Sequence analysis and genomics

5. Introduction to Phylogenetics

Dr. Katja Nowick

Group leader TFome and Transcriptome Evolution
Bioinformatics group
Paul-Flechsig-Institute for Brain Research
www. nowicklab.info
nowick@bioinf.uni-leipzig.de

4. Add on to HMMs

Alice and Bob

Alice and Bob talk together daily over the telephone about what they did that day. Bob is only interested in three activities: walking in the park, shopping, and cleaning his apartment. What he does depend on the weather:

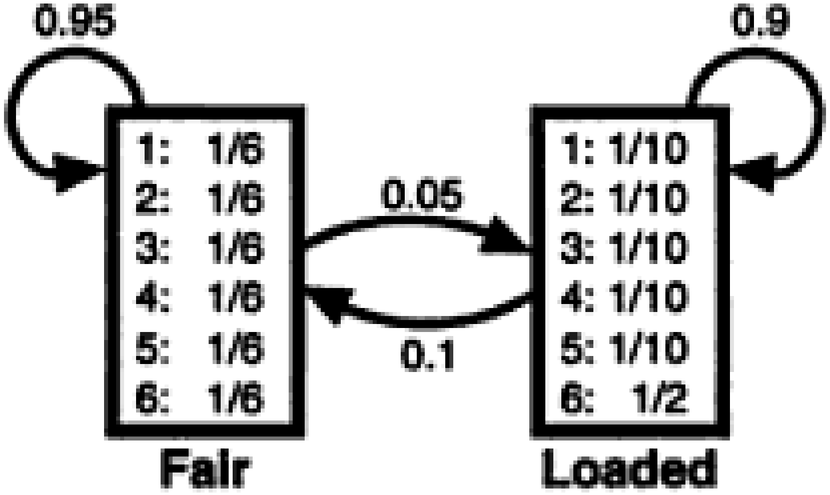
```
emission_probability = { 'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
                         'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1}
                       }
```

Alice has no definite information (information is *hidden)* about the weather where Bob lives, but she knows general trends.:

```
transition_probability = { 'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
                           'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6}
                         }
```

Based on what Bob tells her he did each day, Alice tries to guess what the weather must have been like.

The occasional dishonest casino



States: Fair, Loaded
Transition probabilities: 0.05, 0.1
Emission probabilities:
in fair state all have 1/6
in loaded state: 6 has 50%

Rolls: 31511624644664424531143631656626566666511664531326512456366646316 …
Die: FFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFFFFLLLLLLLLLLLL …

Would be useful to know when the loaded dice is in play to bet for the 6

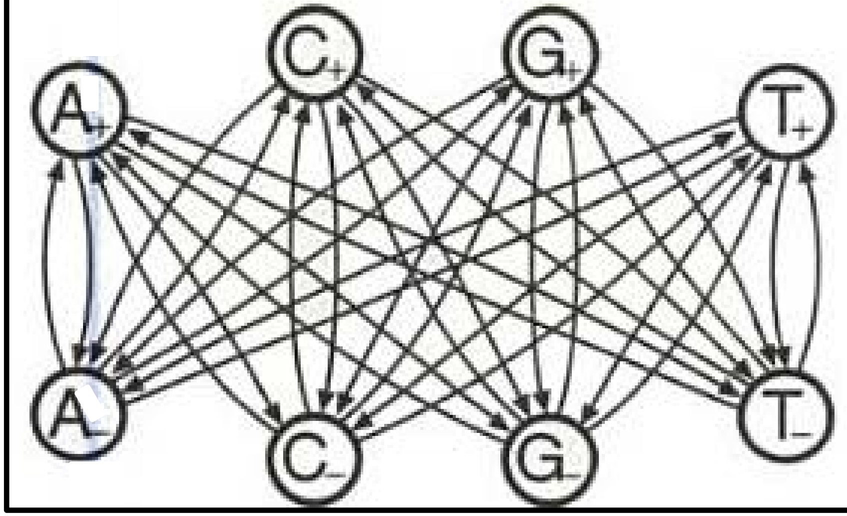
CpG islands

Hall mark of promoter regions, rich in CG dinucleotides

States: CpG island +
non-CpG island -

Transition probabilities: higher to go from CpG island to non-CpG island

Emission probabilities: how often a certain nucleotide is followed by another one
in CpG island C often followed by G



+ all transitions within each state (not shown)

Based on how often we observe CG dinucleotides, we can guess if we are in a CpG island

Alice and Bob

Alice and Bob talk together daily over the telephone about what they did that day. Bob is only interested in three activities: walking in the park, shopping, and cleaning his apartment. What he does depend on the weather:

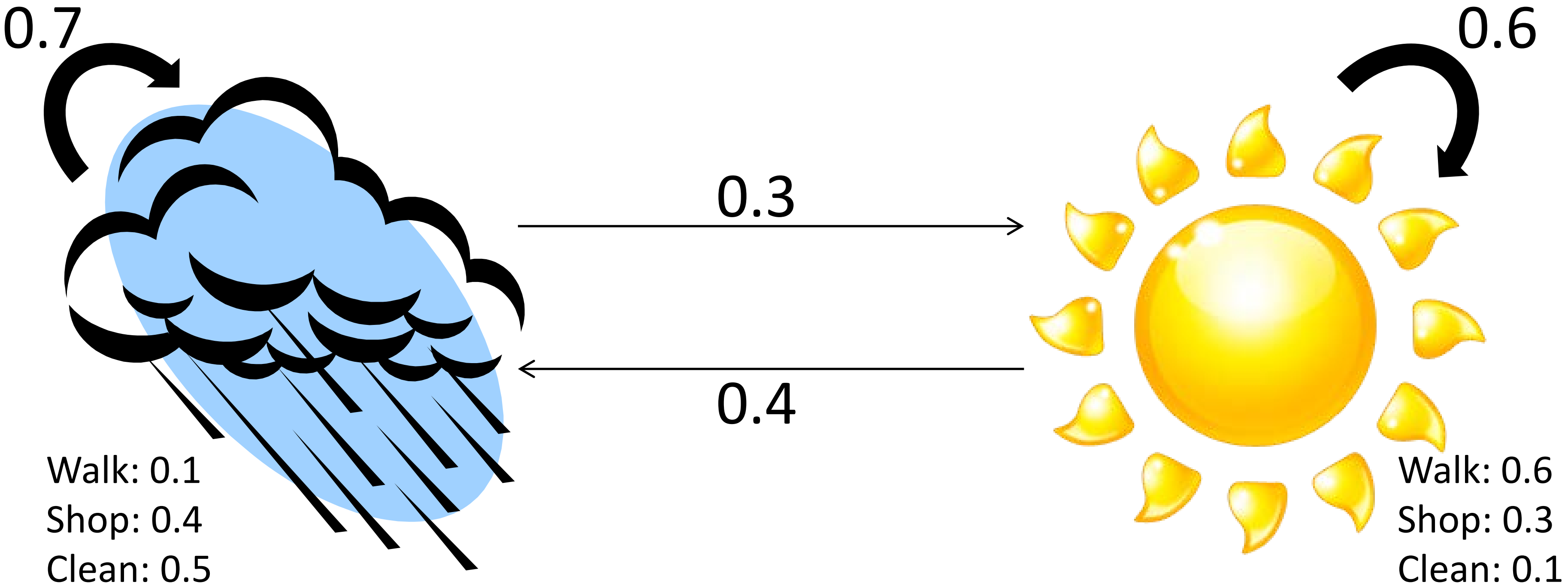
```
emission_probability = { 'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
                         'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1}
                       }
```

Alice has no definite information (information is *hidden)* about the weather where Bob lives, but she knows general trends.:

```
transition_probability = { 'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
                           'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6}
                         }
```

Based on what Bob tells her he did each day, Alice tries to guess what the weather must have been like.

Alice and Bob



0.7

0.6

0.3

0.4

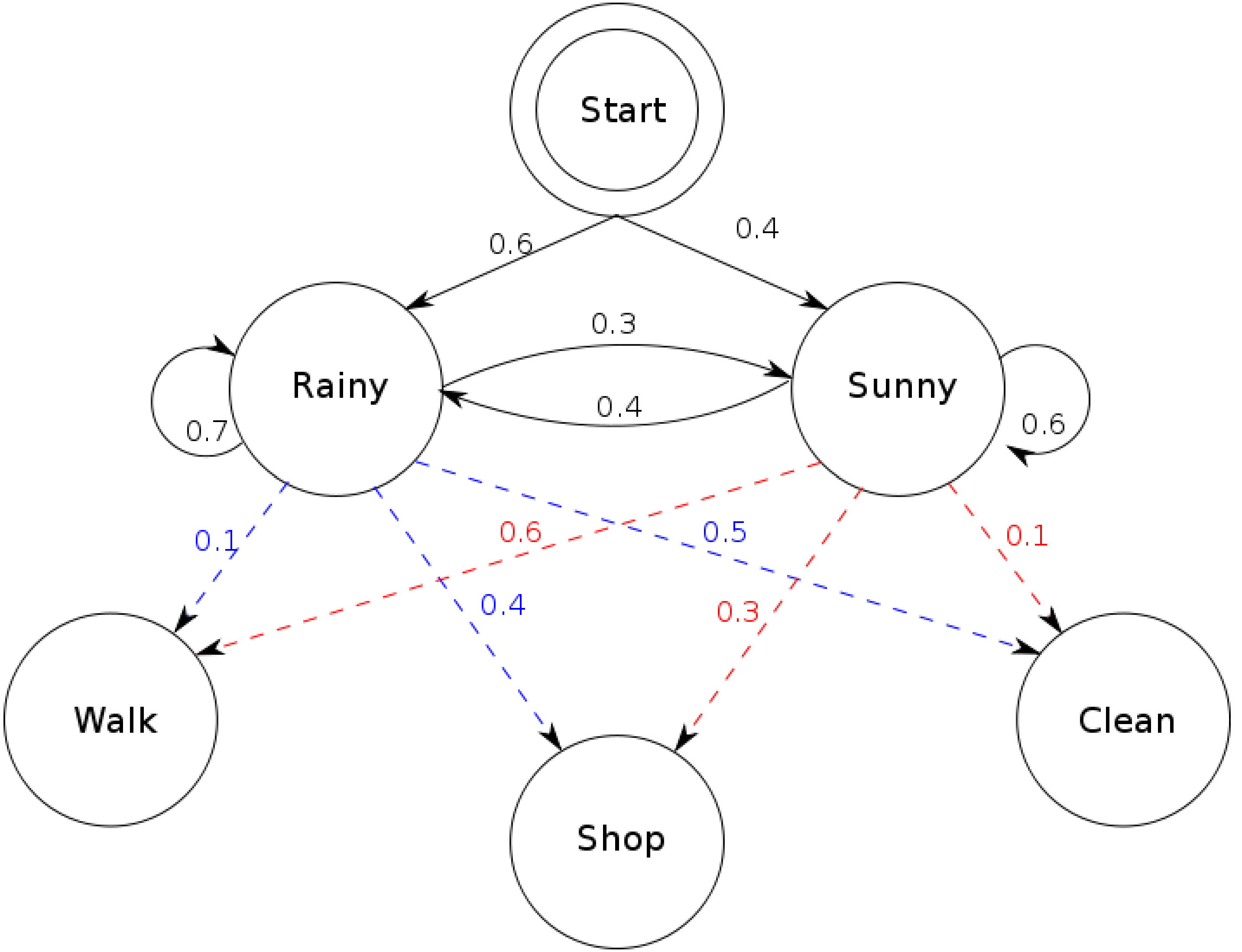
Walk: 0.1
Shop: 0.4
Clean: 0.5

Walk: 0.6
Shop: 0.3
Clean: 0.1

```
emission_probability = { 'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
                         'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1}
                       }

transition_probability = { 'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
                           'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6}
                         }
```

Alice and Bob

Just another way to represent the model:



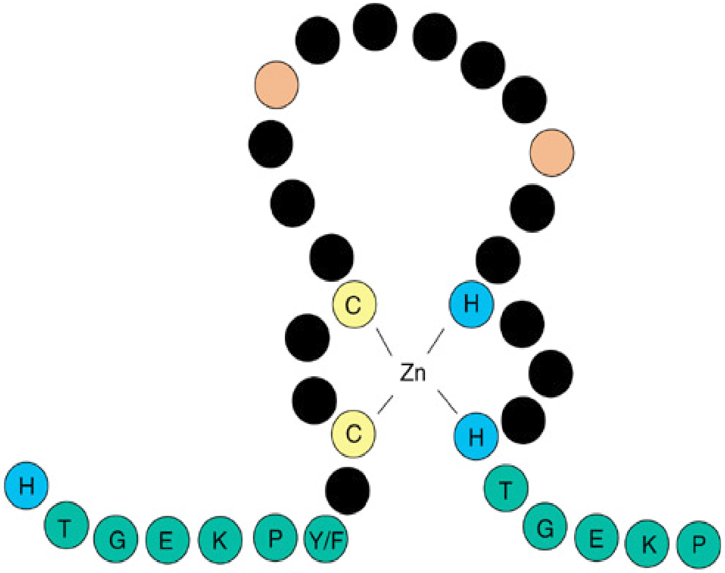
Profile HMMs

Most frequent application of HMMs in biological data
Good for domain finding

e.g. Find all ZNF genes in the platypus genome

Let's assume you have all human and mouse ZNF genes

```
E C G E C G K S F S S N V N L K G H Q R I   H T G E R P Y
E C G E C G K S F S Q N V M L K H H Q R I   H T G E R P Y
E C G E C G K S F S W N V L L K S H Q R I   H T G E R P Y
K C G E C G K S F S S N M K L K L H Q R I   H T G E R P Y
E C G E C G K S F S S N V H L K P H Q R I   H T G E R P Y
K C G E C E K S F S R K P G L S I H Q R I   H T E V R P Y
K C G E C E K S F S I K P S L S M H Q R I   H T E V R P Y
K C G E C E K S F S L K P P L S K H Q R I   H T E V R P Y
K C G E C D K S F S V K P T L G Y H Q R I   H T E V R P Y
K C G E C E K S F S R K P R L S E H Q R I   H T E V R P Y
K C G E C G K S Y I S K G H L R R H Q R M H T G E R P Y
K C G E C G K S Y I P K G S L R I H Q R M H T G E R P Y
K C G E C G K S Y I H K G T L R T H Q R M H T G E R P Y
K C G E C G K S Y I L K G H L R G H Q R M H T G E R P Y
K C G E C G K S Y I S K G L L R I H Q R M H T G E R P Y
K C G D C G K S F N E K G I L R S H Q R V H T T E R P Y
K C G D C G K S F N W K G H L R Y H Q R V H T T E R P Y
K C G D C G K S F N E K G P L R W H Q R V H T T E R P Y
K C G D C G K S F N Y K G L L R S H Q R V H T T E R P Y
K C G D C G K S F N G K G K L R L H Q R V H T T E R P Y
```



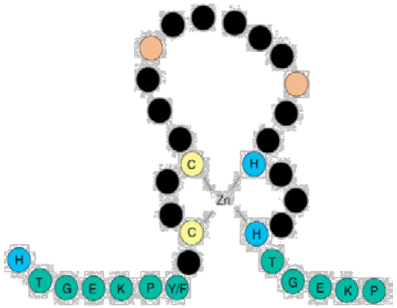
We could use the known ZNF genes to BLAST for homologues in the platypus genome
Or make PW alignments with all platypus genes

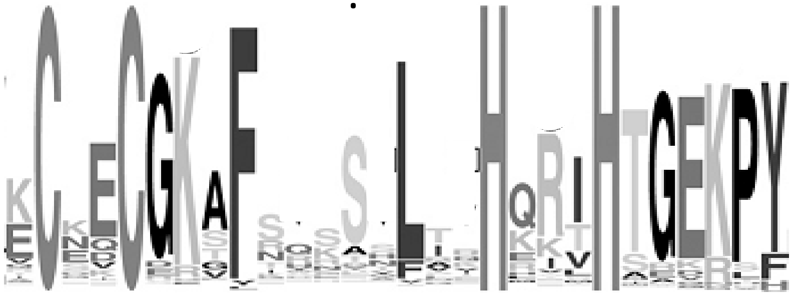
both might miss platypus specific / highly diverged ZNF genes

Profile HMMs

Or we could make a model of the family of ZNF domains and screen the genome for this model

```
E C G E C G K S F S S N V N L K G H Q R I  H T G E R P Y
E C G E C G K S F S Q N V M L K H H Q R I  H T G E R P Y
E C G E C G K S F S W N V L L K S H Q R I  H T G E R P Y
K C G E C G K S F S S N M K L K L H Q R I  H T G E R P Y
E C G E C G K S F S S N V H L K P H Q R I  H T G E R P Y
K C G E C E K S F S R K P G L S I H Q R I  H T E V R P Y
K C G E C E K S F S I K P S L S M H Q R I  H T E V R P Y
K C G E C E K S F S L K P P L S K H Q R I  H T E V R P Y
K C G E C D K S F S V K P T L G Y H Q R I  H T E V R P Y
K C G E C E K S F S R K P R L S E H Q R I  H T E V R P Y
                   ⋮
```





2nd and 5th position always C
18th and 22nd position always H
End often something like GERPY

PSSM
(Position Specific Scoring Matrix):
Some rule to convert frequencies to scores
But doesn't take gaps into account

	A	C	D
1	-1789	911	-1713
2	-1075	-662	-1597
3	-12231	5880	-11456
4	-1505	71	-244
5	-1624	-246	189
6	-12231	5880	-11456
7	-1719	-668	-514

. . .

⋮

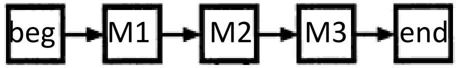
Profile HMMs

1. MSA:

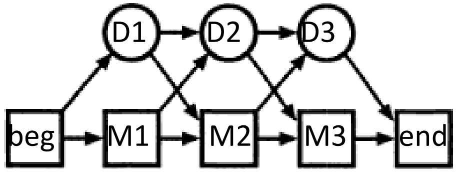
```
Bat:   AG---C
Rat:   A-AG-C
Cat:   AG-AA-
Gnat:  --AAAC
Goat:  AG---C

       12...3
```

2. Decide which columns to mark as match states
(e.g. all columns with more less than 50% deletions; here 1,2,3)

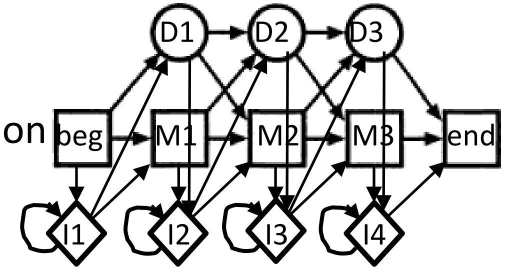
3. Start making the HMM
(matches are the states in our HMM; here M1-M3
residues are the emissions, have certain probability according to MSA)



4. Add deletions to HMM
(are states that do not emit any residue; here D1-D3
you can go from any residue to any other residue
by using a sequence of deletions)



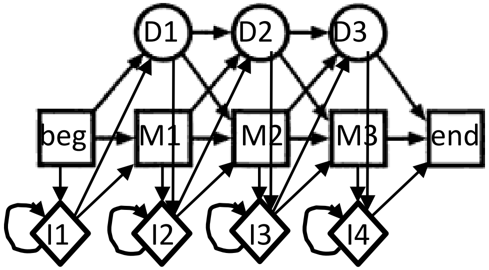
5. Add insertions to HMM
(they are treated as new states; here I1-I5
can emit any residue, probability usually taken from background distribution
there can be multiple insertions)



Profile HMMs

1. MSA:

```
Bat:   AG---C
Rat:   A-AG-C
Cat:   AG-AA-
Gnat:  --AAAC
Goat:  AG---C

       12...3
          .
          .
          .
```



Model position: 0 1 2 3

6. Get transition and emission probabilities
(count/observe transitions and emissions from MSA;
to avoid zero probabilities add pseudocountes,
e.g. add 1 to each observation;
here without pseudocounts)

		0	1	2	3
Match emissions	**A**	-	4	0	0
	C	-	0	0	4
	G	-	0	3	0
	T	-	0	0	0
Insert emissions	**A**	0	0	6	0
	C	0	0	0	0
	G	0	0	1	0
	T	0	0	0	0
State transitions	**M-M**	4	3	2	4
	M-D	1	1	0	0
	M-I	0	0	1	0
	I-M	0	0	2	0
	I-D	0	0	1	0
	I-I	0	0	4	0
	D-M	-	0	0	1
	D-D	-	1	0	0
	D-I	-	0	2	0

Profile HMMs

e.g. Find all ZNF genes in the platypus genome

Let’s assume you have all human and mouse ZNF genes
Use them to build/train your HMM

```
E C G E C G K S F S S N V N L K G H Q R I H T G E R P Y
E C G E C G K S F S Q N V M L K H H Q R I H T G E R P Y
E C G E C G K S F S W N V L L K S H Q R I H T G E R P Y
K C G E C G K S F S S N M K L K L H Q R I H T G E R P Y
E C G E C G K S F S S N V H L K P H Q R I H T G E R P Y
K C G E C E K S F S R K P G L S I H Q R I H T E V R P Y
K C G E C E K S F S I K P S L S M H Q R I H T E V R P Y
K C G E C E K S F S L K P P L S K H Q R I H T E V R P Y
K C G E C D K S F S V K P T L G Y H Q R I H T E V R P Y
K C G E C E K S F S R K P R L S E H Q R I H T E V R P Y
K C G E C G K S Y I S K G H L R R H Q R M H T G E R P Y
K C G E C G K S Y I P K G S L R I H Q R M H T G E R P Y
K C G E C G K S Y I H K G T L R T H Q R M H T G E R P Y
K C G E C G K S Y I L K G H L R G H Q R M H T G E R P Y
K C G E C G K S Y I S K G L L R I H Q R M H T G E R P Y
K C G D C G K S F N E K G I L R S H Q R V H T T E R P Y
K C G D C G K S F N W K G H L R Y H Q R V H T T E R P Y
K C G D C G K S F N E K G P L R W H Q R V H T T E R P Y
K C G D C G K S F N Y K G L L R S H Q R V H T T E R P Y
K C G D C G K S F N G K G K L R L H Q R V H T T E R P Y
```

train →

		0	1	…	28
Match emissions	A				
	C				
	D				
	E				
	…				
Insert emissions	A				
	C				
	D				
	E				
	…				
State transitions	M-M				
	M-D				
	M-I				
	I-M				
	I-D				
	I-I				
	D-M				
	D-D				
	D-I				

Scan the translated platypus genome with this HMM
Finds everything that looks like a ZNF domain, hence also diverged and platypus specific ZNF proteins

How do we get the transition and emission probabilities if the path is unknown?

- In the ZNF example the path was known, so we could just count observations
- In the casino example the path is unknown …

Typically split data in training set and test set

Baum-Welch algorithm:

<u>Training:</u>

Initialization: Start the HMM with some arbitrary model parameters

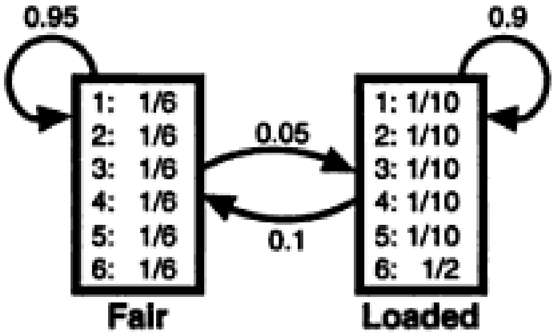
Recurrence:

```
Set all transition A and emission E variables to their pseudocounts or to zero
For each sequence j = 1 .. n
        Calculate how likely is j given the current HMM
        Calculate how much j contributes to current A and E
Calculate new model parameters using maximum likelihood estimation: $a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}}$ and $e_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')}$
Calculate new log likelihood of the model
```

Stop: when change in log likelihood is less than a predefined threshold
or when predefined maximum number of iterations in reached

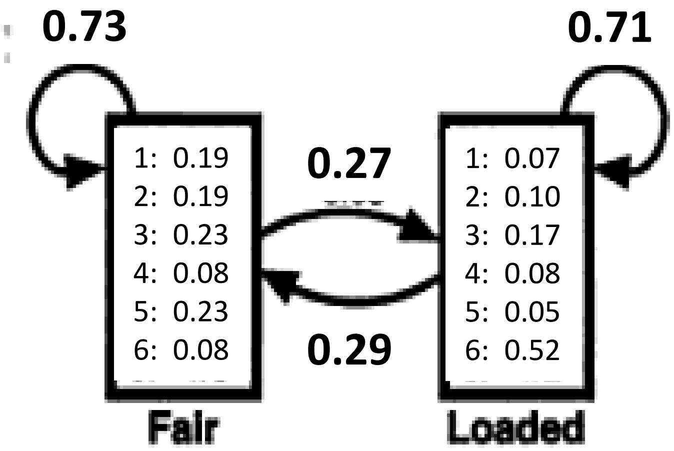
How do we get the transition and emission probabilities if the path is unknown?

“real” model:



Would be useful to know when the loaded dice is in play to bet for the 6

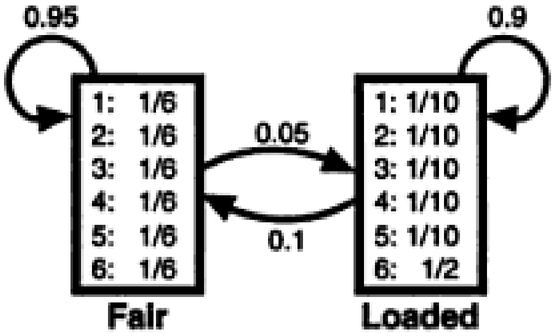
Estimated model based on 300 rolls:



Rolls: 31511624644664424531143631656626566666511664531326512456366646316 …
Die: FFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFFFFLLLLLLLLLLLL …
Est: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFFFFLLLLLLLLLLLL …

How do we get the transition and emission probabilities if the path is unknown?

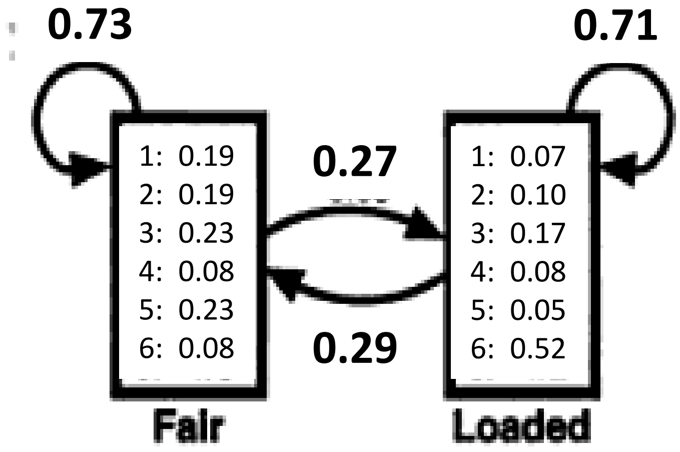
“real” model:



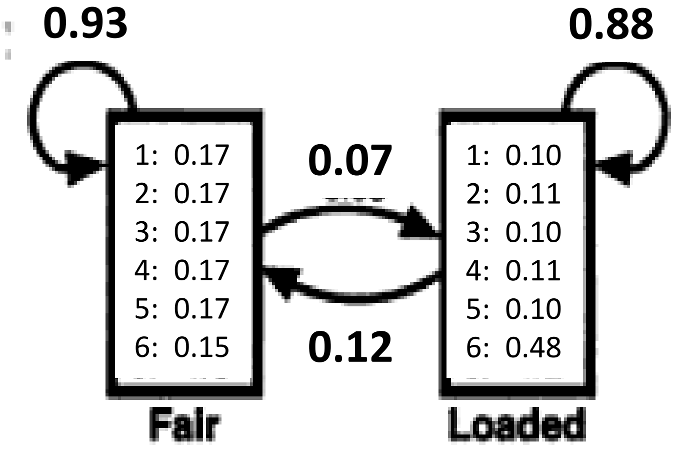
Would be useful to know when the loaded dice is in play to bet for the 6

Model estimate:

Collected observations of 300 rolls

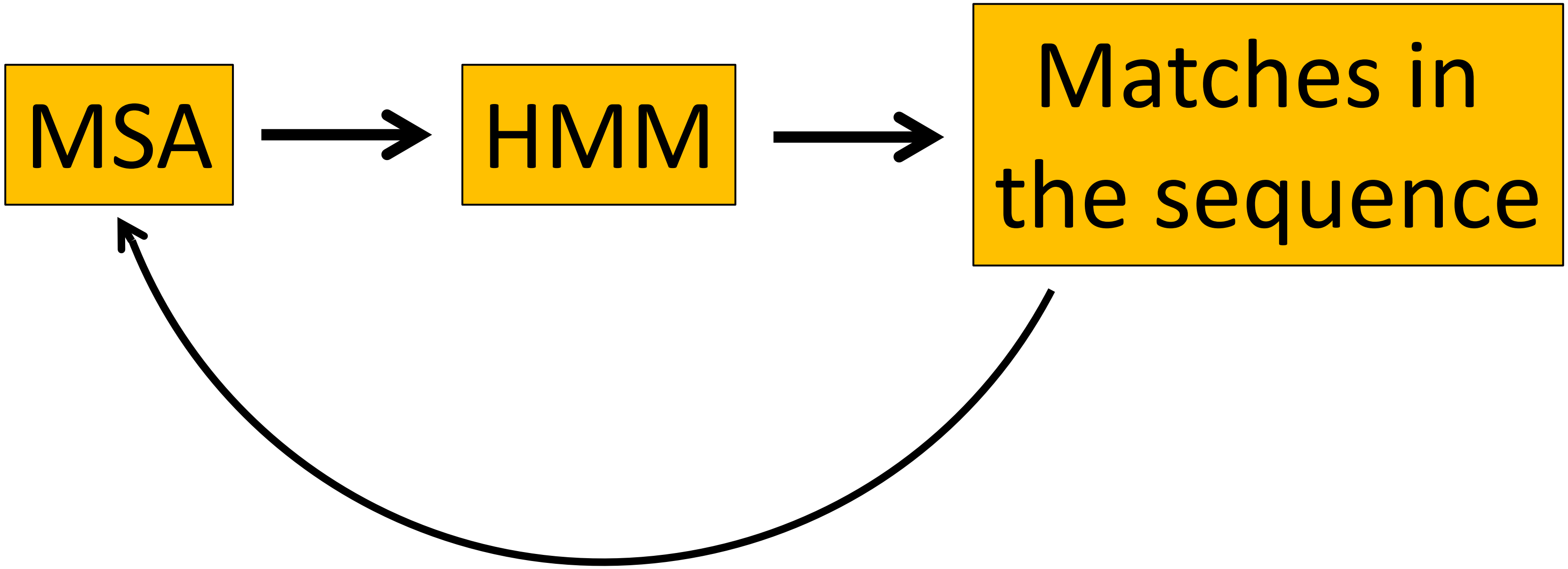


Collected observations of 30.000 rolls



The more observations the better the model

HMMER



http://hmmer.janelia.org/

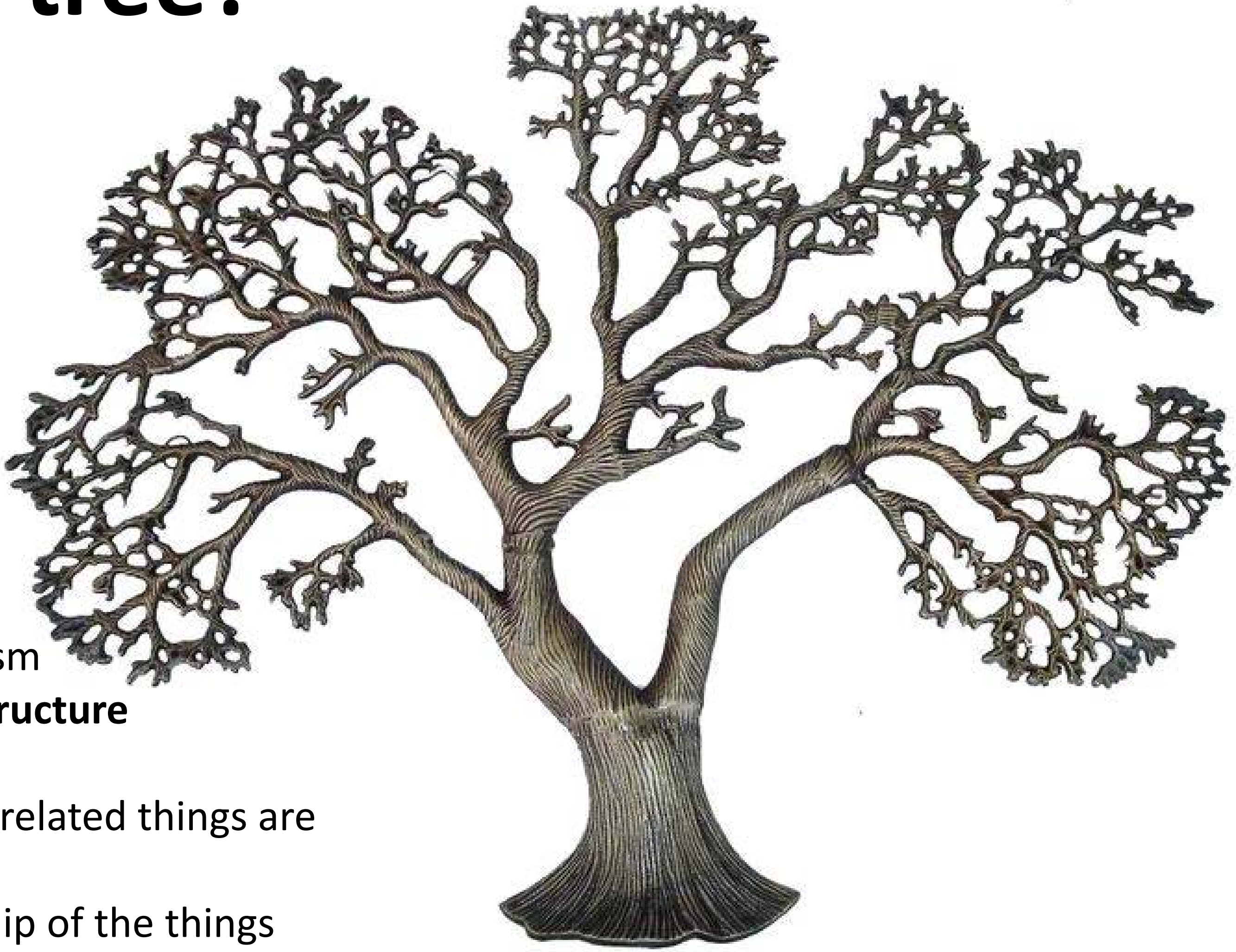
Sequence analysis and genomics

5. Introduction to Phylogenetics

Dr. Katja Nowick

Group leader TFome and Transcriptome Evolution
Bioinformatics group
Paul-Flechsig-Institute for Brain Research
www. nowicklab.info
nowick@bioinf.uni-leipzig.de

What is a tree?



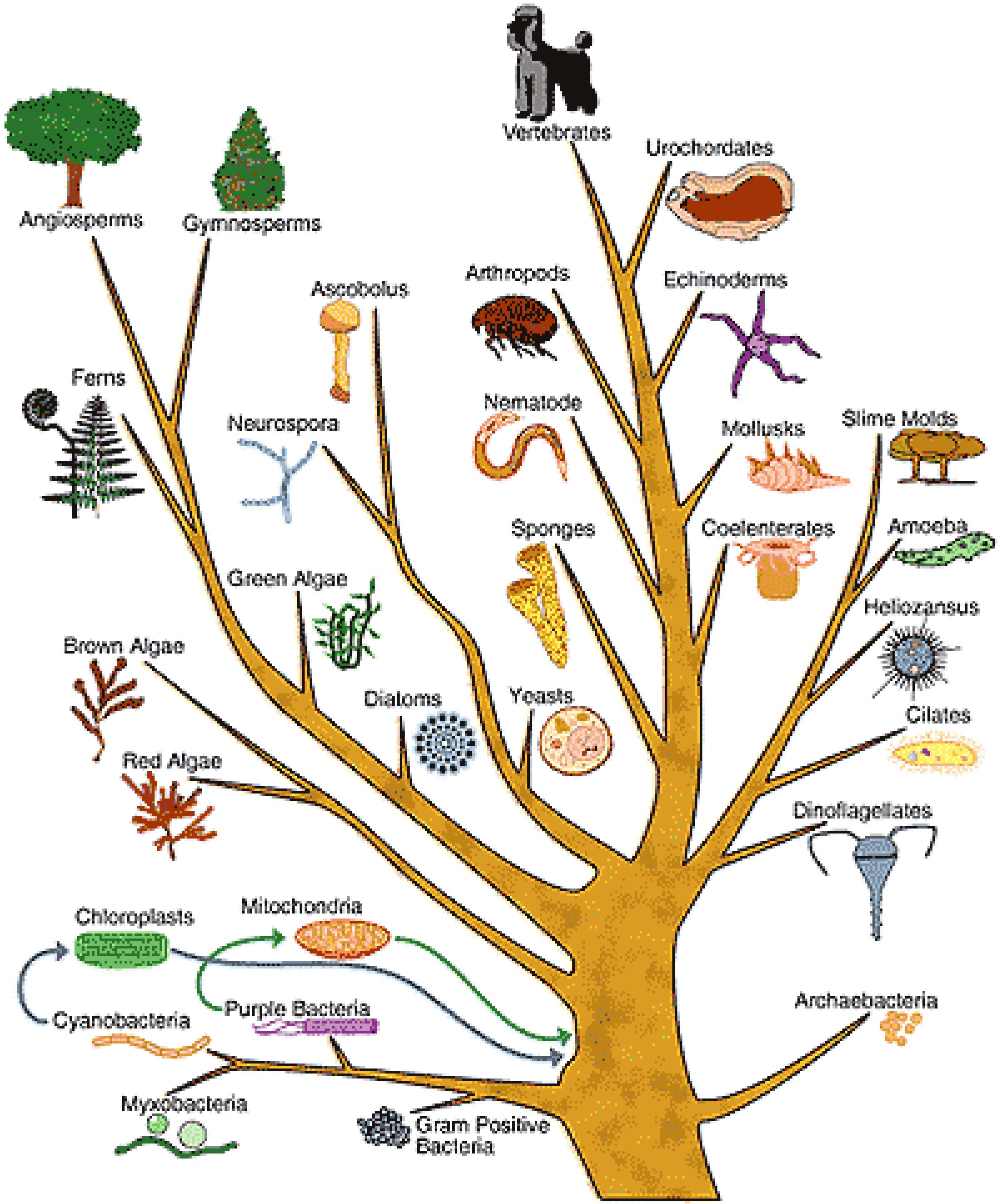
Tree = a biological organism

= a mathematical structure

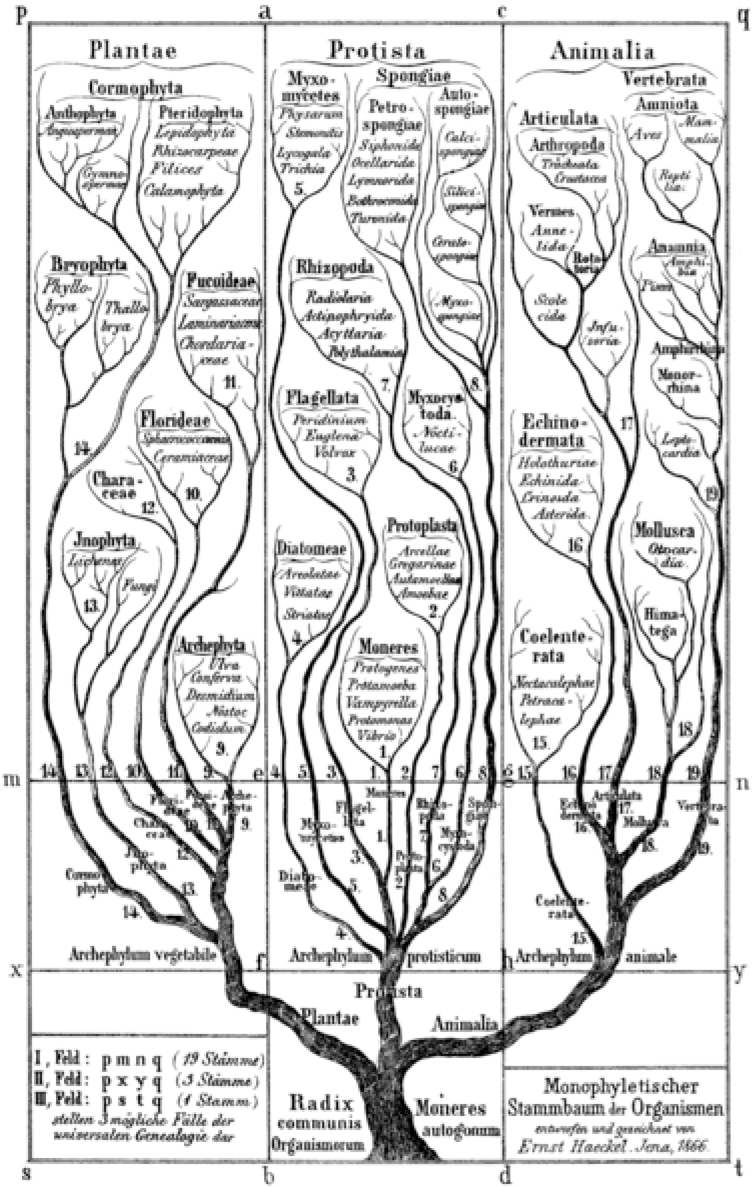
Trees reflect how similar/related things are
Leaves = Things
Branches show relationship of the things

Examples:
Species tree – how similar are species
Gene trees – how similar are genes
…

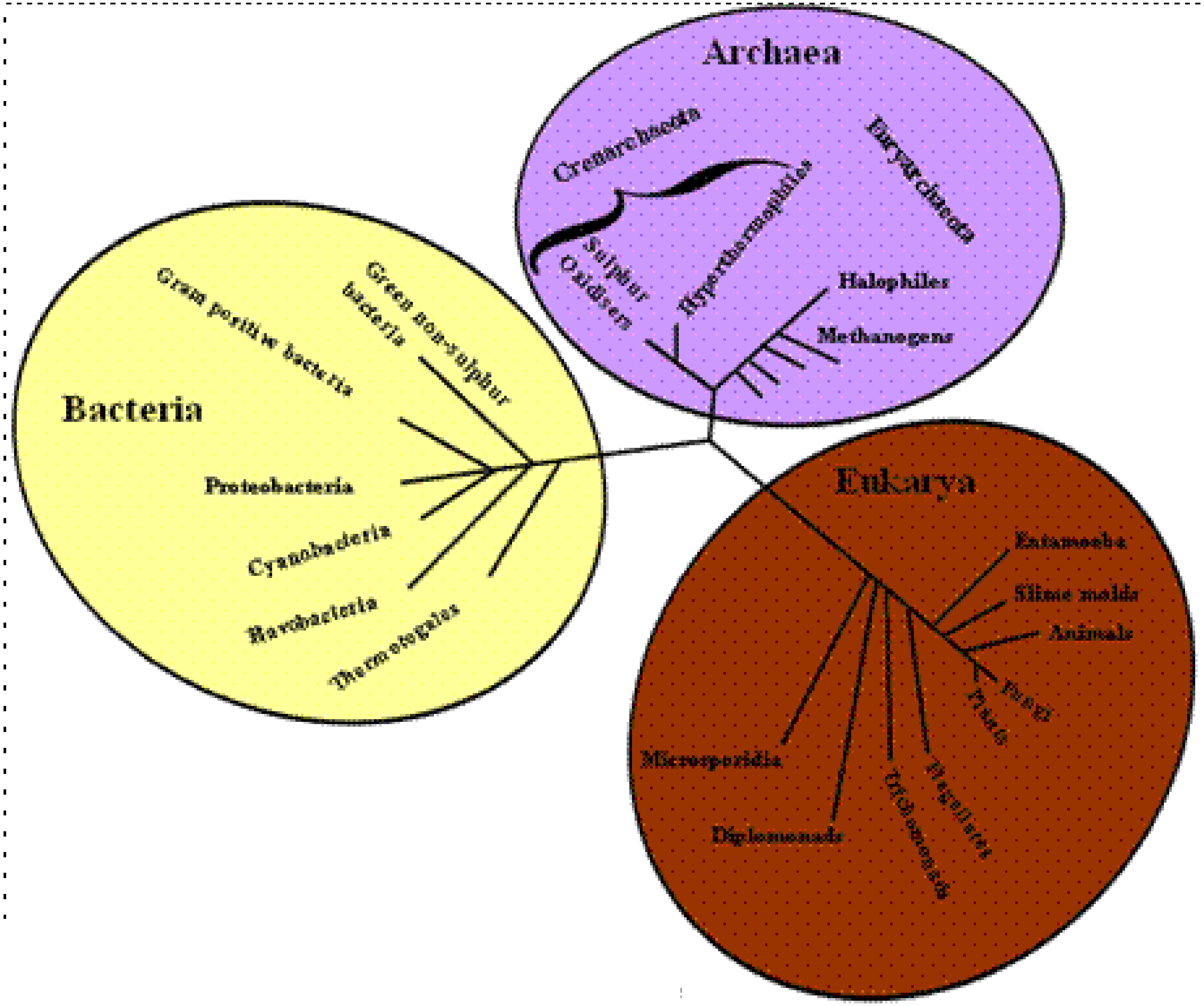
A species tree



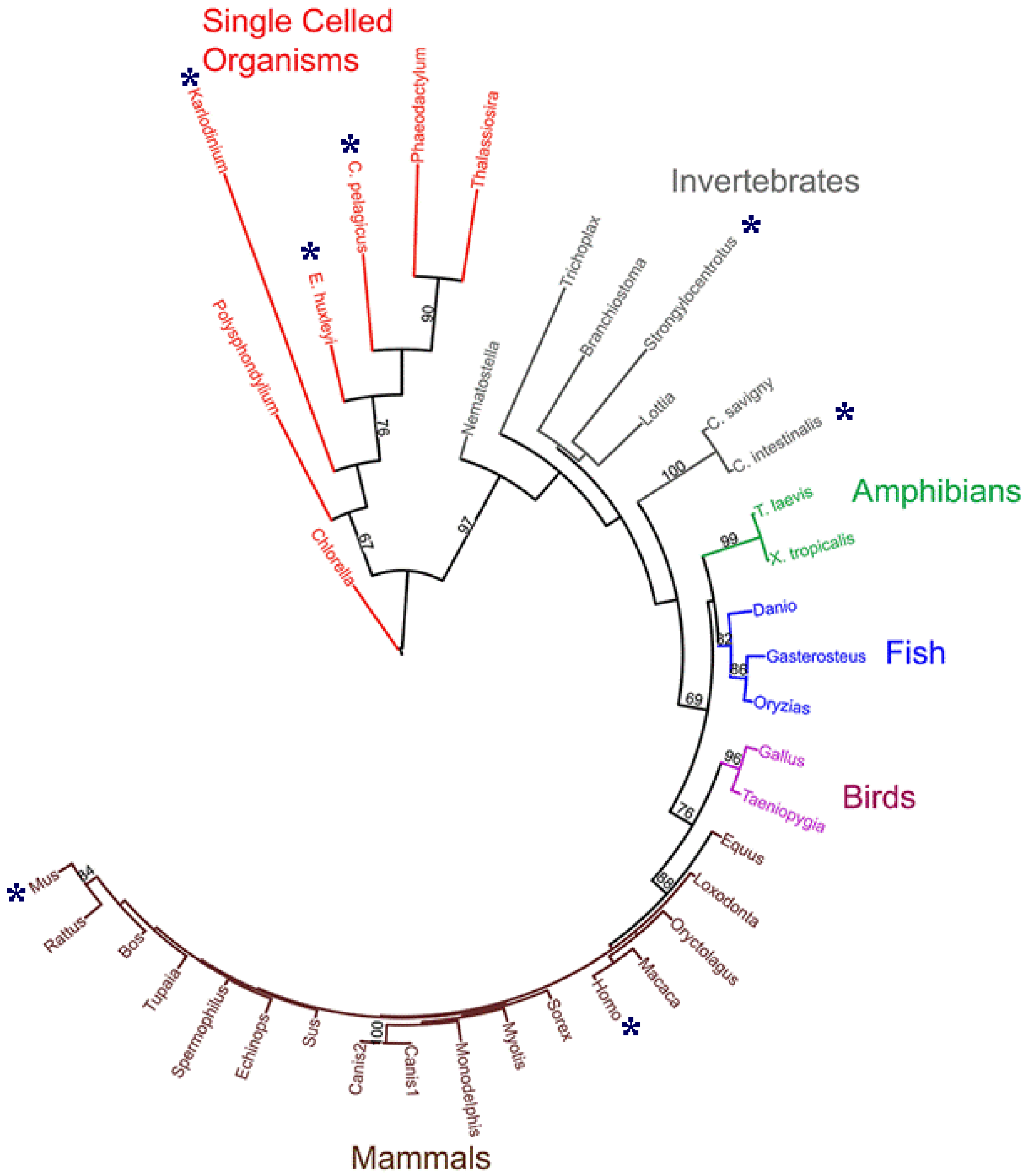
Haeckel’s tree of life: another species tree



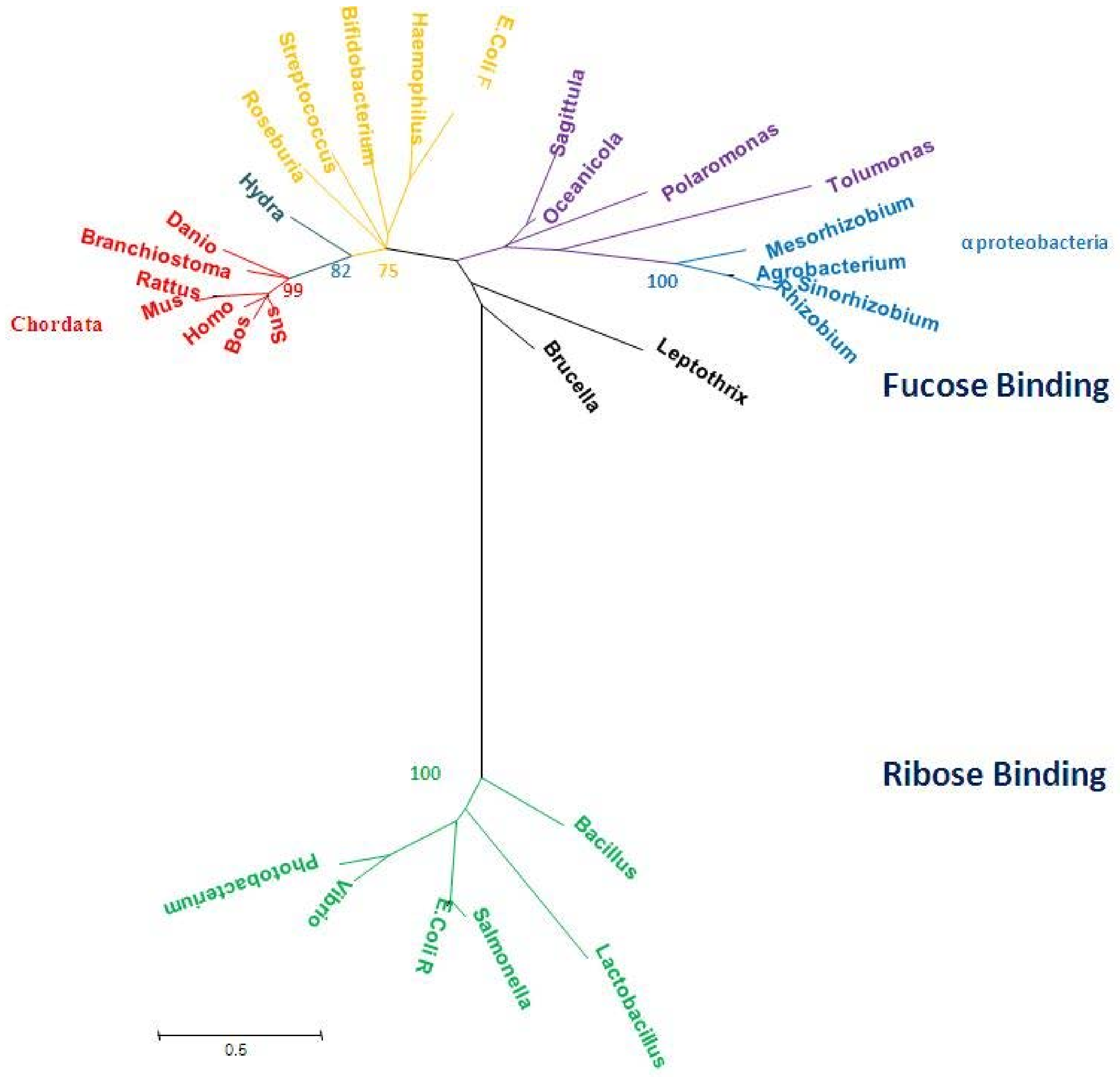
Tree of life: Another species tree



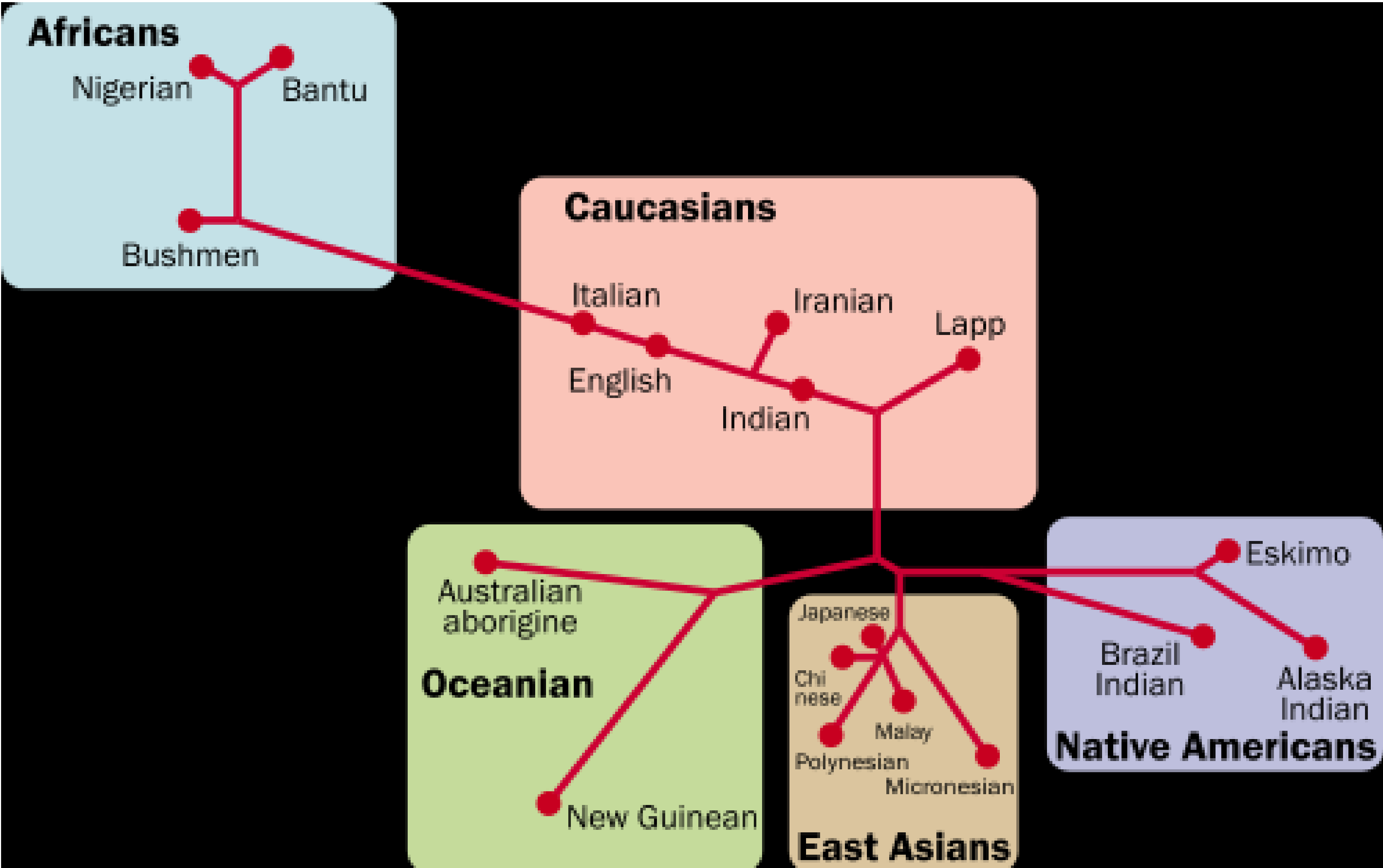
Another visualization of a species tree



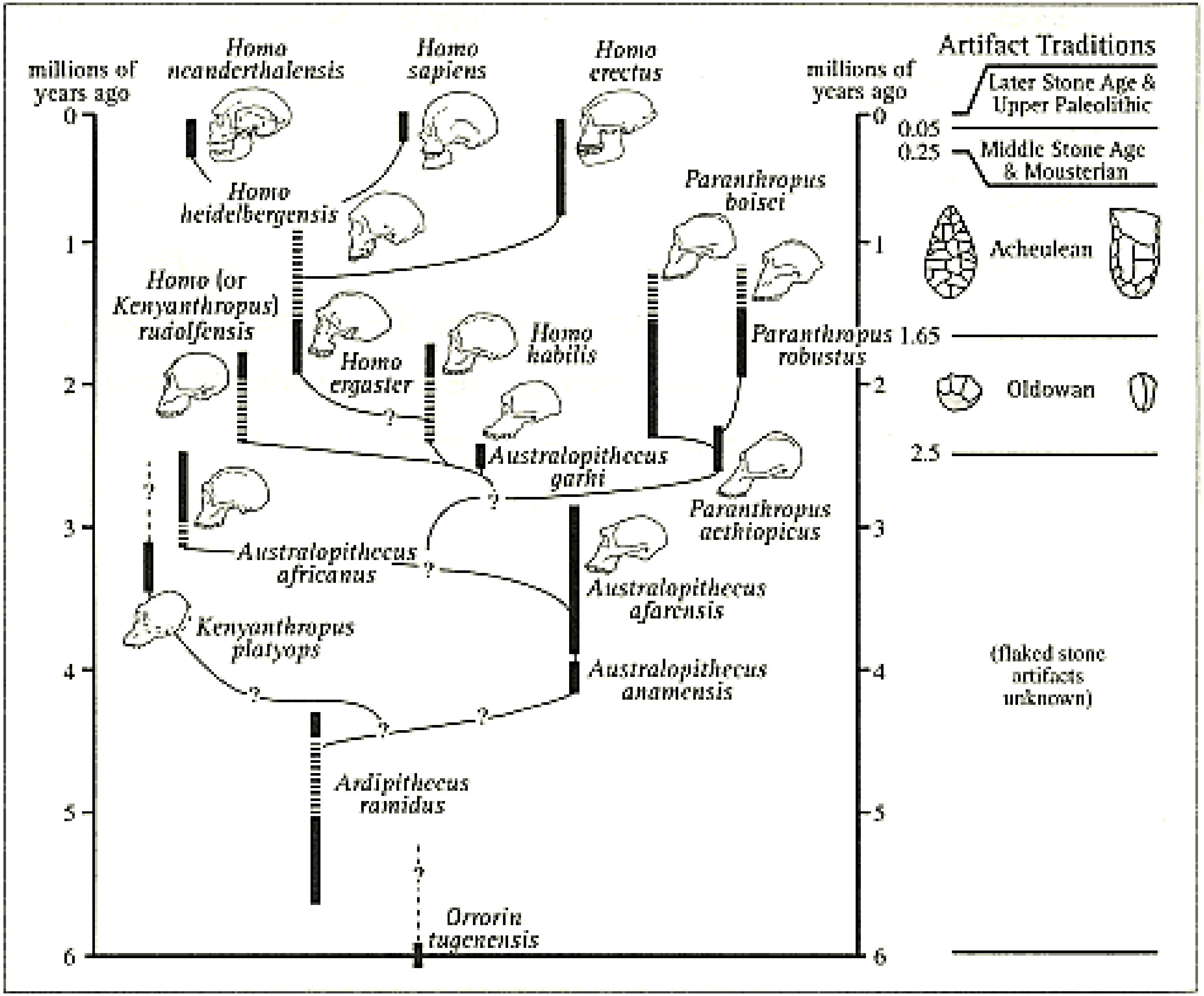
Another visualization of a species tree



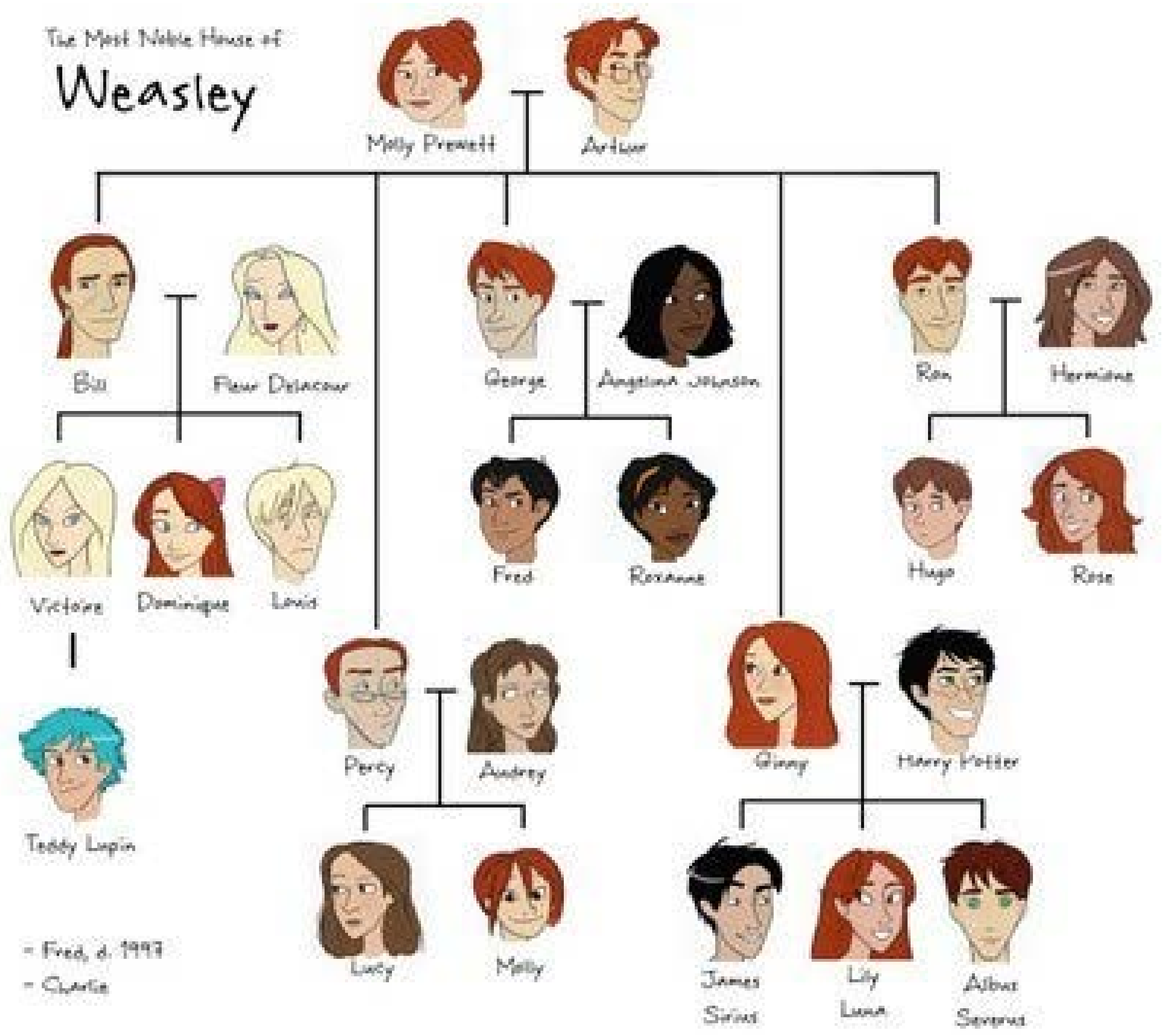
Tree of human populations



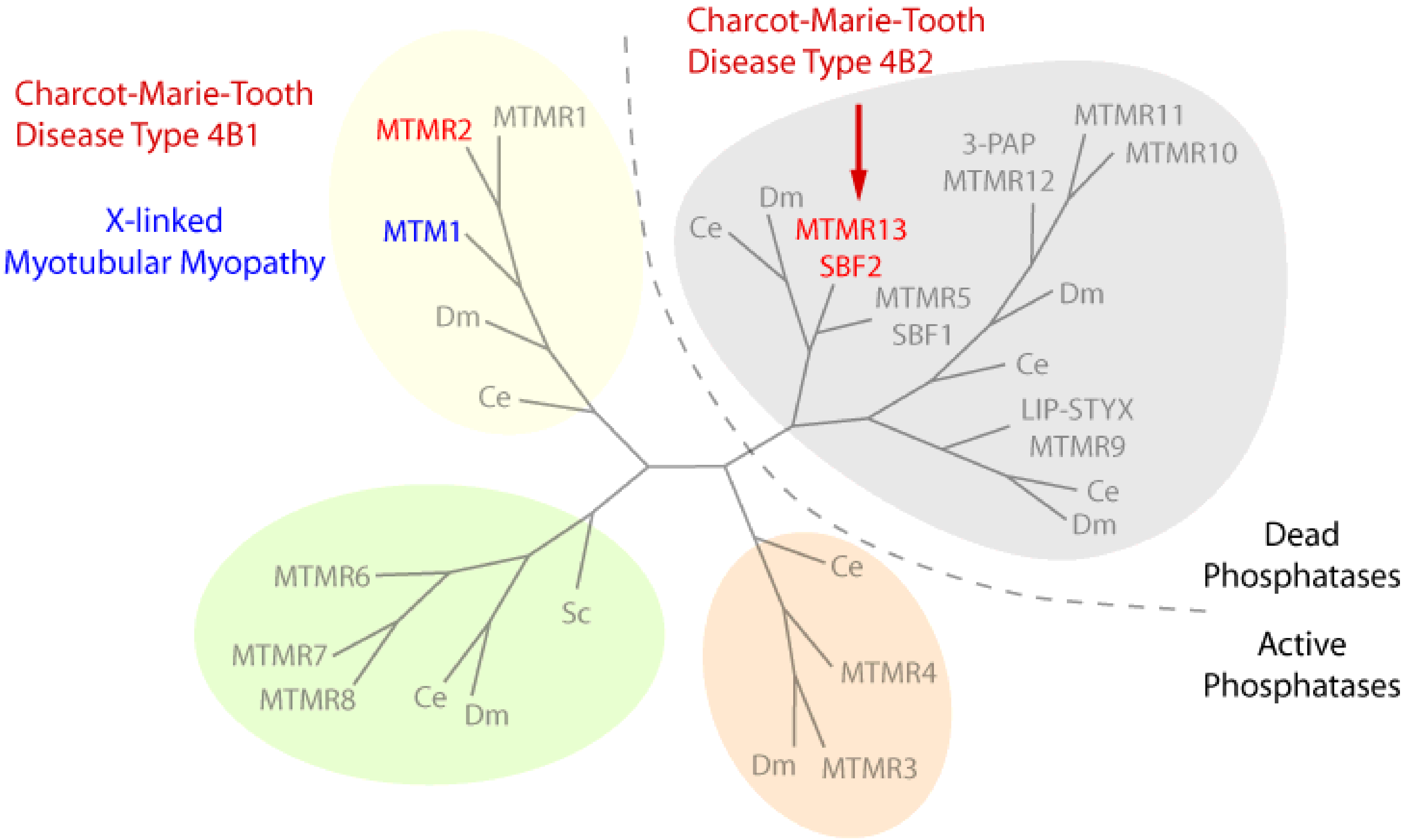
Tree of human species



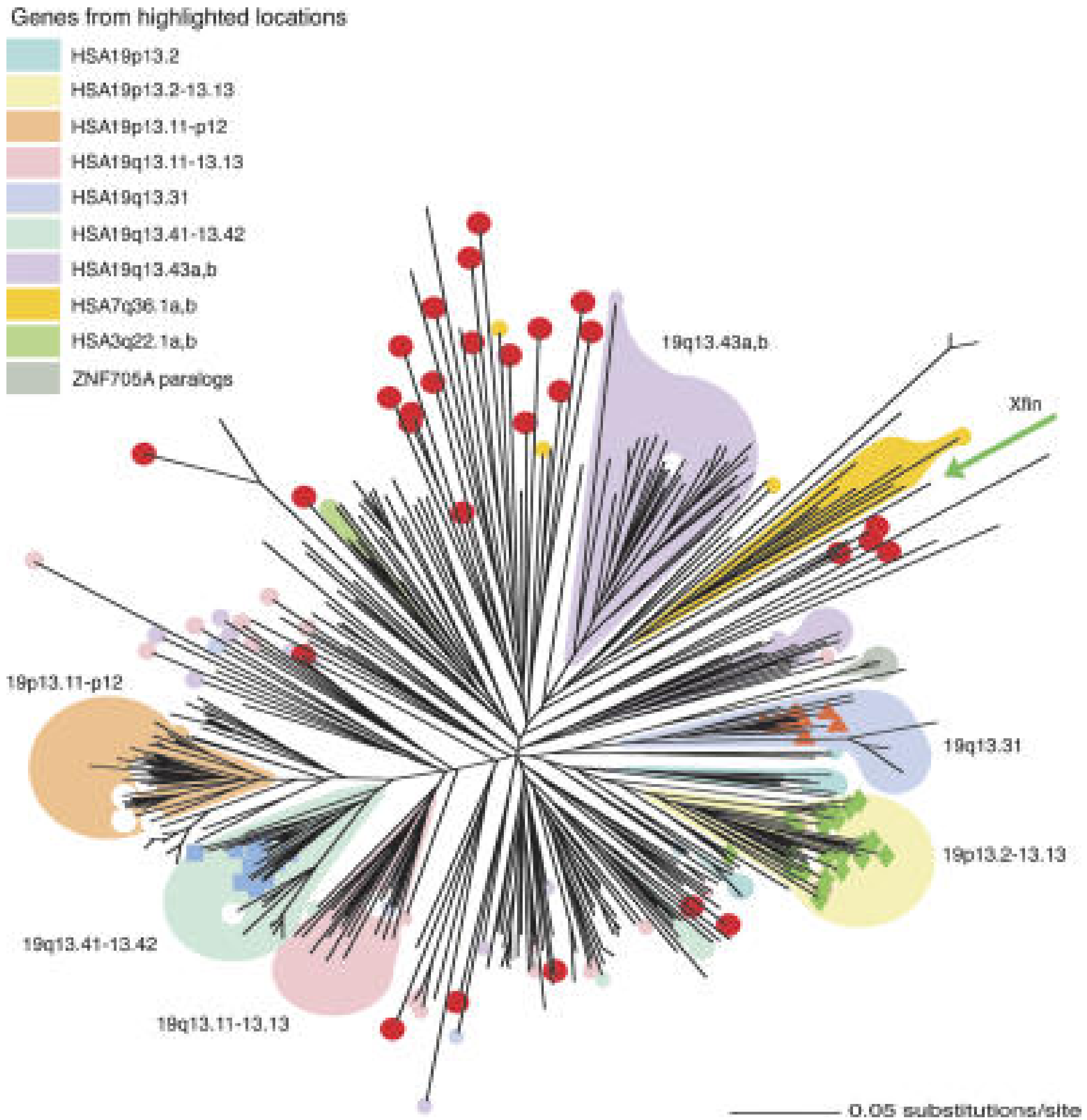
A family tree



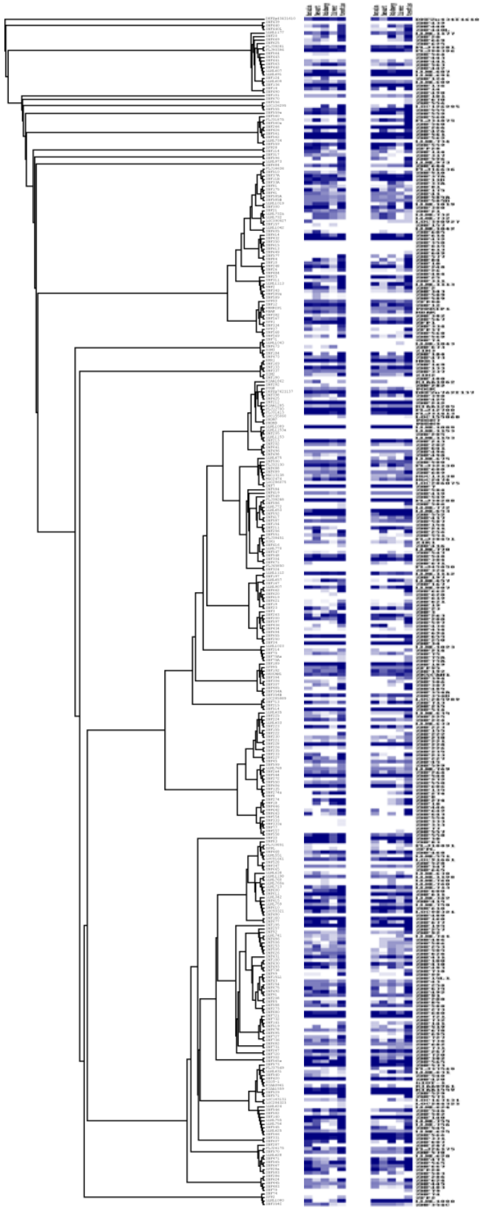
A gene tree


Charcot-Marie-Tooth
Disease Type 4B2
Charcot-Marie-Tooth
Disease Type 4B1
X-linked
Myotubular Myopathy
MTMR2
MTMR1
MTM1
Dm
Ce
Dm
Ce
MTMR13
SBF2
MTMR5
SBF1
3-PAP
MTMR12
MTMR11
MTMR10
Dm
Ce
LIP-STYX
MTMR9
Ce
Dm
Dead
Phosphatases
Active
Phosphatases
MTMR6
MTMR7
MTMR8
Ce
Dm
Sc
Ce
MTMR4
Dm
MTMR3

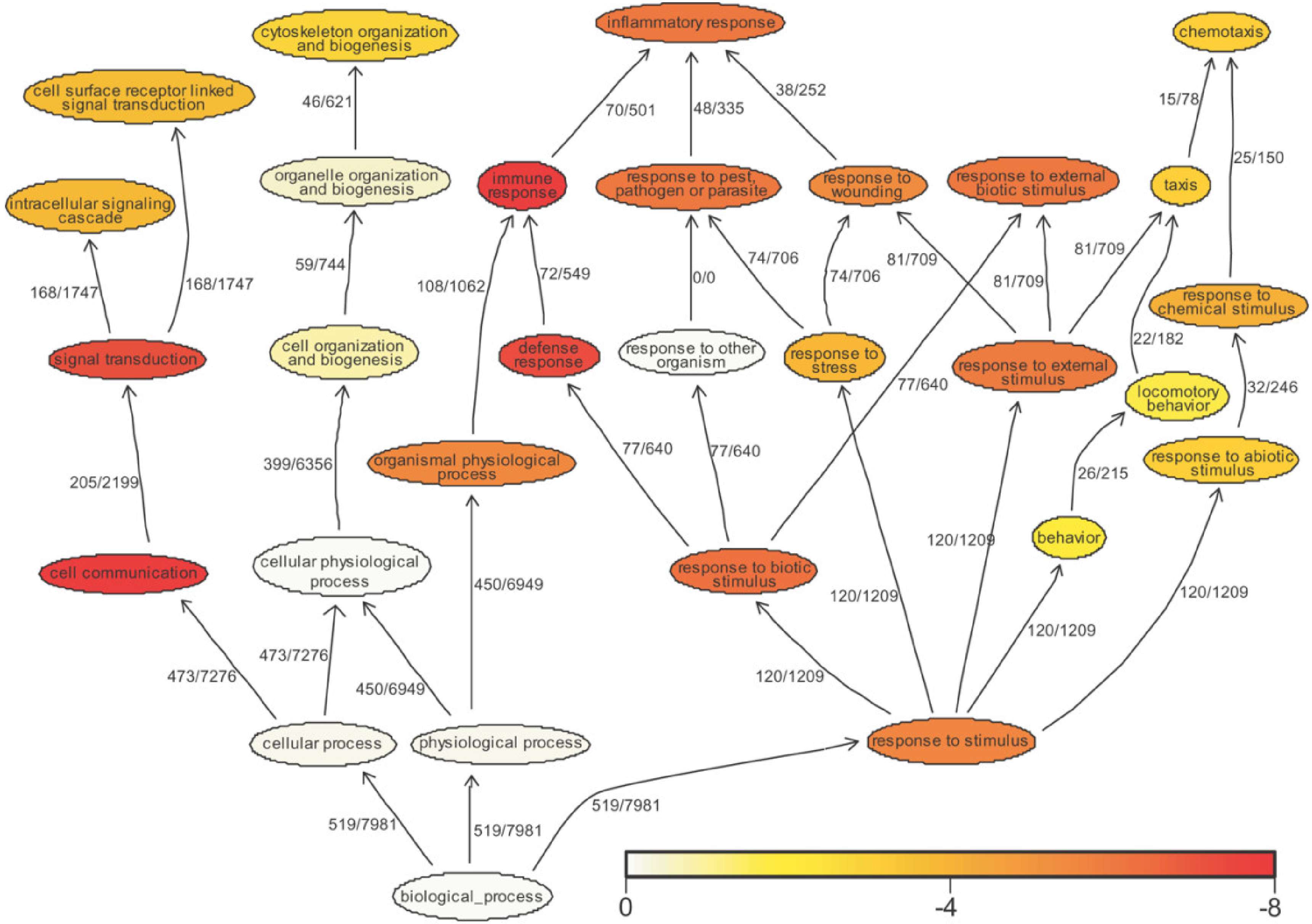
Another gene tree



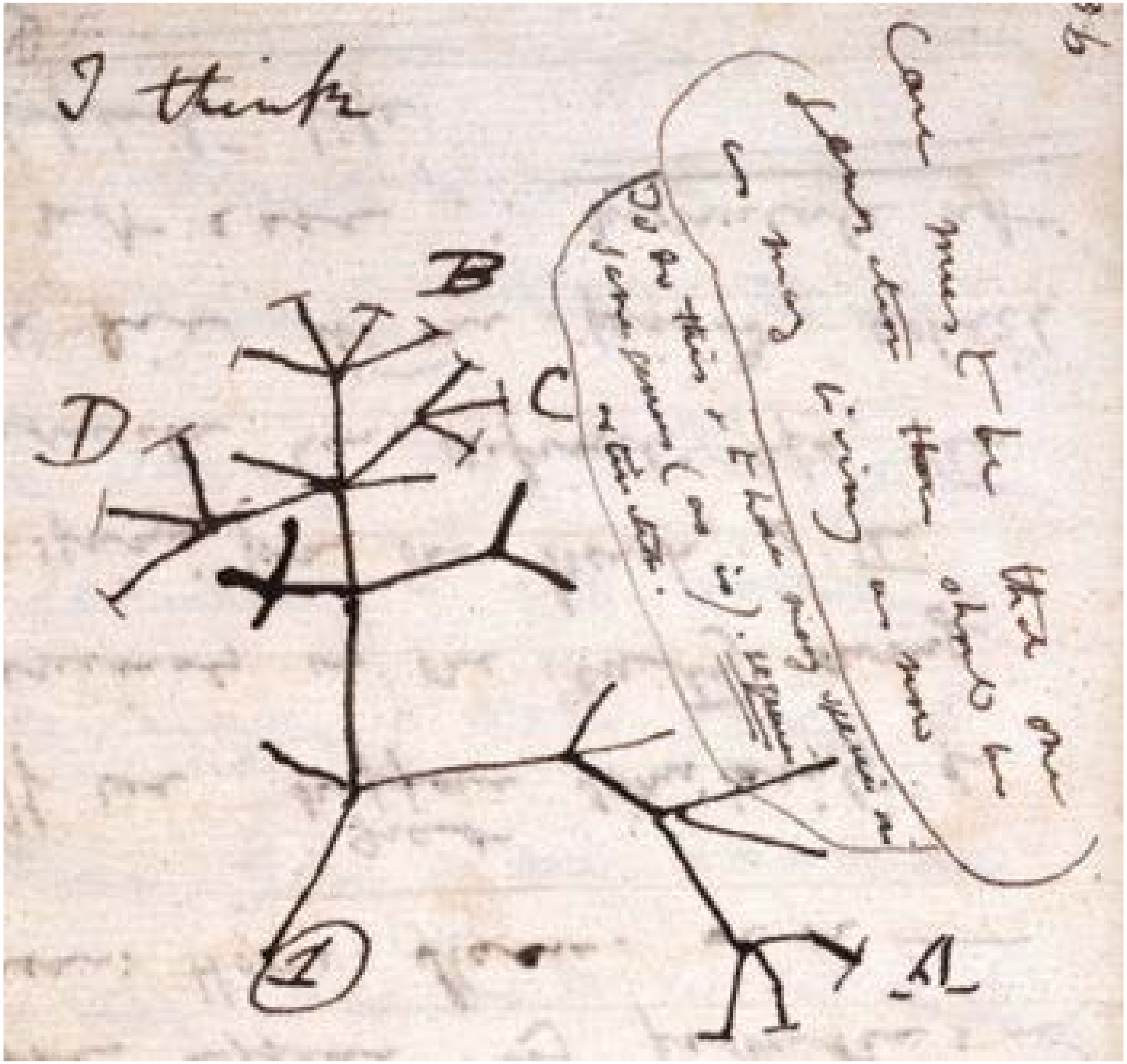
A tree of gene expression patterns in multiple tissues



A tree of Gene Ontology groups



The first drawn tree (Darwin)



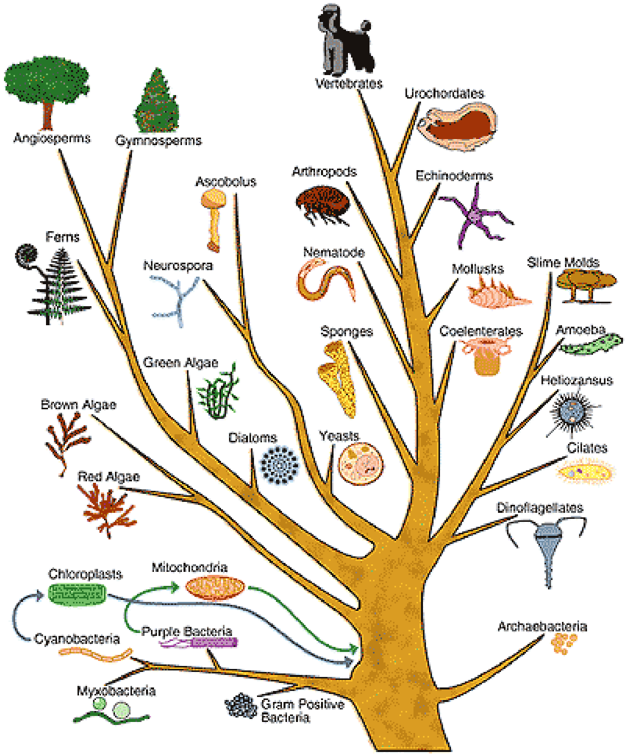
Case must be that one generation then should be as many living as now. To do this & to have many species in same genus (as is) requires extinction.

Topic today: Evolutionary trees / phylogenetic trees

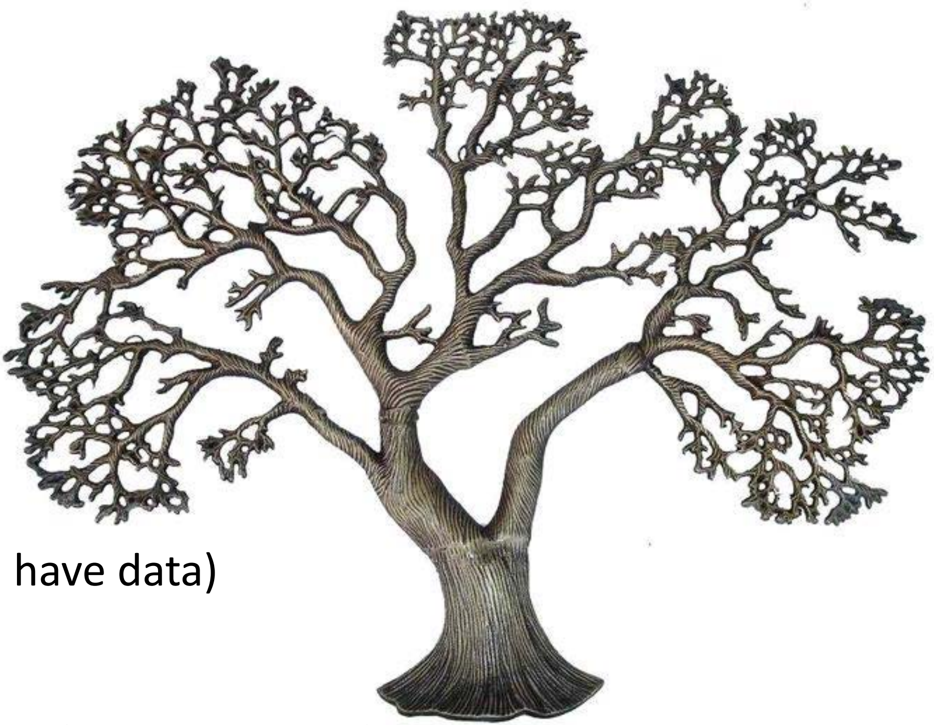
All in life is related by common ancestry

Phylogenetics refers to the evolutionary relatedness of organisms (species, populations …)

But all the following methods can be used for any type of data as long as characters have more than one state



Terminology



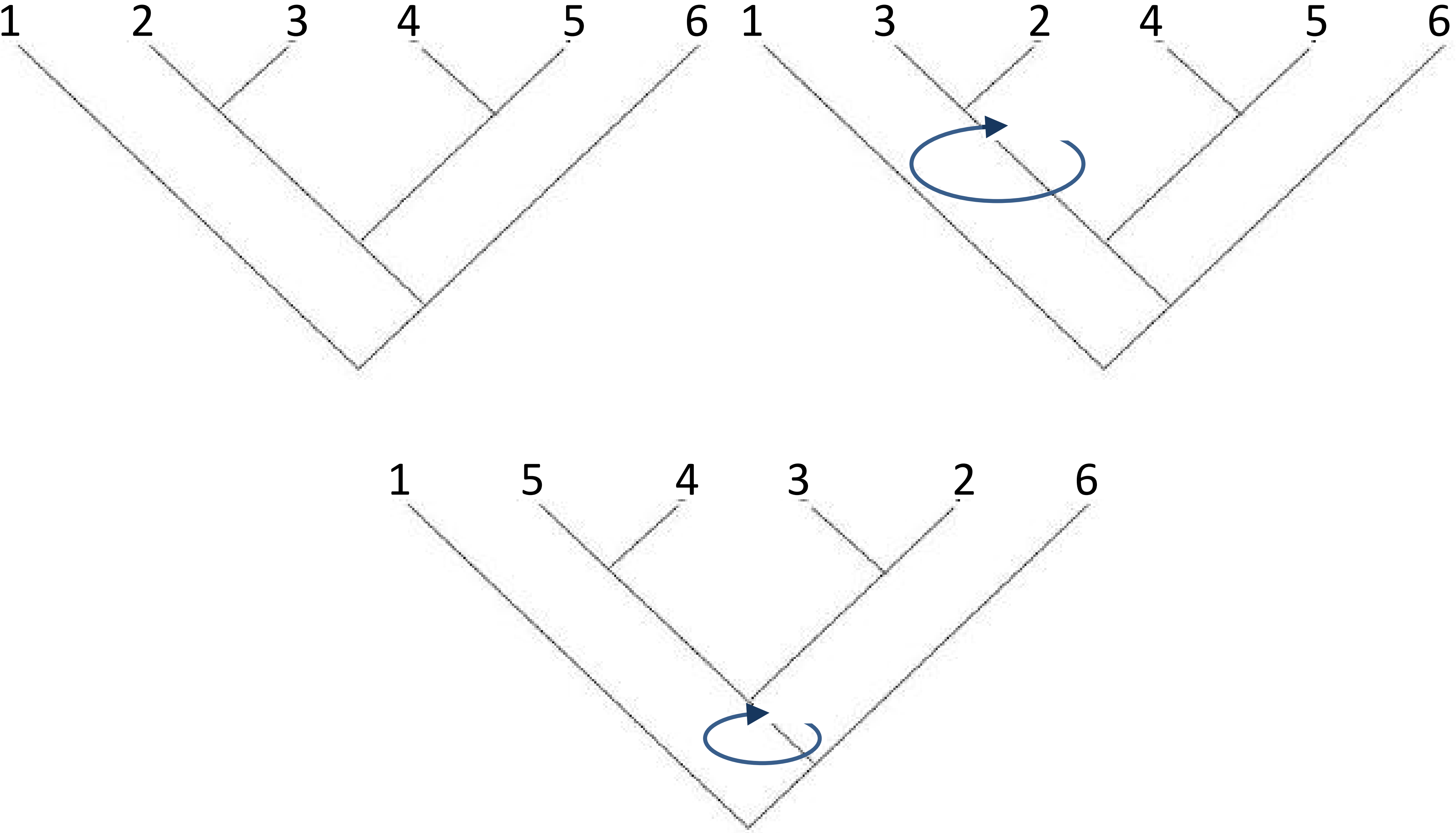
Tree = a mathematical structure

Trees reflect how similar/related things are
Things = nodes in the tree, e.g. species
Terminal nodes (leaves) = species (for which we have data)
Internal nodes = inferred ancestors
Branches (edges) show relationship
Length of the branch can reflect evolutionary time (weighted trees)

Terminology differs between disciplines

Usually data for ancestors are sparse (only fossils),
so the evolutionary history has to be reconstructed based on living species

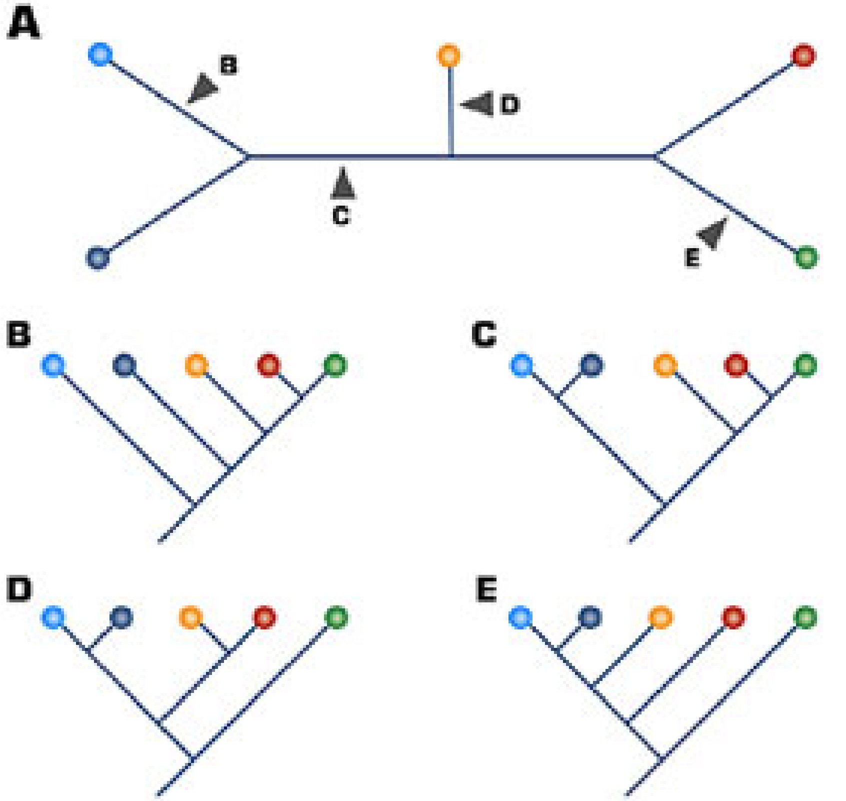
Branches can be freely rotated



Unrooted vs rooted trees

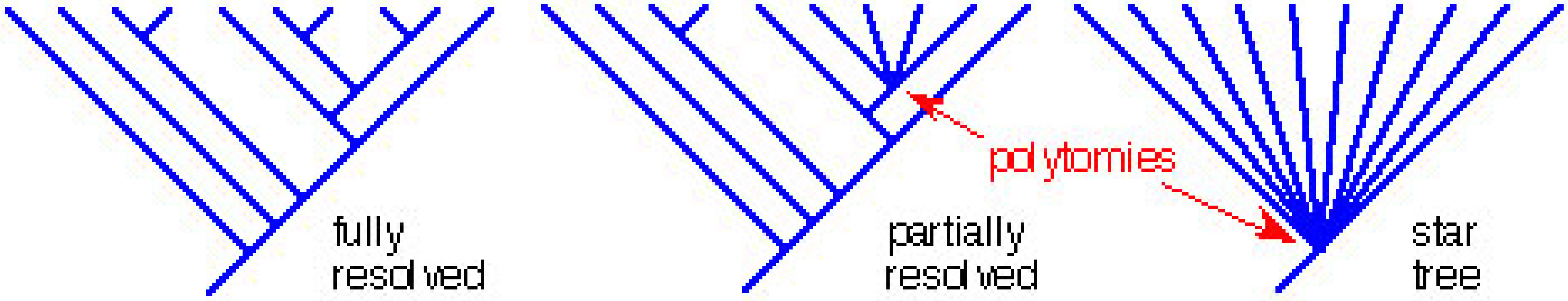
Unrooted

Rooted
Choosing different branches as roots



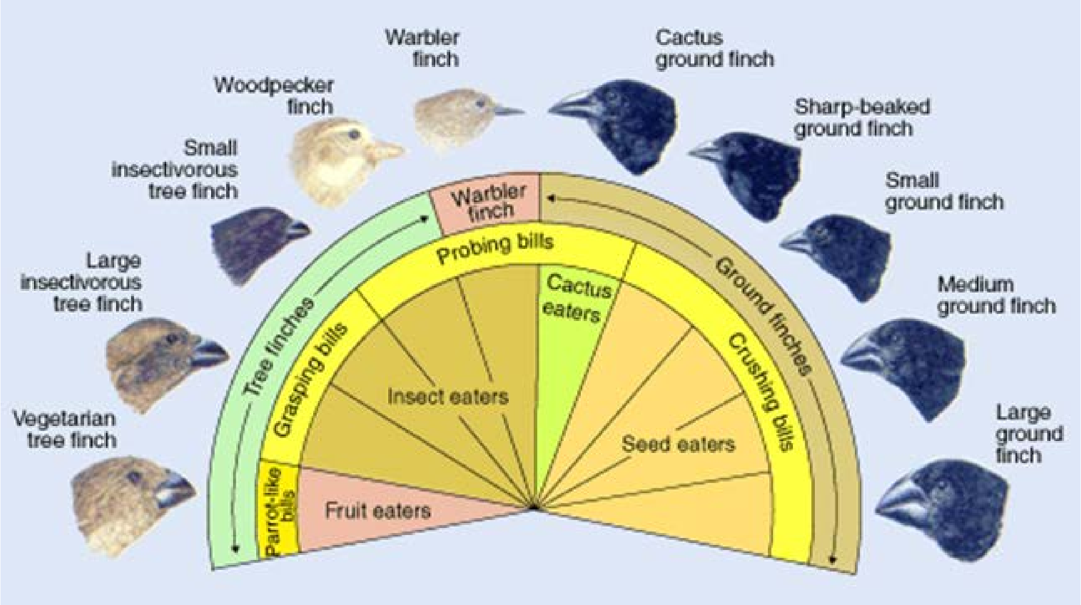
Only rooted trees have an evolutionary direction

Resolved vs unresolved trees

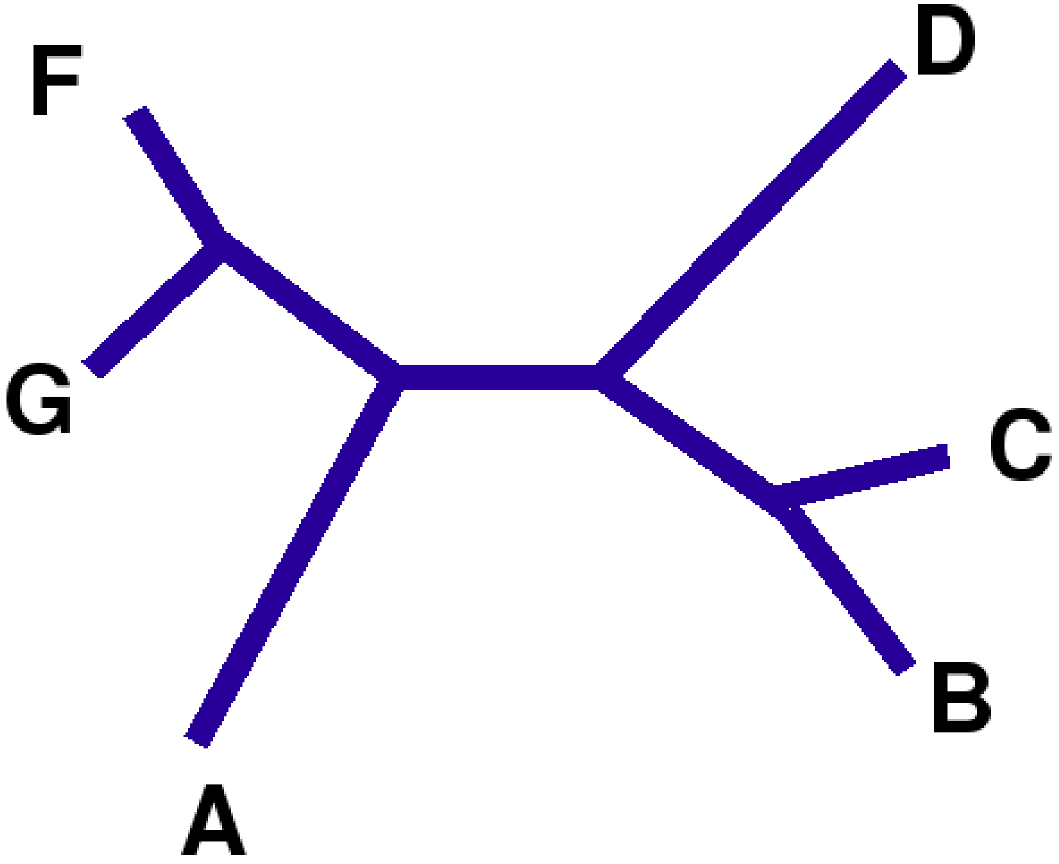


Polytomy = when a node has more than 3 branches (more than 1 ancestor + two descendants)
Either the lineages really diverged at the same time (or very rapidly)
or we don't know what the real divergence pattern is

Example of a star tree:
Radiation of Darwin finches



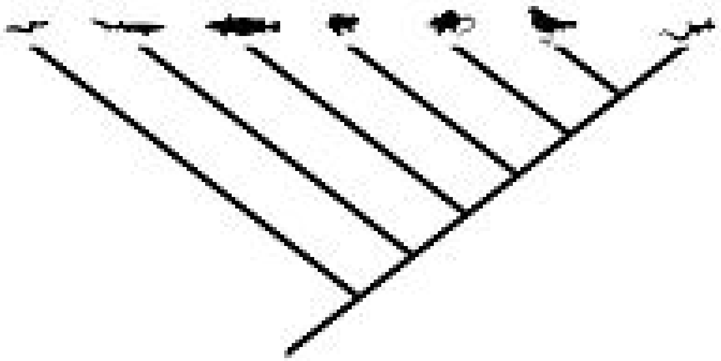
Newick format

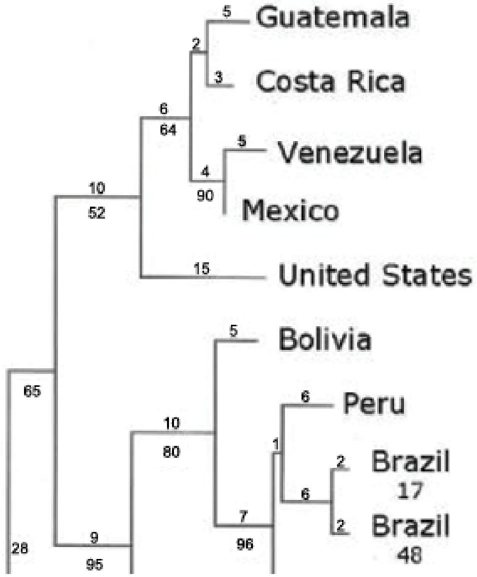


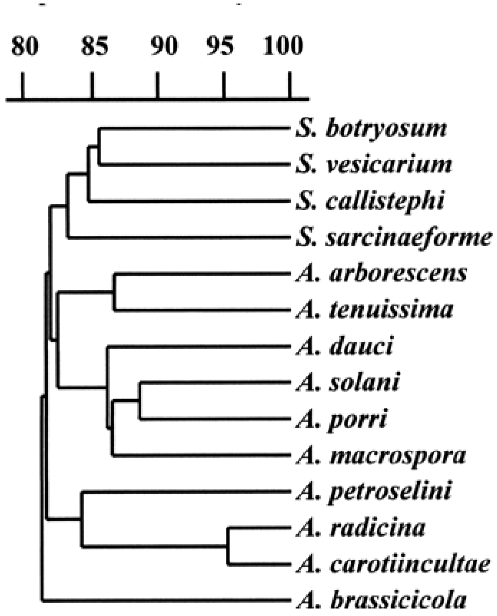
Newick format used by many computer programs:

(((F , G) , A) , (B , C) , D)

Different types of trees







Cladogram

Most simple tree
Just shows relative recency
Branch length has no meaning

Phylogram

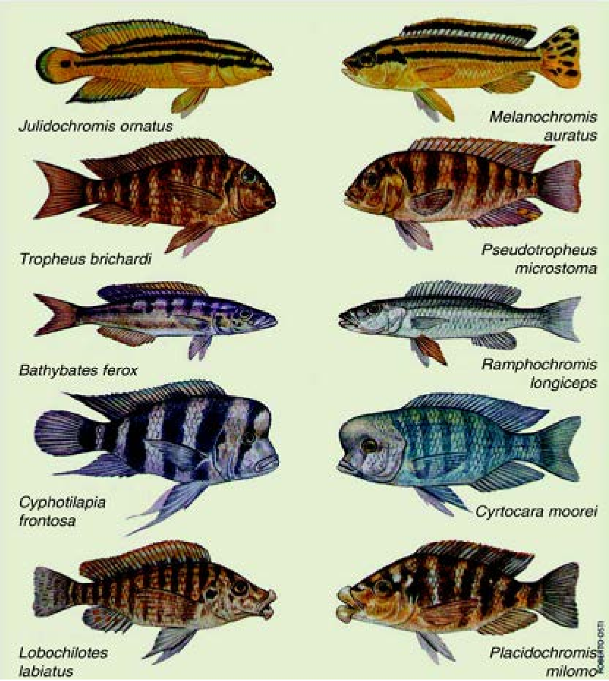
Additive tree
Branch length reflects number of changes

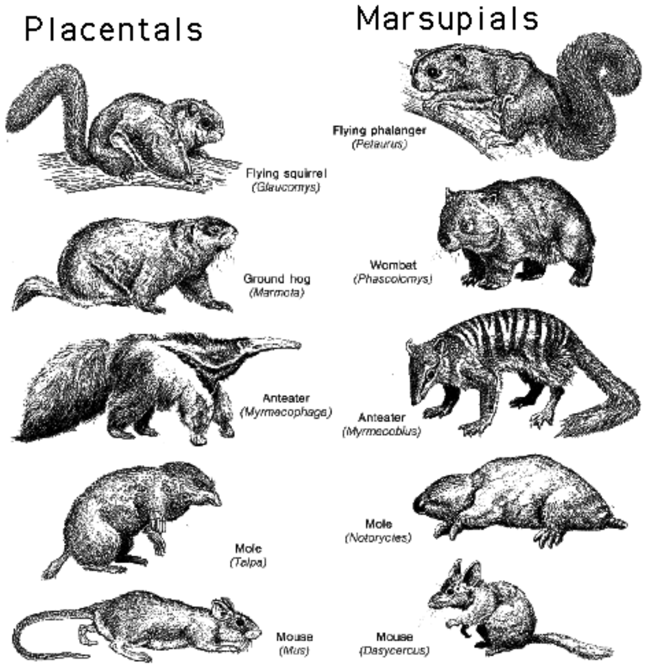
Dendrogram

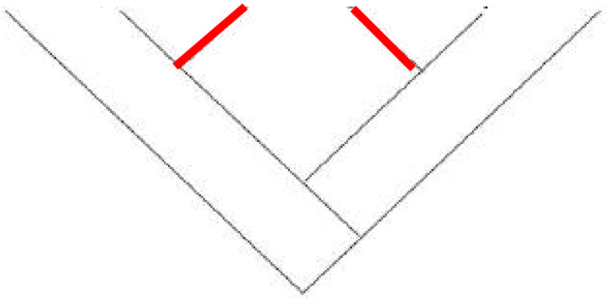
Ultrametric tree
Special form of a phylogram
All tips equal length from root
Axis = divergence time assuming molecular clock

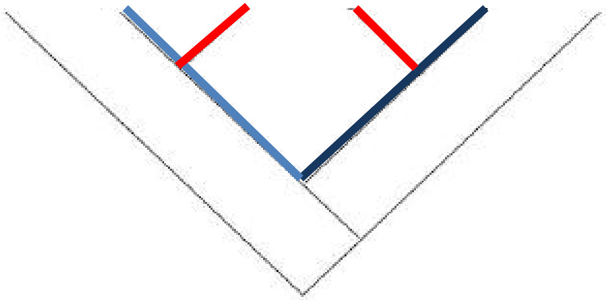
Homoplasy

Homoplasy: similarity acquired independently, i.e. not by descent









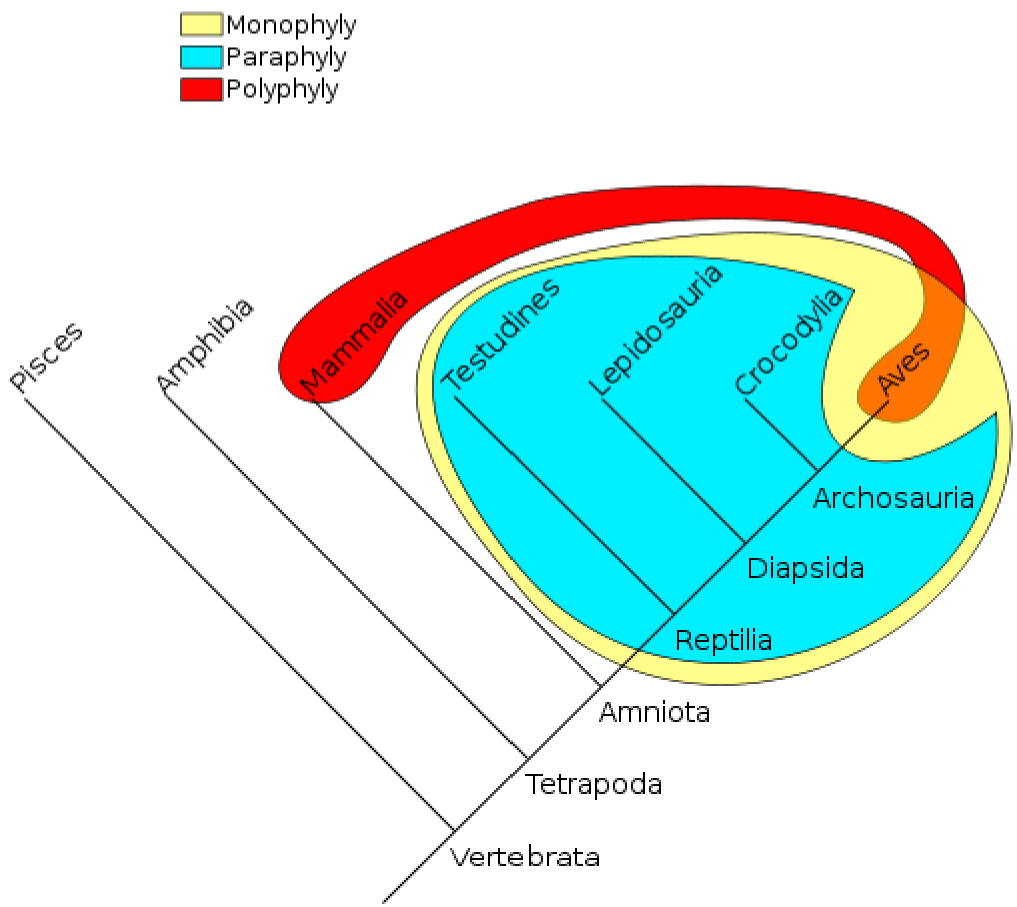
Parallel evolution

Same characters from the same ancestral condition
cichlids in Lake Malawi and Lake Tanganyika

Convergent evolution

Same characters from different ancestral condition

Phyla grouping



Monophyletic:

All birds and reptiles are believed to have descended from **a single common ancestor** (yellow).

Paraphyletic:

"Modern reptile" (cyan) is a grouping that contains a common ancestor, but **does not contain all descendants** of that ancestor (birds are excluded).

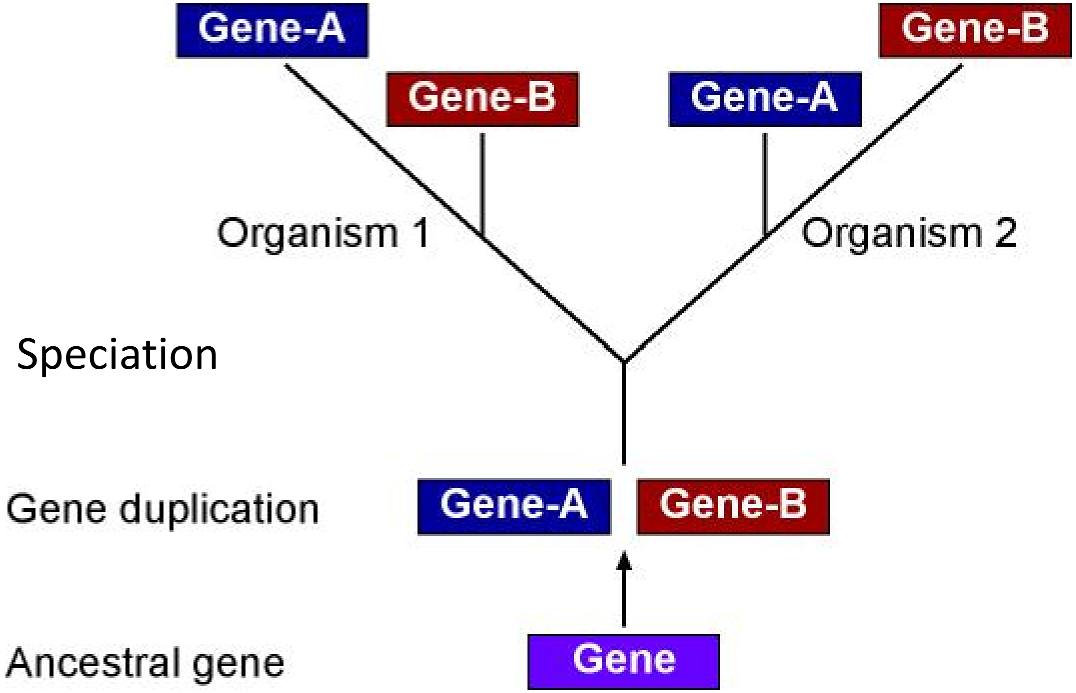
Polyphyletic:

A grouping such as warm-blooded animals would include only mammals and birds (red/orange); members of this grouping **do not include the most recent common ancestor**

Gene tree and species tree do not always agree

Homology:

1. **Orthologous genes**
   created by speciation event
2. **Paralogous genes**
   created by gene duplication

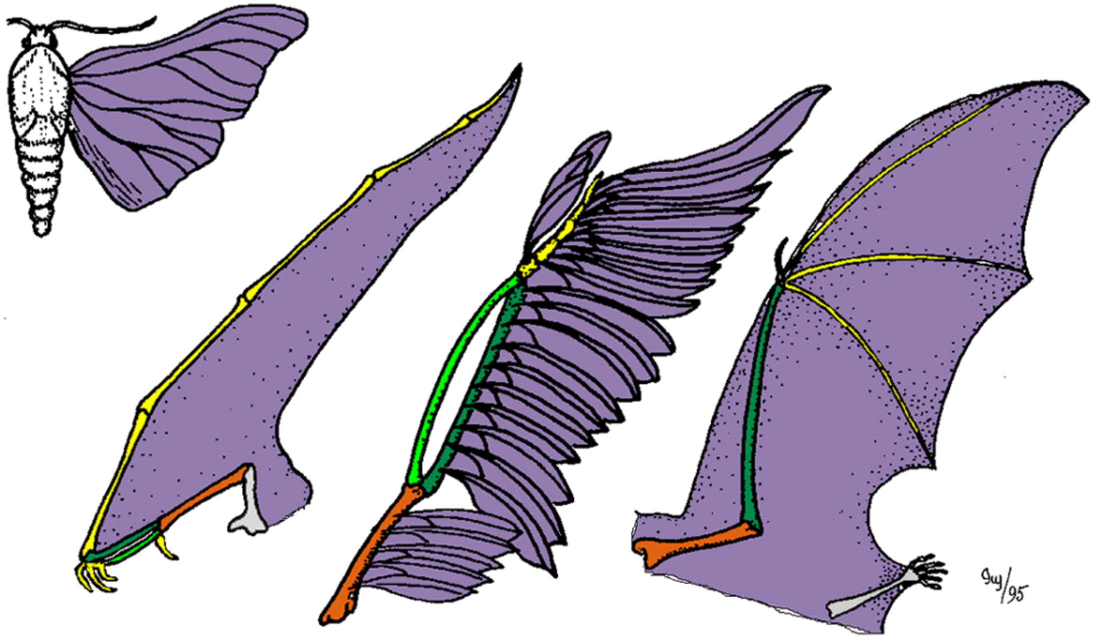


→ **To learn about species relationship use only orthologous genes**

Characters for trees

Traditionally, morphological characters have been used to infer phylogenies

For any phylogenetic analysis, always need to look at many characters:
For example, birds and bats have wings, while crocodiles and humans do not. If these were the only data available, we would tend to group crocodiles with humans, and birds with bats



Molecular phylogenetics uses DNA, protein sequence characters

Characters for trees

You can take any kind of character, as long as it has more than one state, e.g. eye color: blue, brown

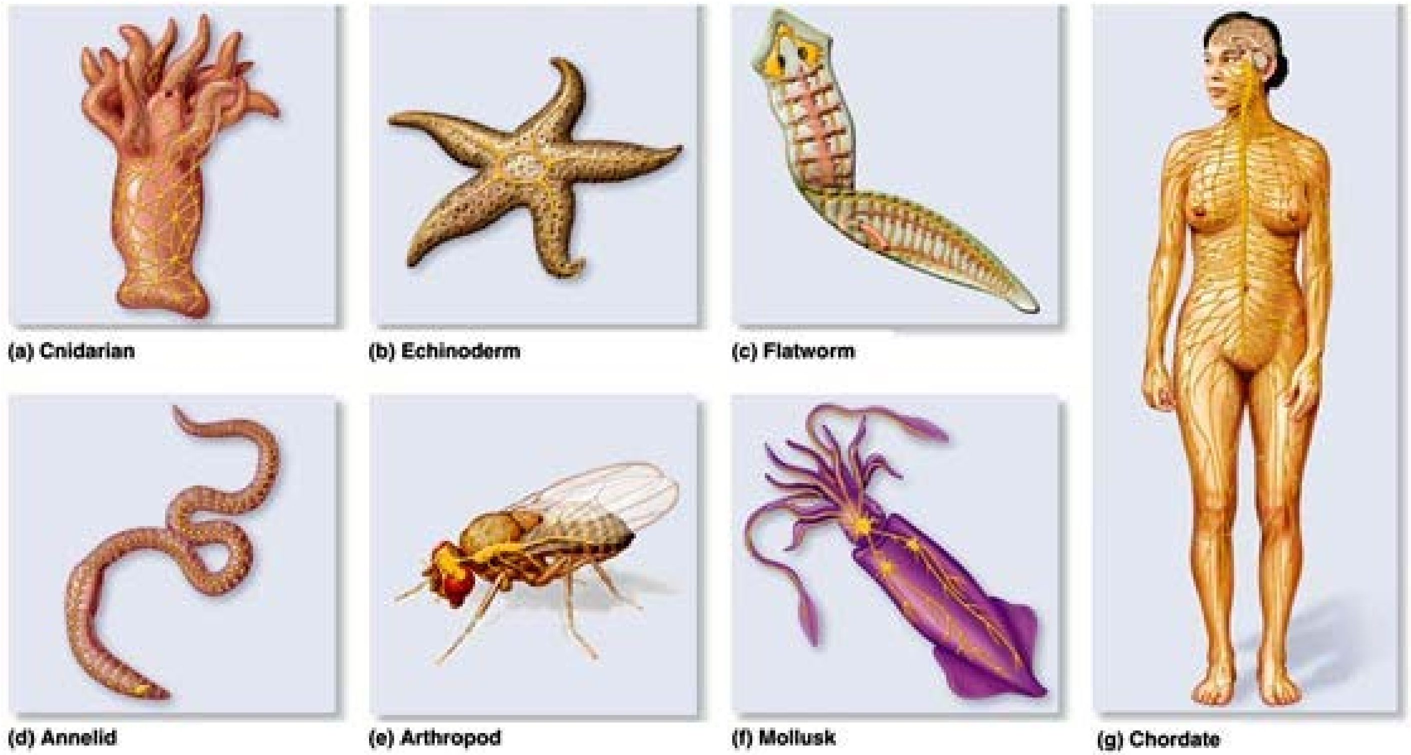
What about gray and green eyes? Make an extra attribute/character state or put them together with blue or brown depending on what is more similar

Character states for DNA: AGTC
for protein: A,C,G,K,L,H,...

How to treat missing data, e.g. an animal has no eyes, or the color is unknown for whatever reason?
Usually put a "?" or "-" or "X", or "N",
the latter things are common for sequence data

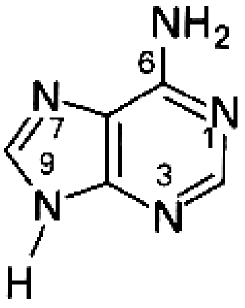
Characters states can be ordered

If going from one state to another requires more than one step:
e.g. cannot go directly from simple brain in cnidaria to complex brain in humans with a lot of different brain regions, gyri, sulci ; or dorsal vs ventral nervous system

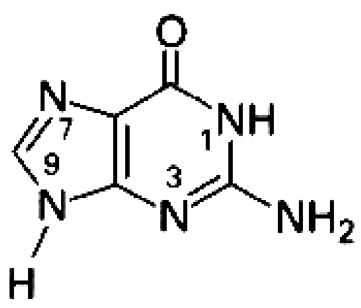


Characters states changes can be weighted

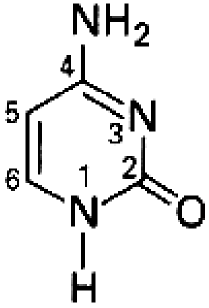
Give more weight to changes that are less common:
e.g. transversions (A-C, A-T, G-C, G-T changes) are less common than transitions
changes in 3rd codon position are more common than in 1st or 2nd (might be neutral)



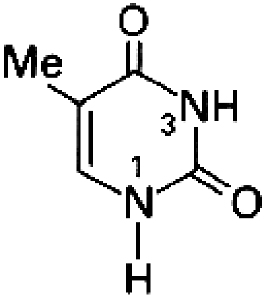
adenine



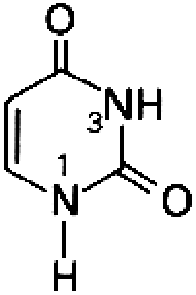
guanine



cytosine



thymine



uracil

Look at more than one character

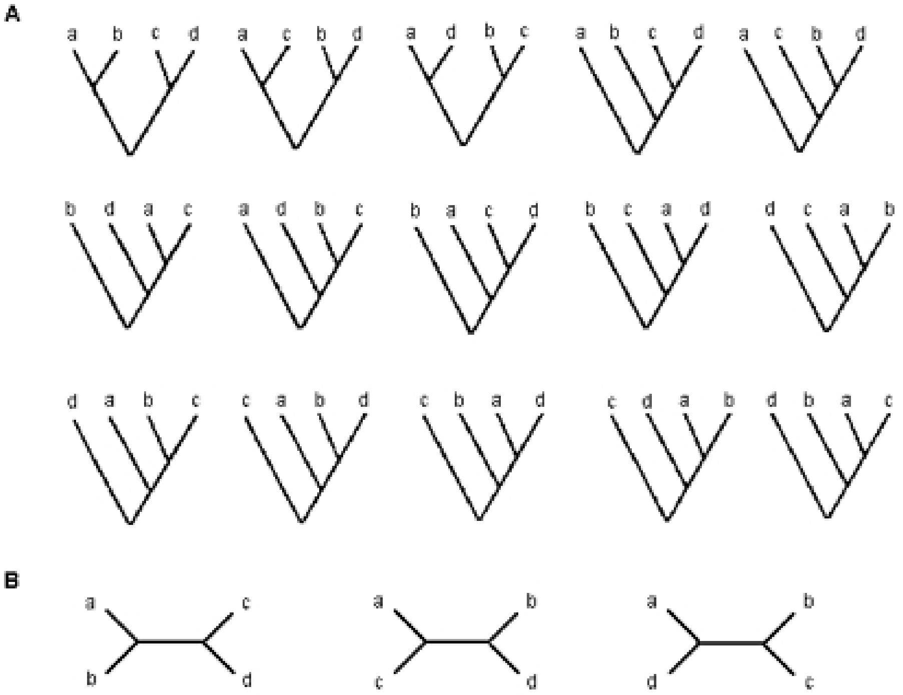
Idea: if two species share a trait that is not in a third, the two are more related to each other

Group dolphin and giraffe because they have a placenta
Or group fish and dolphin because they live in water (parallel evolution)
So depending on the character, we might end up with a different tree

→ Need to find the optimal tree taking into account all parameters

Number of possible trees can be enormous

All possible trees of 4 species
(rooted / unrooted)



Each taxon represents a new sample for every character, but, more importantly, it (usually) represents a new combination of character states
Ten species gives over two million possible unrooted trees
→ How to find the right tree?

It’s a NP-complete problem (NP= non-deterministic polynomial)
i.e. for any reasonable number of characters it is often impossible to find the optimal tree
→ We need some heuristics (“quick and dirty”)
→ Different methods might produce different tress

Tree building methods

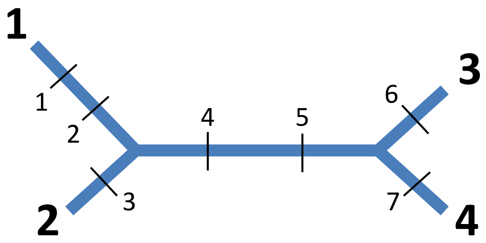
Sequences → Distances

sites

sequence	1	2	3	4	5	6	7
1	T	T	A	T	T	A	A
2	A	A	T	T	T	A	A
3	A	A	A	A	A	T	A
4	A	A	A	A	A	A	T

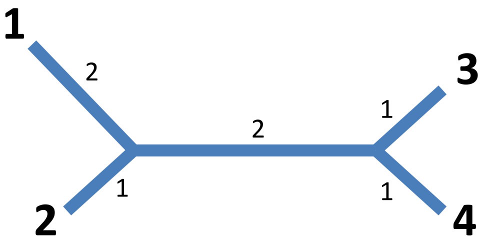
sequence	1	2	3
2	3		
3	5	4	
4	5	4	2

sequences



Discrete method
e.g. Parsimony tree

The 7 substitutions are placed on the 5 branches



Distance method
e.g. Neighbor-joining tree

The 7 substitutions are apportioned over the 5 branches

Sum of the branch length is the same in both trees: 7
But in the parsimony tree we see which site contributes to the length of each branch

Tree building methods

Clustering methods

Start with three species
Add the others in a step-wise fashion

Easy to implement
Fast, good for large datasets
Produce only one tree

Not possible to evaluate the resulting trees compared to alternatives

Information about which sites contribute to branch length missing

e.g. Neighbor-joining tree

Optimality criteria

Make all possible trees

Evaluate (score) trees to find the tree that best explains the data = the tree with the fewest number of evolutionary steps

Computationally intense

Provides information about which sites contribute to branch length missing

e.g. Maximum Parsimony tree
Maximum Likelihood

Neighbor-joining method

Clustering method
Distance method

Start by making a distance matrix:

Sequences → **Distances**

sites

sequence

	1	**2**	**3**	**4**	**5**	**6**	**7**
1	T	T	A	T	T	A	A
2	A	A	T	T	T	A	A
3	A	A	A	A	A	T	A
4	A	A	A	A	A	A	T

sequence

2	3		
3	5	4	
4	5	4	2
	1	**2**	**3**

sequences

Algorithm:

1. Based on the current distance matrix calculate the matrix Q (defined below).
2. Find the pair of taxa in Q with the lowest value. Create a node on the tree that joins these two taxa (i.e., join the closest neighbors, as the algorithm name implies).
3. Calculate the distance of each of the taxa in the pair to this new node.
4. Calculate the distance of all taxa outside of this pair to the new node.
5. Start the algorithm again, considering the pair of joined neighbors as a single taxon and using the distances calculated in the previous step.

Neighbor-joining method

1. Based on the current distance matrix calculate the matrix Q

Distance matrix

	A	B	C	D
A	0	7	11	14
B	7	0	6	9
C	11	6	0	7
D	14	9	7	0

→

Q matrix

	A	B	C	D
A	0	−40	−34	−34
B	−40	0	−34	−34
C	−34	−34	0	−40
D	−34	−34	−40	0

$$Q(i,j) = (r-2)d(i,j) - \sum_{k=1}^{r} d(i,k) - \sum_{k=1}^{r} d(j,k)$$

r= # taxa
i, j = taxa (A,B,C,D)
k = always the other taxa

$$\text{AB: } Q(A,B) = (4-2) * \overset{AB}{7} - (\overset{AB}{7}+\overset{AC}{11}+\overset{AD}{14}) - (\overset{AB}{7}+\overset{BC}{6}+\overset{BD}{9})$$

$$= 14 - 32 - 22$$

$$= -40$$

$$\text{AC: } Q(A,C) = (4-2) * \overset{AC}{11} - (\overset{AB}{7}+\overset{AC}{11}+\overset{AD}{14}) - (\overset{AC}{11}+\overset{BC}{6}+\overset{DC}{7})$$

$$= 22 - 32 - 24$$

$$= -34$$

…

Neighbor-joining method

2. Find the pair of taxa in Q with the lowest value. Create a node on the tree that joins these two taxa (i.e., join the closest neighbors, as the algorithm name implies).

Q matrix

	A	**B**	**C**	**D**
A	0	−40	−34	−34
B	−40	0	−34	−34
C	−34	−34	0	−40
D	−34	−34	−40	0

Join taxa A and B to a new node: u

Neighbor-joining method

3. Calculate the distance of each of the taxa in the pair to this new node.

Distance of A and B to the new node:

Distance matrix

	A	**B**	**C**	**D**
A	0	7	11	14
B	7	0	6	9
C	11	6	0	7
D	14	9	7	0

u = the new node
f,g = the joined taxa (A,B)

$$d(f,u) = \frac{1}{2}d(f,g) + \frac{1}{2(r-2)}\left[\sum_{k=1}^{r} d(f,k) - \sum_{k=1}^{r} d(g,k)\right]$$

$$\begin{aligned} \text{A: } d(A,u) &= ½*\overset{AB}{7} + 1/2(4-2) * [(\overset{AB}{7}+\overset{AC}{11}+\overset{AD}{14}) – (\overset{AB}{7}+\overset{BC}{6}+\overset{BD}{9})] \\ &= 3.5 + ¼ * [\ 32 – 22\ ] \\ &= 3.5 + 2.5 \\ &= 6 \end{aligned}$$

$$d(g,u) = d(f,g) - d(f,u)$$

B: d(B,u) = 7 – 6 =1

Neighbor-joining method

4. Calculate the distance of all taxa outside of this pair to the new node.

Distance of C and D to the new node:

$$d(u,k) = \frac{1}{2}[d(f,k) + d(g,k) - d(f,g)]$$

C: d(u,C) = ½ * [11 + 6 – 7]
= ½ * 10
= 5

D: d(u,D) = ½ * [14 + 9 -7]
= ½ * 16
= 8

u = the new node
k = taxa to calculate distance for (C,D)

Distance matrix

	A	**B**	**C**	**D**
A	0	7	11	14
B	7	0	6	9
C	11	6	0	7
D	14	9	7	0

New Distance matrix

	AB	**C**	**D**
AB	0	5	8
C	5	0	7
D	8	7	0

Neighbor-joining method

5. Start the algorithm again, considering the pair of joined neighbors as a single taxon and using the distances calculated in the previous step.

New Distance matrix

	AB	C	D
AB	0	5	8
C	5	0	7
D	8	7	0

→

New Q matrix

	AB	C	D
AB	0	-20	-20
C	-20	0	-20
D	-20	-20	0

$$Q(i,j) = (r-2)d(i,j) - \sum_{k=1}^{r} d(i,k) - \sum_{k=1}^{r} d(j,k)$$

r= # taxa: now 3

i,j = taxa (AB,C,D)

AB-C: $Q(AB,C) = (3-2) * \overset{AB-C}{5} - (\overset{AB-C}{5} + \overset{AB-D}{8}) - (\overset{AB-C}{5}+\overset{CD}{7})$

$= 5 - 13 - 12$

$= -20$

AB-D: $Q(AB,D) = (3-2)*\overset{AB-D}{8} - (\overset{AB-C}{5} + \overset{AB-D}{8}) - (\overset{AB-D}{8} + \overset{CD}{7})$

$= 8 - 13 - 15$

$= -20$

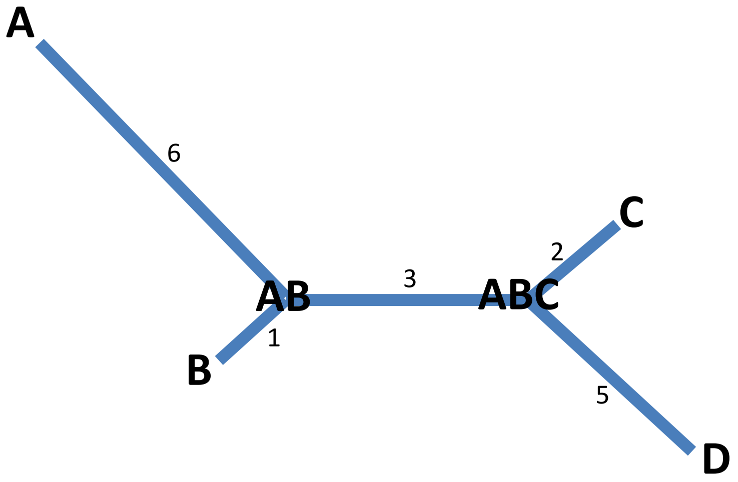
CD: …

Neighbor-joining method

Join C to AB

New Distance matrix

	ABC	**D**
ABC	0	5
D	5	0



Distance A to AB was 6
Distance B to AB was 1
Distance AB to ABC is 3
Distance C to ABC is 2
Distance D to ABC is 5

Algorithm produces tree with the shortest total branch length
unrooted, but outgroup can be added
root = where the edge of the outgroup meets the tree
no molecular clock assumed
uses a greedy algorithm: solve one problem at the time, then the next problems (step-wise adding of the next nodes)
it's computationally efficient but might not find the optimal tree (but often it finds the optimal tree or something close)

Tree building methods

Clustering methods

Start with three species
Add the others in a step-wise fashion

Easy to implement
Fast, good for large datasets
Produce only one tree

Not possible to evaluate the resulting trees compared to alternatives

Information about which sites contribute to branch length missing

e.g. Neighbor-joining tree

Optimality criteria

Make all possible trees

Evaluate (score) trees to find the tree that best explains the data = the tree with the fewest number of evolutionary steps

Computationally intense

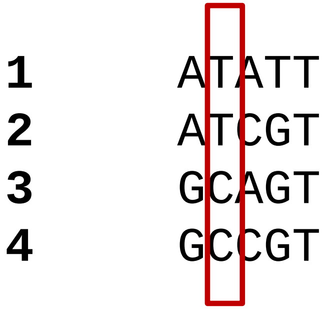
Provides information about which sites contribute to branch length missing

e.g. Maximum Parsimony tree
Maximum Likelihood

Maximum parsimony

- **Small parsimony problem**

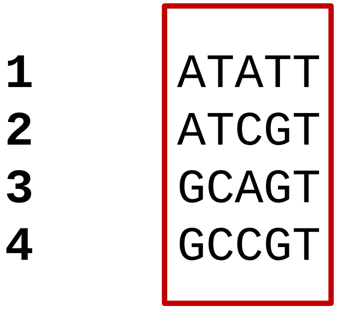
Tree is given
Only one character is analyzed
Goal: find out what the ancestor
at each node was



1	ATATT
2	ATCGT
3	GCAGT
4	GCCGT

- **Large parsimony problem**

Tree is unknown
Multiple characters
Goal: find the tree and ancestors



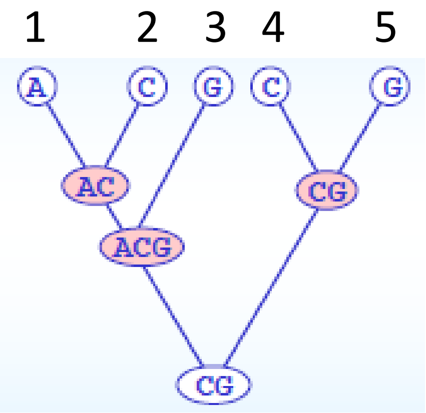
1	ATATT
2	ATCGT
3	GCAGT
4	GCCGT

Small parsimony problem

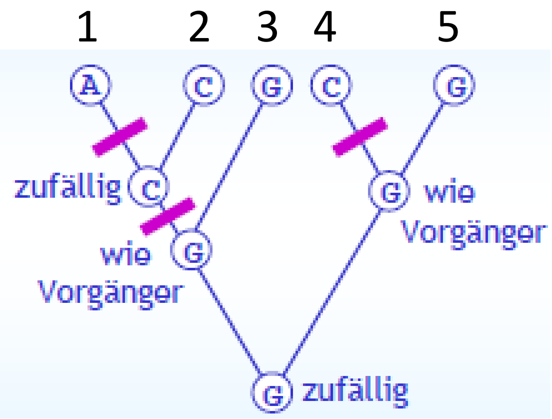
Fitch algorithm:
Calculate the minimal number of mutations that explains the evolution given a particular tree
Because every character is independent, this can be calculated for each character

1	AAATT
2	ACCGT
3	GGAGT
4	GCCGT
5	AGATT

1. Start with leaves and go to root
Assign to each node v a letter of the alphabet:
if v is a leaf $S(v)=\{v_c\}$; v_c = character of node v
if v is an intermediate node of nodes u and w
if $S(u) \cap S(w) \neq \{\}$: $S(v) = S(u) \cap S(w)$
if $S(u) \cap S(w) = \{\}$: $S(v) = S(u) \cup S(w)$



2. Start at root and go to leaves to solve intermediate nodes
Assign to each node (except leaves) a letter of the alphabet
if v follows u and $u_c \in S(v)$: $v_c = u_c$
otherwise v_c = random $\in S(v)$



3 assignments were random → loss of information = 3

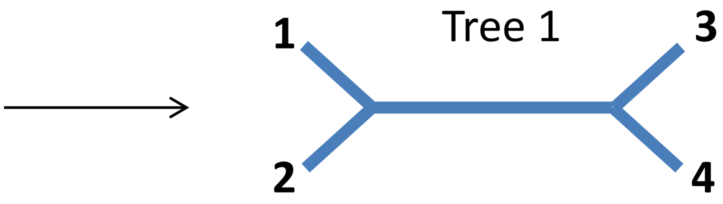
Large parsimony problem

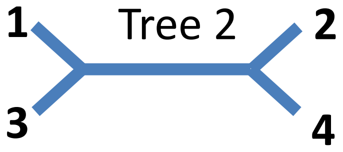
- **For a multiple characters (e.g. the whole sequence)**

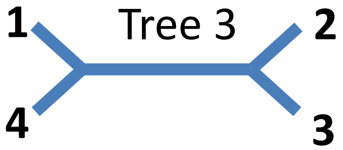
Method:

1. Make all possible trees
2. Score them by the total number of character state changes required (= "evolutionary steps") to explain distribution of each character
3. Pick the tree that infers the least steps/number of changes = the most parsimonious

1	ATATT
2	ATCGT
3	GCAGT
4	GCCGT

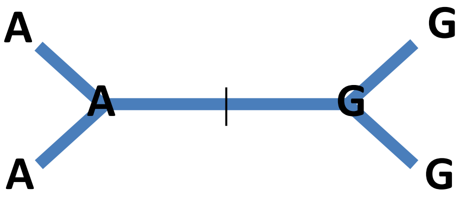


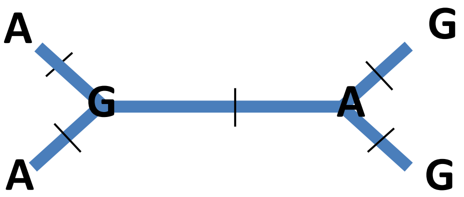




For the first site, Tree 1:

(two examples of what the ancestral version (internal node) could have looked like)





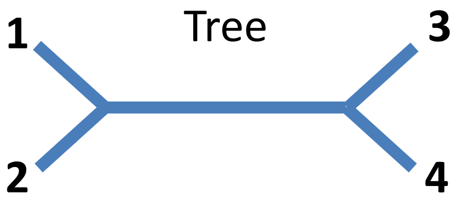
Only one change required
= more parsimonious

Five change required

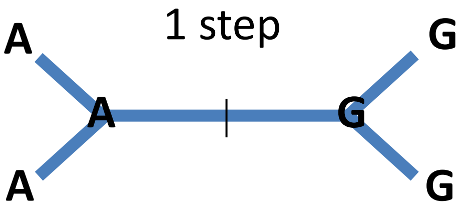
Large parsimony problem

1	ATATT
2	ATCGT
3	GCAGT
4	GCCGT

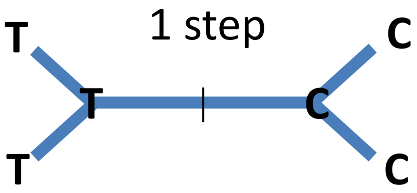
→

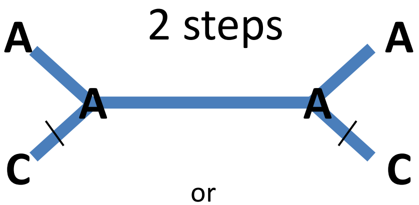


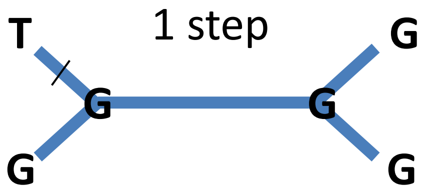
For the first site:

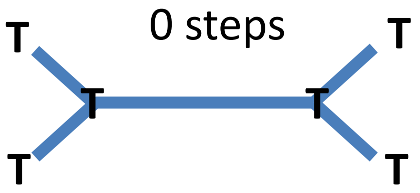


For the other sites:

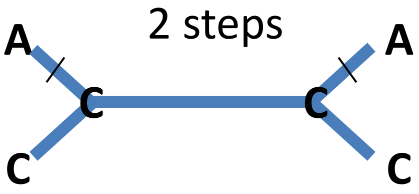








or



Not informative for phylogeny because all sites are the same

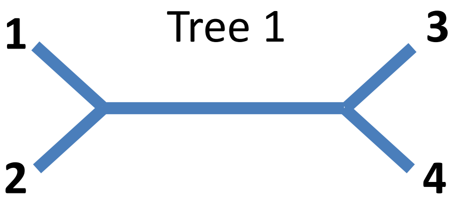
$$L = \sum_{i=1}^{k} l_i$$

Total length of the tree L = total # evolutionary changes/steps = sum of length l of each site k; here: 1 + 1 + 2 + 1 + 0 = 5

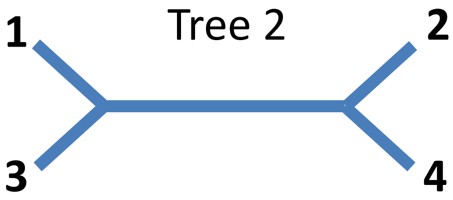
Large parsimony problem

1	ATATT
2	ATCGT
3	GCAGT
4	GCCGT

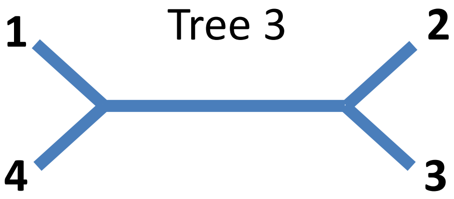
Alternative trees:


Tree 1

1 + 1 + 2 + 1 + 0 = 5


Tree 2

2 + 2 + 1 + 1 + 0 = 6


Tree 3

2 + 2 + 2 + 1 + 0 = 7

Tree with shortest total branch length
= most parsimonious tree

Large parsimony problem

After scoring all trees find the most-parsimonious trees (MPTs)

often exist a number of equally MPTs
if too many, maybe because of too many missing data; insufficient data to resolve the tree completely
often only parts of the tree (sub trees) differ between the MPTs, e.g. a few taxa jump around
can make a consensus tree

Large parsimony problem

Finding the optimal tree (the tree with the best score) is computationally expensive
→ a possibility is to start with one good tree, perturb it and see if the score gets better
→ Or make a tree for each character and then a consensus tree

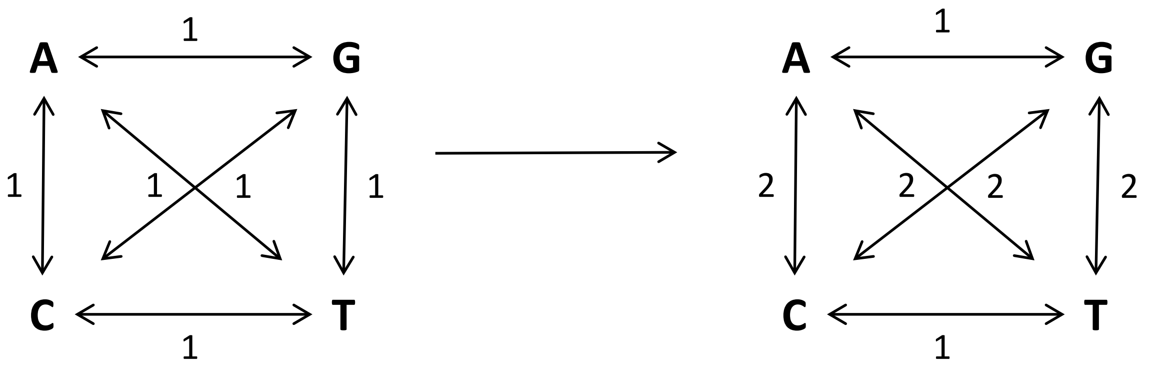
Trees are unrooted, but you can chose an outgroup

Trees do not reflect divergence times

Maximum parsimony

Are all changes equally likely?

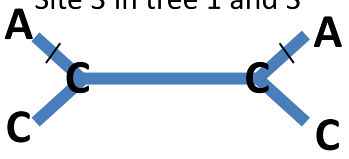
e.g. transversions are rarer than transitions, i.e. less likely
→ Assign higher costs to transversions; e.g. 1 transversion counts as 2 steps



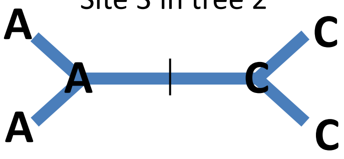
	A	C	G	T
A	0	2	1	2
C	2	0	2	1
G	1	2	0	2
T	2	1	2	0

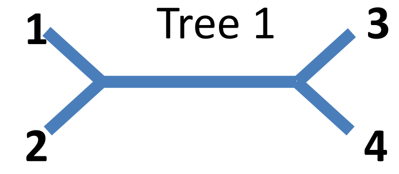
1 ATATT
2 ATCGT
3 GCAGT
4 GCCGT

Site 3 in tree 1 and 3

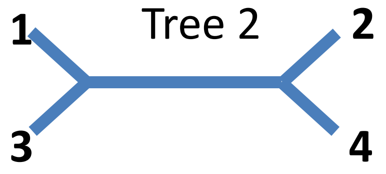


Site 3 in tree 2

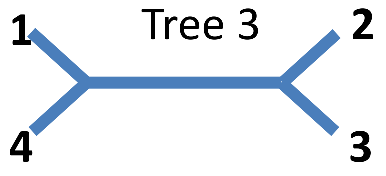




1 + 1 + 4 + 1 + 0 = 7



2 + 2 + 2 + 1 + 0 = 7



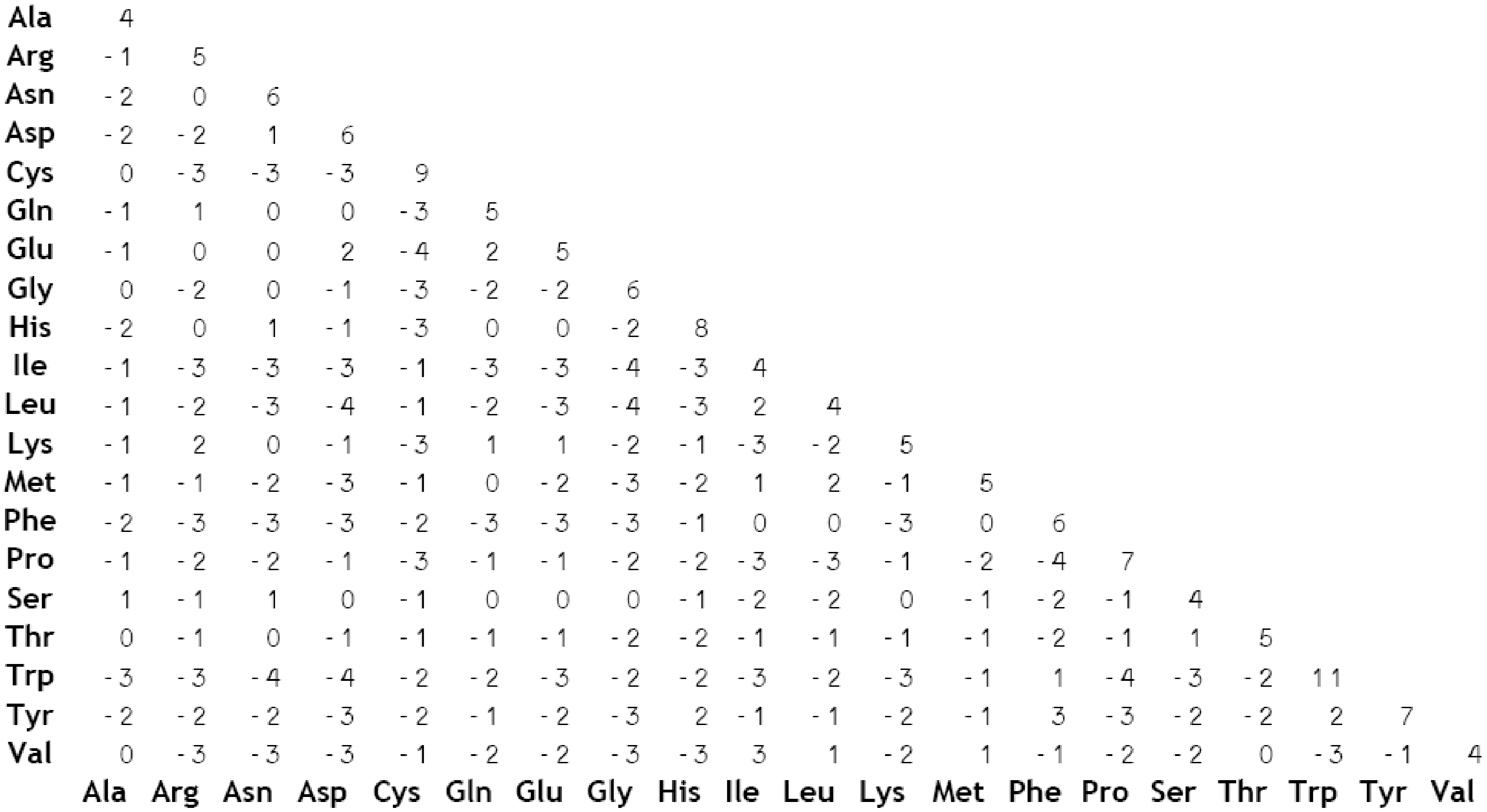
2 + 2 + 4 + 1 + 0 = 9

→ Now tree 1 and 2 are equally parsimonious

Maximum parsimony

Are all changes equally likely?

Substitution matrices for protein alignments: PAM, BLOSUM



Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4
	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val

Maximum parsimony

Are all changes equally likely?

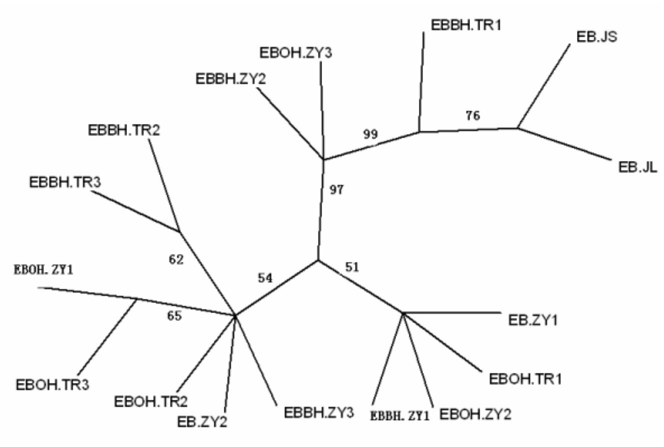
Some sites might be highly conserved (e.g. functional domains of a protein) while others might be rapidly changing (e.g. intronic sites)
Rapidly changing sites might be saturated and therefore misleading for the tree

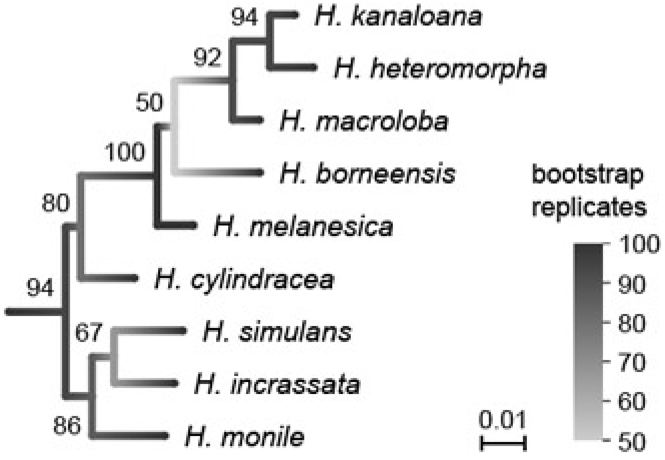
→ Give sites different weights
This will affect the total length of the tree: $L = \sum_{i=1}^{k} l_i \longrightarrow L = \sum_{i=1}^{k} w_i * l_i$

How well supported is the tree?

Bootstrap method:

Randomly pick characters from your dataset (e.g. columns in the alignment)
Make a random dataset that has the same size as your real dataset
Characters are picked with replacement
Do this multiple times, typically 1000 times
How often do you find a certain branch among the random trees?
More suitable for Neighbor-joining than Maximum Parsimony, because computationally intense





Commonly used phylogenetic programs

Phylip (**PHYL**ogeny **I**nference **P**ackage)
PAUP (**P**hylogenetic **A**nalysis **U**sing **P**arsimony)
Clustal