

Day 2

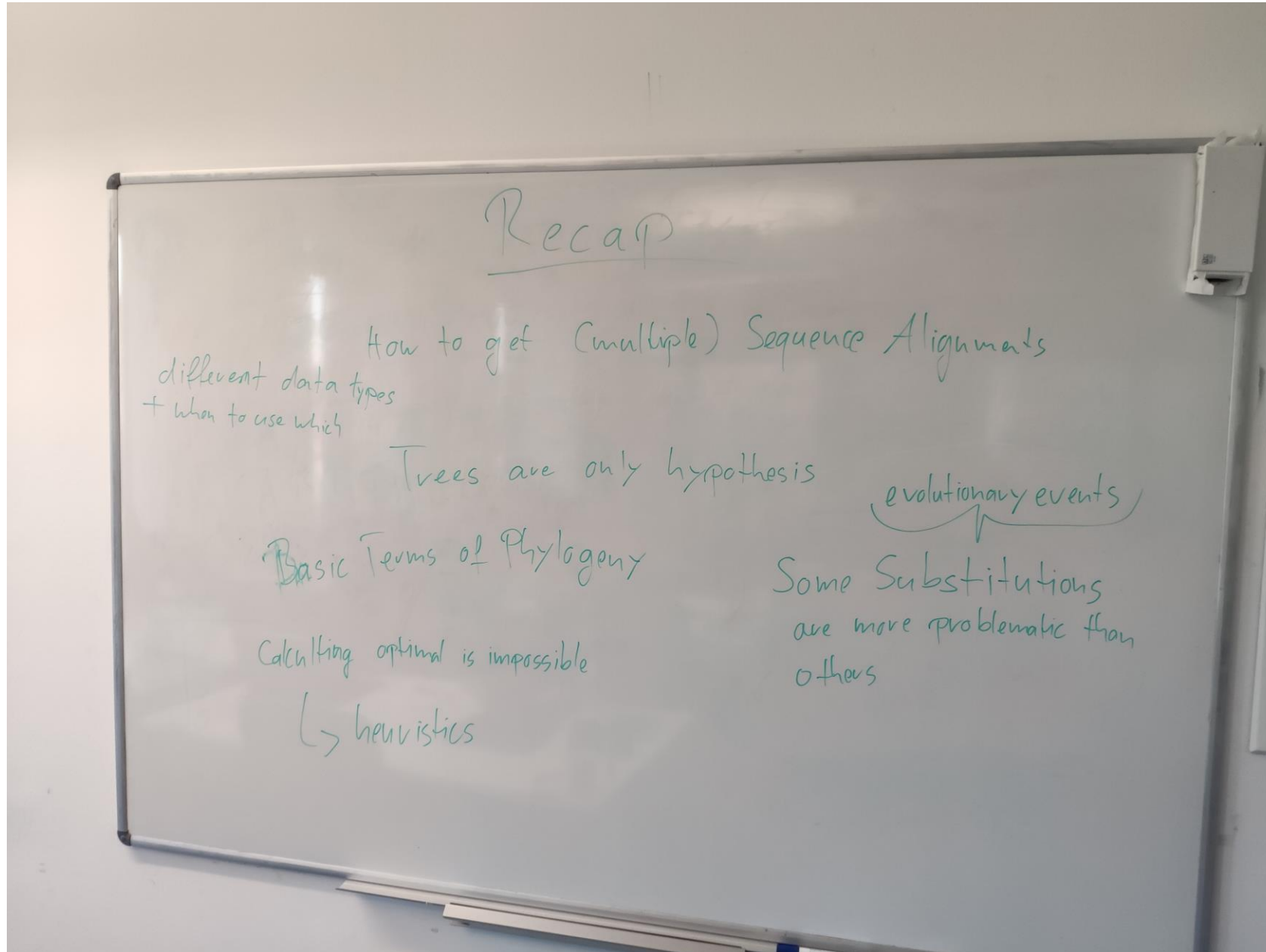
Pathogen phylogenetics: From sequences to trees

Workshop

14.01.2020 – 17.01.2020

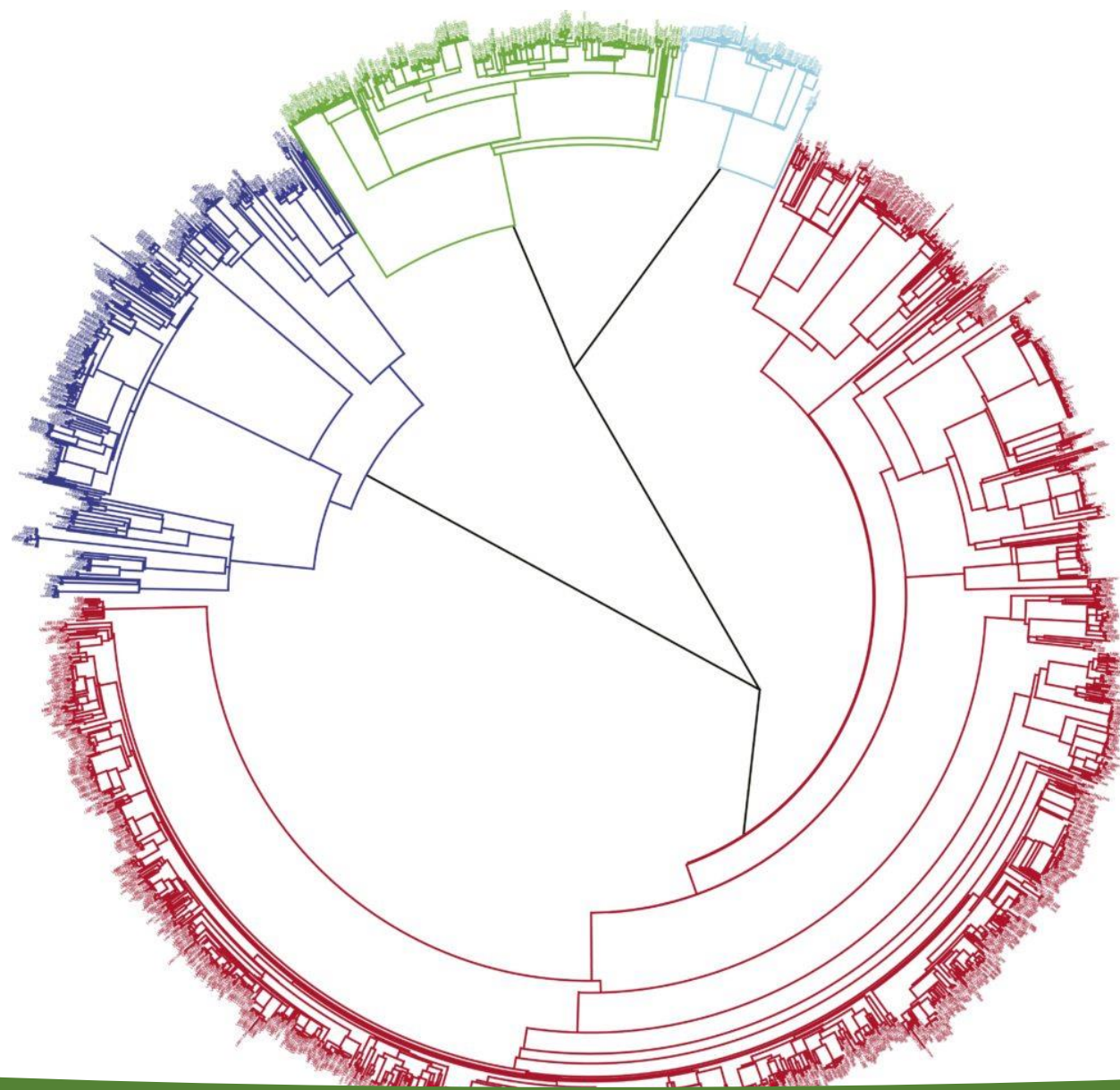


Recap Of Workshop Day 1



Objectives Of The Day

- You know the fundamental differences between Maximum Parsimony, distance based and Maximum Likelihood methods and when to use which approach.
- You have a good idea how sequence evolution can be modelled mathematically using distance models.
- You know that the better your sequence alignment, than the better your distance models and than the better your phylogenetic trees.



Algorithmic Phylogenetics

- Building trees from alignments. -

Algorithmic Phylogenetics

- ~ is the application of computational algorithms to phylogenetic analyses.
- Four main approaches:
 - Parsimony analysis
 - Distance matrix approaches
 - Maximum Likelihood analysis
 - Bayesian inference
- They are all (mainly) based on sequence alignments.



“All things being equal, the simplest solution tends to be the best one.”

William of Ockham

Parsimony

- The dream of a perfect evolution. -

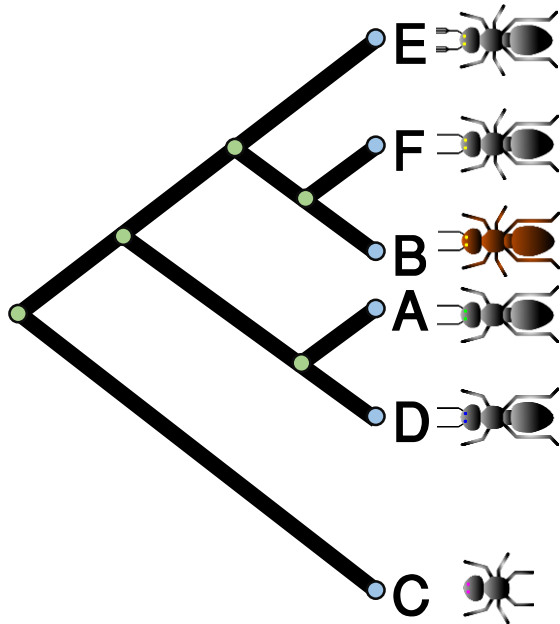
Algorithmic Phylogenetics: The Parsimony Principle

- If in a given phylogenetic tree every observed characteristic evolved exactly once, then this tree follows the **perfect phylogeny** principle.
- We know this does not hold true for the tree of life (e.g. convergent evolution).
→ in particular violated at molecular level
- If in a given phylogenetic tree every observed characteristic evolved as few times as possible, then this tree follows the **parsimony** principle.

Algorithmic Phylogenetics: Maximum Parsimony

- **Maximum parsimony:** Identify the phylogenetic tree that minimizes the total amount of character changes of the given taxa.
- Historically used for morphological data but is also applicable on molecular data.

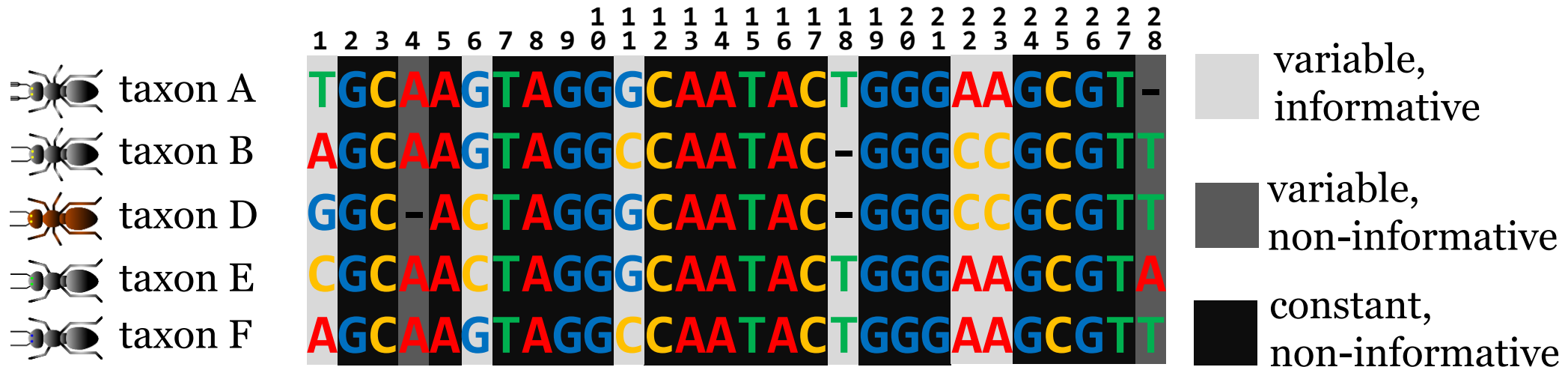
feature matrix



Species	No. of legs	No. of antennae	Color	Eyes	Segments
A	4	2	Black	Green	3
B	6	2	Brown	Yellow	3
C	6	0	Black	Purple	1
D	4	2	Black	Blue	3
E	6	6	Black	Yellow	3
F	6	2	Black	Yellow	3

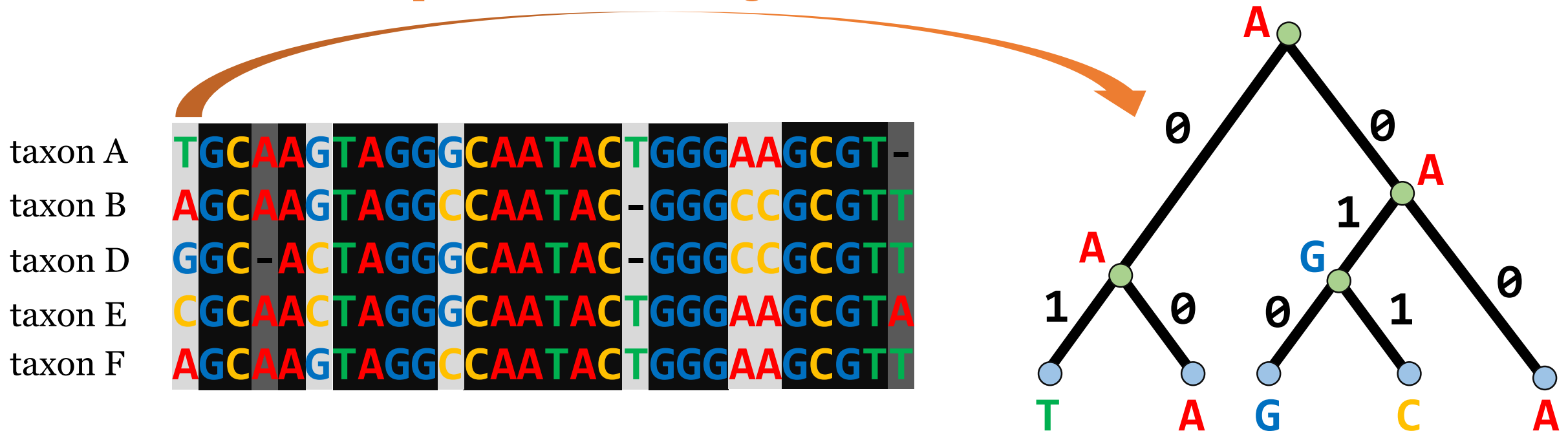
Algorithmic Phylogenetics: Maximum Parsimony

- Within a sequence alignment every column (site) corresponds to one discrete characteristic (feature).
- Not every site holds informative character changes.



Algorithmic Phylogenetics: Small Parsimony

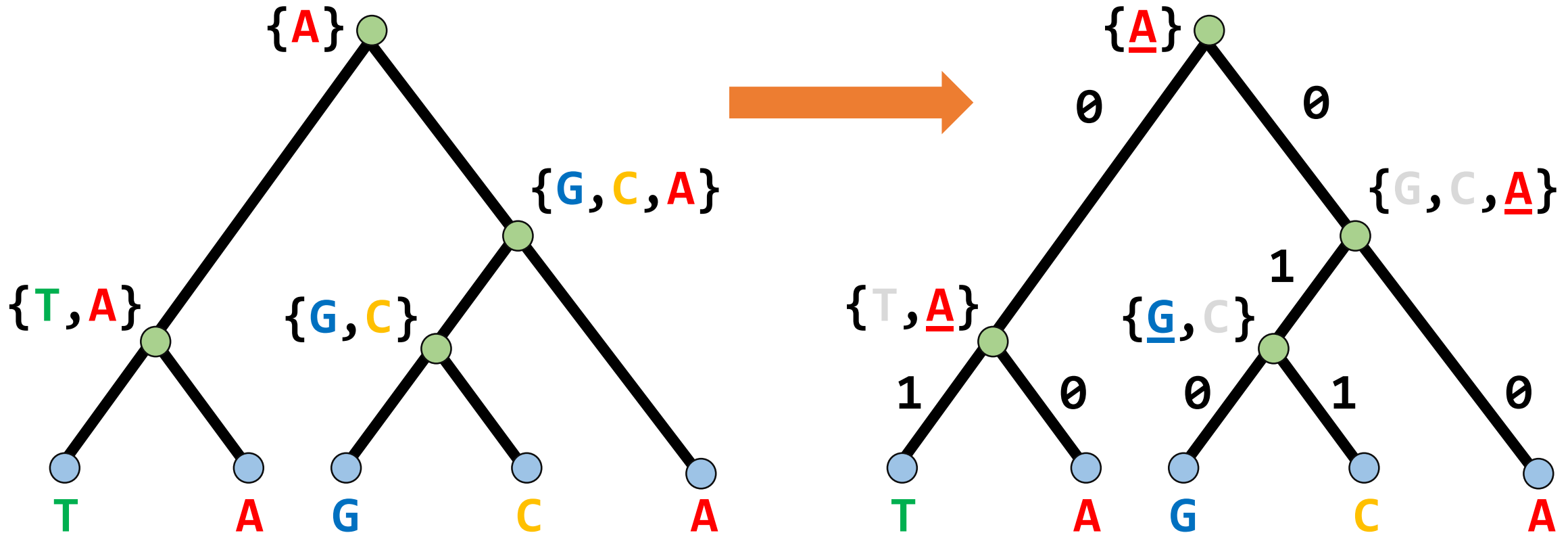
- Before solving the Maximum parsimony problem, we will first consider the **small parsimony** problem.
- Given an informative site of a sequence alignment and a phylogenetic tree, what is the **most-parsimonious assignment** of the inner nodes of this tree?



- The most-parsimonious assignment gives us the **optimal score** for this particular tree and site (in our example the optimal score would be 3).

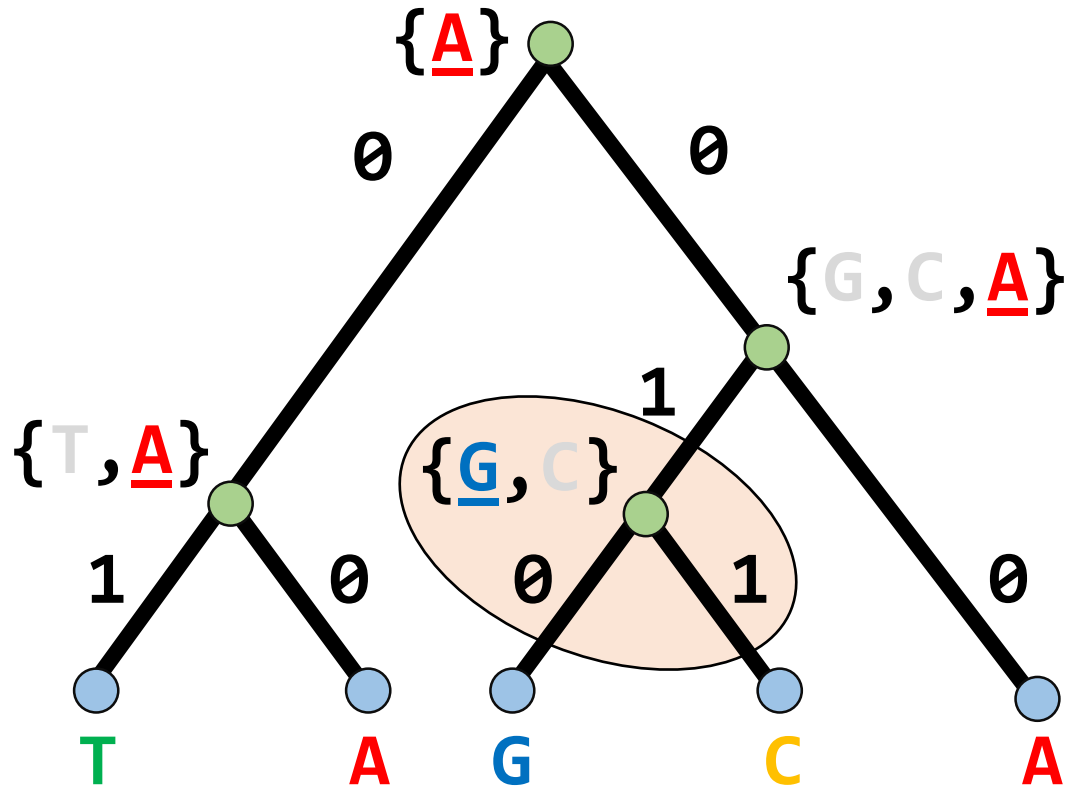
Algorithmic Phylogenetics: Small Parsimony

- The **Fitch algorithm** can efficiently solve the small parsimony problem:
 1. Determine the character sets for every node, starting at the leaves and progressing to the root.
 2. Starting at the root, go through the tree and select for each inner node the character from its character set that corresponds to that of its parent node. If this is not possible, choose any other character from its set.

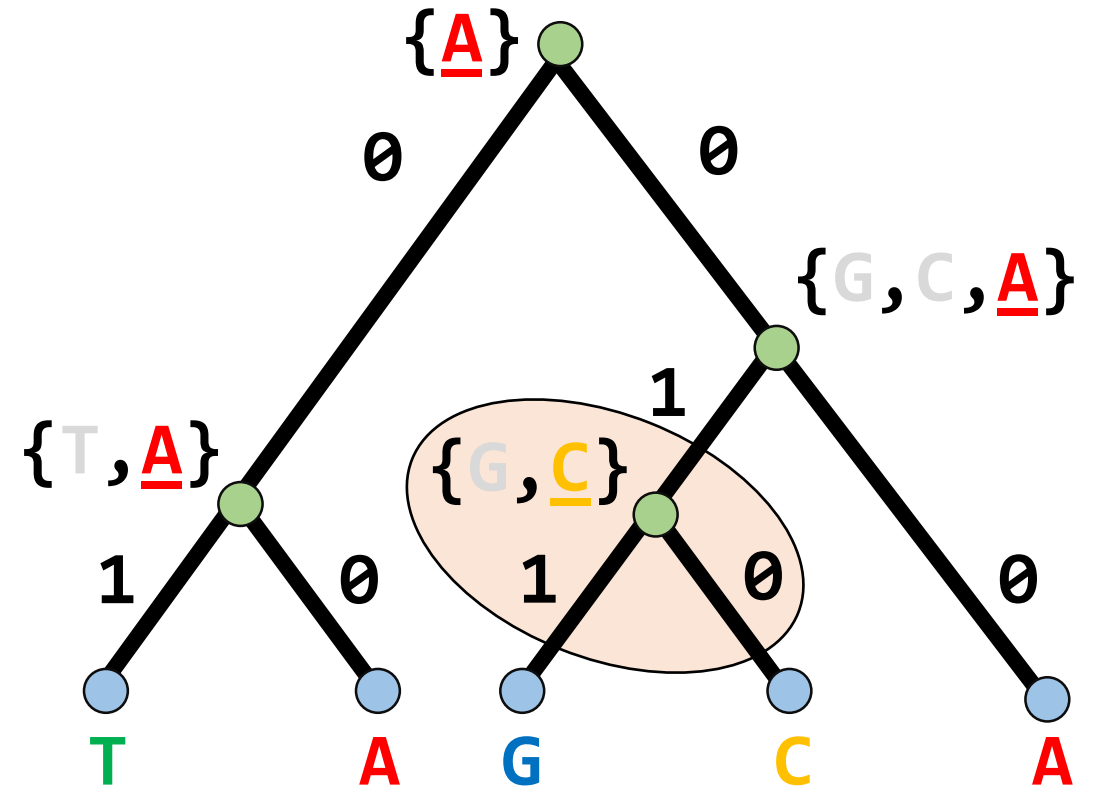


Algorithmic Phylogenetics: Small Parsimony

- The **small parsimony score** is calculated by summing up all character changes of the tree. Multiple optimal solutions may be possible.



score = 3

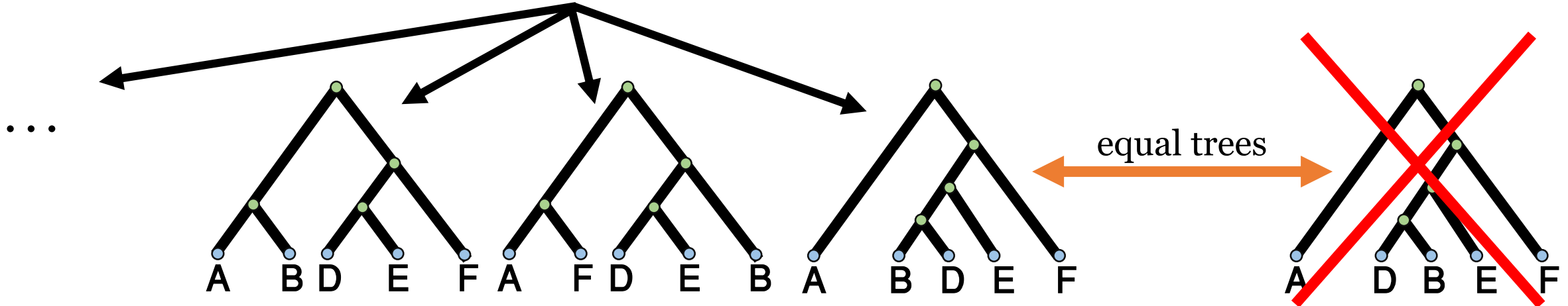


score = 3

Algorithmic Phylogenetics: Maximum Parsimony

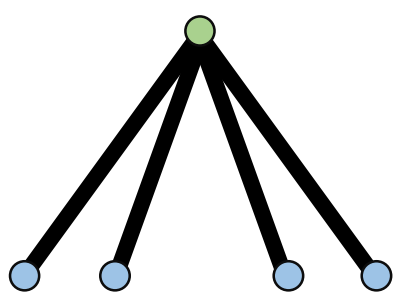
- Since we do not know the true phylogenetic tree for our taxa, we have to find the most-parsimonious one → **Maximum parsimony problem**.
- Given an informative site of a sequence alignment:
 1. Calculate the small parsimony score for every possible tree.
 2. Choose the tree with the lowest score.

taxon A	T	G	C	A	A	G	T	A	G	G	C	A	A	T	A	C	T	G	G	G	A	A	G	C	G	T
taxon B	A	G	C	A	A	G	T	A	G	G	C	A	A	T	A	C	-	G	G	G	C	C	G	C	G	T
taxon D	G	G	C	-	A	C	T	A	G	G	G	C	A	A	T	A	C	-	G	G	G	C	C	G	C	G
taxon E	C	G	C	A	A	C	T	A	G	G	G	C	A	A	T	A	C	T	G	G	G	A	A	G	C	G
taxon F	A	G	C	A	A	G	T	A	G	G	C	A	A	T	A	C	T	G	G	G	A	A	G	C	G	T

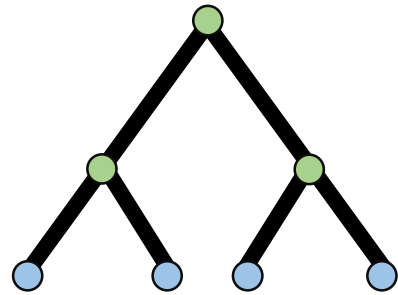


Algorithmic Phylogenetics: Maximum Parsimony

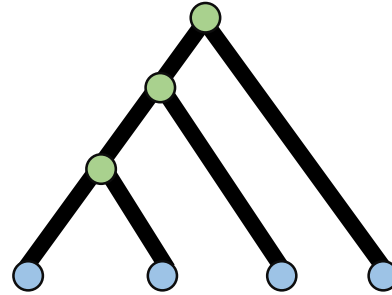
- How many possible (including non-binary) rooted trees with 4 taxa exist?
→ 26 different trees



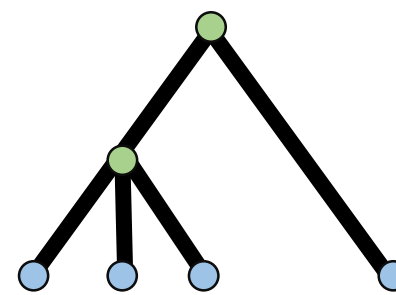
1 tree



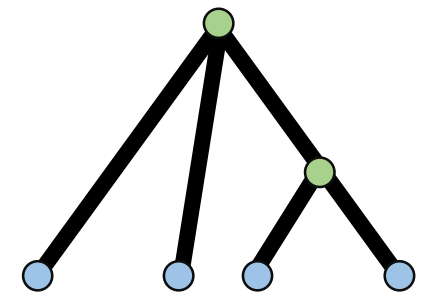
3 trees



12 trees



4 trees



6 trees


- How many possible (including non-binary) rooted trees with 5 taxa exist?
→ 166 different trees
- How many possible (**excluding non-binary**) rooted trees with n taxa exist?

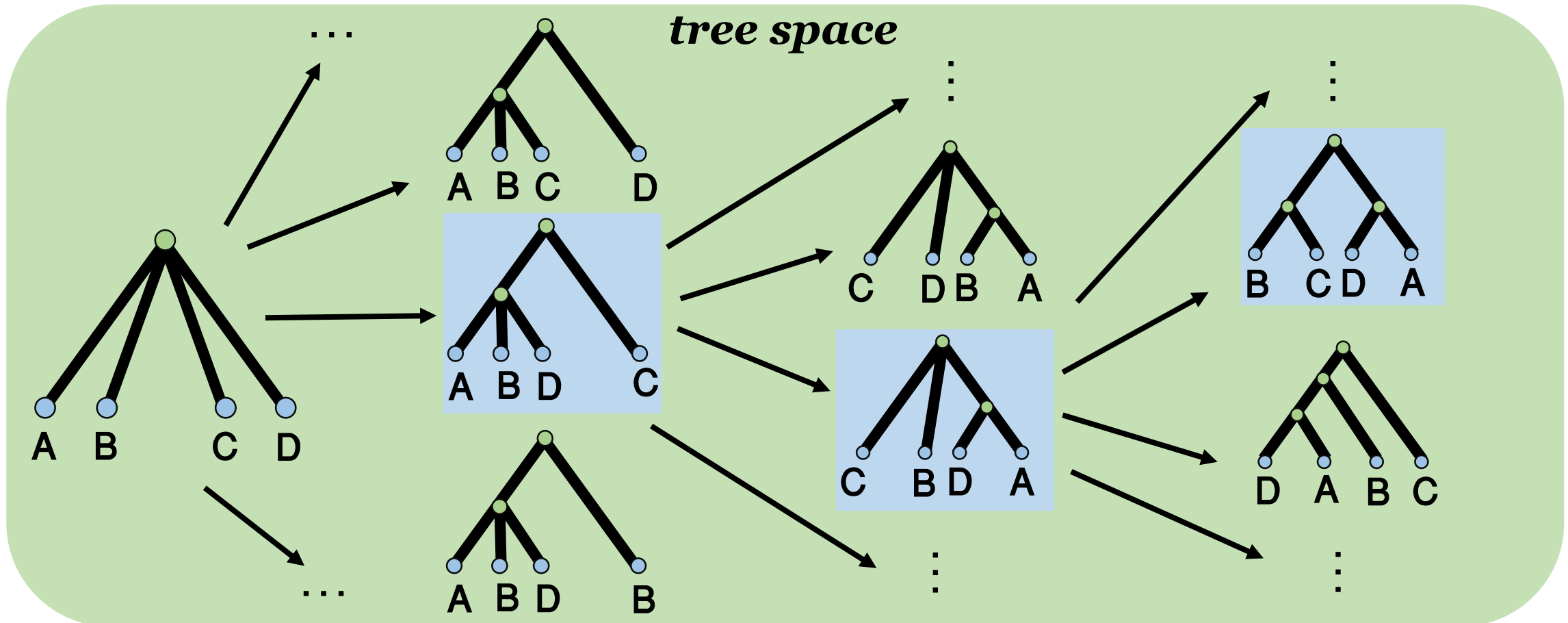
$$\rightarrow \prod_{j=3}^n (2j - 5)$$

$n = 10 \rightarrow$ more than 34.000.000 trees

$n = 30 \rightarrow$ more than 5×10^{38} trees

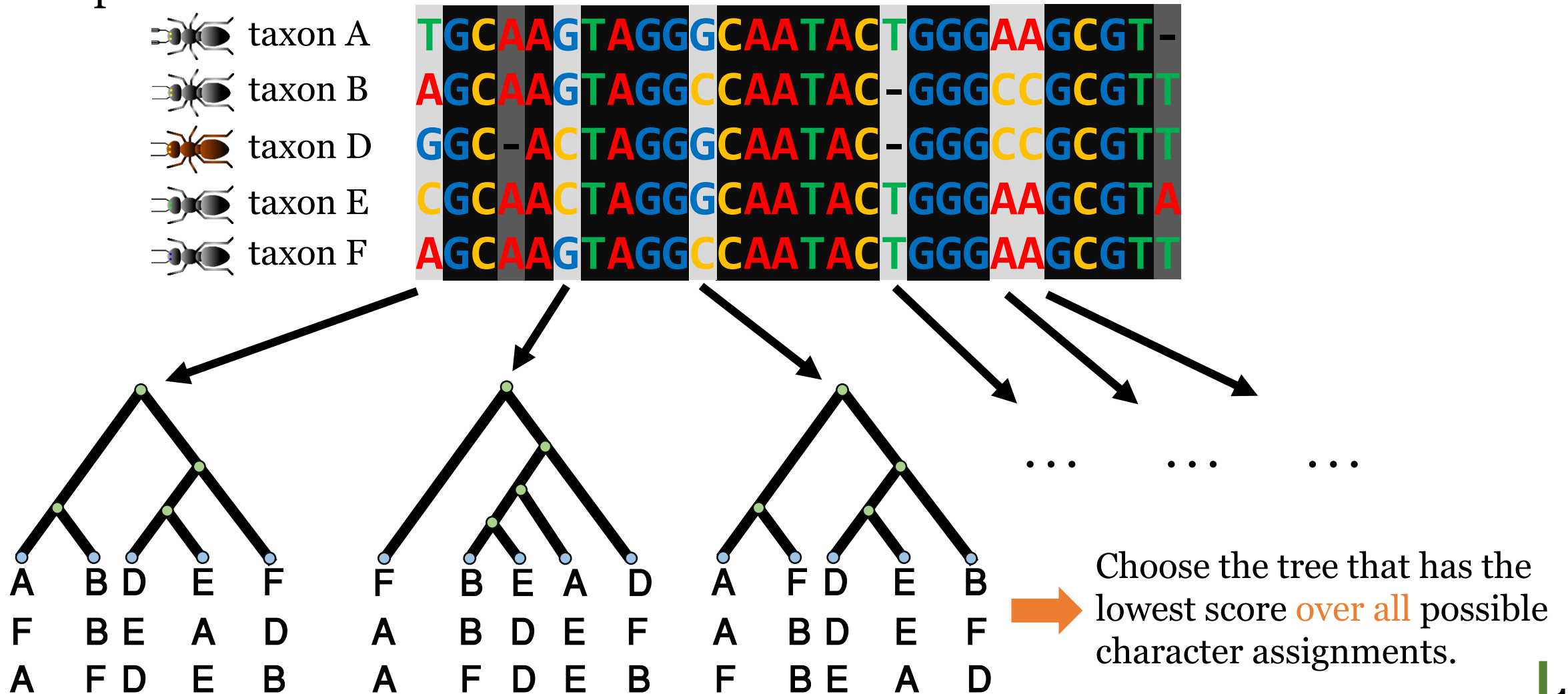
Algorithmic Phylogenetics: Maximum Parsimony

- The Maximum Parsimony problem is NP-hard. 
- We use heuristics to (hopefully) find the most-parsimonious tree for a given informative site (e.g. using *star-decomposition*) in the *tree space*.



Algorithmic Phylogenetics: Maximum Parsimony

- How to handle conflicting trees when multiple informative sites are present?



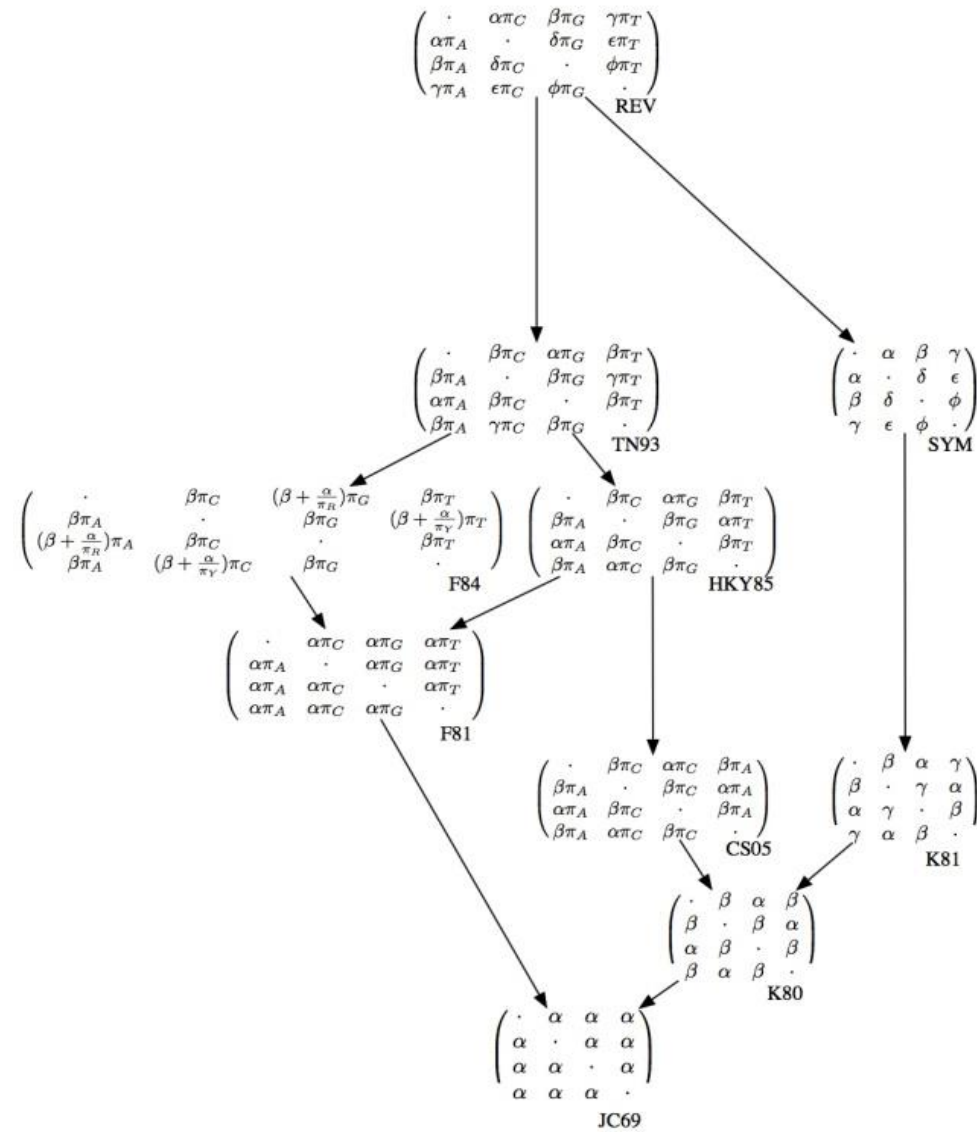
Algorithmic Phylogenetics: (Dis)advantages Of Parsimony Analysis

Advantages

- simple and intuitive concept
- applicable to morphological and molecular data
- relatively fast

Disadvantages

- evolution is not parsimonious
- no explicit model of sequence evolution used
- often multiple optimal trees exist for a given data set

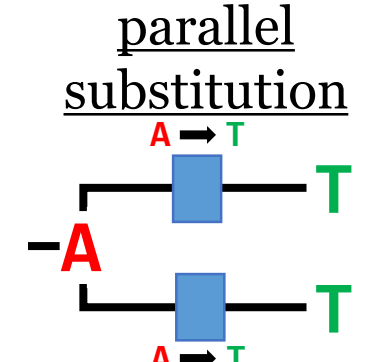
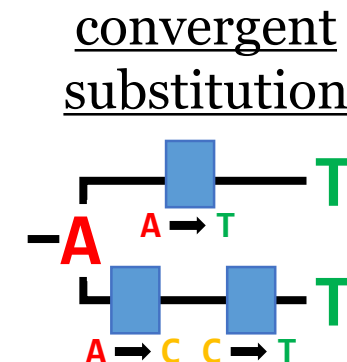
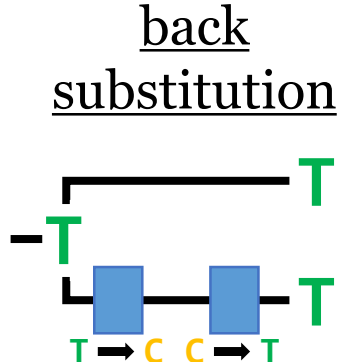
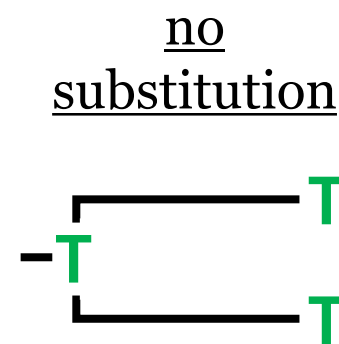
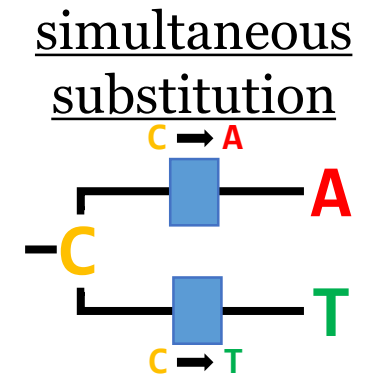
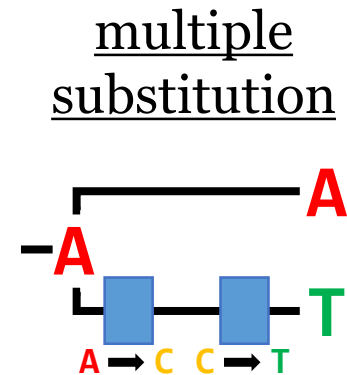
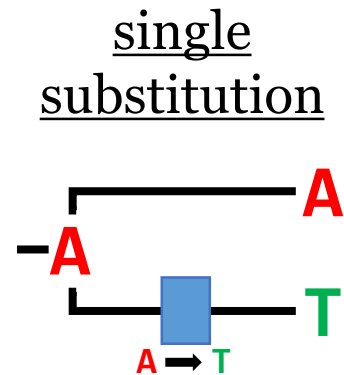
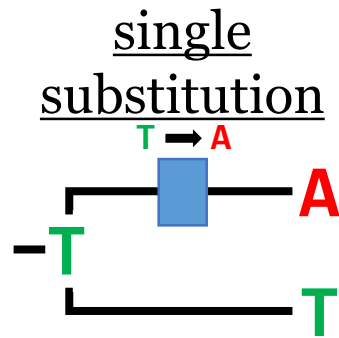


Sequence evolution

- Phylogenetics next top model. -

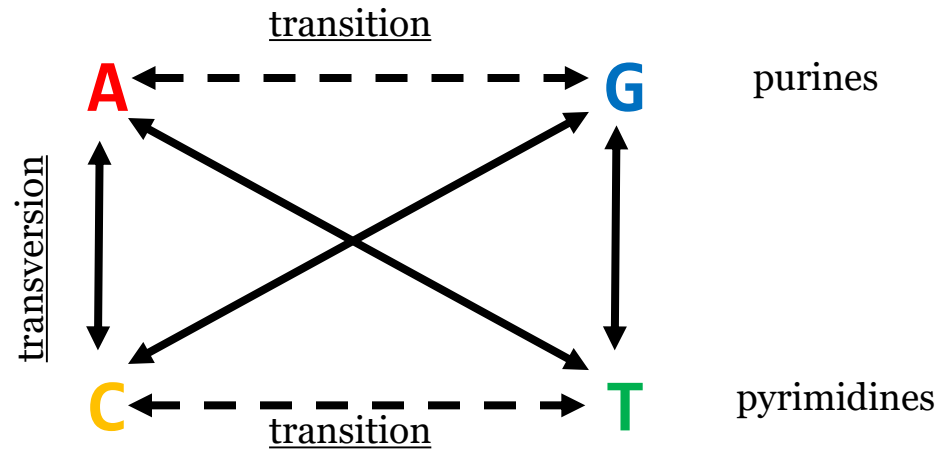
Sequence Evolution: Basics

- In theory, DNA sequences are bad evolutionary characteristics, because there are only four possible character states.
- Since we (mostly) do only know the genomic sequences of current living organisms, we do not know how many substitutions really occurred during the evolution of orthologous sequences.



Sequence Evolution: Basics

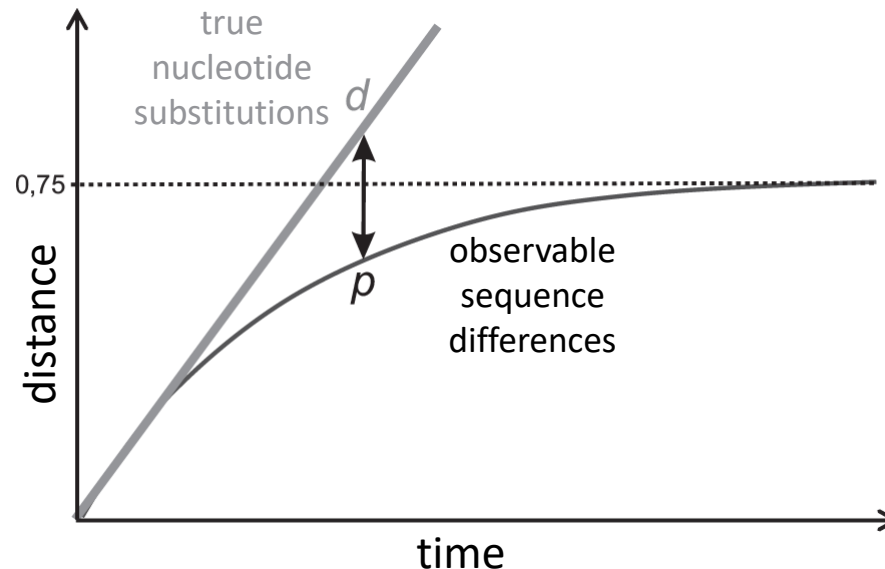
- We also know some substitutions occur more frequently than others.



- Additionally, some genomic positions have higher substitution rates than others. (silent mutations, compensatory mutations, ...)
- We need *distance models* to accurately estimate the “amount” of evolution that happened between two sequences.
- Those sequence evolution models are used for Distance-matrix methods, Maximum Likelihood and Bayesian Inference approaches.

Sequence Evolution: The Uncorrected p-Distance

- Given the pairwise sequence alignment of length 1000 with 20 mismatches and without gaps.
→ The two sequences have a relative distance of $20/1000 = 0.02 = 2\%$
- This distance is called the *uncorrected p-distance* of observable changes.
- To correct the p-distance we can use different distance models to estimate the true amount of substitutions that occurred between both sequences.



Sequence Evolution: Substitution Probabilities

- The **probability** of nucleotide i being substituted to nucleotide j over time is denoted as P_{ij} .
- The set of all possible substitutions $\{P_{ij}\}$ can be displayed as a **substitution-probability-matrix**, denoted as P .

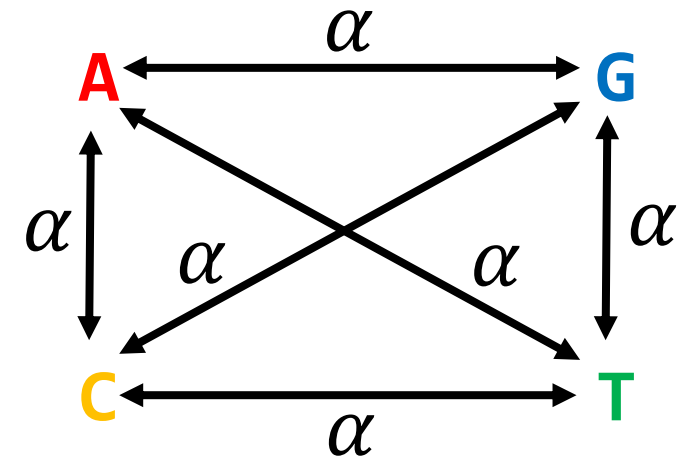
$$P = \begin{matrix} & \begin{matrix} \text{A} & \text{C} & \text{G} & \text{T} \end{matrix} \\ \begin{pmatrix} \cdot & P_{AC} & P_{AG} & P_{AT} \\ P_{CA} & \cdot & P_{CG} & P_{CT} \\ P_{GA} & P_{GC} & \cdot & P_{GT} \\ P_{TA} & P_{TC} & P_{TG} & \cdot \end{pmatrix} & \begin{matrix} \text{A} \\ \text{C} \\ \text{G} \\ \text{T} \end{matrix} \end{matrix}$$

- The probability of no substitution is $P_{ii} = 1 - \sum_{j \neq i} P_{ij}$

Sequence Evolution: The Jukes-Cantor Model

- Assumptions:
 - The substitution probability of nucleotide i to nucleotide j is always equal and denoted as α .
 - The relative amount of nucleotides is always equal.

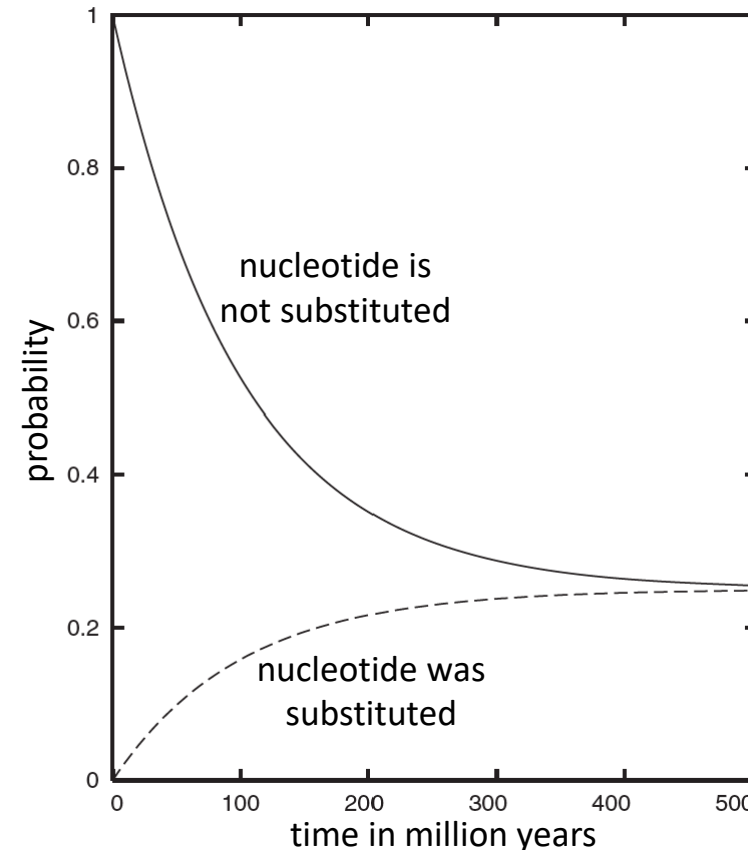
$$P = \begin{pmatrix} . & \alpha & \alpha & \alpha \\ \alpha & . & \alpha & \alpha \\ \alpha & \alpha & . & \alpha \\ \alpha & \alpha & \alpha & . \end{pmatrix} = \begin{pmatrix} 1 - 3\alpha & \alpha & \alpha & \alpha \\ \alpha & 1 - 3\alpha & \alpha & \alpha \\ \alpha & \alpha & 1 - 3\alpha & \alpha \\ \alpha & \alpha & \alpha & 1 - 3\alpha \end{pmatrix}$$



Sequence Evolution: The Jukes-Cantor Model

- But how do we get from our substitution matrix P to the actual probability that there was a substitution at a specific site after some time t ?
- We need the help of **stochastic processes**, which will give us the function $P_{ij}(t)$ that describes the probability of nucleotide i being substituted to j after some time t .

$$P_{ij}(t) = \begin{cases} \frac{1}{4} - \frac{1}{4}e^{-4\alpha t}, & \text{if } i \neq j \\ \frac{1}{4} + \frac{3}{4}e^{-4\alpha t}, & \text{if } i = j \end{cases}$$



if we assume
that $t = 10^{-8}$
substitutions per
position per year

Sequence Evolution: The Jukes Cantor Model

- Doing some more mathematics, we receive the distance correction formula:

$$K = -\frac{3}{4} \ln\left(1 - \frac{4}{3}p\right)$$

← This is our uncorrected
p-distance

- Example:

TTTGC-AGTGGGGCAATACTCGGAAGCGTG
CCGGCAACTAGGGCAATAC-GGGCCGCGTT

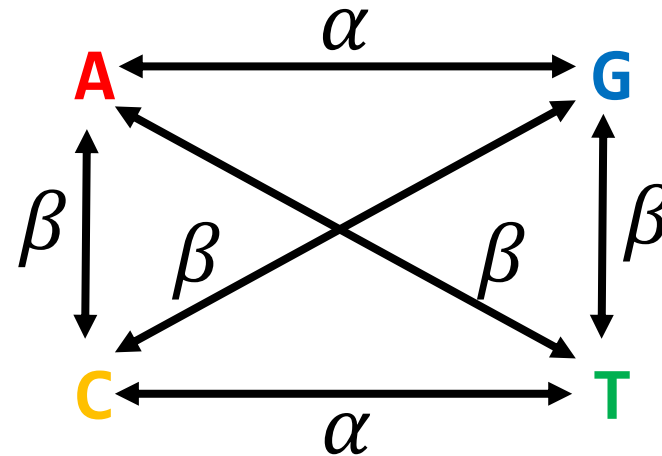
$$p = 7/29 = 0.2413$$

$$K = 0,1265$$

Sequence Evolution: The Kimura-Two-Parameter-Model

- Assumptions:
 - The substitution probability of nucleotide i to nucleotide j are α for transitions and β for transversions.
 - The relative amount of nucleotides is always equal.

$$P = \begin{pmatrix} . & \beta & \alpha & \beta \\ \beta & . & \beta & \alpha \\ \alpha & \beta & . & \beta \\ \beta & \alpha & \beta & . \end{pmatrix}$$

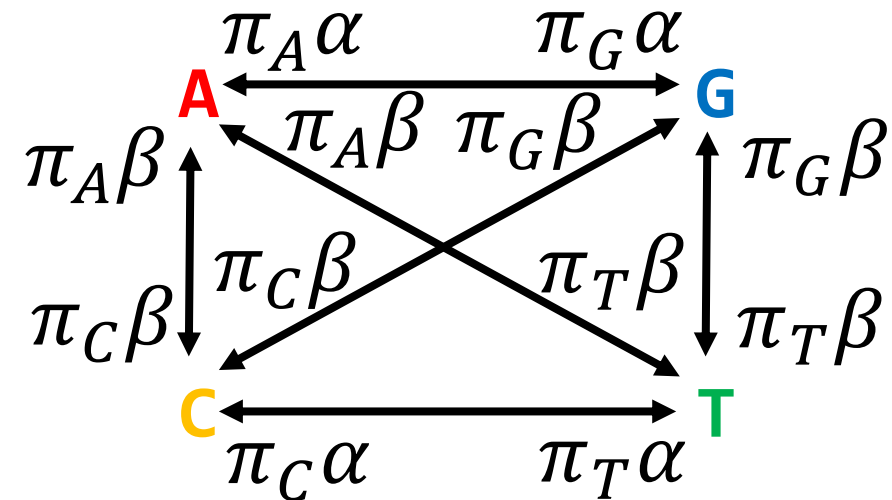


- Distance correction: $K = \frac{1}{2} \ln \frac{1}{(1-2p-q)} + \frac{1}{4} \ln \frac{1}{(1-2q)}$

Sequence Evolution: The HKY85-Model

- Assumptions:
 - The substitution probability of nucleotide i to nucleotide j are α for transitions and β for transversions.
 - The relative amounts of nucleotides can vary with $\pi_A, \pi_C, \pi_G, \pi_T$ being the relative amounts of **A**, **C**, **G** and **T**, respectively.

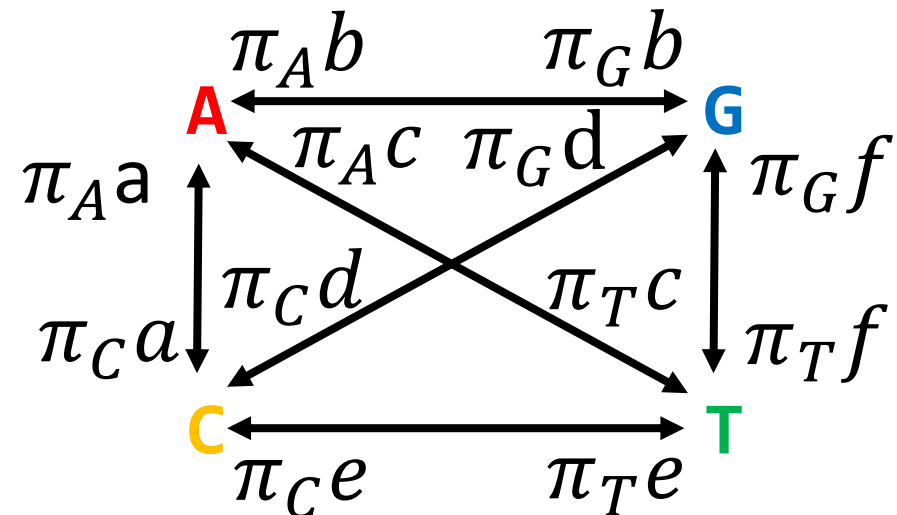
$$P = \begin{pmatrix} . & \pi_C \beta & \pi_G \alpha & \pi_T \beta \\ \pi_A \beta & . & \pi_G \beta & \pi_T \alpha \\ \pi_A \alpha & \pi_C \beta & . & \pi_T \beta \\ \pi_A \beta & \pi_C \alpha & \pi_G \beta & . \end{pmatrix}$$



Sequence Evolution: The GTR- Γ Model

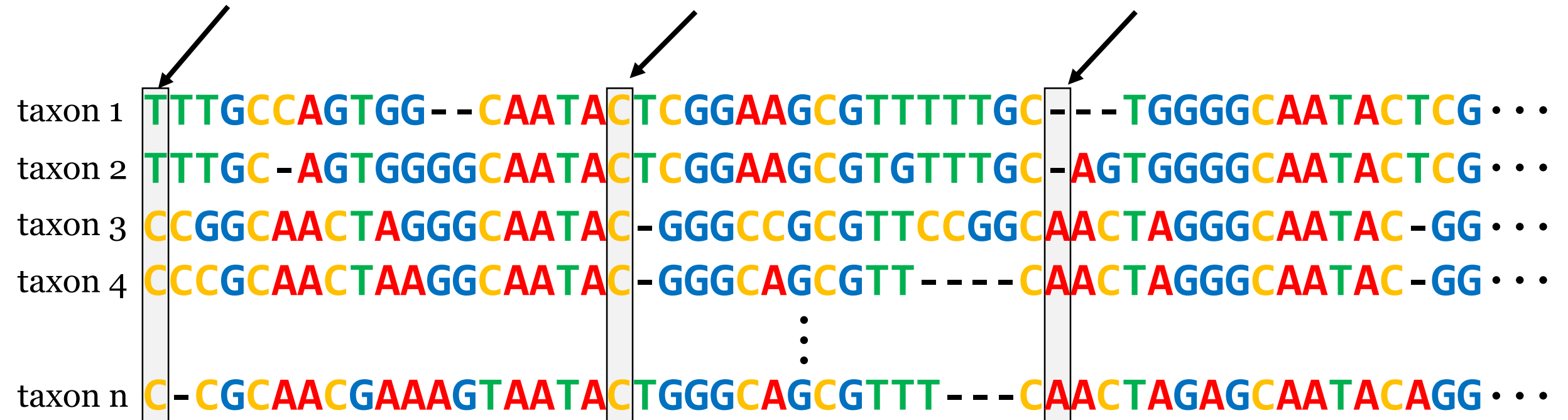
- The *General Time Reversible Gamma* model was published in 1990 but still is one of the most common distance models used today.
- Assumptions:
 - The substitution probability for each nucleotide pair i and j are individual but reversible (e.g. $A \rightarrow G = G \rightarrow A$).
 - The relative amounts of nucleotides can vary with $\pi_A, \pi_C, \pi_G, \pi_T$ being the relative amounts of **A**, **C**, **G** and **T**, respectively.
 - Each character site (i.e. alignment column) has its own substitution-matrix P_k .

$$P = \begin{pmatrix} . & \pi_C a & \pi_G b & \pi_T c \\ \pi_A a & . & \pi_G d & \pi_T e \\ \pi_A b & \pi_C d & . & \pi_T f \\ \pi_A c & \pi_C e & \pi_G f & . \end{pmatrix}$$



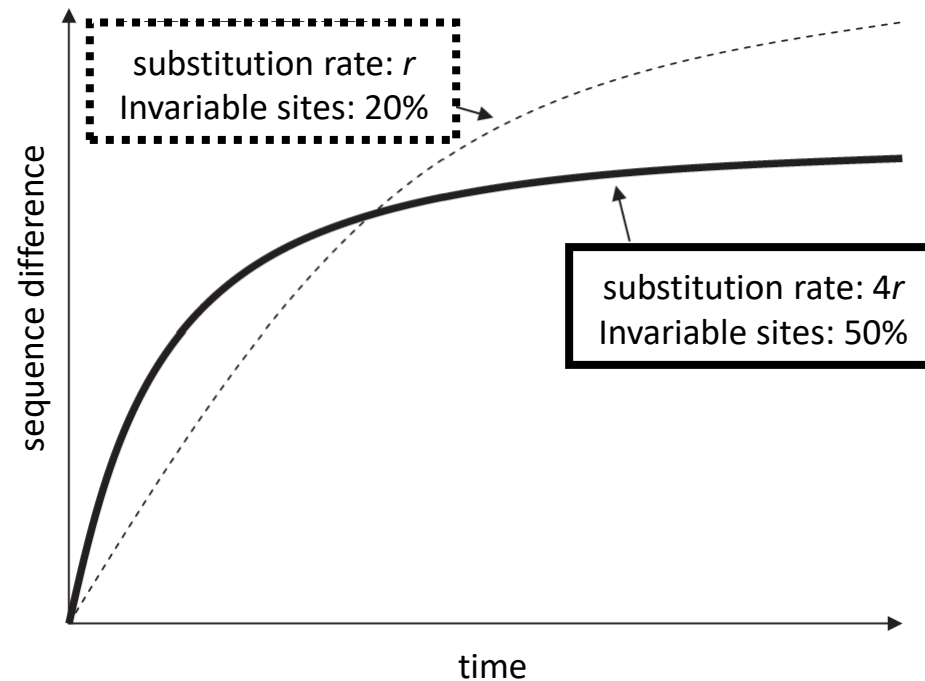
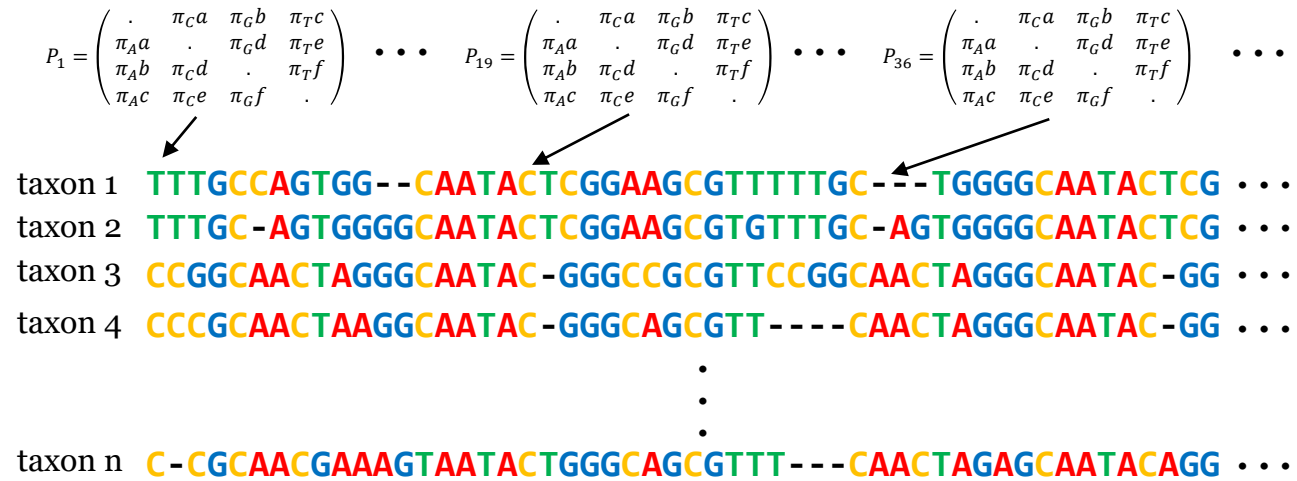
Sequence Evolution: The GTR- Γ Model

$$P_1 = \begin{pmatrix} . & \pi_{Ca} & \pi_{Gb} & \pi_{Tc} \\ \pi_{Aa} & . & \pi_{Gd} & \pi_{Te} \\ \pi_{Ab} & \pi_{Cd} & . & \pi_{Tf} \\ \pi_{Ac} & \pi_{Ce} & \pi_{Gf} & . \end{pmatrix} \quad \dots \quad P_{19} = \begin{pmatrix} . & \pi_{Ca} & \pi_{Gb} & \pi_{Tc} \\ \pi_{Aa} & . & \pi_{Gd} & \pi_{Te} \\ \pi_{Ab} & \pi_{Cd} & . & \pi_{Tf} \\ \pi_{Ac} & \pi_{Ce} & \pi_{Gf} & . \end{pmatrix} \quad \dots \quad P_{36} = \begin{pmatrix} . & \pi_{Ca} & \pi_{Gb} & \pi_{Tc} \\ \pi_{Aa} & . & \pi_{Gd} & \pi_{Te} \\ \pi_{Ab} & \pi_{Cd} & . & \pi_{Tf} \\ \pi_{Ac} & \pi_{Ce} & \pi_{Gf} & . \end{pmatrix} \quad \dots$$

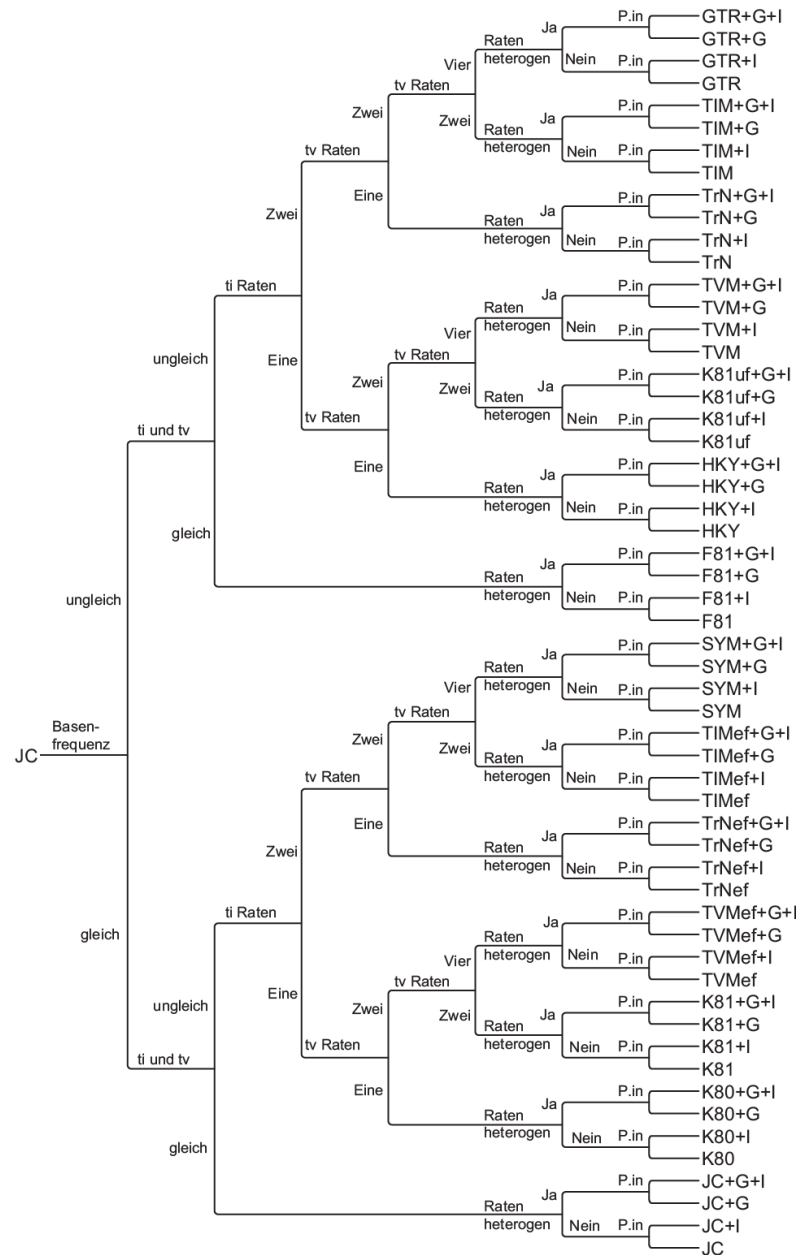


- All the parameters of the individual substitution-matrices P_k are **estimated** from the input data (i.e. the given sequence alignment).

Sequence Evolution: The GTR-Γ Model



Sequence Evolution: The Tree of Distance Models



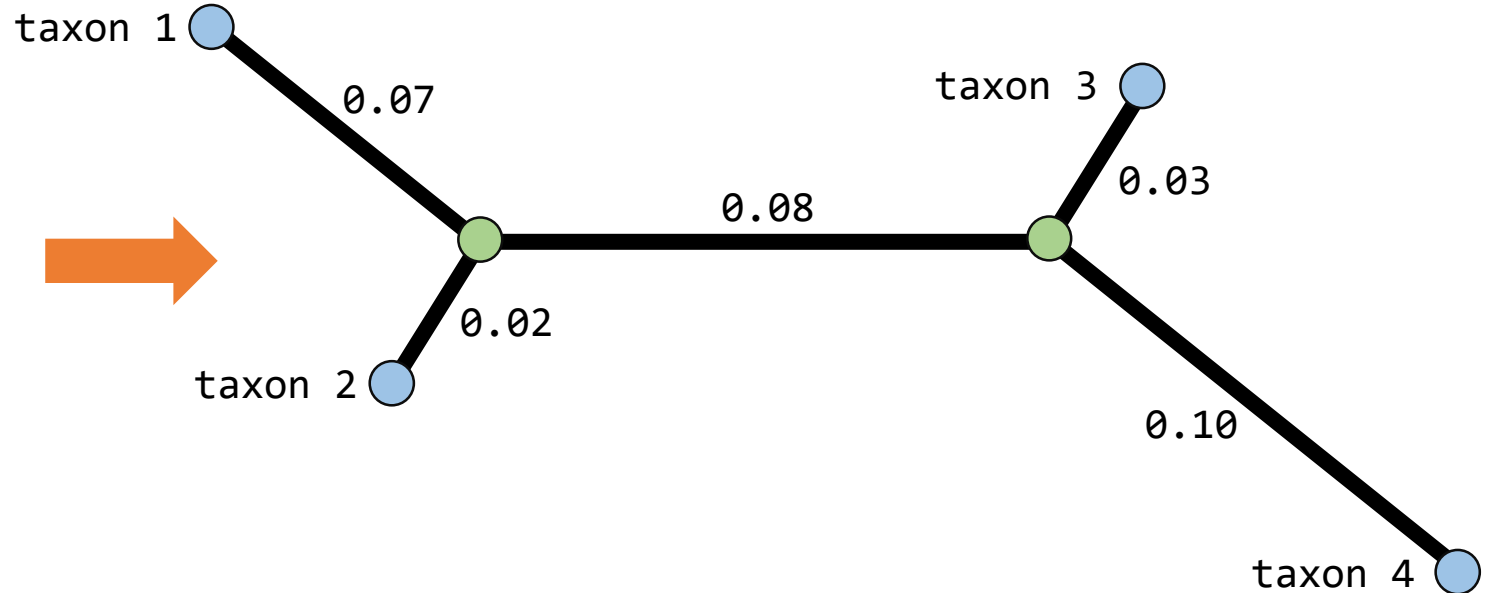


- *The quick'n'dirty approaches.* -

Algorithmic Phylogenetics: Distance Matrix Approaches

- Assuming we have a really good (i.e. *additive*) distance matrix of our input sequences and a given phylogenetic tree, then we can assign a distance value to each edge in the tree which also perfectly match the pairwise distances of our matrix.

	taxon 1	taxon 2	taxon 3	taxon 4
taxon 1	-	0.09	0.18	0.25
taxon 2		-	0.13	0.20
taxon 3			-	0.13
taxon 4				-

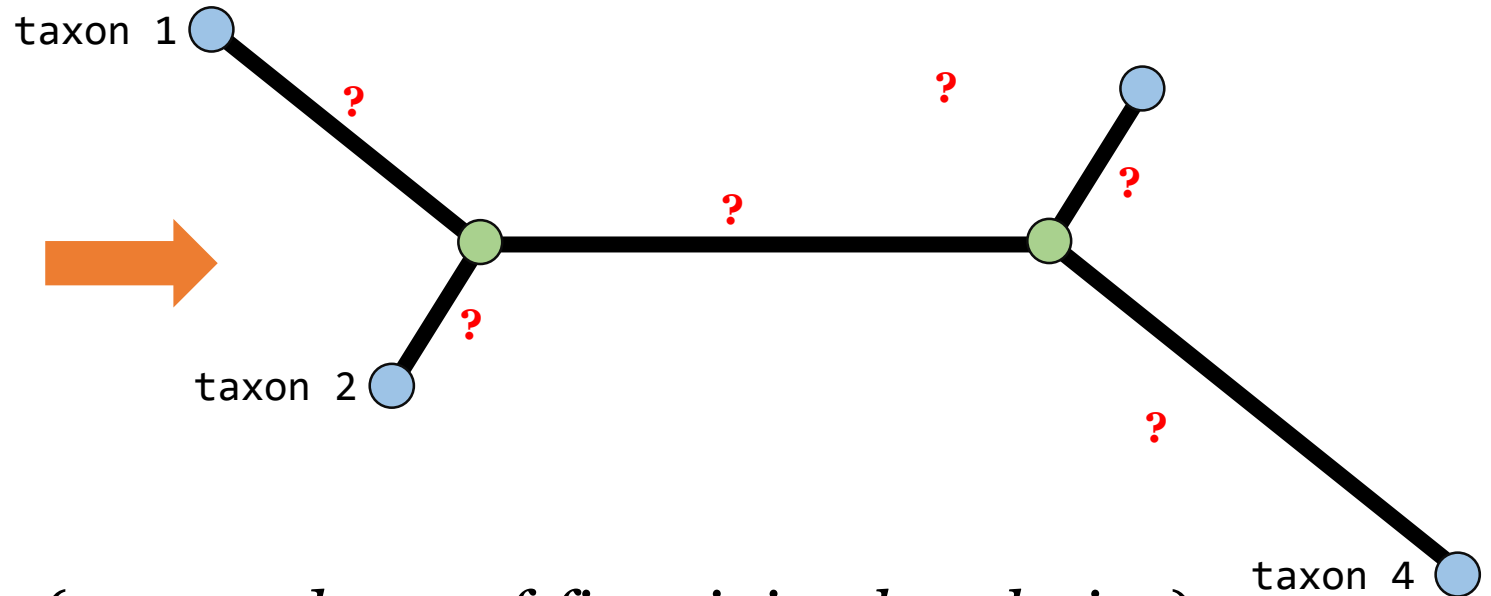


- Even better: if our distance matrix is *additive* then we can just reconstruct the correctly matching phylogenetic tree.

Algorithmic Phylogenetics: Distance Matrix Approaches

- In reality, our distance matrices are **almost never additive**.

	taxon 1	taxon 2	taxon 3	taxon 4
taxon 1	-	0.11	0.18	0.25
taxon 2		-	0.15	0.20
taxon 3			-	0.11
taxon 4				-



- There are several algorithms (e.g. *goodness-of-fit*, *minimal evolution*) that try to match the edge distances as closely as possible to the pairwise distances in a given distance matrix and for a given phylogenetic tree.
- To find the most fitting tree we have to search the *tree space* (→ similar to Maximum Parsimony problem)

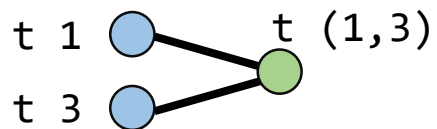
Algorithmic Phylogenetics: Clustering

- We can use our distance matrix to construct a fitting phylogenetic tree, instead of searching for the most fitting tree in *tree space*.
- For that, **clustering algorithms** are used, all following the same principle:
 1. Find the two taxa ***i*** and ***j*** that have the smallest distance
→ those two taxa must be siblings and have to be connected by an internal node named ***(i, j)***.
 2. Calculate the distances on the two edges between ***i*** and ***j***.
 3. Insert a new taxon ***(i, j)*** into the distance matrix and calculate the distance between the new taxon ***(i, j)*** and all remaining taxa, except ***i*** and ***j***.
 4. Remove ***i*** and ***j*** from the distance matrix.
 5. Repeat steps 1-4 until the tree is complete.

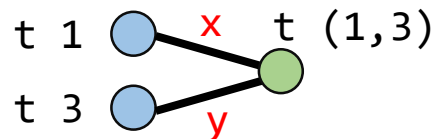
Algorithmic Phylogenetics: Clustering

Step 1

	t1	t2	t3	t4	t5
t1	-		min		
t2		-			
t3			-		
t4				-	
t5					-



Step 2



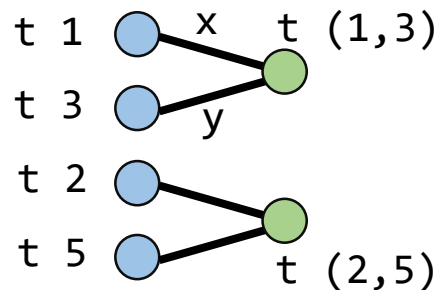
Step 3+4

	t (1,3)	t2	t4	t5
t (1,3)	-			
t2		-		
t4			-	
t5				-

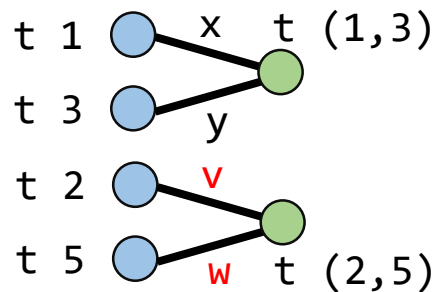
Step 5

Step 1

	t (1,3)	t2	t4	t5
t (1,3)	-			
t2		-		min
t4			-	
t5				-



Step 2



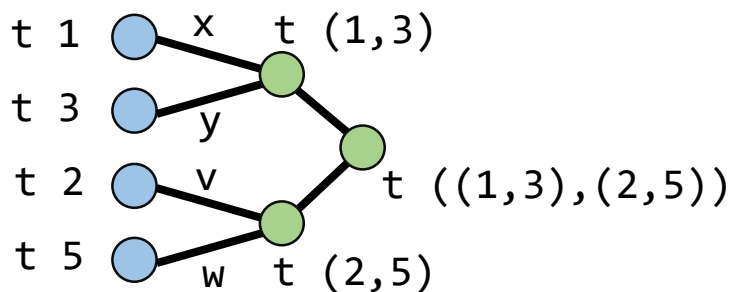
Step 3+4

	t (1,3)	t (2,5)	t4
t (1,3)	-		
t (2,5)		-	
t4			-

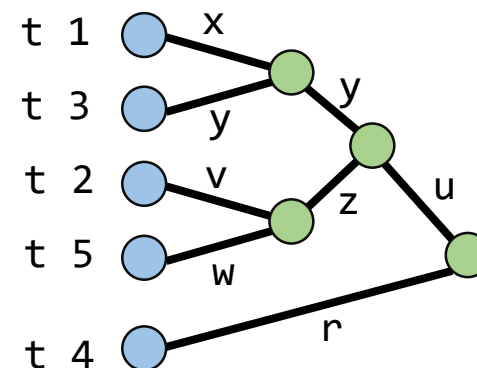
Step 5

Step 1

	t (1,3)	t (2,5)	t4
t (1,3)	-		
t (2,5)		-	
t4			-



...



Algorithmic Phylogenetics: UPGMA & WPGMA

- Use the following two clustering algorithms to build a phylogenetic tree from the given distance matrix.

UPGMA

- Find the two nodes ***i*** and ***j*** that have the smallest distance K_{ij} and connect them by an internal node (***i,j***).
- Assign to the two branches ***i*** → (***i,j***) and ***j*** → (***i,j***) the distance $\frac{K_{ij}}{2}$.
- Calculate the distances between (***i,j***) and every other node ***k*** (except node ***i*** and node ***j***) by:

$$K_{k,(i,j)} = K_{ki} * \left(\frac{n_i}{n_i + n_j} \right) + K_{kj} * \left(\frac{n_j}{n_i + n_j} \right)$$

where n_i and n_j are the total number of leaves beneath node ***i*** and node ***j***, respectively.

- Remove the nodes ***i*** and ***j*** from the distance matrix.
- Repeat steps 1-4 until the tree is complete.

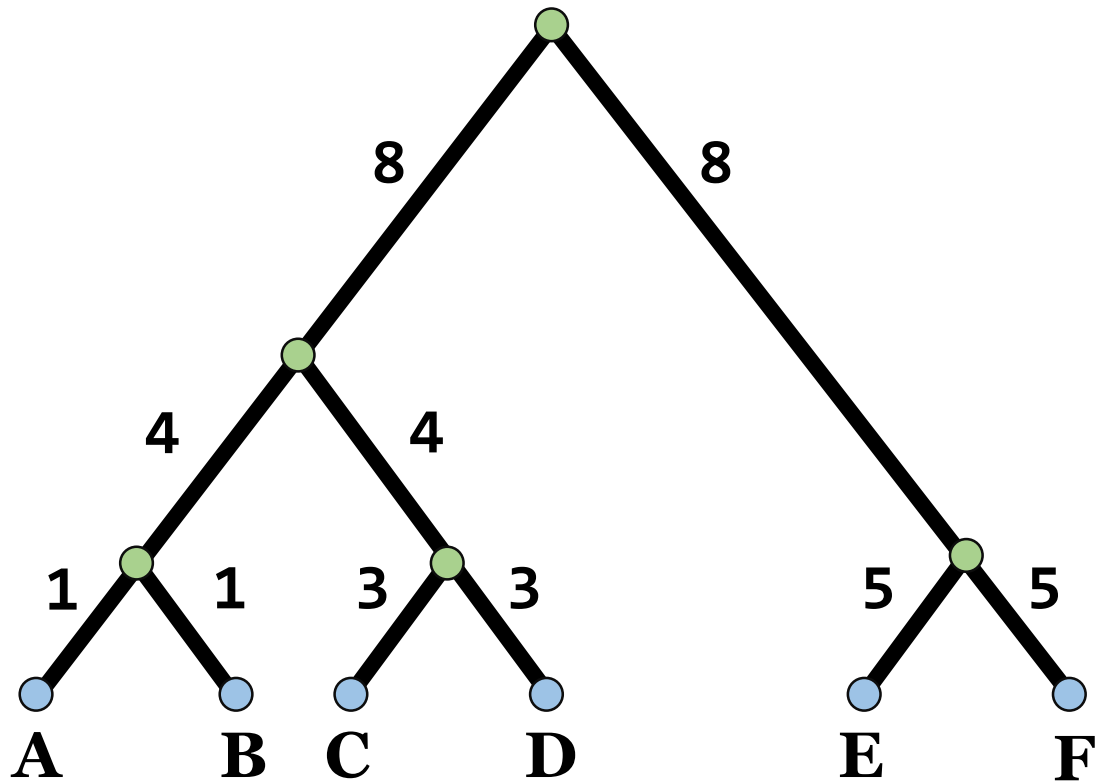
WPGMA

$$K_{k,(i,j)} = \frac{K_{ki}}{2} + \frac{K_{kj}}{2}$$

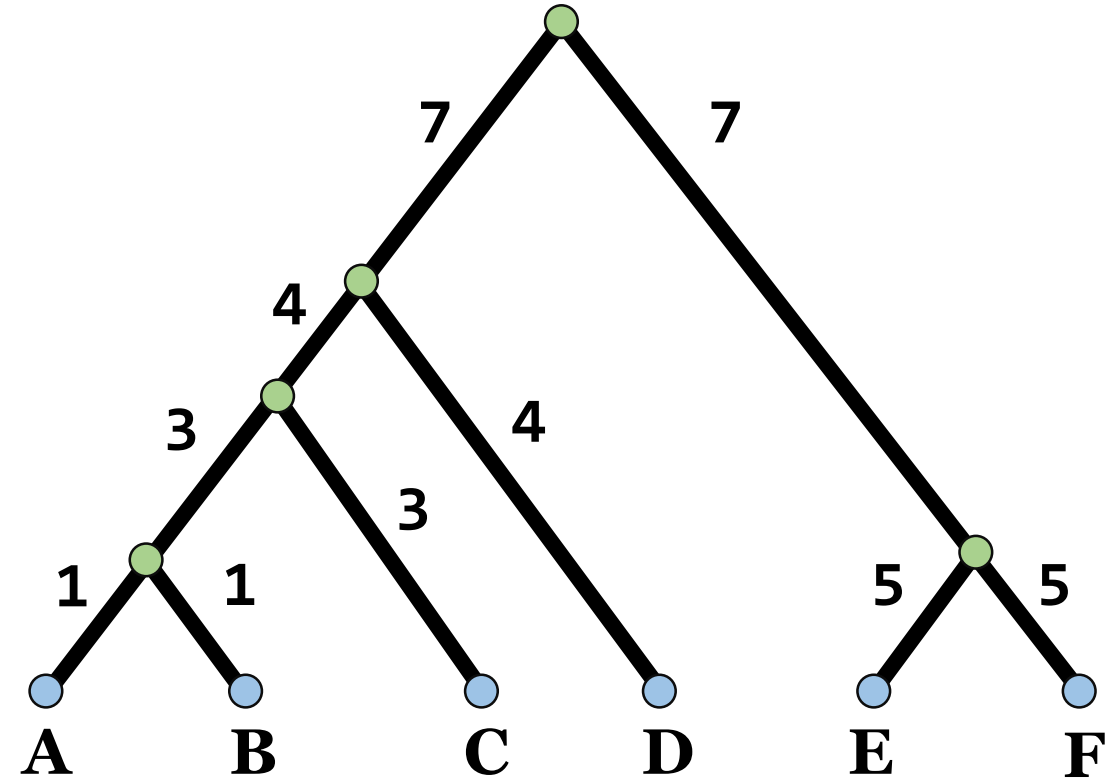
	A	B	C	D	E	F
A	0	2	8	12	18	18
B		0	4	8	18	18
C			0	6	18	18
D				0	8	12
E					0	10
F						0

Algorithmic Phylogenetics: UPGMA & WPGMA

UPGMA



WPGMA



Algorithmic Phylogenetics: Neighbor Joining

- UPGMA and WPGMA work best if the given distance matrix is an *ultra metric*, but in reality, this is even less common than an *additive* distance matrix.
- Today, the **Neighbor Joining** clustering algorithm is most commonly used and performs still considerably well even if the distance matrix is neither an *ultra metric* nor *additive*.
- It works very similar to UPGMA/WPGMA but with more complicated distance calculations at step 2 and step 3 of the clustering algorithm.
- Neighbor Joining is usually used to swiftly calculate an **approximation** to the phylogenetic tree that is searched.
 - Often used as a starting point for tree searches in *tree space*.

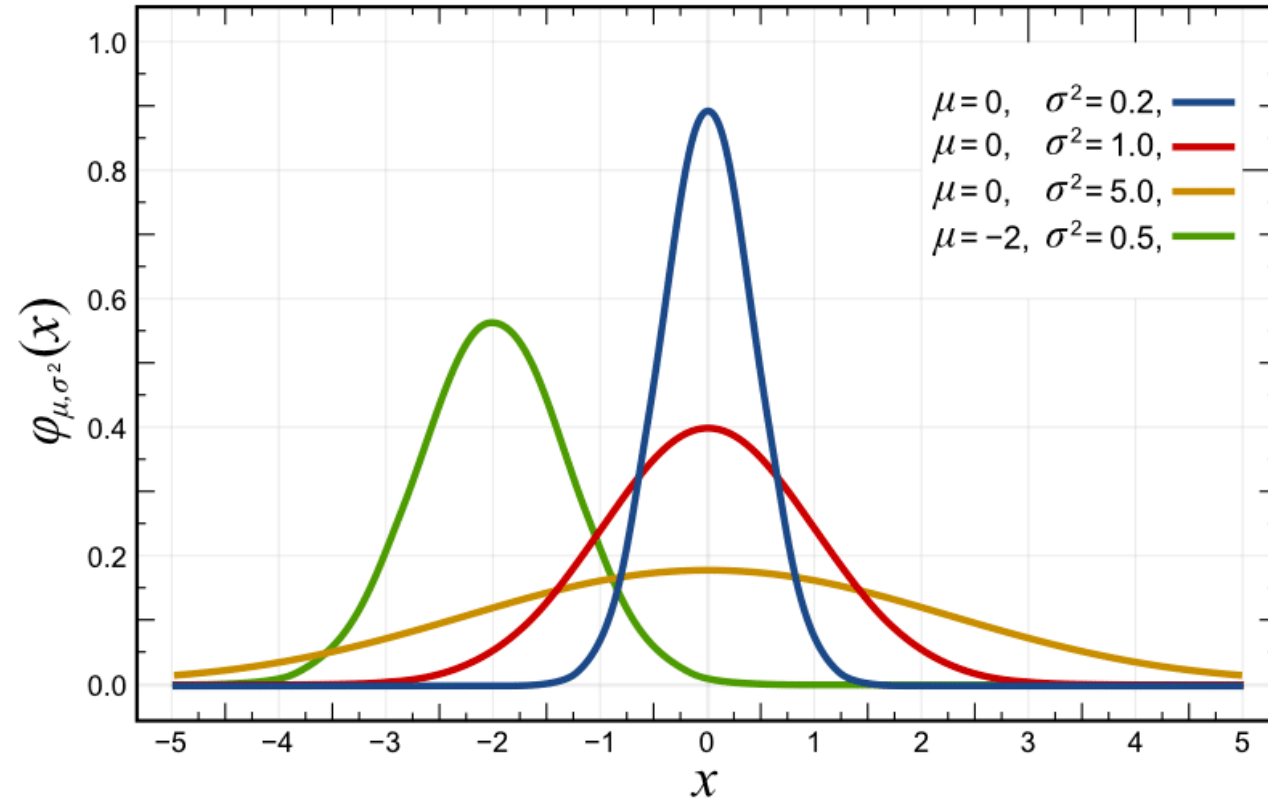
Algorithmic Phylogenetics: (Dis)advantages Of Distance Matrix approaches

Advantages

- Super fast
- Provide exactly one tree
- Make use of distance models
(i.e. more realistic models of evolution
than Parsimony)

Disadvantages

- Highly depend on several criteria
that the distance matrix has to fulfill
(*additivity, ultra metric*)
- Provide only one tree



Maximum likelihood

- It's all about probabilities. -

- A *random variable* X is a variable that can have different possible values, depending on the outcome of a random process → each possible value represents the probability of a certain outcome.

(To be more precise: X is a function that assigns a probability to each possible result of a random process.)

- Easy example:
 - We have a fair, six-sided dice and our random variable X is the number rolled on the dice.
 - The possible outcomes of one dice role are $\{1,2,3,4,5,6\}$ and they are called *elementary events*.
 - Since the dice is fair, they all have the same probability, denoted $P(X)$, with $P(X = 1) = \frac{1}{6}$, $P(X = 2) = \frac{1}{6}$, $P(X = 3) = \frac{1}{6}$, ... , $P(X = 6) = \frac{1}{6}$



- Easy example continued:
 - We have a fair, six-sided dice and our random variable X is the number rolled on the dice.
 - The possible outcomes of one dice role are $\{1,2,3,4,5,6\}$ and they are called *elementary events*.
 - Since the dice is fair, they all have the same probability, denoted $P(X)$, with $P(X = 1) = \frac{1}{6}$, $P(X = 2) = \frac{1}{6}$, $P(X = 3) = \frac{1}{6}$, ... , $P(X = 6) = \frac{1}{6}$
 - Now we can also calculate the probabilities of *non-elementary events*, like $P(X = \text{"Odd number"})$ which is a combination of elementary events
$$P(X = \text{"Odd number"}) = P(X = 1) + P(X = 3) + P(X = 5) = \frac{1}{2}$$



- More complex example:
 - We have two fair, six-sided dice (one green and one blue) and our random variable X is the pair of numbers rolled on both dice.
 - The possible *elementary events* of our our random variable X are $\{(1,1), (2,1), (3,1), (4,1), (5,1), (6,1), (1,2), \dots, (6,6)\}$.
 - Since both dice are fair, all *elementary events* have the same probability, with $P(X = (1,1)) = \frac{1}{36}$, $P(X = (2,1)) = \frac{1}{36}$, ... , $P(X = (6,6)) = \frac{1}{36}$
 - We can calculate the probabilities of *non-elementary events*, like $P(X = \text{"number on green dice is odd and on blue greater than 5"}) =$
$$P((1,6)) + P((3,6)) + P((5,6)) = \frac{3}{36} = \frac{1}{12}$$



- We can also calculate *conditional probabilities* $P(X = A|B)$ that reads like “What is the probability of event A given that *event B already happened?*”

- **Bayes' theorem:** $P(A|B) = \frac{P(B|A)*P(A)}{P(B)} = \frac{P(A \cap B)}{P(B)}$ ← „ $A \cap B$ “ means that event A and event B happen at the same time.

- Example: $P(X = \text{"sum of both dice is 8"} \mid \text{"number on both dice is different"})$

$$P(X) = \frac{4/36}{30/36} = \frac{1/9}{5/6} = \frac{2}{15}$$

Algorithmic Phylogenetics: Probabilities On Sequences

- We can use the concept of random variables also for sequences:
 - X is a nucleotide sequence with the nucleotide frequencies
 $P(X = \text{"A"}) = \frac{4}{9}, P(X = \text{"C"}) = \frac{1}{9}, P(X = \text{"G"}) = \frac{1}{9}, P(X = \text{"T"}) = \frac{3}{9}$
 - We can calculate the probability of any given nucleotide sequence:

$$P(\text{T T T A C G G T A}) = \\ P(\text{T}) * P(\text{T}) * \dots * P(\text{A}) = \left(\frac{4}{9}\right)^2 * \left(\frac{1}{9}\right)^3 * \left(\frac{3}{9}\right)^4 \approx 0.00003345$$

Algorithmic Phylogenetics: Probabilities On Sequences

- But what if we do not know the nucleotide frequencies of our sequence?

$$P(X = \text{"A"}) = ?, P(X = \text{"C"}) = ?, P(X = \text{"G"}) = ?, P(X = \text{"T"}) = ?$$

- We can make a *hypothesis* H and can calculate how likely our given data D fits this hypothesis:

- $H: P(\text{"A"}) = \frac{1}{4}, P(\text{"C"}) = \frac{1}{4}, P(\text{"G"}) = \frac{1}{4}, P(\text{"T"}) = \frac{1}{4}$

- $D: \text{"TTTACGGTA"}$

$$\text{Likelihood} = P(D|H) = P\left(\text{TTTACGGTA} \mid \text{A}=\text{C}=\text{G}=\text{T}=\frac{1}{4}\right) = \left(\frac{1}{4}\right)^9 \approx 3,8 * 10^{-6}$$



The *Maximum Likelihood analysis* tries to identify the hypothesis H that maximizes the likelihood of the given data D .

Algorithmic Phylogenetics: Probabilities On Sequences

- We can make a *hypothesis* H and can calculate how likely our given data D fits this hypothesis:
 - $H: P(\text{"A"}) = \frac{1}{4}, P(\text{"C"}) = \frac{1}{4}, P(\text{"G"}) = \frac{1}{4}, P(\text{"T"}) = \frac{1}{4}$
 - $D: \text{"TTTACGGTA"}$

$$\text{Likelihood} = P(D|H)$$

$$= P(D_1|H) * P(D_2|H) * P(D_3|H) * \dots * P(D_n|H)$$

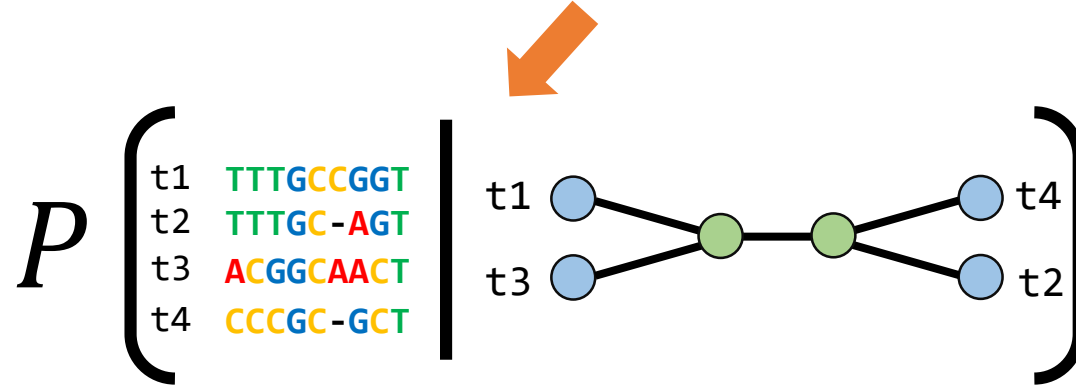
$$= \prod_{i=1}^n P(D_i|H)$$

$$\begin{aligned} \text{Likelihood} &= P\left(\text{TTTACGGTA} \mid \text{A}=\text{C}=\text{G}=\text{T}=\frac{1}{4}\right) \\ &= P\left(\text{T} \mid \text{A}=\text{C}=\text{G}=\text{T}=\frac{1}{4}\right) * P\left(\text{T} \mid \text{A}=\text{C}=\text{G}=\text{T}=\frac{1}{4}\right) * P\left(\text{T} \mid \text{A}=\text{C}=\text{G}=\text{T}=\frac{1}{4}\right) * P\left(\text{A} \mid \text{A}=\text{C}=\text{G}=\text{T}=\frac{1}{4}\right) * \dots * P\left(\text{A} \mid \text{A}=\text{C}=\text{G}=\text{T}=\frac{1}{4}\right) \\ &= \left(\frac{1}{4}\right)^9 \approx 3,8 * 10^{-6} \end{aligned}$$

Algorithmic Phylogenetics: Back To Phylogenetics

- The ideas of *conditional probabilities* and *Maximum Likelihood*, can also be applied to phylogenetics analysis.
- Our *hypothesis H* is a given phylogenetic tree and our data *D* is a multiple sequence alignment.

$$\text{Likelihood} = P(D|H) = P(\text{sequence alignment} | \text{tree})$$



Algorithmic Phylogenetics: Back To Phylogenetics

- We know that the *total likelihood* of our data (i.e. alignment) matching the given tree equals the product of the likelihoods of the single data points (i.e. alignment columns) matching the given tree.

$$\text{Likelihood} = P(D|H) = \prod_{i=1}^n P(D_i|H)$$

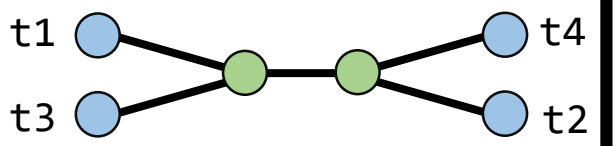
$$P \left(\begin{array}{c} \text{TTTGCCGGT} \\ \text{TTTGC-AGT} \\ \text{ACGGCAACT} \\ \text{CCCGC-GCT} \end{array} \middle| \begin{array}{c} \text{t1} \text{ } \text{t4} \\ \text{t3} \text{ } \text{t2} \end{array} \right) =$$

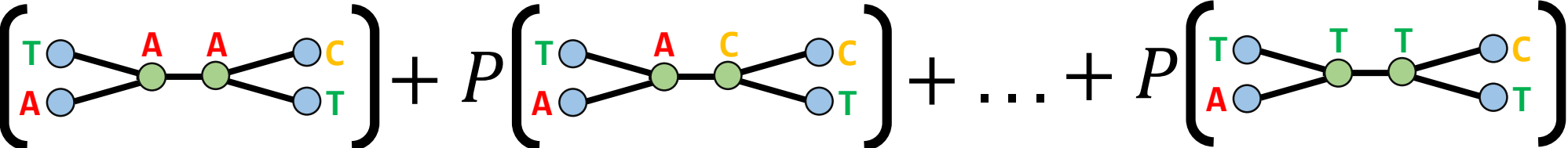
$$P_1 \left(\begin{array}{c} \text{t1 T} \\ \text{t2 T} \\ \text{t3 A} \\ \text{t4 C} \end{array} \middle| \begin{array}{c} \text{t1} \text{ } \text{t4} \\ \text{t3} \text{ } \text{t2} \end{array} \right) * P_2 \left(\begin{array}{c} \text{t1 T} \\ \text{t2 T} \\ \text{t3 C} \\ \text{t4 C} \end{array} \middle| \begin{array}{c} \text{t1} \text{ } \text{t4} \\ \text{t3} \text{ } \text{t2} \end{array} \right) * \dots$$

- The single data point probabilities P_1, \dots, P_n are also termed *character probabilities*.

Algorithmic Phylogenetics: Back To Phylogenetics

- To calculate the *character probabilities* we have to consider all possible *substitution scenarios* of our current data point and the given tree.

$$P_1 \left(\begin{array}{c} t1 \text{ T} \\ t2 \text{ T} \\ t3 \text{ A} \\ t4 \text{ C} \end{array} \middle| \begin{array}{c} t1 \text{ } \\ t3 \text{ } \end{array} \begin{array}{c} \text{ } \\ \text{ } \end{array} \begin{array}{c} t4 \\ t2 \end{array} \right) =$$


$$\underbrace{P \left(\begin{array}{c} \text{T} \\ \text{A} \end{array} \begin{array}{c} \text{ } \\ \text{ } \end{array} \begin{array}{c} \text{A} \\ \text{A} \end{array} \begin{array}{c} \text{C} \\ \text{T} \end{array} \right) + P \left(\begin{array}{c} \text{T} \\ \text{A} \end{array} \begin{array}{c} \text{ } \\ \text{ } \end{array} \begin{array}{c} \text{A} \\ \text{C} \end{array} \begin{array}{c} \text{C} \\ \text{T} \end{array} \right) + \dots + P \left(\begin{array}{c} \text{T} \\ \text{A} \end{array} \begin{array}{c} \text{ } \\ \text{ } \end{array} \begin{array}{c} \text{T} \\ \text{T} \end{array} \begin{array}{c} \text{C} \\ \text{T} \end{array} \right)}$$


In our example, there are 16 possible *substitution scenarios*.

Algorithmic Phylogenetics: Back To Phylogenetics

- To calculate the probability of a *substitution scenario* we have to consider the probability of each single substitution within the scenario.

$$P \left(\begin{array}{c} \text{Tree with 4 tips: T, A, C, T} \\ \text{Internal nodes: A, A} \end{array} \right) = \underbrace{P_{A \leftrightarrow T} * P_{A \leftrightarrow T} * P_{A \leftrightarrow A} * P_{A \leftrightarrow C} * P_{A \leftrightarrow T}}$$

- And finally, the probabilities of these single substitutions are obtained from the distance model used (e.g. Jukes-Cantor, YK2, GTR- Γ , ...).

Algorithmic Phylogenetics: Maximum Likelihood Analysis For Trees

$$P \left[\begin{array}{c} \text{Tree with 4 tips: T, A, C, T} \\ \text{Internal nodes: A, A} \end{array} \right] = P_{A \leftrightarrow T} * P_{A \leftrightarrow T} * P_{A \leftrightarrow A} * P_{A \leftrightarrow C} * P_{A \leftrightarrow T}$$

Calculate probability of *substitution scenarios* based on single substitution probabilities.

$$P \left[\begin{array}{c} \text{Tree with 4 tips: T, A, C, T} \\ \text{Internal nodes: A, A} \end{array} \right]$$

Calculate *character probabilities* based on substitution scenarios.

$$P_1 \left[\begin{array}{c} \text{t1 T} \\ \text{t2 T} \\ \text{t3 A} \\ \text{t4 C} \end{array} \middle| \begin{array}{c} \text{Tree with 4 tips: t1, t2, t3, t4} \\ \text{Internal nodes: } \end{array} \right]$$

Calculate *likelihood* of the sequence alignment matching the given phylogenetic tree based on the *character probabilities*.

$$P \left[\begin{array}{c} \text{t1 TTTGCGGT} \\ \text{t2 TTTGC-AGT} \\ \text{t3 ACGGCAACT} \\ \text{t4 CCCGC-GCT} \end{array} \middle| \begin{array}{c} \text{Tree with 4 tips: t1, t2, t3, t4} \\ \text{Internal nodes: } \end{array} \right]$$

We have to search the *tree space* again to find the tree with the *maximal likelihood* for our alignment.

Algorithmic Phylogenetics: (Dis)advantages Of Maximum Likelihood Analysis

Advantages

- We have an actual likelihood for the resulting trees → we have some kind of quality measurement for the trees
→ we can rank different resulting trees
- Make use of distance models
(i.e. more realistic models of evolution than Parsimony)

Disadvantages

- Computationally very demanding
→ much slower than Parsimony and distance matrix approaches

What we have learned today

- You know the fundamental differences between Maximum Parsimony, distance based and Maximum Likelihood methods and when to use which approach.
- You have a good idea how sequence evolution can be modelled mathematically using distance models.
- You know that the better your sequence alignment, than the better your distance models and than the better your phylogenetic trees.