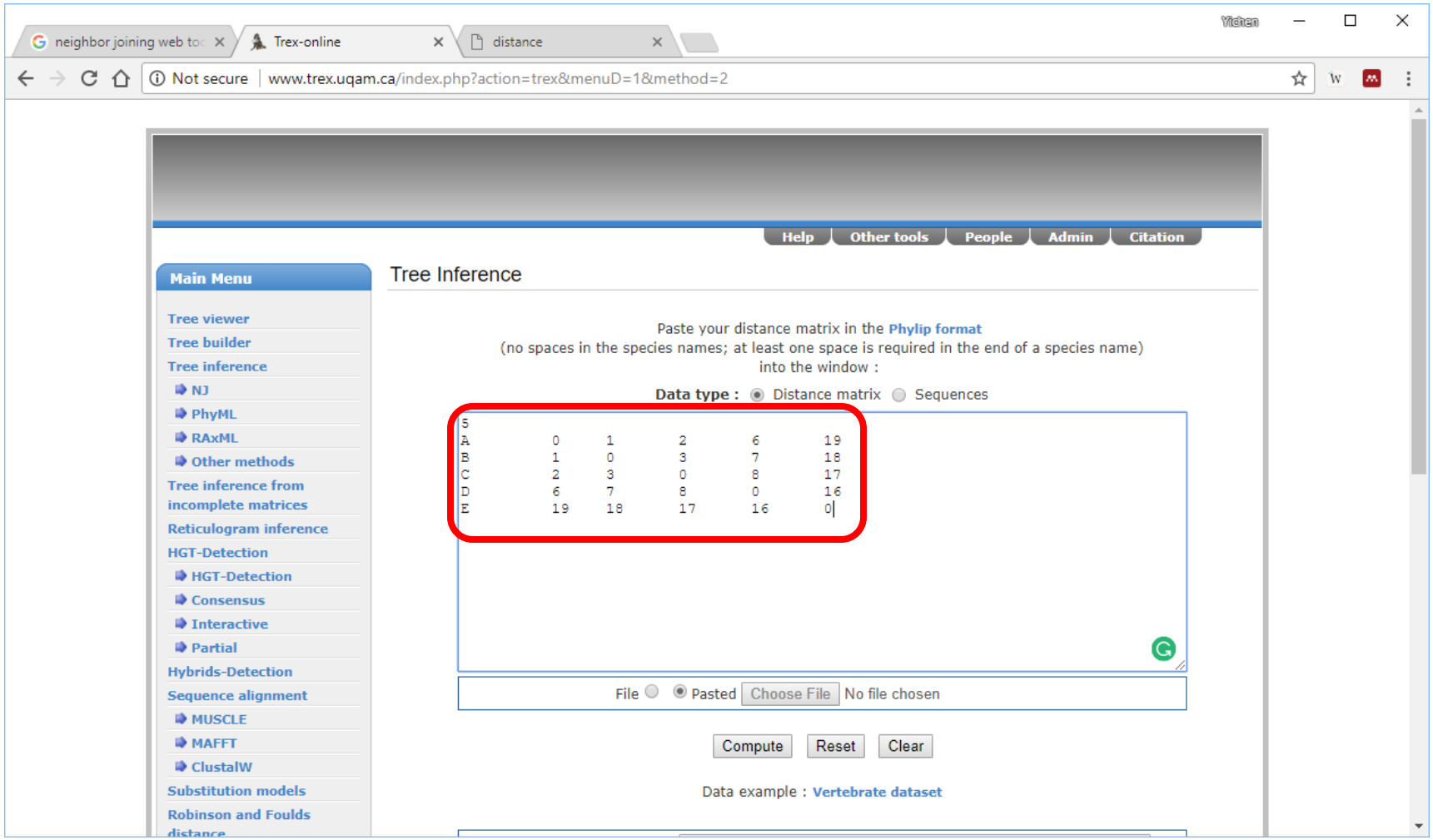
- RAxML (**R**andomized **Ax**elerated **M**aximum **L**ikelihood) is a program for sequential and parallel Maximum Likelihood based inference of large phylogenetic trees (Reference: Stamatakis, A. 2006. Bioinformatics 22:2688–2690).

- ProtTest *(David Posada, University of Vigo, Spain)* - estimates the empirical model of aminoacid substitution that fits the data best among 64 candidate models. PROTTEST calculates AIC, AICc and BIC values, and obtain a rank of model fits, model-averaged parameter estimates, or measures of parameter importance. Mac OSX, Windows and Linux versions are available for downloading.

- Phylemon2 - a suite of web-tools for molecular evolution, phylogenetics and phylogenomics (Reference:Sánchez, R. et al. 2011.Nucl. Acids Res. 39/suppl_2/W470)

- POWER (**P**hyl**O**genetic **We**b **R**epeater) - allows users to carry out phylogenetic analysis on most programs of PHYLIP package repeatedly. POWER provide two pipelines to process the analysis. One of them includes multiple sequence alignment (MSA) at the beginning of the pipeline whereas the other begin phylogenetic analysis with aligned sequence. Very user friendly. (Reference: C. Y. Lin, et al. 2005. Nucl. Acids Res. **33:**

Step 3: input the distance matrix



The Phylip format

neighbor joining web to | Trex-online | distance

evolution.genetics.washington.edu/phylip/doc/distance.html

The input format for distance data is straightforward. The first line of the input file contains the number of species. There follows species data, starting, as with all other programs, with a species name. The species name is ten characters long, and must be padded out with blanks if shorter. For each species there then follows a set of distances to all the other species (options selected in the programs' menus allow the distance matrix to be upper or lower triangular or square). The distances can continue to a new line after any of them. If the matrix is lower-triangular, the diagonal entries (the distances from a species to itself) will not be read by the programs. If they are included anyway, they will be ignored by the programs, except for the case where one of them starts a new line, in which case the program will mistake it for a species name and get very confused.

For example, here is a sample input matrix, with a square matrix:

```
     5
Alpha      0.000 1.000 2.000 3.000 3.000
Beta       1.000 0.000 2.000 3.000 3.000
Gamma      2.000 2.000 0.000 3.000 3.000
Delta      3.000 3.000 3.000 0.000 1.000
Epsilon    3.000 3.000 3.000 1.000 0.000
```

and here is a sample lower-triangular input matrix with distances continuing to new lines as needed:

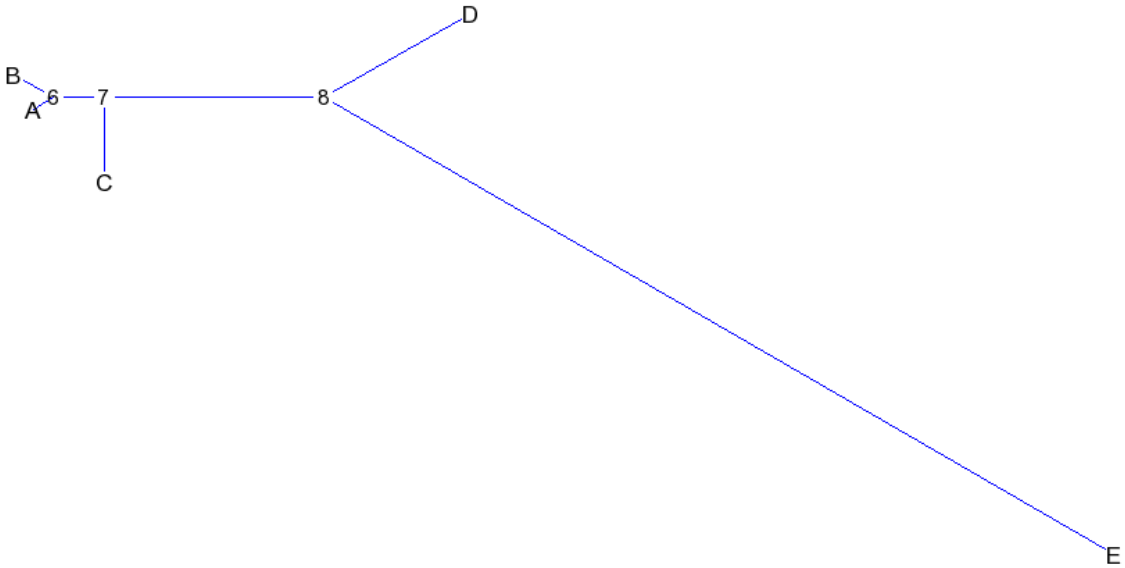
```
   14
Mouse
Bovine      1.7043
Lemur       2.0235  1.1901
Tarsier     2.1378  1.3287  1.2905
Squir Monk  1.5232  1.2423  1.3199  1.7878
Jpn Macaq   1.8261  1.2508  1.3887  1.3137  1.0642
Rhesus Mac  1.9182  1.2536  1.4658  1.3788  1.1124  0.1022
Crab-E.Mac  2.0039  1.3066  1.4826  1.3826  0.9832  0.2061  0.2681
BarbMacaq   1.9431  1.2827  1.4502  1.4543  1.0629  0.3895  0.3930  0.3665
Gibbon      1.9663  1.3296  1.8708  1.6683  0.9228  0.8035  0.7109  0.8132
  0.7858
Orang       2.0593  1.2005  1.5356  1.6606  1.0681  0.7239  0.7290  0.7894
  0.7140  0.7095
Gorilla     1.6664  1.3460  1.4577  1.5935  0.9127  0.7278  0.7412  0.8763
  0.7966  0.5959  0.4604
Chimp       1.7320  1.3757  1.7803  1.7119  1.0635  0.7899  0.8742  0.8868
  0.8288  0.6213  0.5065  0.3502
Human       1.7101  1.3956  1.6661  1.7599  1.0557  0.6933  0.7118  0.7589
  0.8542  0.5612  0.4700  0.3097  0.2712
```

5					
A	0	1	2	6	19
B	1	0	3	7	18
C	2	3	0	8	17
D	6	7	8	0	16
E	19	18	17	16	0

Note that the name "Mouse" in this matrix must be padded out by blanks to the full length of 10 characters.

Step 4: show the results

- Tree in the Newick format:
  - (D:2.5000,E:13.5000,(C:1.2500,(A:0.3333,B:0.6667):0.7500):3.2500);
  - Note: the above Newick format is for the tree rooted at node “8”.



Comparison

- The tree produced by the web tool is almost the same as the one constructed manually.

- There are some possible reasons if the results are different:
  - Different ways to break ties
  - Different formulas used in the web tool

Maximum parsimony

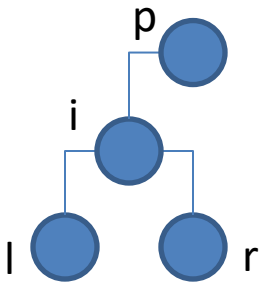
- Rational:
  - Mutations are rare
  - A tree is likely to be true if it involves few mutations
- "Small parsimony" problem:
  - Sequences at the leaf nodes and the rooted tree structure are given
  - Find the <u>ancestral sequences</u>
  - Note: if the tree structure is not known, the problem is NP hard

Maximum parsimony

	Large parsimony	Small parsimony
Observed sequences	Given	Given
Ancestral sequences	Need to work out	Need to work out
Tree topology	Need to work out	Given
Number of mutations	Minimum among all tree topologies	Minimum subject to the given tree topology
Algorithms	No efficient algorithms known (NP hard)	Simple version, extended version

Small parsimony (simple version)

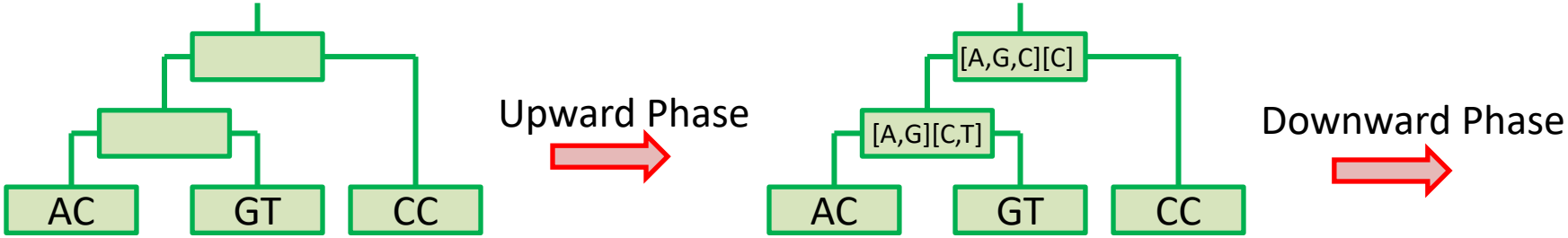
- Upward Phase:
  - If $S_{left\text{-}child} \cap S_{right\text{-}child} = \{\ \}$,
    then $S_{current} \leftarrow S_{left\text{-}child} \cup S_{right\text{-}child}$
  - If $S_{left\text{-}child} \cap S_{right\text{-}child} \neq \{\ \}$,
    then $S_{current} \leftarrow S_{left\text{-}child} \cap S_{right\text{-}child}$

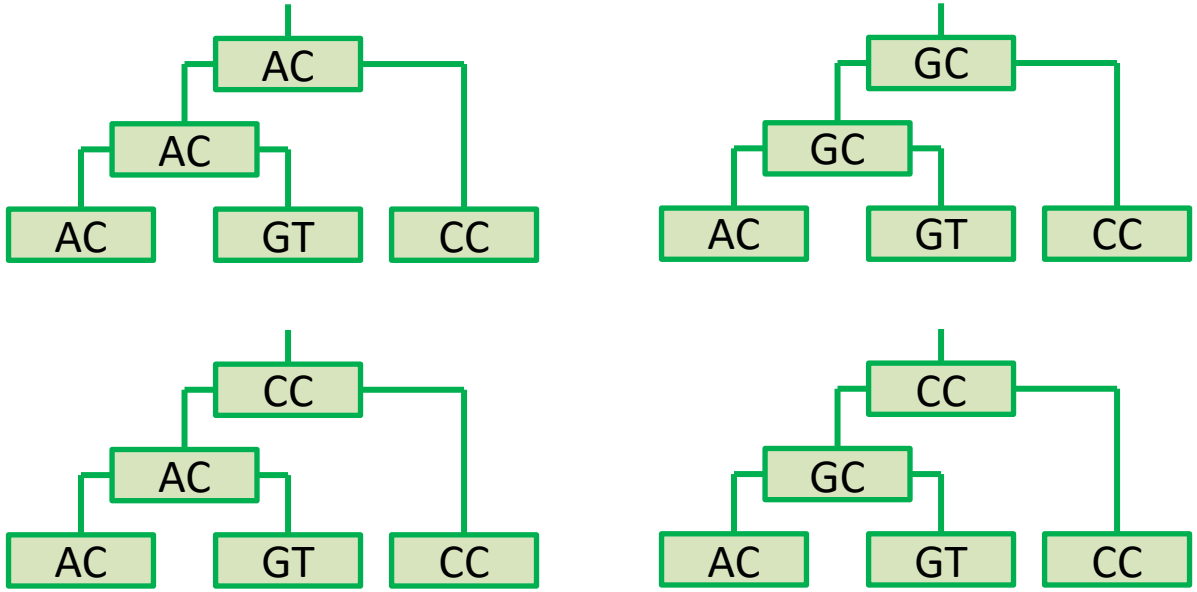


- Downward Phase:

  $C_{root} \leftarrow$ any one in S_{root}
  - If $C_{parent} \in S_{current}$ then $C_{current} \leftarrow C_{parent}$
  - If $C_{parent} \notin S_{current}$ then $C_{current} \leftarrow$ any one in $S_{current}$

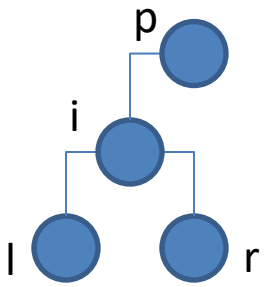
Example: small parsimony (simple version)





Small parsimony (extended version)

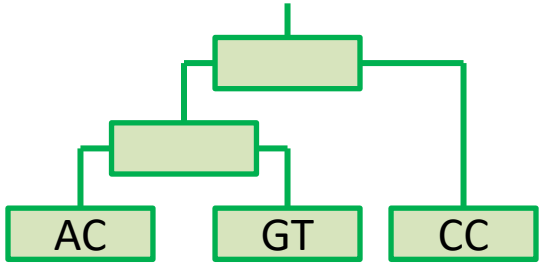
- Upward Phase (same as before):
  - If $S_{left\text{-}child} \cap S_{right\text{-}child} = \{\ \}$,
    then $S_{current} \leftarrow S_{left\text{-}child} \cup S_{right\text{-}child}$
  - If $S_{left\text{-}child} \cap S_{right\text{-}child} \neq \{\ \}$,
    then $S_{current} \leftarrow S_{left\text{-}child} \cap S_{right\text{-}child}$



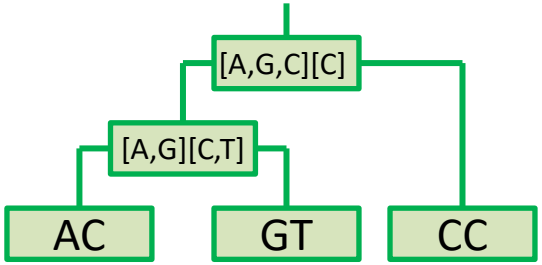
- Downward Phase (majority vote):

  In turn, $C_{root} \leftarrow$ one in S_{root}
  - Compare the characters in $\{C_{parent}\}$, S_{child1}, and S_{child2} and find those having the most appearances
  - In turn, choose one to be $C_{current}$

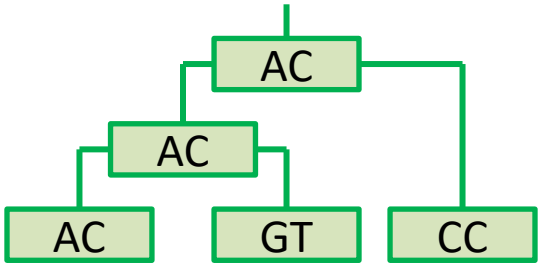
Example: small parsimony (extended version)

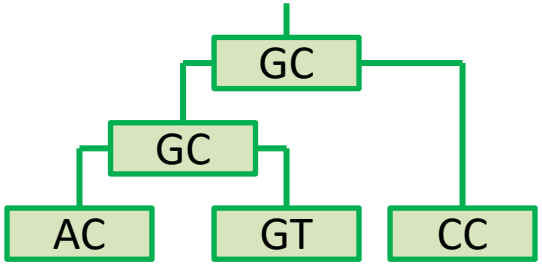


Upward Phase

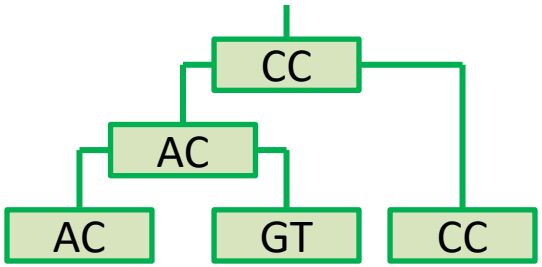


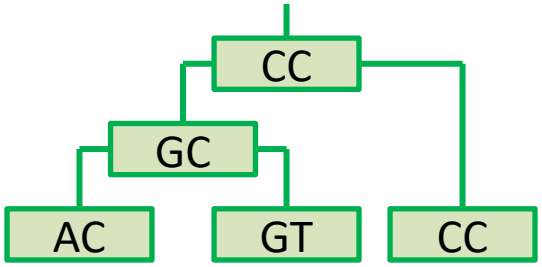
Downward Phase

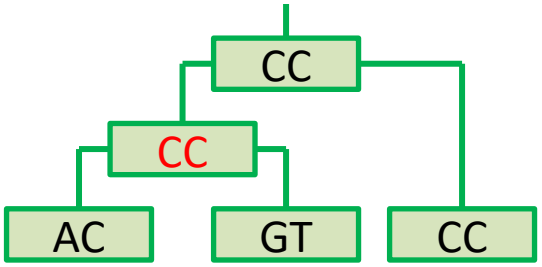




Found in extended version but not in simple version

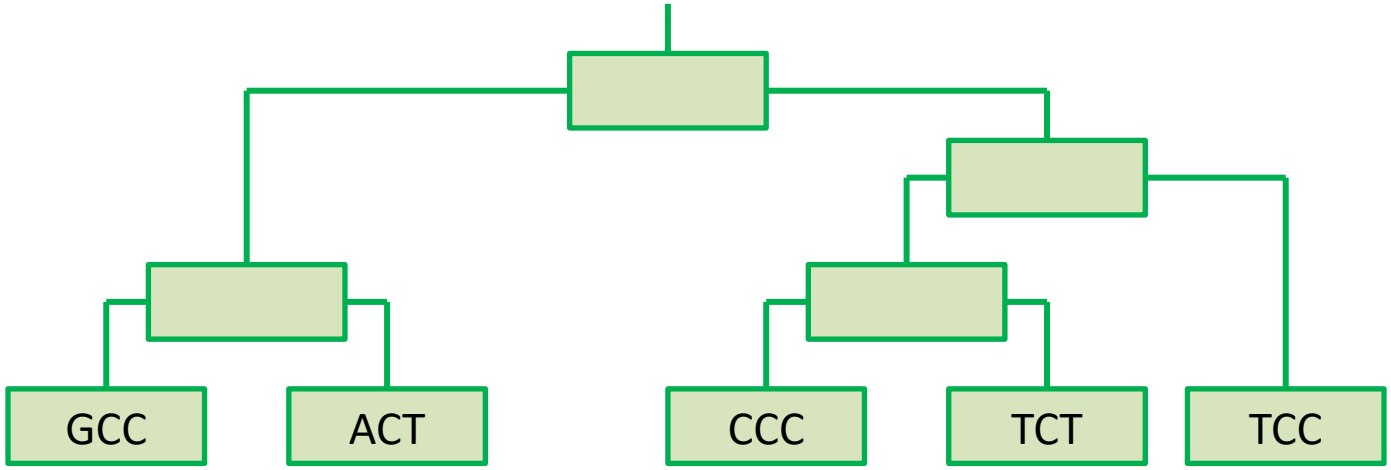






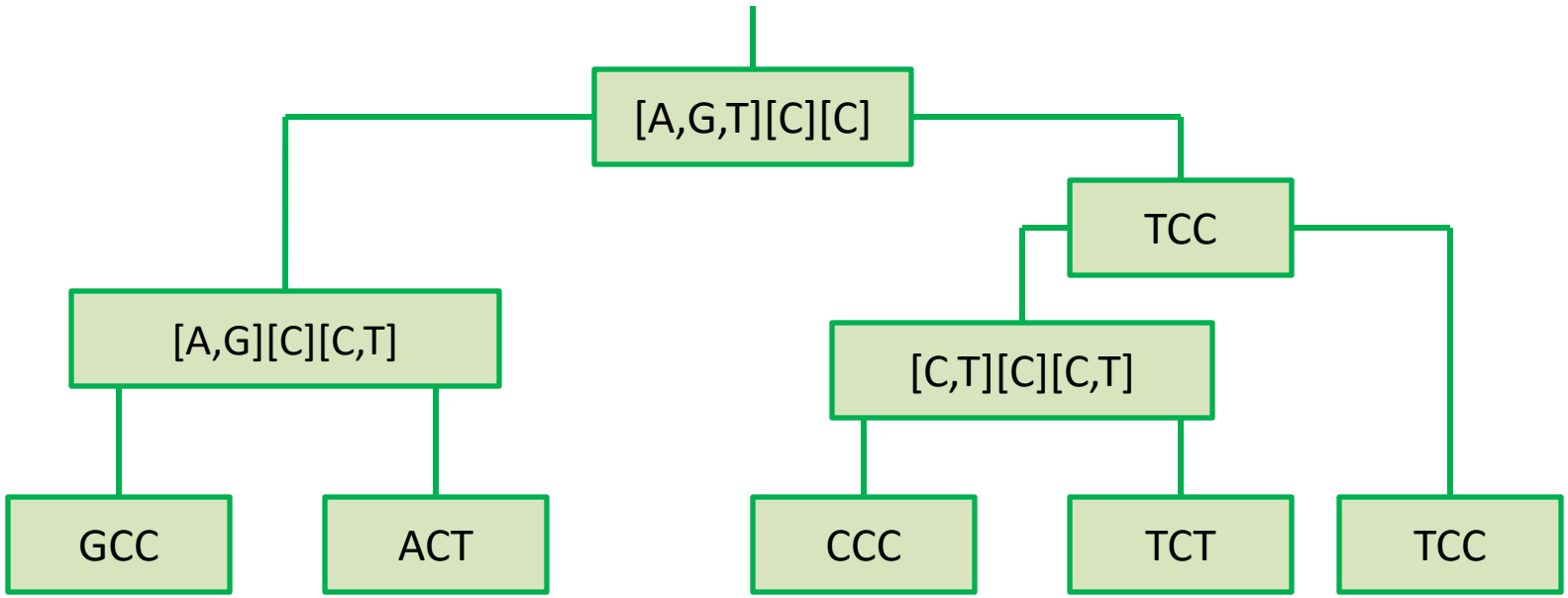
Exercise 1: Small parsimony

- You are given the following tree topology. Considering each site independently, find:
  - the ancestral sequences such that the number of mutations is minimum, using both **simple** and **extended** versions;
  - the optimal number of mutations.



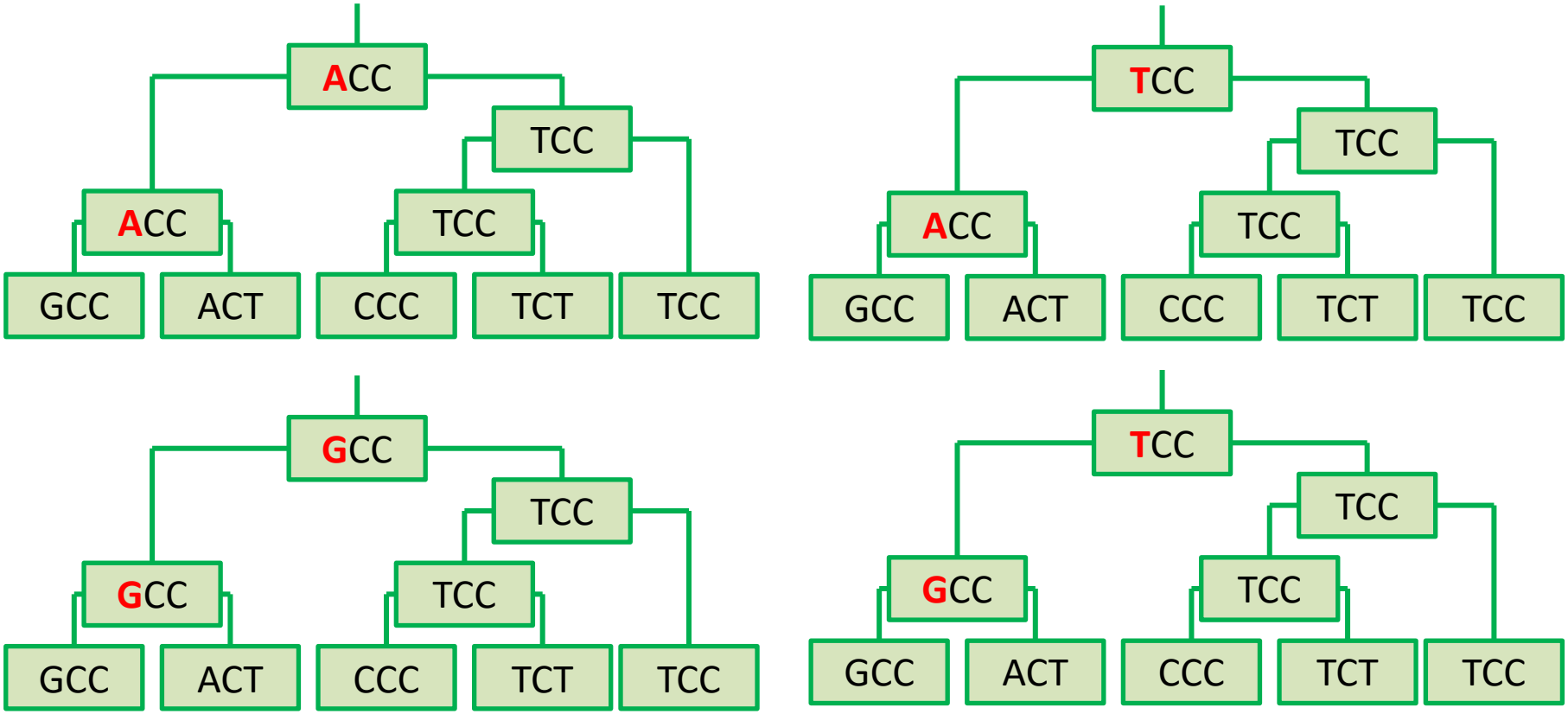
Answer: Small parsimony

- Upward phase



Answer: Small parsimony (simple version)

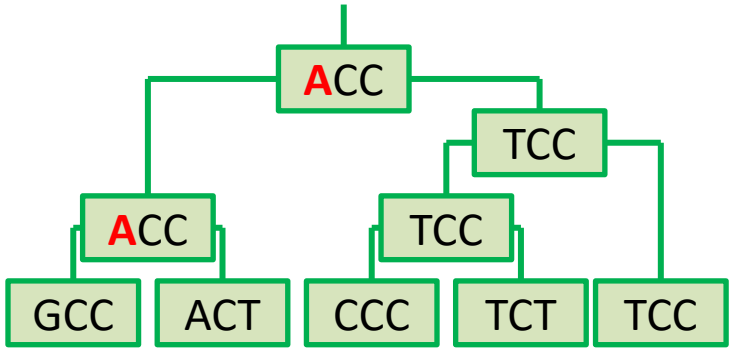
- Downward phase

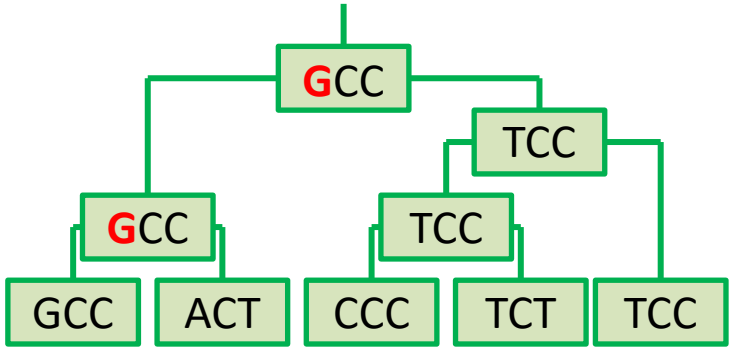


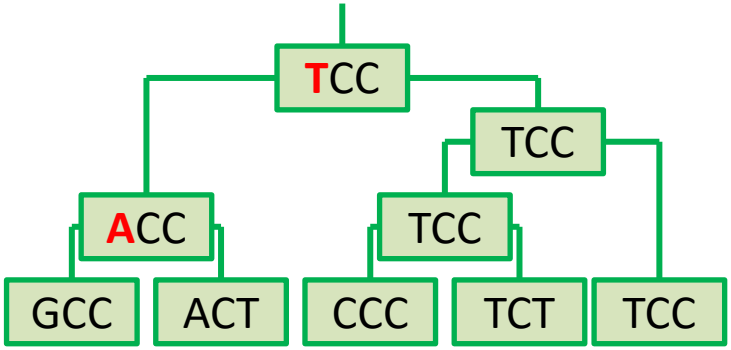
Number of mutations: 5

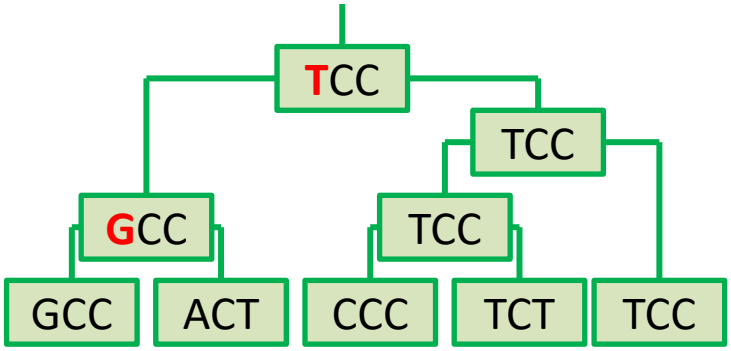
Answer: Small parsimony (extended version)

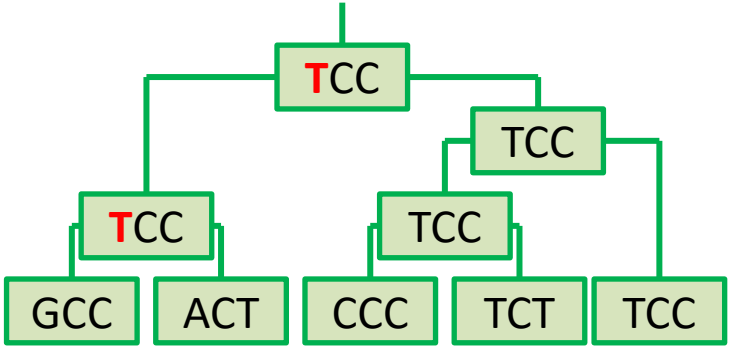
- Downward phase











Number of mutations: 5

Maximum likelihood

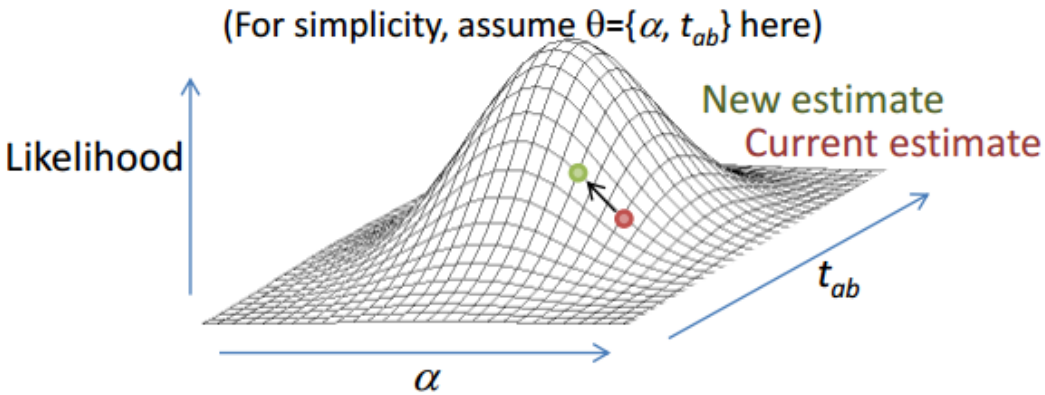
- Maximum likelihood: find parameters (θ) of a model such that the probability of having the observed data under this model, $\Pr(X|\theta)$, is maximized.
- Small likelihood problem:
  - Tree topology is given
  - Find <u>the mutation rates and divergence times</u>
  - There are effective heuristic methods

Maximum likelihood

	Large likelihood	Small likelihood
Observed sequences	Given	Given
Ancestral sequences	Consider all possible cases	Consider all possible cases
Time of divergence	Need to work out	Need to work out
Mutation rate	Need to work out	Need to work out
Tree topology	Need to work out	Given
Likelihood	Maximum among all tree topologies	Maximum subject to the given tree topology
Algorithms	No efficient algorithms known (NP hard)	Effective heuristic methods

Small likelihood

- We try different time of divergence and mutation rates, and change them a little bit each iteration
- After many trials, the likelihood will converge to some local (possibly global) maximum



- Key points:
  - Evaluate likelihood given time of divergence and mutation rate using dynamic programming
  - Compare the value of likelihood

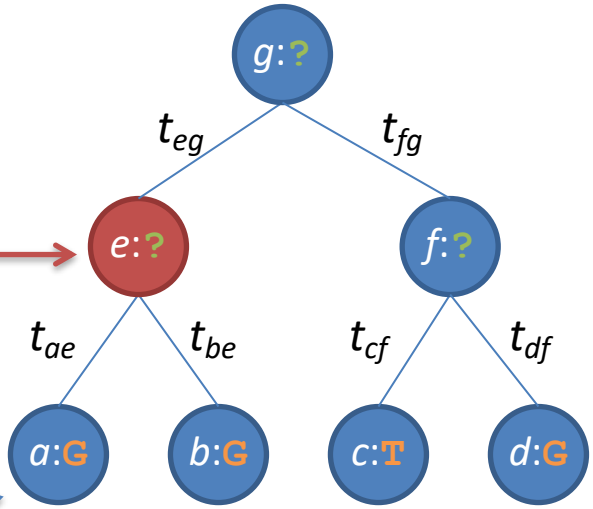
Image credit: http://www.absoluteastronomy.com/topics/Hill_climbing

Computing likelihood [Optional]

- Define table V, where entry V(i,x) is the likelihood of the sub-tree rooted at i when **the parent of i** takes character x
- Example:
  - Likelihood =
    Pr(g:A) V(e,A) V(f,A) +
    Pr(g:C) V(e,C) V(f,C) +
    Pr(g:G) V(e,G) V(f,G) +
    Pr(g:T) V(e,T) V(f,T)
  - V(***e***,**A**) =
    Pr(***e***:A|***g***:**A**,t_{eg}) V(a,A) V(b,A) +
    Pr(***e***:C|***g***:**A**,t_{eg}) V(a,C) V(b,C) +
    Pr(***e***:G|***g***:**A**,t_{eg}) V(a,G) V(b,G) +
    Pr(***e***:T|***g***:**A**,t_{eg}) V(a,T) V(b,T)
  - V(***a***,**A**) = Pr(***a***:G|***e***:**A**, t_{ae})

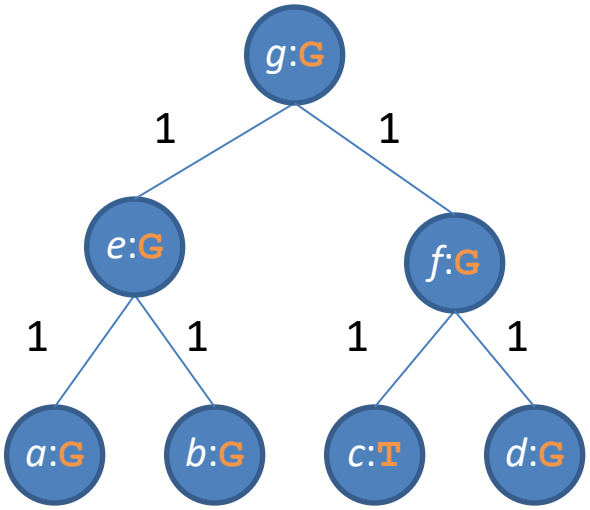
For internal nodes

For leaf nodes



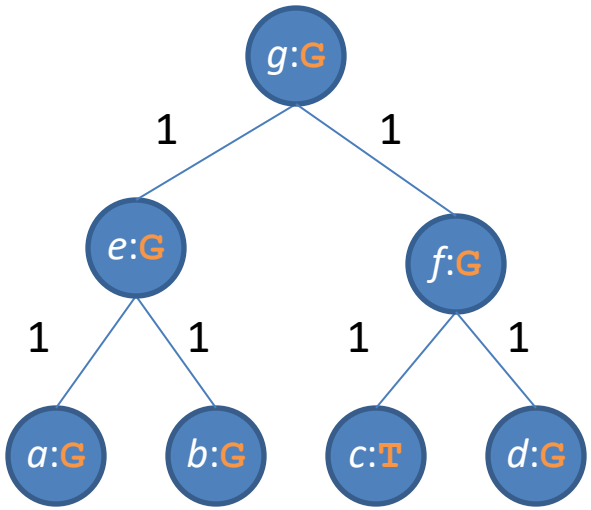
Exercise: Computing Likelihood [Optional]

- Using Jukes-Cantor model with mutation rate = 0.01, divergence time are drawn on the branches of the tree. Assume the probability of having A, C, G and T are equal (0.25) for the sequence *g*. Calculate the likelihood of the tree.



Answer: Computing Likelihood [Optional]

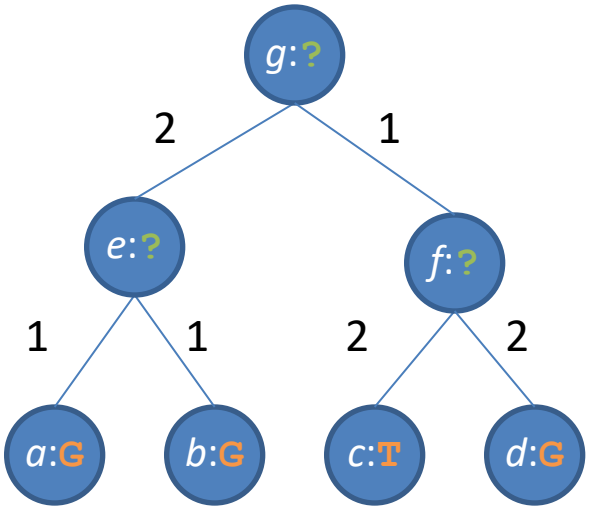
Node	Likelihood
a	$v_a = 0.97$
b	$v_b = 0.97$
c	$v_c = 0.01$
d	$v_d = 0.97$
e	$v_e = (0.97)v_a\, v_b = (0.97)(0.97)(0.97) = 0.912673$
f	$v_f = (0.97)v_c\, v_d = (0.97)(0.01)(0.97) = 0.009409$
g	$v_g = (0.25)v_e\, v_f = (0.25)(0.912673)(0.009409)$ $= 0.00214683506$



A Particular Case	Probability
1 mutation	0.01
1 no change	0.97

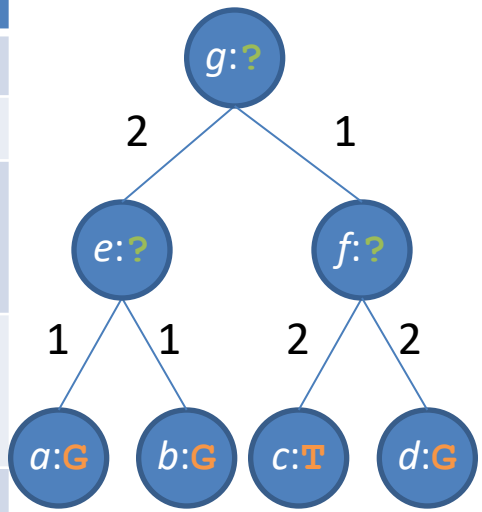
Exercise: Likelihood [Optional]

- Given that mutation rate = 0.1, divergence time are drawn on the branches of the tree. Assume the probability of having A, C, G, or T are equal for the sequence *g*, using Jukes-Cantor model, find the likelihood of this tree.



Exercise: Likelihood [Optional]

X	A	C	G	T
V(a,X)	0.1	0.1	0.7	0.1
V(b,X)	0.1	0.1	0.7	0.1
V(c,X)	2(0.1)(0.7) +2(0.1)(0.1) = 0.16	2(0.1)(0.7) +2(0.1)(0.1) = 0.16	2(0.1)(0.7) +2(0.1)(0.1) = 0.16	(0.7)(0.7) +3(0.1)(0.1) = 0.52
V(d,X)	2(0.1)(0.7) +2(0.1)(0.1) = 0.16	2(0.1)(0.7) +2(0.1)(0.1) = 0.16	(0.7)(0.7) +3(0.1)(0.7) = 0.52	2(0.1)(0.7) +2(0.1)(0.1) = 0.16
V(e,X)	(0.52)(0.1)(0.1) +(0.16)(0.1)(0.1) +(0.16)(0.7)(0.7) +(0.16)(0.1)(0.1) = 0.0868	(0.16)(0.1)(0.1) +(0.52)(0.1)(0.1) +(0.16)(0.7)(0.7) +(0.16)(0.1)(0.1) = 0.0868	(0.16)(0.1)(0.1) +(0.16)(0.1)(0.1) +(0.52)(0.7)(0.7) +(0.16)(0.1)(0.1) = 0.2596	(0.16)(0.1)(0.1) +(0.16)(0.1)(0.1) +(0.16)(0.7)(0.7) +(0.52)(0.1)(0.1) = 0.0868
V(f,X)	(0.7)(0.16)(0.16) +(0.1)(0.16)(0.16) +(0.1)(0.16)(0.52) +(0.1)(0.52)(0.16) = 0.0349696	(0.1)(0.16)(0.16) +(0.7)(0.16)(0.16) +(0.1)(0.16)(0.52) +(0.1)(0.52)(0.16) = 0.0349696	(0.1)(0.16)(0.16) +(0.1)(0.16)(0.16) +(0.7)(0.16)(0.52) +(0.1)(0.52)(0.16) = 0.07168	(0.1)(0.16)(0.16) +(0.1)(0.16)(0.16) +(0.1)(0.16)(0.52) +(0.7)(0.52)(0.16) = 0.07168

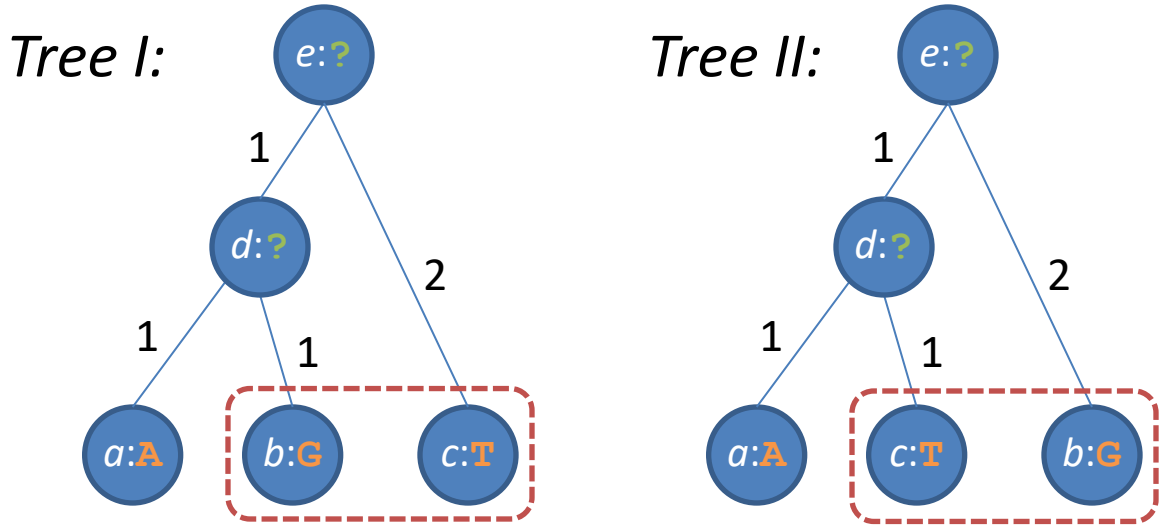


A Particular Case	Probability
1 mutation	0.1
1 no change	0.7
1 mutation + 1 no change	(0.1)(0.7)
2 mutations	(0.1)(0.1)
2 no changes	(0.7)(0.7)

Likelihood = (0.25)(0.0868)(0.0349696) +(0.25)(0.0868)(0.0349696)
+(0.25)(0.2596)(0.07168) +(0.25)(0.0868)(0.07168) = 0.00772516864

Example: Comparison of Likelihood [Optional]

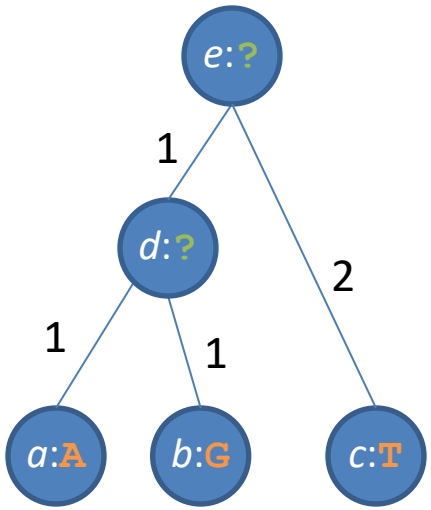
- Given that transition rate = 0.1, transversion rate = 0.01, divergence time are drawn on the branches of the tree. Assume the probability of having A, T, C, and G are 0.2, 0.2, 0.3 and 0.3 respectively for the sequence *e*. Using Kimura two-parameter model, determine which tree is more likely.



Example: Comparison of Likelihood [Optional]

X	A	C	G	T
V(*a*,*X*)	0.88	0.01	0.1	0.01
V(*b*,*X*)	0.1	0.01	0.88	0.01
V(*c*,*X*)	2(0.01)(0.88) +2(0.1)(0.01) = 0.0196	2(0.1)(0.88) +2(0.01)(0.01) = 0.1762	2(0.01)(0.88) +2(0.1)(0.01) = 0.0196	(0.88)(0.88) +2(0.01)(0.01) +(0.1)(0.1) = 0.7846
V(*d*,*X*)	(0.88)(0.88)(0.1) +(0.01)(0.01)(0.01) +(0.1)(0.1)(0.88) +(0.01)(0.01)(0.01) = 0.086242	(0.01)(0.88)(0.1) +(0.88)(0.01)(0.01) +(0.01)(0.1)(0.88) +(0.1)(0.01)(0.01) = 0.001858	(0.1)(0.88)(0.1) +(0.01)(0.01)(0.01) +(0.88)(0.1)(0.88) +(0.01)(0.01)(0.01) = 0.086242	(0.01)(0.88)(0.1) +(0.1)(0.01)(0.01) +(0.01)(0.1)(0.88) +(0.88)(0.01)(0.01) = 0.001858

Tree I:



Likelihood = (0.2)(0.086242)(0.0196)
+(0.3)(0.001858)(0.1762)
+(0.3)(0.086242)(0.0196)
+(0.2)(0.001858)(0.7846) = 0.00123494284

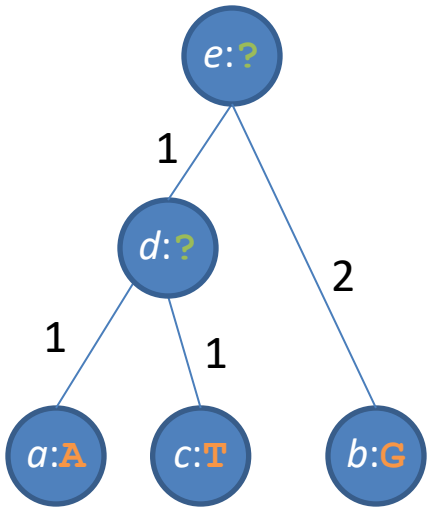
A Particular Case	Probability
1 transition (A<->G, C<->T)	0.1
1 transversion	0.01
1 no change	0.88

A Particular Case	Probability
1 transition + 1 transversion	(0.1)(0.01)
1 transition + 1 no change	(0.1)(0.88)
1 transversion + 1 no change	(0.01)(0.88)
2 transitions	(0.1)(0.1)
2 transversions	(0.01)(0.01)
2 no changes	(0.88)(0.88)

Example: Comparison of Likelihood [Optional]

X	A	C	G	T
V(a,X)	0.88	0.01	0.1	0.01
V(c,X)	0.01	0.1	0.01	0.88
V(b,X)	2(0.1)(0.88) +2(0.01)(0.01) = 0.1762	2(0.01)(0.88) +2(0.1)(0.01) = 0.0196	(0.88)(0.88) +2(0.01)(0.01) +(0.1)(0.1) = 0.7846	2(0.01)(0.88) +2(0.1)(0.01) = 0.0196
V(d,X)	(0.88)(0.88)(0.01) +(0.01)(0.01)(0.1) +(0.1)(0.1)(0.01) +(0.01)(0.01)(0.88) = 0.007942	(0.01)(0.88)(0.01) +(0.88)(0.01)(0.1) +(0.01)(0.1)(0.01) +(0.1)(0.01)(0.88) = 0.001858	(0.1)(0.88)(0.01) +(0.01)(0.01)(0.1) +(0.88)(0.1)(0.01) +(0.01)(0.01)(0.88) = 0.001858	(0.01)(0.88)(0.01) +(0.1)(0.01)(0.1) +(0.01)(0.1)(0.01) +(0.88)(0.01)(0.88) = 0.007942

Tree II:



Likelihood = (0.2)(0.007942)(0.1762)
+(0.3)(0.001858)(0.0196)
+(0.3)(0.001858)(0.7846)
+(0.2)(0.007942)(0.0196) = 0.0007592698

Previously, Likelihood of Tree I = 0.00123494284

So, Tree I is more likely than Tree II.

A Particular Case	Probability
1 transition + 1 transversion	(0.1)(0.01)
1 transition + 1 no change	(0.1)(0.88)
1 transversion + 1 no change	(0.01)(0.88)
2 transitions	(0.1)(0.1)
2 transversions	(0.01)(0.01)
2 no changes	(0.88)(0.88)

Check list

- What are the beliefs in maximum parsimony?
- What are the differences between the simple version and extended version of parsimony algorithm?
- What are the similarities and differences among the four phylogenetic tree reconstruction algorithms (i.e., maximum parsimony, maximum likelihood, UPGMA, Neighbor-joining)?