

Introduction to Bioinformatics for Computer Scientists

Lecture 5

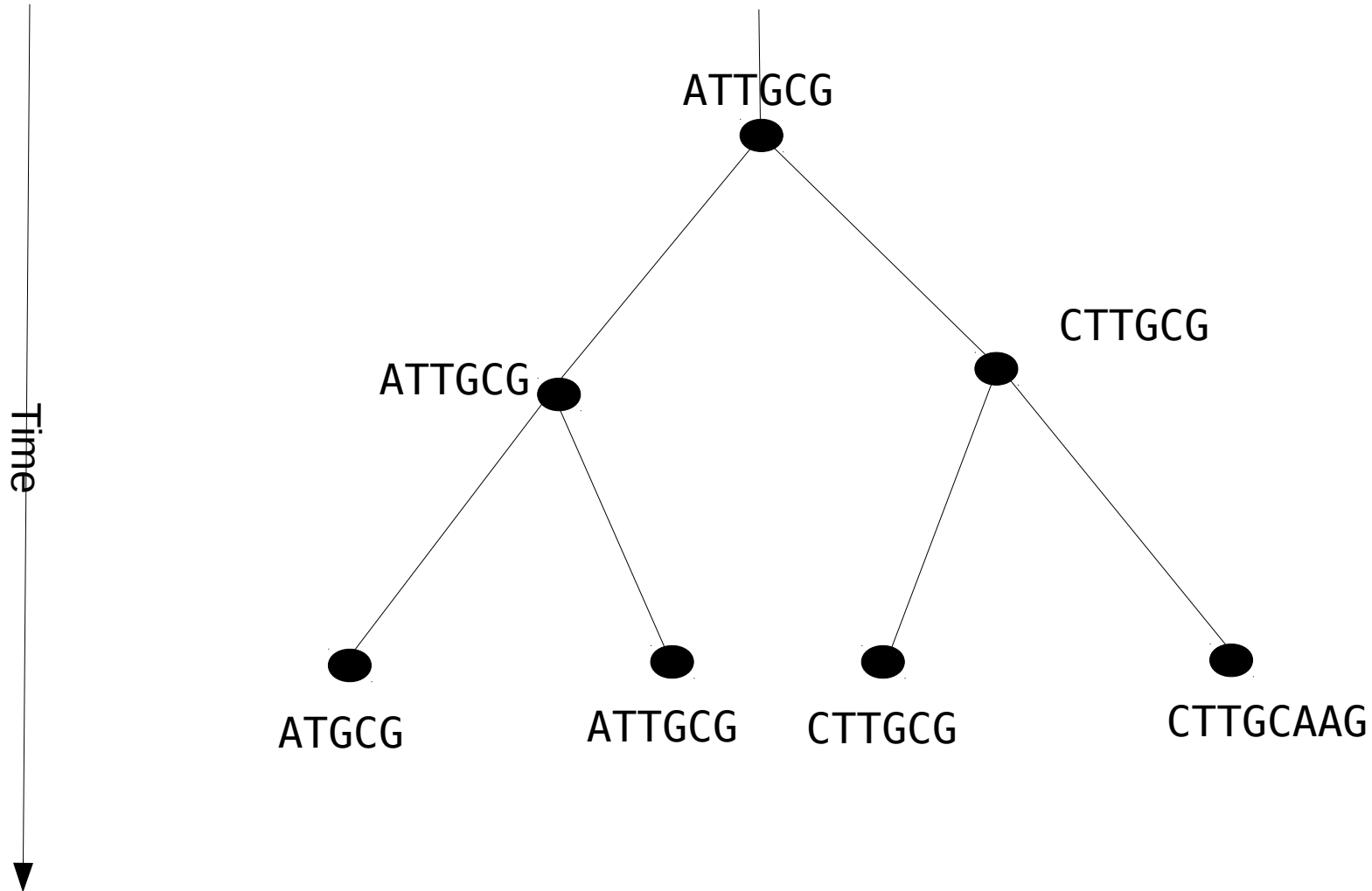
Plan for next lectures

- Today: Multiple Sequence Alignment
- Lecture 6: Introduction to phylogenetics
- Lecture 7: Phylogenetic search algorithms
- Lecture 8 (Alexis): Statistical Models of Evolution
- Lecture 9 (Pierre): Discrete Operations on Trees

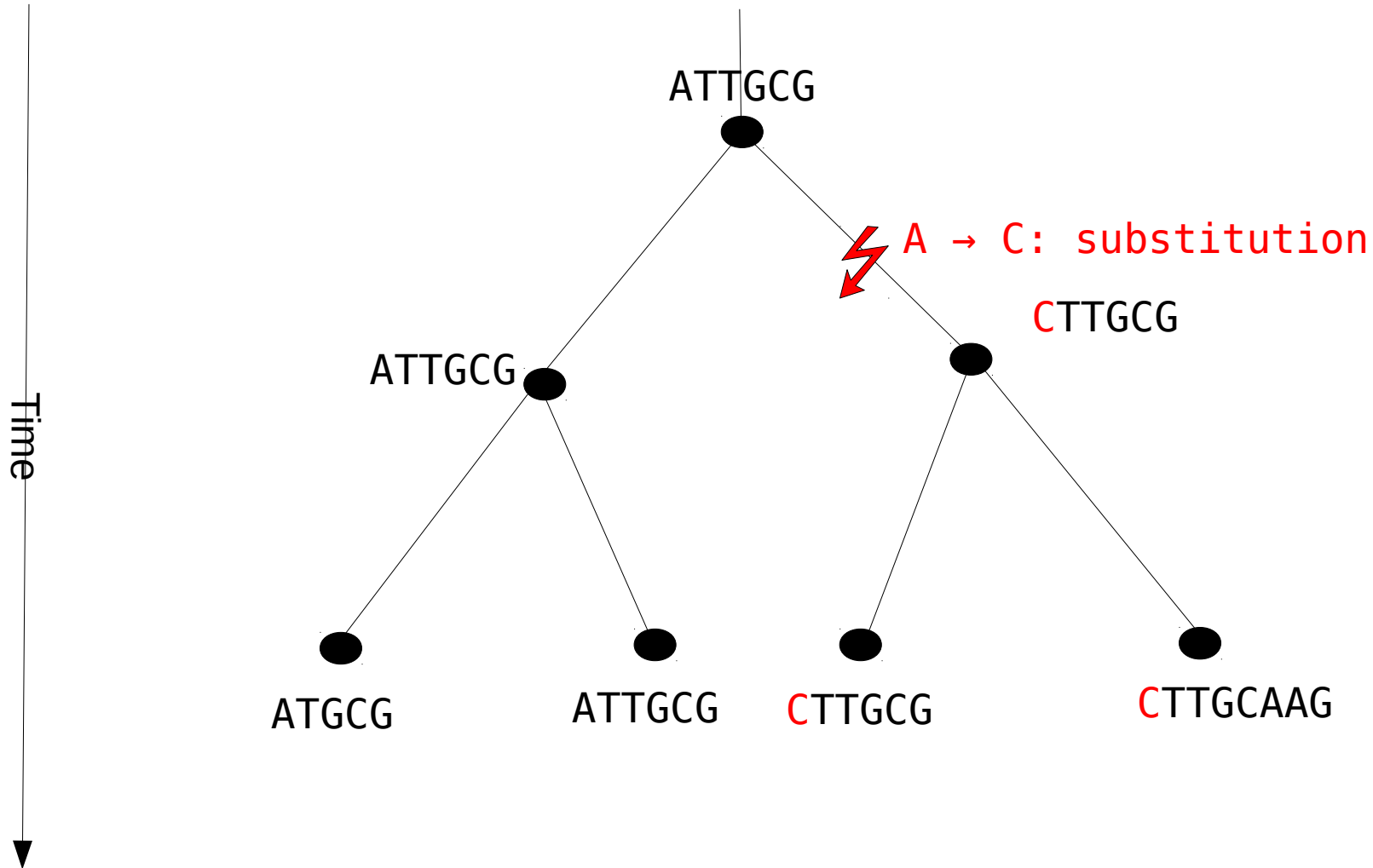
Multiple Sequence Alignment

- What are we trying to reconstruct?

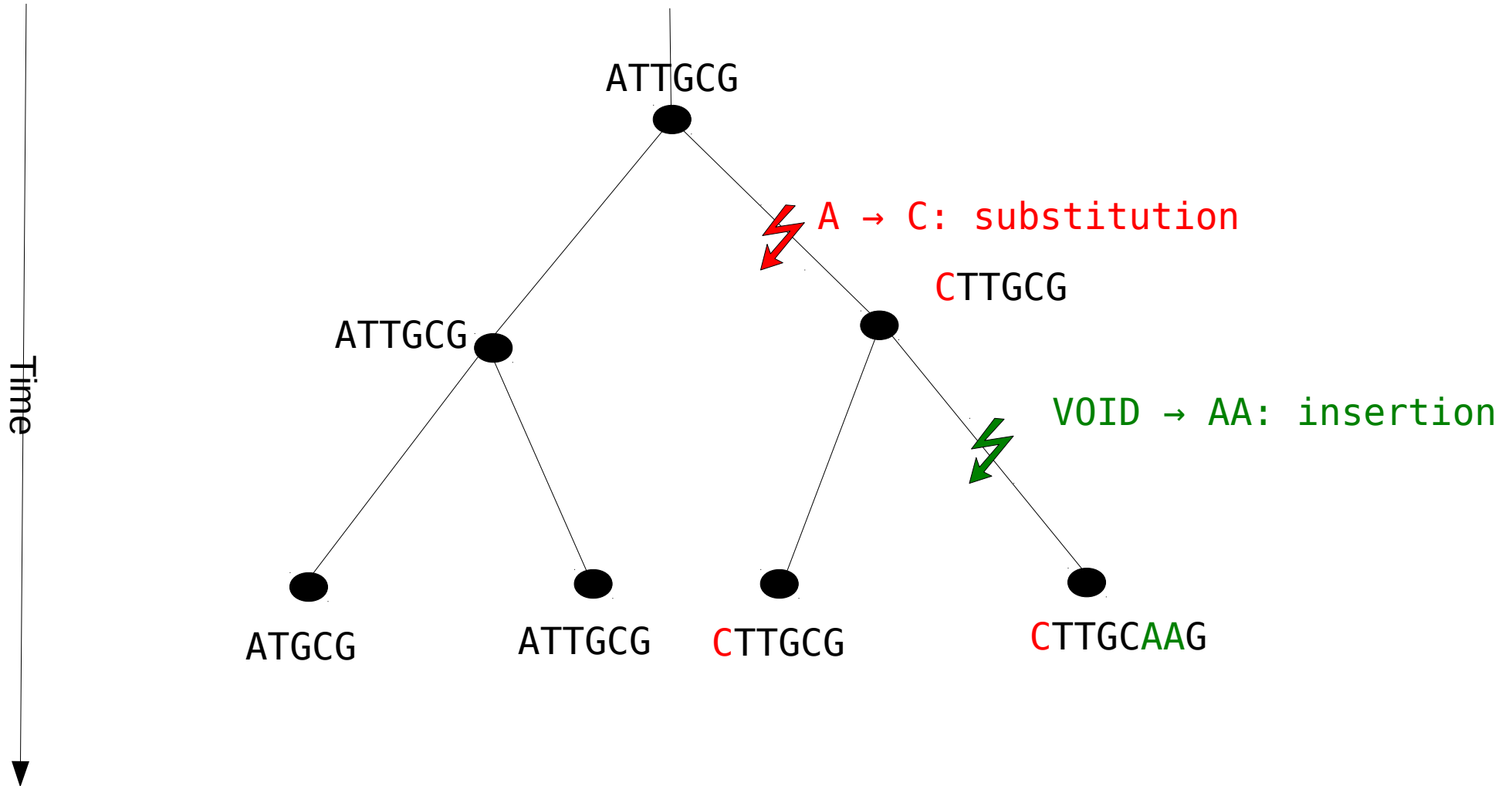
Insertions, Deletions & Substitutions



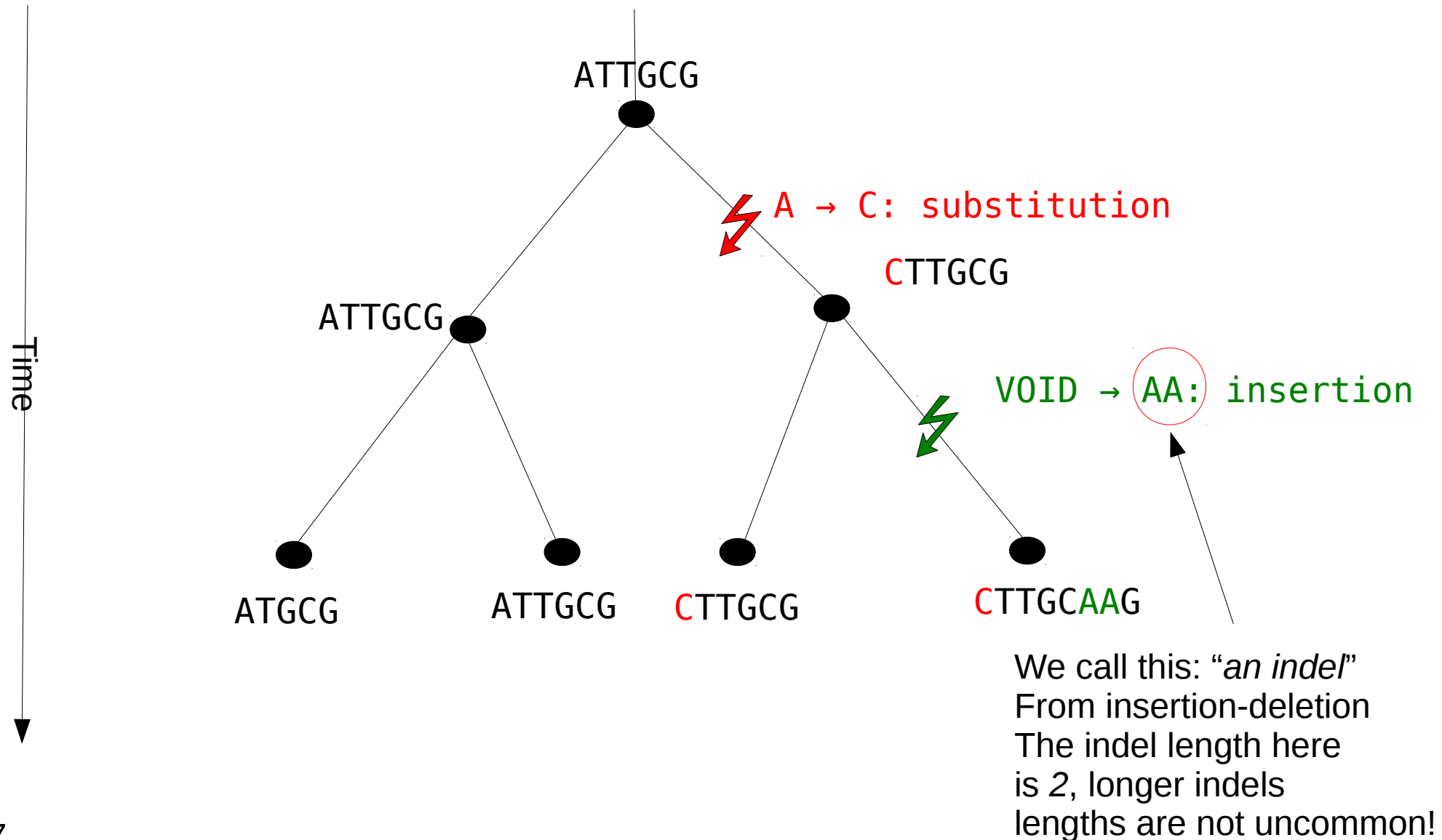
Insertions, Deletions & Substitutions



Insertions, Deletions & Substitutions

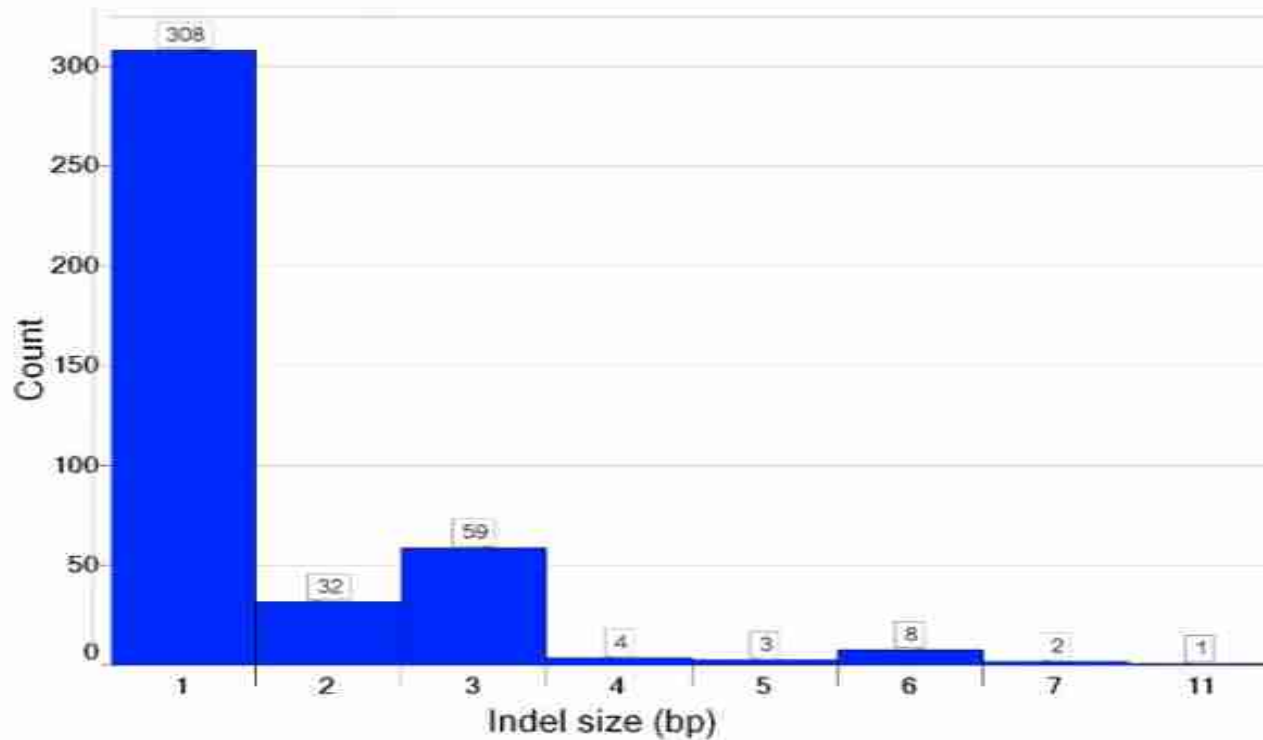


Insertions, Deletions & Substitutions



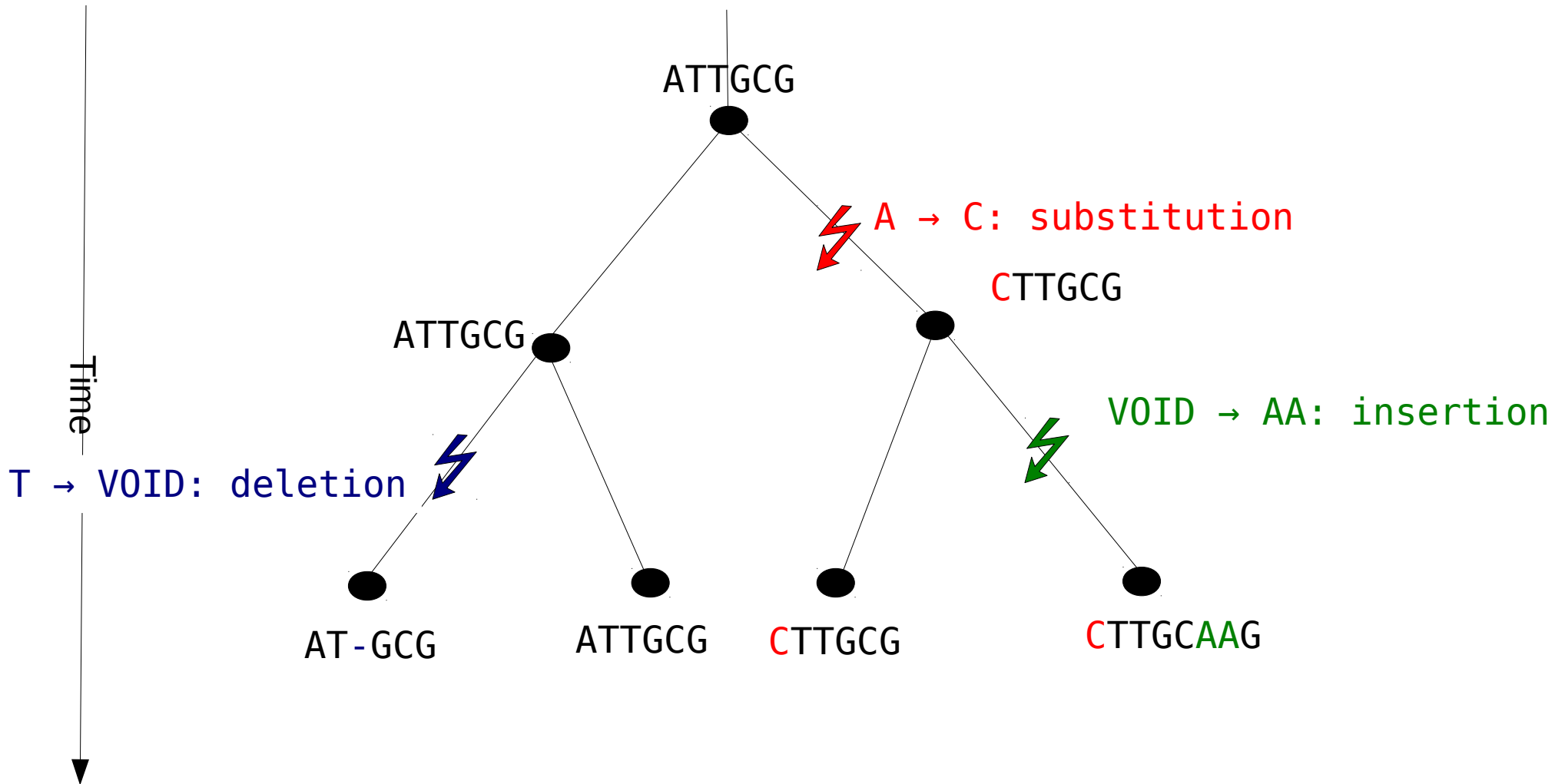
Indel size distribution

- Why are indels of size 3 rather frequent?

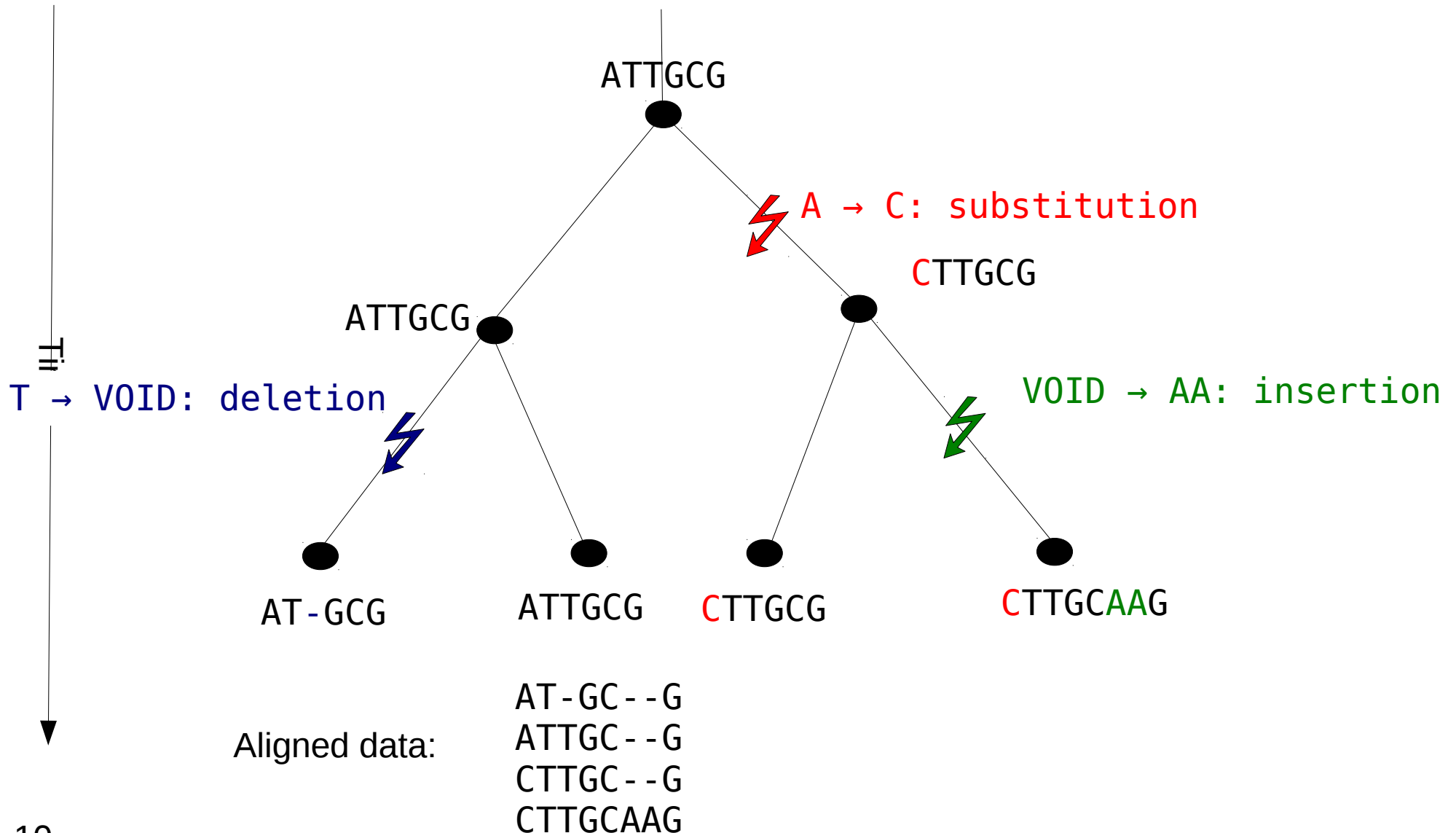


Indel size distribution in coding regions of cattle genomes.

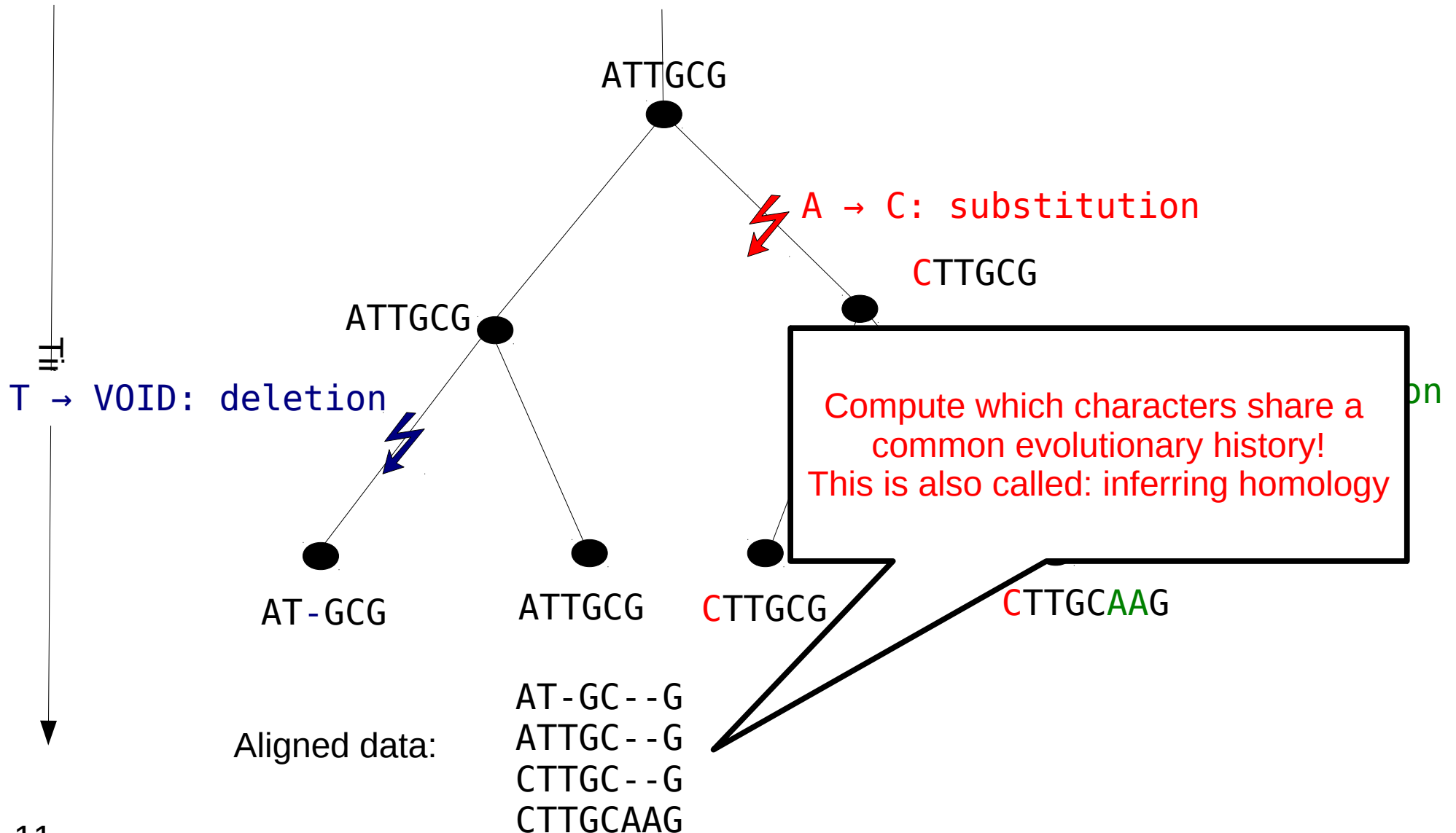
Insertions, Deletions & Substitutions



Insertions, Deletions & Substitutions



Insertions, Deletions & Substitutions



Multiple Sequence Alignment

- So far:
 - Comparing two sequences
 - Mapping a sequence/read to a reference genome
- 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 (conserved) parts of a *protein family*
- *Protein family* → group of evolutionarily 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, \dots, 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

s1	M	Q	P	I	L	L	L
s2	M	L	R	-	L	L	-
s3	M	K	-	I	L	L	L
s4	M	P	P	V	L	I	L



Alignment site/Alignment column

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

s1	M	Q	P	I	L	L	L
s2	M	L	R	-	L	L	-
s3	M	K	-	I	L	L	L
s4	M	P	P	V	L	I	L

Alignment site/Alignment column

MSA Terminology

s1	M	Q	P	I	L	L	L
s2	M	L	R	-	L	L	-
s3	M	K	-	I	L	L	L
s4	M	P	P	V	L	I	L

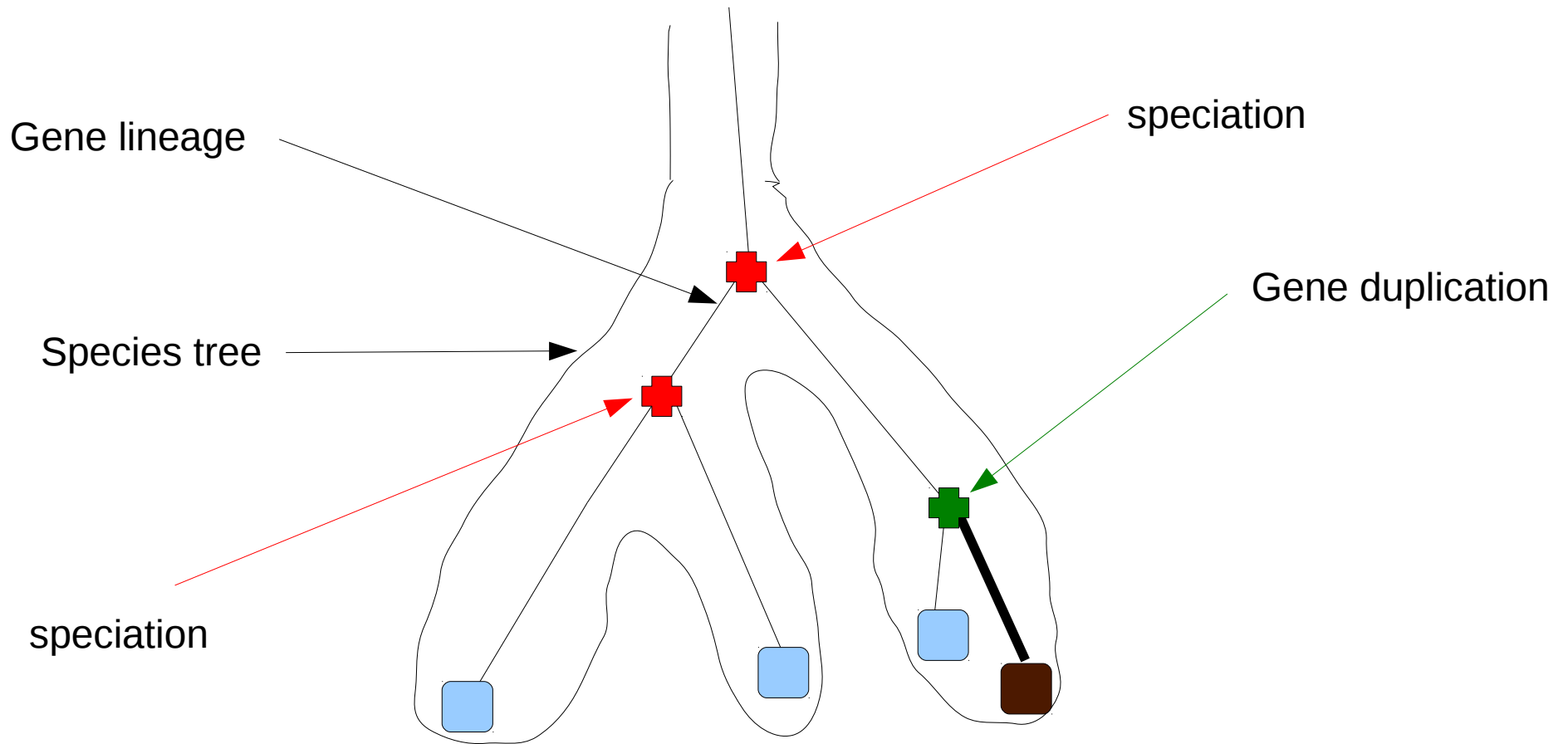
Homologous characters:
Characters that share a common evolutionary history

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

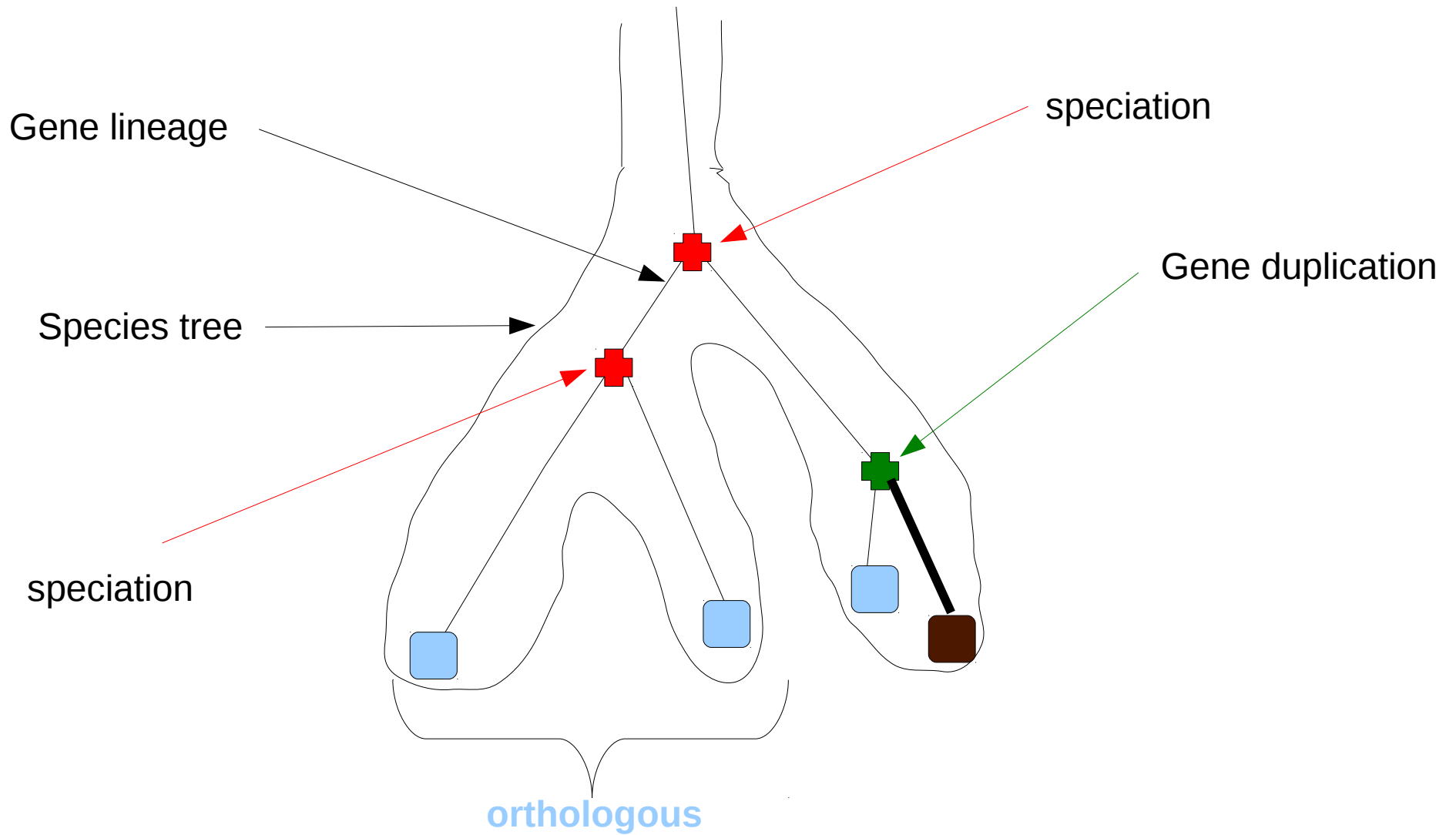
They could be similar by chance or via Convergent evolution (see slides later-on)

Alignment site/Alignment column

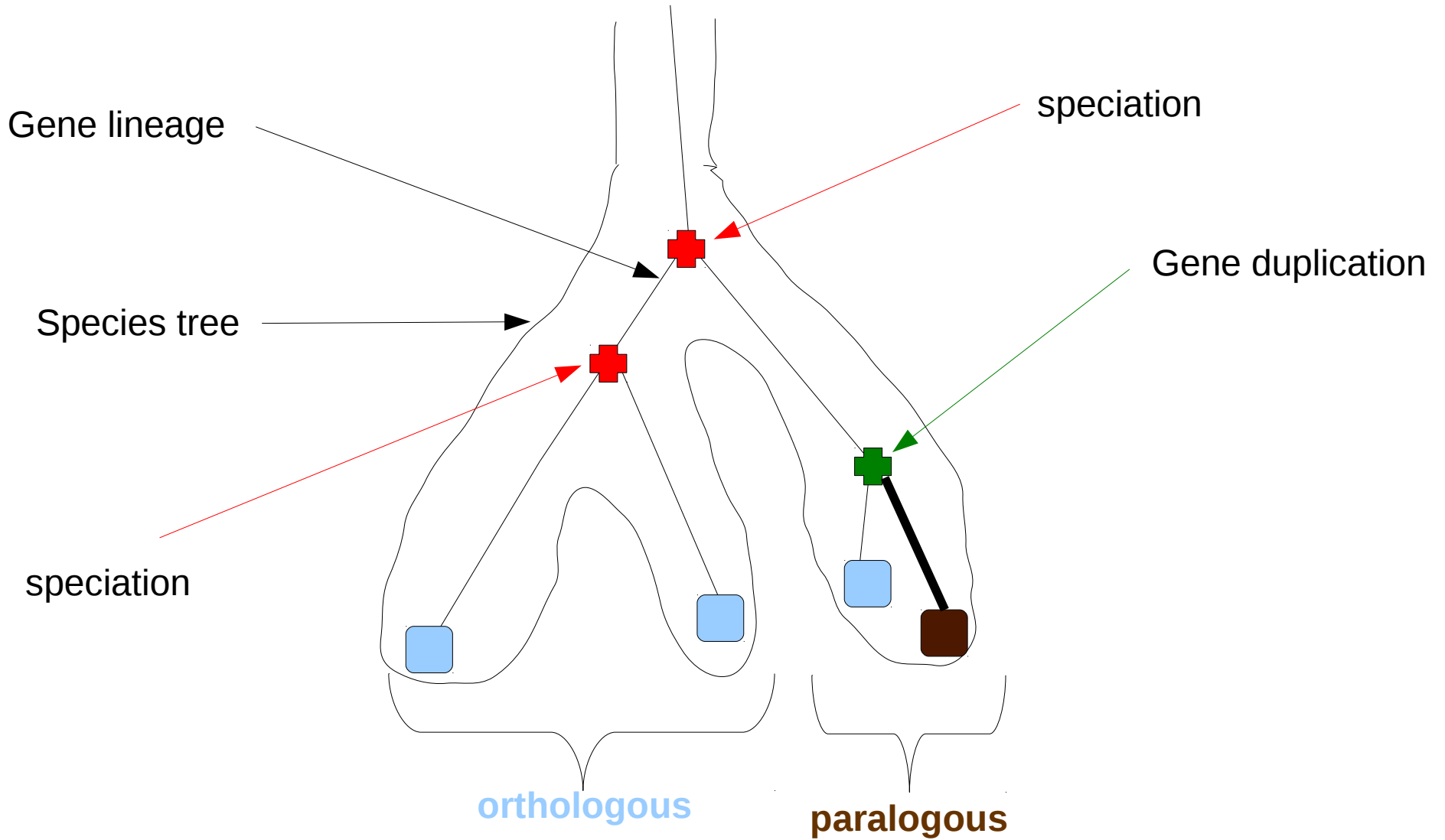
Orthology



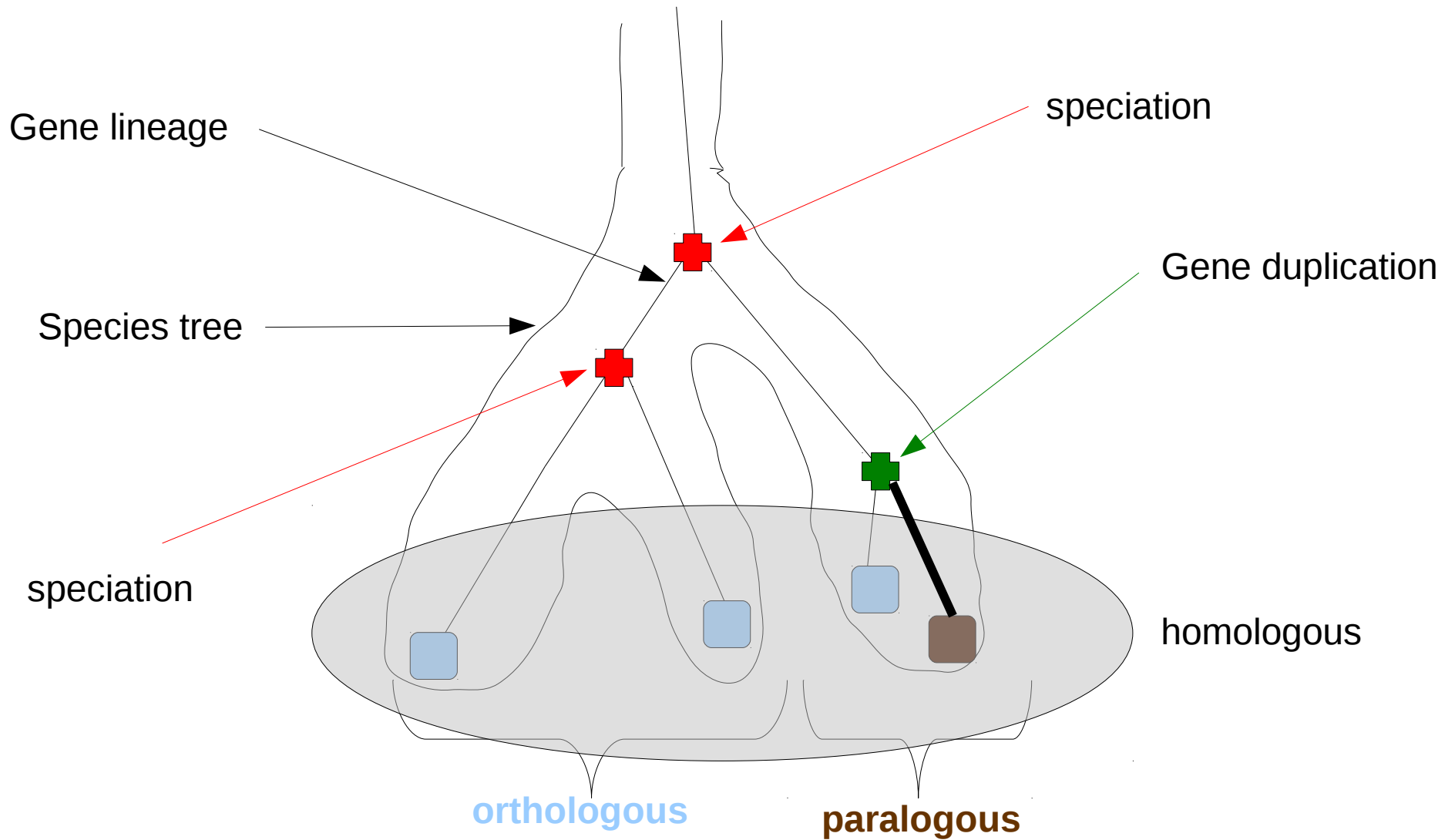
Orthology



Orthology



Orthology

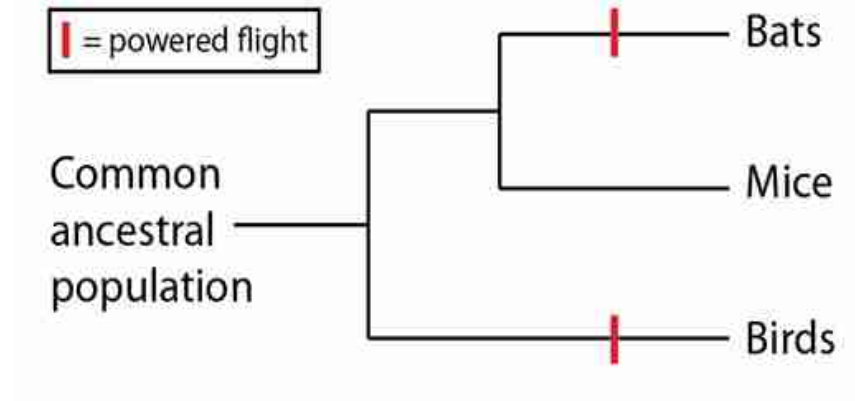
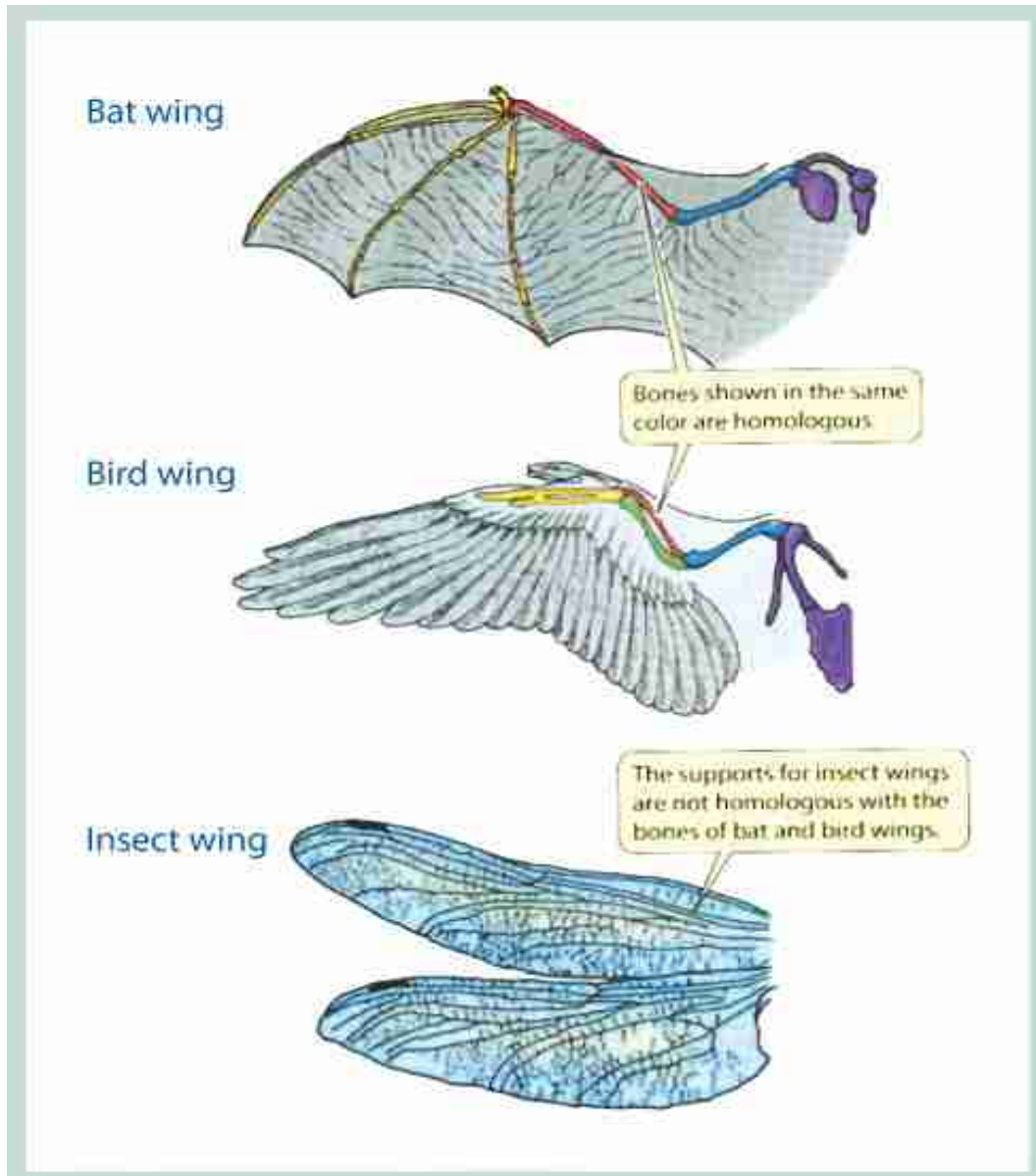


Homology

- 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
- Let's assume that we have a set of n orthologous sequences s_1, \dots, s_n and see how we can 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 of all pairwise scores between characters of a site

SP an example

- $SP\text{-score}(l, -, l, V) =$
 $p(l, -) + p(l, l) + p(l, V) + p(-, l) + p(-, V) + p(l, V)$
- 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 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

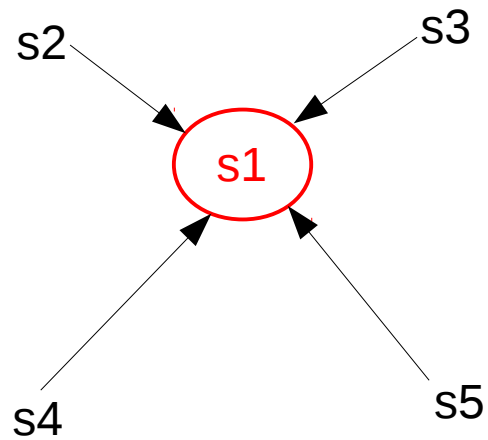
s2: ATGGCCATT

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--

← Gaps inserted

s3: ATC-CAATTTT

s1: ATTGCCATT--

← "Once a gap, always a gap"

s4: ATCTTC-TT--

s1: ATTGCCATT--

s5: ACTGACC----

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

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

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 - -

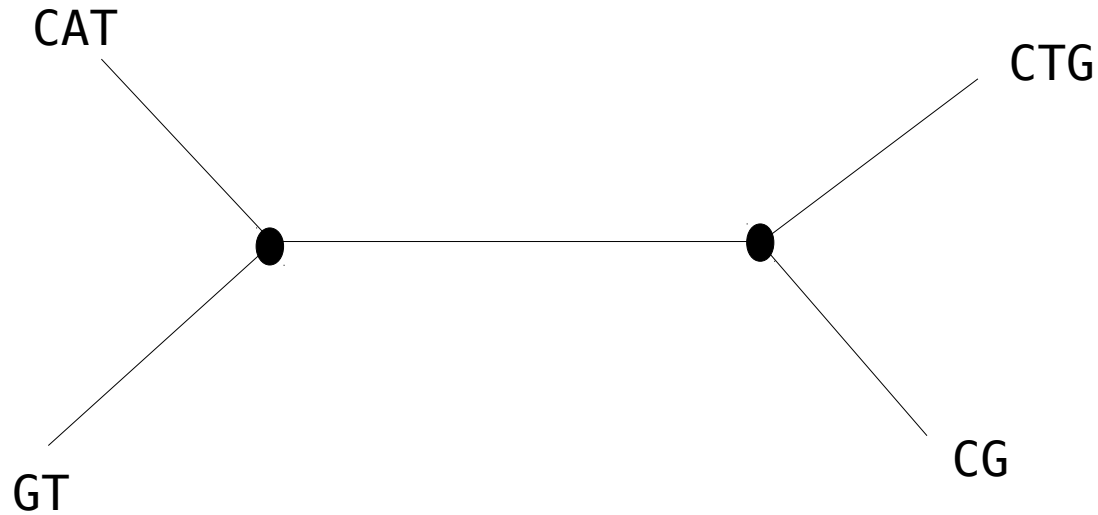
Shift right!

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.

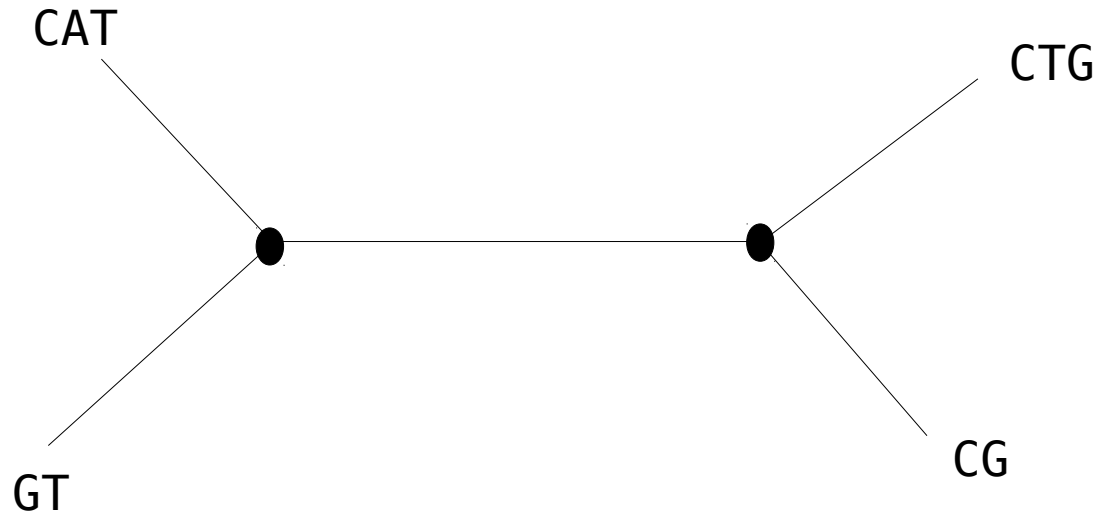
Tree Alignment

- 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

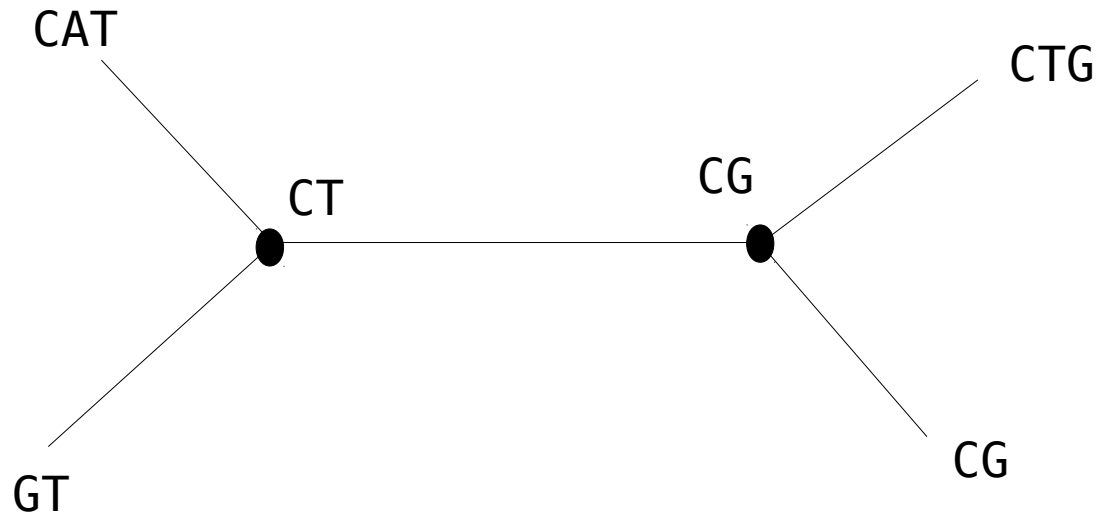


Tree Alignment

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

$p(a,b) := 0$ if $a \neq b$

$p(a,-) := -1$

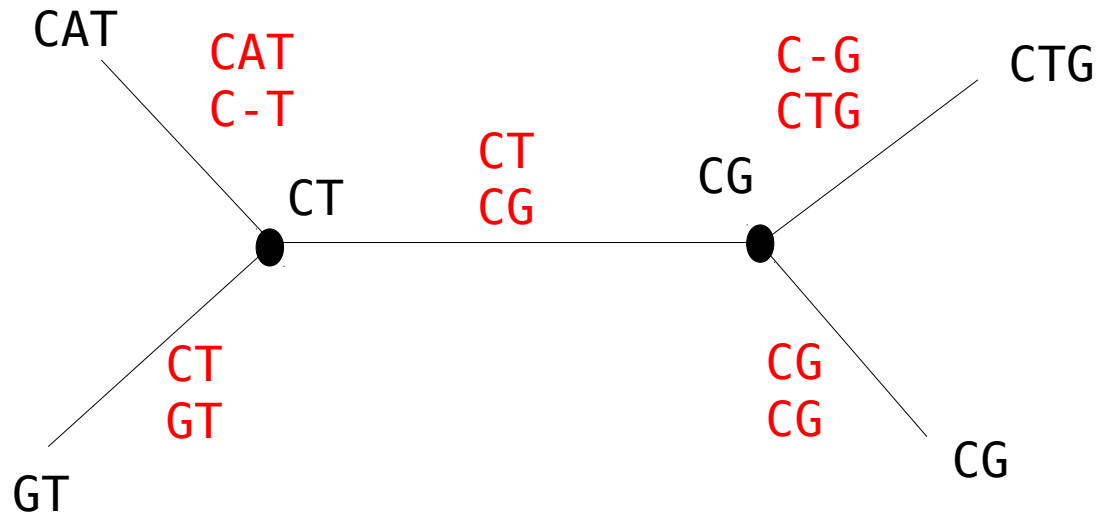


Tree Alignment

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

$p(a,b) := 0$ if $a \neq b$

$p(a,-) := -1$

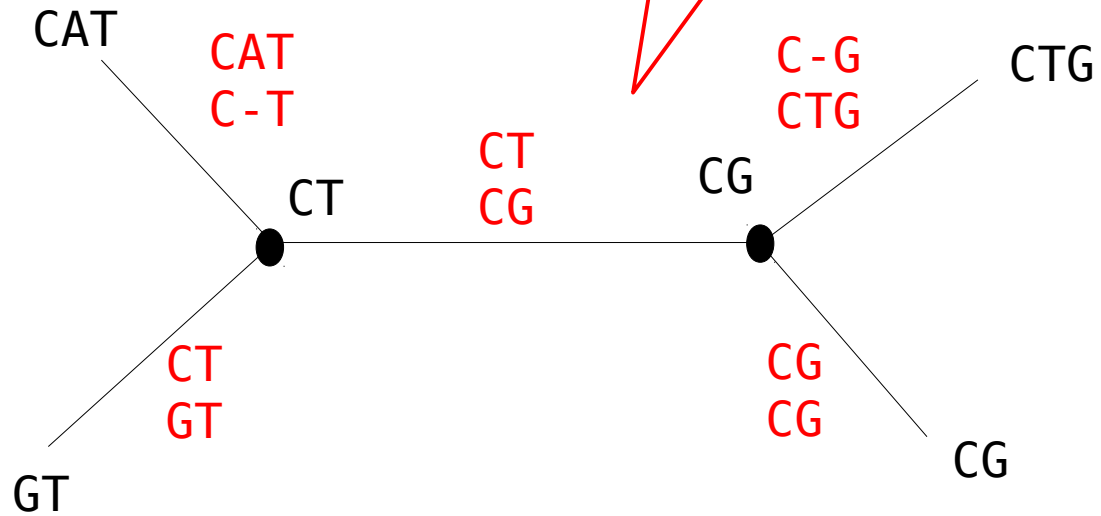


Tree Alignment

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

$p(a,b) := 0$ if $a \neq b$

$p(a,-) := -1$

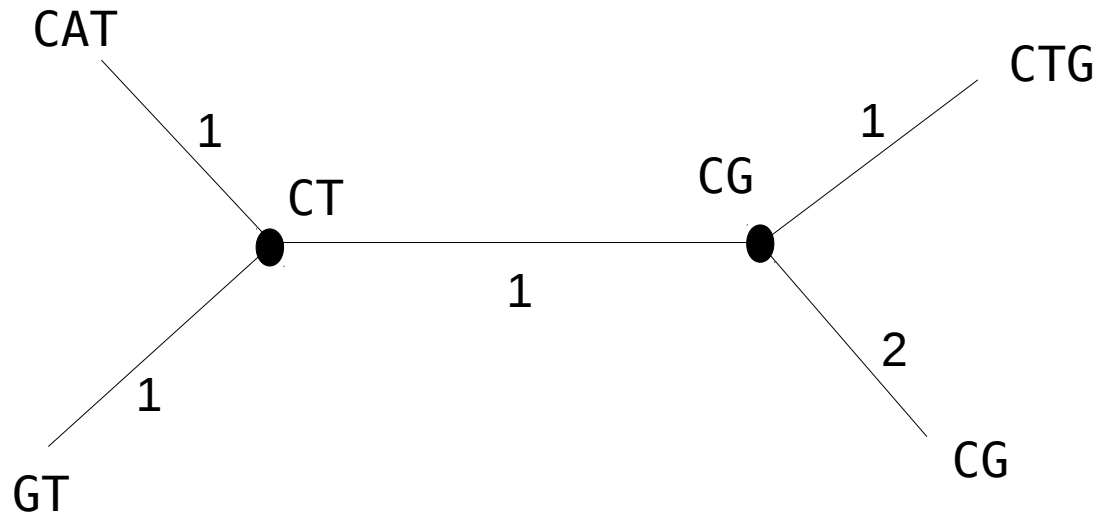


Tree Alignment

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

$p(a,b) := 0$ if $a \neq b$

$p(a,-) := -1$

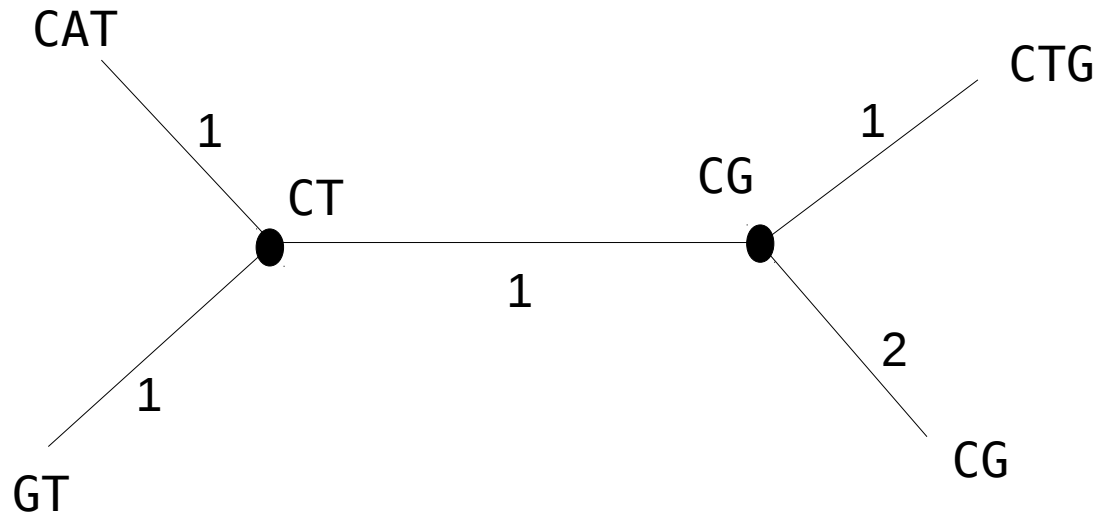


Tree Alignment

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

$p(a,b) := 0$ if $a \neq b$

$p(a,-) := -1$



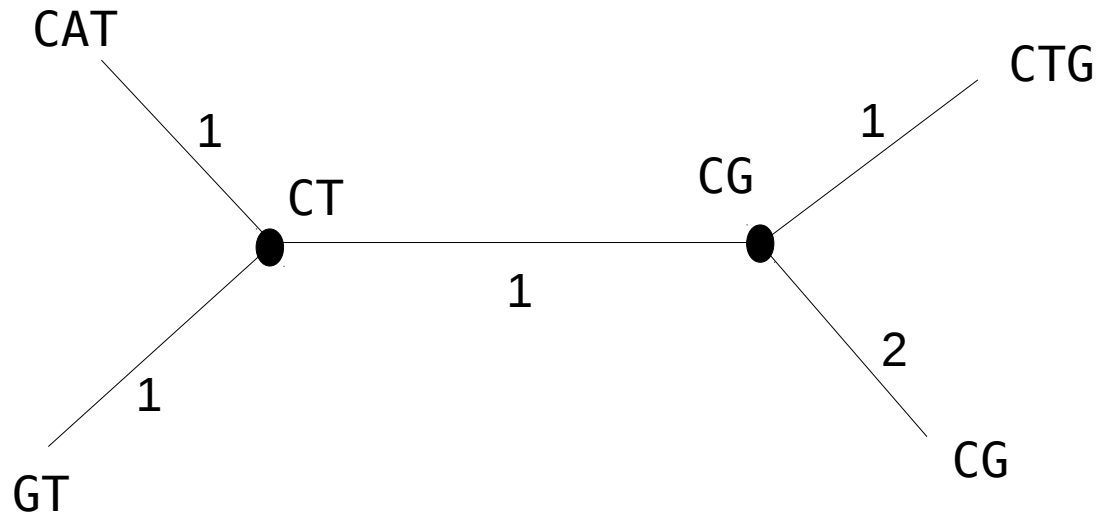
Overall score: 6 → maximize this score

Tree Alignment

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

$p(a,b) := 0$ if $a \neq b$

$p(a,-) := -1$



Overall score: 6 → maximize this score

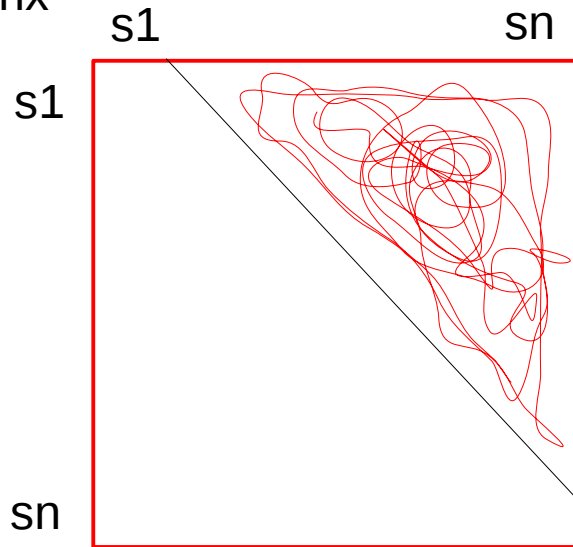
This problem is NP-hard because we don't have the ancestral states

Tree-Based Alignment

- Hen and egg problem
 - we need a 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

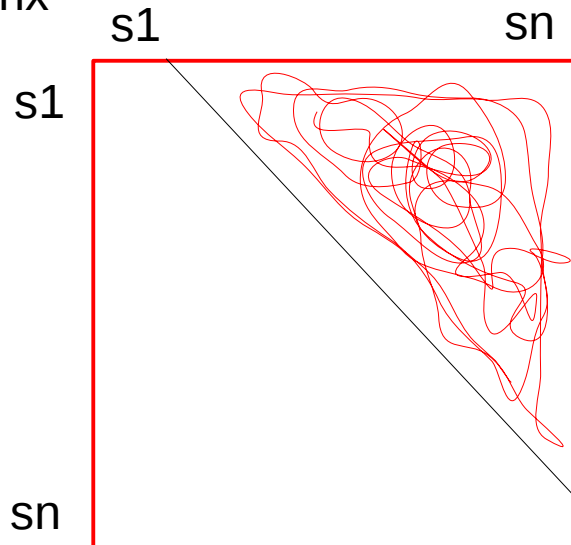
Practical approaches

Build a pair-wise
distance matrix



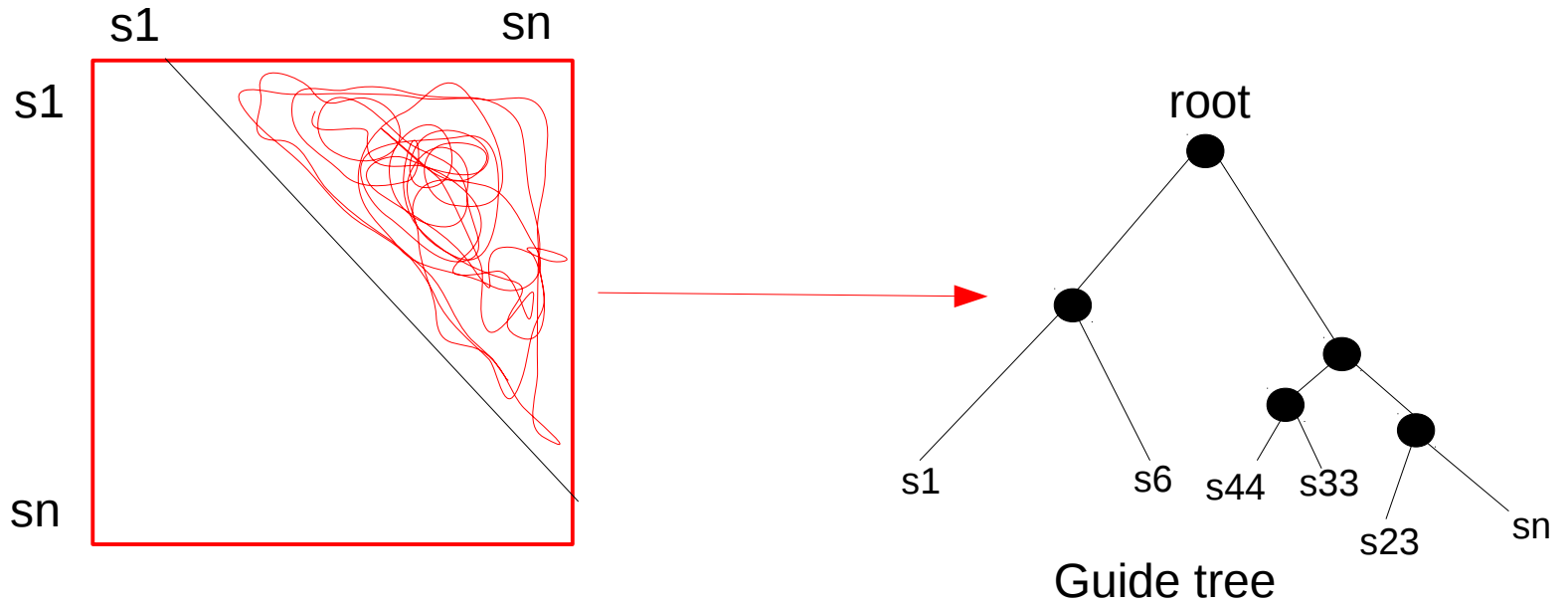
Practical approaches

Build a pair-wise distance matrix

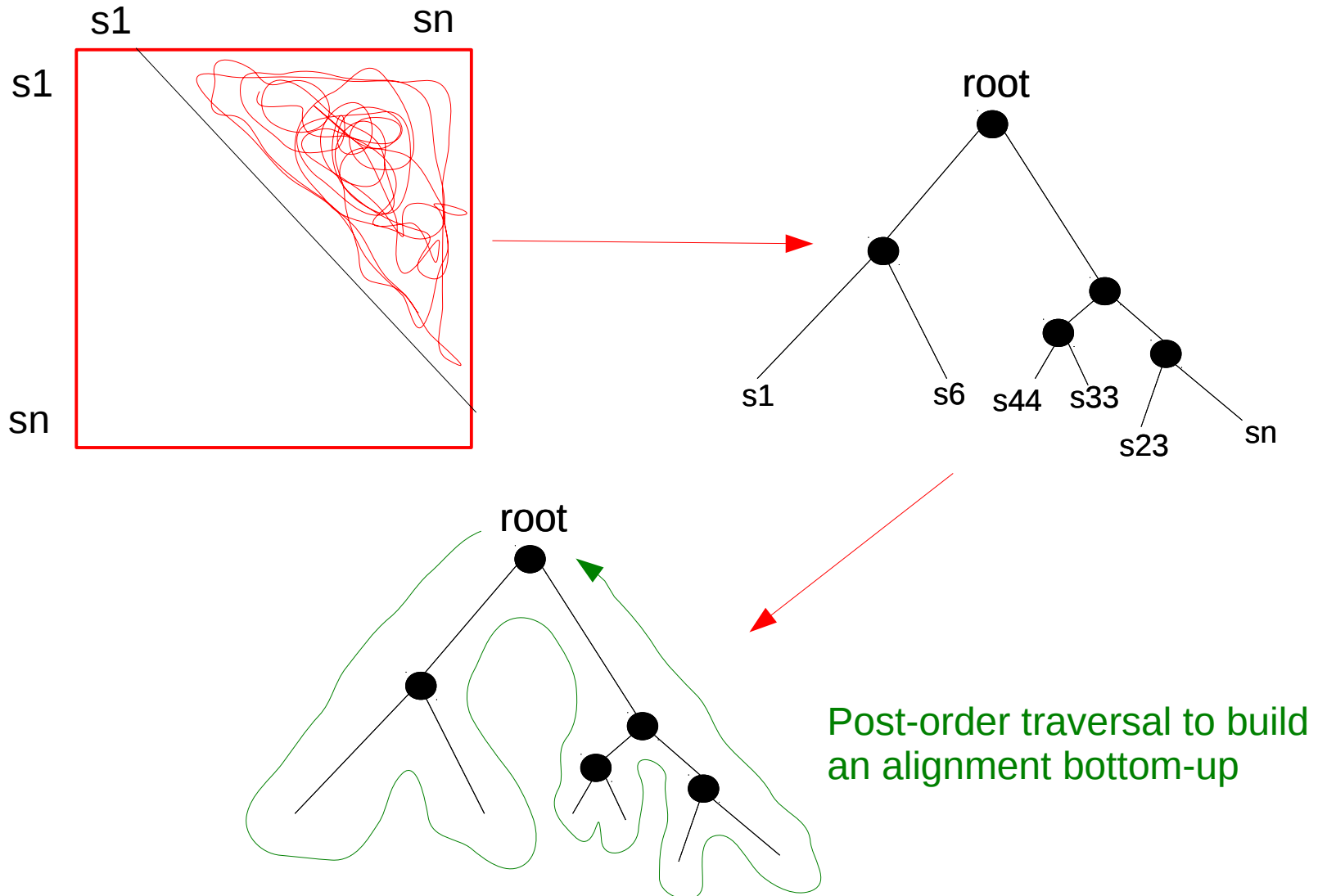


Computation of pair-wise distance matrix
Using pair-wise alignment scores can be time and memory-intensive due to $O(n^2)$ complexity
One may use approximate distance methods based on *k-mers*
(remember last lecture!)

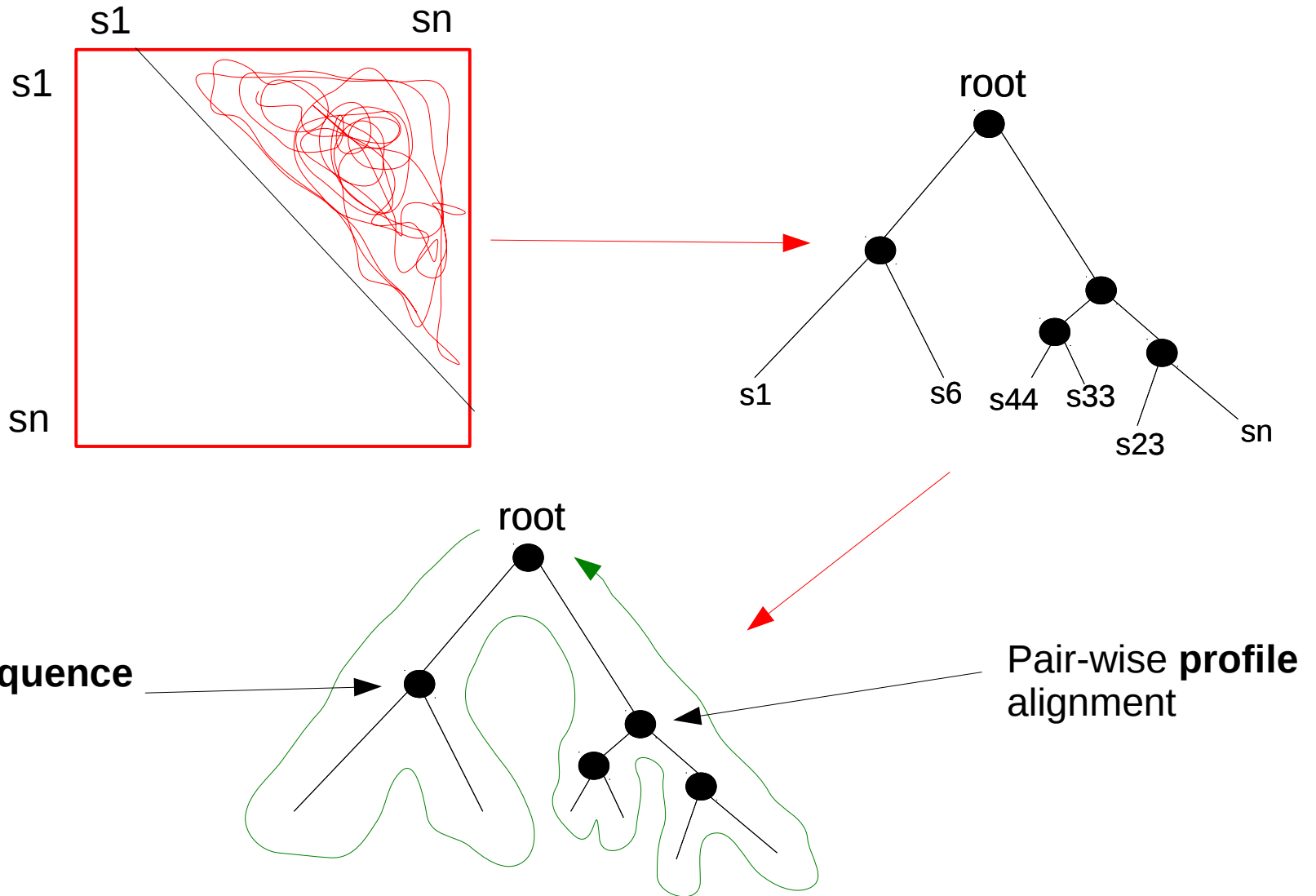
Practical approaches



Practical approaches



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
- 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**

Progressive MSA



AC



ATG



TCG

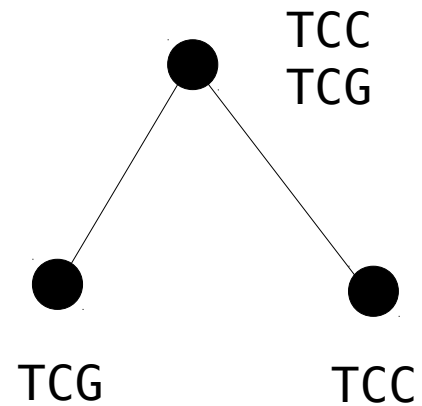
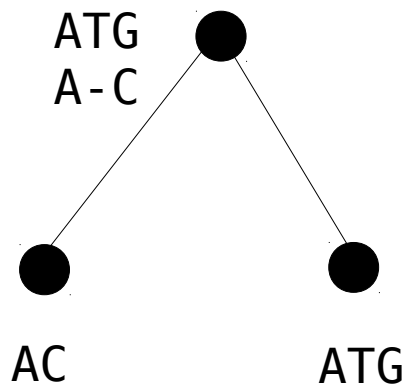


TCC

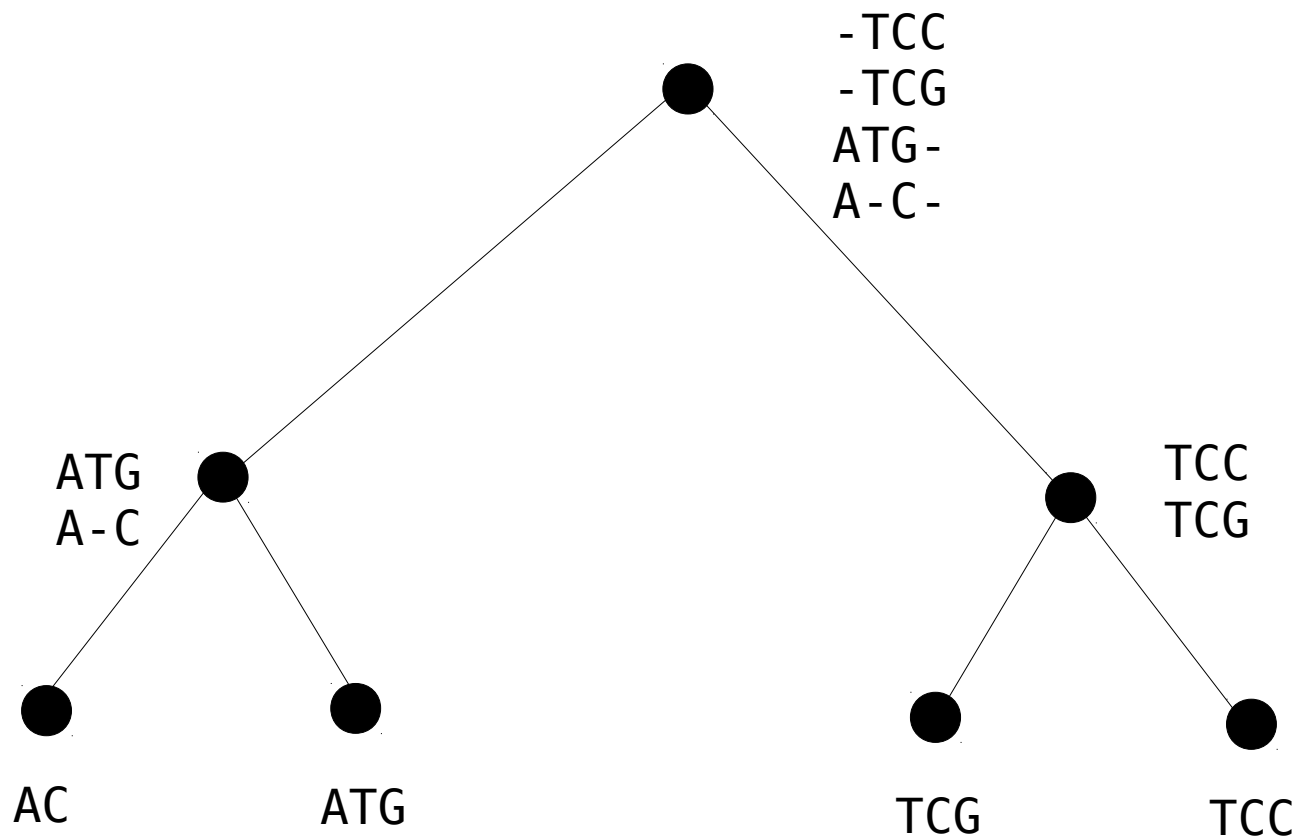
Progressive MSA



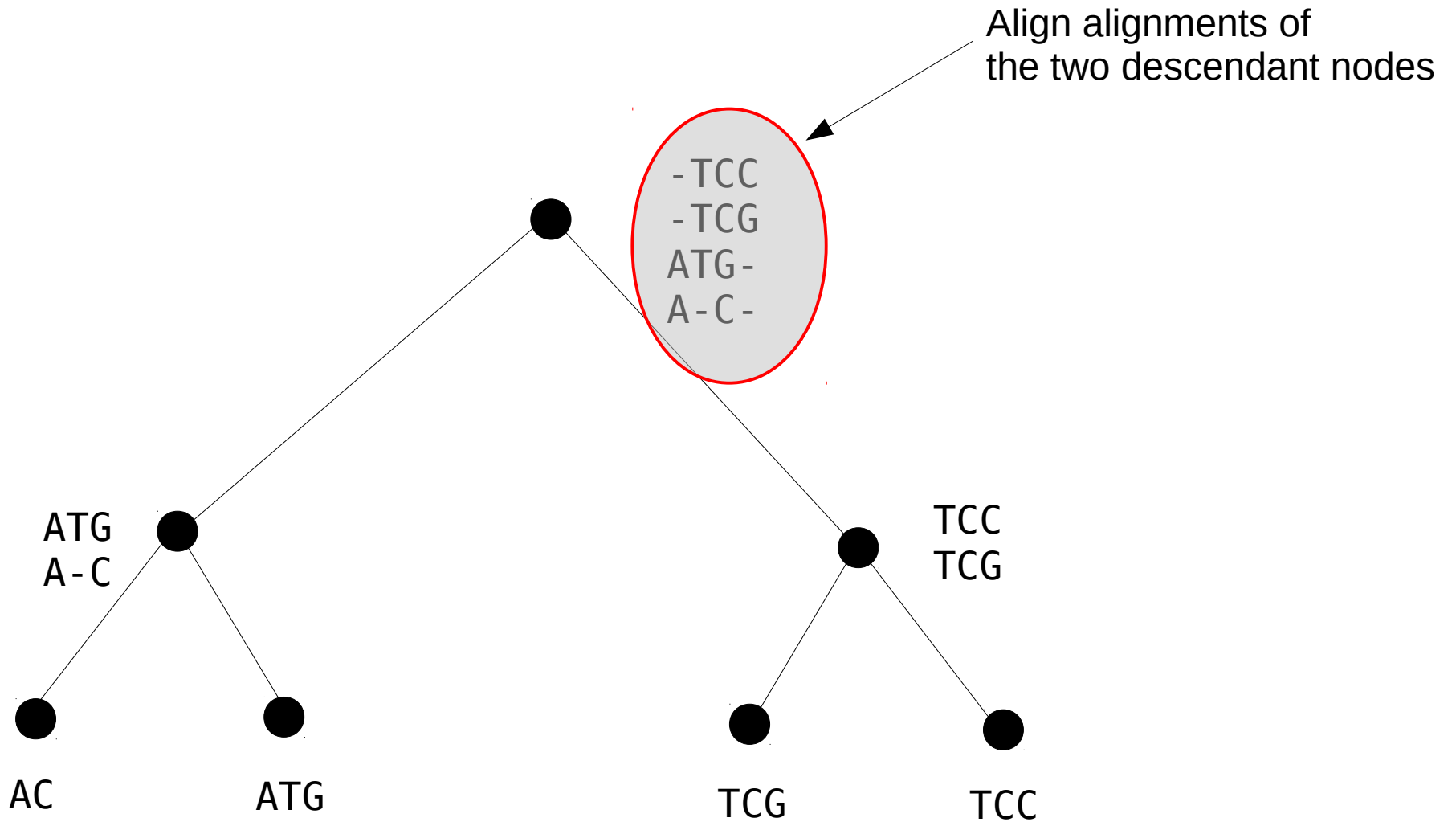
Progressive MSA



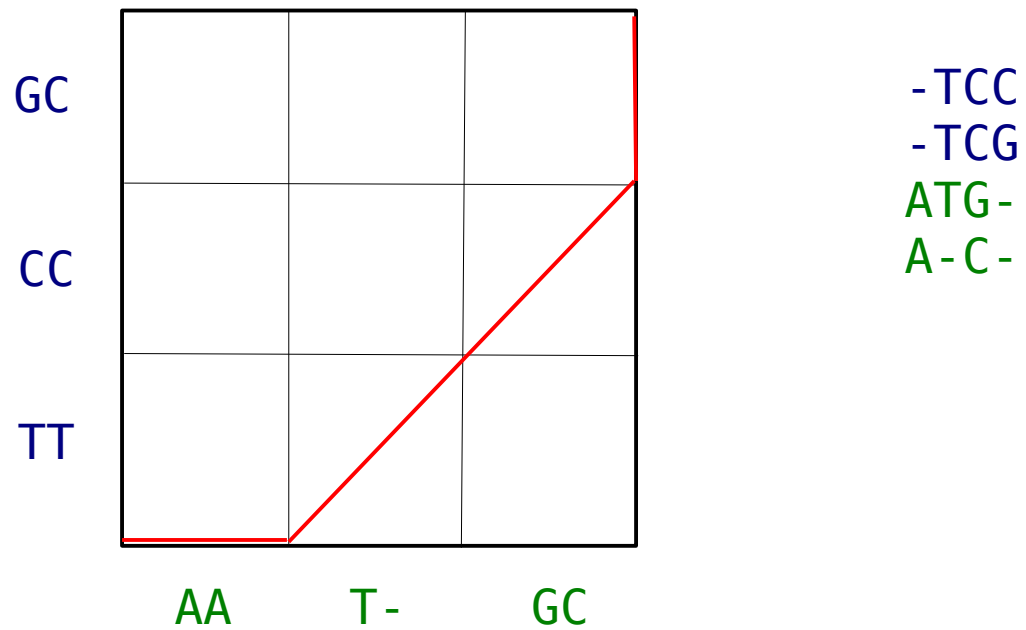
Progressive MSA



Progressive MSA

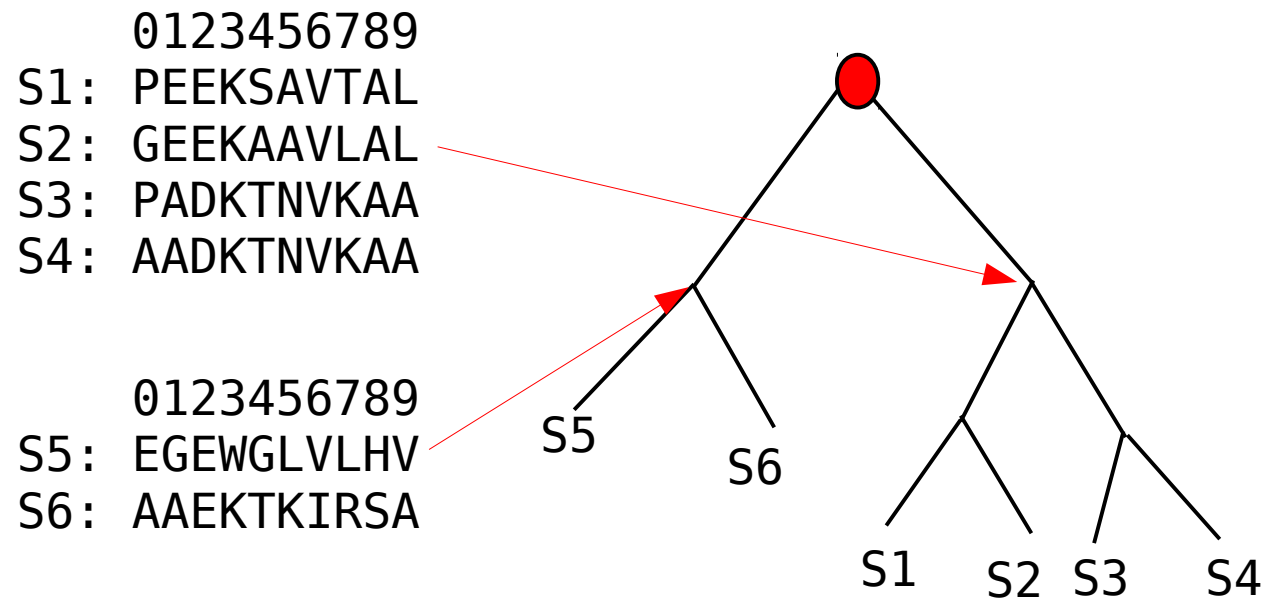


Profile Alignment



Profile Alignment

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

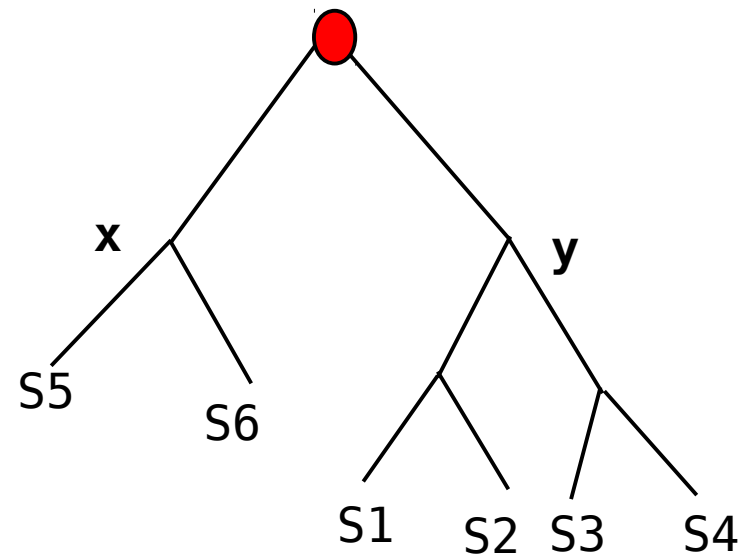


Profile Alignment

- 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



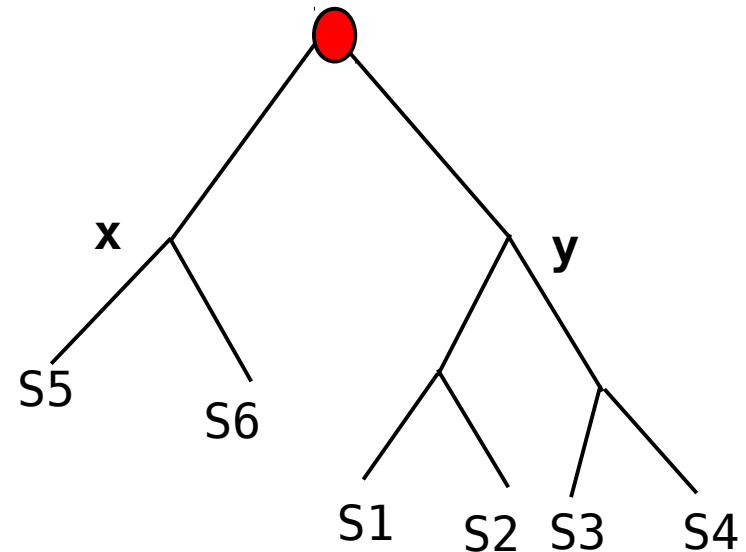
Compute score between position 6 of **x** and position 7 of **y**

Profile Alignment

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

	0	1	2	3	4	5	6	7	8	9
S1:	P	E	E	K	S	A	V	T	A	L
S2:	G	E	E	K	A	A	V	L	A	L
S3:	P	A	D	K	T	N	V	K	A	A
S4:	A	A	D	K	T	N	V	K	A	A

	0	1	2	3	4	5	6	7	8	9
S5:	E	G	E	W	G	L	V	L	H	V
S6:	A	A	E	K	T	K	I	R	S	A



Weighted average over all 8 (2 * 4) possibilities:

Score: $1/8 * [p(T,V) + p(T,I) + p(L, V) + p(L, I) + p(K,V) + p(K,I) + p(K,V) + p(K,I)]$

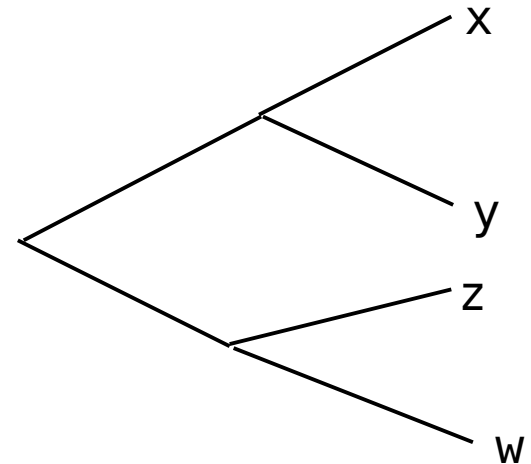
Problems with progressive MSA

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

x: GAAGTT
y: GAC-**TT** → frozen by initial alignment

z: GAA**CTG**
w: GTA**CTG** } y: GA-**CTT**

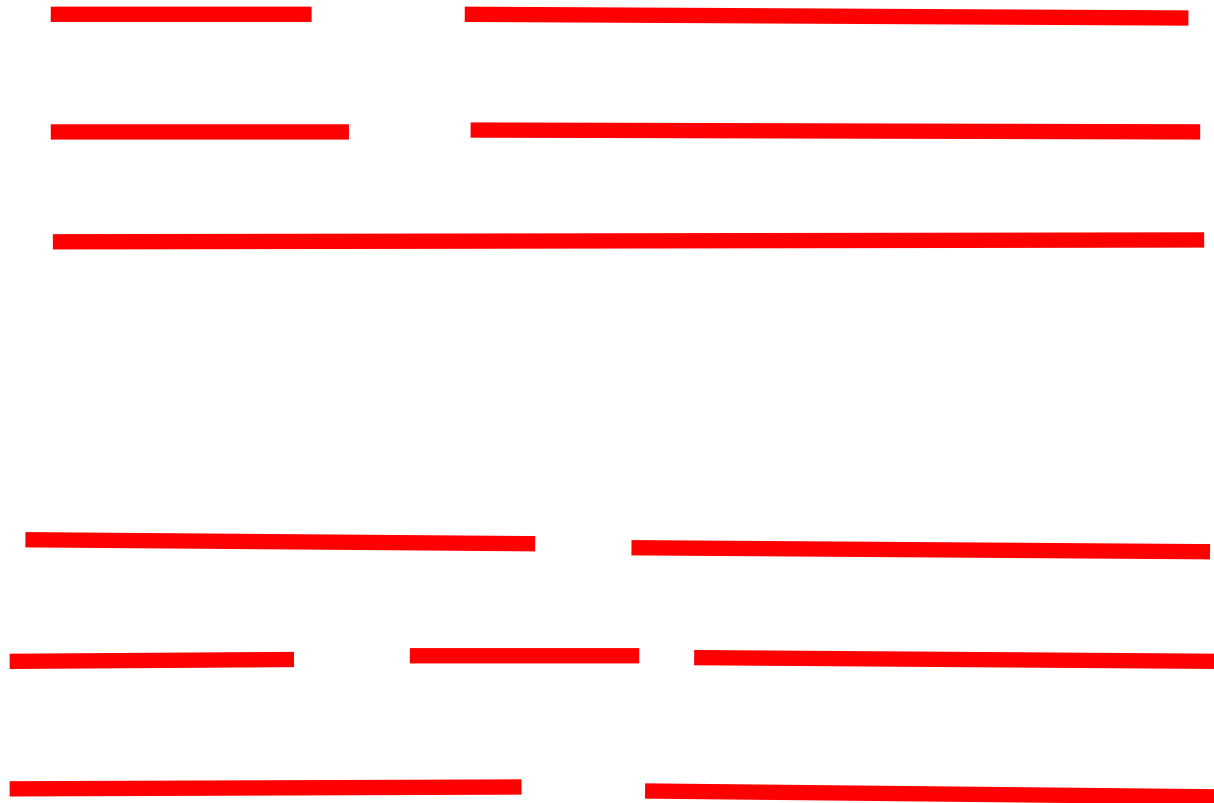
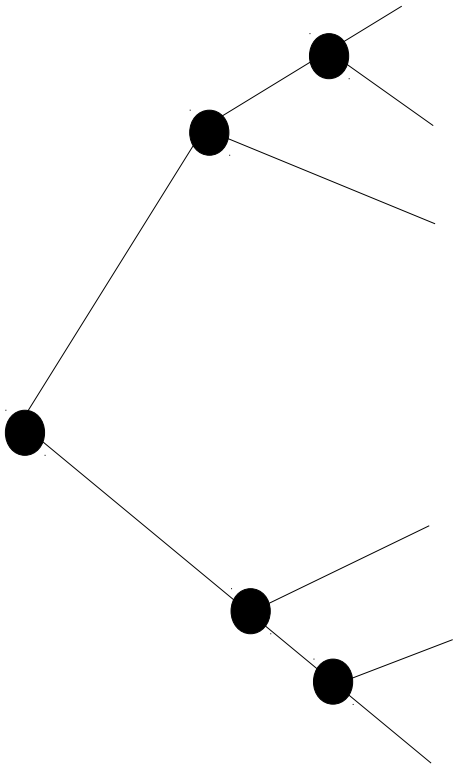
should be flipped



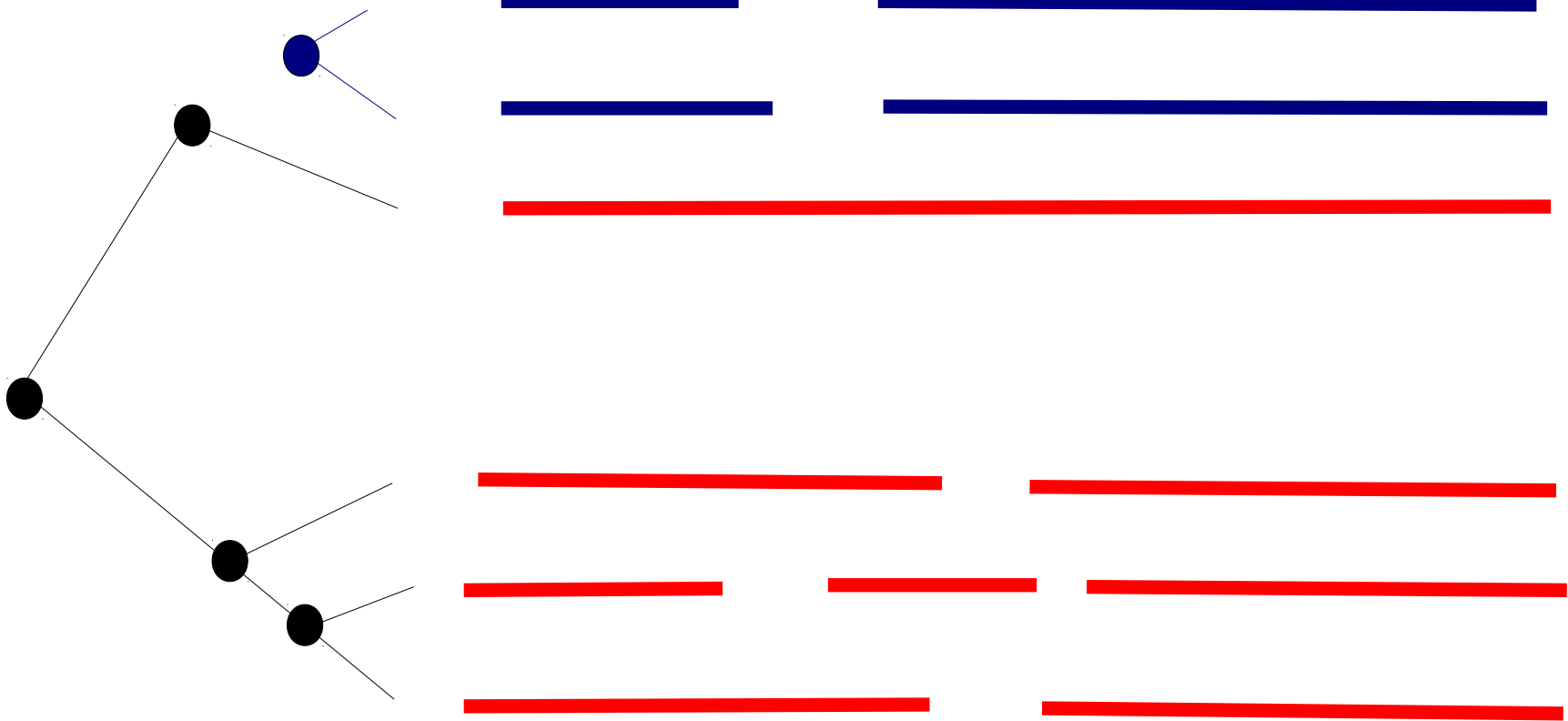
Iterative Progressive MSA

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

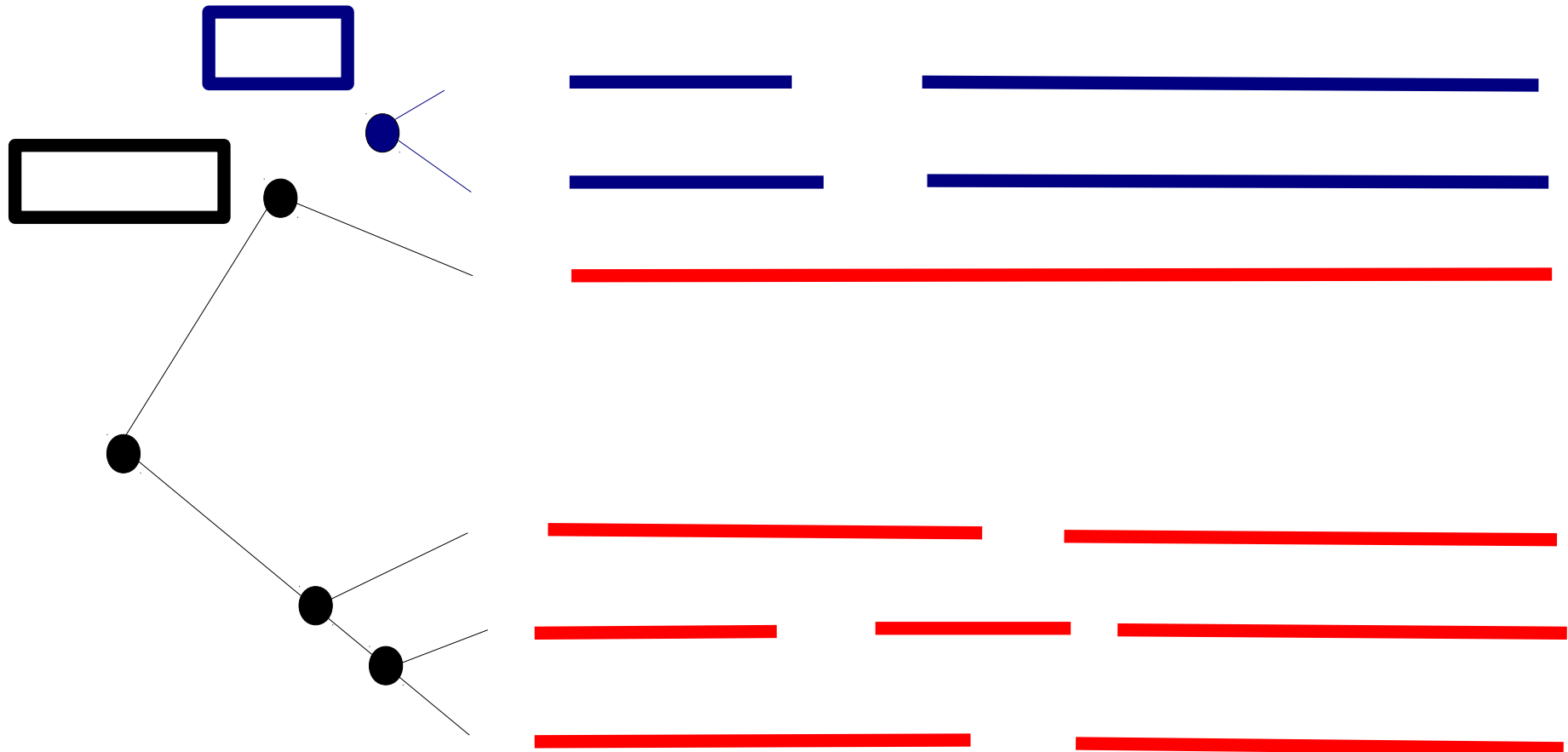
MUSCLE Re-Finement



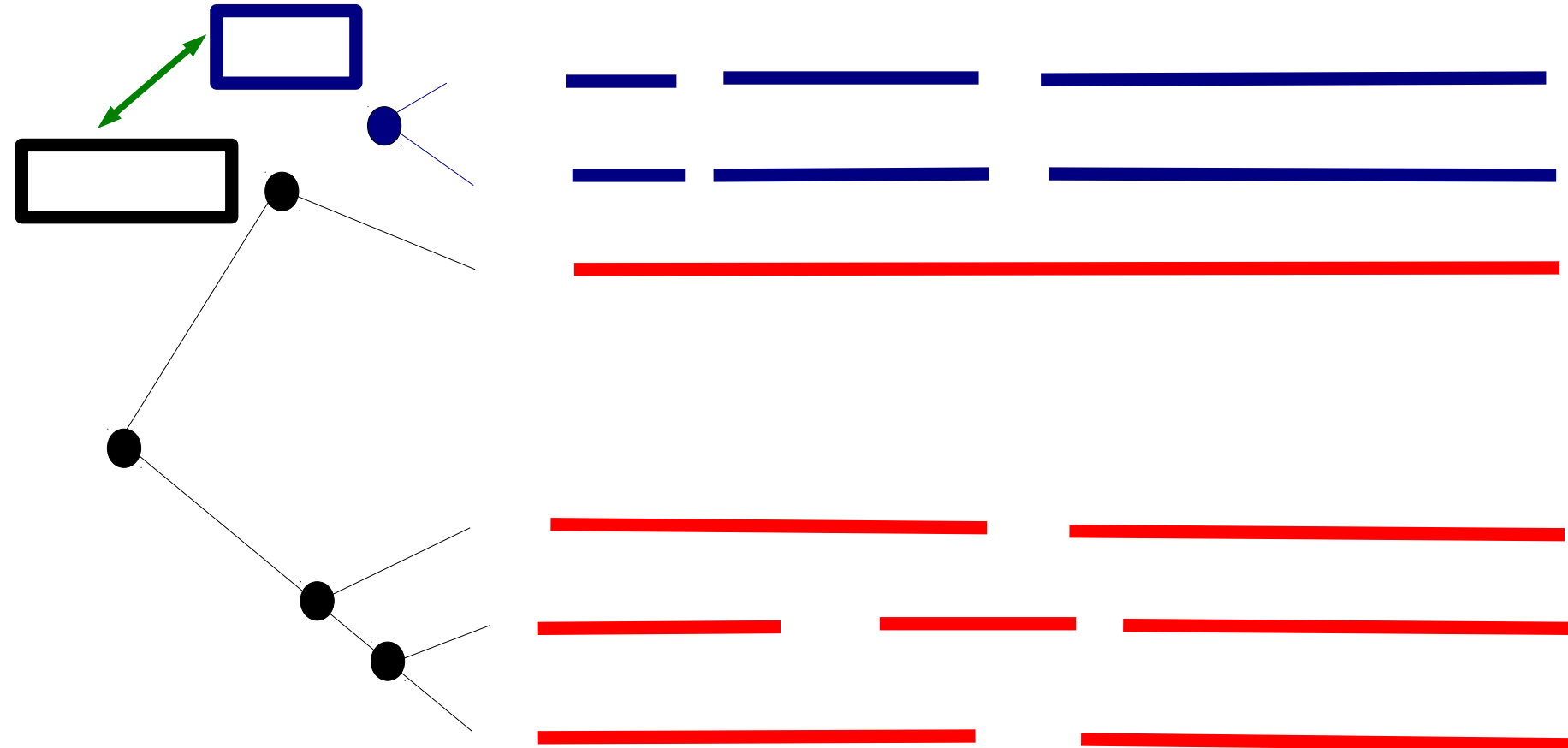
MUSCLE Re-Finement



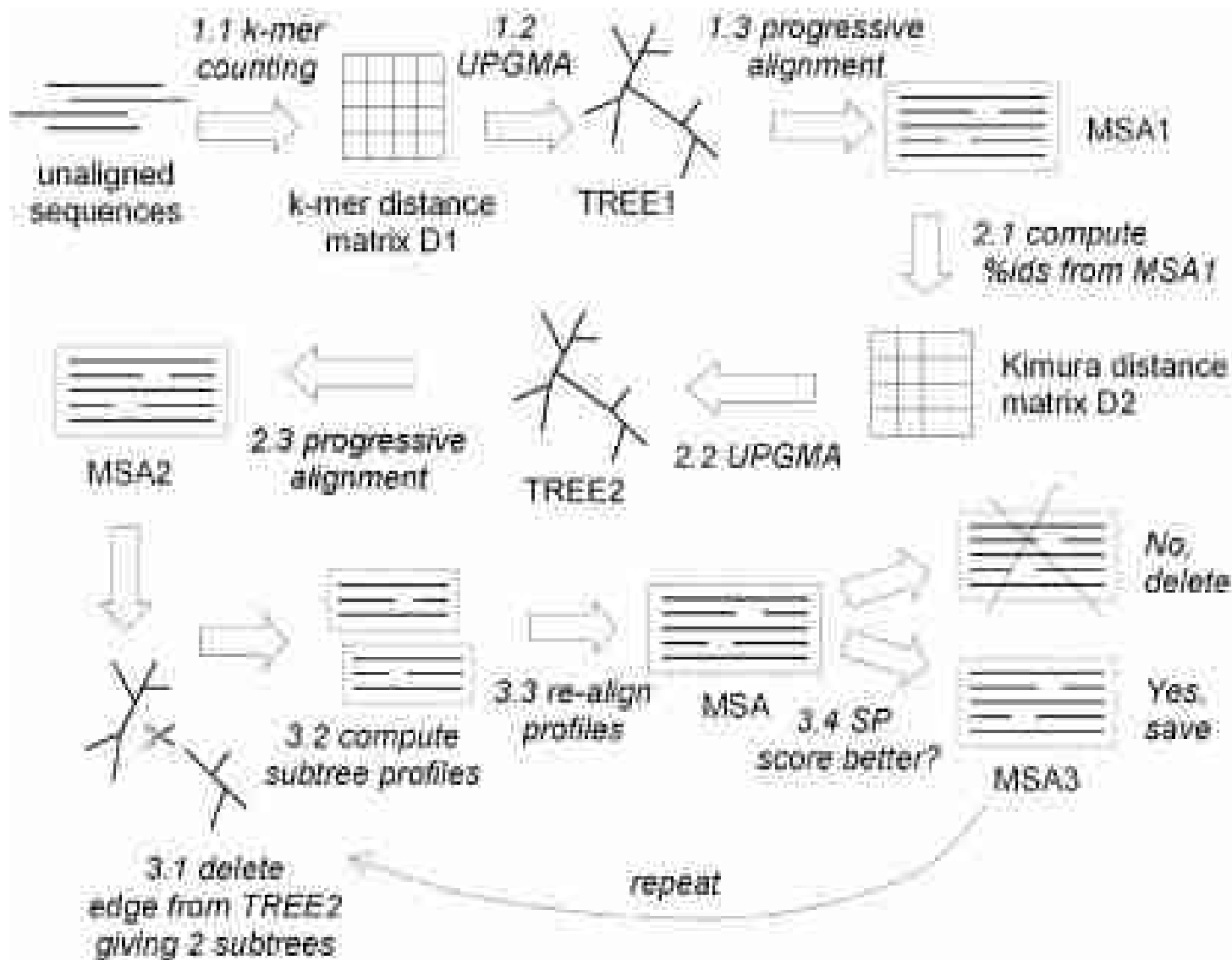
MUSCLE Re-Finement



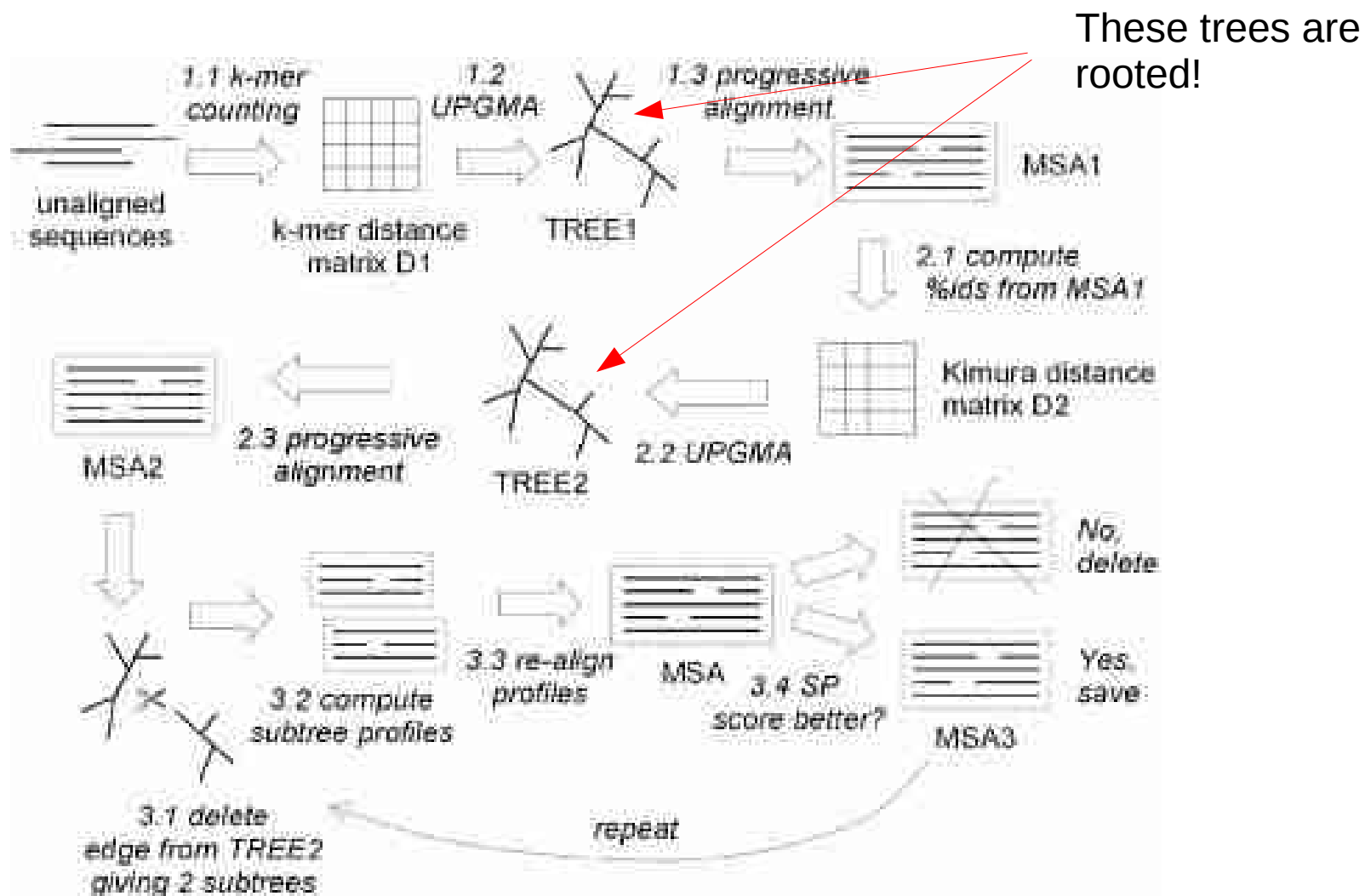
MUSCLE Re-Finement



MUSCLE Details



MUSCLE Details



MUSCLE Refinement

1. An edge/branch is chosen from *TREE2* (edges are visited in order of decreasing distance from the root)
2. *TREE2* is divided into two subtrees by deleting the edge. The profile of the multiple alignment in each subtree is computed.
3. A new multiple alignment is produced by re-aligning the two profiles.
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.

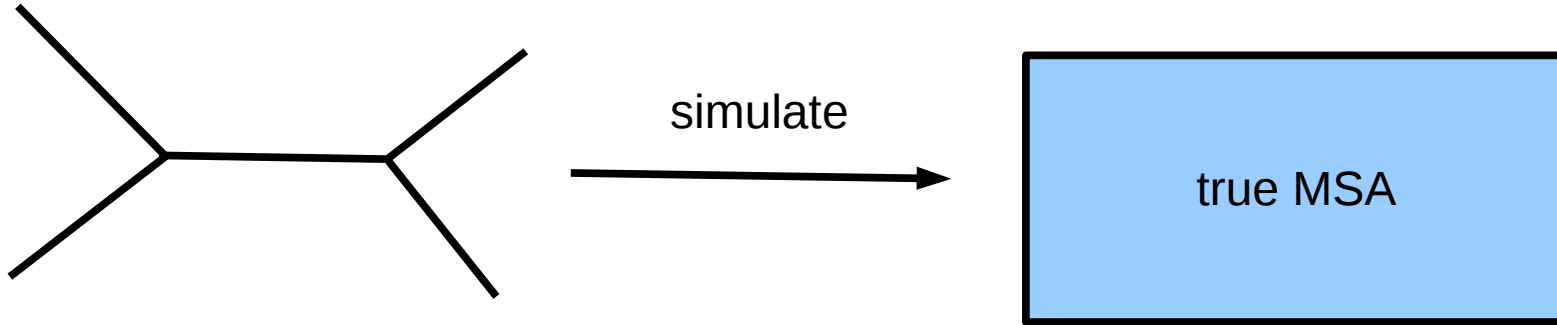
Motif-based approaches

- Find a small motif (substring) common to all sequences
- Called: anchor, block, region, q-gram *etc*
- If motif is found → shift sequences such that the motifs are “in alignment”
- Then, align regions around these motifs using for instance progressive alignment

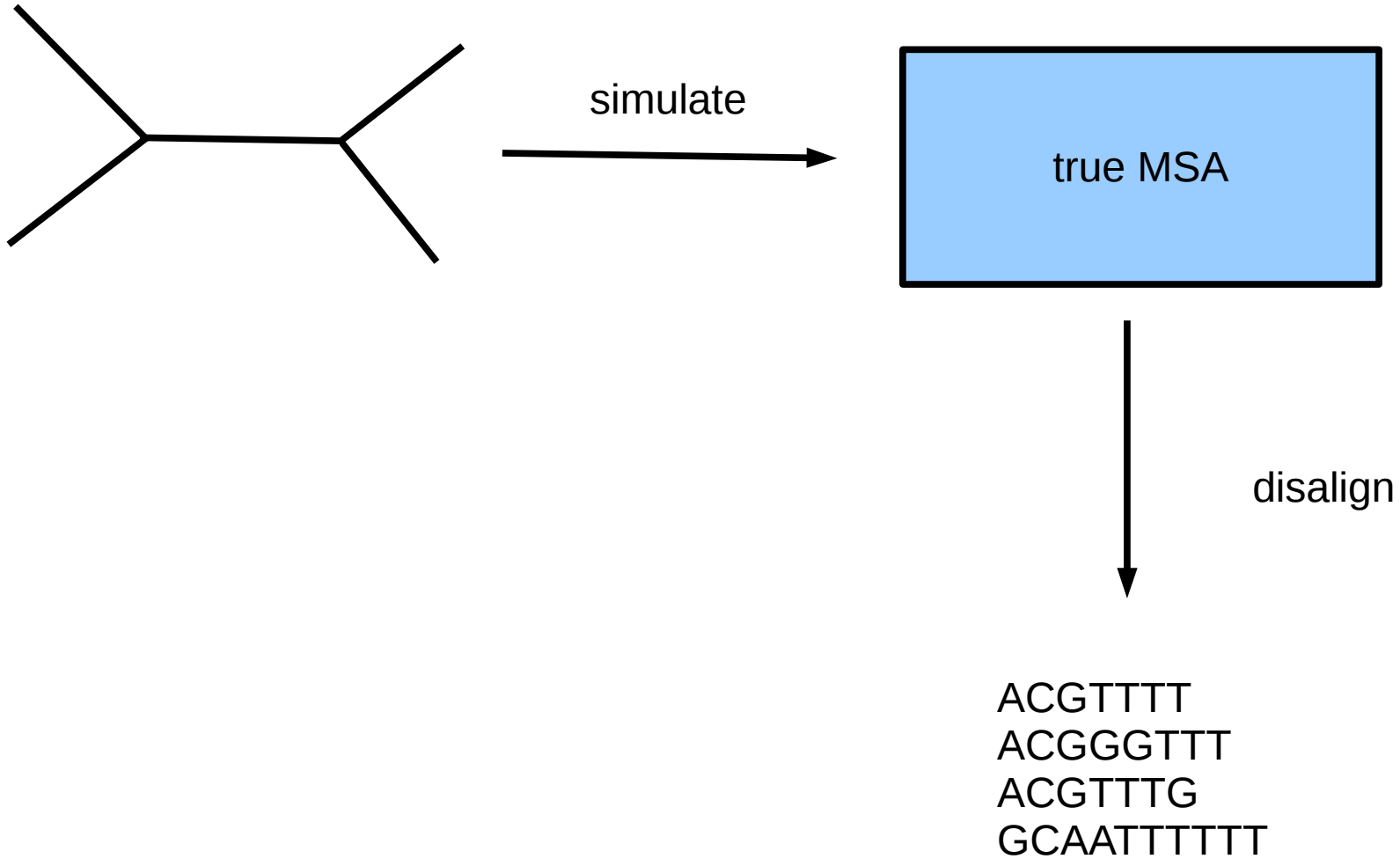
Benchmarking MSAs

- 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

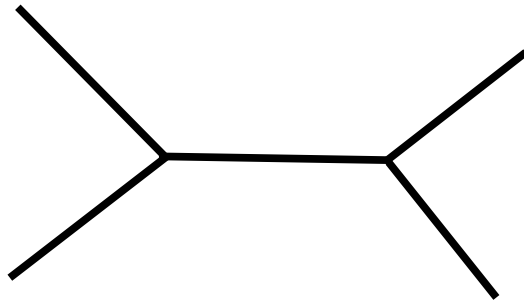
Simulation



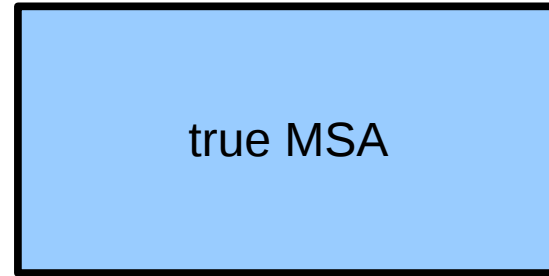
Simulation



Simulation



simulate

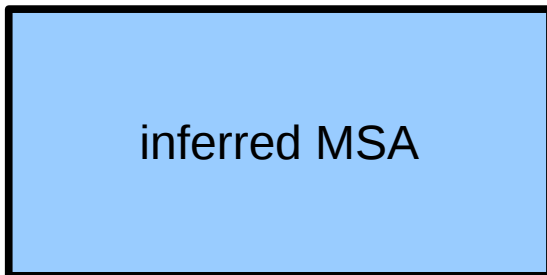


disalign

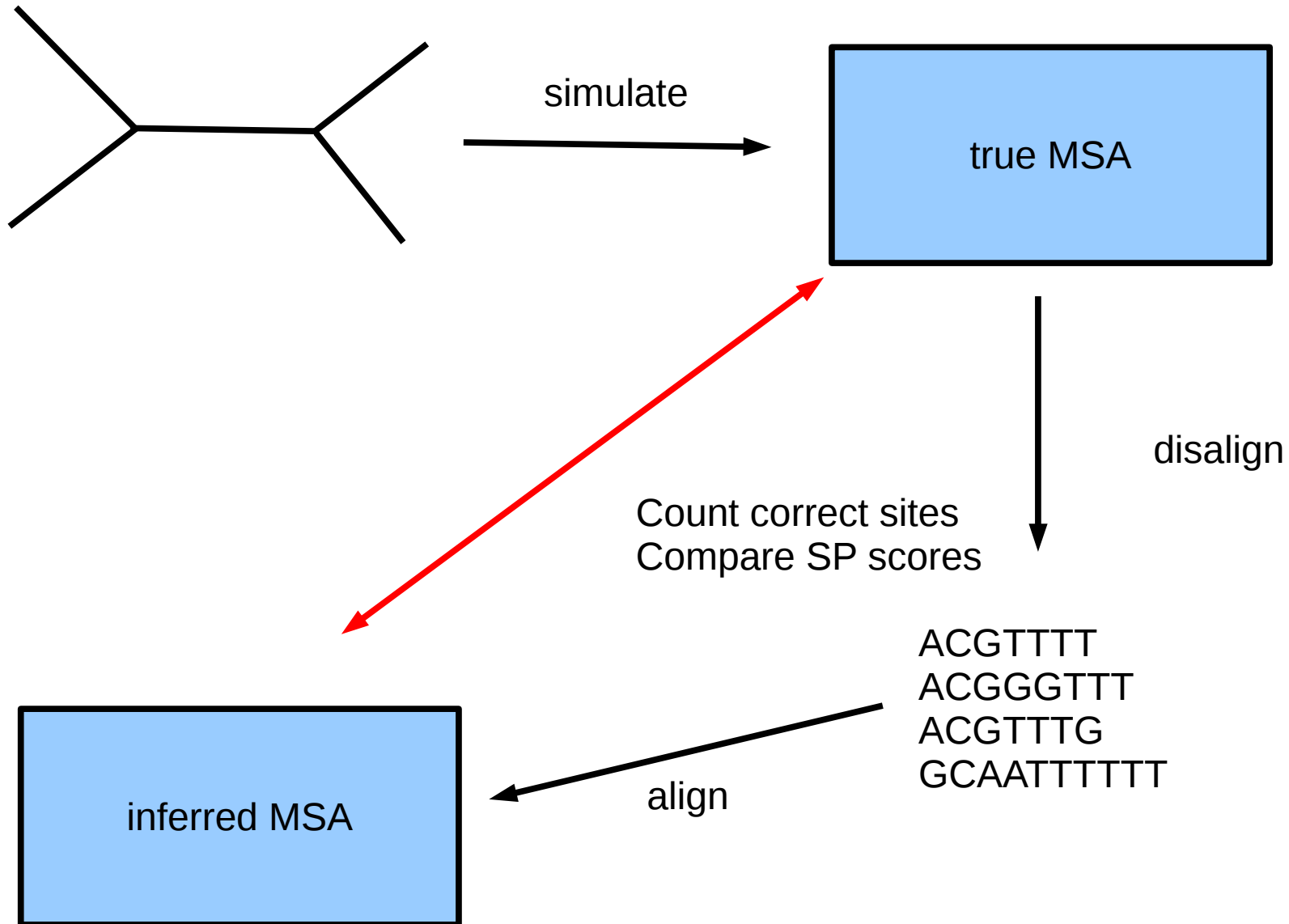


ACGTTTT
ACGGGTTT
ACGTTTG
GCAATTTTTT

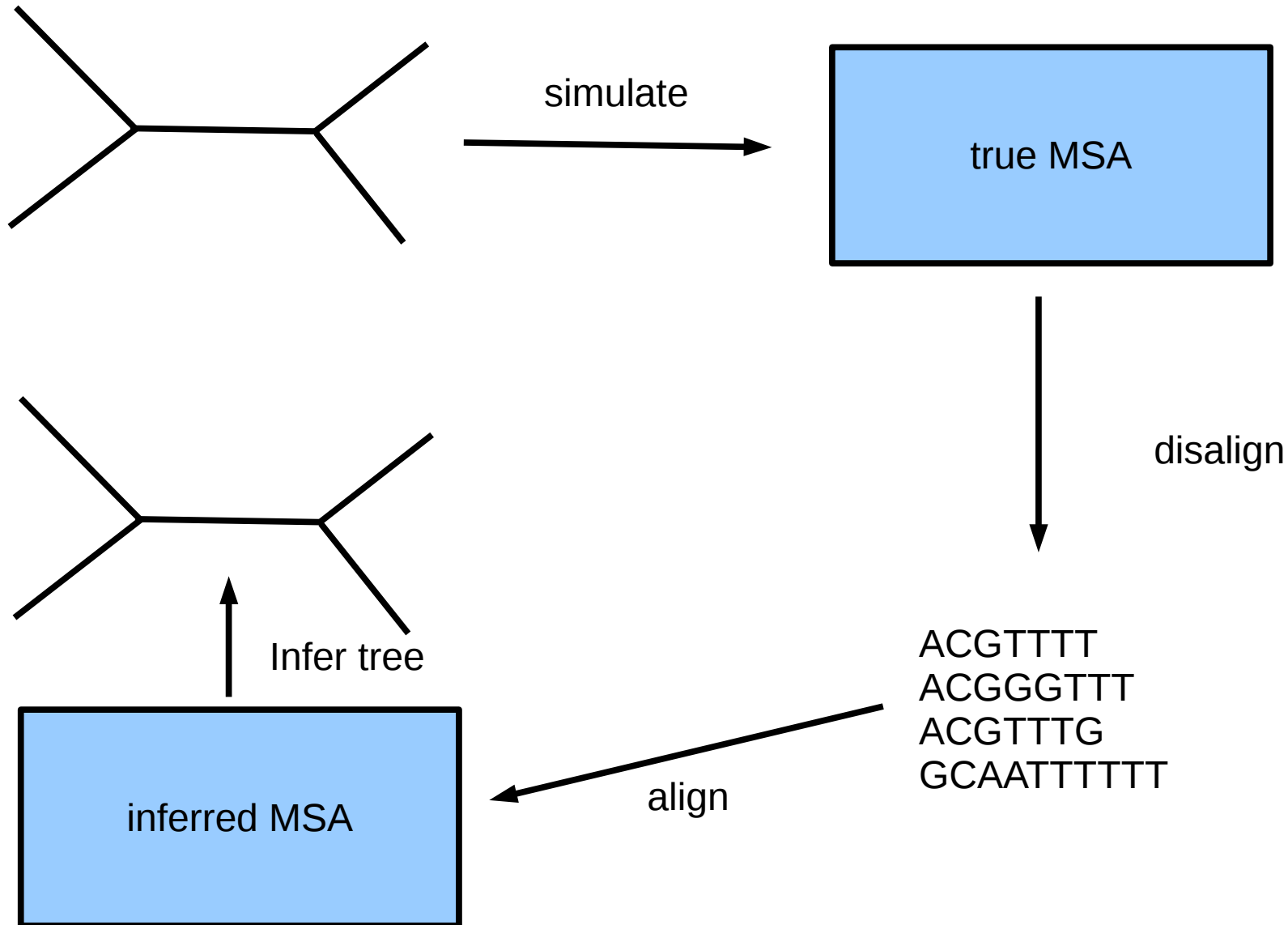
align



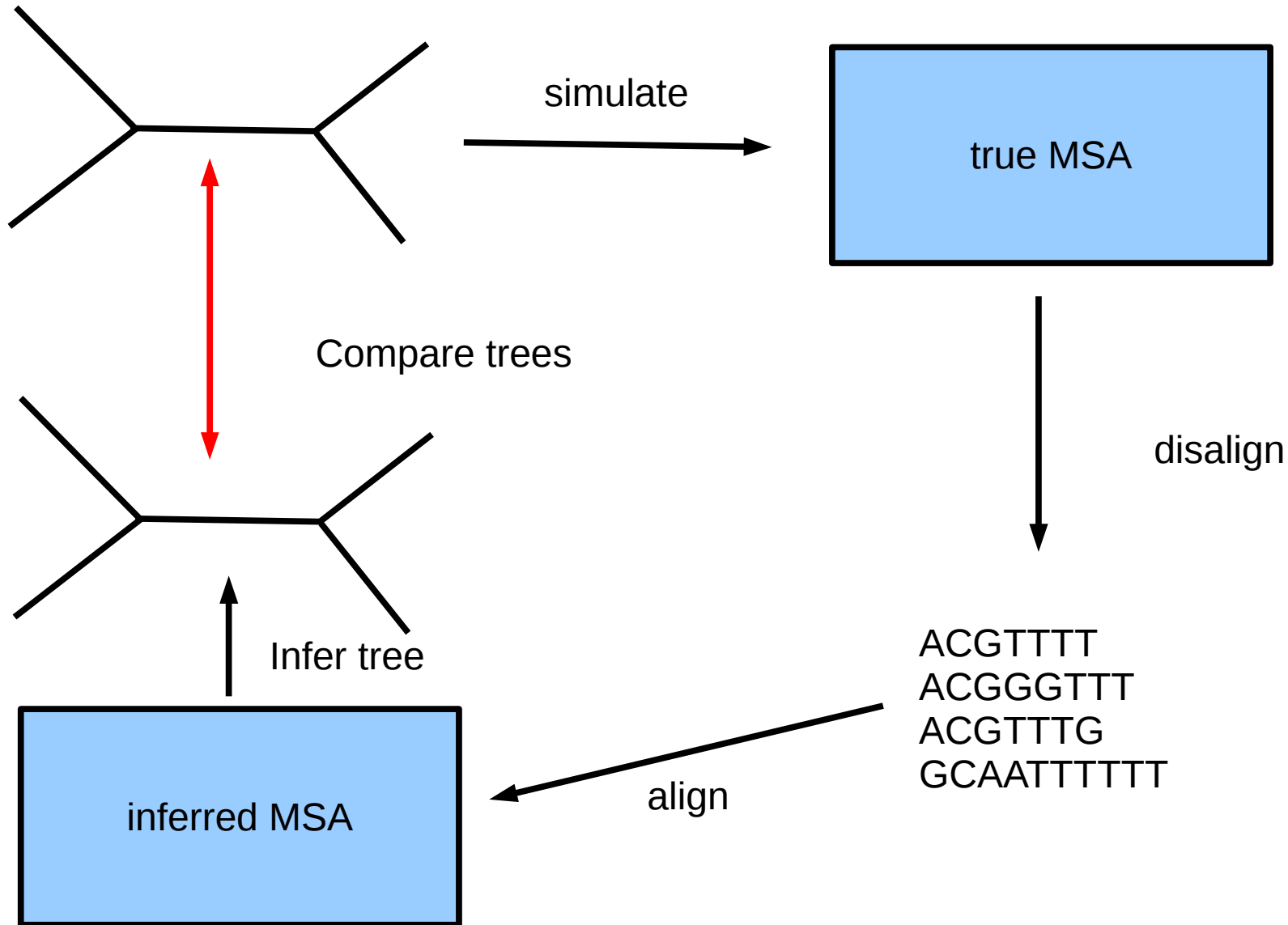
Simulation



Simulation



Simulation



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 heuristics with performance guarantees
- However, practical approaches use ad hoc heuristics that typically perform better
- Classes of algorithms
 - Progressive MSA
 - Progressive iterative MSA
 - Motif-based approaches
 - Statistical MSA (not covered)
 - Phylogeny-aware MSA (not covered)
 - Simultaneous MSA & tree inference (not covered)