Introduction to Bioinformatics for Computer Scientists

Lecture 5

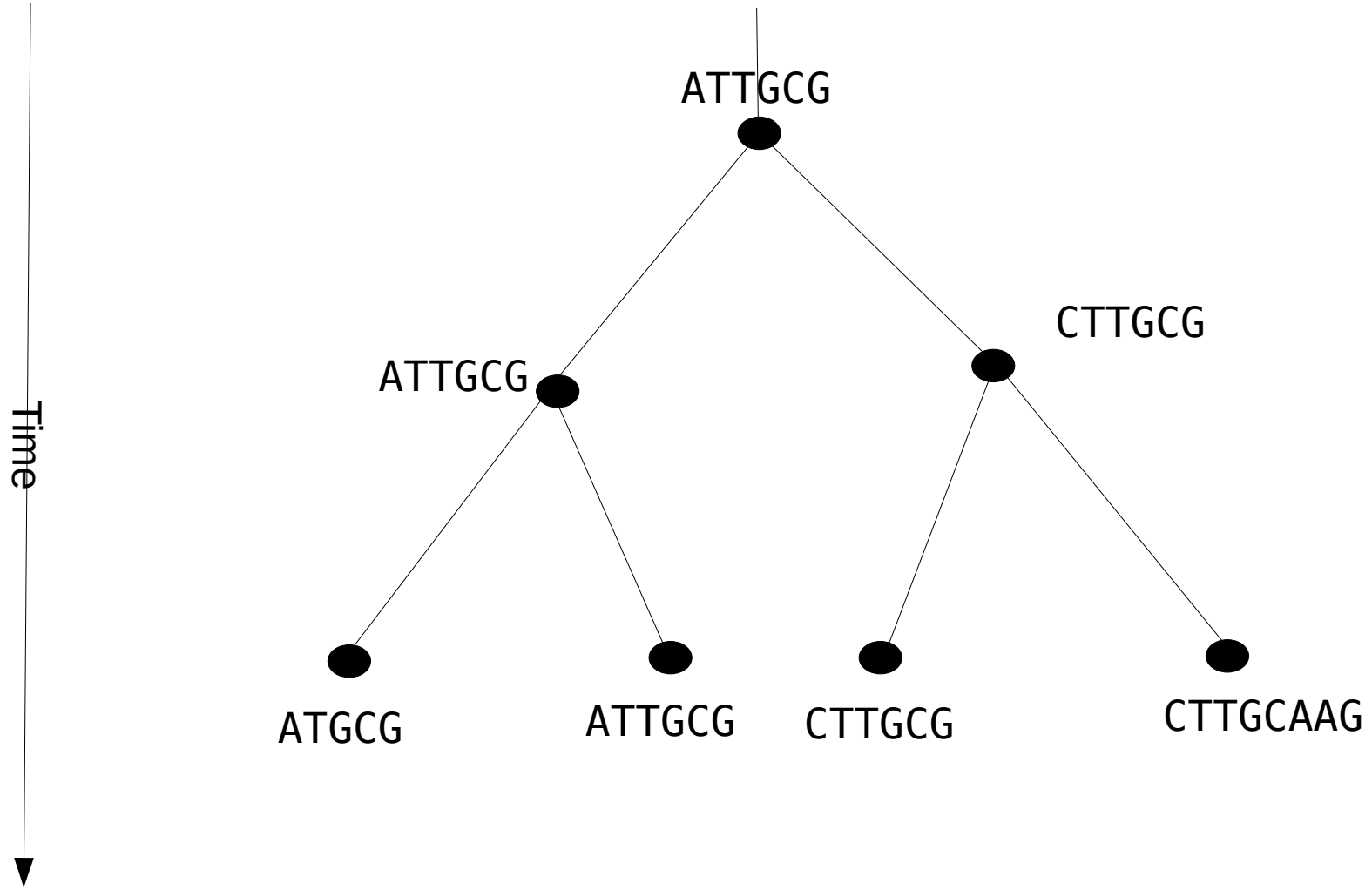
Plan for next lectures

- Today:
  - Multiple Sequence Alignment
  - Introduction to phylogenetics
- Next time:
  - Introduction to phylogenetics (continued)
  - Phylogenetic search algorithms

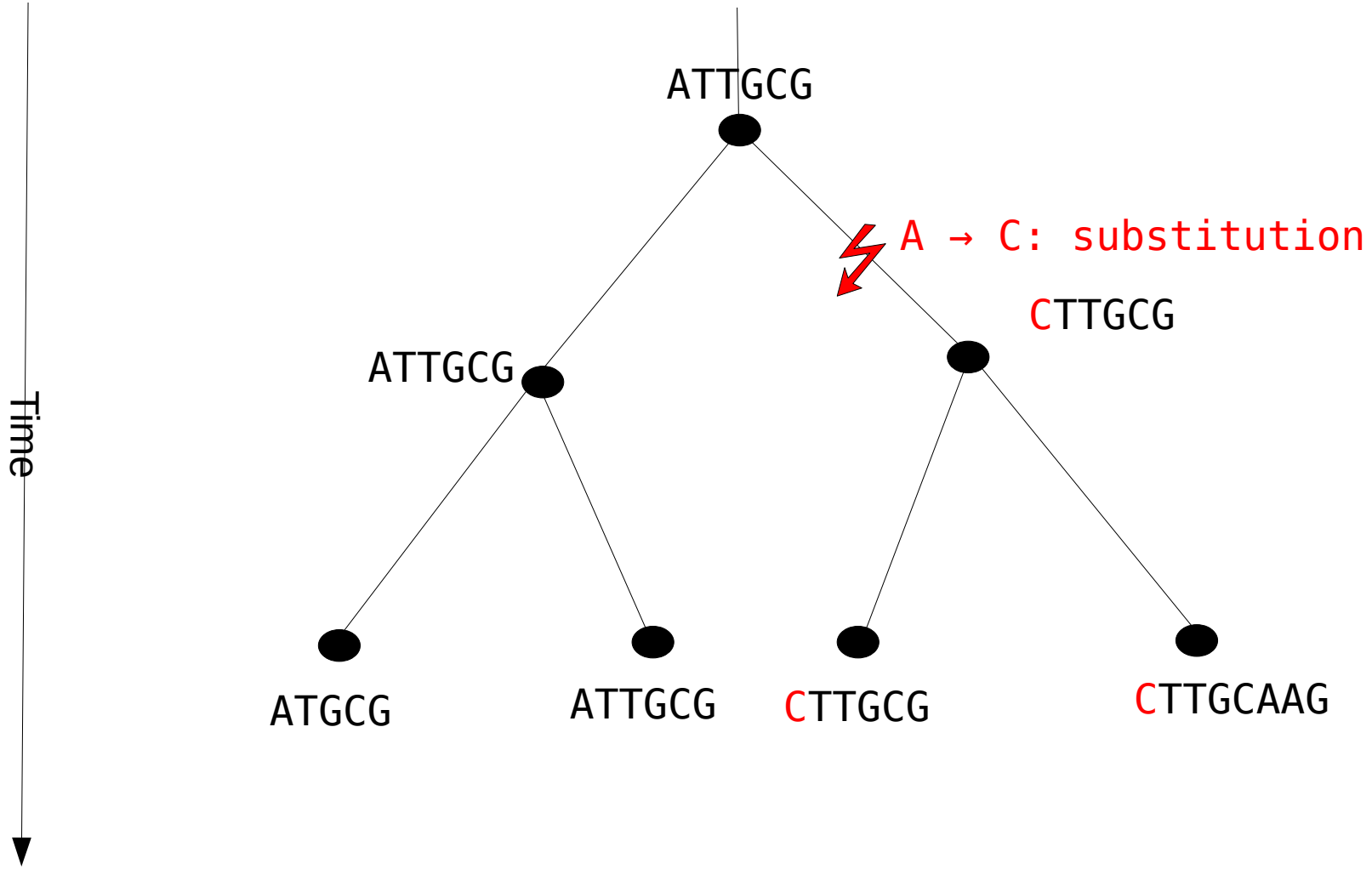
Multiple Sequence Alignment

- What are we trying to reconstruct?

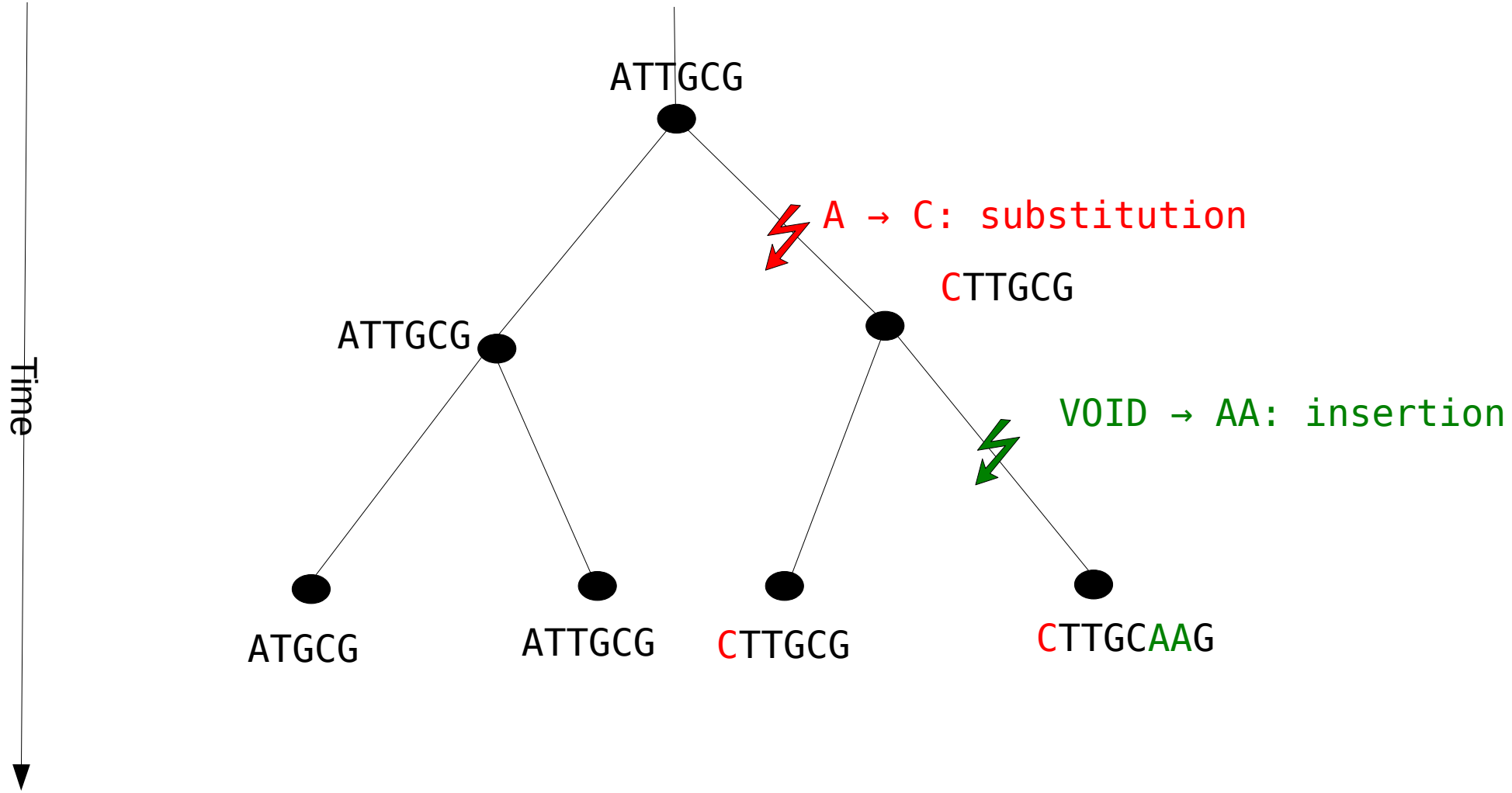
Insertions, Deletions & Substitutions



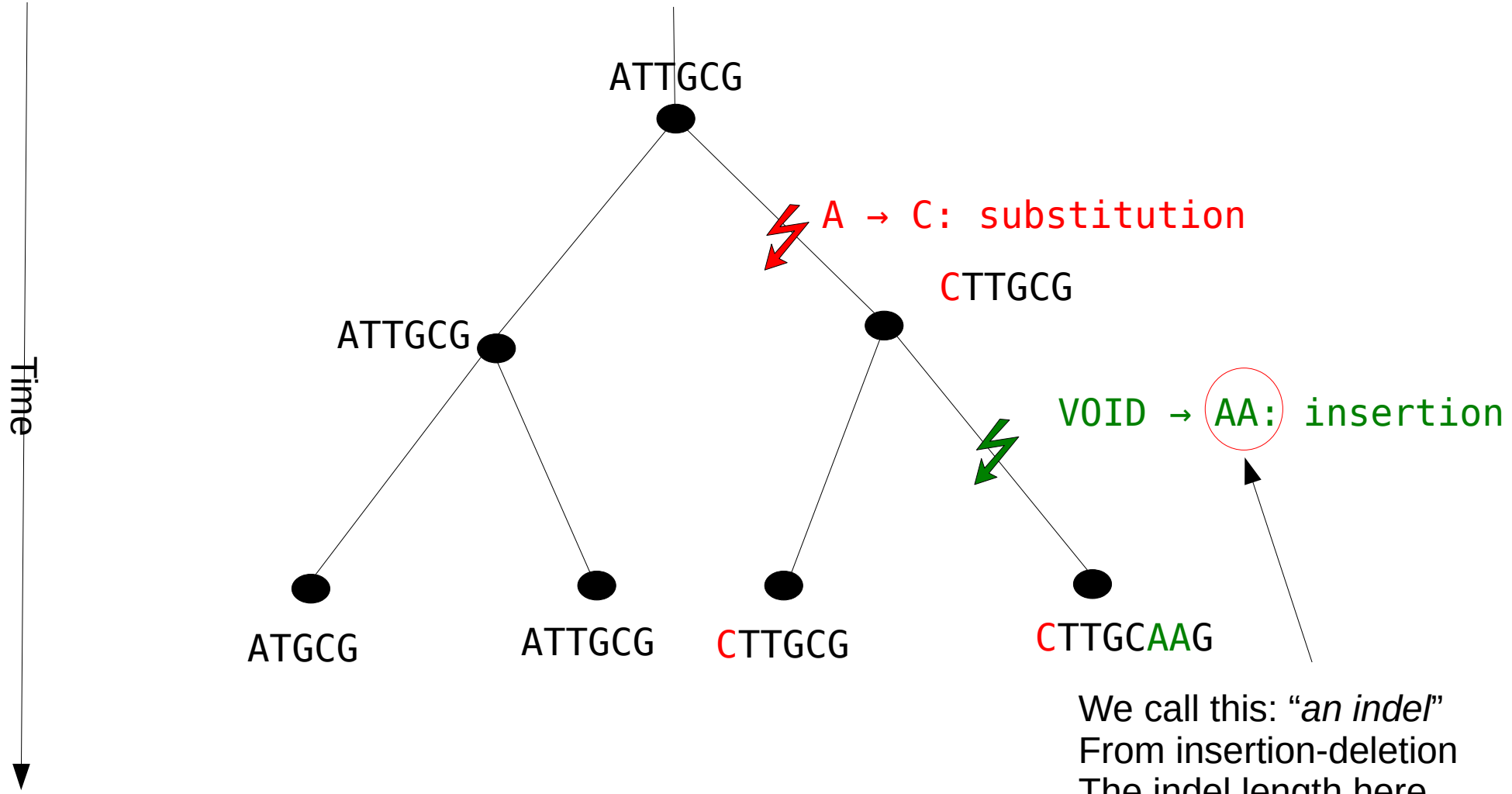
Insertions, Deletions & Substitutions



Insertions, Deletions & Substitutions



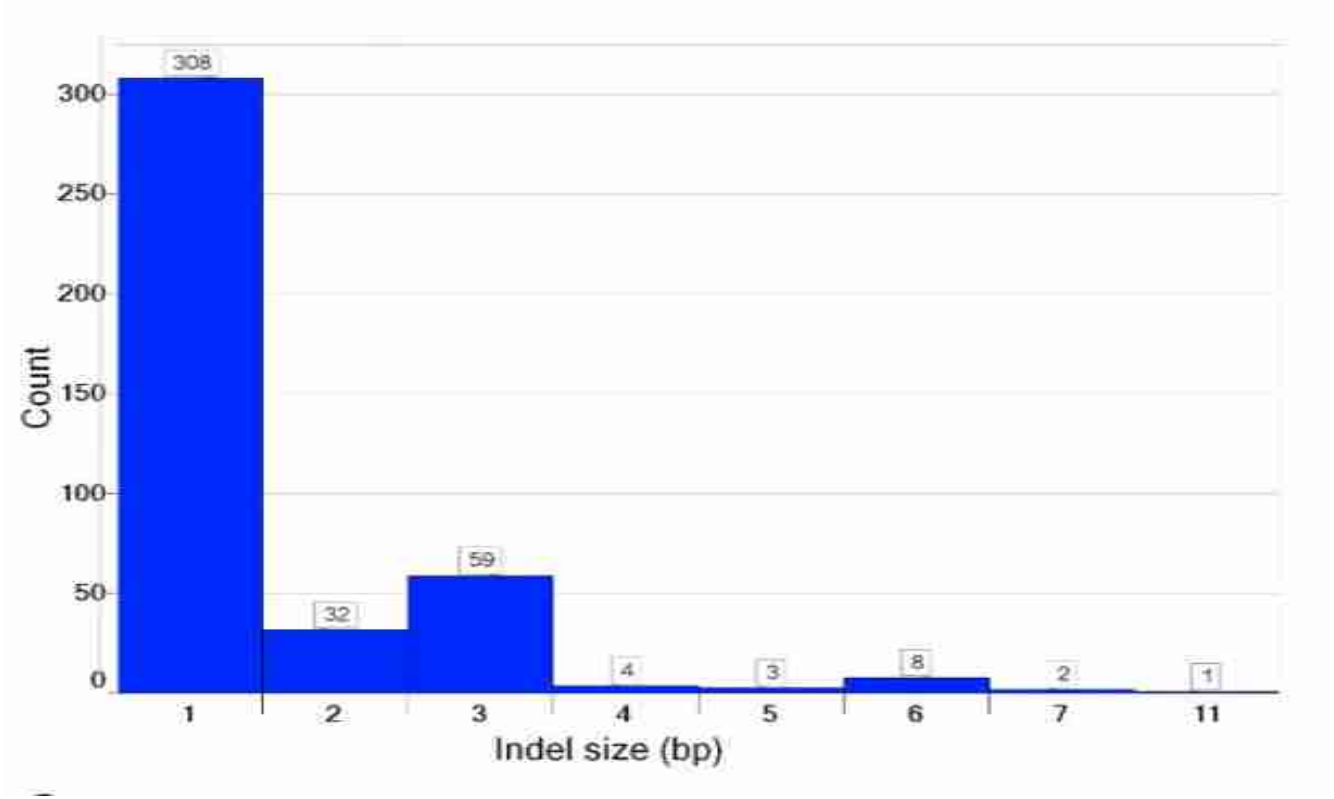
Insertions, Deletions & Substitutions



We call this: "*an indel*"
From insertion-deletion
The indel length here
is *2*, longer indel
lengths are not uncommon!

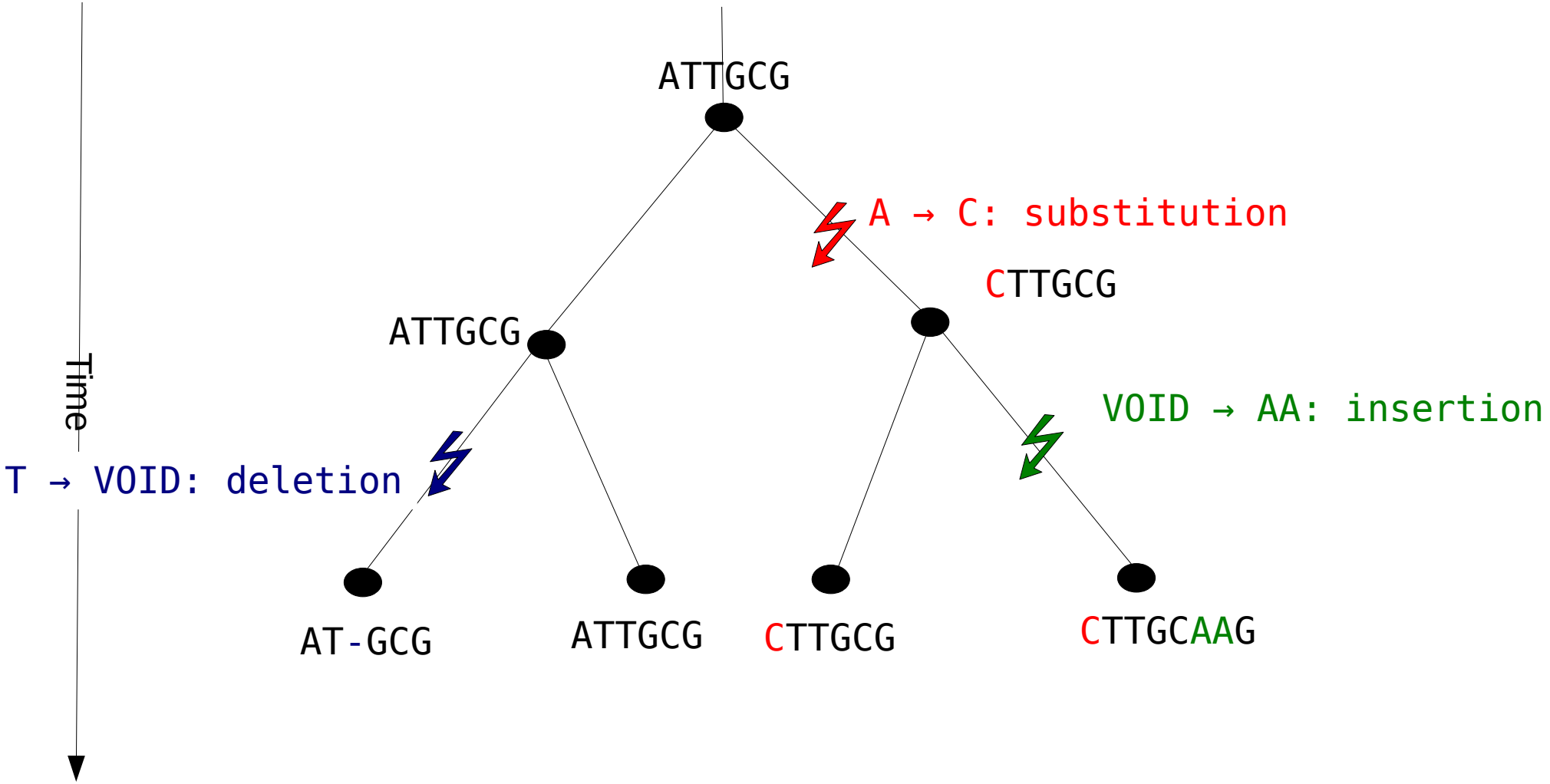
Indel size distribution

- Why are indels of size *3* rather frequent?

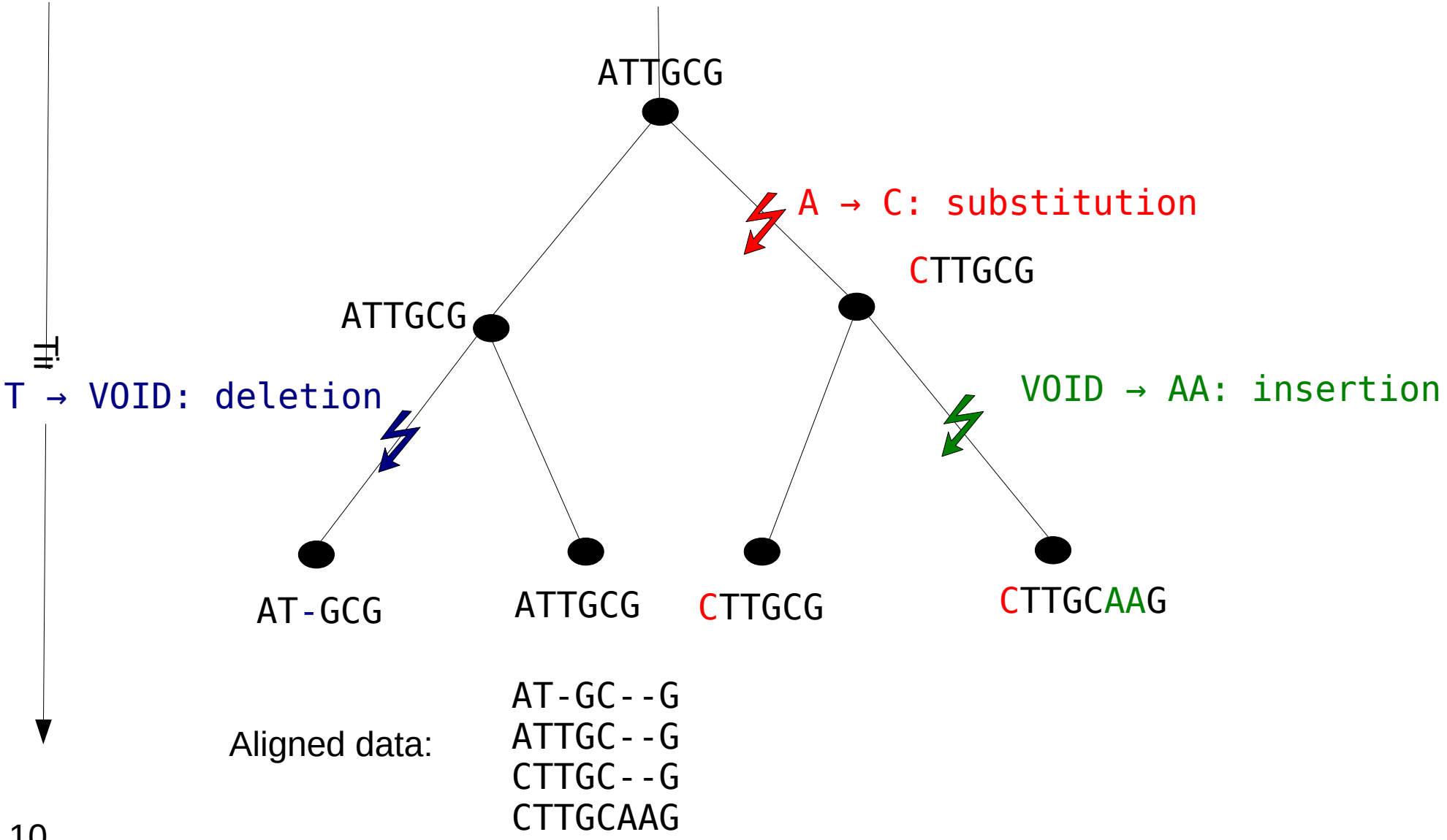


Indel size distribution in *coding regions* of *cattle genomes*

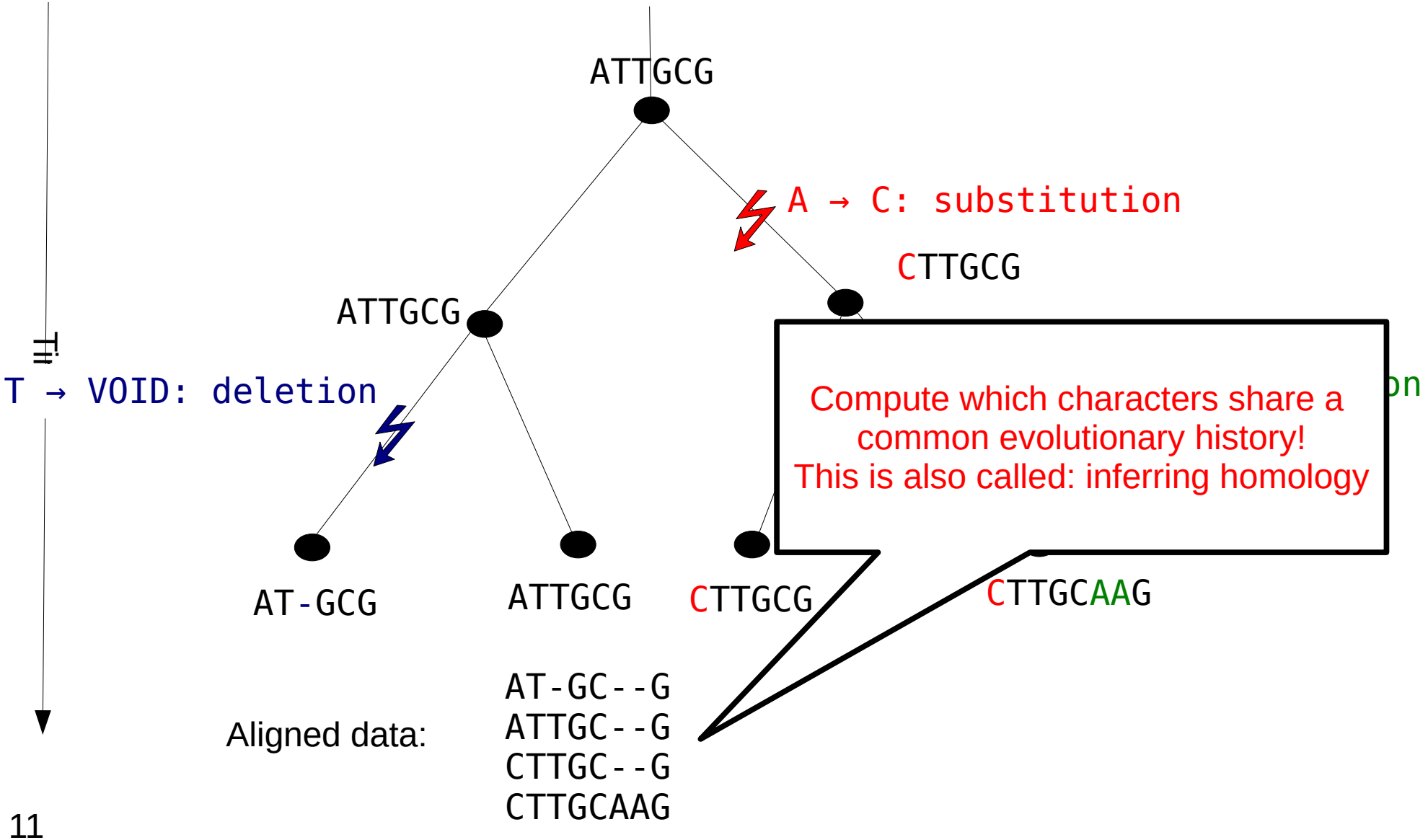
Insertions, Deletions & Substitutions



Insertions, Deletions & Substitutions



Insertions, Deletions & Substitutions



Multiple Sequence Alignment

- So far
  - Comparing two sequences (Lukas' lecture)
  - Mapping a sequence/read to a reference genome (Alexey's lecture)
- What do we do when we want to compare more than two sequences at a time?

  → Multiple Sequence Alignment (MSA)
- Open question: How do we assess the quality/accuracy of MSA algorithms?

  → nice review paper: "Who watches the watchmen?" http://arxiv.org/abs/1211.2160

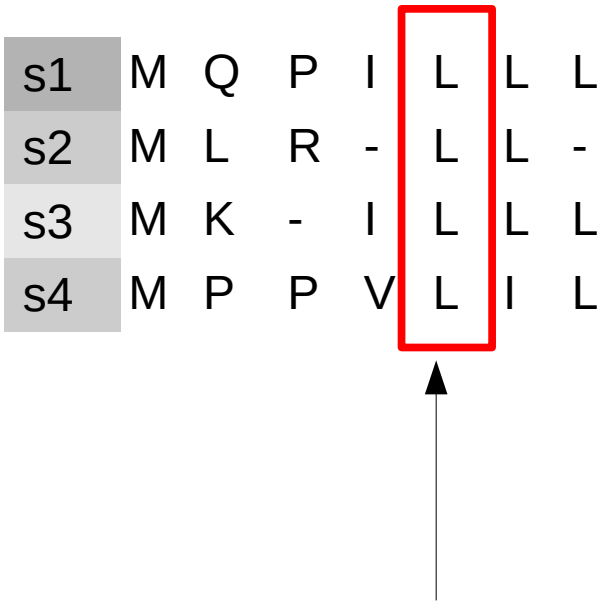
Why do we need MSAs?

- Input for phylogenetic reconstruction
- Discover important (e.g., conserved) parts of a *protein family*
- *Protein family* → group of evolutionary related genes/proteins in different species with similar function/structure
- ***Family*** has a different meaning than in taxonomy!

MSA

- Generalization of pair-wise sequence alignment problem
- Given *n* orthologous sequences $s_1,\ldots,s_n$ of different lengths, insert gaps "-" such that:
  - All sequences have the same length
  - Some criterion is optimized
  - *Corresponding (homologous) characters* in s_i and s_j are aligned to each other (in the same alignment column/site)
  - Columns/sites that entirely consist of gaps are **not** allowed

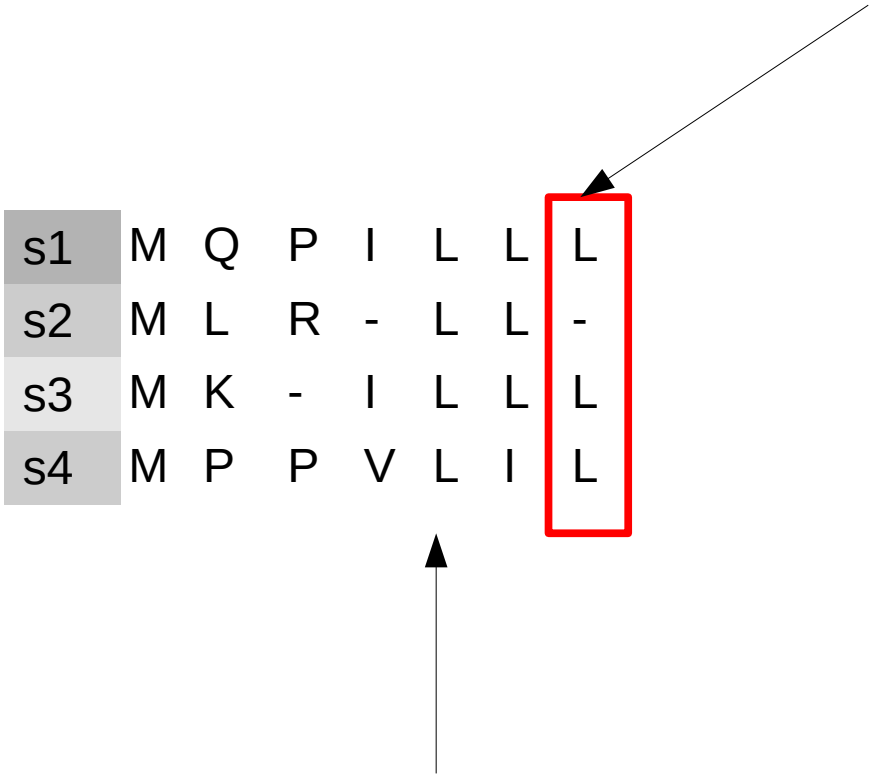
MSA Terminology



Orthologous sequences:
Sequences in different species that have evolved from the same **ancestral** gene

→ sequences that share a common evolutionary history

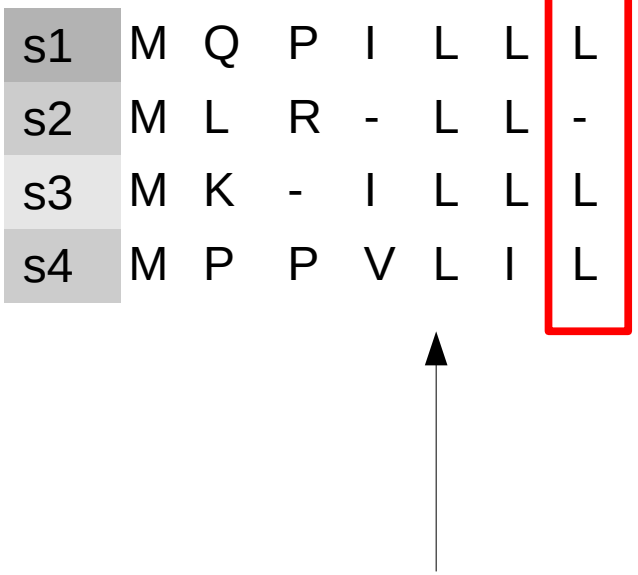
MSA Terminology



Homologous characters:
Characters that share a common evolutionary history

Alignment site/Alignment column

MSA Terminology



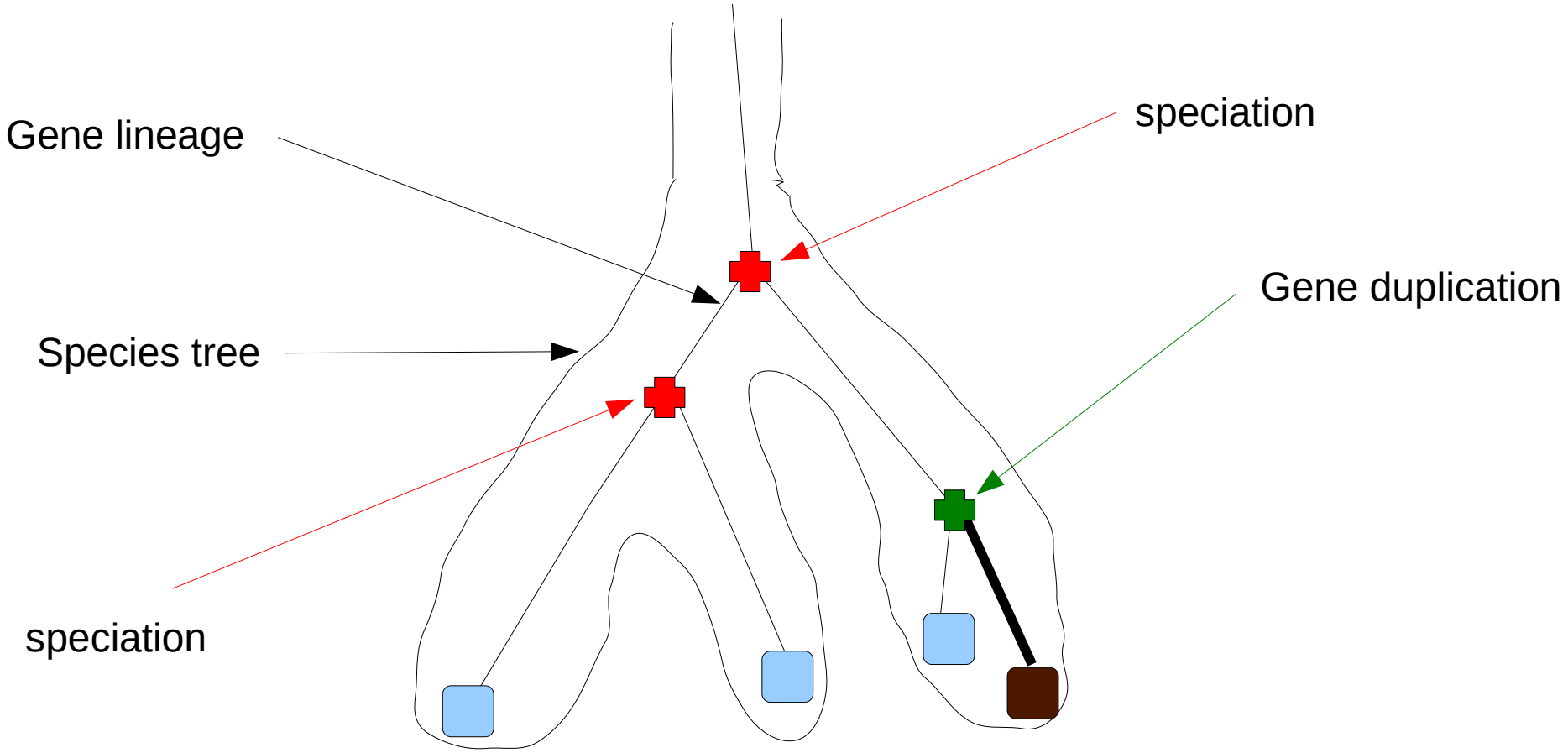
Alignment site/Alignment column

Homologous characters:
Characters that share a common evolutionary history

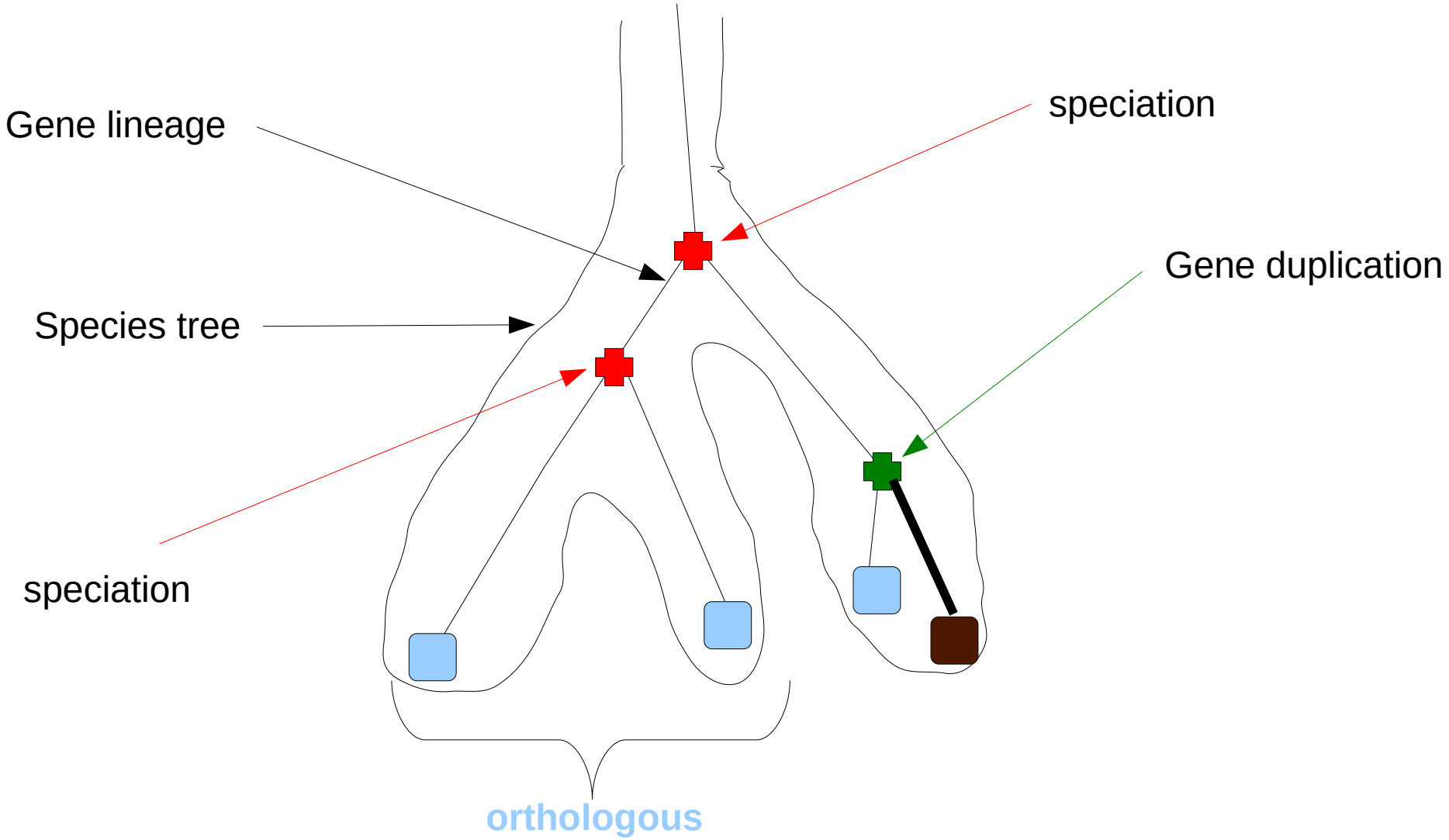
Note that, in this column the characters are similar (*analogous*), but this does not automatically induce homology!

They could be similar by chance or via convergent evolution (see slides later-on)

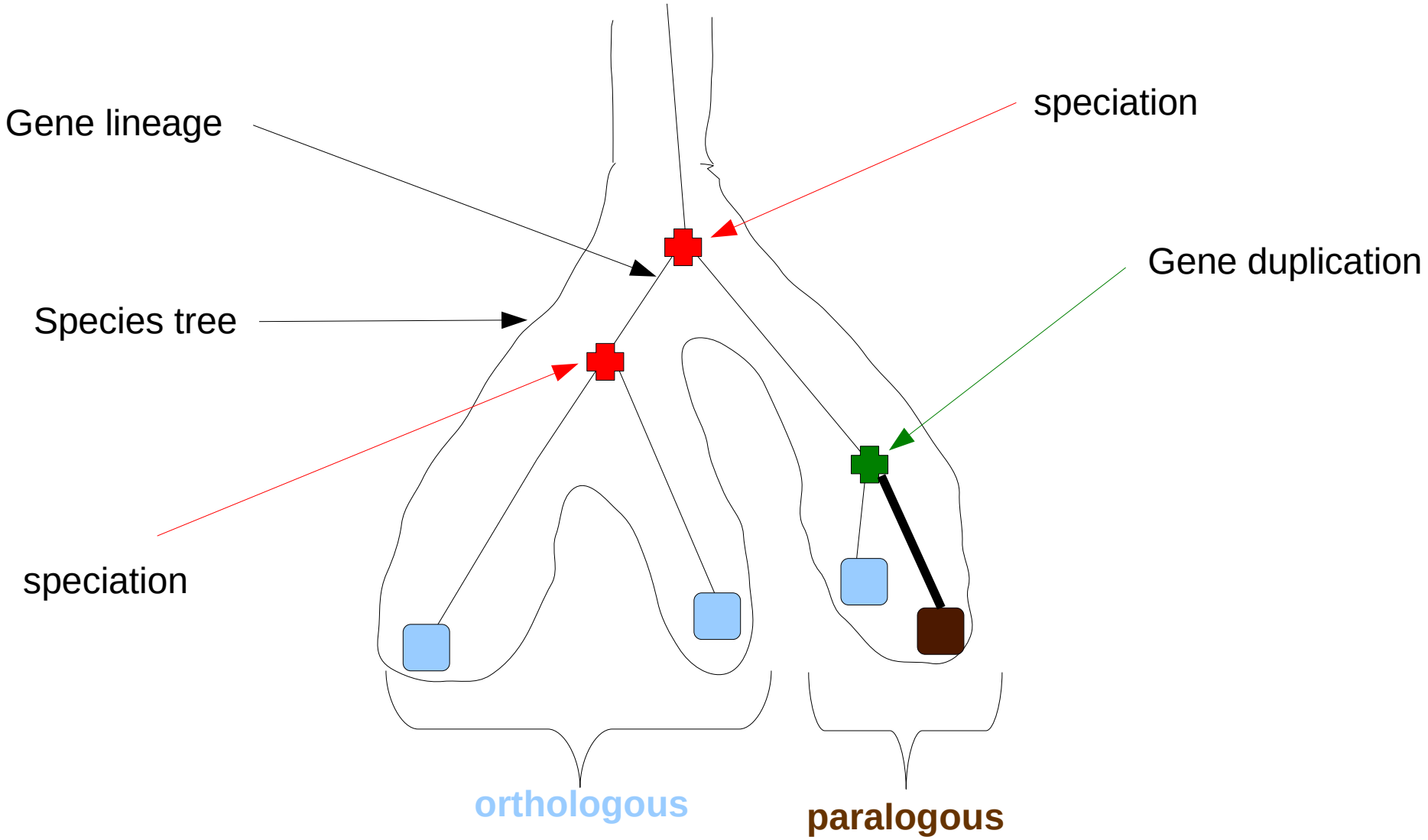
Orthology


Gene lineage
speciation
Gene duplication
Species tree
speciation

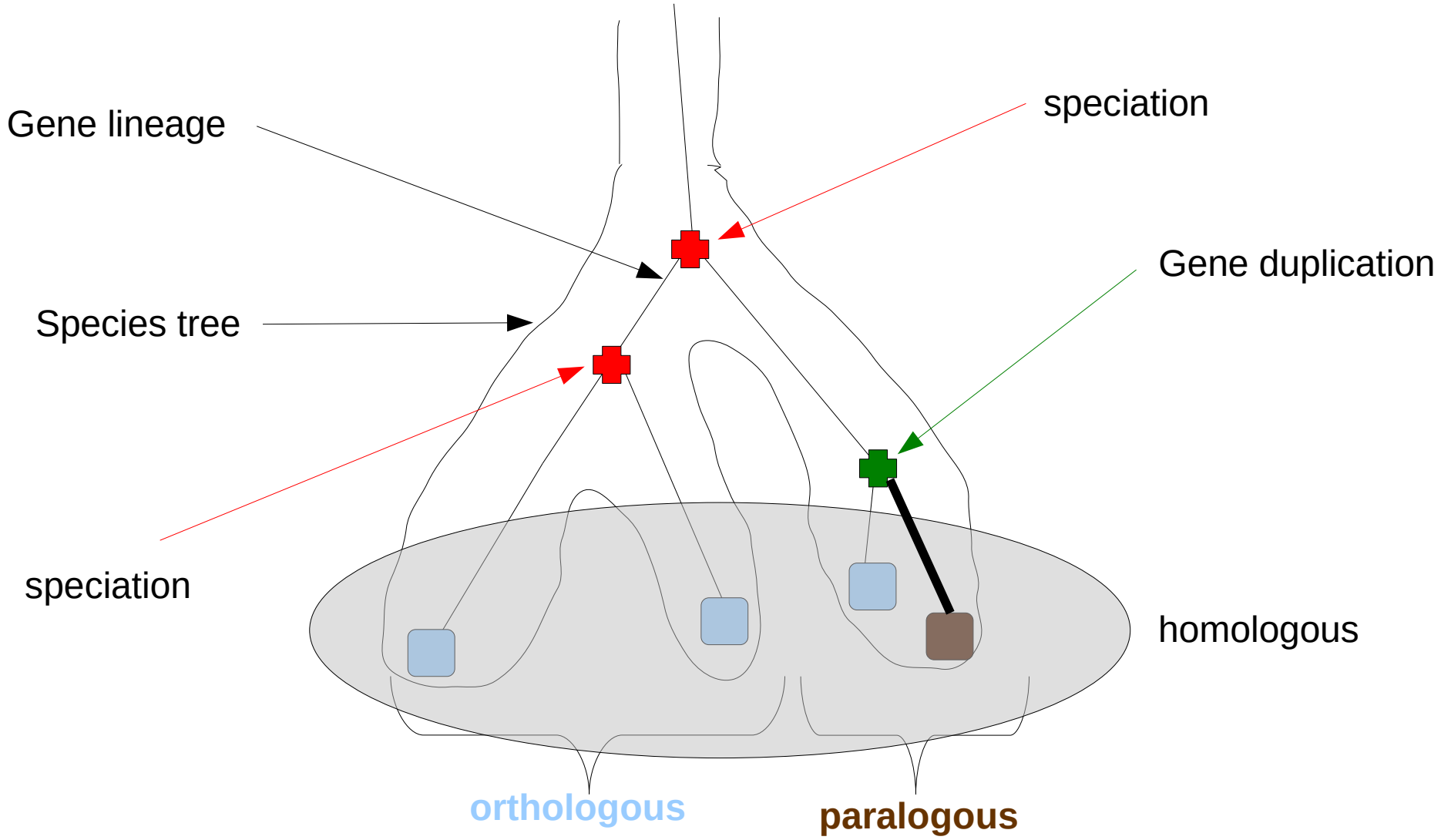
Orthology


Gene lineage
speciation
Gene duplication
Species tree
speciation
orthologous

Orthology

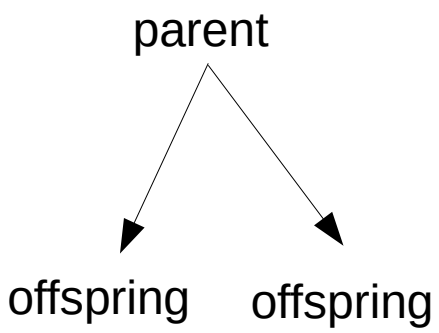

Gene lineage
speciation
Gene duplication
Species tree
speciation
orthologous
paralogous

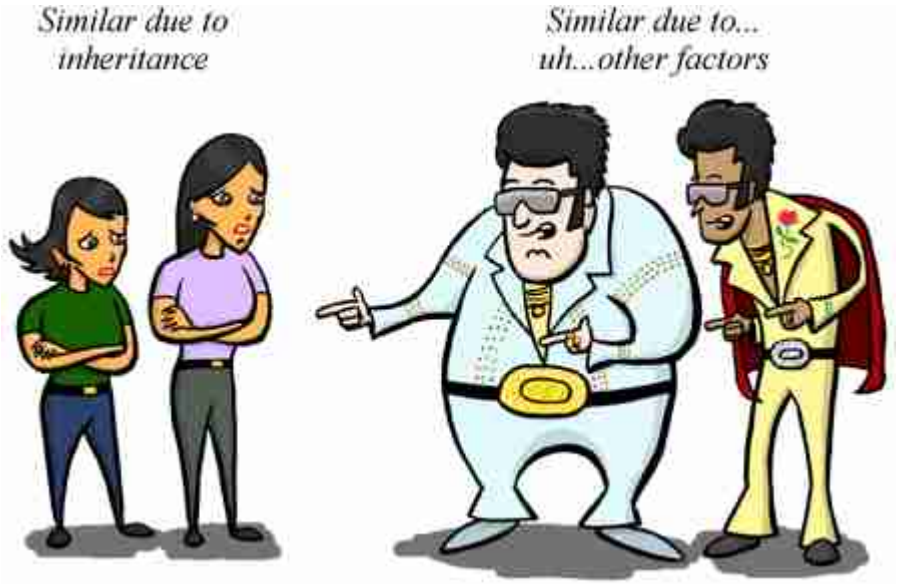
Orthology

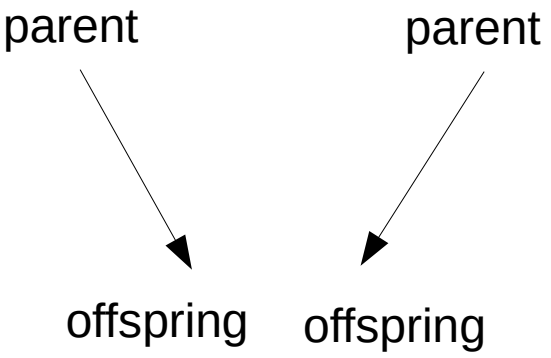

Gene lineage
speciation
Gene duplication
Species tree
speciation
homologous
orthologous
paralogous

Homology

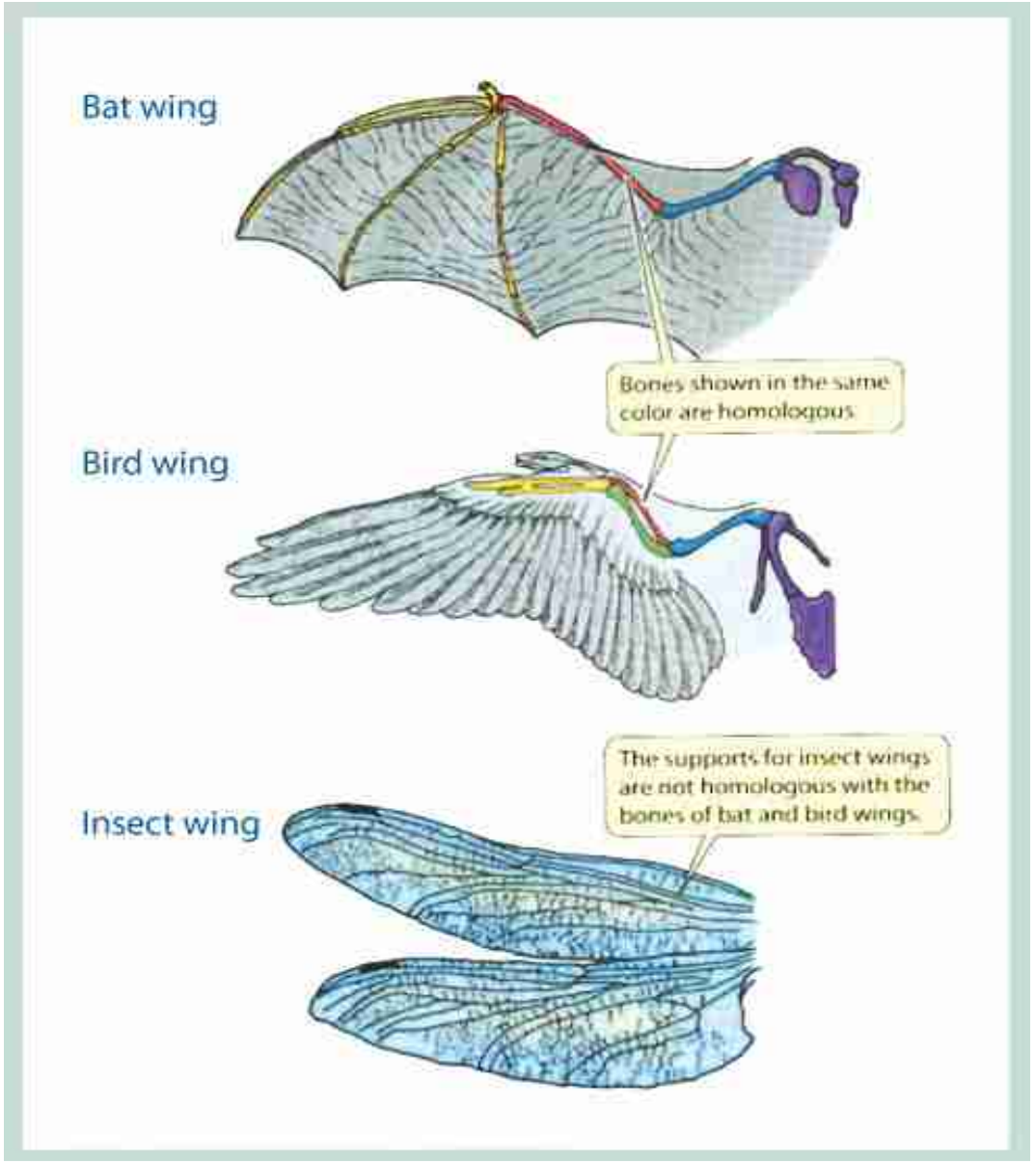
- High sequence similarity does not automatically induce homology
  - Same sequence (gene function) can have evolved independently twice → convergent evolution
  - For short sequences: similar by chance

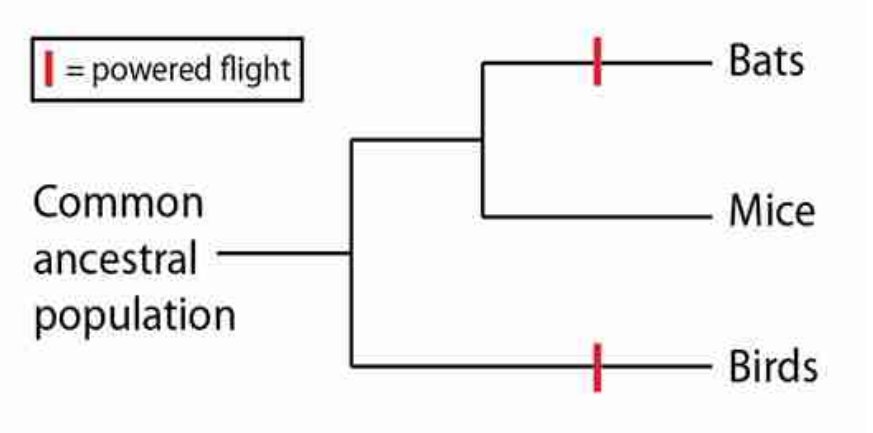






Convergent Evolution





Orthology Assignment

- Numerous methods available
- Will not be covered here → difficult problem
- Henceforth let us assume that we are given a set of *n* orthologous sequences $s_1,\ldots,s_n$ and want to align them

Alignment Criteria

- How do we define alignment quality?
- There are different criteria
  - The SP (sum of pairs) measure
  - Real data benchmarks
  - Curated alignments (based on protein structure)
  - Evolutionary measures
  - Simulations

Alignment Criteria

- How do we define alignment quality?
- There are different criteria
  - **The SP (sum of pairs) measure**
  - Real data benchmarks
  - Curated alignments (based on protein structure)
  - Evolutionary measures
  - Simulations

The SP measure

- **SP**: *sum-of-pairs* score
- Score each MSA site and then add up the scores over all sites
  - Penalize mismatches and gaps
  - Favor matches
  - The per-site score is defined as the sum over all pairwise scores between characters of a site

SP an example

- *SP-score(I, -, I, V)* =

  p(I,-) + *p(I, I)* + *p(I, V)* + *p(-, I)* + *p(-, V)* + *p(I, V)*

- Where *p()* is the penalty function and *p(-,-) := 0*
- Given a MSA with *n* sequences and *m* sites we can thus compute the overall score as:

```
sp = 0;
for(i = 0; i < m; i++)
  sp += SP-score(sites[i]);
```

An example

s1	A	A	G	A	A	-	A
s2	A	T	-	A	A	T	G
s3	C	T	G	-	G	-	G

Using the edit distance for *p()* the score is:

2 + 2 + 2 + 2 + 2 + 2 + 2 = 14

Note that, we can also compute this as the sum of pair-wise edit distances between the aligned sequences:

e(s1,s2) + e(s1,s3) + e(s2,s3) = 4 + 5 + 5

Keep in mind that, *p(-,-) := 0*

The SP measure

- Note that, this is only **one way** to quantify the quality of an alignment
- One can build alignment algorithms that optimize the *SP* measure
- However, alignments (MSAs) with larger *SP* scores may better represent the true evolutionary history of the characters!

How can we extend pair-wise alignment to triple-wise alignment?

- Any ideas?
- What is the time and space complexity?

SP-based optimization

- We can extend the dynamic programming approach for pair-wise sequence alignment to *n* sequences for calculating an *SP-optimal* MSA
- Assume that all *n* sequences have equal length *m*
  - Storing the dynamic programming matrix requires $O(m^n)$ space
  - And the lower bound for time is also $O(m^n)$ because all m^n entries need to be computed → consider an example with *n:= 3*
- As you can imagine, computing the *SP-optimal* MSA is NP-complete

SP-based MSA

- NP-complete
- Not granted that *SP* is the correct (biologically most plausible) criterion!
- Depends on -arbitrary- choice of scoring function *p()*
- We need heuristics or approximation algorithms!
- We will have a look at some basic approaches now …

Star Alignment **Approximation**

- Pick a center sequence s_c
- Align all remaining sequences to s_c using a pairwise sequence alignment algorithm
- “Once a gap, always a gap” strategy

  → gaps inserted into s_c can not be removed again
- s_c can be picked by computing all $O(n^2)$ [more precisely: $(n^2 / 2) - n]$ optimal pair-wise alignments and selecting *the* sequence that has the largest similarity to all other sequences

Star Alignment

```
s1: ATTGCCATT
s2: ATGGCCATT
s3: ATCCAATTTT
s4: ATCTTCTT
s5: ACTGACC
```

Star Alignment

s1: ATTGCCATT ← center sequence

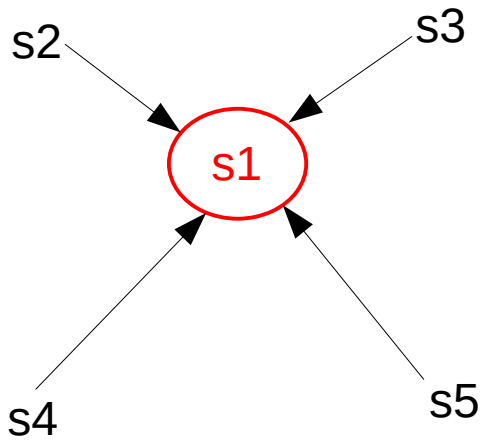
s2: ATGGCCATT

s3: ATCCAATTTT

s4: ATCTTCTT

s5: ACTGACC

Star Alignment



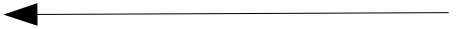
Star Alignment

```
s1: ATTGCCATT
s2: ATGGCCATT

s1: ATTGCCATT--
s3: ATC-CAATTTT

s1: ATTGCCATT
s4: ATCTTC-TT

s1: ATTGCCATT
s5: ACTGACC--
```

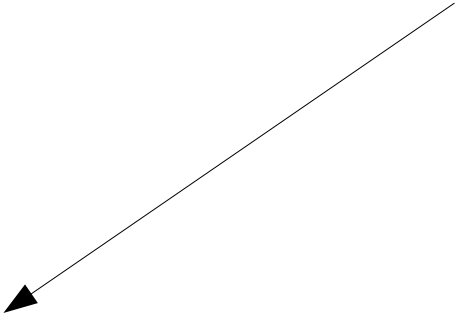
Star Alignment

```
s1: ATTGCCATT
s2: ATGGCCATT

s1: ATTGCCATT--
s3: ATC-CAATTTT

s1: ATTGCCATT--
s4: ATCTTC-TT--

s1: ATTGCCATT--
s5: ACTGACC----
```

Gaps inserted



“Once a gap, always a gap”



The Star Alignment

```
s1: ATTGCCATT--
s2: ATGGCCATT--
s3: ATC-CAATTTT
s4: ATCTTC-TT--
s5: ACTGACC----
```

Another Example

s1:ATTGCCATT

s2:ATGGCCATT

s3:ATCCAATTTT

s4:ATCTTCTT

s5:ATTGCCGATT

Another Example

```
s1:ATTGCCATT
s2:ATGGCCATT

s1:ATTGCCATT--
s3:AT-CCAATTTT

s1:ATTGCCATT
s4:ATCTTC-TT

s1:ATTGCC-ATT
s5:ATTGCCGATT
```

Another Example

s1:ATTGCCATT

s2:ATGGCCATT

s1:ATTGCCATT--

s3:AT-CCAATTTT

s1:ATTGCCATT

s4:ATCTTC-TT

s1:ATTGCC-ATT

s5:ATTGCCGATT

s1:ATTGCCATT

s2:ATGGCCATT

Another Example

s1:ATTGCCATT
s2:ATGGCCATT

s1:ATTGCCATT--
s3:AT-CCAATTTT

s1:ATTGCCATT
s4:ATCTTC-TT

s1:ATTGCC-ATT
s5:ATTGCCGATT

s1:ATTGCCATT
s2:ATGGCCATT

s1:ATTGCCATT--
s2:ATGGCCATT--
s3:AT-CCAATTTT

Another Example

s1:ATTGCCATT
s2:ATGGCCATT

s1:ATTGCCATT--
s3:AT-CCAATTTT

s1:ATTGCCATT
s4:ATCTTC-TT

s1:ATTGCC-ATT
s5:ATTGCCGATT

s1:ATTGCCATT
s2:ATGGCCATT

s1:ATTGCCATT--
s2:ATGGCCATT--
s3:AT-CCAATTTT

s1:ATTGCCATT--
S2:ATGGCCATT--
S3:AT-CCAATTTT
s4:ATCTTC-TT--

Another Example

```
s1:ATTGCCATT
s2:ATGGCCATT

s1:ATTGCCATT--
s3:AT-CCAATTTT

s1:ATTGCCATT
s4:ATCTTC-TT

s1:ATTGCC-ATT
s5:ATTGCCGATT
```

```
s1:ATTGCCATT
s2:ATGGCCATT

s1:ATTGCCATT--
s2:ATGGCCATT--
s3:AT-CCAATTTT

s1:ATTGCCATT--
S2:ATGGCCATT--
S3:AT-CCAATTTT
s4:ATCTTC-TT--

s1:ATTGCC-ATT--
S2:ATGGCC-ATT--
S3:AT-CCA-ATTTT
s4:ATCTTC--TT--
s5:ATTGCCGATT--
```

Shift right!

Pairwise alignment step

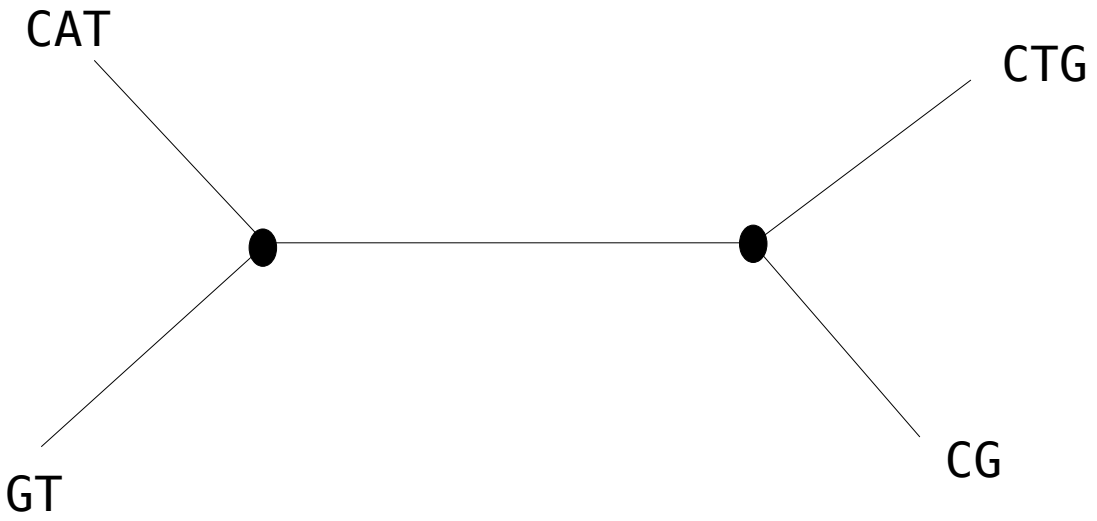
Merging step

Star Alignment Approximation

- Produces an MSA whose *SP* score is *< 2 * optimum*
- Proof omitted
- Reference: D. Gusfield “Efficient methods for multiple sequence alignment with guaranteed error bounds”, *Bulletin of Mathematical Biology*, 1993.

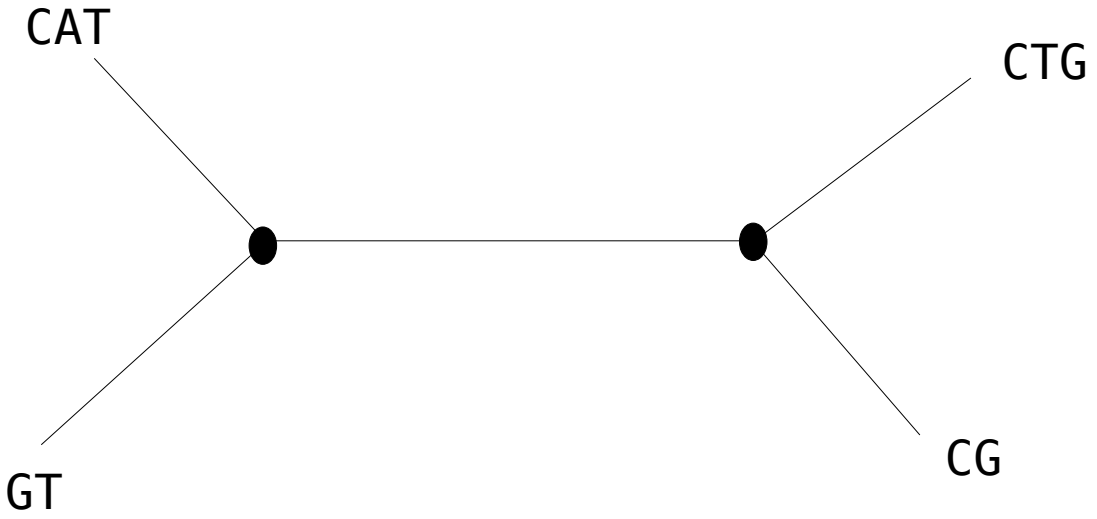
Tree Alignment

- If an evolutionary tree for the sequences is available



Tree Alignment

- Find an assignment of sequences to the inner nodes such that the sum over the **similarity** scores on all branches is maximized

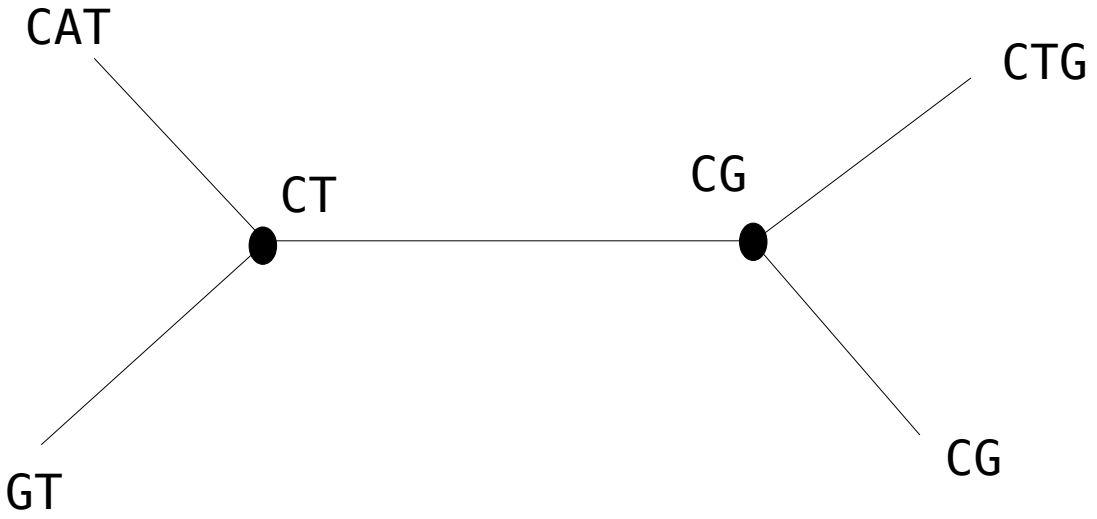


Tree Alignment

p(a,b) := 1 if a = b

p(a,b) := 0 if a ≠b

p(a,-) := -1

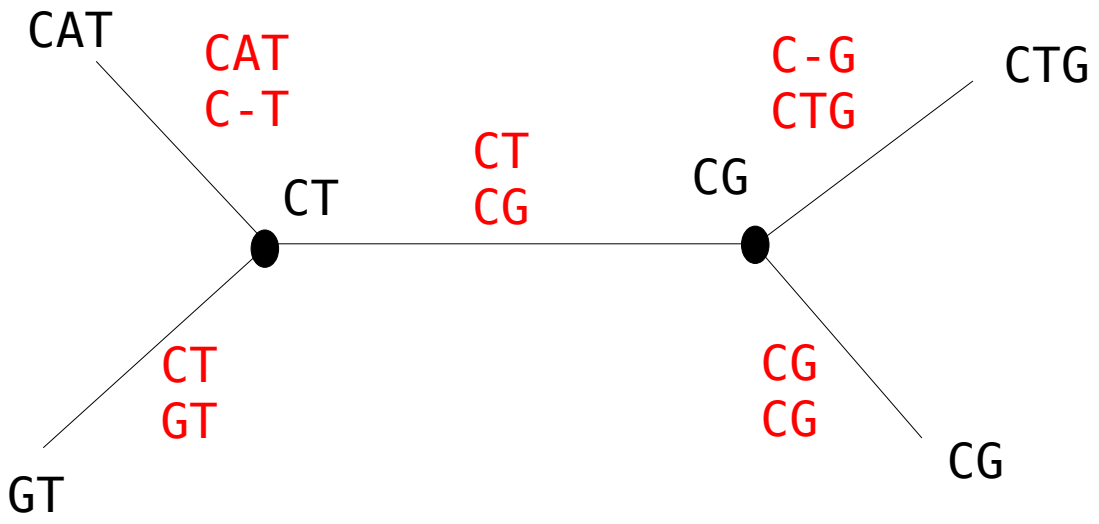


Tree Alignment

p(a,b) := 1 if a = b

p(a,b) := 0 if a ≠b

p(a,-) := -1

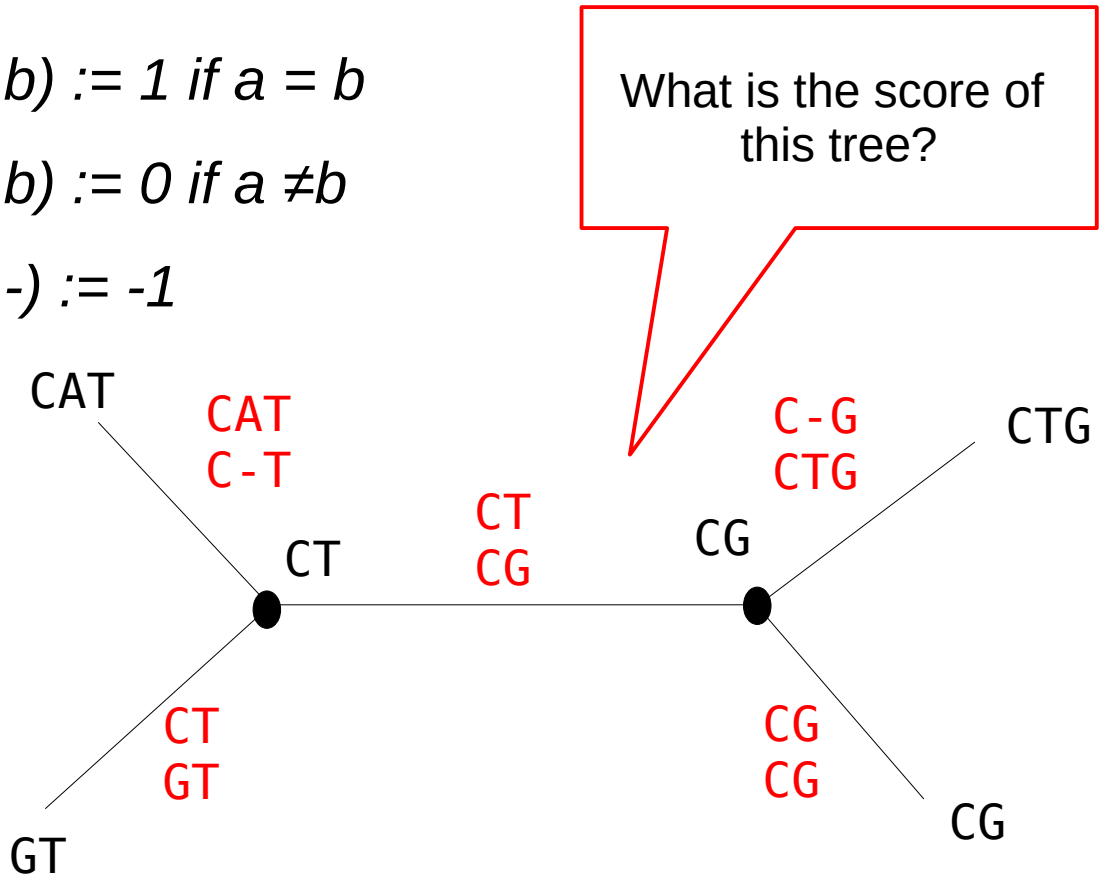


Tree Alignment

p(a,b) := 1 if a = b

p(a,b) := 0 if a ≠b

p(a,-) := -1

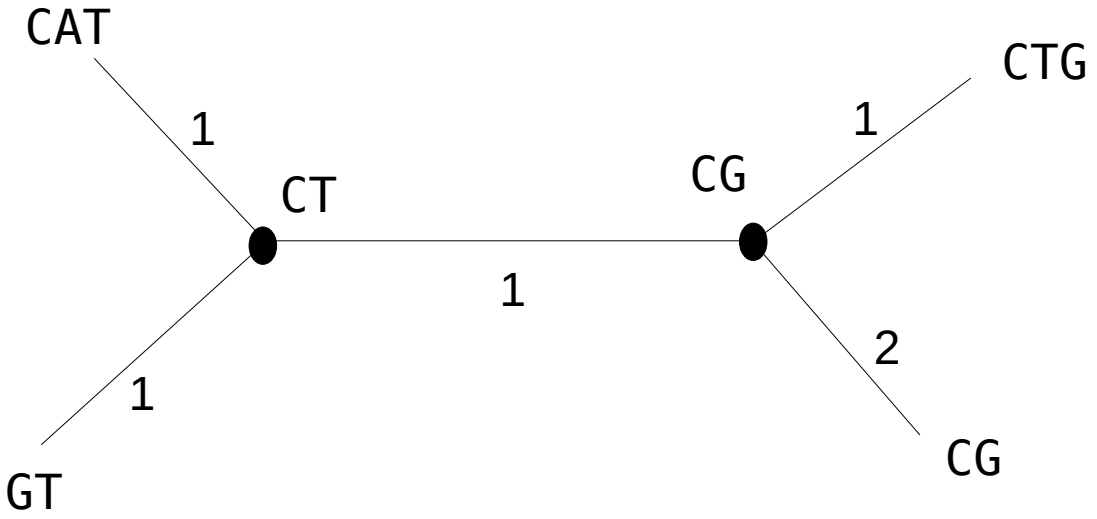


Tree Alignment

p(a,b) := 1 if a = b

p(a,b) := 0 if a ≠b

p(a,-) := -1

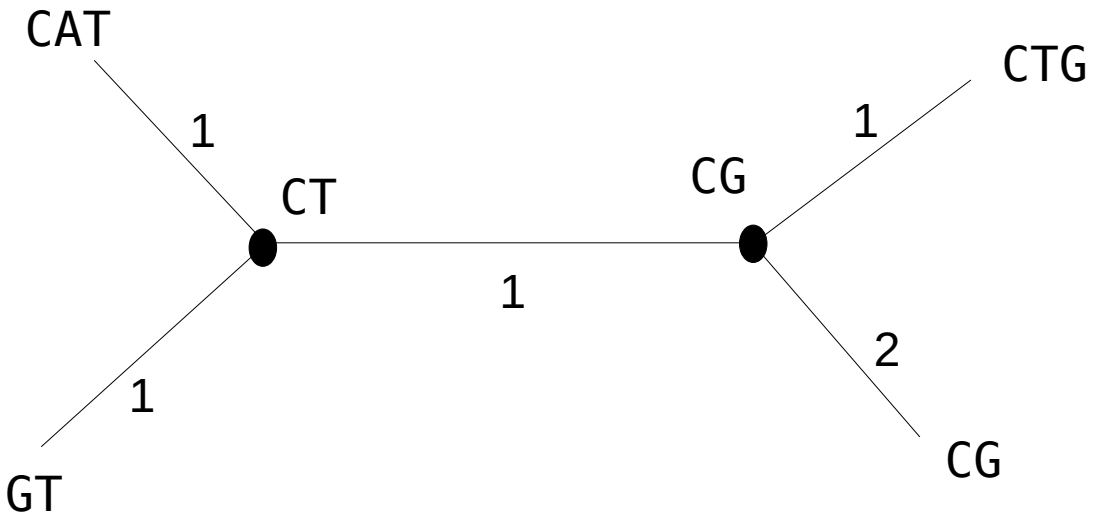


Tree Alignment

p(a,b) := 1 if a = b

p(a,b) := 0 if a ≠b

p(a,-) := -1



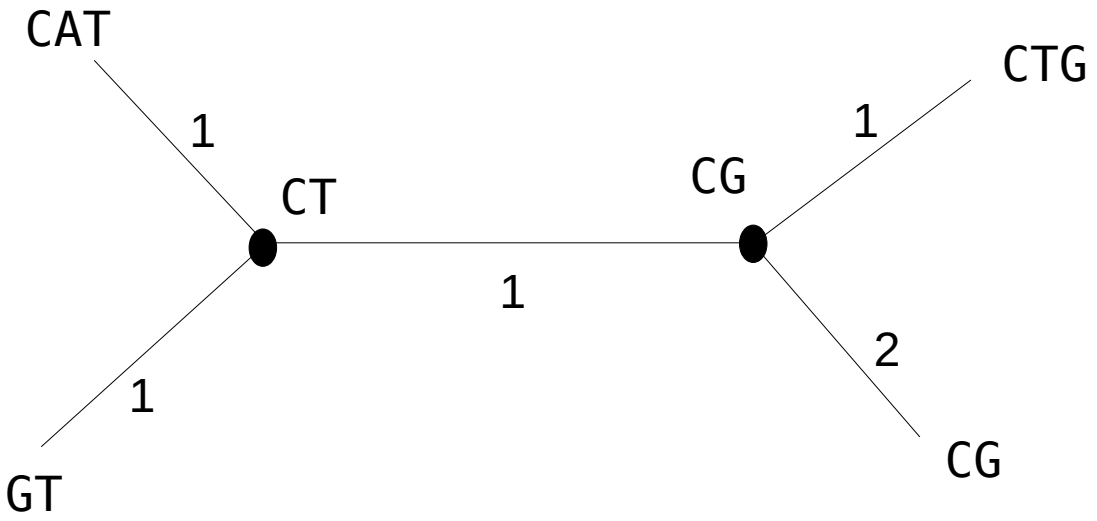
Overall score: 6 → now, maximize this score

Tree Alignment

p(a,b) := 1 if a = b

p(a,b) := 0 if a ≠b

p(a,-) := -1



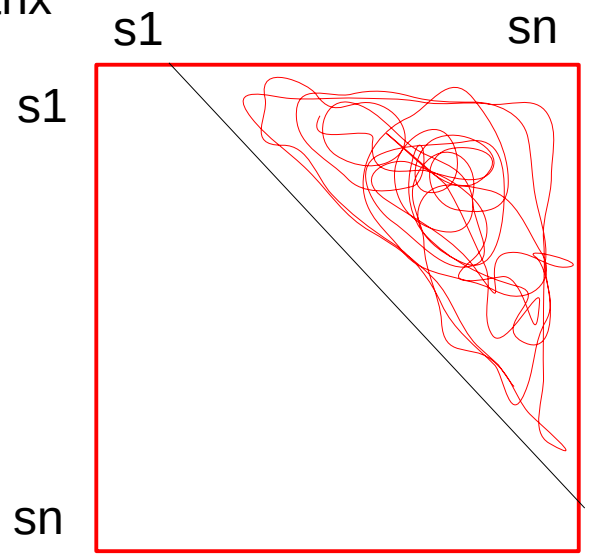
Overall score: 6 → maximize this score
This problem is NP-hard because we don't have the ancestral states

Tree-Based Alignment

- Hen and egg problem
  - → we need an MSA to build a tree
  - → we need a tree to compute a MSA
  - → if the alignment is wrong, the tree might be wrong
  - → if the tree is wrong, the MSA might be wrong
- One idea
  - → simultaneous inference of tree & alignment
  - → very hard problem: trying to solve two generally NP-hard or NP-complete problems simultaneously

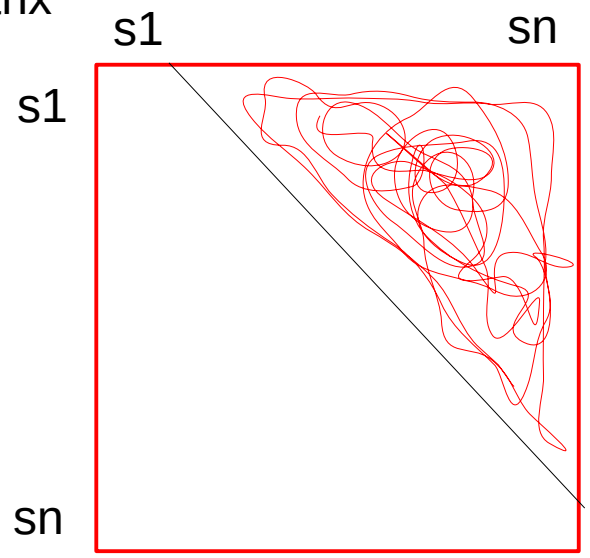
Practical approaches

Build a pair-wise distance matrix



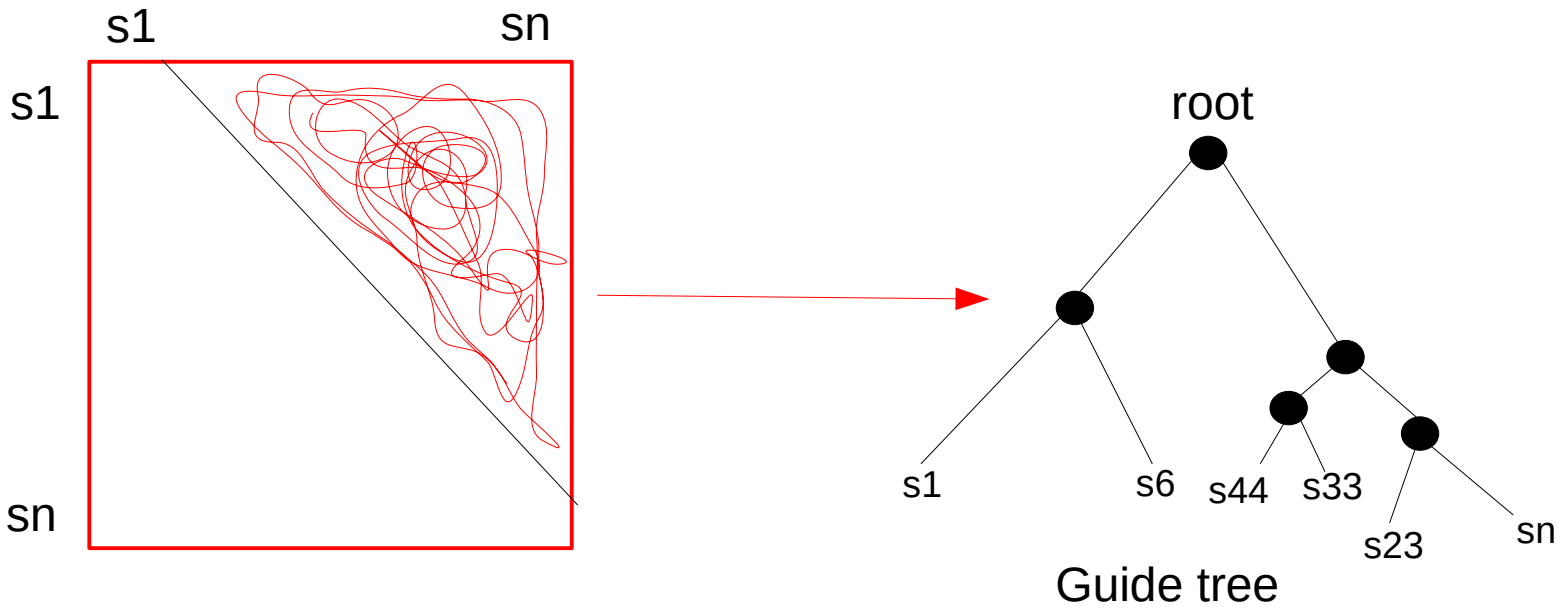
Practical approaches

Build a pair-wise distance matrix

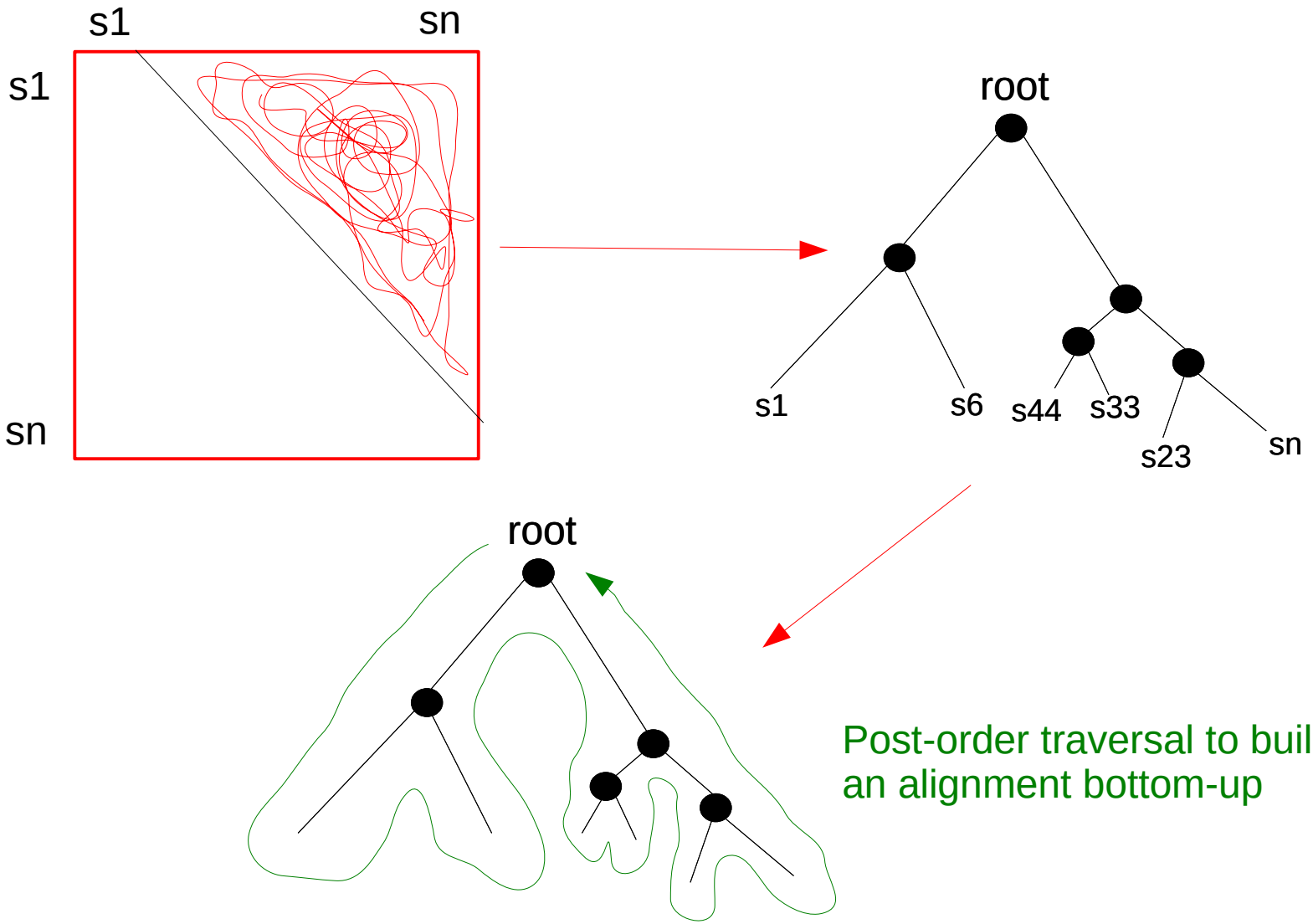


Computation of pair-wise distance matrix
Using pair-wise alignment scores can be time and memory-intensive due to $O(n^2)$ complexity
One may use approximate distance methods based on *k-mers*
(remember last lecture!)

Practical approaches

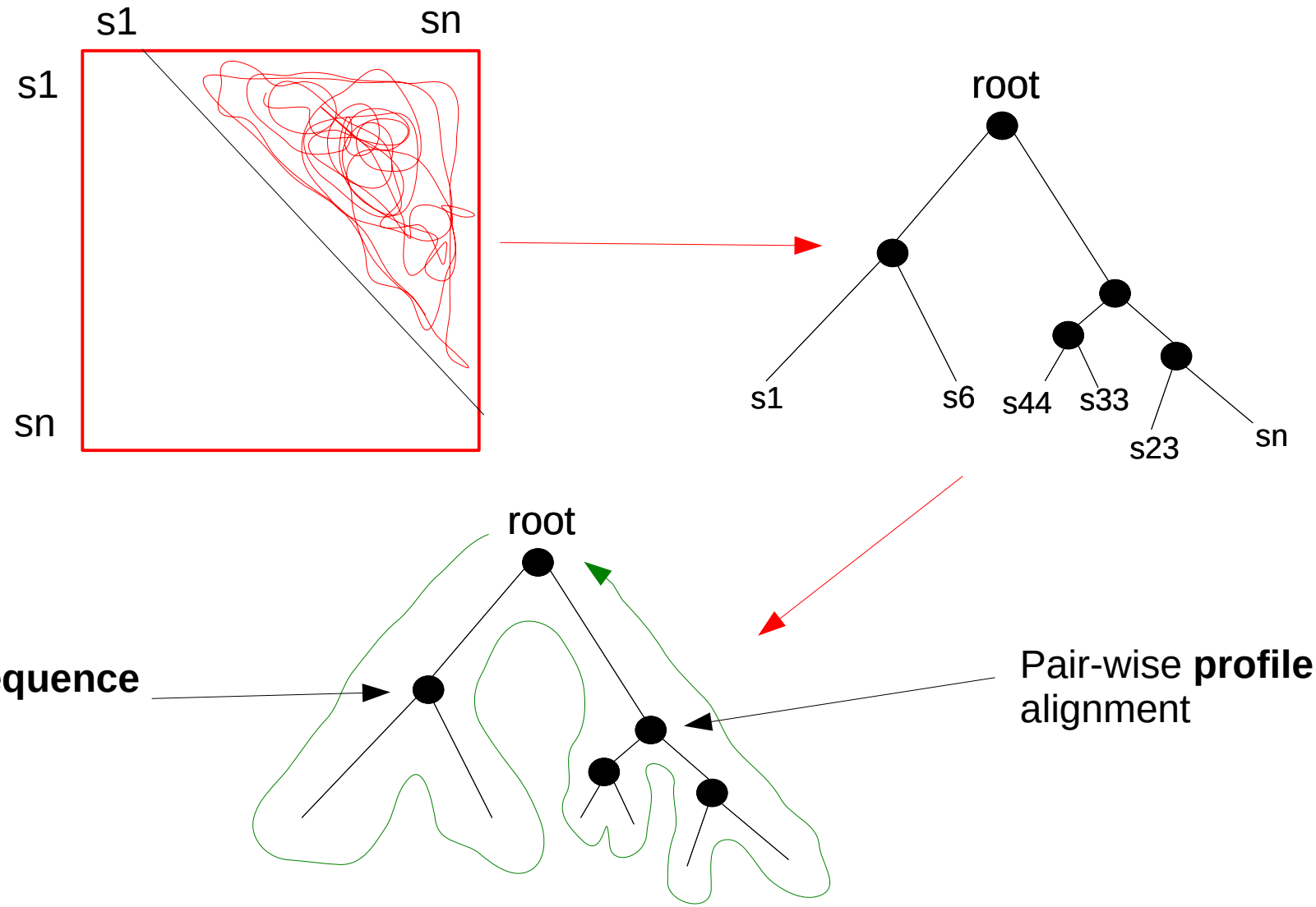


Practical approaches



Post-order traversal to build an alignment bottom-up

Practical approaches



Practical Approaches

- Guide-tree approach
- Compute all $(n^2/2)-n$ pair-wise distances (alignments) between the n sequences
- Use these distances for hierarchical clustering
  - e.g. with the Neighbor Joining (NJ) algorithm → we will see this later-on for tree building/phylogenetic inference
- Use the distance-based tree to calculate pair-wise
  - Sequence-sequence
  - Sequence-profile
  - Profile-profile

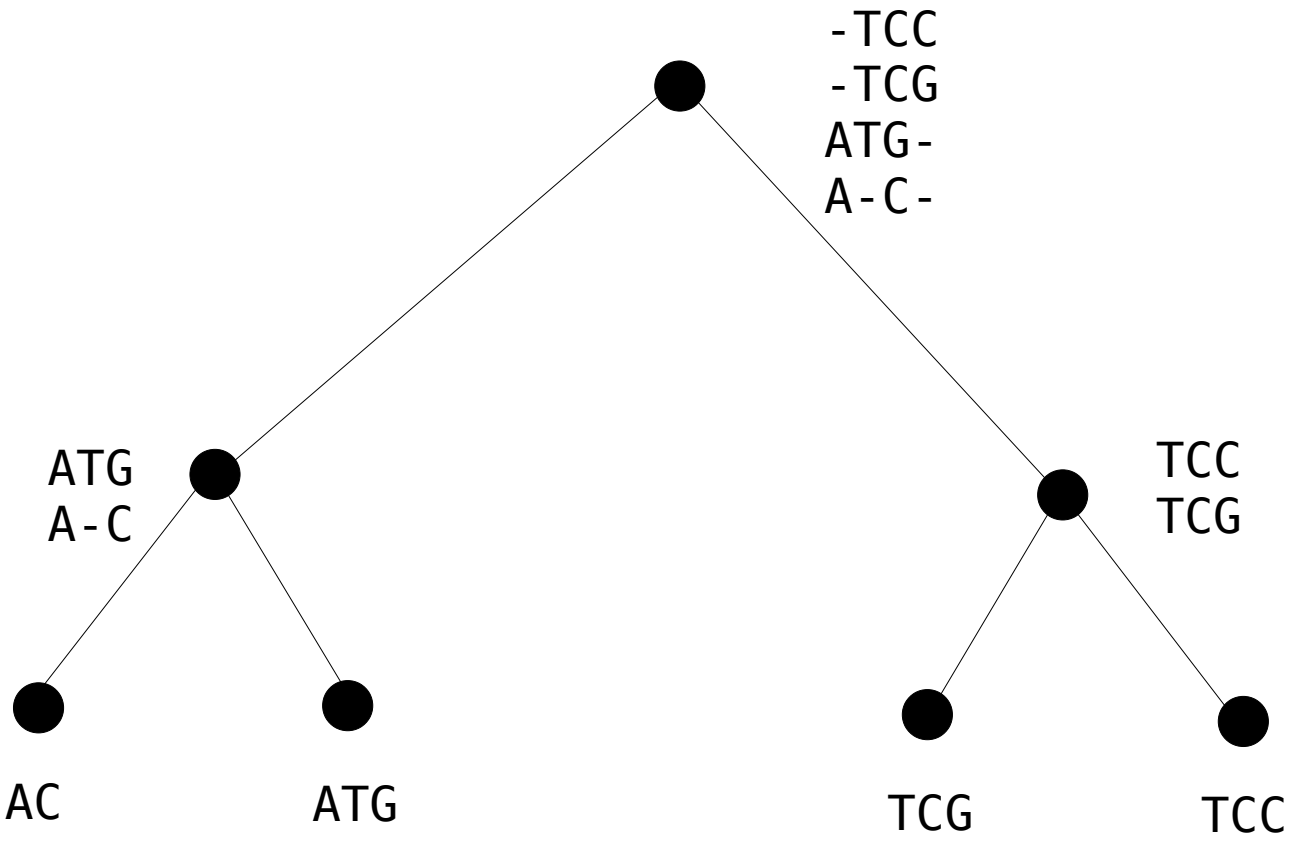
  … alignments bottom up toward the root via a post-order tree traversal
- Many widely-used MSA programs rely on this idea: e.g., **Clustal** family of tools**, T-COFFEE, MUSCLE**

Progressive MSA

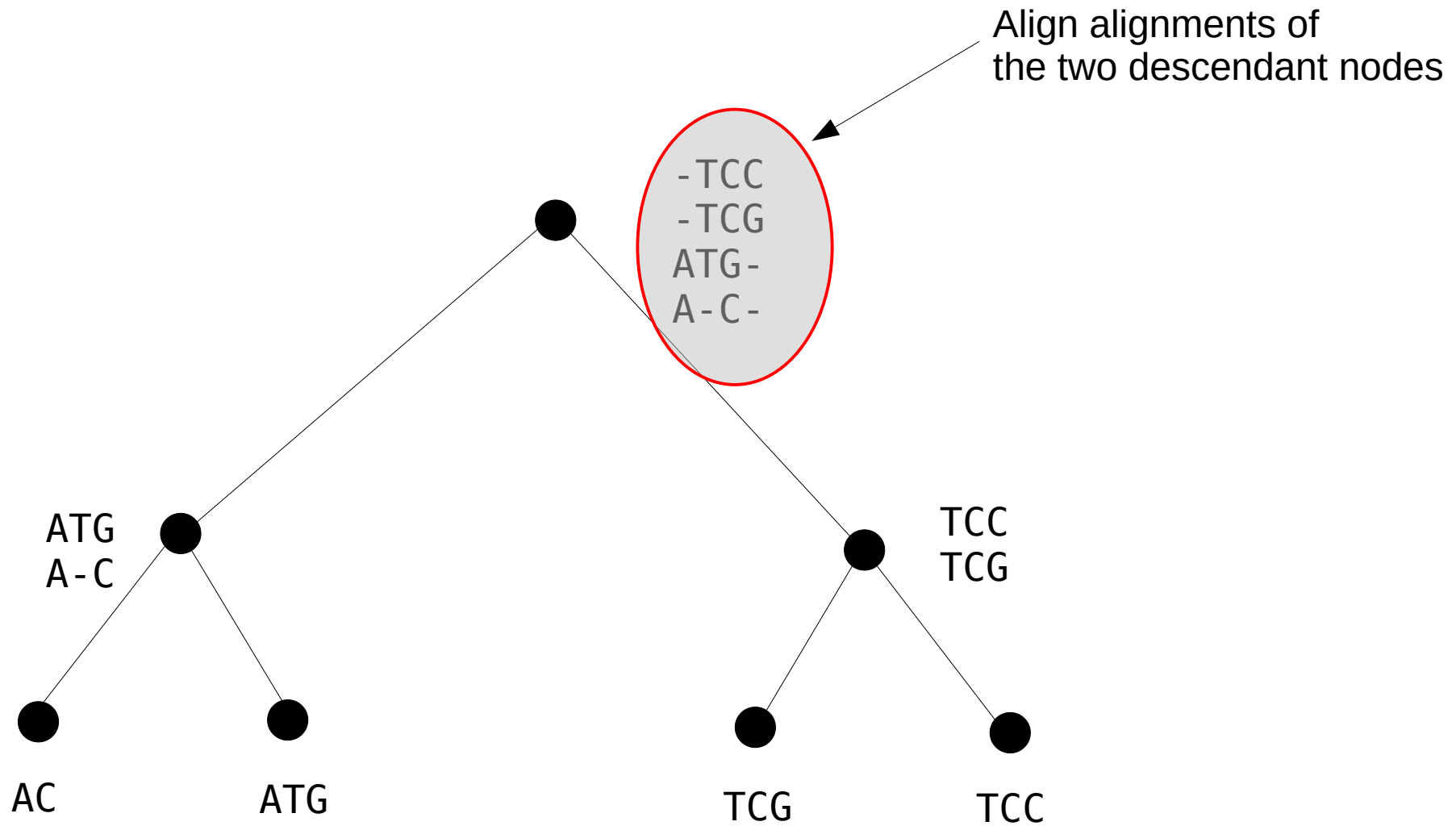
Progressive MSA

Progressive MSA

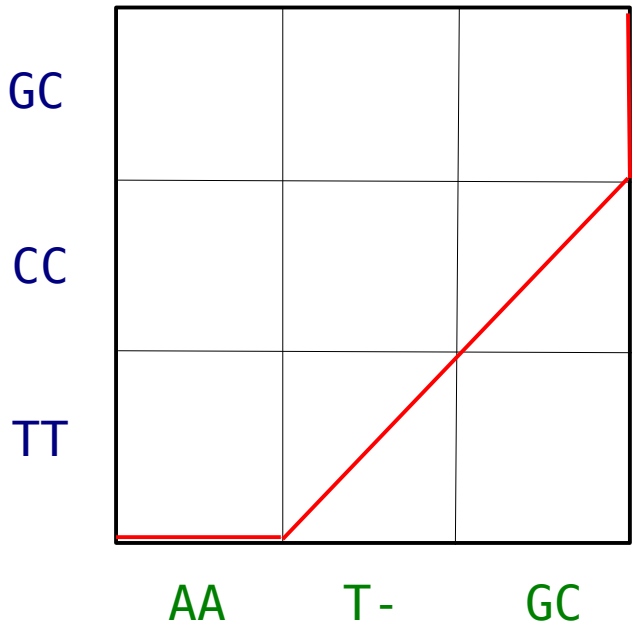
Progressive MSA



Progressive MSA



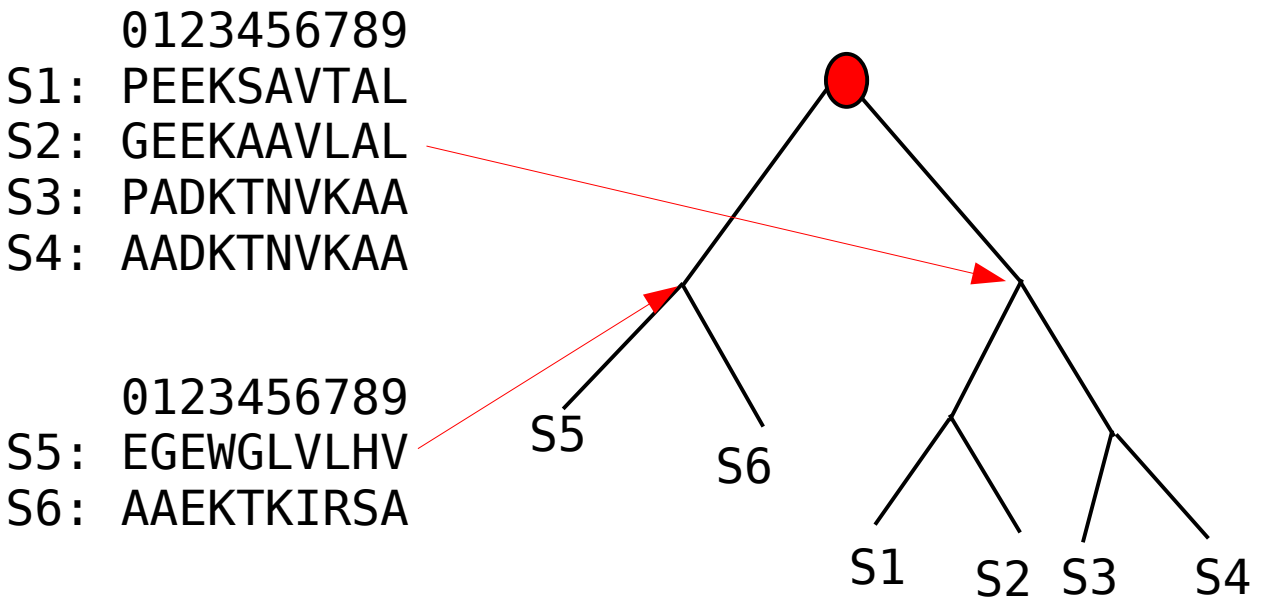
Profile Alignment



-TCC
-TCG
ATG-
A-C-

Profile Alignment

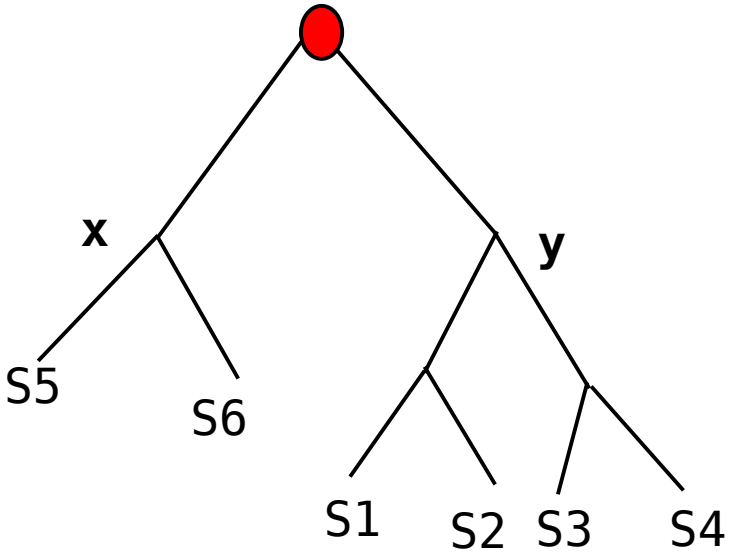
- Generalization of pair-wise sequence alignment to pair-wise profile alignment
- Average over all possibilities



Profile Alignment

- Generalization of pair-wise sequence alignment to pair-wise profile alignment
- Average over all possibilities

```
    0123456789
S1: PEEKSAVTAL
S2: GEEKAAVLAL
S3: PADKTNVKAA
S4: AADKTNVKAA

    0123456789
S5: EGEWGLVLHV
S6: AAEKTKIRSA
```

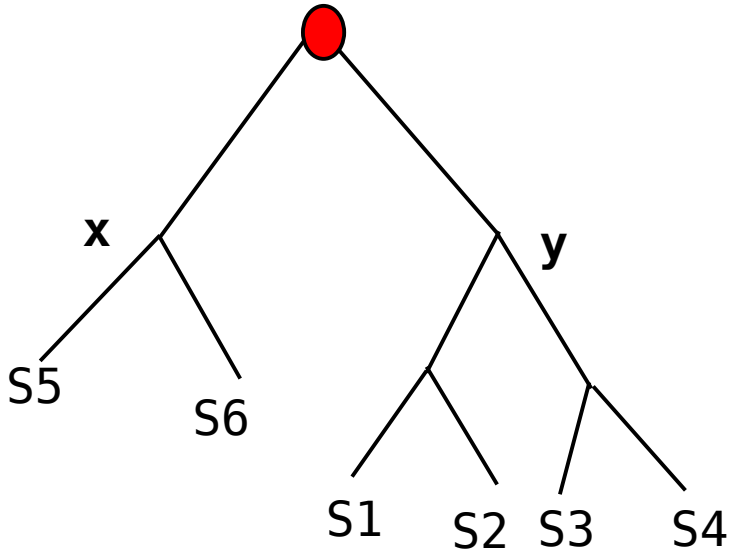


Compute score between position 6 of **x** and position 7 of **y**

Profile Alignment

- Generalization of pair-wise sequence alignment to pair-wise profile alignment
- Average over all possibilities

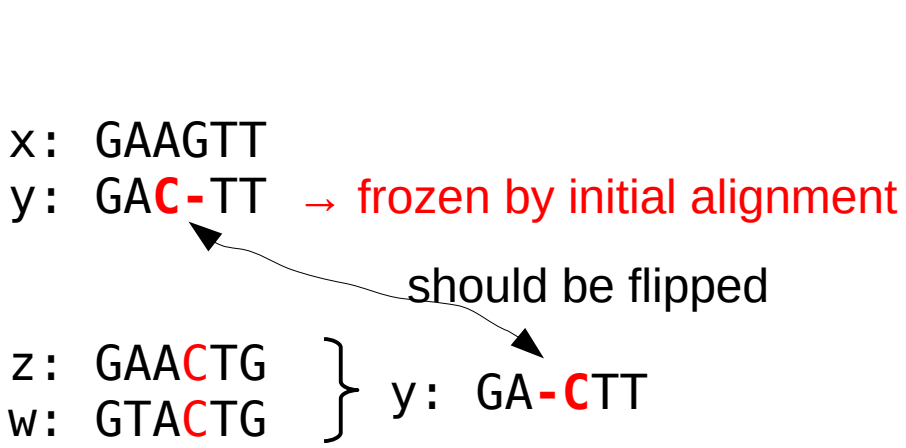
```
    0123456789
S1: PEEKSAVTAL
S2: GEEKAAVLAL
S3: PADKTNVKAA
S4: AADKTNVKAA

    0123456789
S5: EGEWGLVLHV
S6: AAEKTKIRSA
```

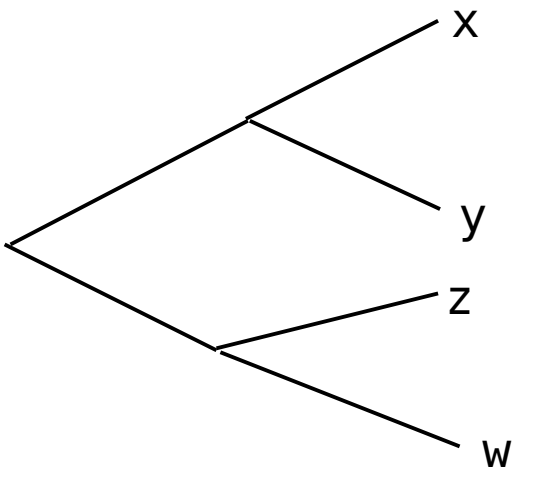


Weighted average over all *8 (2 * 4)* possibilities:
Score: *1/8 * [p(T,V) + p(T,I) + p(L, V) + p(L, I) + p(K,V) + p(K,I) + p(K,V) + p(K,I)]*

Problems with progressive MSA

- Initial pair-wise alignments are “frozen”
- Can't be corrected when new evidence emerges

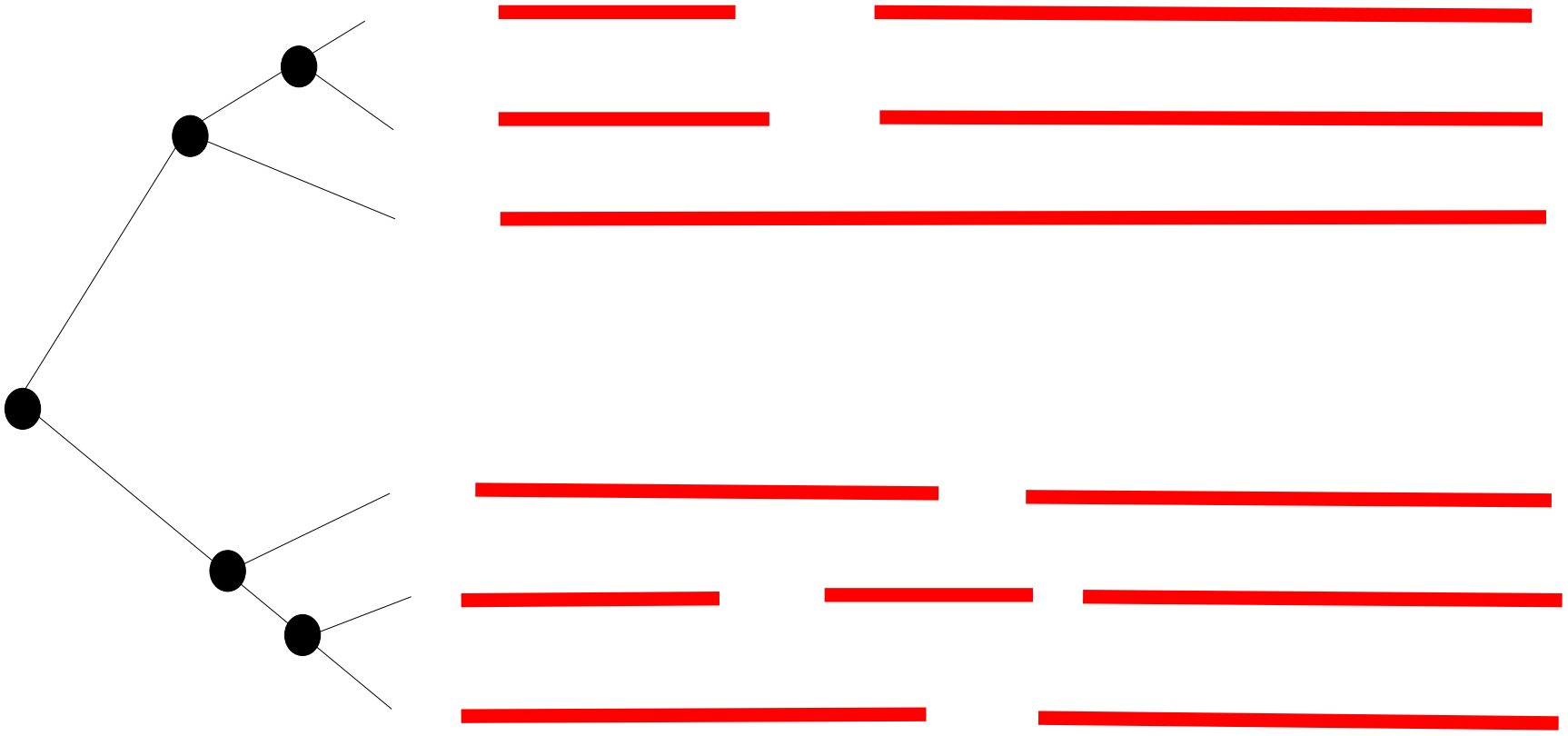




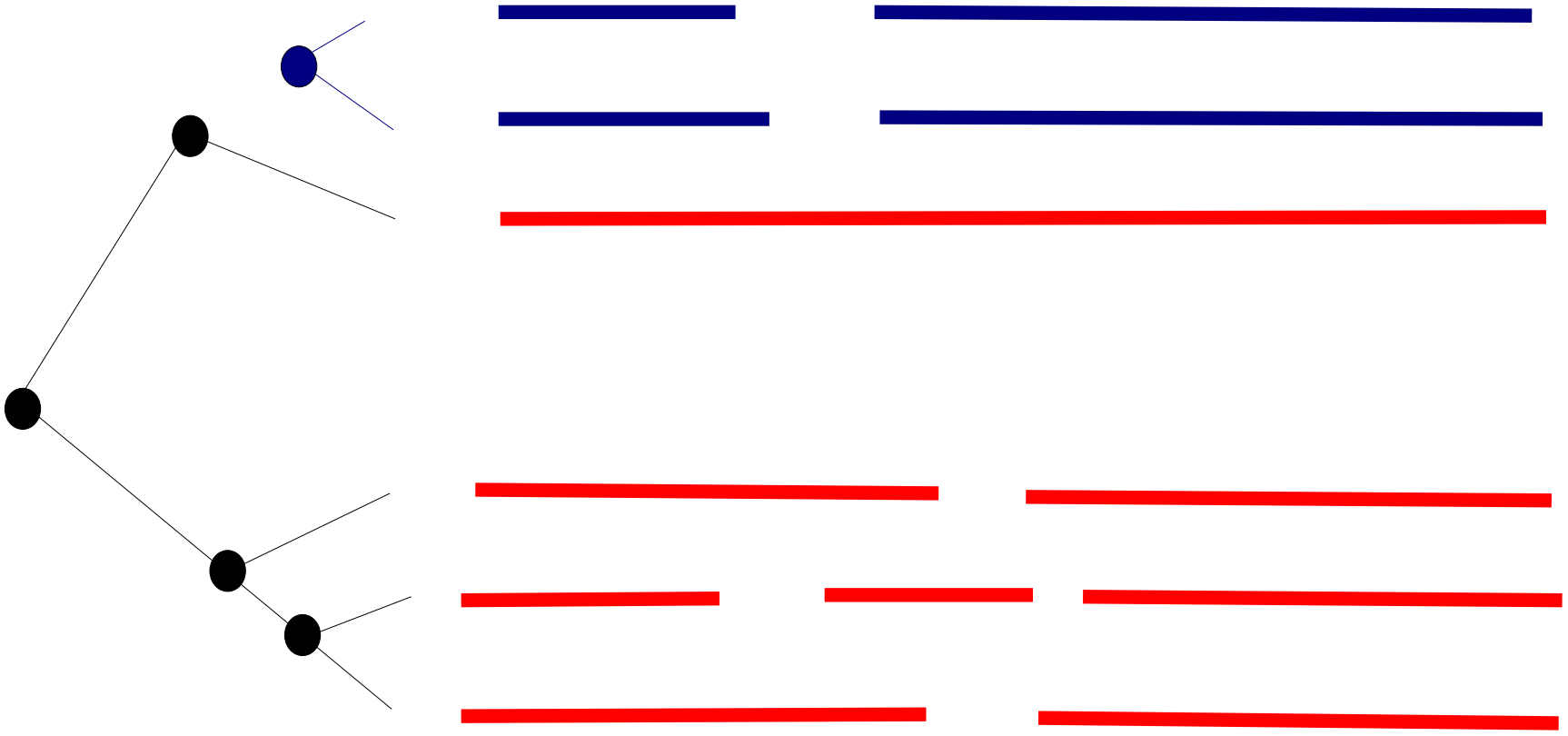
Iterative Progressive MSA

- e.g. MUSCLE, PRRP, MAFFT
- Execute progressive MSA several times to re-fine the alignment

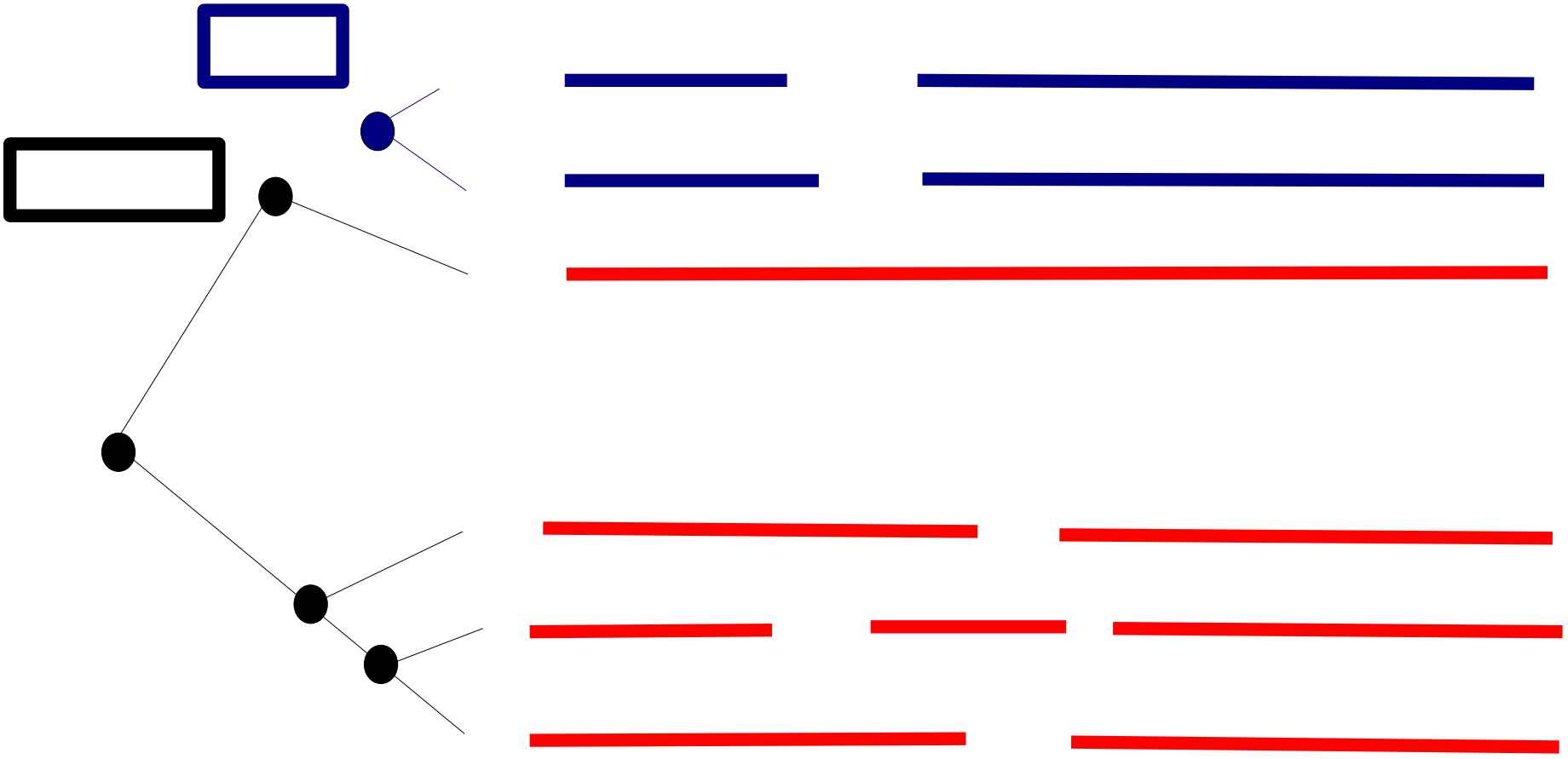
MUSCLE Re-Finement



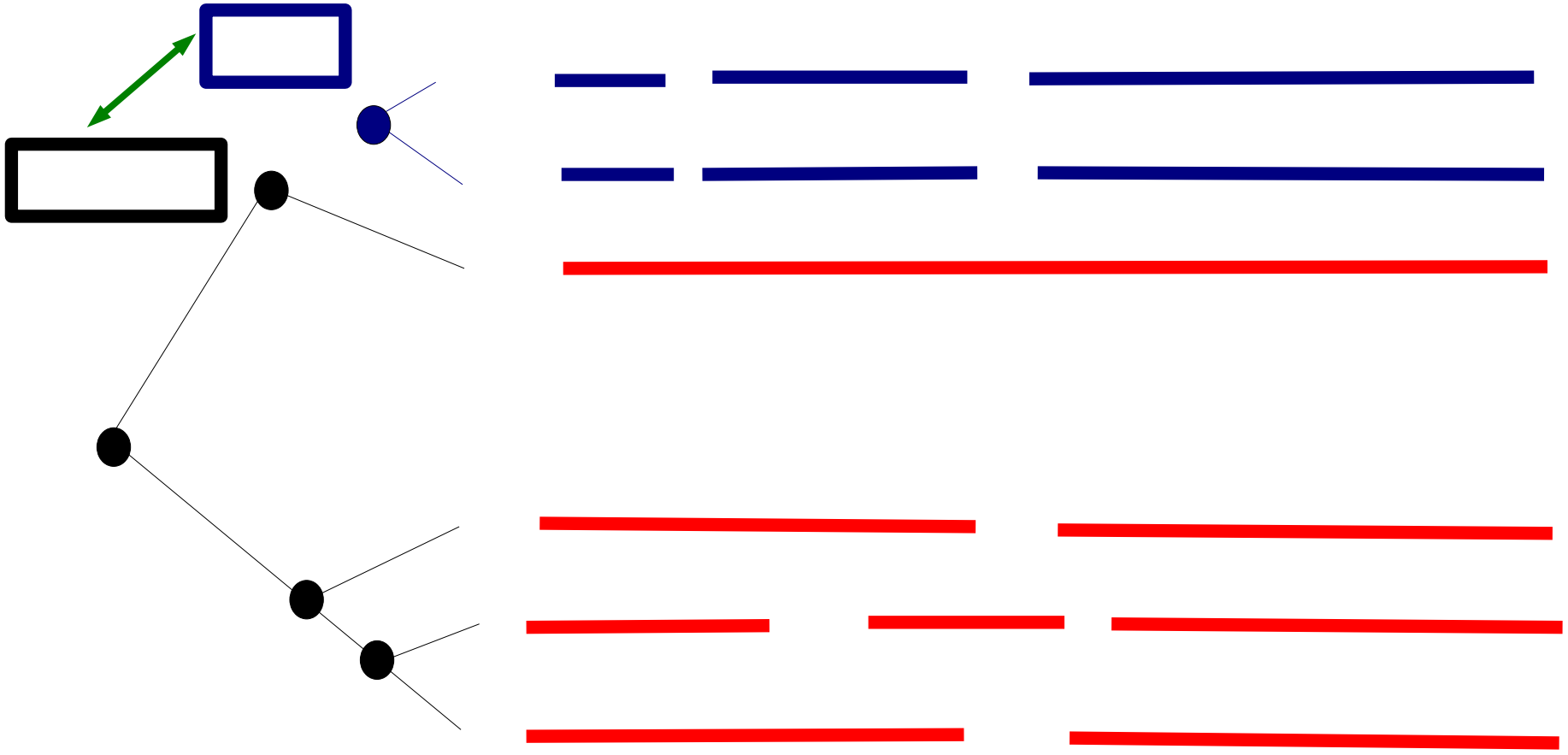
MUSCLE Re-Finement



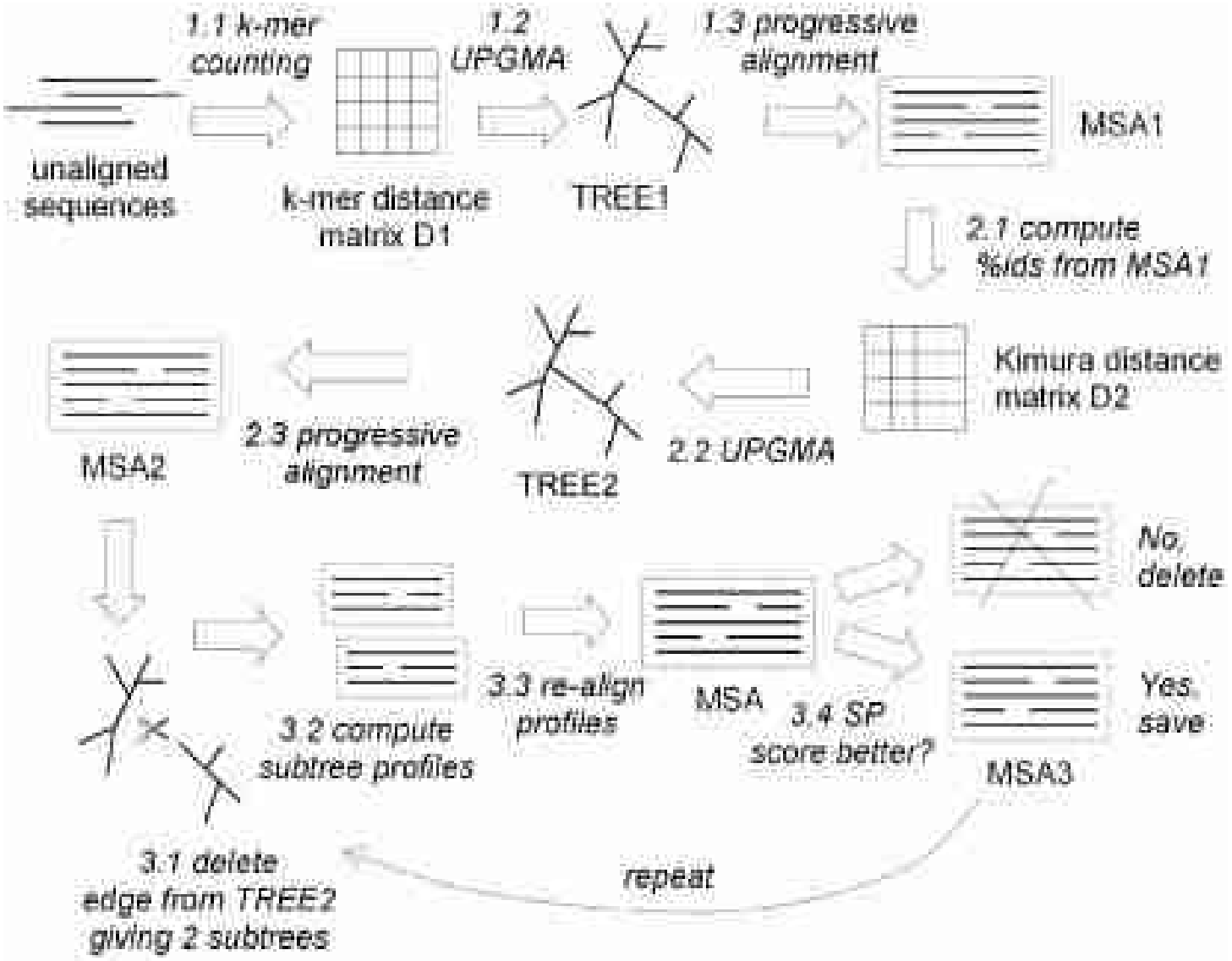
MUSCLE Re-Finement



MUSCLE Re-Finement



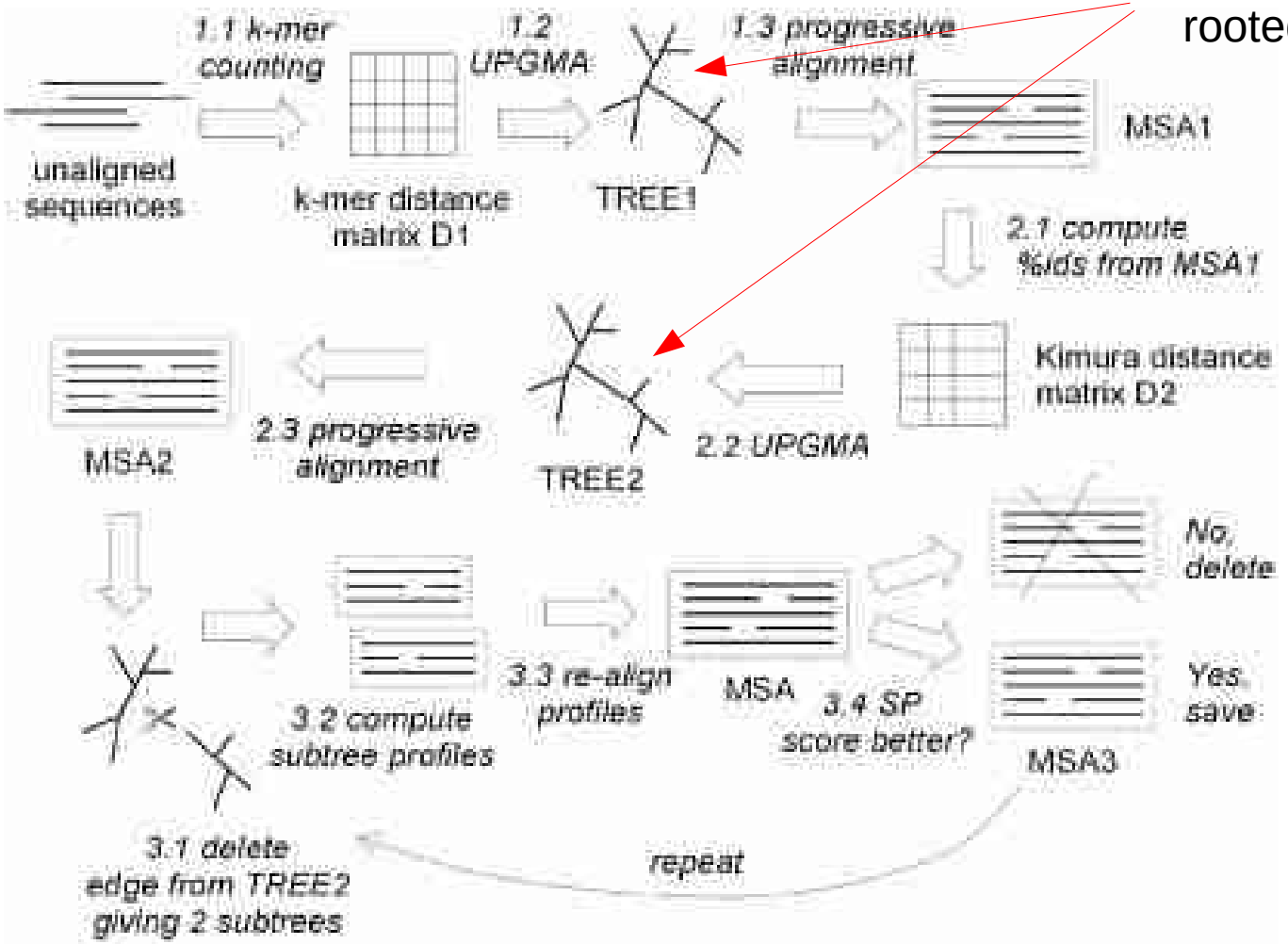
MUSCLE Details



From: “MUSCLE: multiple sequence alignment with high accuracy and high throughput”

MUSCLE Details

These trees are rooted!



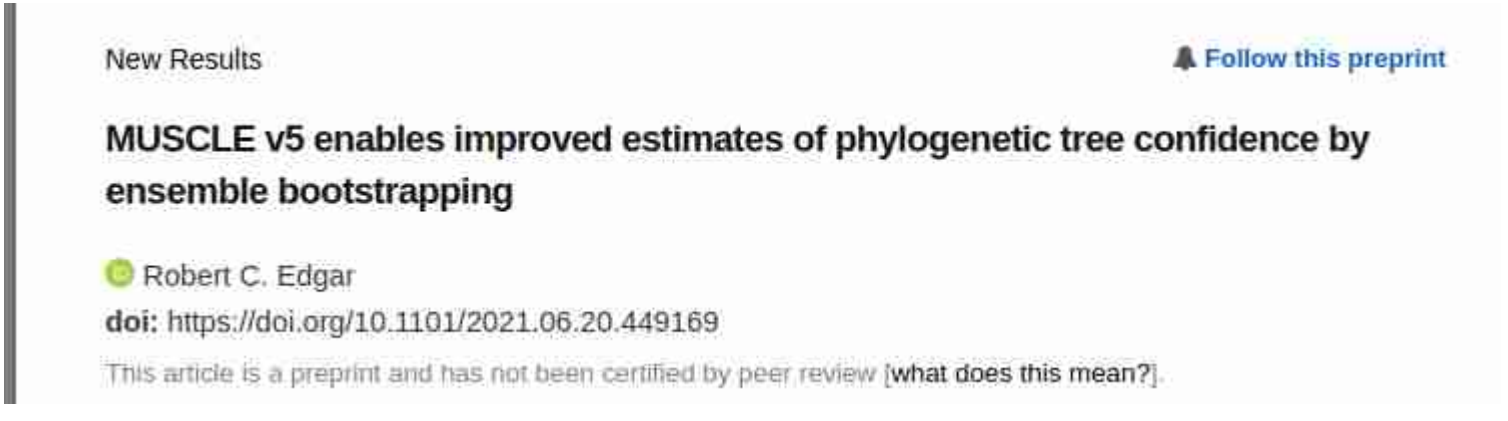
From: “MUSCLE: multiple sequence alignment with high accuracy and high throughput”

MUSCLE Refinement

1. TREE2 is divided into two subtrees by deleting the edge. The profile of the multiple alignment in each subtree is computed.
2. A new multiple alignment is produced by re-aligning the two profiles.
3. An edge/branch is chosen from *TREE2* (edges are visited in order of decreasing distance from the root)
4. If the *SP* score is improved, the new alignment is kept, otherwise it is discarded.
5. Steps 1. - 4. are repeated until convergence or until a user-defined limit is reached.

Alignment Uncertrainty

- The MSA depends heavily on the guide tree
- The MSA depends heavily on the penalty matrix used
- Instead of using a single MSA better use an *ensemble* of MSAs for downstream analyses that captures these two sources of uncertainty
- A recent preprint by the MUSCLE guy:


New Results

Follow this preprint

MUSCLE v5 enables improved estimates of phylogenetic tree confidence by ensemble bootstrapping

Robert C. Edgar

doi: https://doi.org/10.1101/2021.06.20.449169

This article is a preprint and has not been certified by peer review [what does this mean?].

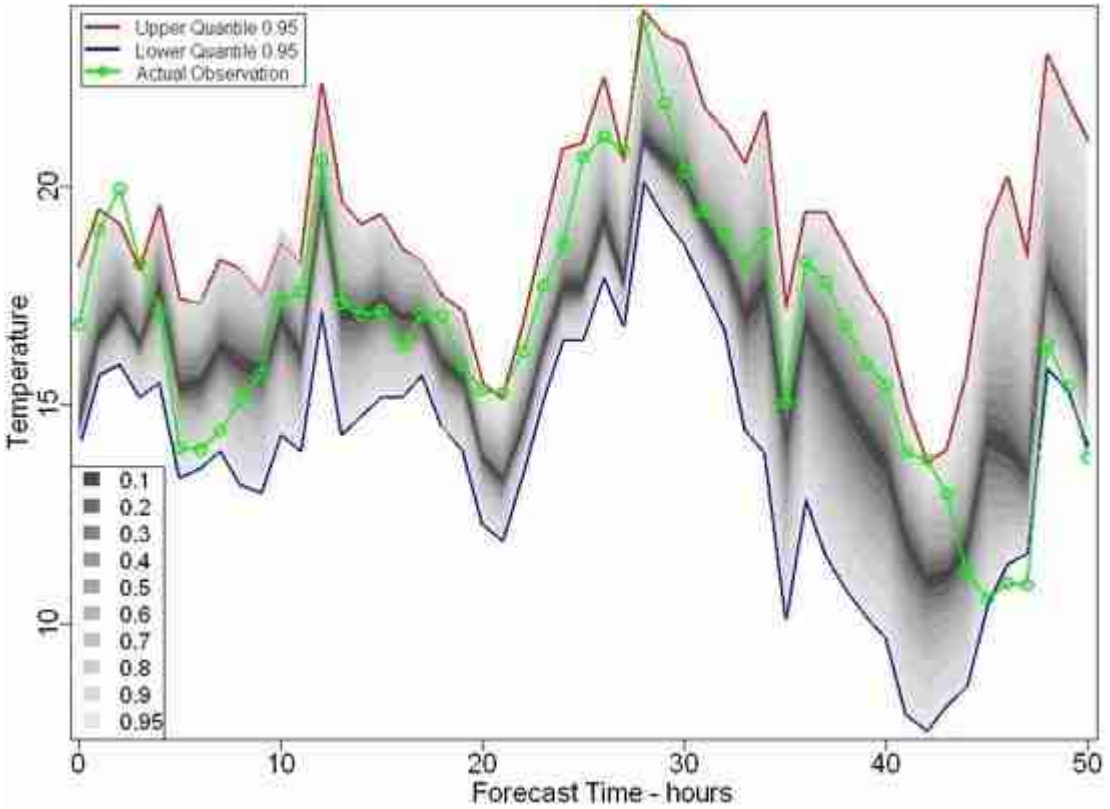
What is an ensemble?

- Ensemble forecasting is a numerical weather prediction method. Instead of making a single forecast of the most likely weather, a set (or ensemble) of forecasts is produced. This set of forecasts aims to give an indication of the range of possible future states of the atmosphere.
- Multiple simulations are conducted to account for uncertainty sources in forecast models:

  (1) errors due to imperfect initial conditions

  (2) errors introduced because of imperfect models
- In general, this approach can be used for probabilistic forecasts of any dynamical system, not just for weather prediction.
- See: https://en.wikipedia.org/wiki/Ensemble_forecasting

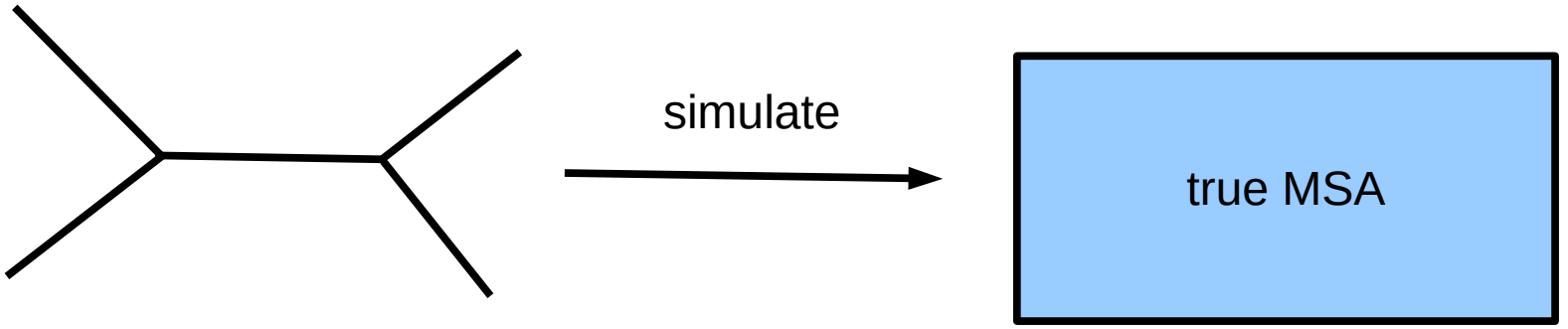
Temperature Forecast Ensemble



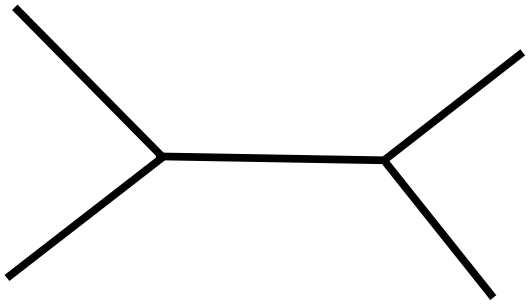
Benchmarking MSAs

- MSA benchmarks → mostly structural protein data that has been manually aligned to reflect the protein structure
  - Databases: BALiBASE 2.0, OXBench, PREFAB, etc
- Simulation
  - → focus on alignment
  - → focus on phylogeny

Simulation



Simulation



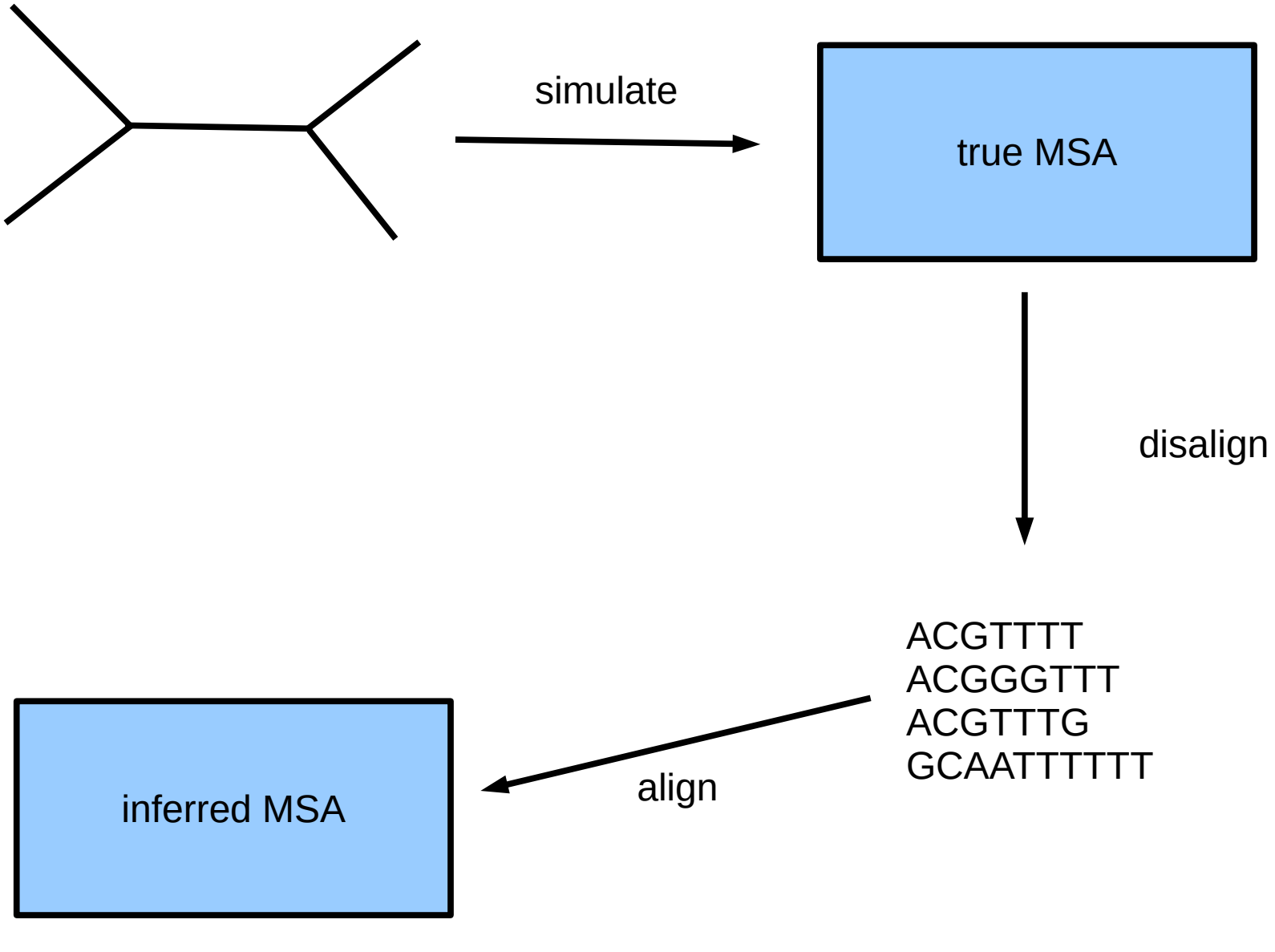
simulate

true MSA

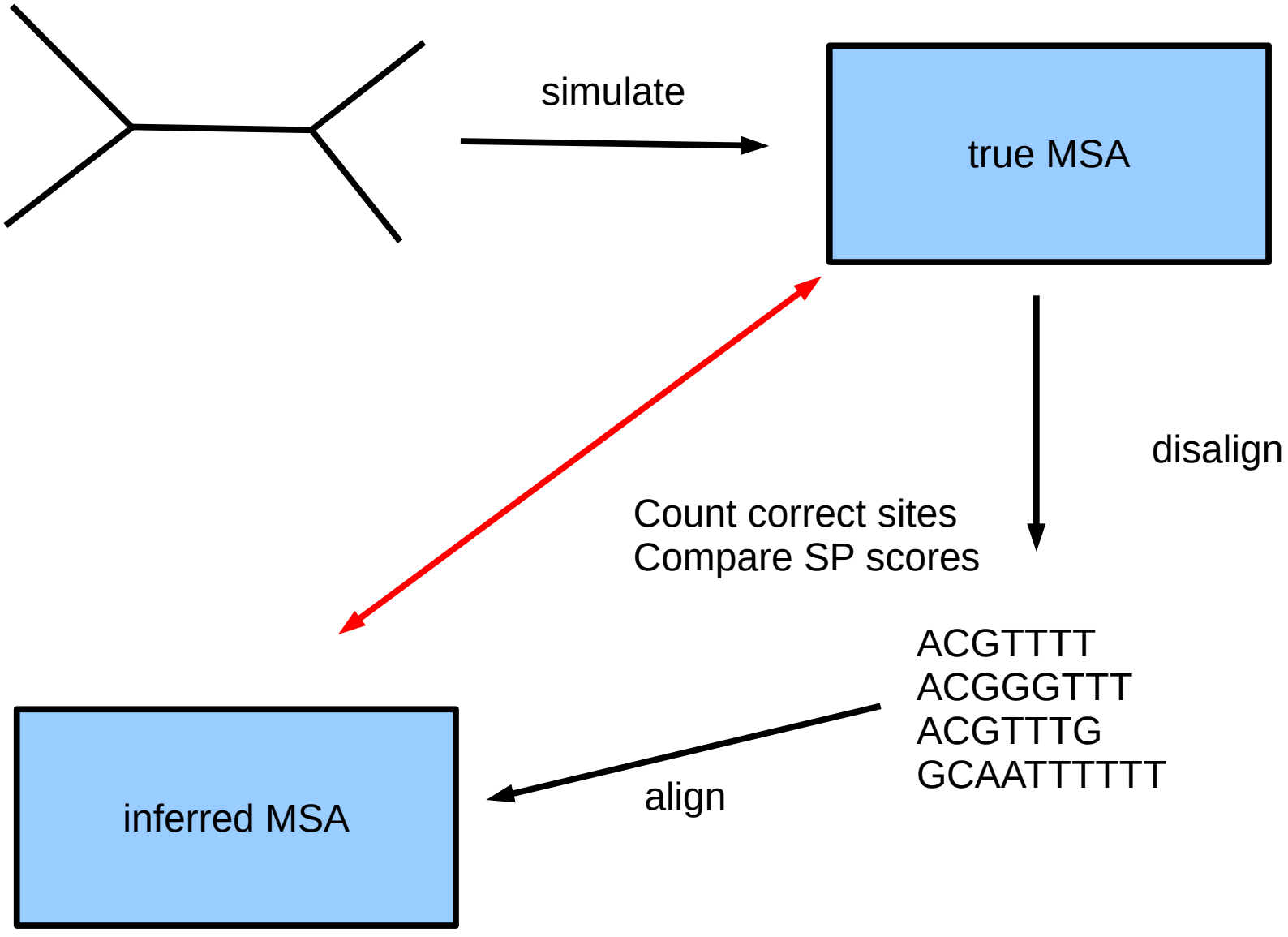
disalign

ACGTTTT
ACGGGTTT
ACGTTTG
GCAATTTTTT

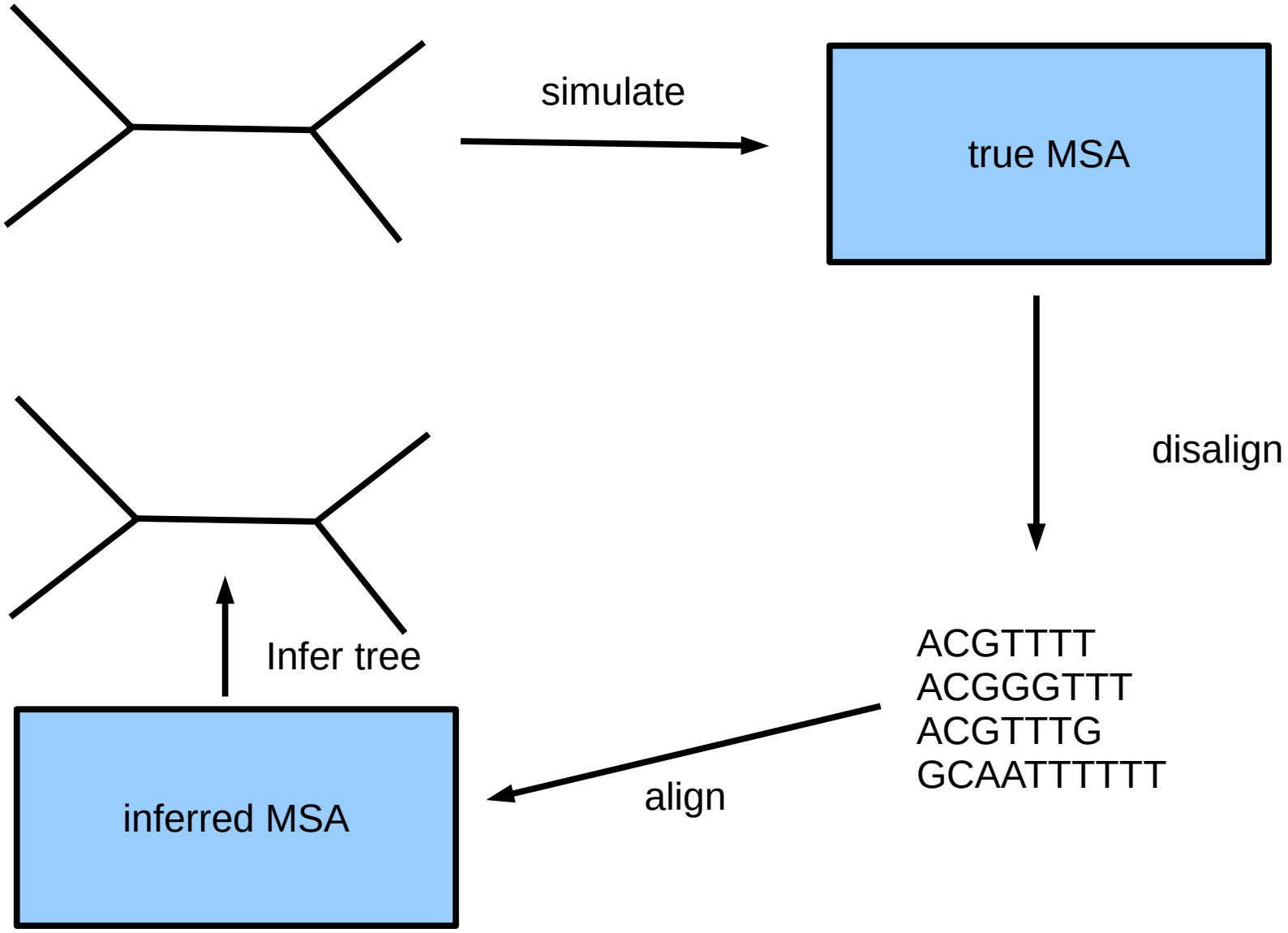
Simulation


simulate
true MSA
disalign
ACGTTTT
ACGGGTTT
ACGTTTG
GCAATTTTTT
align
inferred MSA

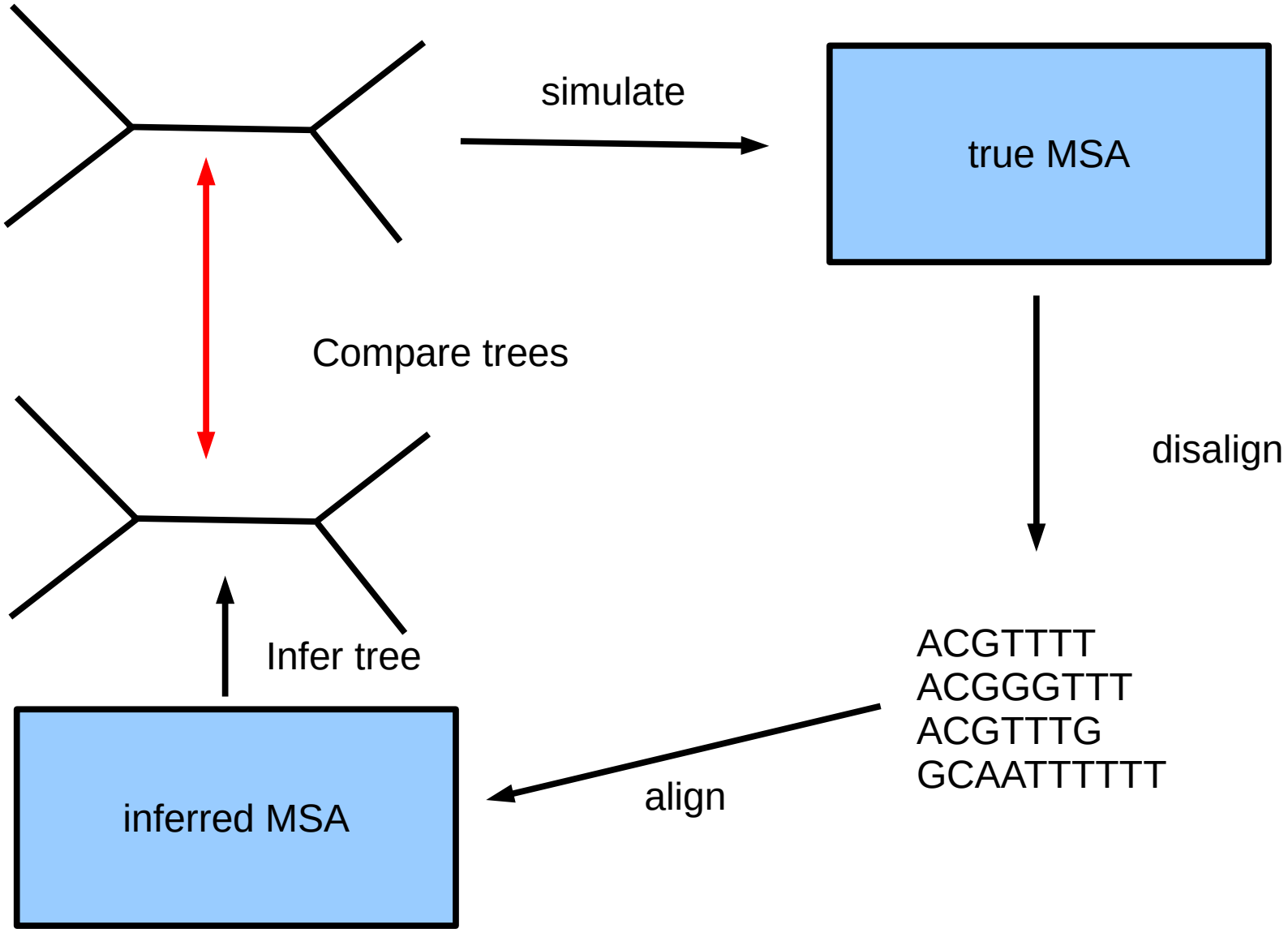
Simulation


simulate
true MSA
disalign
Count correct sites
Compare SP scores
ACGTTTT
ACGGGTTT
ACGTTTG
GCAATTTTTT
align
inferred MSA

Simulation


simulate
true MSA
disalign
ACGTTTT
ACGGGTTT
ACGTTTG
GCAATTTTTT
align
inferred MSA
Infer tree

Simulation


simulate
true MSA
Compare trees
disalign
Infer tree
ACGTTTT
ACGGGTTT
ACGTTTG
GCAATTTTTT
inferred MSA
align

Summary

- MSA is generally difficult due to lack of objective criteria
- MSA as defined per *SP-score* is NP-complete
- Tree-alignment MSA is also NP-complete
- There exist approximation algorithms with performance guarantees
- However, practical approaches use ad hoc heuristics that typically perform better
- Classes of algorithms
  - Progressive MSA
  - Progressive iterative MSA
  - Statistical MSA (not covered)
  - Phylogeny-aware MSA (not covered)
  - Simultaneous MSA & tree inference (not covered)

Time for a break :-)

- Up next: **Introdutcion to Phylogenetics**

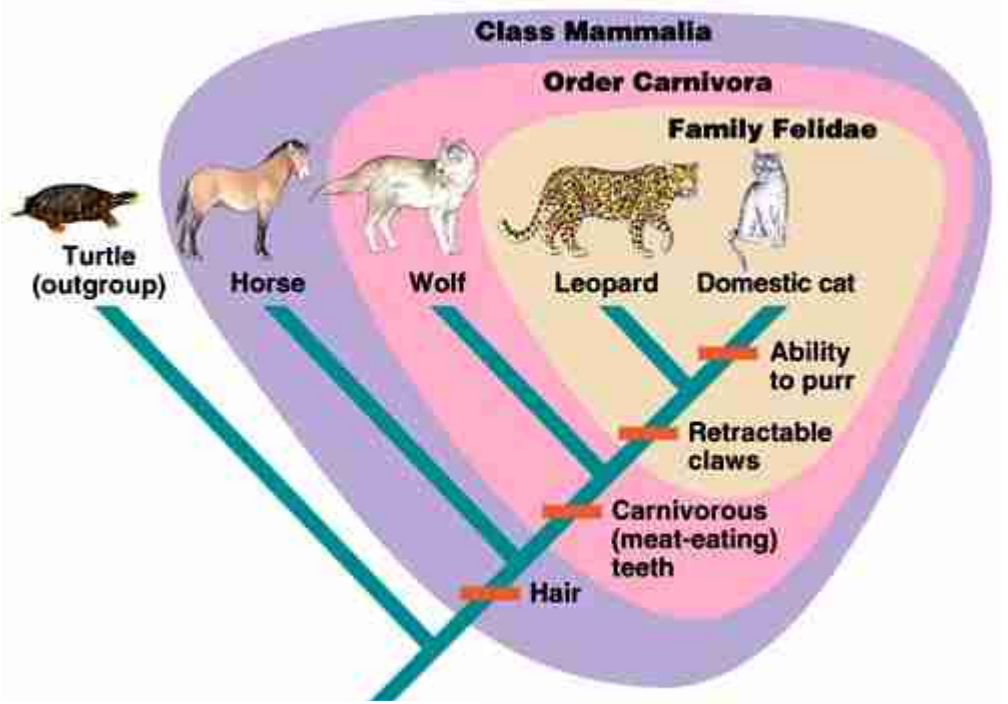
The story so far

- Biological Terminology: RNA, DNA, genes, genomes, etc
- Pair-wise Sequence Alignment
- Sequence Comparison
- Genome Assembly
- Multiple Sequence Alignment

The story so far

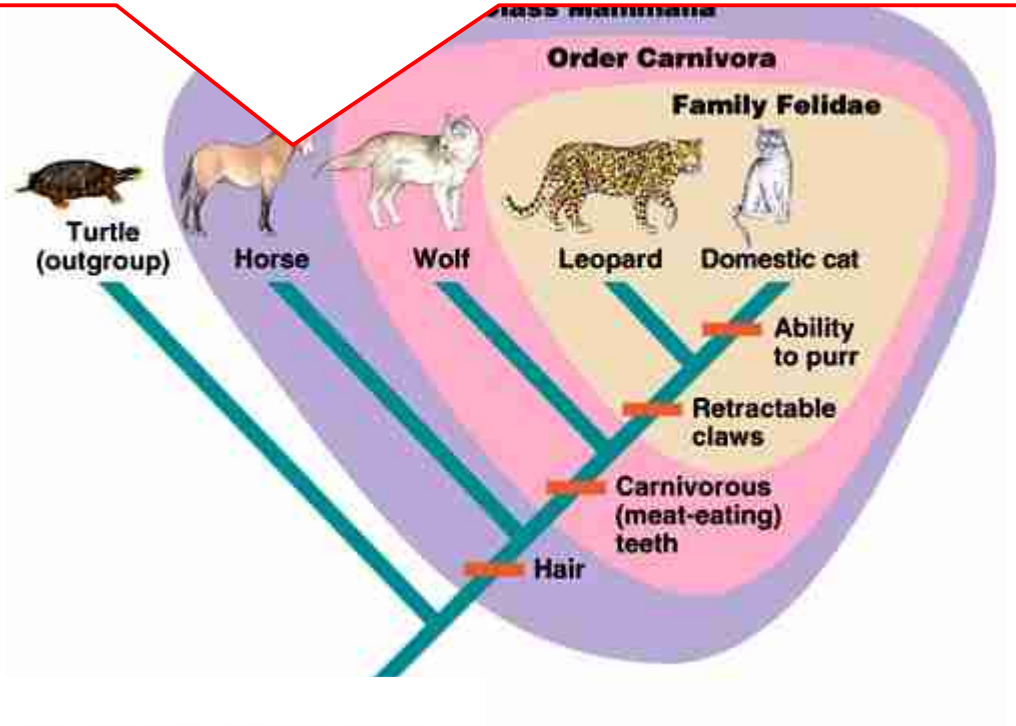
- Biological Terminology: RNA, DNA, genes, genomes, etc
- Pair-wise Sequence Alignment
- Sequence Comparison
- Genome Assembly
- Multiple Sequence Alignment
- Phylogenetic Inference

A Taxonomy


Class Mammalia
Order Carnivora
Family Felidae
Turtle (outgroup)
Horse
Wolf
Leopard
Domestic cat
Ability to purr
Retractable claws
Carnivorous (meat-eating) teeth
Hair

A Taxonomy

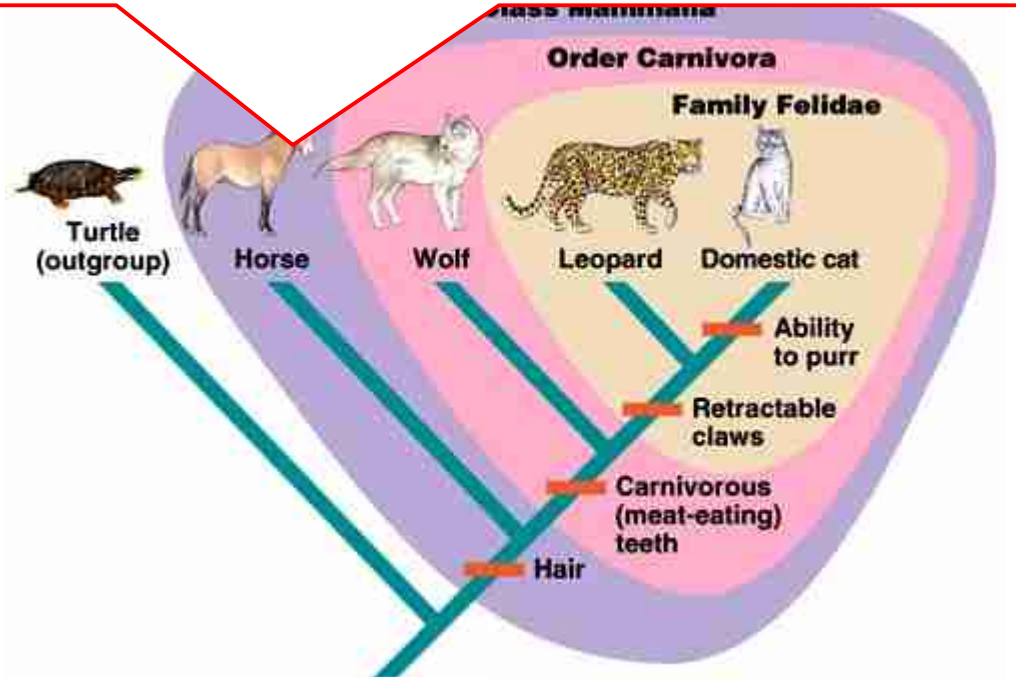
First systematic classification of living beings by Aristotele *384 -382* BC
Some terms still in use today, e.g., classification of animals into *Vertebrates* versus *Invertebrates*



A Taxonomy

First systematic classification of living beings by Aristotele *384 -382* BC
Some terms still in use today, e.g., classification of animals into *Vertebrates* versus *Invertebrates*

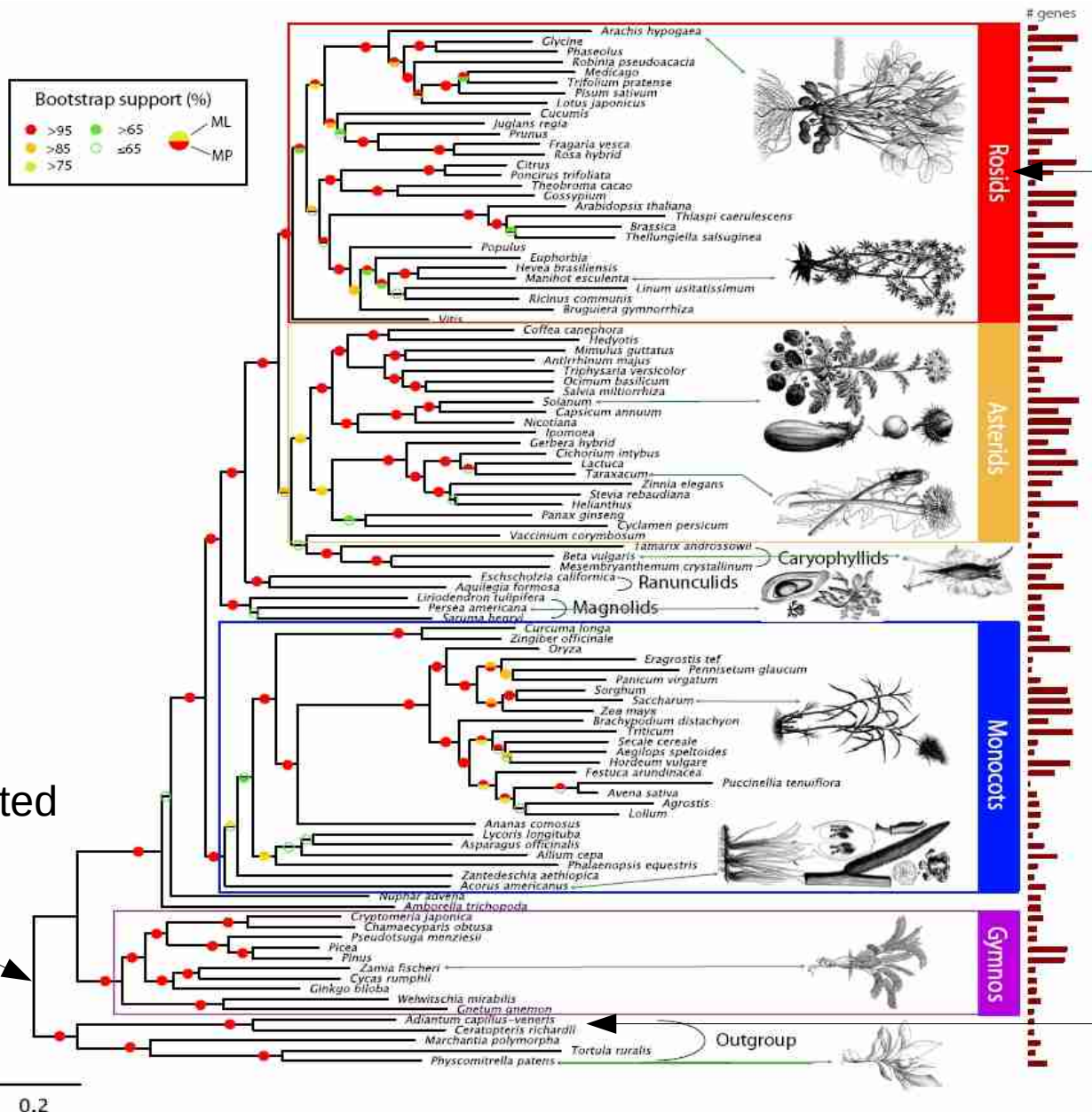
Wirbeltiere



Taxonomy

- Group biological organisms (species) into groups with similar characteristics
- Define characteristics of groups at different hierarchy levels, e.g., animals > mammals > great apes
- Taxonomic ranks
  - Domain → three domains of life
  - Kingdom
  - Phylum
  - Class
  - Order
  - Family
  - Genus
  - Species

A Phylogeny or Phylogenetic Tree



A taxonomic subclass

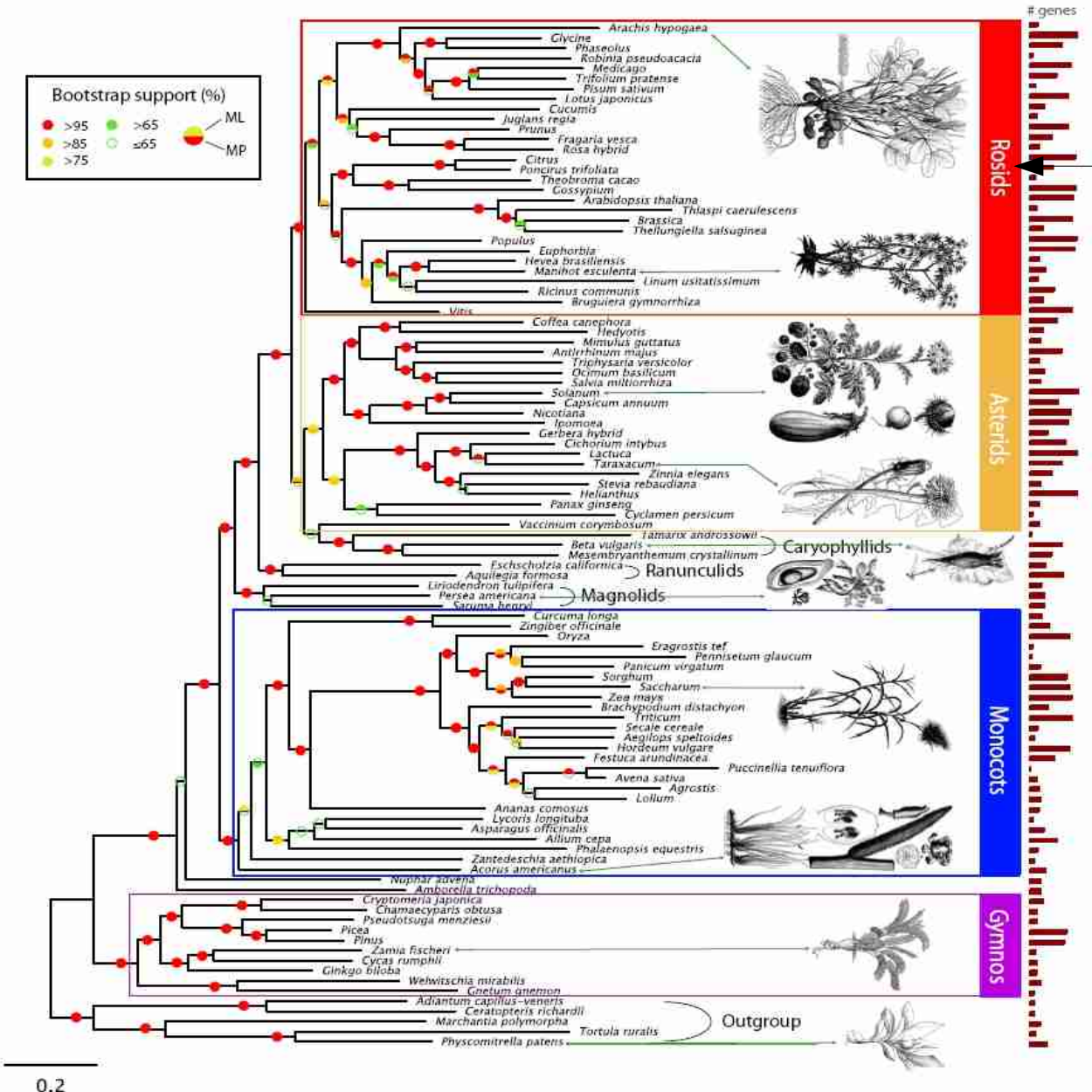
The ingroup

This tree is unrooted

99

The outgroup

A Phylogeny or Phylogenetic Tree



In Phylogenetics such a subtree is often also called *Lineage*!

Phylogeny

- An unrooted strictly binary tree
- Leafs are labeled by *extant* (currently living) organisms represented by their DNA/Protein sequences

  → we can also sequence ancient DNA, see, for instance, the neandertal genome: “The complete genome sequence of a Neanderthal from the Altai Mountains”, *Nature* 2013

  → depends on temperature, time, and other environmental conditions

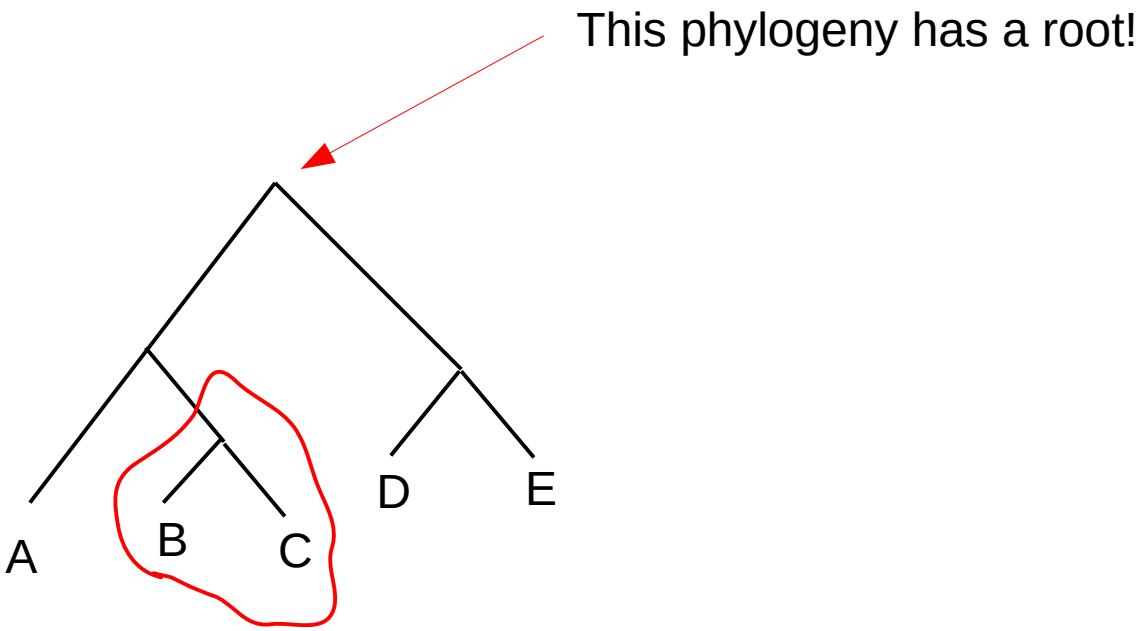
  → up to 300,000 years back, see

  http://www.pnas.org/content/110/39/15758.abstract
- Inner nodes represent *hypothetical common ancestors*
- *Outgroup*: one or more closely related, but different species → allows to root the tree

Taxon

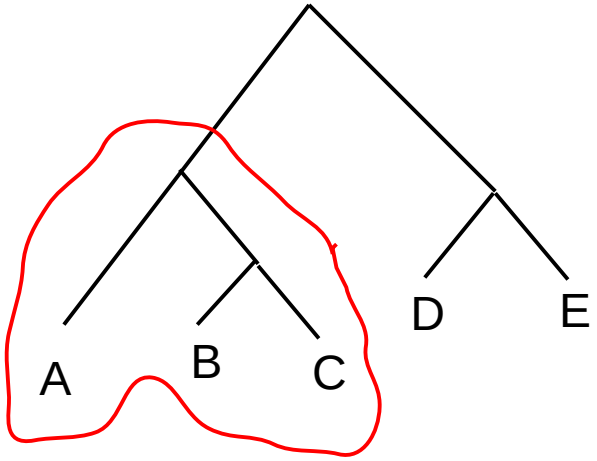
- Used to denote clades/subtrees in phylogenies or taxonomies
- A group of one or more species that form a biological unit
- As defined by taxonomists
  - → subject of controversial debates
  - → part of the culture/fuzziness of Biology
- In phylogenetics we often refer to a single leaf as taxon
  - → the plural of taxon is *taxa*
  - → we often say that a tree with *n* leaves (sequences) has *n* taxa

Some more terminology



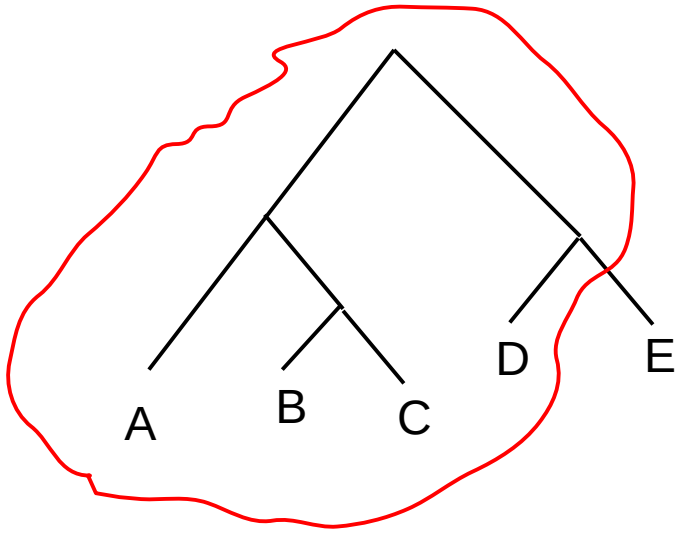
B and **C** are a *monophyletic* group; they are sister species

Some more terminology



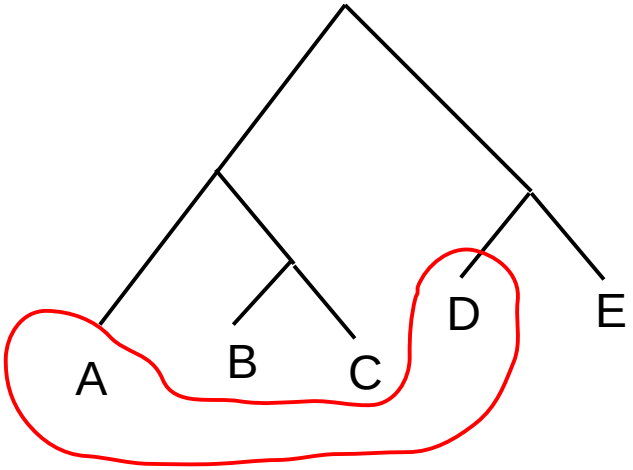
(A,B,C) is a *monophyletic* group; it is sister to **(D, E)**

Some more terminology



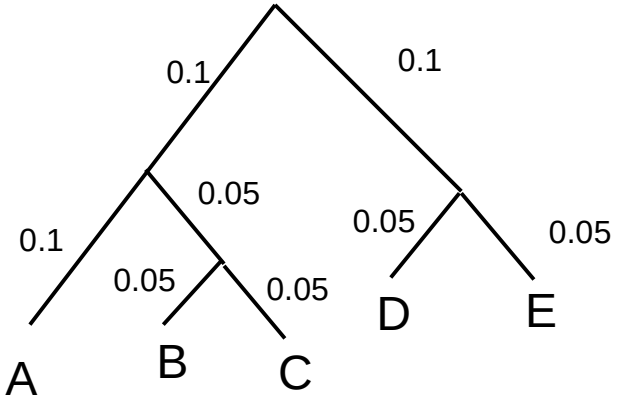
(A,B,C,D) is *paraphyletic* → **E** is excluded

Some more terminology



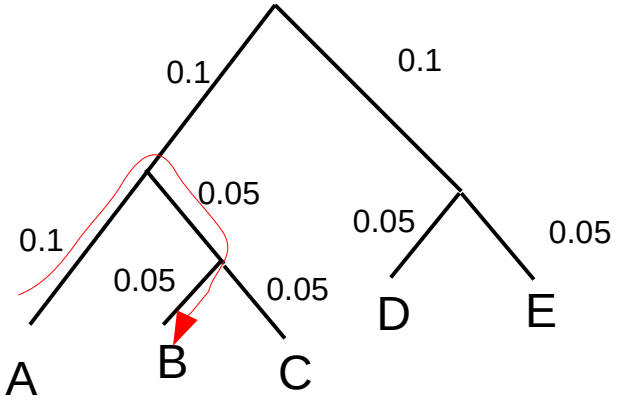
(A,D) is a *polyphyletic* group → their most recent common ancestor (MRCA) is excluded

Some more terminology



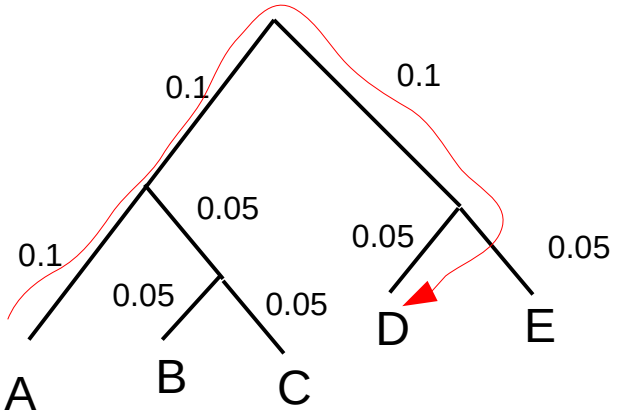
Tree-based or *patristic distance* between two taxa:
Sum over branch lengths along the path in the tree, e.g.:

Some more terminology



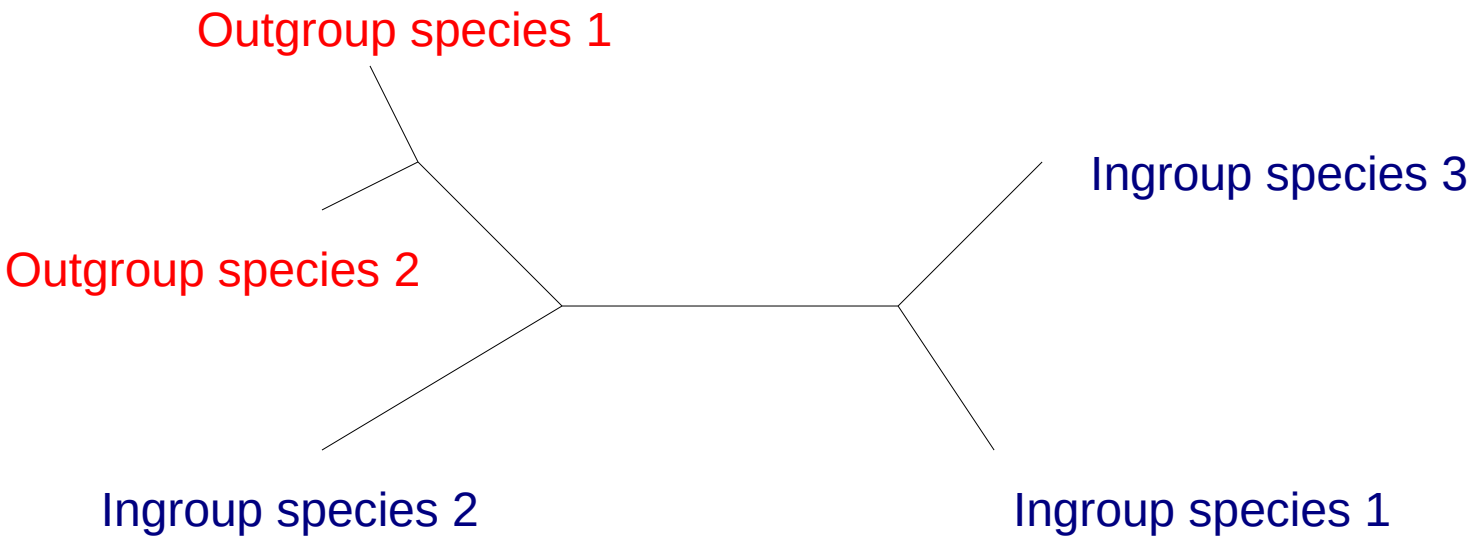
Tree-based or *patristic distance* between two taxa:
Sum over branch lengths along the path in the tree, e.g.:
A ↔ **B**: 0.2

Some more terminology

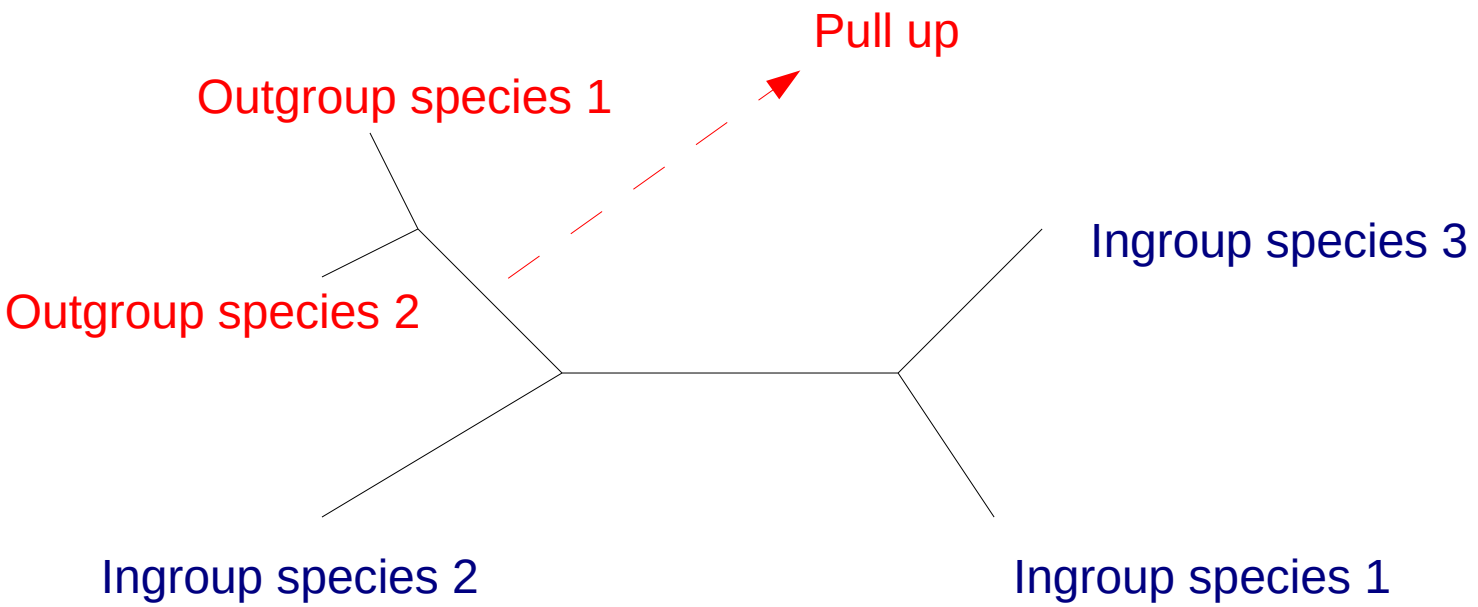


Tree-based or *patristic distance* between two taxa:
Sum over branch lengths along the path in the tree, e.g.:
A ↔ **B**: 0.2
A ↔ **D**: 0.35

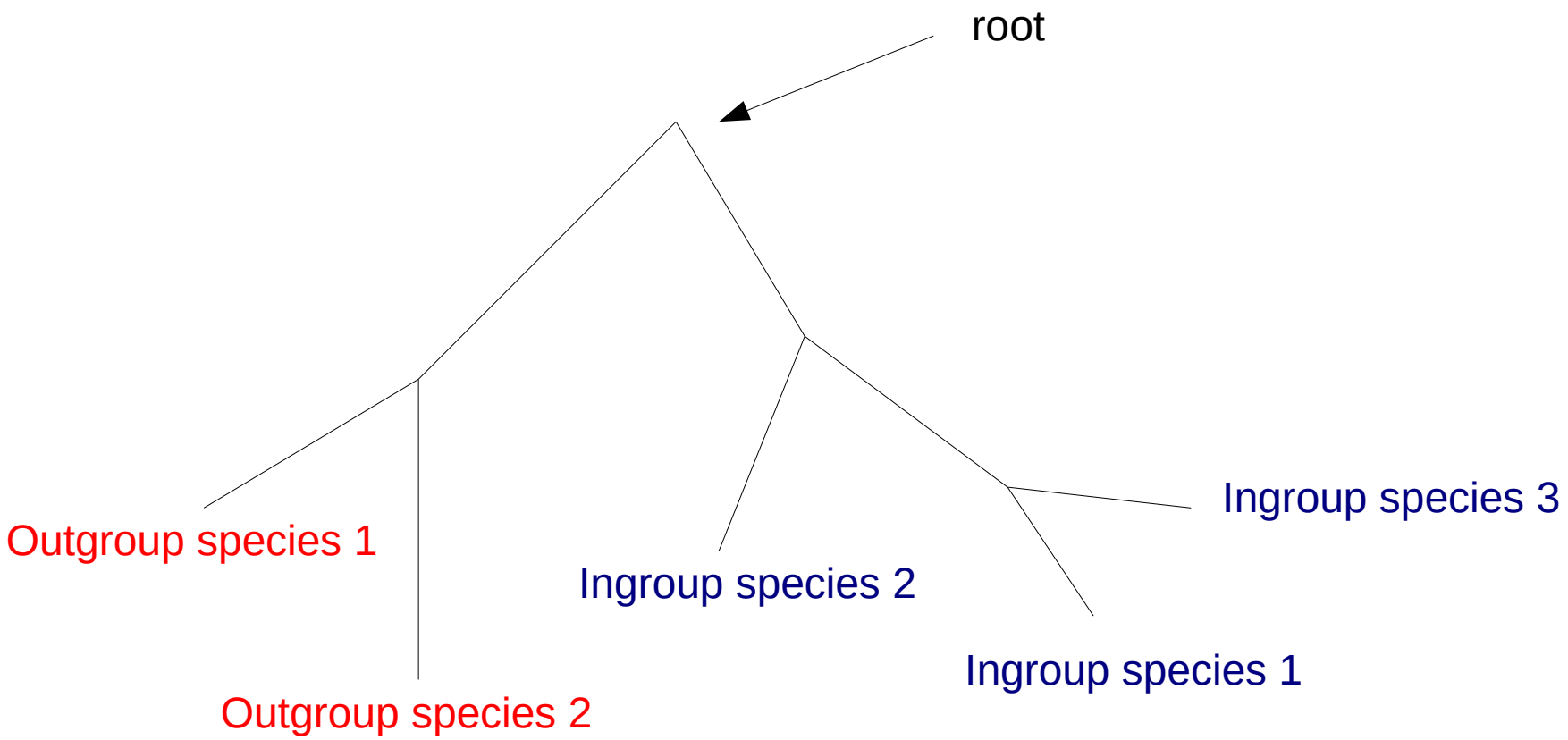
Tree Rooting



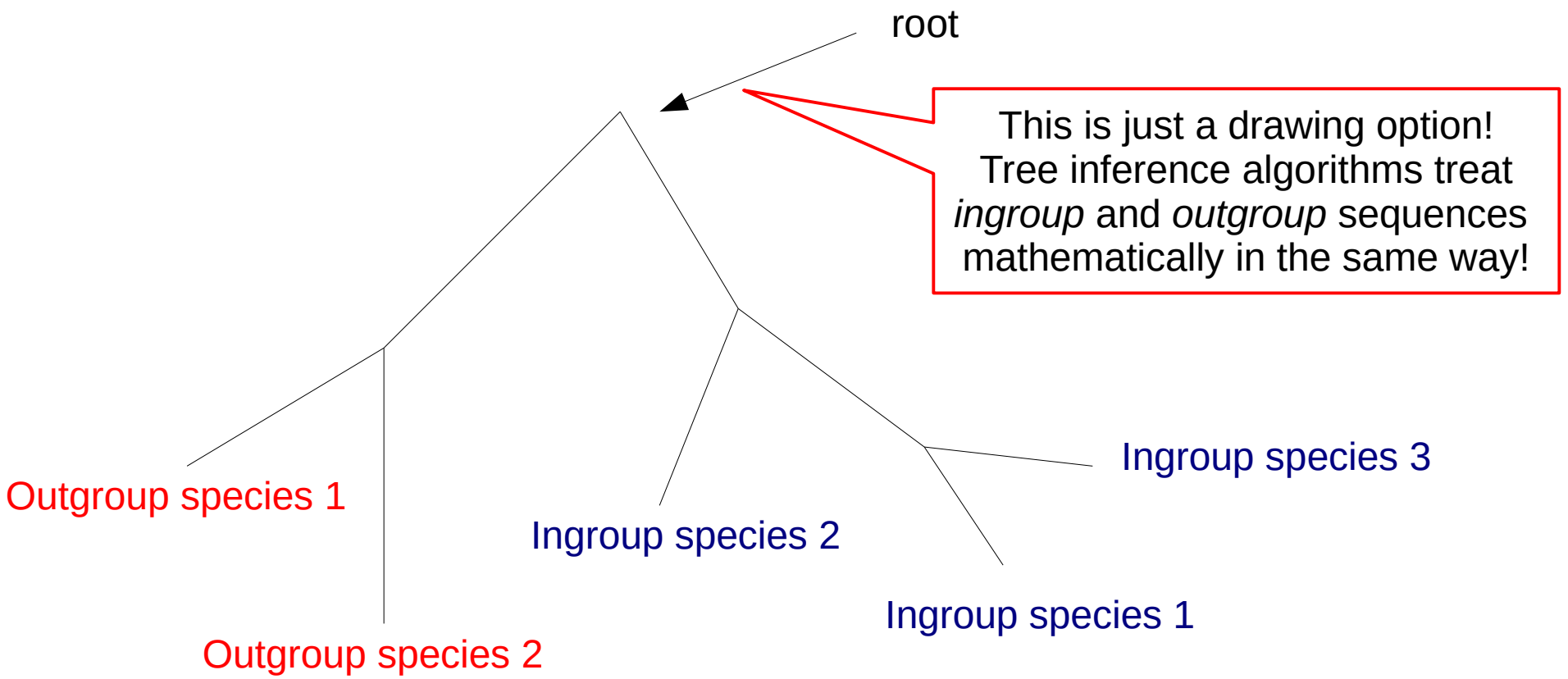
Tree Rooting



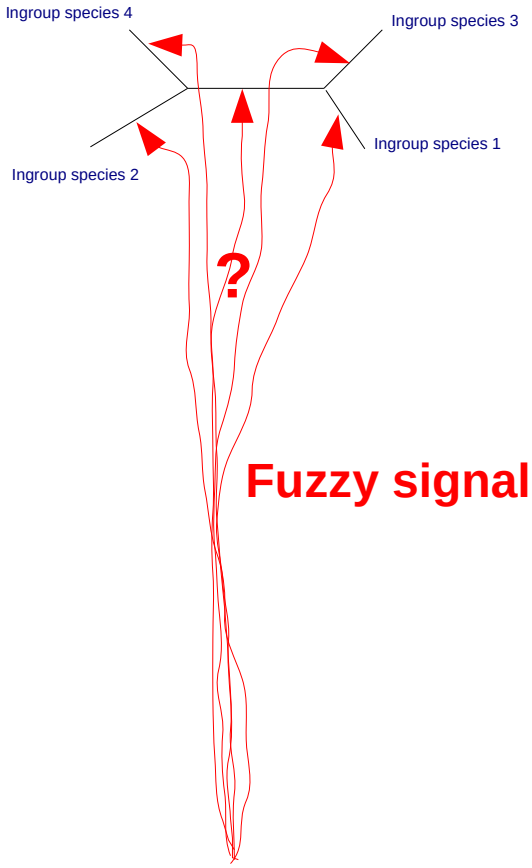
Tree Rooting

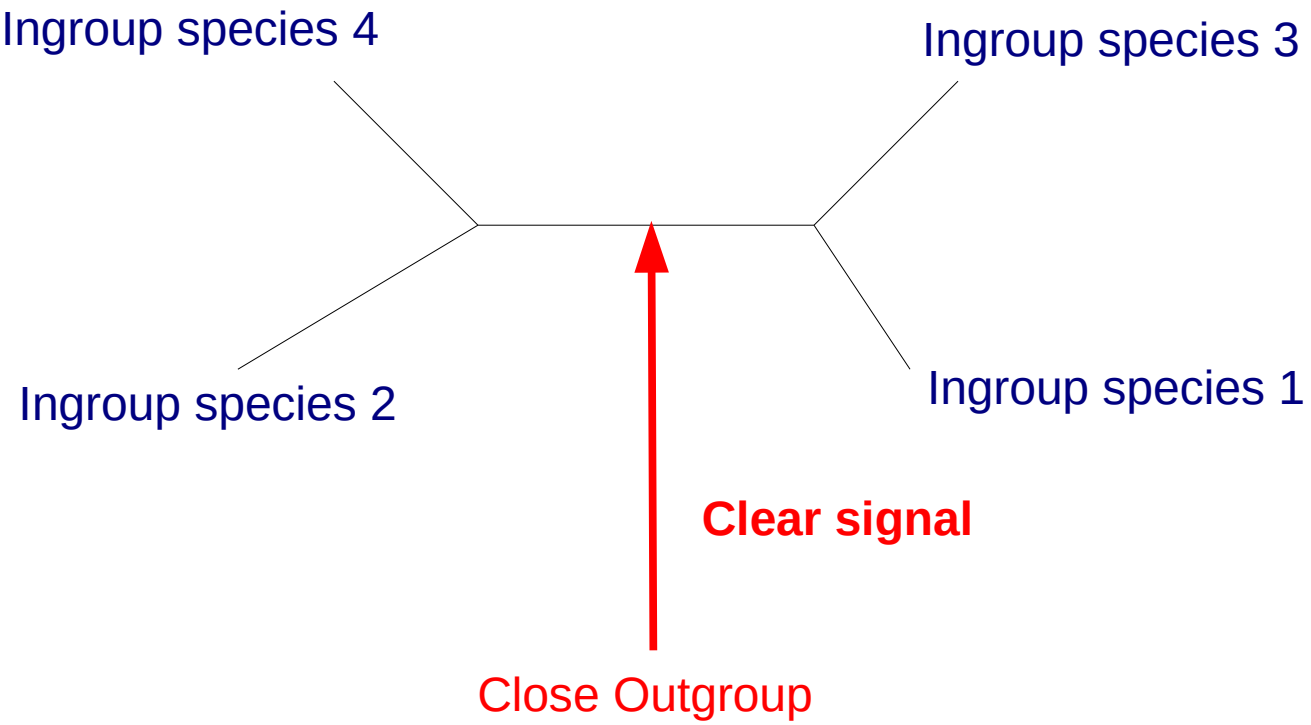


Tree Rooting

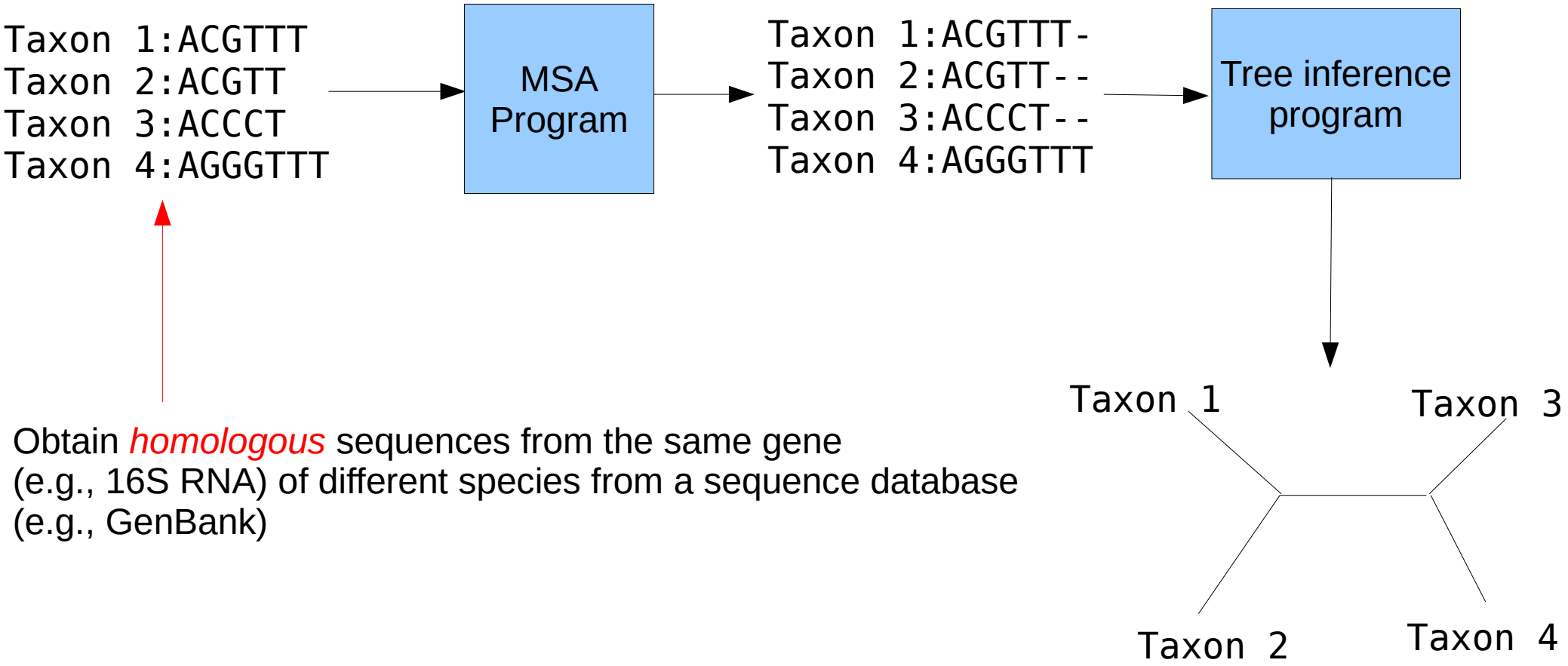


Outgroup Choice



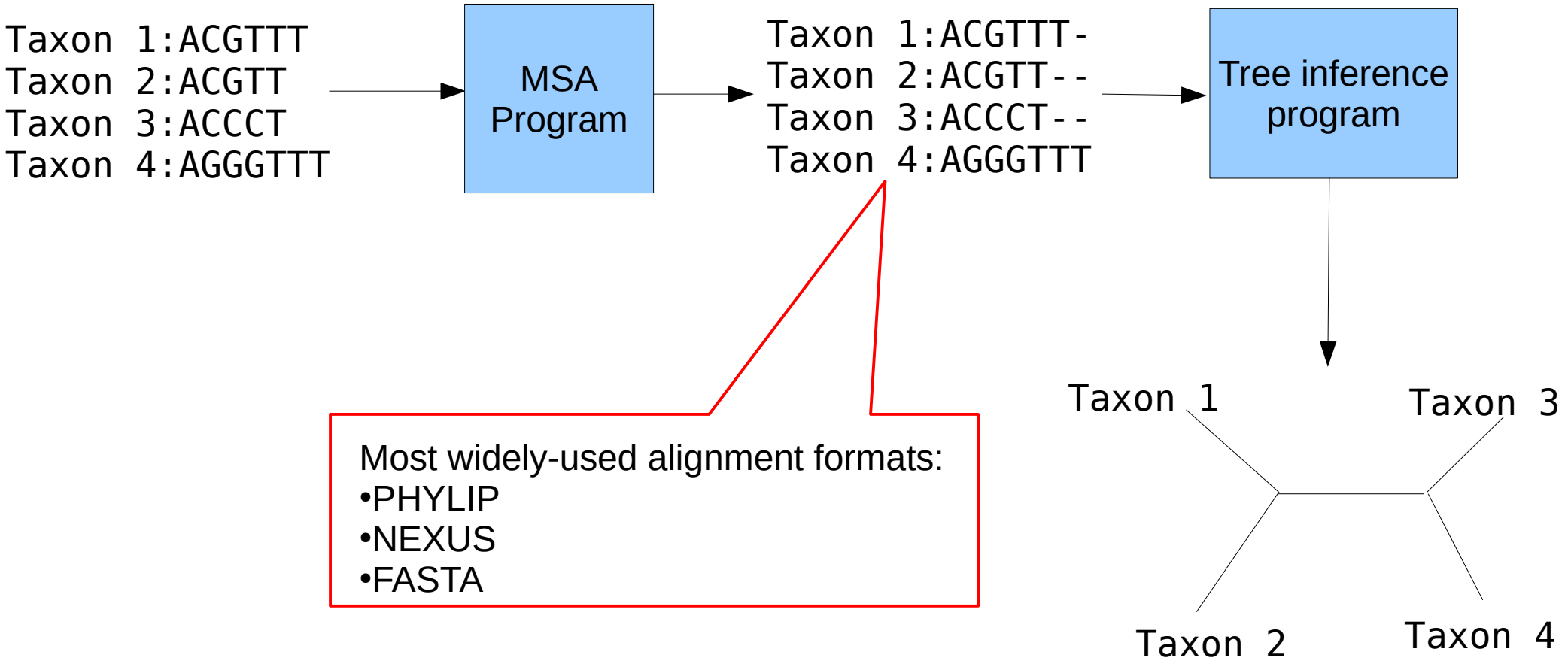


Tree Inference

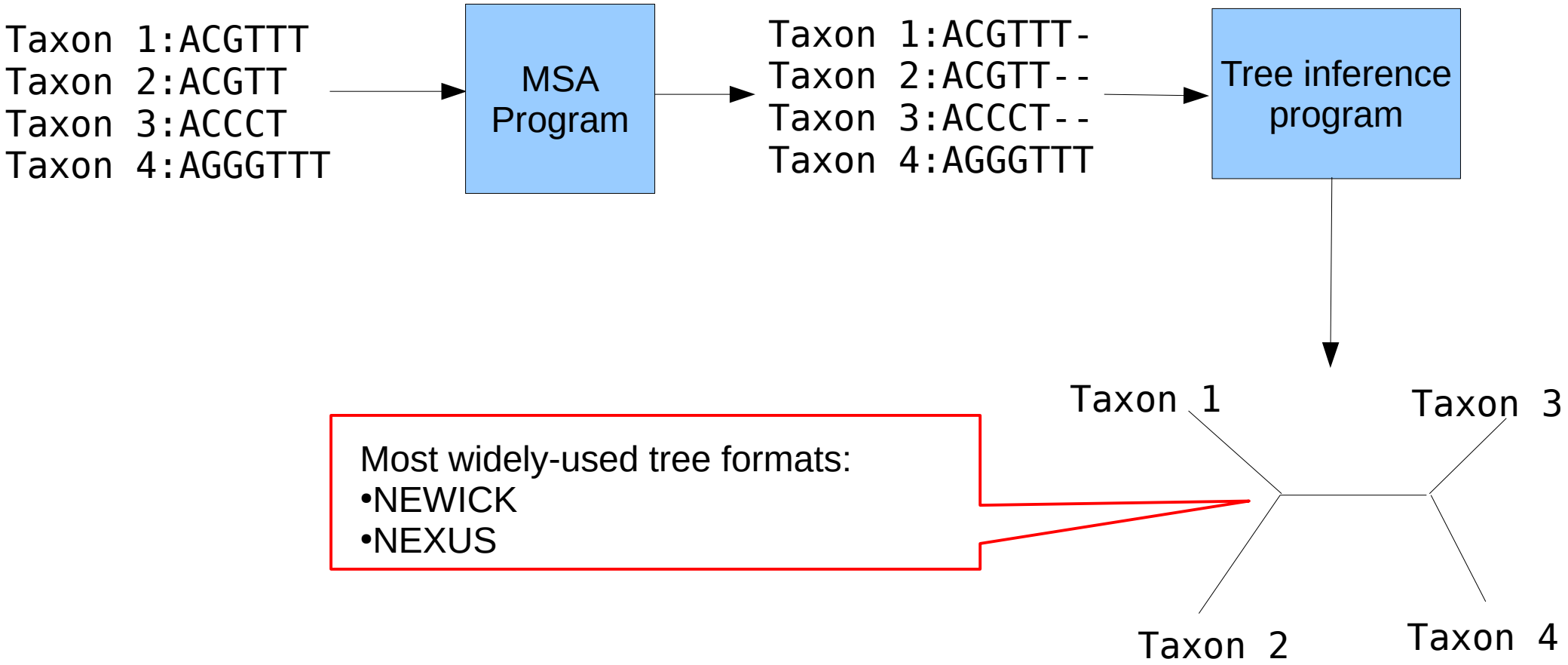


Obtain *homologous* sequences from the same gene
(e.g., 16S RNA) of different species from a sequence database
(e.g., GenBank)

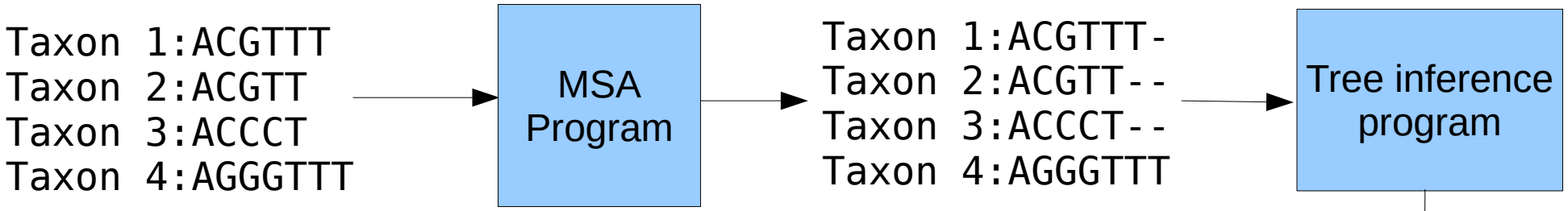
Tree Inference



Tree Inference



Tree Inference

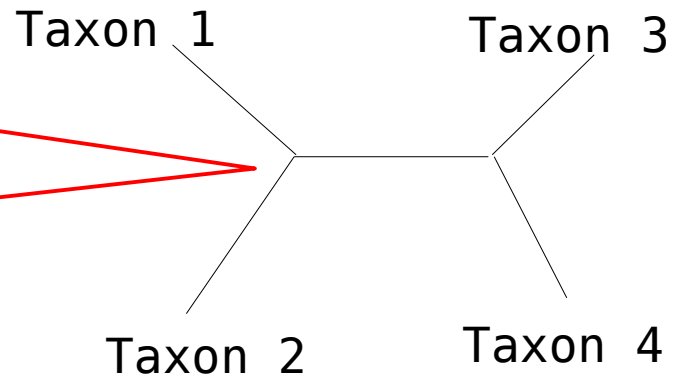


Newick example: Remember that this is an unrooted tree!

(Taxon1, Taxon2, (Taxon3,Taxon4));

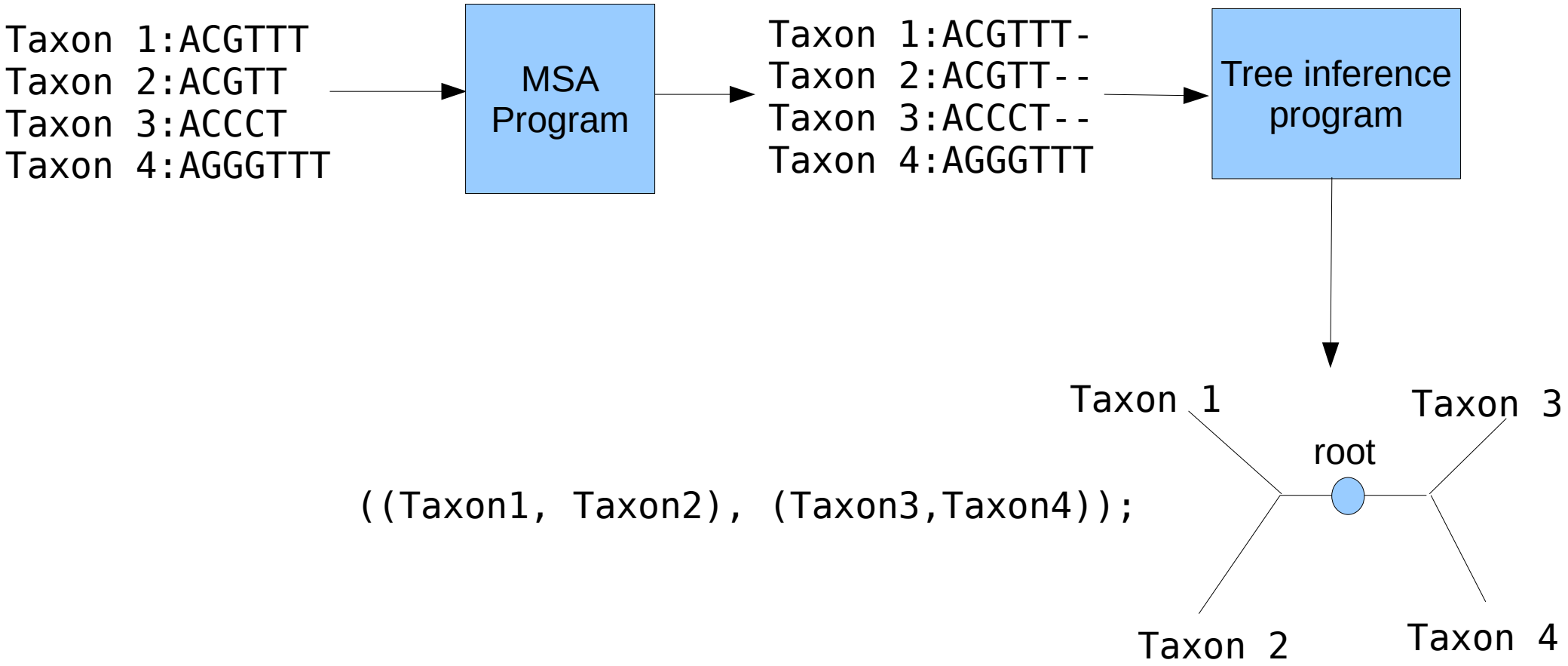
or

((Taxon1, Taxon2), Taxon3,Taxon4);



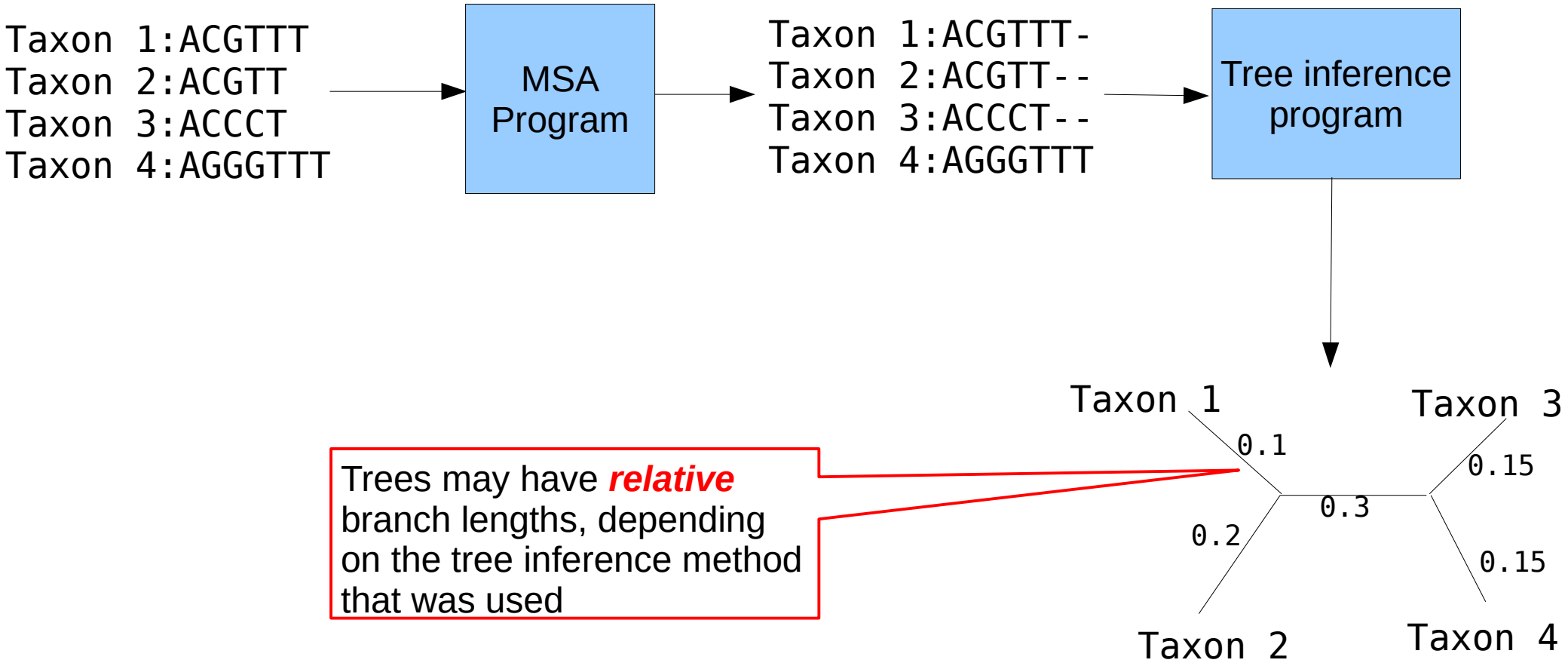
Top level trifurcation

Tree Inference

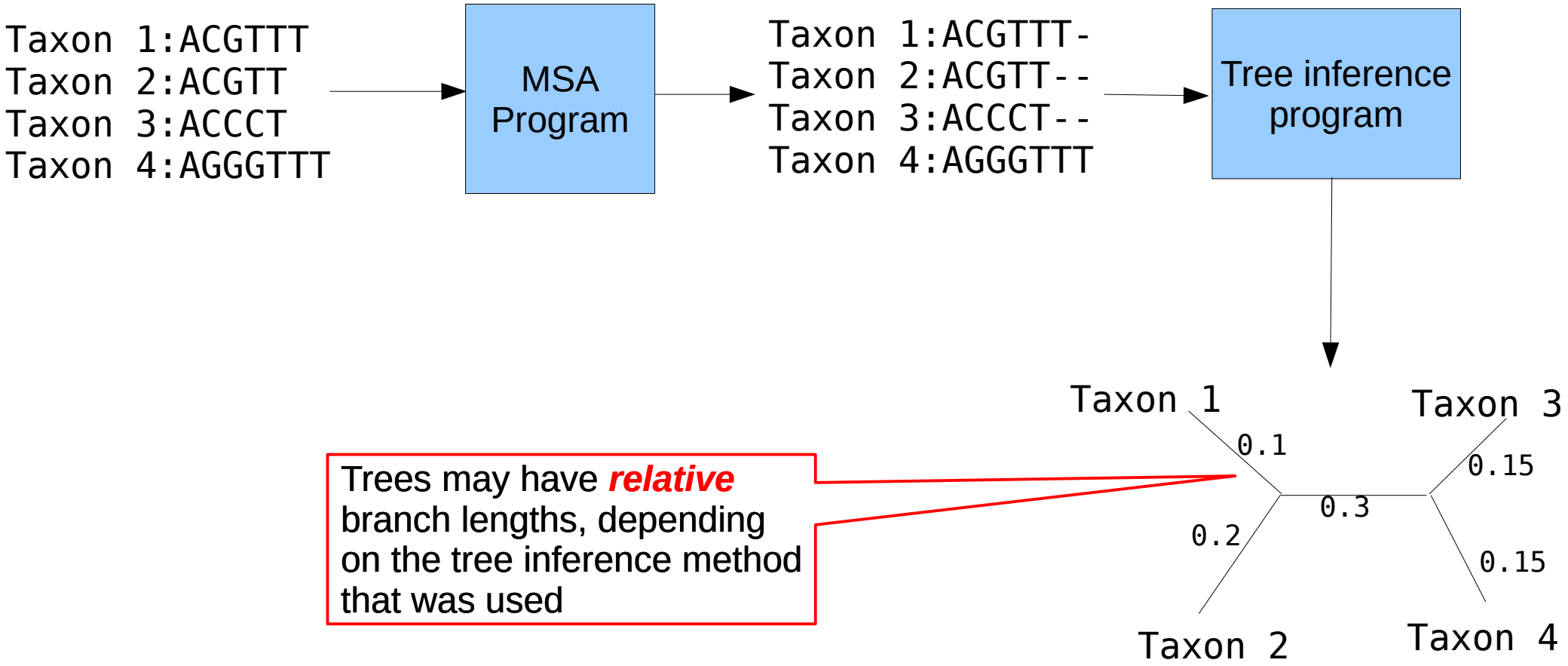


((Taxon1, Taxon2), (Taxon3,Taxon4));

Tree Inference



Tree Inference



Newick format with branch lengths:

```
(Taxon1:0.1,Taxon2:0.2,(Taxon3:0.15,Taxon4:0.15):0.3);
```

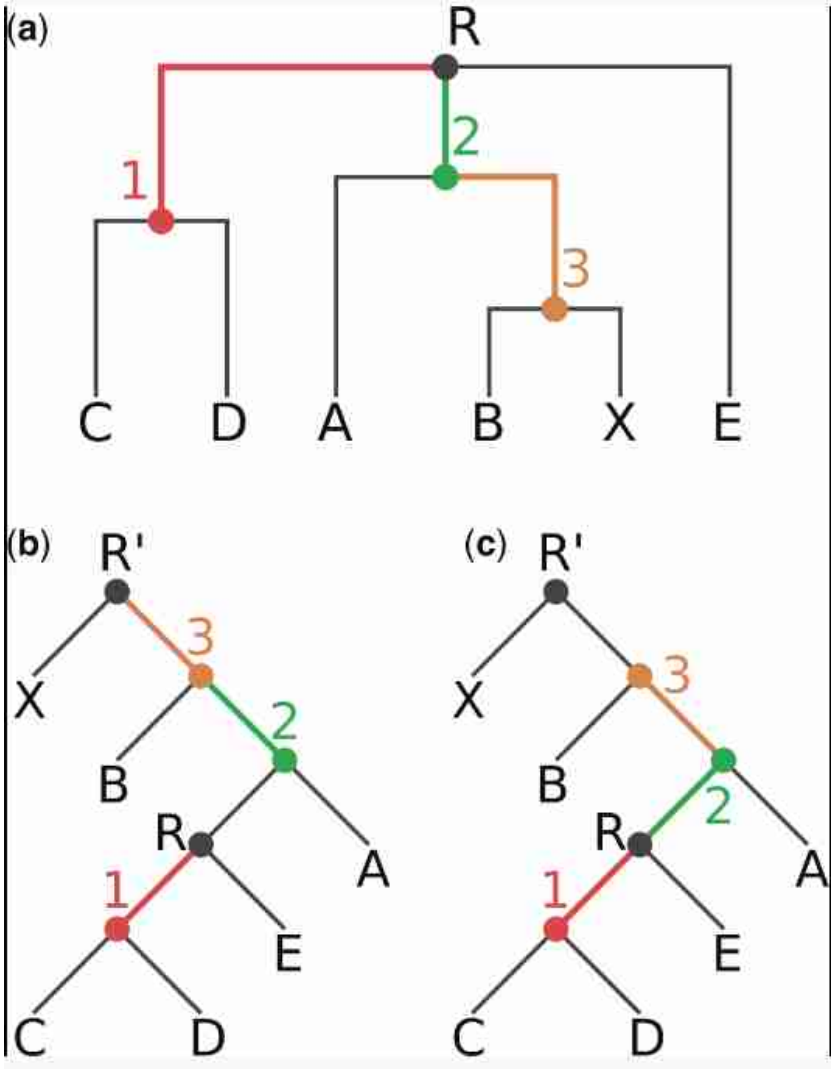
Problems with Newick tree format

- Except for branch length values: no way to associate meta-data to branch lengths
- However, there is important meta-data, e.g., branch support: how well is a branch in the tree supported?

  → ad hoc solution: represent branch support values as node meta-data!

  → this causes problems

Problems with Newick tree format

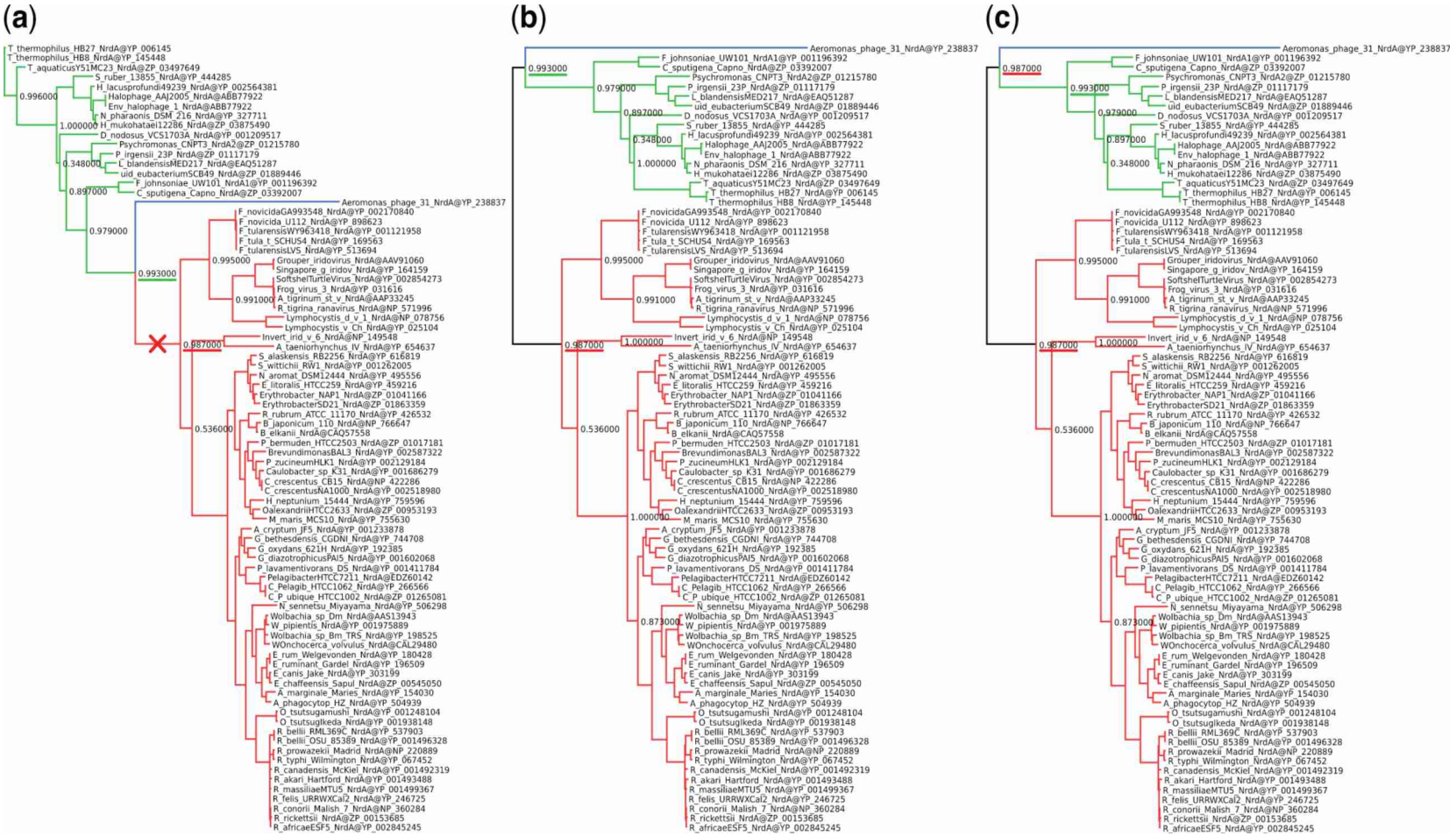


Branch support values represented as node meta-data can be assigned incorrectly to branches after re-rooting.

About 50% of the tools we checked had this Problem. For details see: https://academic.oup.com/mbe/article/34/6/1535/3077051

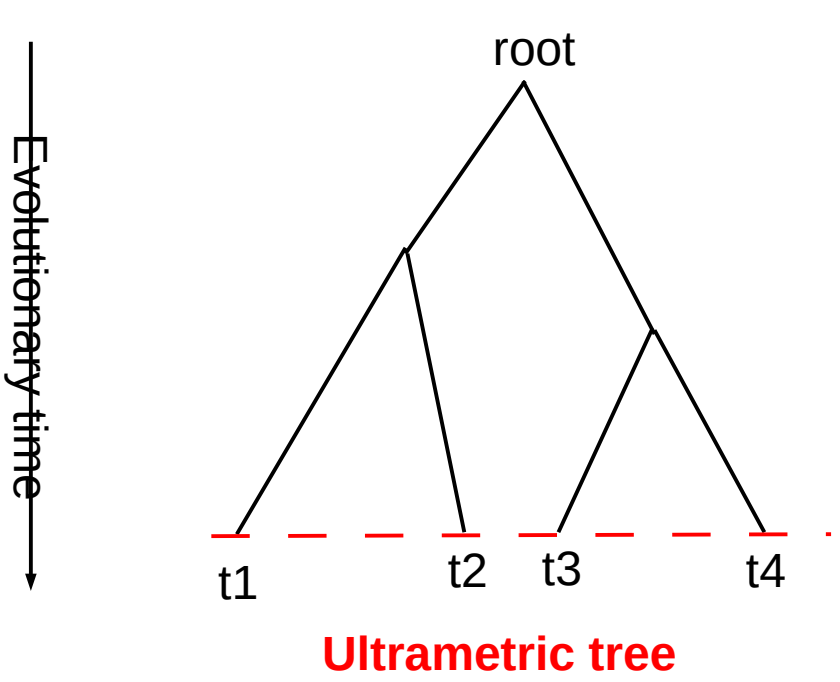
Which representation is correct?

A real example

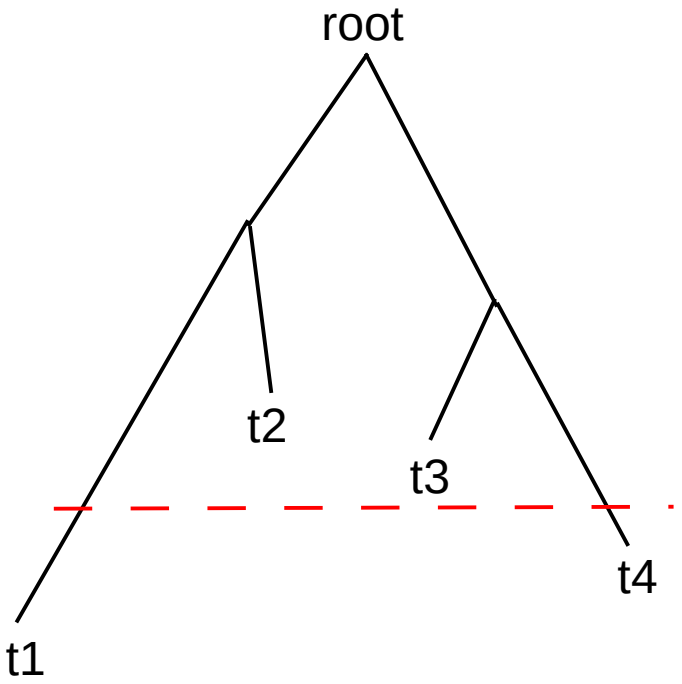


a) original tree
b) re-rooted tree with shifted support values
c) re-rooted tree with correct support values

Tree Shapes

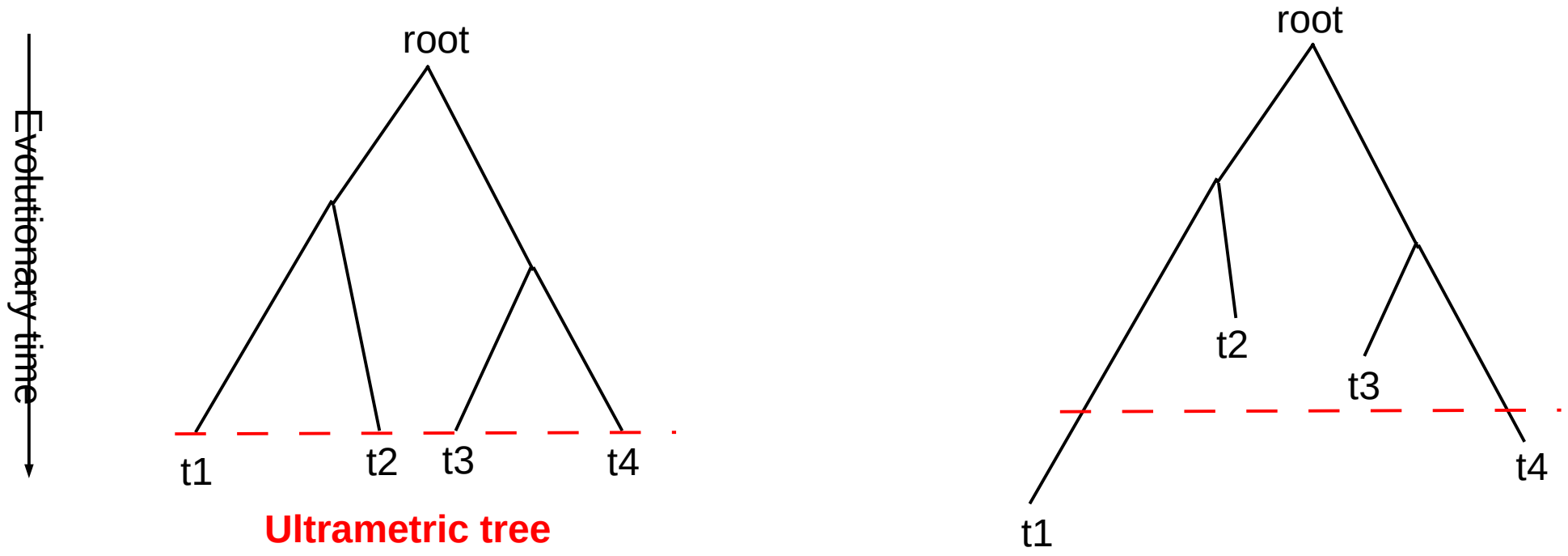


Ultrametric tree

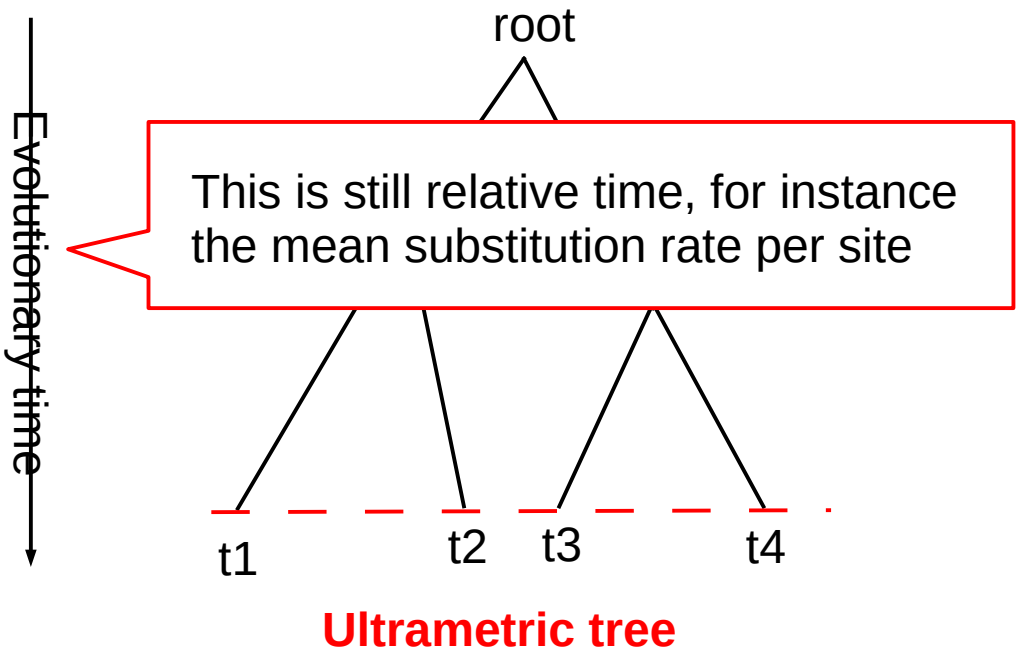


Non-ultrametric tree

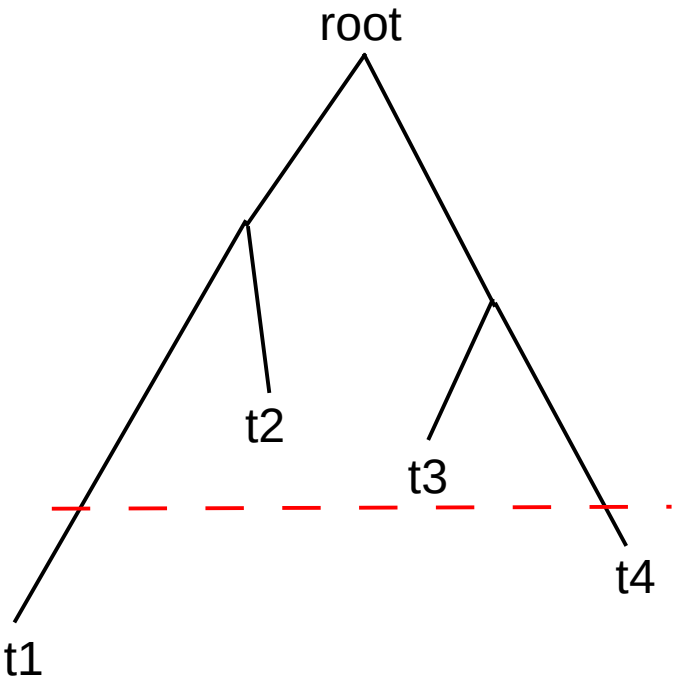
Tree Shapes


Evolutionary time
root
t1
t2
t3
t4
Ultrametric tree
root
t2
t3
t1
t4
Non-ultrametric tree
Most tree inference
models/algorithms/programs
produce non-ultrametric trees

Tree Shapes

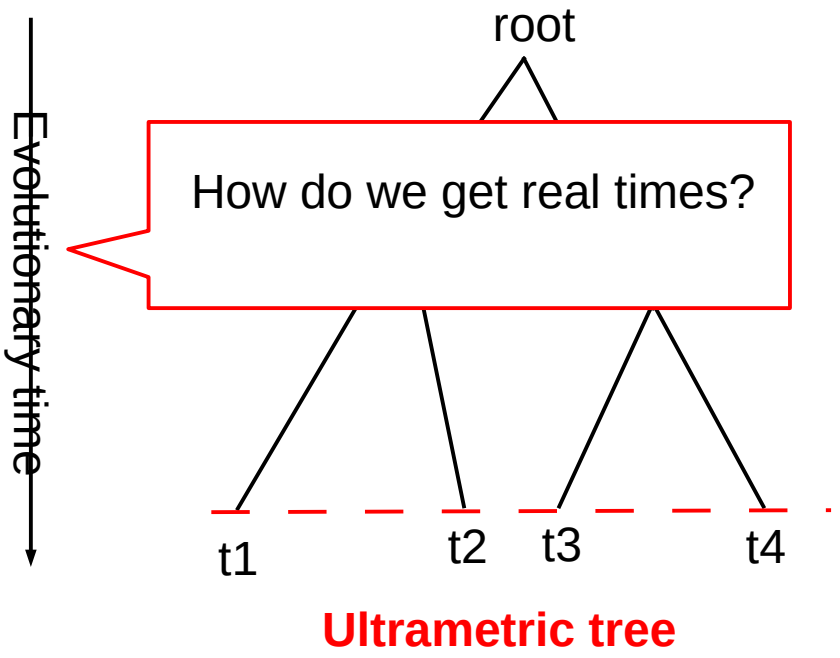


Ultrametric tree

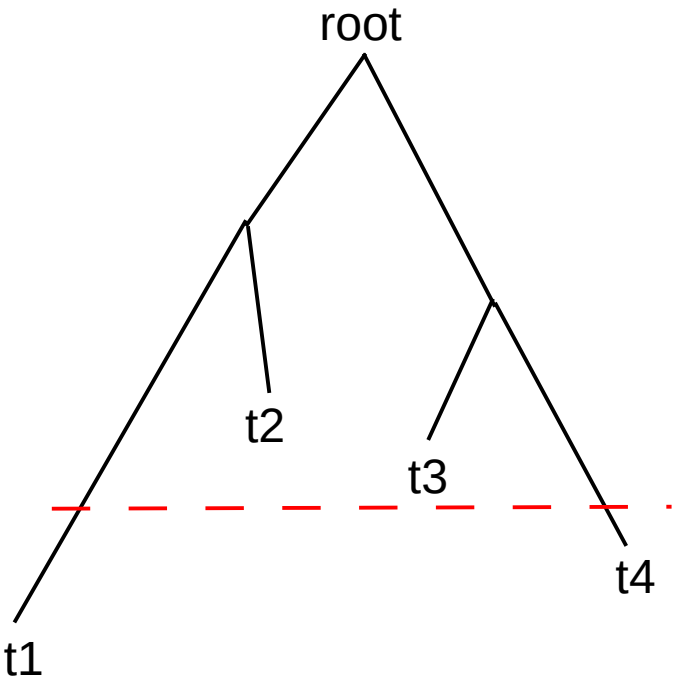


Non-ultrametric tree

Tree Shapes

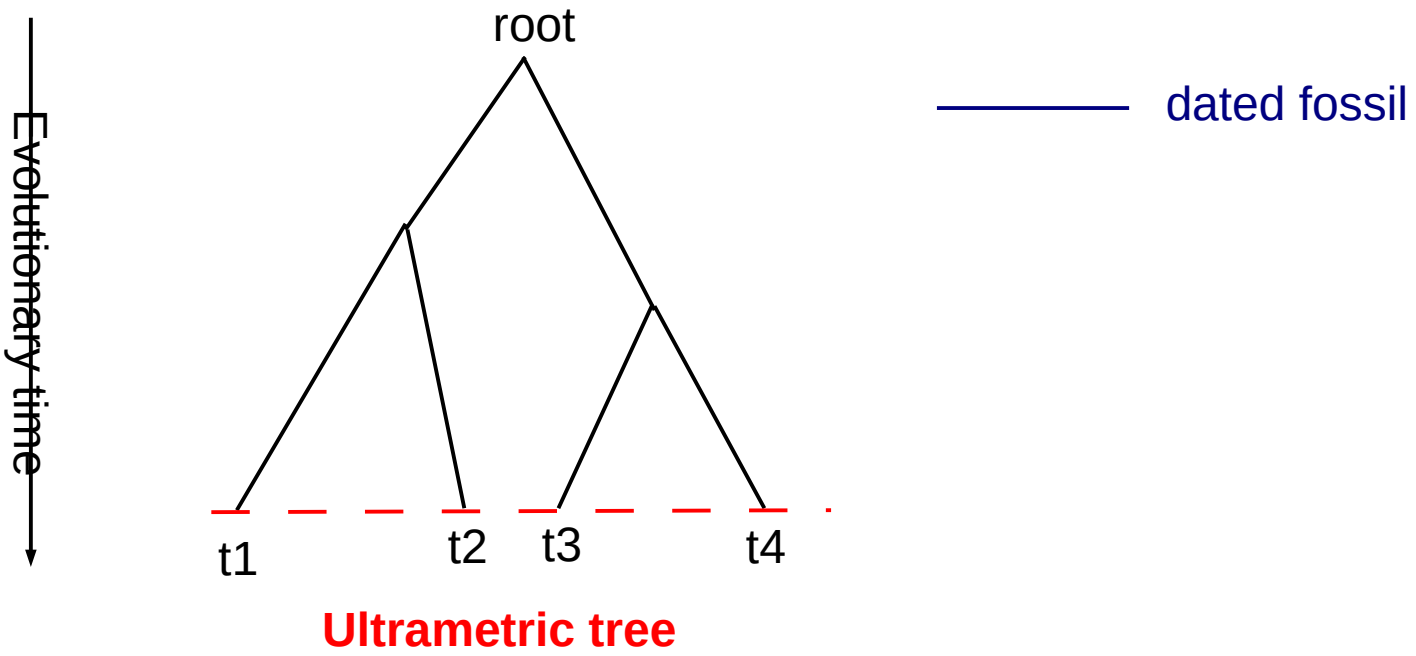


Ultrametric tree

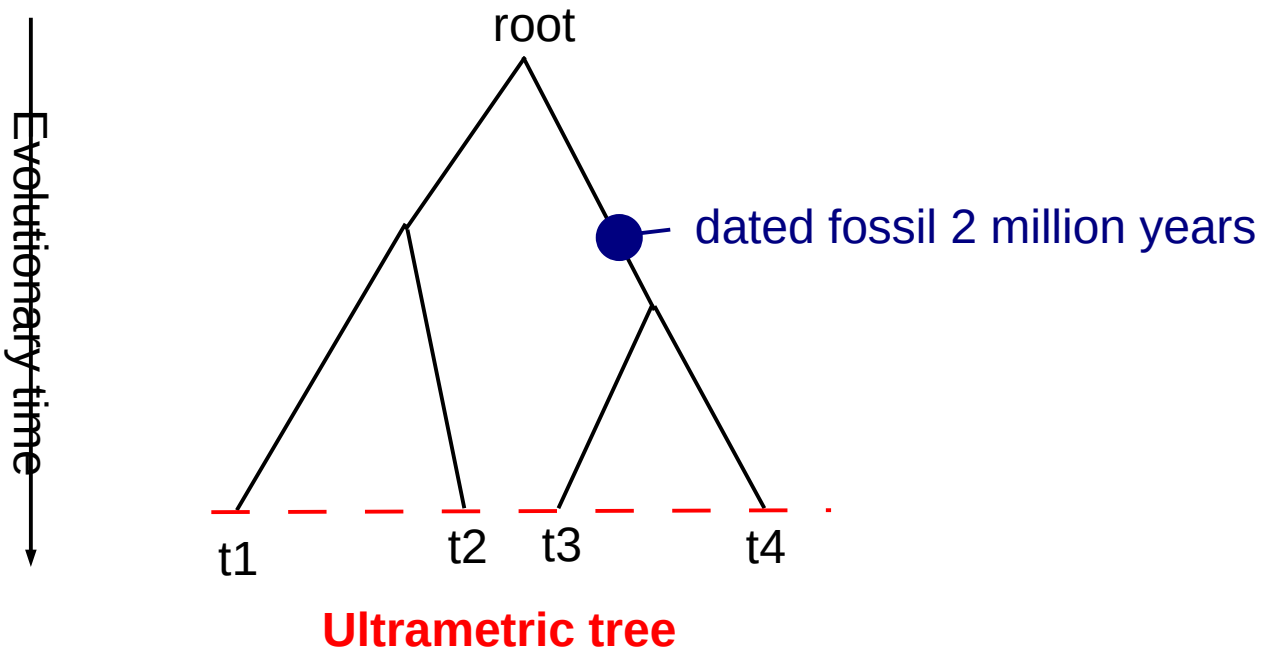


Non-ultrametric tree

Dating Trees

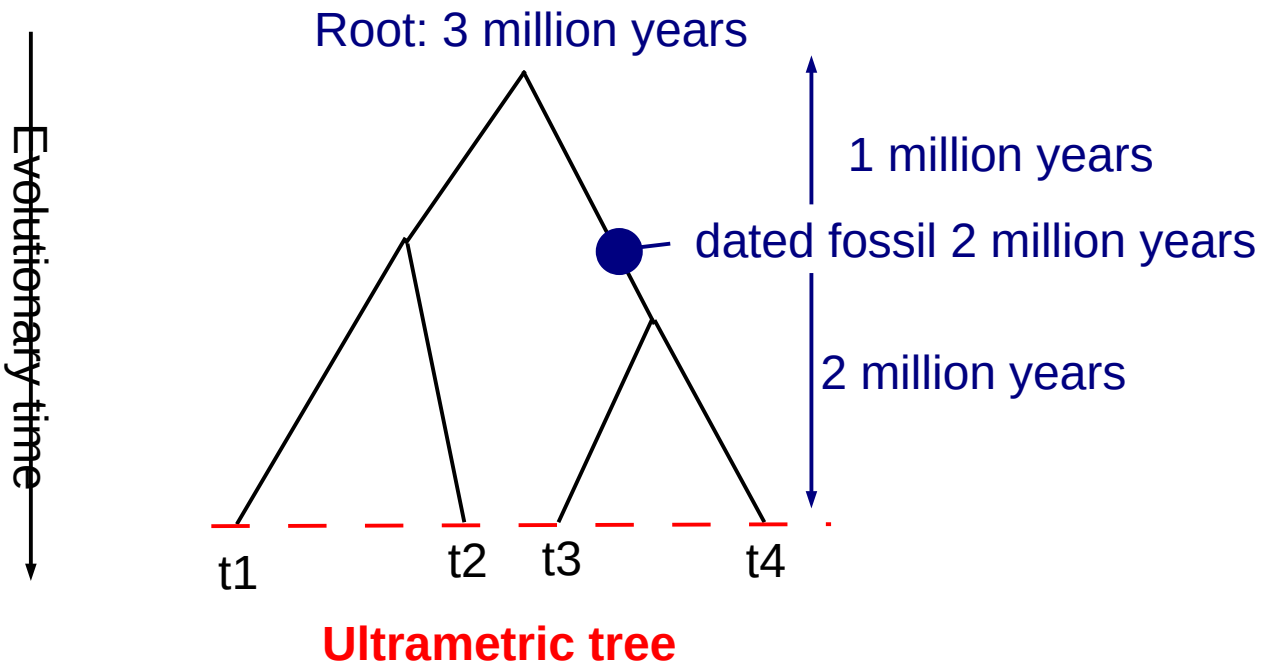


Dating Trees



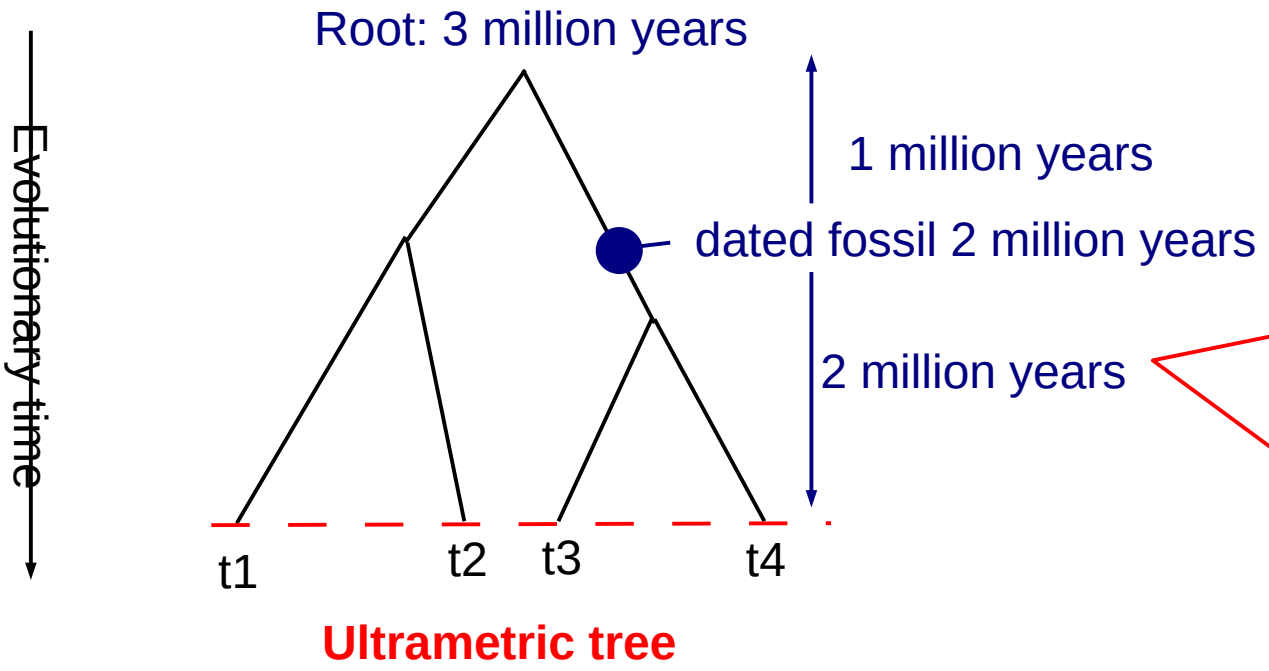
Ultrametric tree

Dating Trees



Ultrametric tree

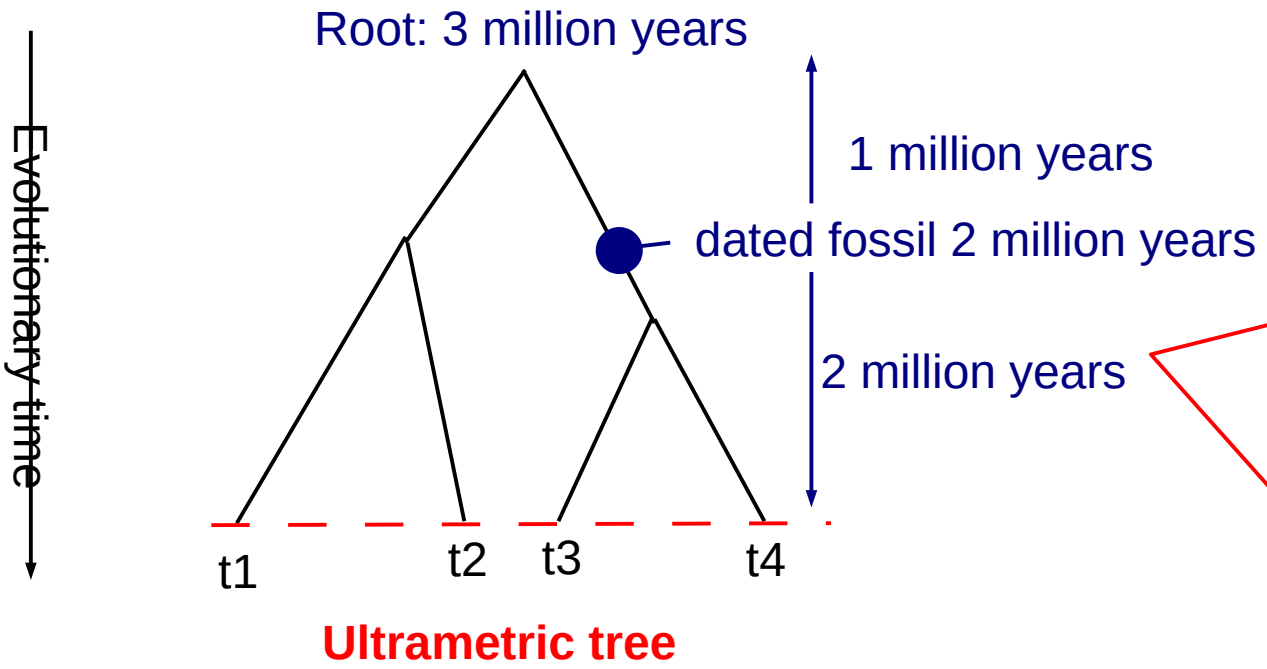
Dating Trees



Ultrametric tree

We need a rooted & ultrametric tree!
→ rooting with outgroups
→ ultrametricity with programs
for *divergence time estimation*
→ active research area
→ most codes rely on the phylogenetic likelihood function and Bayesian Statistics (MCMC methods)

Dating Trees



Ultrametric tree

But how do we place the fossil?
→ typically no DNA data available

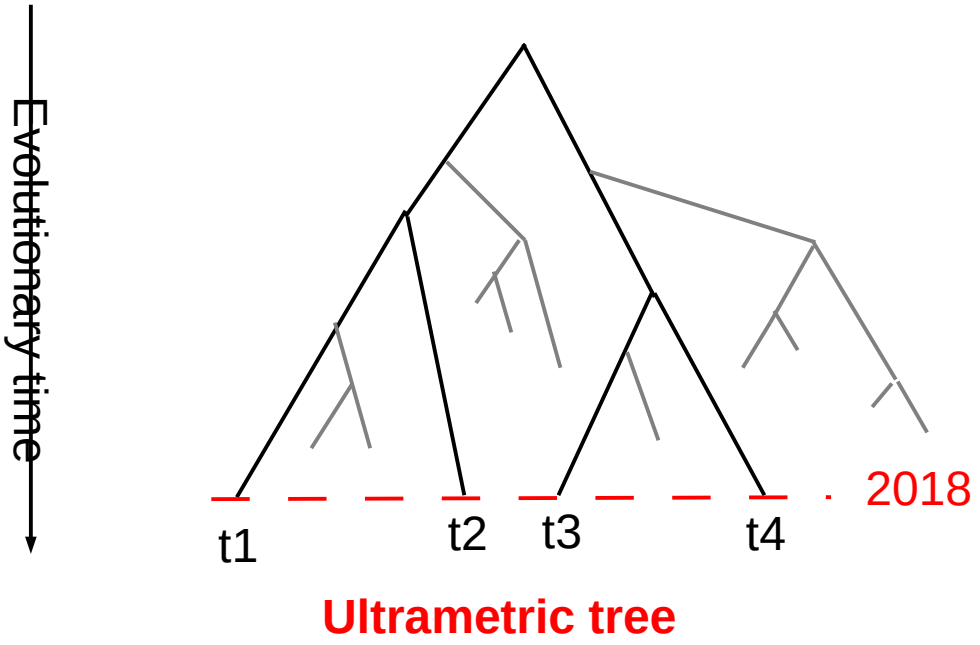
Fossil placement:
→ ad hoc using empirical knowledge
→ computationally using morphological data

The input for a phylogenetic analysis need not be molecular data!

We can also use sequences of morphological traits ("Merkmale")!

e.g. for trees of natural languages

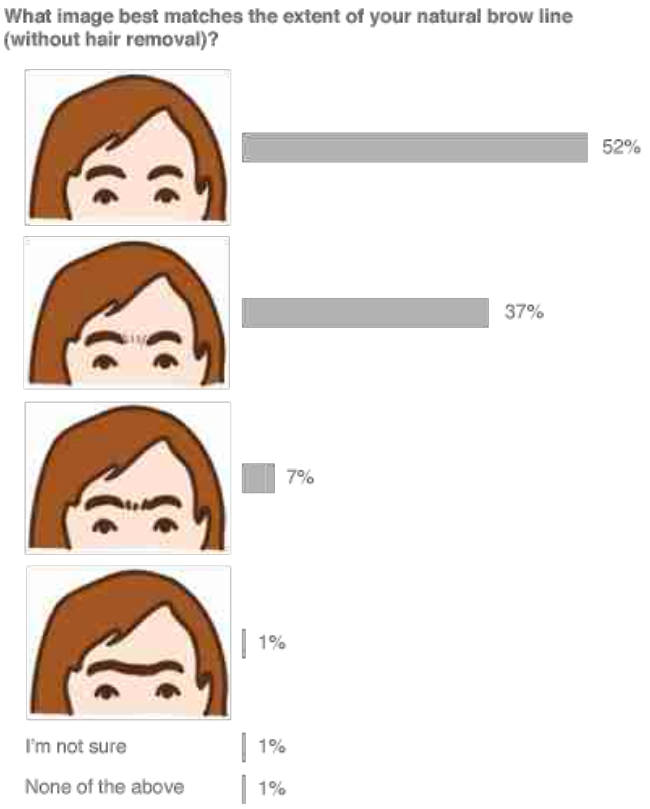
Remember that we deal with extant species!



Ultrametric tree

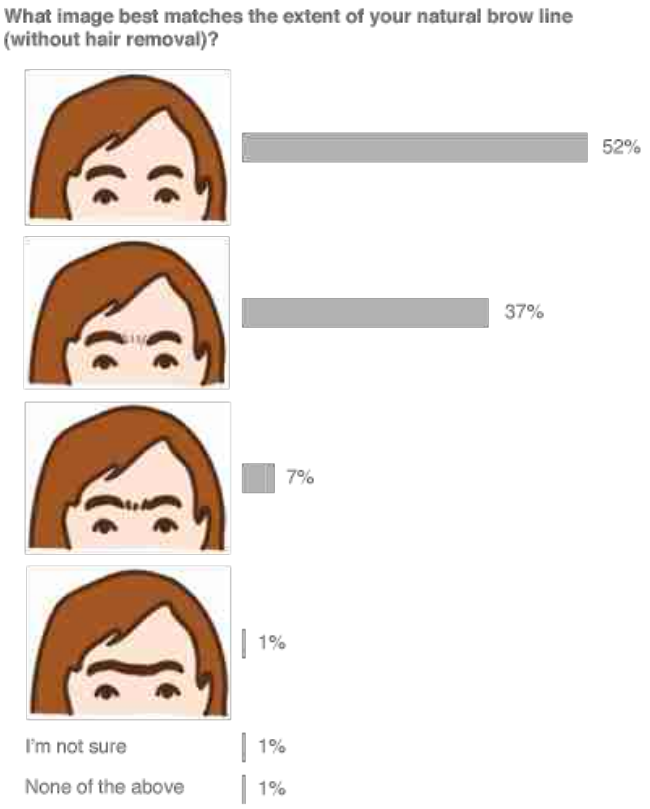
Morphological Traits

```
t1: 1000
t2: 0100
t3: 0010
T4: 0001

or:

t1: 0
t2: 1
t3: 2
t4: 3
```



Morphological Traits

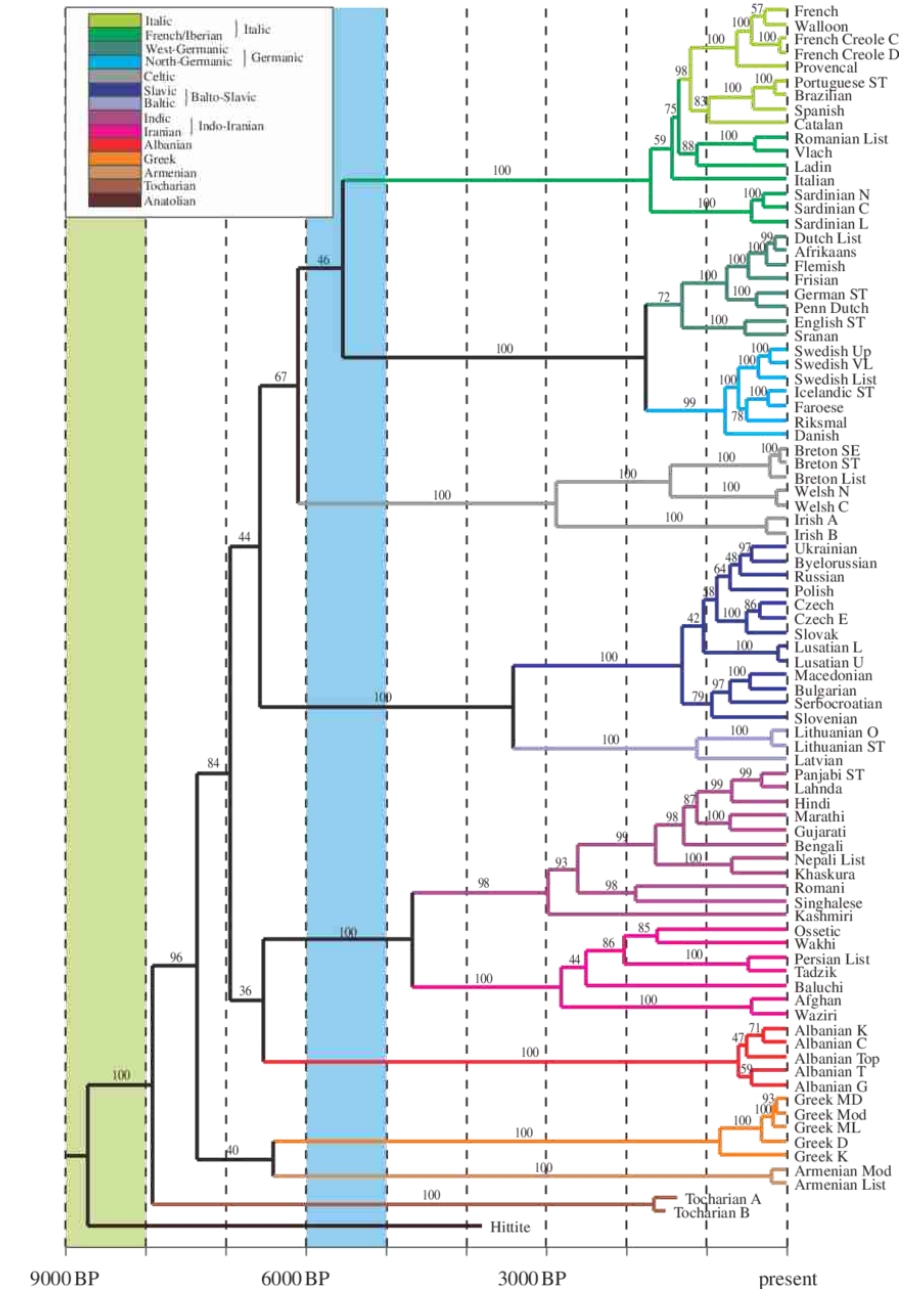
```
t1: 1000
t2: 0100
t3: 0010
T4: 0001

or:

t1: 0
t2: 1
t3: 2
t4: 3
```



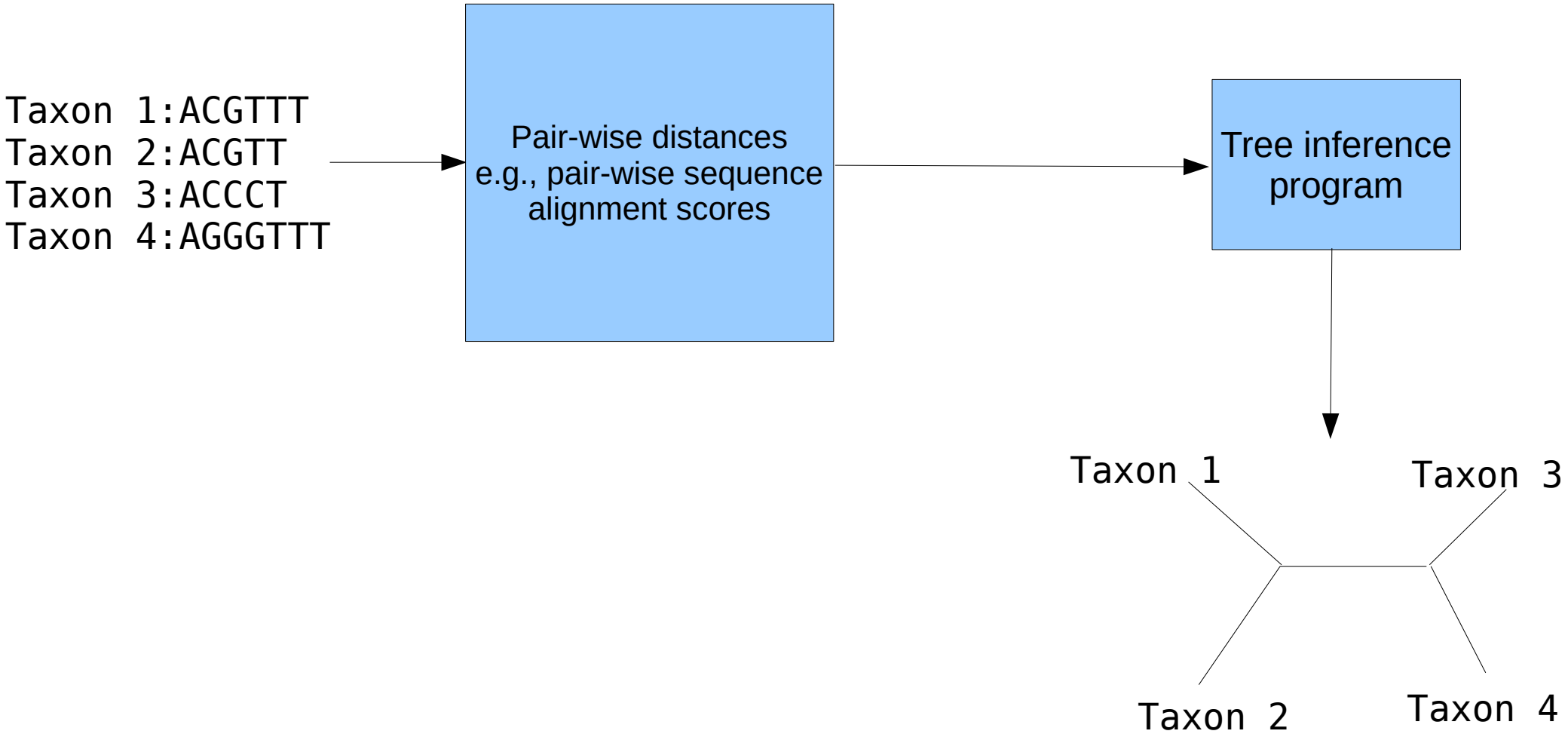
Traits need not be discrete,
they can also be continuous, e.g., bone ratios

Language Evolution

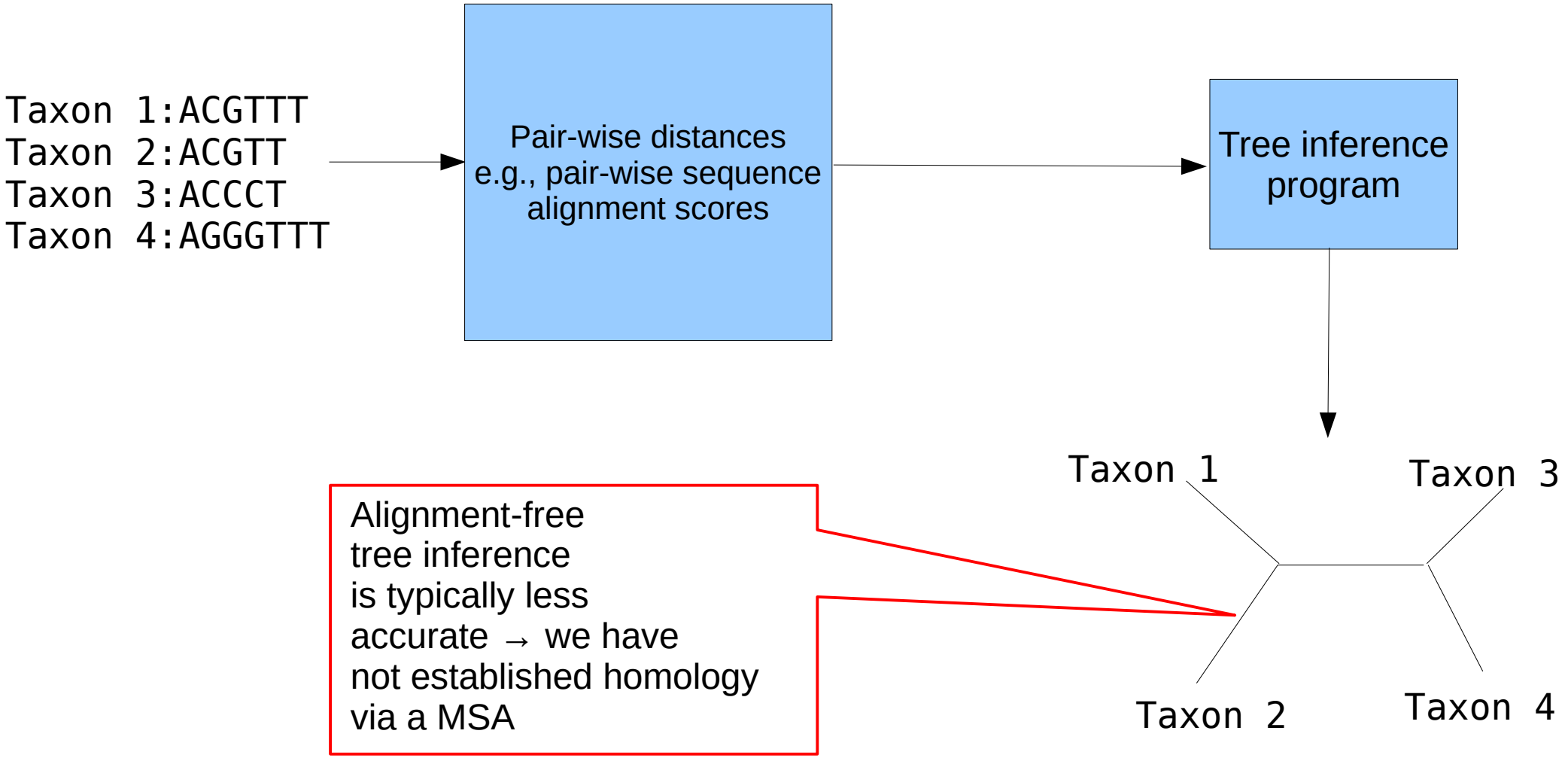
- Phylogenetic methods can also be used to infer trees of natural languages
- Input types
  - Lexical data
  - Morphosyntactic data
  - Sound data



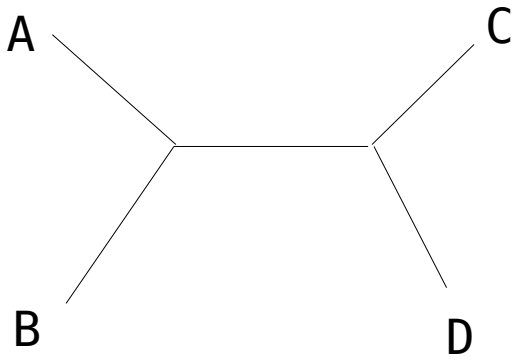
Alignment-Free Tree Inference

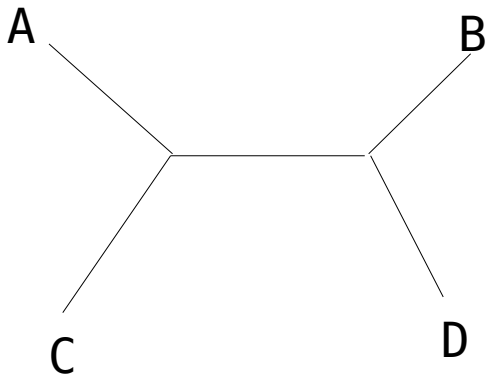


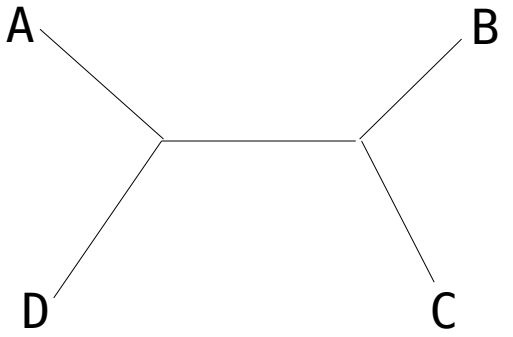
Alignment-Free Tree Inference



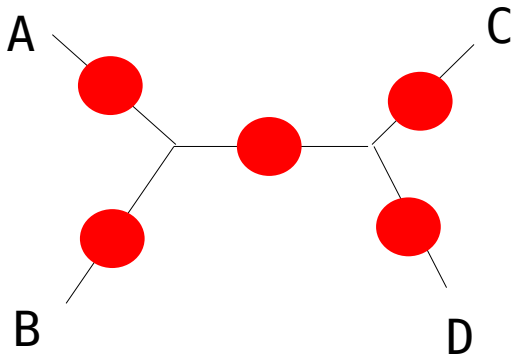
How many unrooted 4-taxon trees exist?

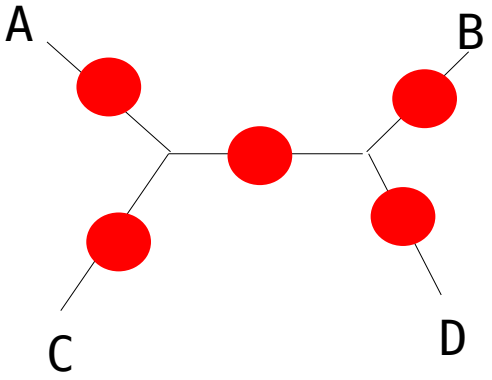


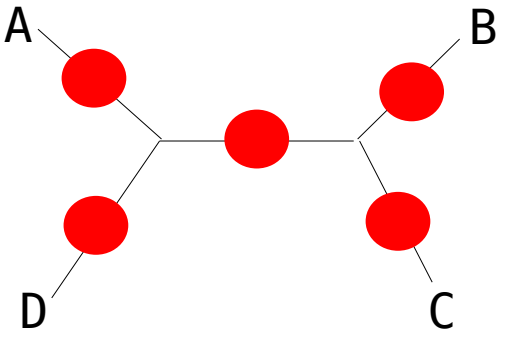




How many rooted 4-taxon trees exist?



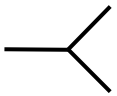




Tree Counts

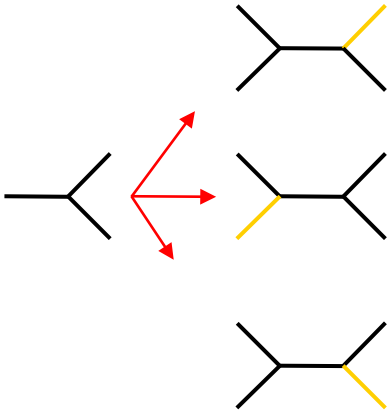
- Unrooted binary trees
  - *4* taxa → *3* distinct trees
  - A tree with *n* taxa has *n-2* inner nodes
  - And *2n-3* branches
- Rooted binary trees
  - *4* taxa → *3* unrooted trees * *5* branches each (rooting points) = *15* trees
  - *n-1* inner nodes
  - *2n-2* branches

The number of trees



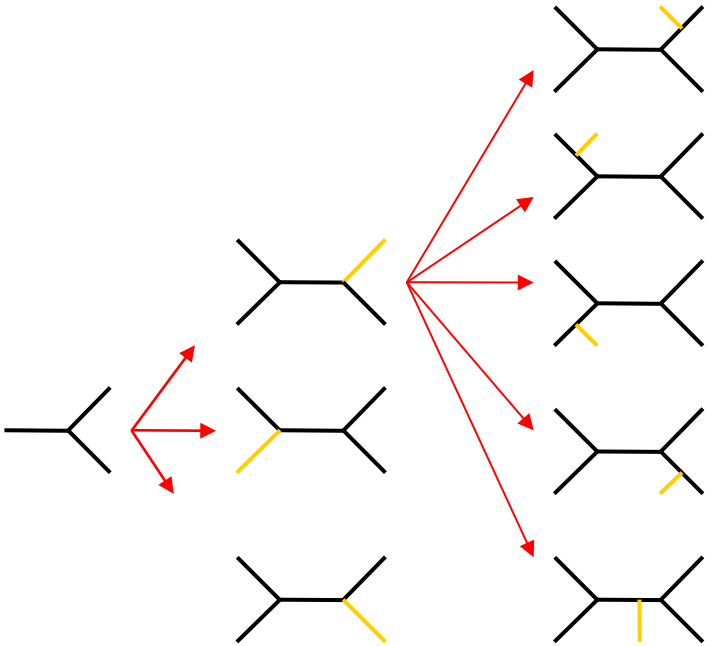
3 taxa = 1 tree

The number of trees



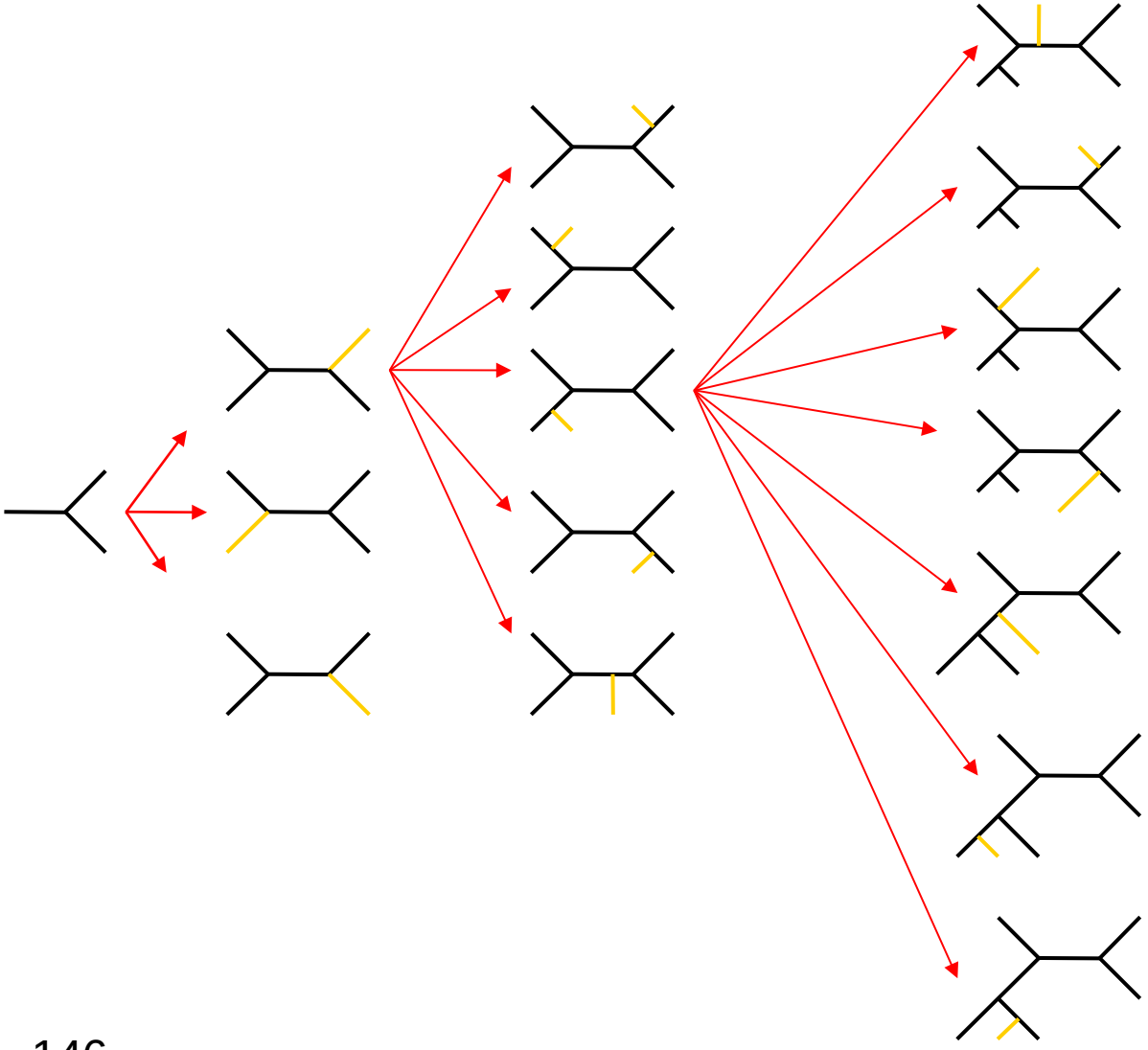
4 taxa: 3 trees
u: # trees of size 4-1 := 1
v: # branches in a tree of size 4-1 := 3
Number of unrooted binary trees with 4 taxa: u * v = 3

The number of trees



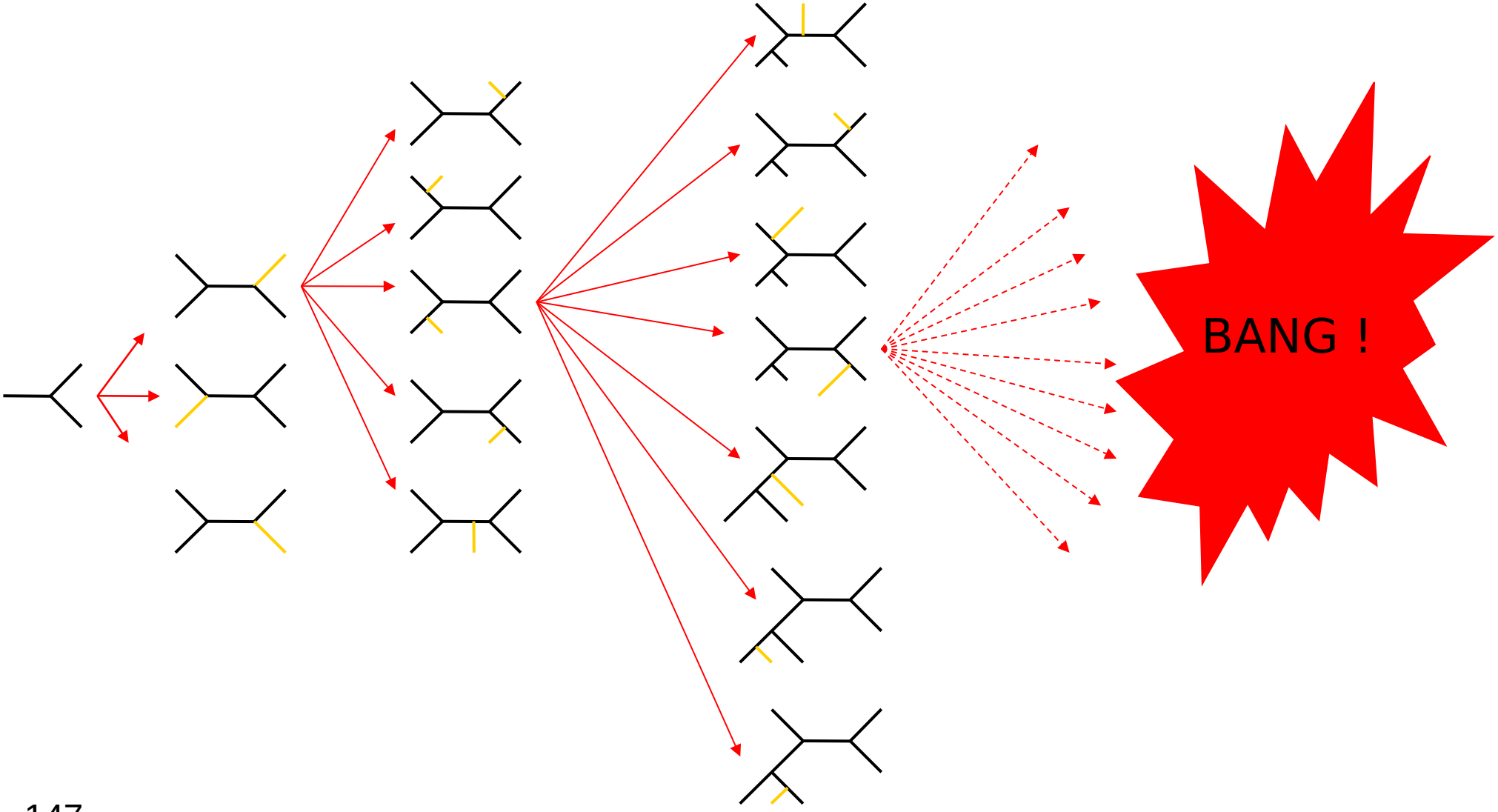
5 taxa: 15 trees
u = 3
v = 5
Number of unrooted trees with 5 taxa: 3 * 5 = 15

The number of trees



6 taxa: 105 trees
u = 15
v = 7
u * v = 105

The number of trees explodes!



Some Numbers

Number of Organisms	Number of alternative Trees
3	1
4	3
5	15
6	105
7	945
10	2.027.025
15	7.905.853.580.625
20	$2.21 * 10^{20}$
50	$2.84 * 10^{70}$

Table 2.1: Number of possible trees for phylogenies with 3–50 organisms

Equation for the number of unrooted trees

- Simple proof via induction

$$\prod_{i=3}^{n}(2i-5)$$

- The number of rooted trees for *n* taxa simply is the number of unrooted trees for *n+1* taxa
- The additional (*n+1*th) taxon represents all possible rootings for all unrooted trees with *n* taxa

trees with 2000 tips

```
stamatak@exelixis:~/Desktop/GIT/TreeCounter$ ./treeCounter -n 2000


GNU GPL tree number calculator released June 2011 by Alexandros Stamatakis

Number of unrooted binary trees for 2000 taxa: 3004963817421165615163291006568181498137723207423701308950495404301263652525830821082768599668824700046435273521426563428829589150234460006314939691306329704360561848618774654822779912235368092334555631999108345976931267565250128998674331877528114019609916315223670306091217357097623798477054676677795324797182614385273338226727784250737252849916669687584403510579587020686505817687044663181237429010214385064324713609344916670211359697569403006662526464792691245510314942366195542824118277625114848758254581227914289801132648902674033761294712745767036267579086843169660718609847941818865957214557044744572288661729053583520744253688123124010661315694886196094119564673620034257524133527757508582916109642257572769976799140828334321016132740165283099380390459232769069003597291970994073934956348620389901074268728229759746553771022576726768428580118772249501062181173405232082653973429622273525365905158656313832720311198419874675997386463182903203832523085979979922161012272157808052481458312068440167606239306009711616729715504728487799634337531348994230372437347879131989085953764070134849446113877572576952408702461720107874297380462275052545706689372319418206440706891884003870590289772197516454495975821662130620506461776109948566373416818358498932907699338206780105243728461492403422961155182609778228619192672071295189589360099591309742330723163825184281103305710174411568843051318658775443763085003114511107238370397074651822320404061547082730786299575493310312752086167006607912980142622300565123522718063819509335872651728623589020520016144361756075654286471422126613004434807084067501589247673166341539540575074474994909831496473031080411401891849735912811228378774049884834056210242056642446386009389965085742961947269054301528123752651096581528469979703679217112903556809818079169587951614159281049528179855847292534447864424435998085315372047968149694659917686145337010519859285771574824559433772423695825762426630169463204824951822559392874031776234338810486046309751915569238711675130952134150988167154643078623526062377864068386804246902527491139319276802611515990582603886733172930713673903403618637463980605764836474670274446727880885337074254421922726677747003329403201038288035112689026255183096791948358678929370163768175304820633894387149793115235369822962511116307148294599211620803302684762013335690441089668145436150905155877581167977001256391215111623744417049737170460402948110411482228646613191882199757138336835207252605520276982397461321849524926489705079039836025625560628985228883956135787417565766488999260873286612630642543260248979229113560071640573984516375245243376943755857384725545564397599604255914640112221144755235573176239973057747183956531217416532295986675901294116123924072209325036967312488449155375921065065601541672077415923624086866767534828651296488873905970757880247339346347084815901163977279774748041731626870091672873561216422684681606831989598012603764856153127816116895872151231233087600634733810972531184233396403909373783950668355787353078863586464005632994949063118742402902779272693300322445377595797224873456891511458557078385054168166766742581130195806362190750079029503108820909727174813643698947397107993277770067630173061756653873972603777717300844134394051236690555449324861650825399577950363267049478442934988531727973481777971465671751511788763964340693324580763461107342143281950499096808740273976889147045174720555436948966874260147724189469321290245331733418828677319465354411330210086657508171324034264758048921862366346160795594372051639515409694980648624230979694721116966596158004178839922322646284989423522769263911243607675089976717896834593378907423463545571935306656153799002779162656363618619740485909382340622354597697331372136593437175855906644396461328300113672601934068706442339489199215304385281541659630118549423634863524585746642834609062916279564926584723003608555598989761916293248140094592489899468468862322586017055146890564983727600392748706955046378881974169942904910285318041077651600726338472163890378227001384359959973026572719880432466431235975855521719696051392101022659632478878309774053331315517629781528807186526031763272648280944993704562580999380530584976995700802893798080149029010052938984722799471678048216894241591182842576964647865057315325178302336307298251692210346584265894447464916123854689718507968172909139032182834111184821384767728316548653212317382004131990510518967022201887049585687180509590730360693040293721603896891760558767695538231809370582625708389838740909846865663427139750001329183510594332172987982524370750827208795985943715766766015578269966034319775262330889962587800628009560944416932377944955441033696586261556256010669390303203878970983673786087056641433585106111658314520424513208508589994932364831689671194951671619567622707090697388958885579562466641536561723549301807394004760529801721771391686788000277851966173070061284517307582503735643102065112443730825229625040453160590741343881872563477913830660590931880252231008534017684026140153961698919207514710803375770884974014183459975397205987868206487911606496985817760115397205849822698907181349432691801821173318806365391089368981171489135745668054280748517017585826663963357018935444983266976283509265792220174637219027311964175148994401007963687601782674710701994547321888783274260889667243715747134206000937042513098936305374597842799804031329894172664922904257309583685344162156405572902820662240038632375263809102332698978388604237596256015679752626950798639868104294832333160267216555178120899264677804935741326387137408423885546533836158643451305439624281397279559725995110706314305992615495622958320232708057681156690489586610522030057372529847211874782713671366605866927109487556397485848947591081972703387828443986448674345620095816193031472734596190049931842433797524366248936332124485059719925236685292493053462527641378534132089431289015237380925560459870909127666623296787033288820591349495800740744731433888007245323217473096597419671144414531271327902051010047671014350638857953478447255389801541923317027519896180635152682543173193832925891931530164130548972311128666465492971930479296432829556719092881692091042334122007454242049900872585046208051104875883059495990311188736668509414882172573457635523396403848131821316740835900691640005326225818478376506780445117771732865818989921535830944776535034179687 5

Approximately 3.00 times 10^6328
```

A side-note
The `treeCounter` tool

- Evidently, the tree count can not be computed using normal integers

  → we need an arbitrary precision library

  → I used the GNU GMP (Multiple Precision Arithmetic) library

  → `treeCounter` available as open-source code at

  https://github.com/stamatak

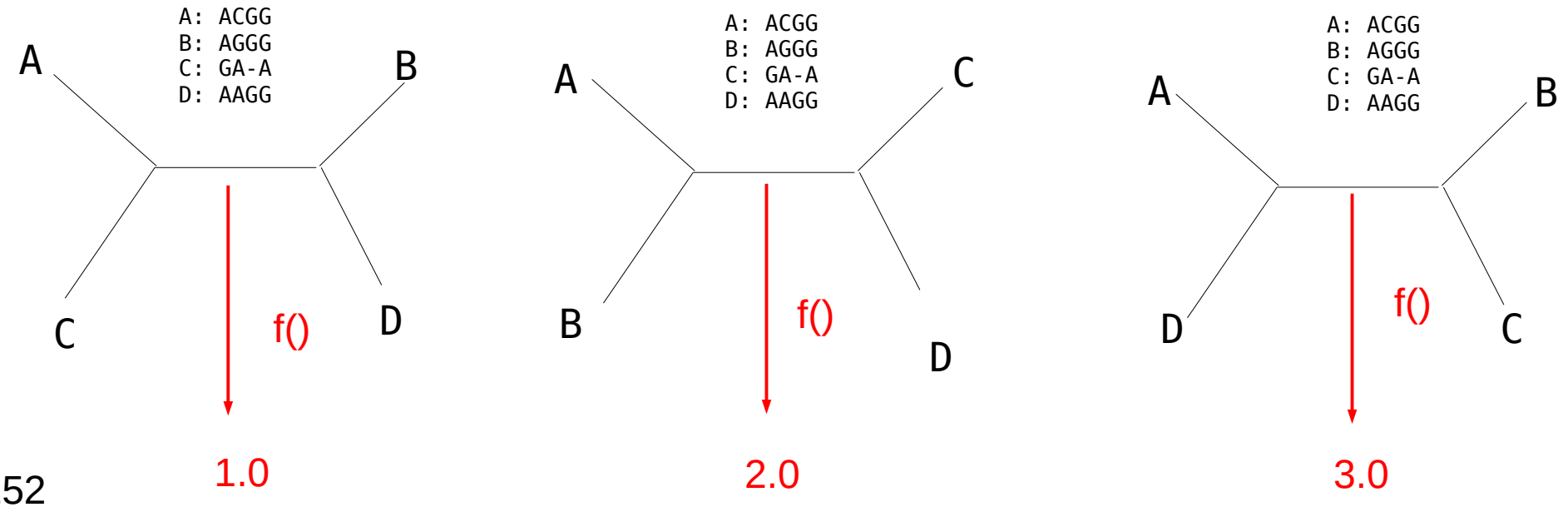
  → Has anybody already used GNU GMP?

Scoring Trees

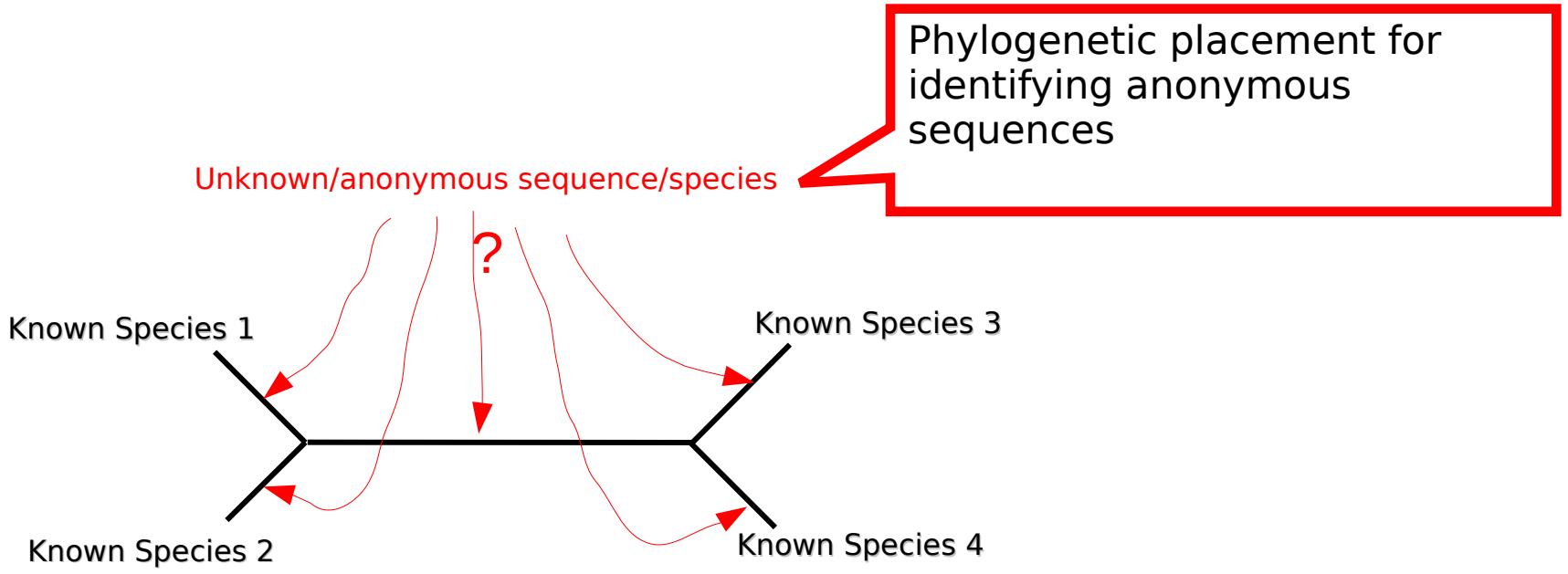
- Now we know how many **unrooted** candidate trees there exist for *n* taxa
- How do we chose among them?

  → we need some scoring criterion f() to evaluate them

  → finding the optimal tree under most criteria is NP-Hard

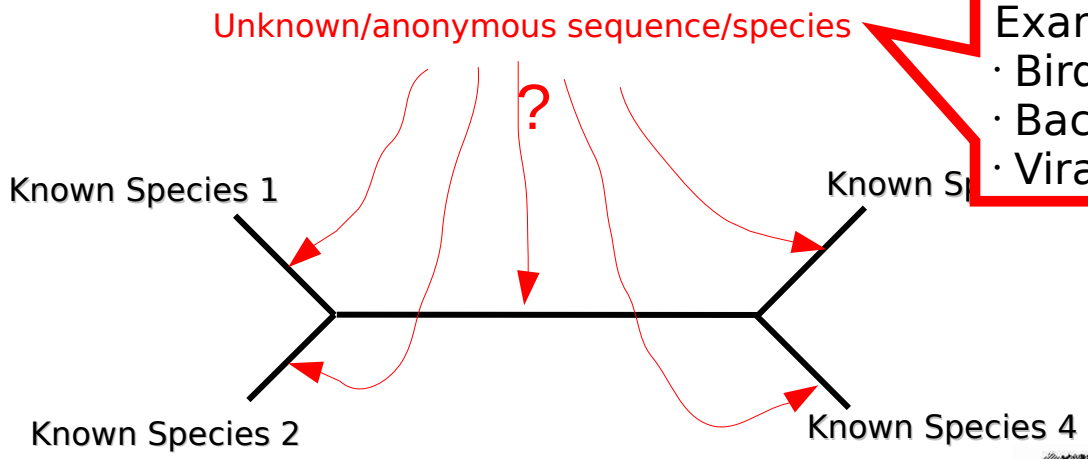


What can we do with Phylogenies?



reference phylogeny

What can we do with Phylogenies?



Phylogenetic placement for identifying anonymous sequences

Examples:

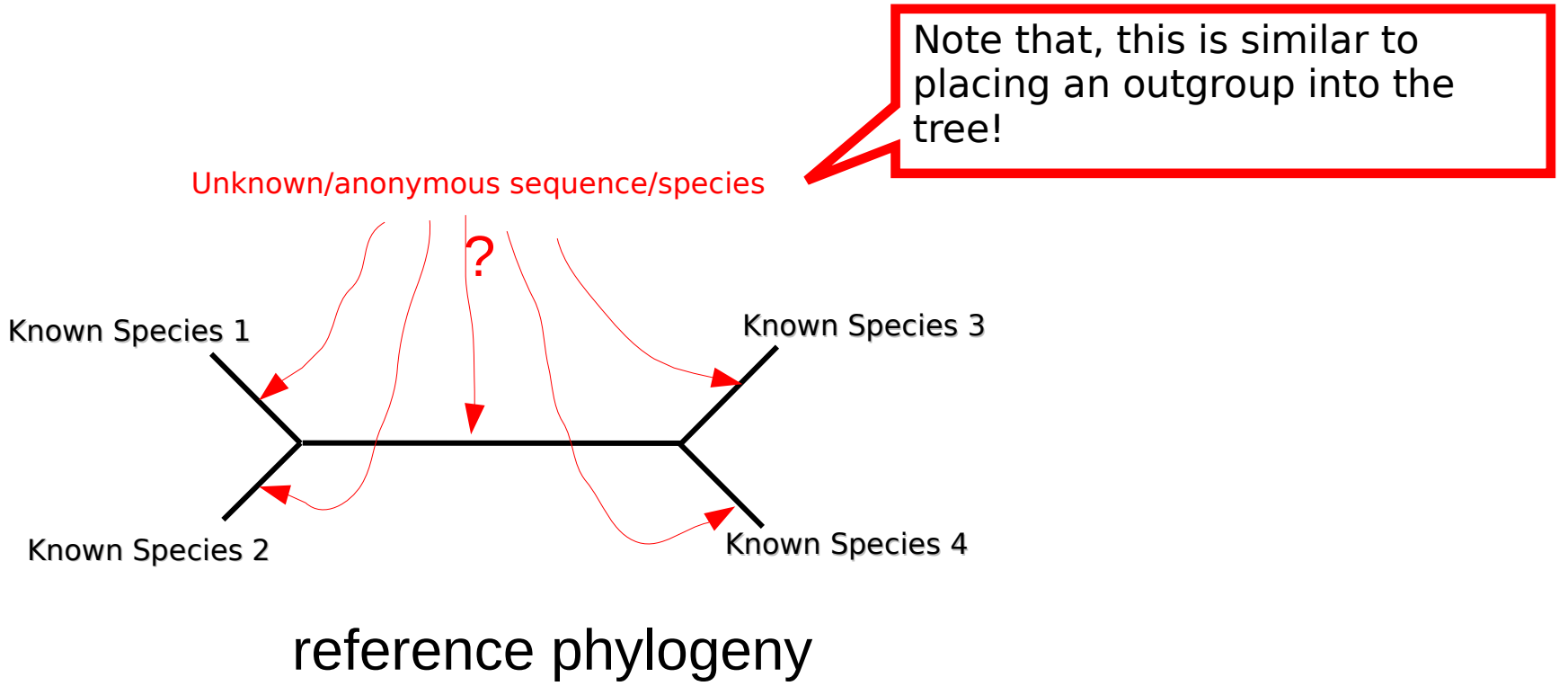
· Bird strike

· Bacteria

· Viral strains

reference phylogeny

What can we do with Phylogenies?



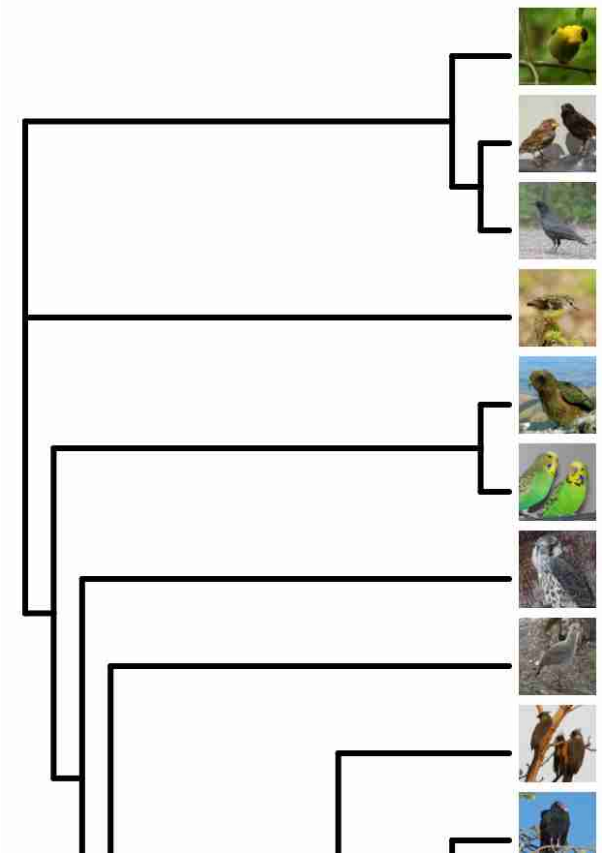
reference phylogeny

The Birdstrike Web-Game

- https://cme.h-its.org/exelixis/eseb/public/en/core/title.html

Aerial Collisions

A research team found out, how birds around the world are related to each other.



Do you see, which birds are closely realted to each other? That is fascinating, right? Click on the images to find out more about the bird on the image.

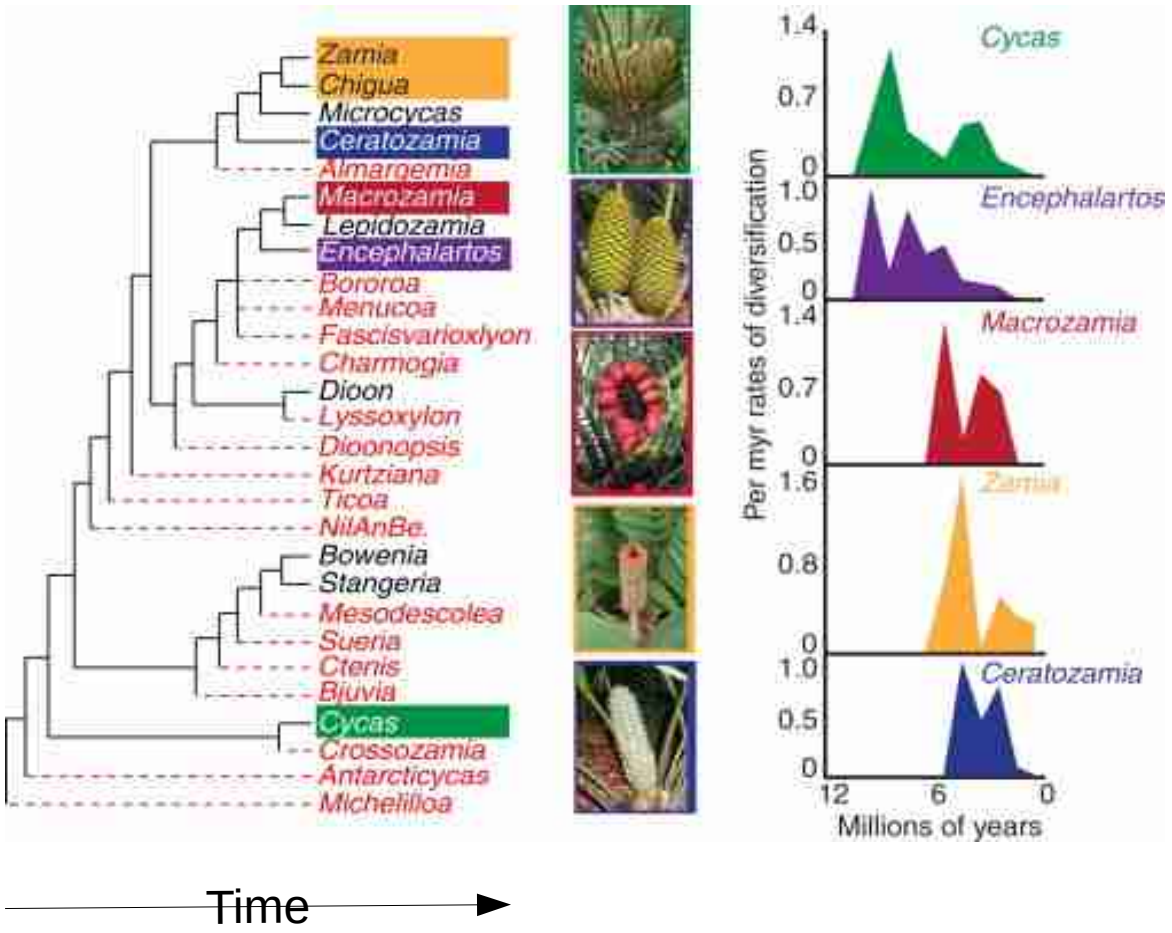
What is a phylogenetic tree?

The molecular laboratory received a bird sample from a plane that should be identified. Can you help them?

Become a real bird researcher

Where do the DNA samples come from? Watch this short movie!

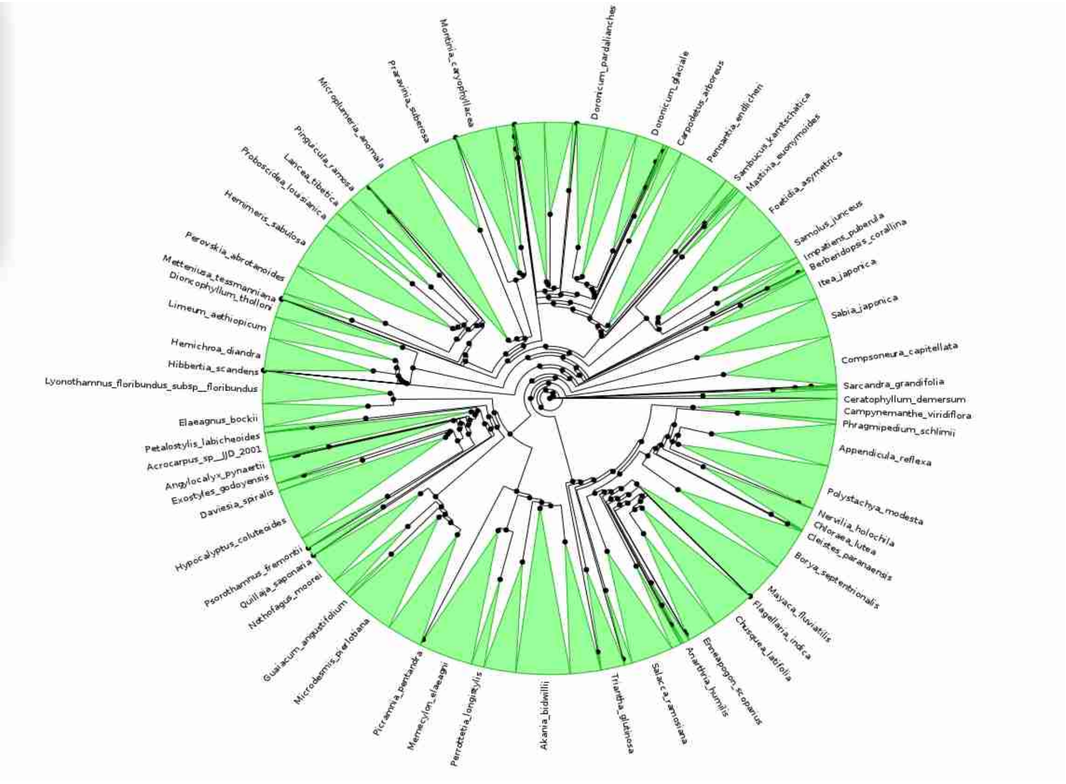
Diversification Rates



From: Charles C. Davis, Hanno Schaefer: “Plant Evolution: Pulses of Extinction and Speciation in Gymnosperm Diversity”, *Current Biology*, 2011.

Diversification Rates

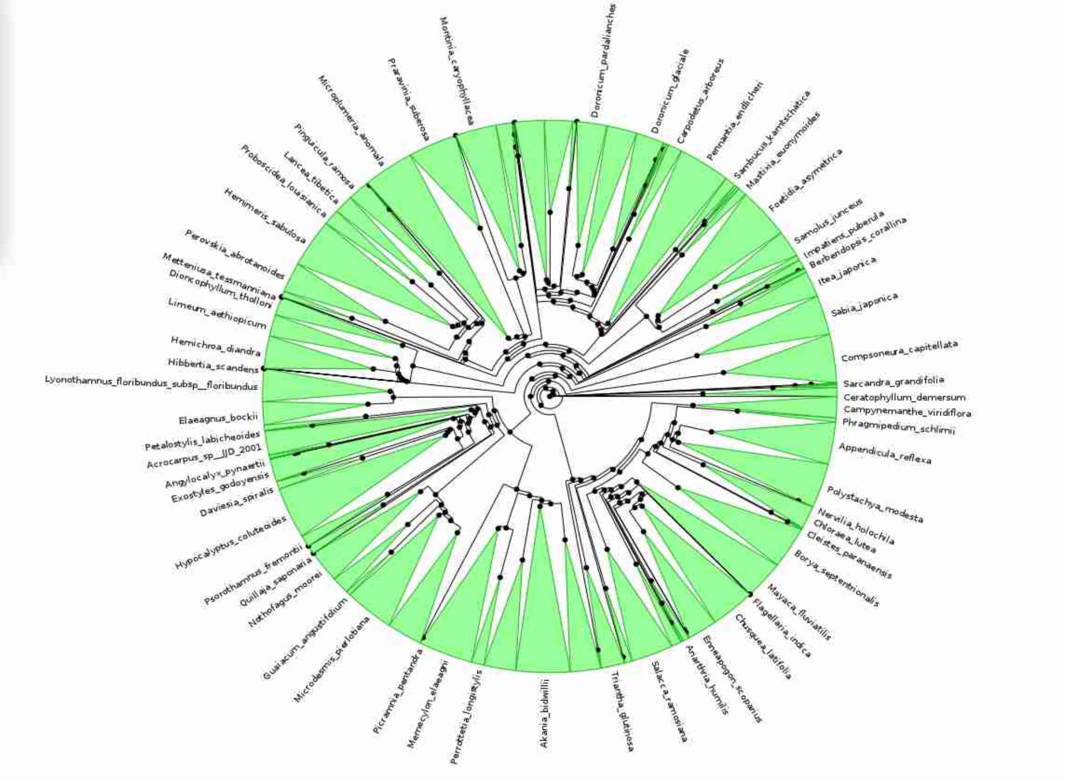
- With former PostDoc Stephen Smith: “Understanding angiosperm diversification using small and large phylogenetic trees”, *American Journal of Botany* 98 (3), 404-414, 2011.
- Largest tree of angiosperms computed to date
- 55,000 taxa



Diversification Rates

- With former PostDoc Stephen Smith: “Understanding angiosperm diversification using small and large phylogenetic trees”, *American Journal of Botany* 98 (3), 404-414, 2011.
- Largest tree of angiosperms computed to date
- 55,000 taxa

Visualizing big trees also represents a challenge → graph drawing & layout algorithms.

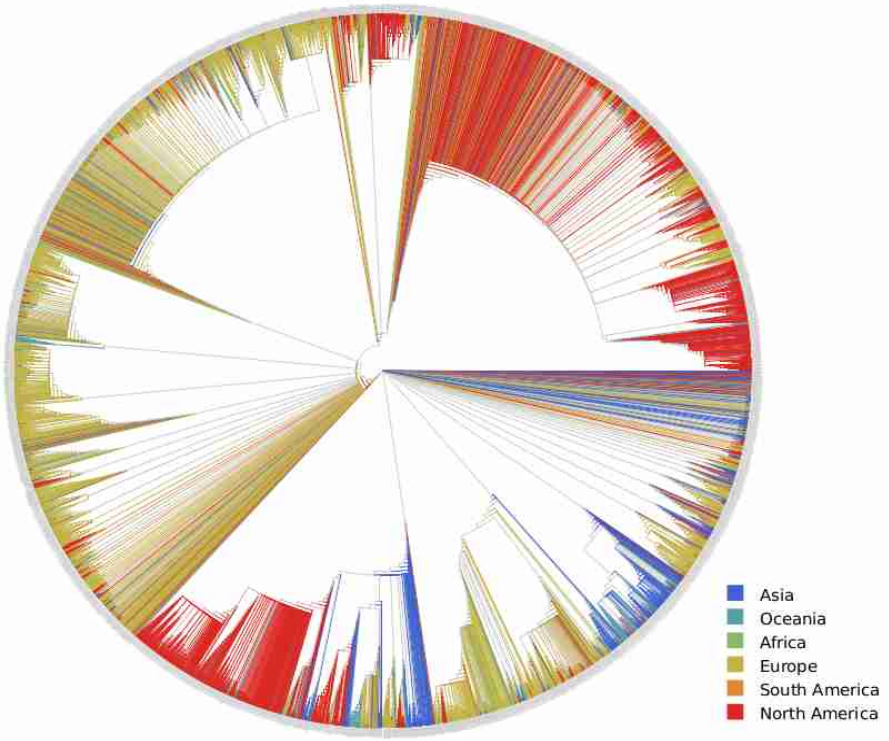


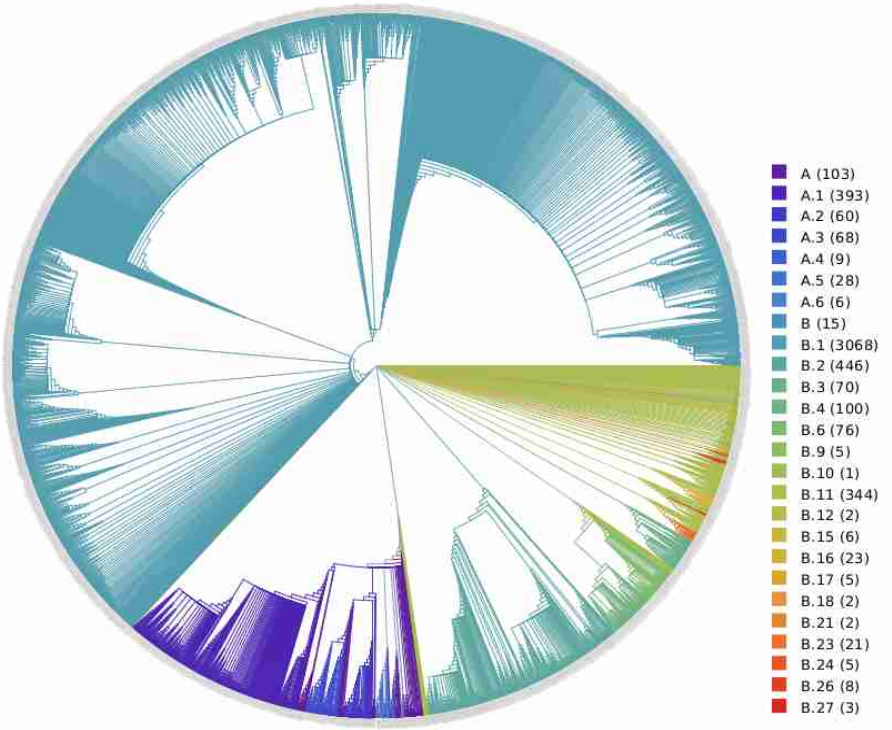
Influenza Outbreaks

And of course SARS-CoV-2

Phylogenetic analysis of SARS-CoV-2 data is difficult

Benoit Morel[*,1], Pierre Barbera[*,1], Lucas Czech[1], Ben Bettisworth[1], Lukas Hübner[1,2], Sarah Lutteropp[1], Dora Serdari[1], Evangelia-Georgia Kostaki[3], Ioannis Mamais[3], Alexey M Kozlov[1], Pavlos Pavlidis[4], Dimitrios Paraskevis[3], and Alexandros Stamatakis[1,2]





Snakebites

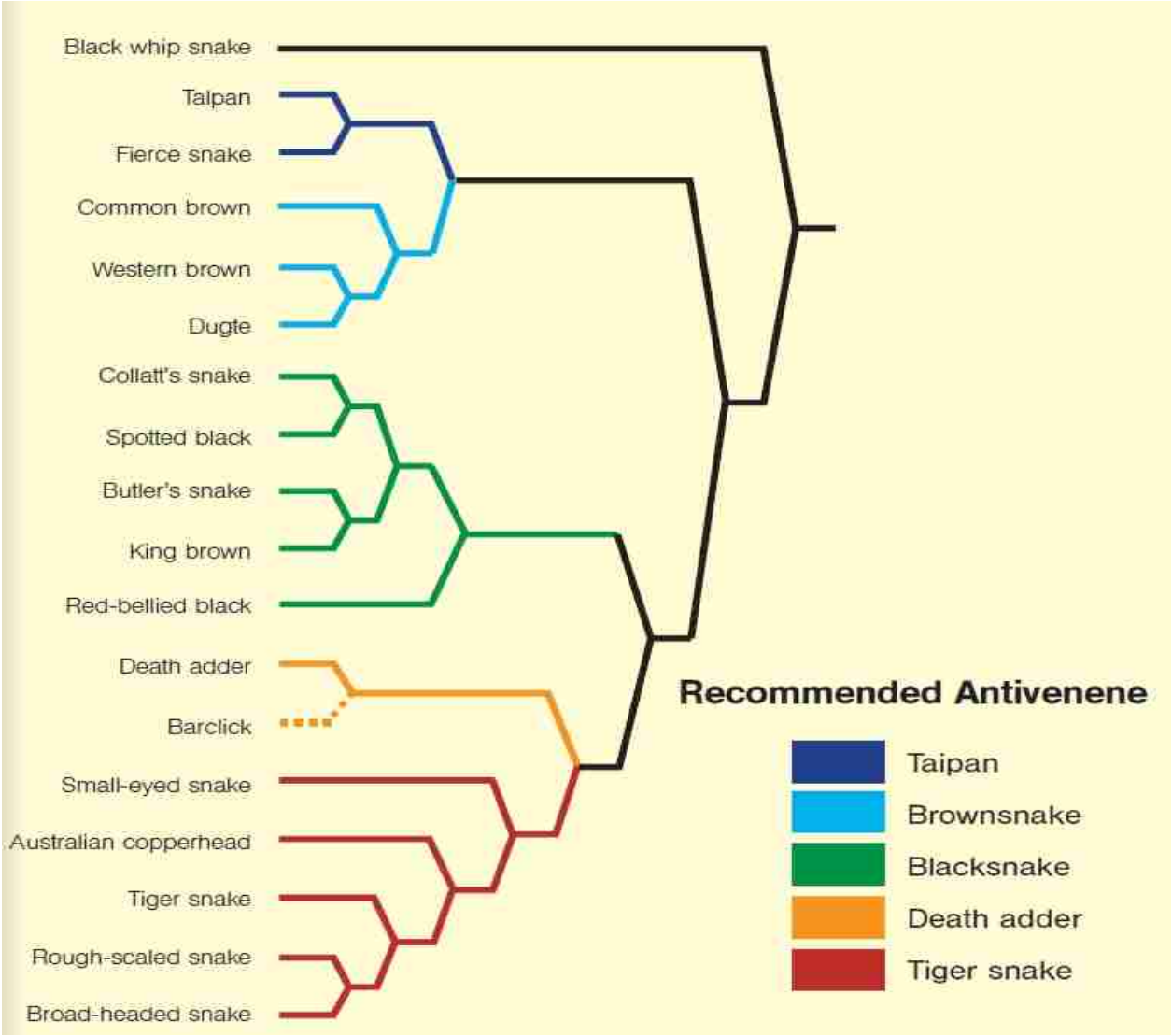
Australia has more poisonous snakes than any other continent, and many people die from snakebites each year. Developing effective antivenins is thus a high priority, but little is known about the venins of most species.

Phylogenetic analysis is helping with this task because venin properties correlate strongly with evolutionary relationships.

Although the red-bellied black snake looks very different from the king brown, it is actually closely related and can be treated with the same antivenin.

Conversely, the western brown looks very similar to the king brown, but it is only distantly related and thus responds best to different antivenin.

The phylogeny is also predictive: the recent demonstration that the poorly-known barclick is closely related to the death adder (orange lineage) predicts that the former is also highly dangerous and might respond to widely-available death adder antivenin.



Snakebites

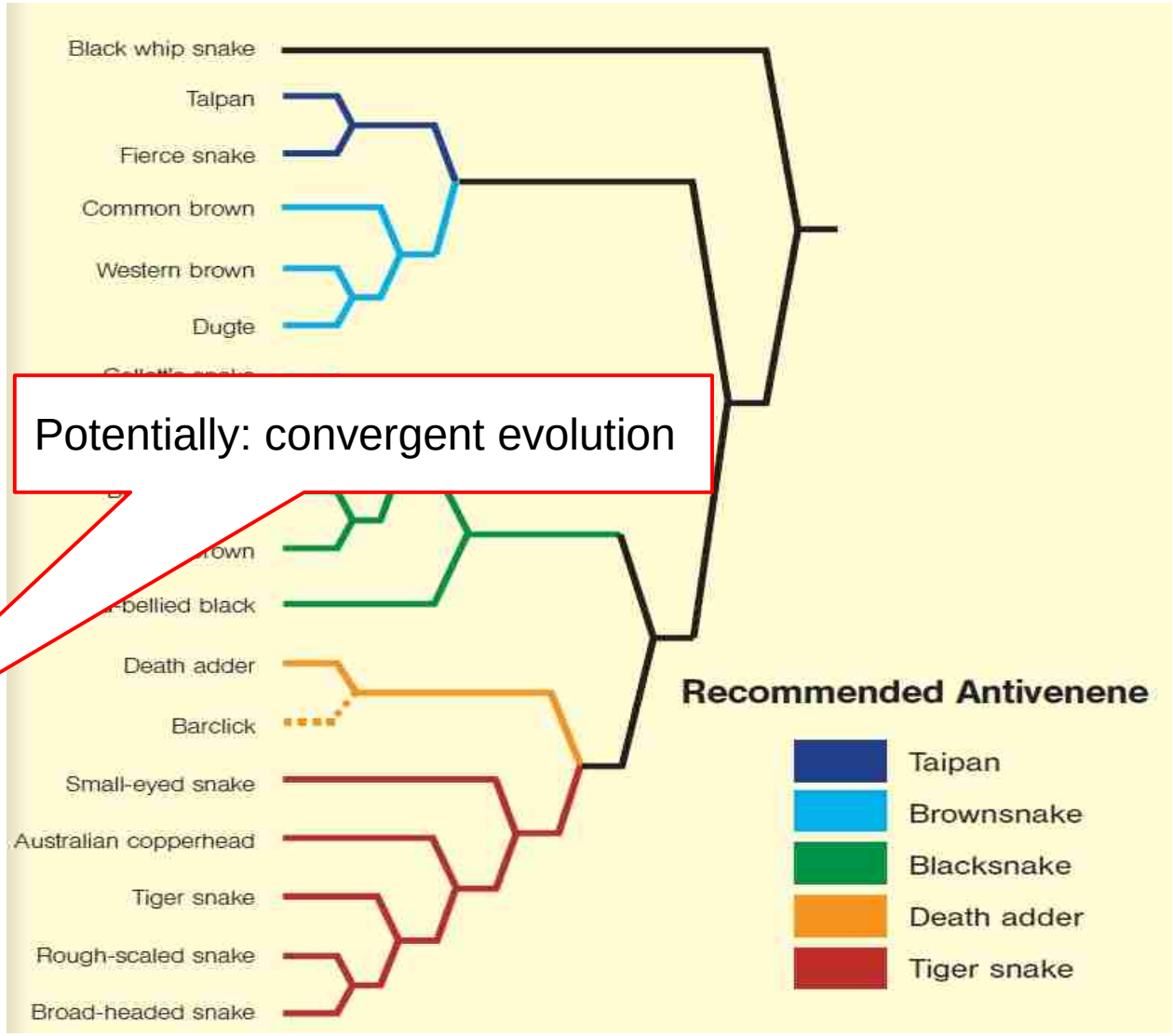
Australia has more poisonous snakes than any other continent, and many people die from snakebites each year. Developing effective antivenins is thus a high priority, but little is known about the venins of most species.

Phylogenetic analysis is helping with this task because venin properties correlate strongly with evolutionary relationships.

Although the red-bellied black snake looks very different from the king brown, it is actually closely related and can be treated with the same antivenin.

Conversely, the western brown looks very similar to the king brown, but it is only distantly related and thus responds best to different antivenin.

The phylogeny is also predictive: the recent demonstration that the poorly-known barclick is closely related to the death adder (orange lineage) predicts that the former is also highly dangerous and might respond to widely-available death adder antivenin.

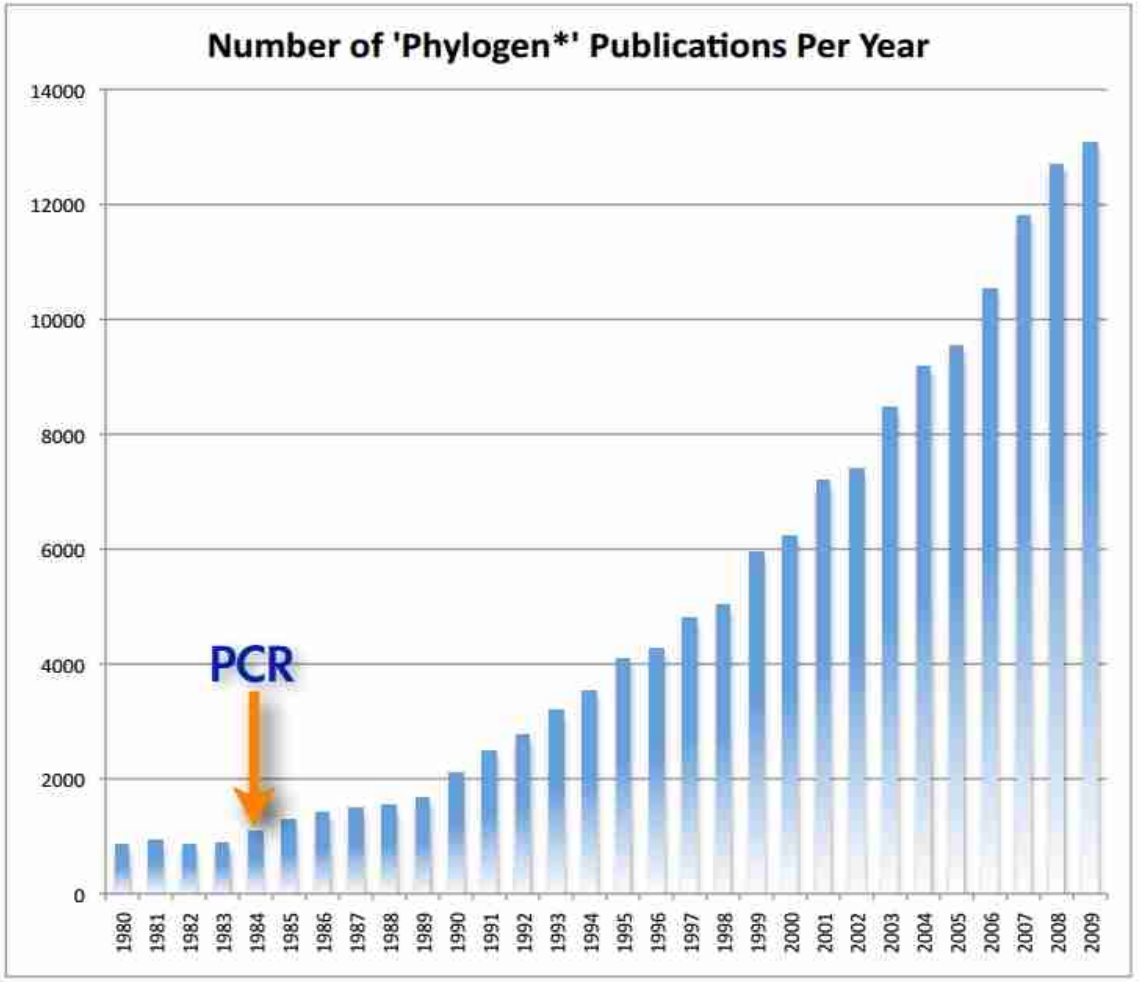


What can we do with phylogenetic trees?

- identifying unknown species
- divergence time estimates
- diversification rates
- viral outbreaks
- forensics → M.L. Metzker, D.P. Mindell, X.M. Liu, R.G. Ptak, R.A. Gibbs, D.M. Hillis: “Molecular evidence of HIV-1 transmission in a criminal case” PNAS: 99(22):14292-7, 2002.

"Nothing in Biology makes sense, except in the light of evolution"

Why this increase in Phylogenetics papers?
Advances in:
- Sequencing technology
- Hardware
- Methods & Tools



Building Trees

- We distinguish between
  - *Distance-based methods*
    - → use MSA to compute a matrix of pair-wise distances
    - → build a tree using these distances
    - → Heuristics (essentially hierarchical clustering methods)
      - → *Neighbor Joining:* NJ
      - → *Unweighted Pair Group Method with Arithmetic Mean:* UPGMA
    - → least-squares method: explicit optimality criterion
  - *Character-based methods*
    - → optimality criteria *f()* operate directly on the MSA & tree
      - → parsimony
      - → maximum likelihood
      - → Bayesian inference
    - → take the current tree topology & MSA to calculate a score
    - → the score tells us how well the MSA data fits the tree

Building Trees

- We distinguish between
  - *Distance-based methods*
    - → use MSA to compute a matrix of pair-wise distances
    - → build a tree using these distances
    - → Heuristics (essentially hierarchical clustering methods)
      - → *Neighbor Joining:* NJ
      - → *Unweighted Pair Group Method with Arithmetic Mean:* UPGMA
    - → least-squares method: explicit optimality criterion
  - *Character-based methods*
    - → optimality criteria *f()* operate directly on the MSA & tree
      - → parsimony
      - → maximum likelihood
      - → Bayesian inference
    - → take the current tree topology & MSA to calculate a score
    - → the score tells us how well the MSA data fits the tree

Less accurate, but faster

Slow, but more accurate

Building Trees

- We distinguish between
  - *Distance-based methods*
    - → use MSA to compute a matrix of pair-wise distances
    - → build a tree using these distances
    - → Heuristics (essentially hierarchical clustering methods)
      - → *Neighbor Joining:* NJ
      - → *Unweighted Pair Group Method with Arithmetic Mean:* UPGMA
    - → least-squares method: explicit optimality criterion
  - *Character-based methods*
    - → optimality criteria *f()* operate directly on the MSA
      - → parsimony
      - → maximum likelihood
      - → Bayesian inference
    - → take the current tree topology & MSA to calculate a score
    - → the score tells us how well the MSA data fits the tree

Memory-intensive!

Less accurate, but faster

Slow, but more accurate

Building Trees

- We distinguish between
  - *Distance-based methods*
    - → use MSA to compute a matrix of pair-wise distances
    - → build a tree using these distances
    - → Heuristics (essentially hierarchical clustering methods)
      - → *Neighbor Joining:* NJ
      - → *Unweighted Pair Group Method with Arithmetic mean:* UPGMA
    - → least-squares method: explicit optimality c
  - *Character-based methods*
    - → optimality criteria *f()* operate directly
      - → parsimony
      - → maximum likelihood
      - → Bayesian inference
    - → take the current tree topology & MSA to calculate a score
    - → the score tells us how well the MSA data fits the tree

Less accurate, but faster

What could be the computational limitation here?

Memory-intensive!

Slow, but more accurate

Building Trees

- We distinguish between
  - *Distance-based methods*
    - → use MSA to compute a matrix of pair-wise distances
    - → build a tree using these distances
    - → Heuristics (essentially hierarchical clus ing methods)
      - → *Neighbor Joining:* NJ
      - → *Unweighted Pair Group Method with Arithmetic M* GMA
    - → least-squares method: explicit optimality criteri
  - *Character-based methods*
    - → optimality criteria *f()* operate directly on t
      - → parsimony
      - → maximum likelihood
      - → Bayesian inference
    - → take the current tree topology & MSA to calculate a score
    - → the score tells us how well the MSA data fits the tree

Less accurate, but faster

Storing this matrix can become problematic memory-wise
→ out-of-core/external memory algorithms
→ e.g.: NINJA tool for Neighbor joining
"Large-scale neighbor-joining with ninja"
T Wheeler,
Algorithms in Bioinformatics, 2009

Out-of-core Algorithms

- Definition from Wikipedia:

  Out-of-core or External memory algorithms are algorithms that are designed to process data that is too large to fit into a computer's main memory at one time. Such algorithms must be optimized to efficiently fetch and access data stored in slow bulk memory such as hard drive or tape drives.

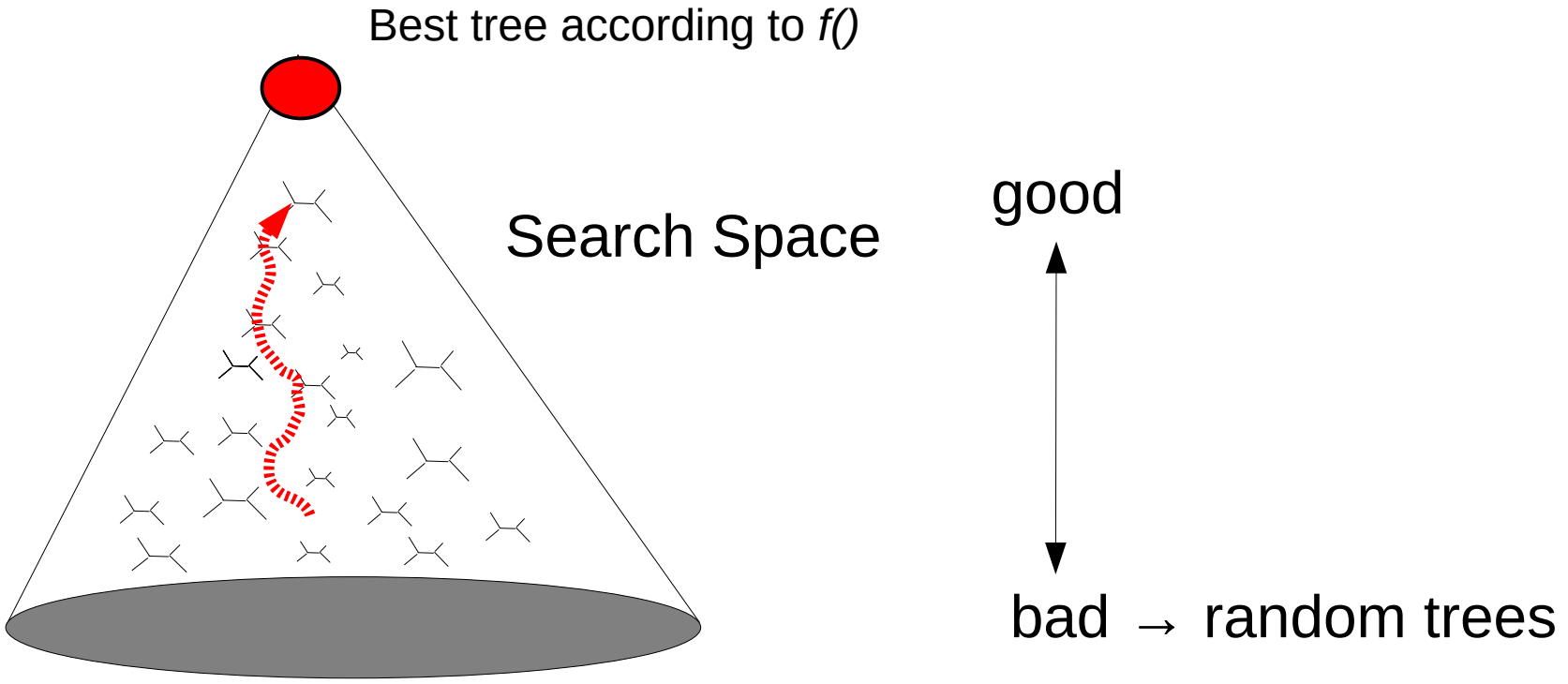
- We do the data transfer RAM ↔ disk explicitly from within the application code by using application-specific knowledge (e.g., about the data access patterns)
- This is to circumvent the paging procedure that would normally be initiated by the OS
- Out-of-core algorithms are typically much faster than the *application-agnostic* paging procedure carried out by the OS
- For an example from phylogenetics see:

  Fernando Izquierdo-Carrasco, Alexandros Stamatakis: "Computing the Phylogenetic Likelihood Function Out-of-Core", *IEEE HICOMB 2011 workshop*, Anchorage, USA, May 2011.

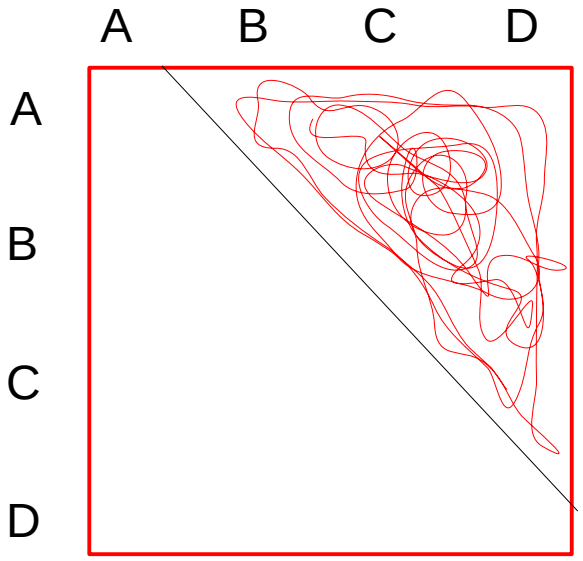
NP-Hardness

- Because of the super-exponential increase in the number of possible trees for *n* taxa ...
- all interesting criteria on trees are NP-hard:
  - Least squares
  - Parsimony → discrete criterion
  - Likelihood → statistical criterion
  - Bayesian → integrate likelihood over entire tree space

Search Space

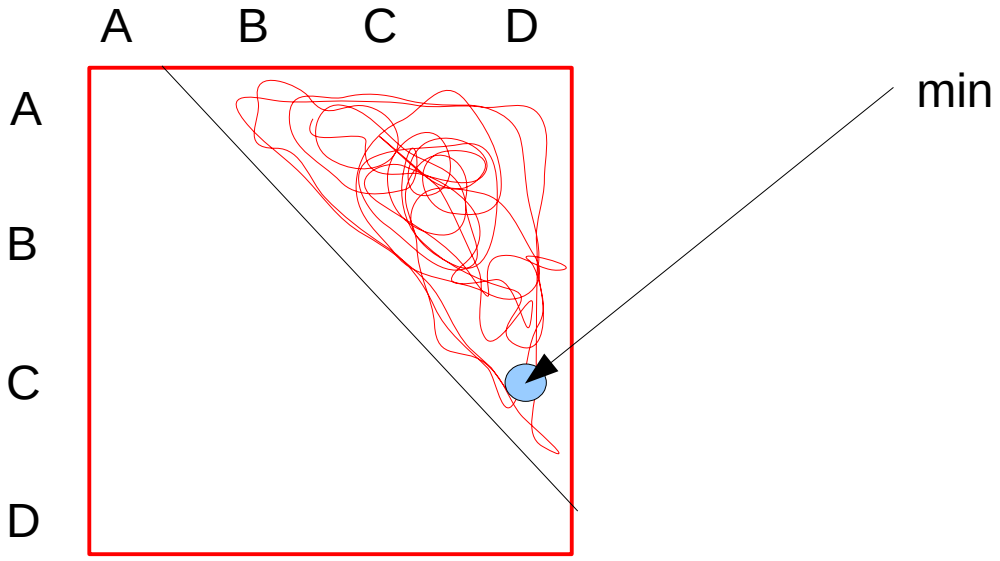


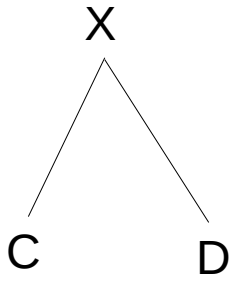
Neighbor Joining → Principle



Given a kind of distance matrix $D_{i,j}$ where $i,j=1...4$

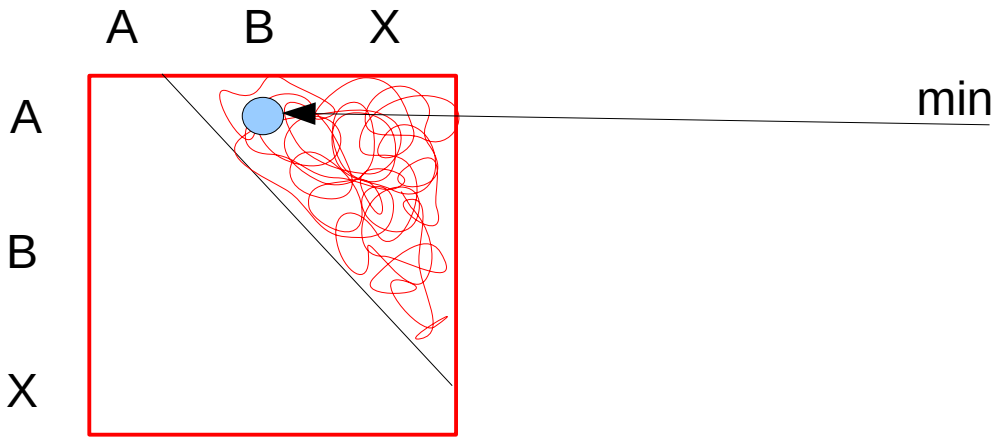
Neighbor Joining → Principle

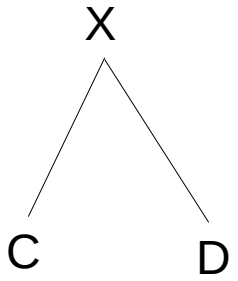




Given a kind of distance matrix $D_{i,j}$ where $i,j=1...4$
Find minimum and merge taxa

Neighbor Joining → Principle





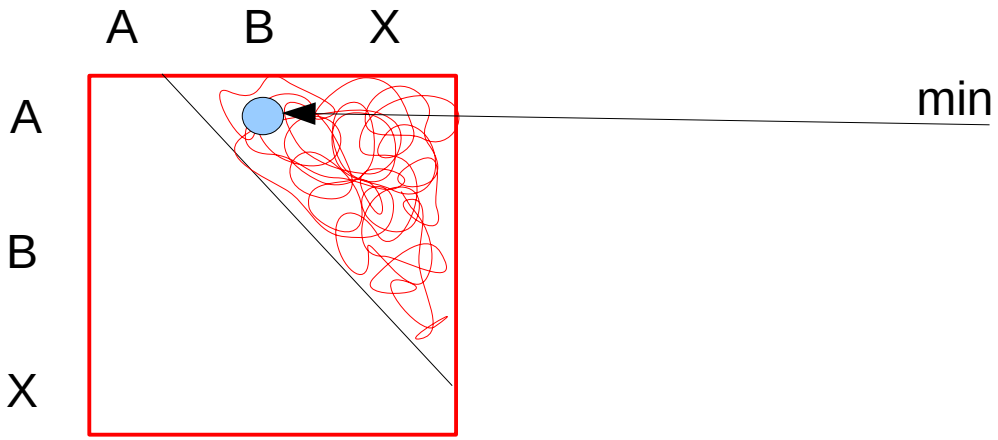
Given a kind of distance matrix $D_{i,j}$ where $i,j=1...4$

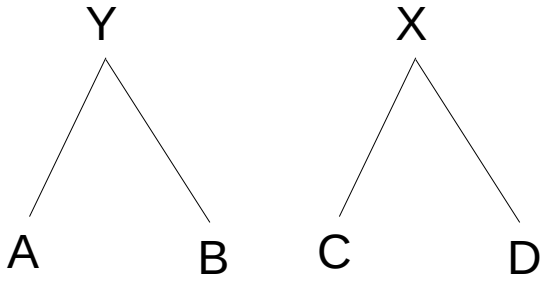
Find minimum and merge taxa

Compute a new distance matrix of size $n-1 = 3$

Find minimum

Neighbor Joining → Principle





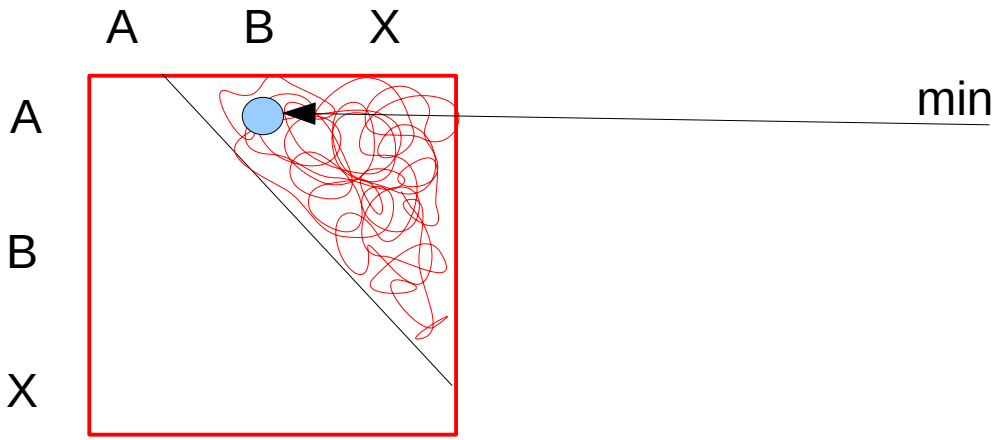
Given a kind of distance matrix $D_{i,j}$ where $i,j=1...4$

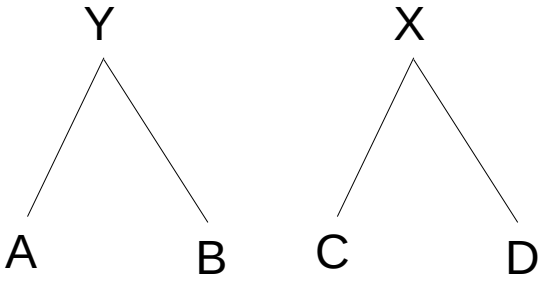
Find minimum and merge taxa

Compute a new distance matrix of size $n-1 = 3$

Find minimum and merge taxa

Neighbor Joining → Principle





Given a kind of distance matrix $D_{i,j}$ where $i,j=1...4$

Find minimum and merge taxa

Compute a new distance matrix of size $n-1 = 3$

Find minimum and merge taxa

Etc.

Space complexity: $O(n^2)$

Time complexity: $O(n^3)$

Key question: how do we compute distance between *X* and *A* or *X* and *B* respectively

→ for progressive alignment we may align the profile of *X* with all remaining sequences

Neighbor Joining Algorithm

- For each tip compute

  $u_i = \Sigma_j D_{ij} / (n-2)$

  → this is in principle the average distance to all other tips

  → the denominator is *n-2* instead of *n,* see below why

- Find the pair of tips, *(i, j)* for which $D_{ij}-u_i-u_j$ is minimal
- Connect the tips *(i,j)* to build a new ancestral node *X*
- The branch lengths from the ancestral node *X* to *i* and *j* are:

  $b_i = 0.5\ D_{ij} + 0.5\ (u_i-u_j)$

  $b_j = 0.5\ D_{ij} + 0.5\ (u_j-u_i)$

- Update the distance matrix:
  → Compute distance between the new node *X* and each remaining tip as follows:

  $D_{ij,k} = (D_{ik}+D_{jk}-D_{ij})/2$

- Replace tips *i* and *j* by the new node *X* which is now treated as a tip
- Repeat until only two nodes remain
  → connect the remaining two nodes with each other

Neighbor Joining Algorithm

	A	B	C	D
A	-	17	21	27
B		-	12	18
C			-	14
D				-

Neighbor Joining Algorithm

	A	B	C	D
A	-	17	21	27
B		-	12	18
C			-	14
D				-

Distance matrix, usually denoted as *D*

i	u_i
A	(17+21+27)/2=32.5
B	(17+12+18)/2=23.5
C	(21+12+14)/2=23.5
D	(27+18+14)/2=29.5

Average distance

Neighbor Joining Algorithm

	A	B	C	D
A	-	17	21	27
B		-	12	18
C			-	14
D				-

i	u_i
A	(17+21+27)/2=32.5
B	(17+12+18)/2=23.5
C	(21+12+14)/2=23.5
D	(27+18+14)/2=29.5

	A	B	C	D
A	-	-39	-35	-35
B		-	-35	-35
C			-	-39
D				-

$D_{ij}-u_i-u_j$

Usually denoted as *Q* matrix

Neighbor Joining Algorithm

	A	B	C	D
A	-	17	21	27
B		-	12	18
C			-	14
D				-

i	u_i
A	(17+21+27)/2=32.5
B	(17+12+18)/2=23.5
C	(21+12+14)/2=23.5
D	(27+18+14)/2=29.5

	A	B	C	D
A	-	-39	-35	-35
B		-	-35	-35
C			-	-39
D				-

$D_{ij}-u_i-u_j$

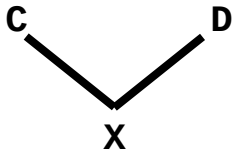
Neighbor Joining Algorithm

	A	B	C	D
A	-	17	21	27
B		-	12	18
C			-	14
D				-

i	u_i
A	(17+21+27)/2=32.5
B	(17+12+18)/2=23.5
C	(21+12+14)/2=23.5
D	(27+18+14)/2=29.5

	A	B	C	D
A	-	-39	-35	-35
B		-	-35	-35
C			-	-39
D				-

$D_{ij}-u_i-u_j$



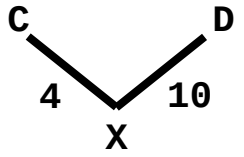
Neighbor Joining Algorithm

	A	B	C	D
A	-	17	21	27
B		-	12	18
C			-	14
D				-

i	u_i
A	(17+21+27)/2=32.5
B	(17+12+18)/2=23.5
C	(21+12+14)/2=23.5
D	(27+18+14)/2=29.5

	A	B	C	D
A	-	-39	-35	-35
B		-	-35	-35
C			-	-39
D				-

$D_{ij}-u_i-u_j$

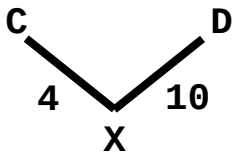


$b_C = 0.5 \times 14 + 0.5 \times (23.5-29.5) = 4$

$b_D = 0.5 \times 14 + 0.5 \times (29.5-23.5) = 10$

Neighbor Joining Algorithm

	A	B	C	D	X
A	-	17	21	27	
B		-	12	18	
C			-	14	
D				-	
X					-

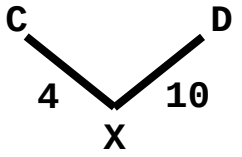


Neighbor Joining Algorithm

	A	B	C	D	X
A	-	17	21	27	
B		-	12	18	
C			-	14	
D				-	
X					-

$$D_{XA} = (D_{CA} + D_{DA} - D_{CD})/2$$
$$= (21 + 27 - 14)/2$$
$$= 17$$

$$D_{XB} = (D_{CB} + D_{DB} - D_{CD})/2$$
$$= (12 + 18 - 14)/2$$
$$= 8$$

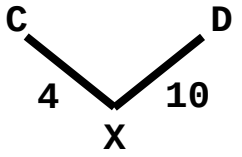


Neighbor Joining Algorithm

	A	B	C	D	X
A	-	17	21	27	17
B		-	12	18	8
C			-	14	
D				-	
X					-

$$D_{XA} = (D_{CA} + D_{DA} - D_{CD})/2$$
$$= (21 + 27 - 14)/2$$
$$= 17$$

$$D_{XB} = (D_{CB} + D_{DB} - D_{CD})/2$$
$$= (12 + 18 - 14)/2$$
$$= 8$$

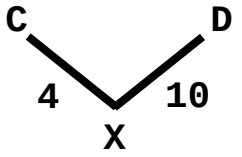


Neighbor Joining Algorithm

	A	B	X
A	-	17	17
B		-	8
X			-

$$D_{XA} = (D_{CA} + D_{DA} - D_{CD})/2$$
$$= (21 + 27 - 14)/2$$
$$= 17$$

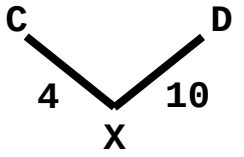
$$D_{XB} = (D_{CB} + D_{DB} - D_{CD})/2$$
$$= (12 + 18 - 14)/2$$
$$= 8$$



Neighbor Joining Algorithm

	A	B	X
A	-	17	17
B		-	8
X			-

i	u_i
A	(17+17)/1 = 34
B	(17+8)/1 = 25
X	(17+8)/1 = 25



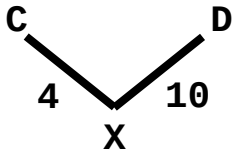
Neighbor Joining Algorithm

	A	B	X
A	-	17	17
B		-	8
X			-

i	u_i
A	(17+17)/1 = 34
B	(17+8)/1 = 25
X	(17+8)/1 = 25

	A	B	X
A	-	-42	-28
B		-	-28
X			-

$D_{ij}-u_i-u_j$



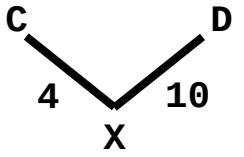
Neighbor Joining Algorithm

	A	B	X
A	-	17	17
B		-	8
X			-

i	u_i
A	(17+17)/1 = 34
B	(17+8)/1 = 25
X	(17+8)/1 = 25

	A	B	X
A	-	-42	-28
B		-	-28
X			-

$D_{ij}-u_i-u_j$



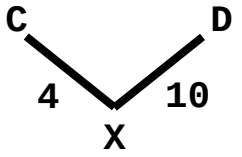
Neighbor Joining Algorithm

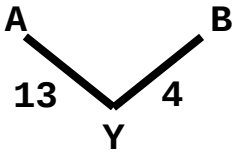
	A	B	X
A	-	17	17
B		-	8
X			-

i	u_i
A	(17+17)/1 = 34
B	(17+8)/1 = 25
X	(17+8)/1 = 25

	A	B	X
A	-	-42	-28
B		-	-28
X			-

$D_{ij}-u_i-u_j$



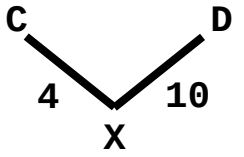


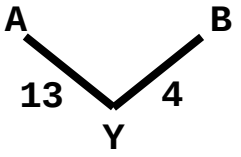
$b_A = 0.5 \times 17 + 0.5 \times (34-25) = 13$

$b_D = 0.5 \times 17 + 0.5 \times (25-34) = 4$

Neighbor Joining Algorithm

	A	B	X	Y
A	-	17	17	
B		-	8	
X			-	
Y				

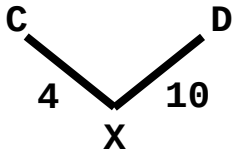


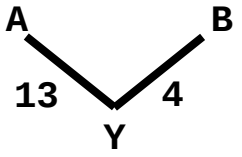


Neighbor Joining Algorithm

	A	B	X	Y
A	-	17	17	
B		-	8	
X			-	4
Y				

$$D_{YX} = (D_{AX} + D_{BX} - D_{AB})/2$$
$$= (17 + 8 - 17)/2$$
$$= 4$$

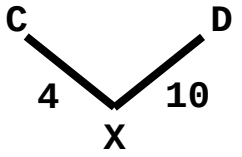


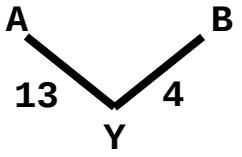


Neighbor Joining Algorithm

	X	Y
X	-	4
Y		-

$$D_{YX} = (D_{AX} + D_{BX} - D_{AB})/2$$
$$= (17 + 8 - 17)/2$$
$$= 4$$

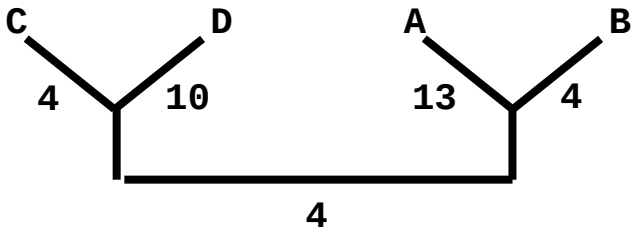




Neighbor Joining Algorithm

	X	Y
X	-	4
Y		-

$$D_{YX} = (D_{AX} + D_{BX} - D_{AB})/2$$
$$= (17 + 8 - 17)/2$$
$$= 4$$



Neighbor Joining Algorithm

	A	B	C	D
A	-	17	21	27
B		-	12	18
C			-	14
D				-