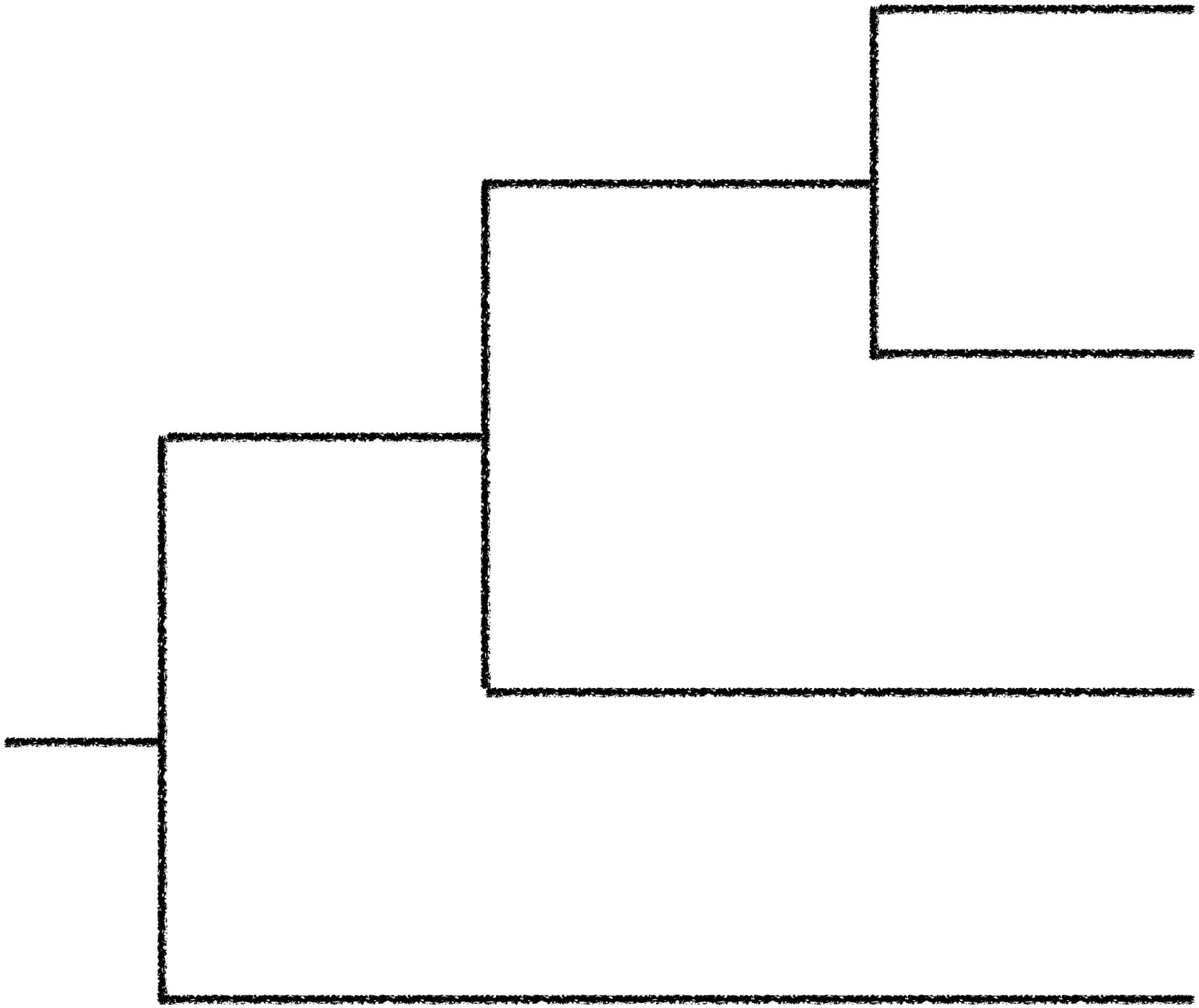
Lecture 7

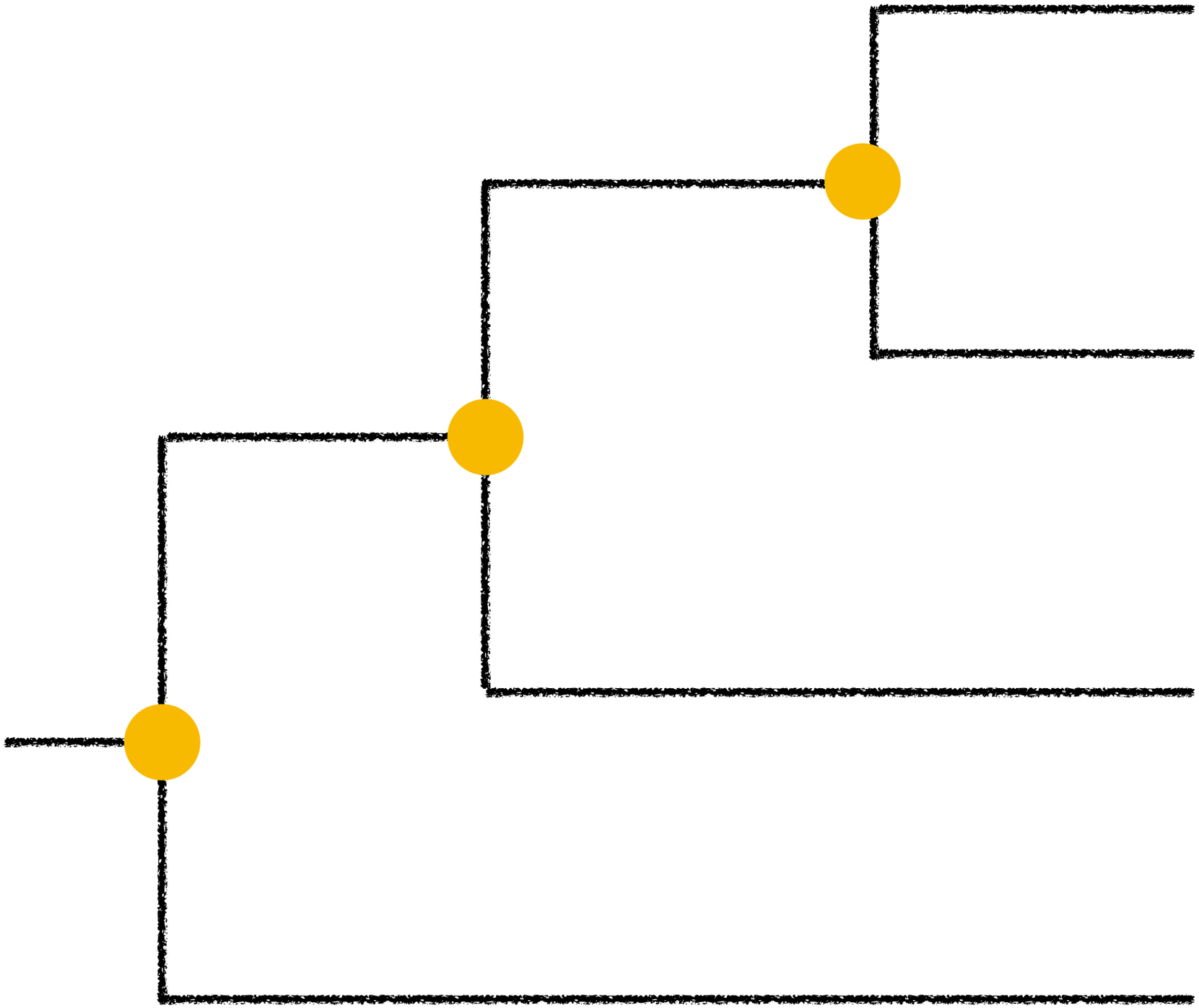
Overview of phylogenetic inference
Botany/Plant Path 563

- **Previous class check-up:**
    - We studied the filtering methods after MSA and some overview of orthology detection methods
- **Learning Objectives:** At the end of today's session, you will be able to
    - explain the overall methodology of phylogenetic inference as well as the main weaknesses
- **No pre-class work**

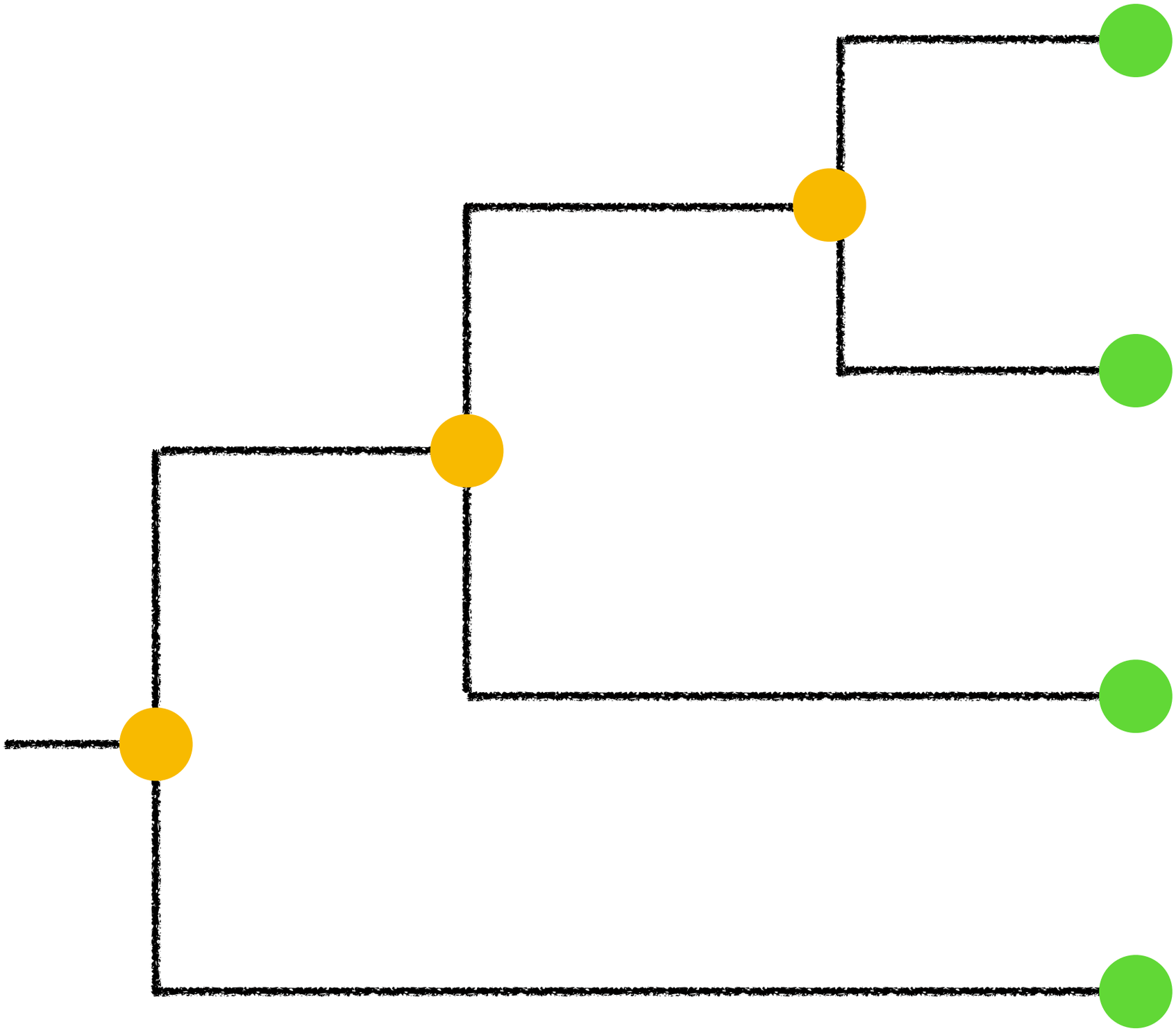
Phylogenetic tree



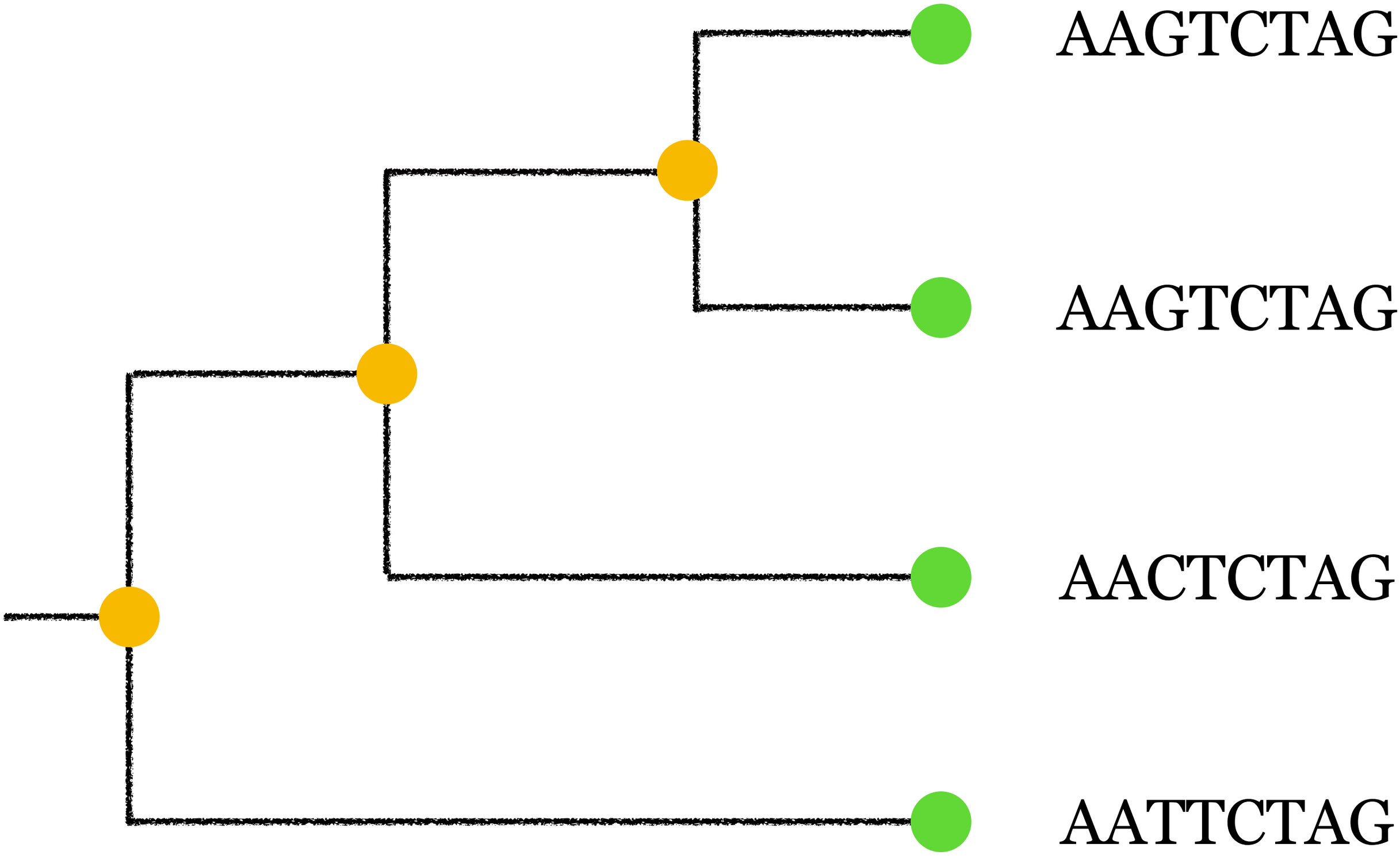
Phylogenetic tree



Phylogenetic tree

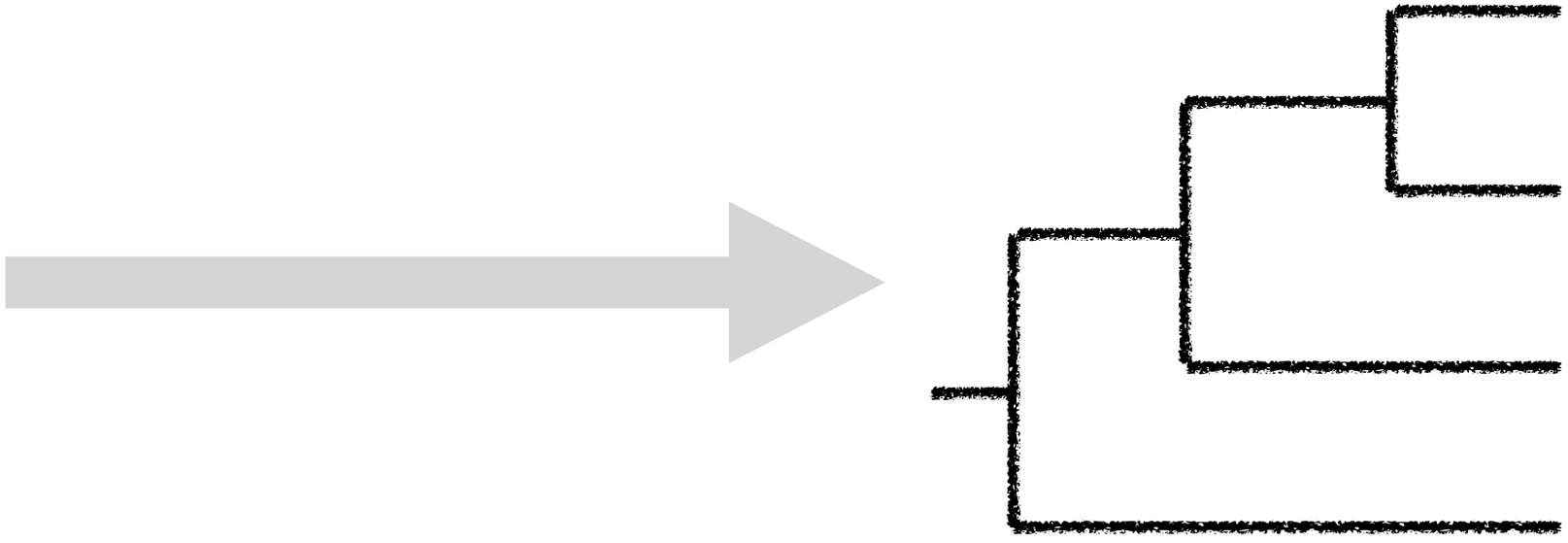


Phylogenetic tree

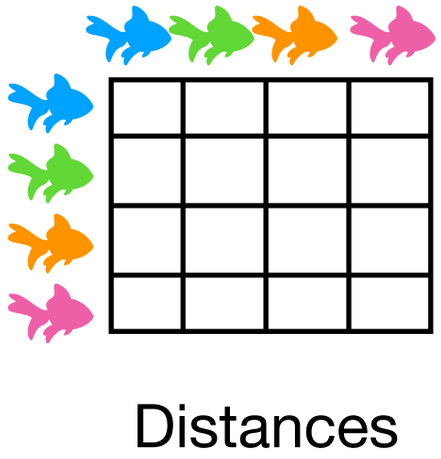


Phylogenetic inference

AAGTCTAG
AAGTCTAG
AACTCTAG
AATTCTAG



Phylogenetic inference



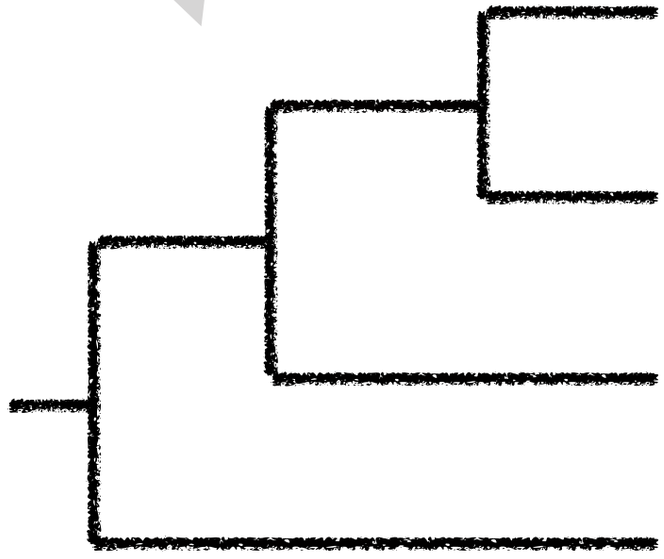
Distances

AAGTCTAG

AAGTCTAG

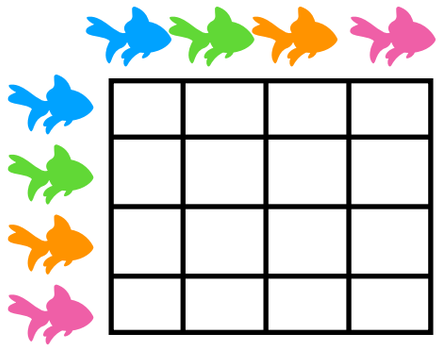
AACTCTAG

AATTCTAG



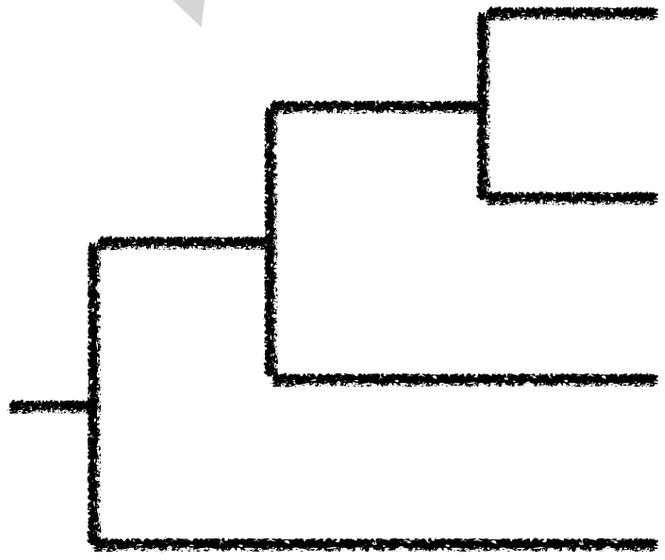
Phylogenetic inference

AAGTCTAG
AAGTCTAG
AACTCTAG
AATTCTAG

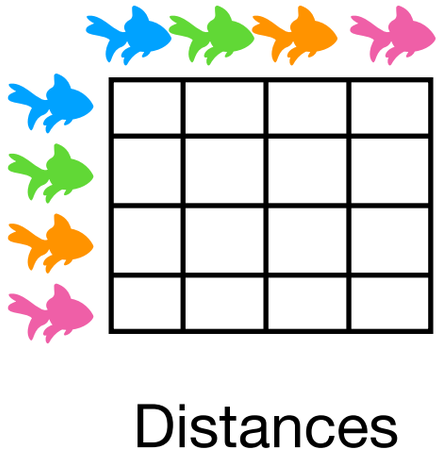


Distances

Loss of info
Fast and scalable

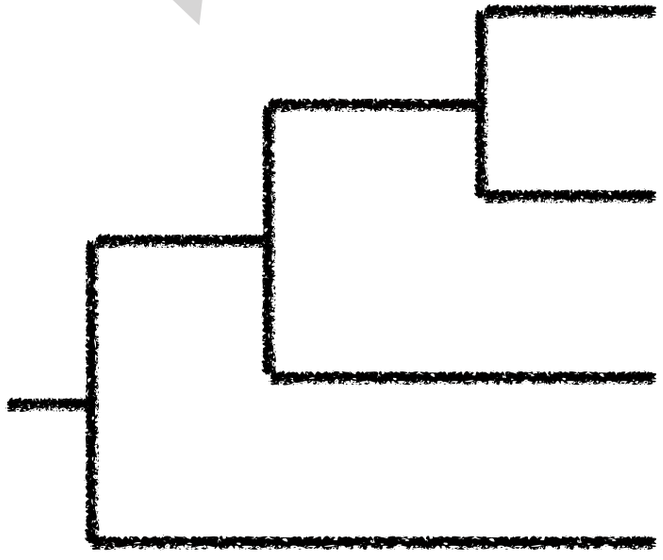


Phylogenetic inference



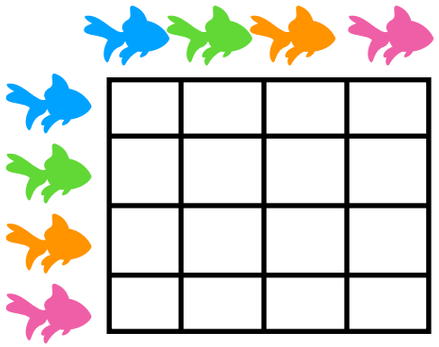
Distances

AAGTCTAG
AAGTCTAG
AACTCTAG
AATTCTAG



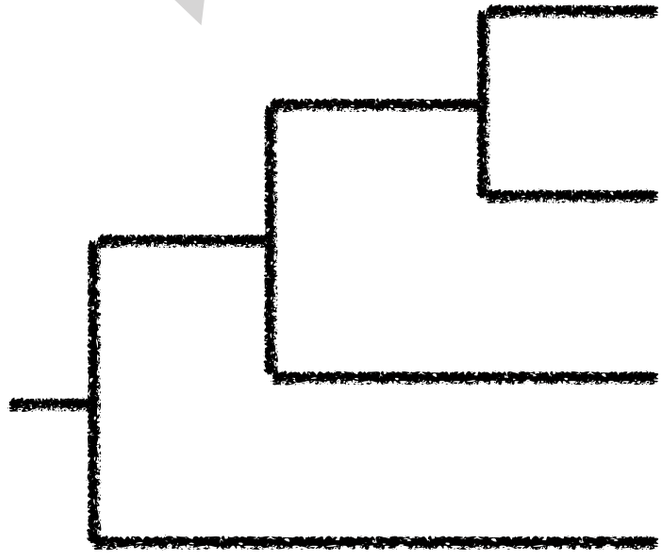
f(| AAGTCTAG
AAGTCTAG
AACTCTAG
AATTCTAG)

Phylogenetic inference



Distances

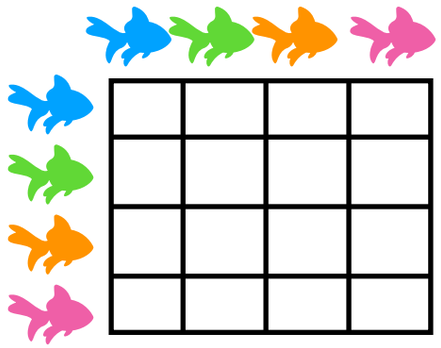
AAGTCTAG
AAGTCTAG
AACTCTAG
AATTCTAG



f( | AAGTCTAG
AAGTCTAG
AACTCTAG
AATTCTAG)

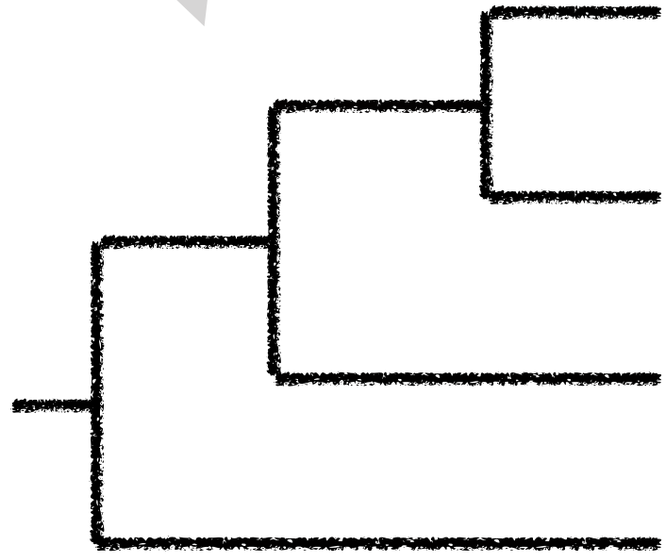
Parsimony score
Likelihood score

Phylogenetic inference

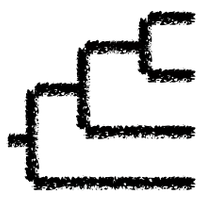


Distances

Criteria:
Find the tree that minimizes the distances and agrees with the observed distances



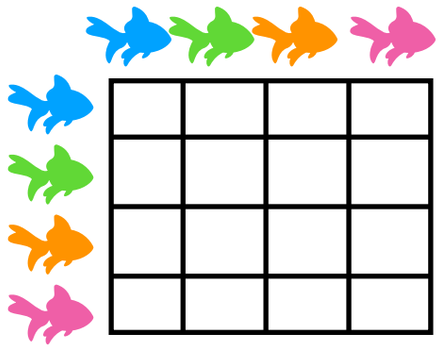
AAGTCTAG
AAGTCTAG
AACTCTAG
AATTCTAG

f( | AAGTCTAG
AAGTCTAG
AACTCTAG
AATTCTAG)

Parsimony score
Likelihood score

Criteria:
Find the tree that maximizes the fit to the data based on the score

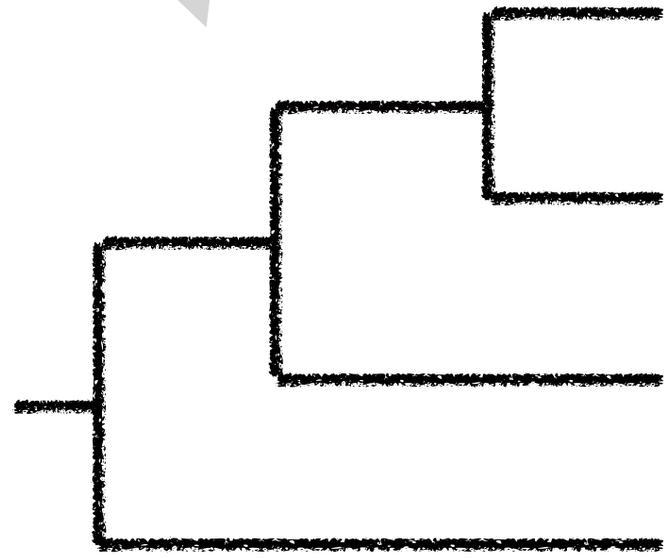
Phylogenetic inference



Distances

AAGTCTAG
AAGTCTAG
AACTCTAG
AATTCTAG

Find a tree that is optimal under certain criteria (score)



f( | AAGTCTAG
AAGTCTAG
AACTCTAG
AATTCTAG)

Parsimony score
Likelihood score

Phylogenetic inference


AAGTCTAG


AAGTCTAG


AACTCTAG


AATTCTAG

Phylogenetic inference

Step 1: Choose the criterion to use: distances, parsimony, likelihood


AAGTCTAG


AAGTCTAG


AACTCTAG

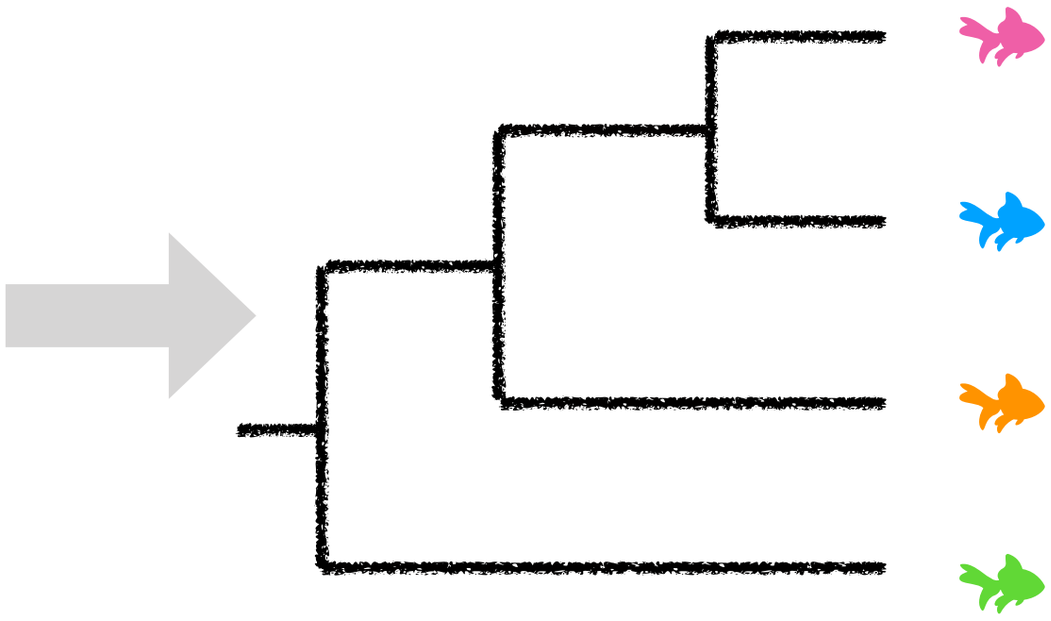

AATTCTAG

Phylogenetic inference

Step 1: Choose the criterion to use: distances, parsimony, likelihood

Step 2: Guess the tree

AAGTCTAG
AAGTCTAG
AACTCTAG
AATTCTAG

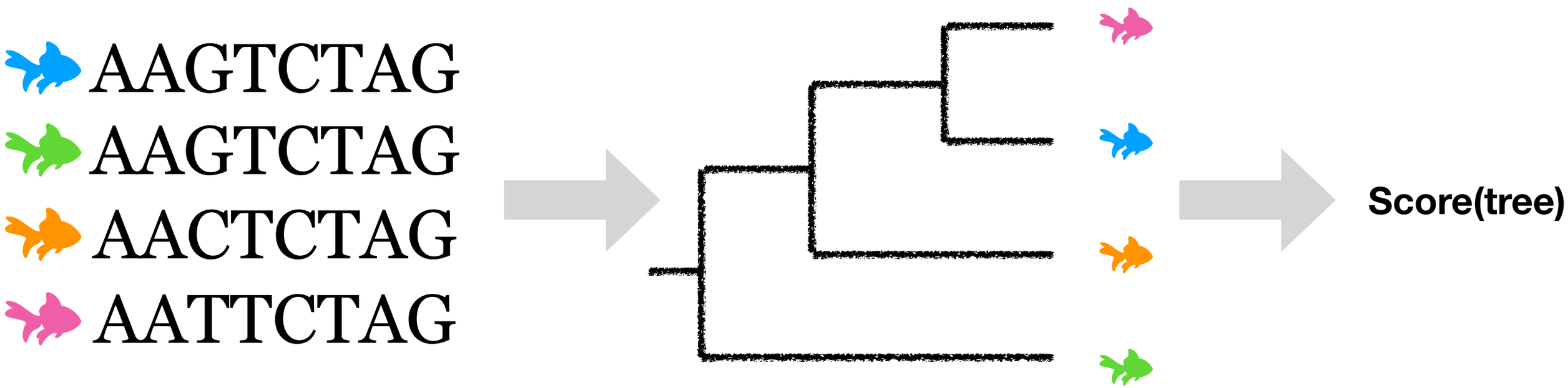


Phylogenetic inference

Step 1: Choose the criterion to use: distances, parsimony, likelihood

Step 2: Guess the tree

Step 3: Evaluate the score of the tree

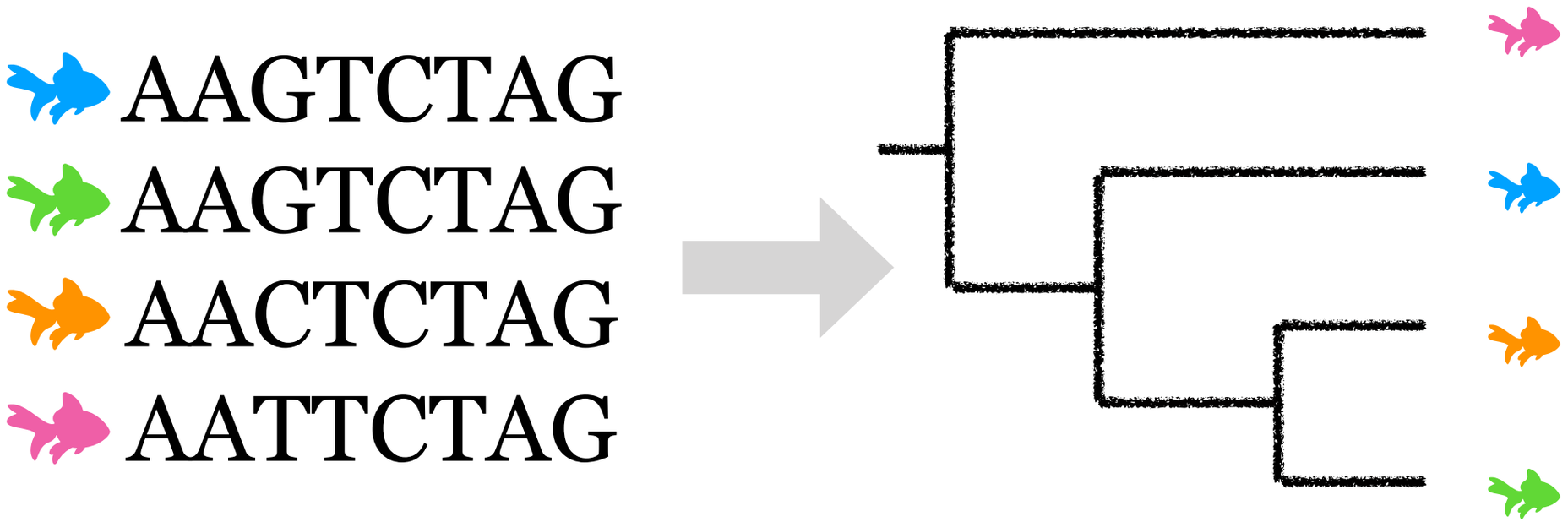


Phylogenetic inference

Step 1: Choose the criterion to use: distances, parsimony, likelihood

Step 2: Guess the tree

Step 3: Evaluate the score of the tree



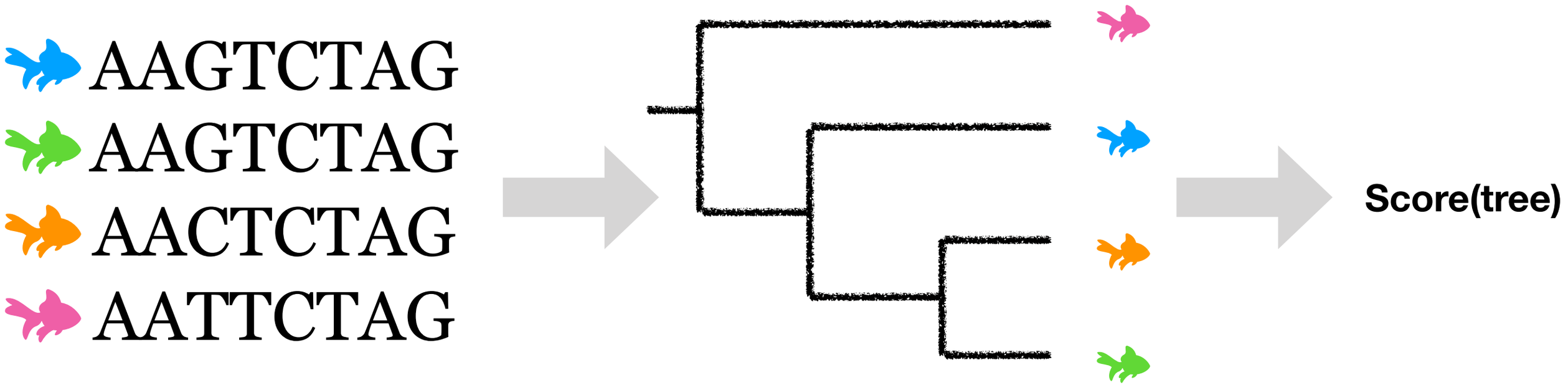
Step 4: Propose new tree

Phylogenetic inference

Step 1: Choose the criterion to use: distances, parsimony, likelihood

Step 2: Guess the tree

Step 3: Evaluate the score of the tree



Step 4: Propose new tree

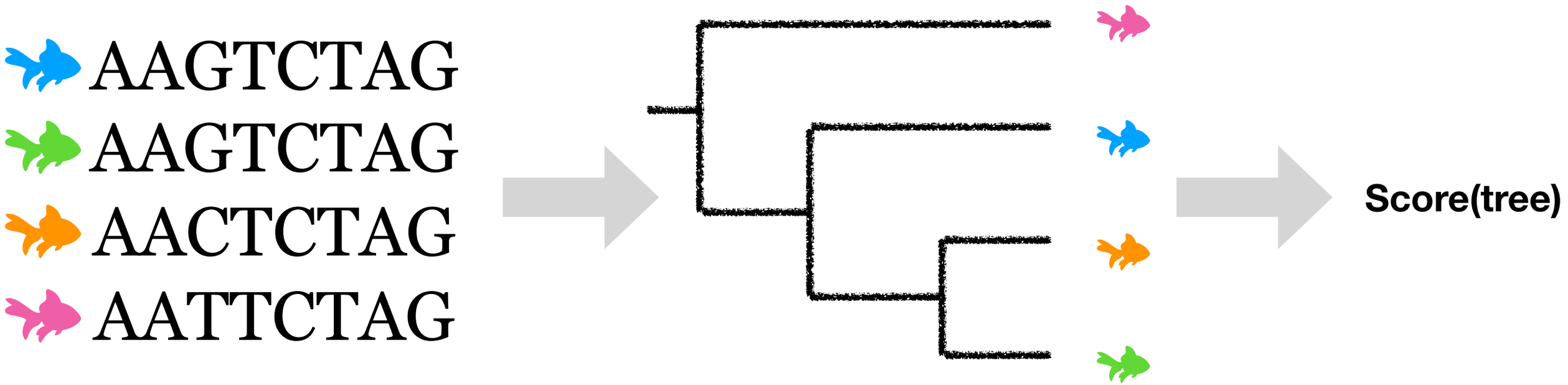
Step 5: Evaluate the score of the tree

Phylogenetic inference

Step 1: Choose the criterion to use: distances, parsimony, likelihood

Step 2: Guess the tree

Step 3: Evaluate the score of the tree



Step 4: Propose new tree

Step 5: Evaluate the score of the tree

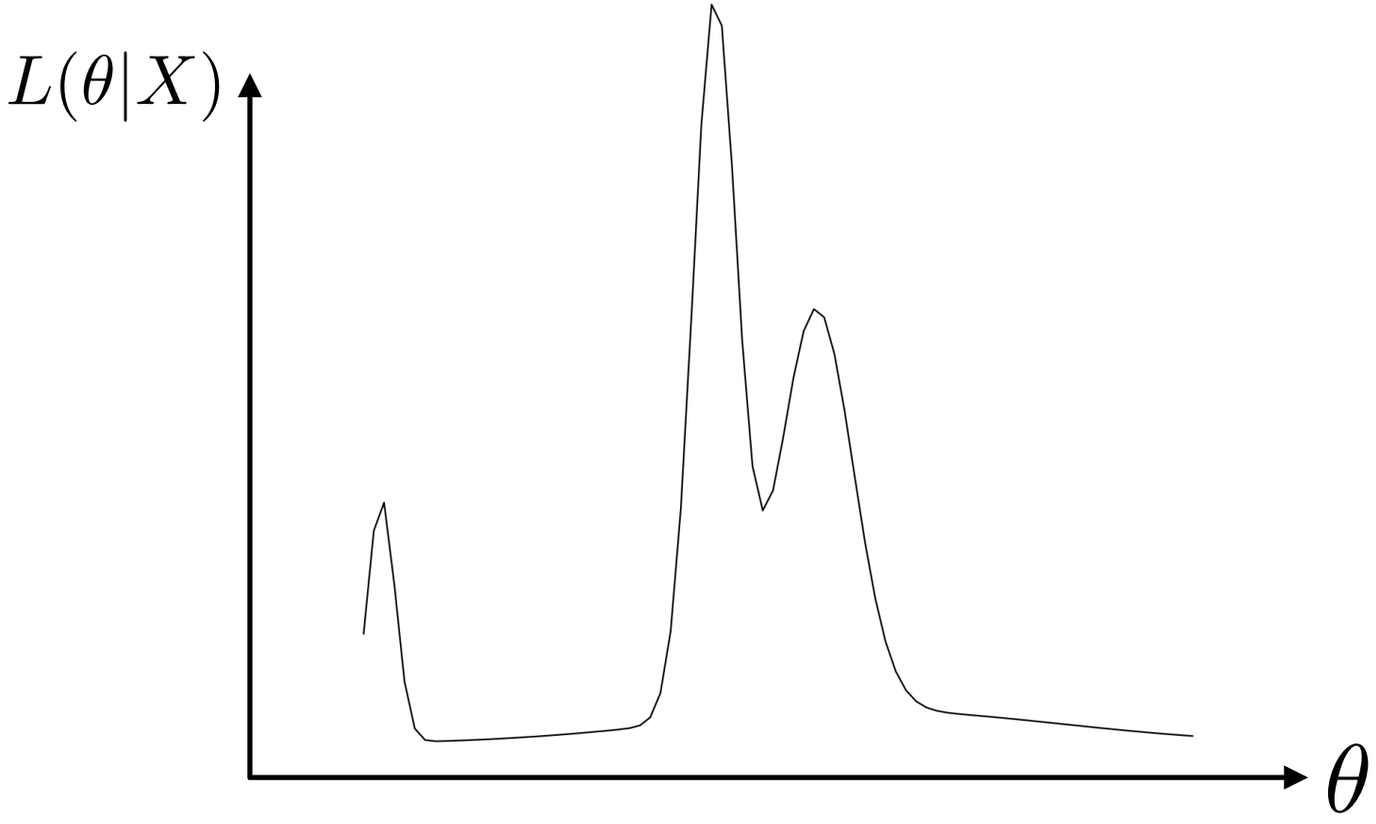
Continue until you scored all trees and found the optimum

Example: maximum likelihood

θ — Parameter of interest ← Tree

X — Data ← Sequences

$L(\theta|X)$ — Likelihood: function of parameter given data ← Substitution Model

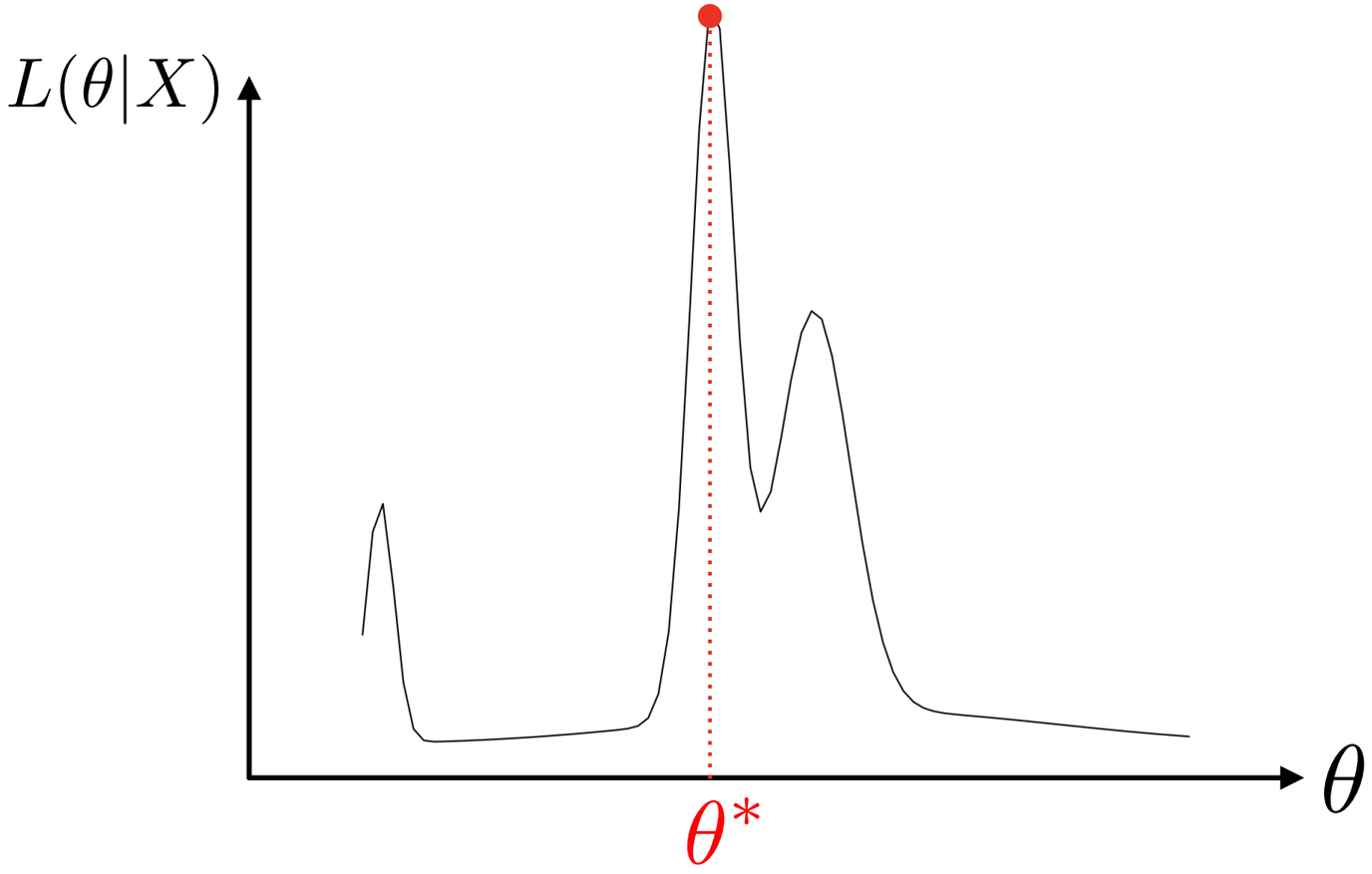


Example: maximum likelihood

θ — Parameter of interest ← Tree

X — Data ← Sequences

$L(\theta|X)$ — Likelihood: function of parameter given data ← Substitution Model

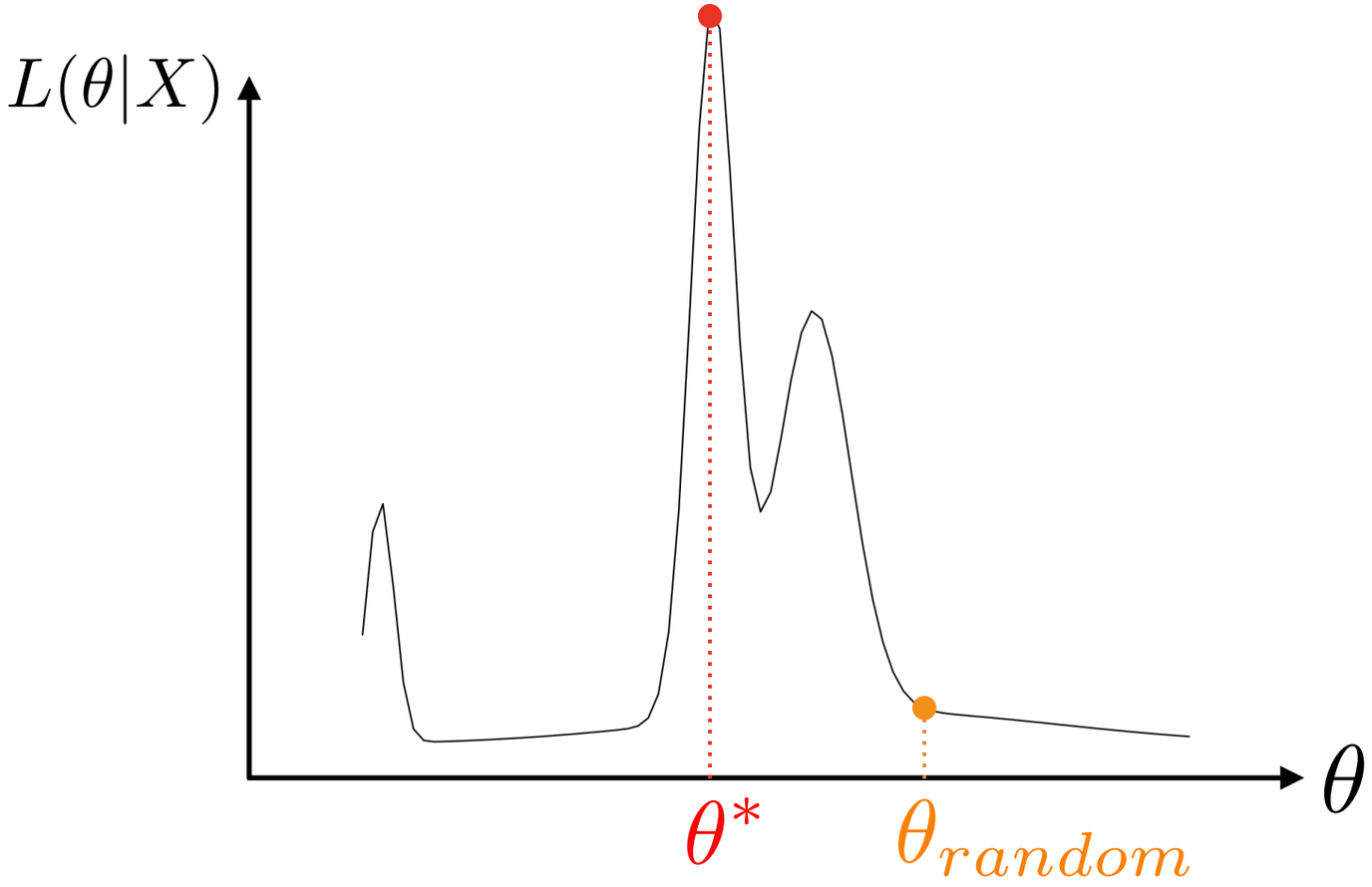


Example: maximum likelihood

θ — Parameter of interest ← Tree

X — Data ← Sequences

$L(\theta|X)$ — Likelihood: function of parameter given data ← Substitution Model

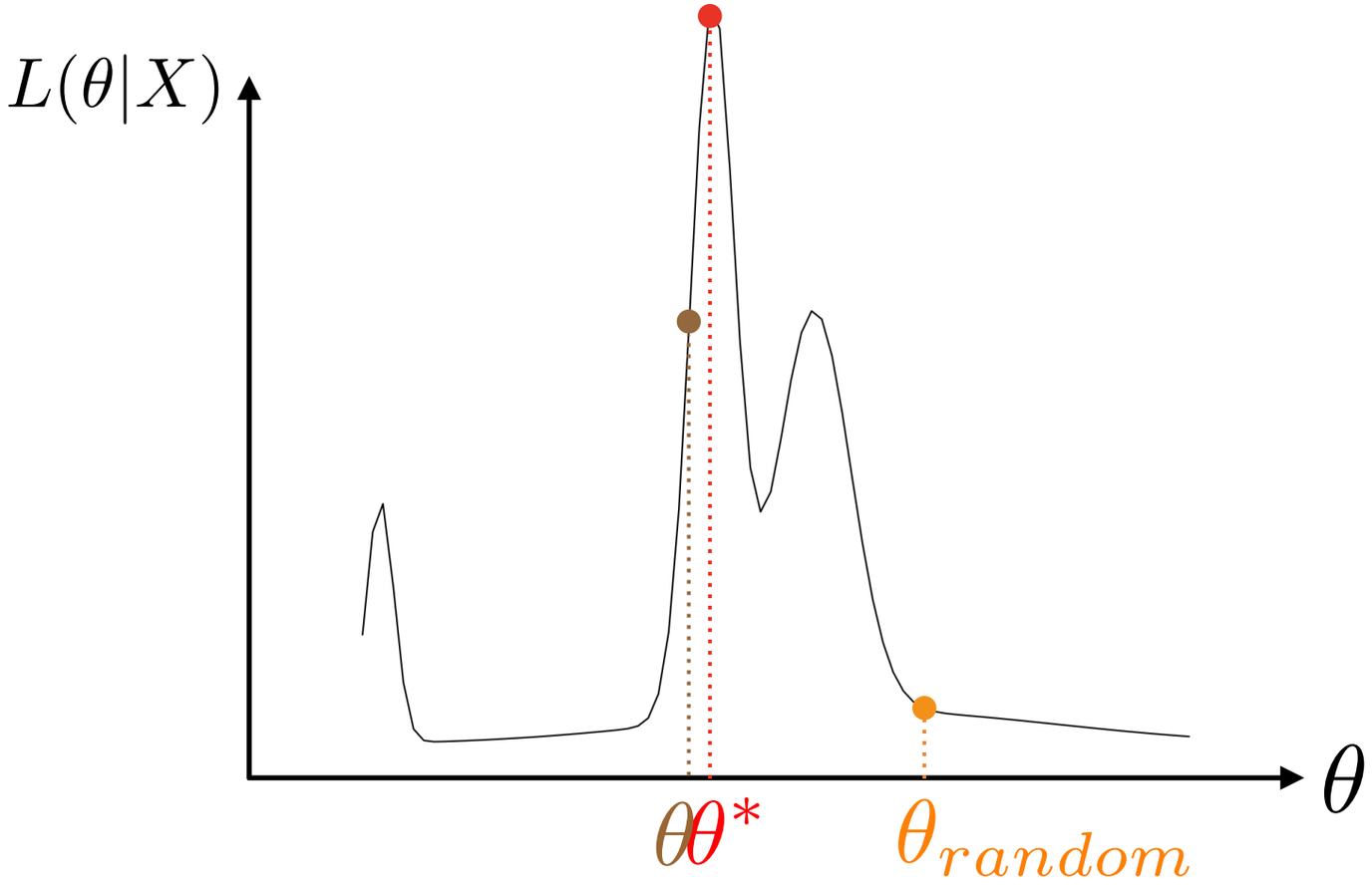


Example: maximum likelihood

θ — Parameter of interest ← Tree

X — Data ← Sequences

$L(\theta|X)$ — Likelihood: function of parameter given data ← Substitution Model

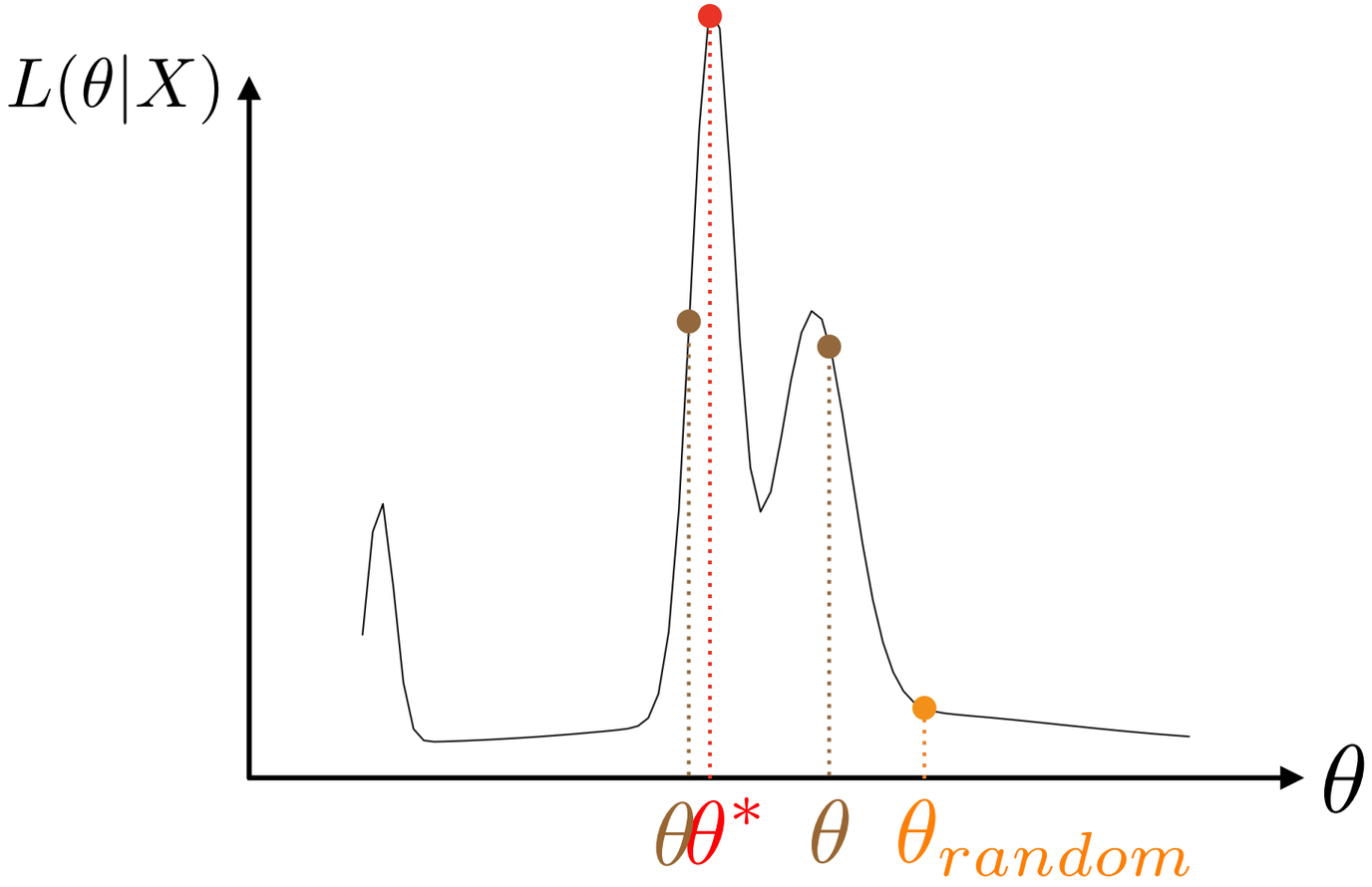


Example: maximum likelihood

θ — Parameter of interest ← Tree

X — Data ← Sequences

$L(\theta|X)$ — Likelihood: function of parameter given data ← Substitution Model

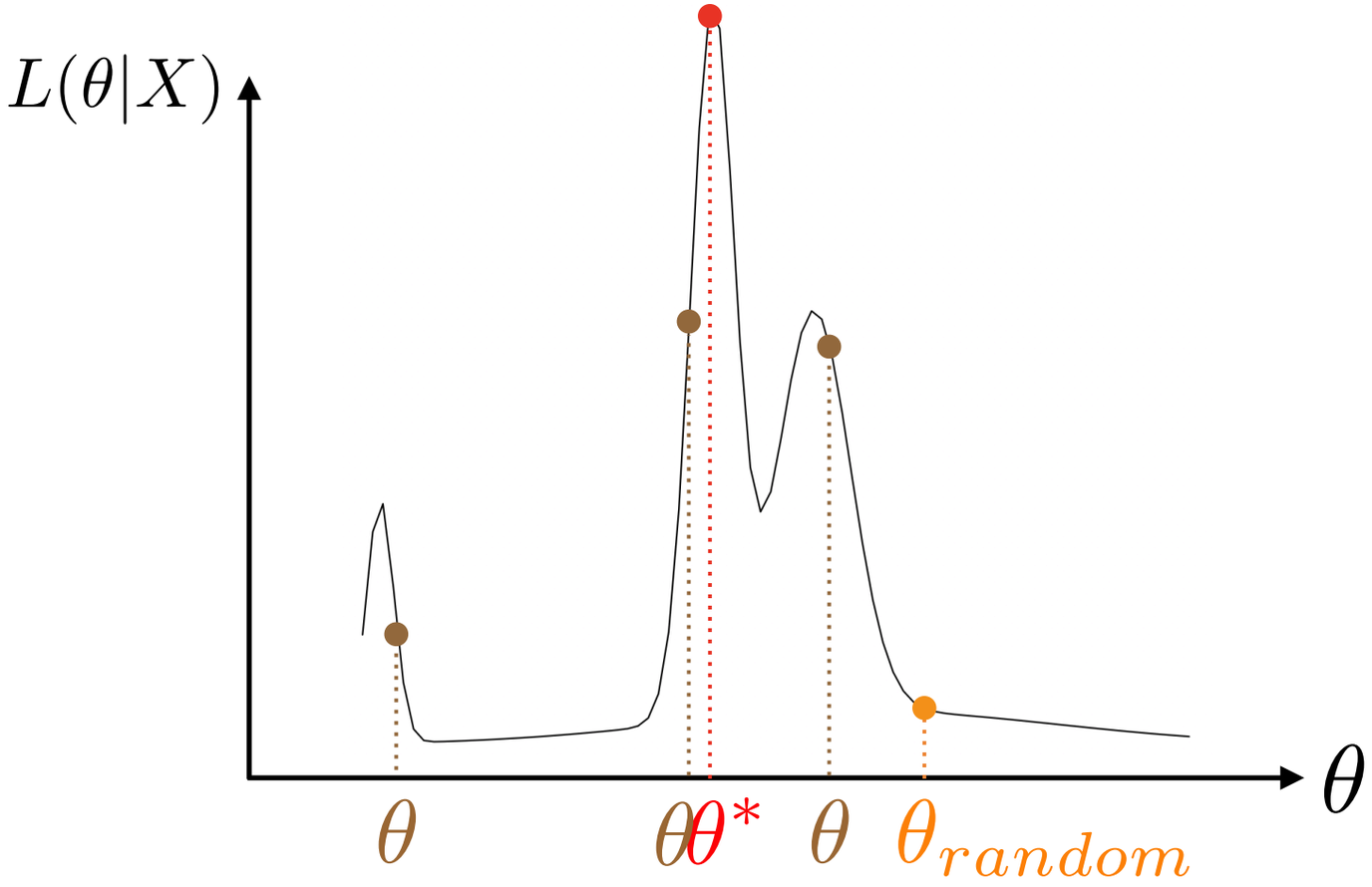


Example: maximum likelihood

θ — Parameter of interest ← Tree

X — Data ← Sequences

$L(\theta|X)$ — Likelihood: function of parameter given data ← Substitution Model

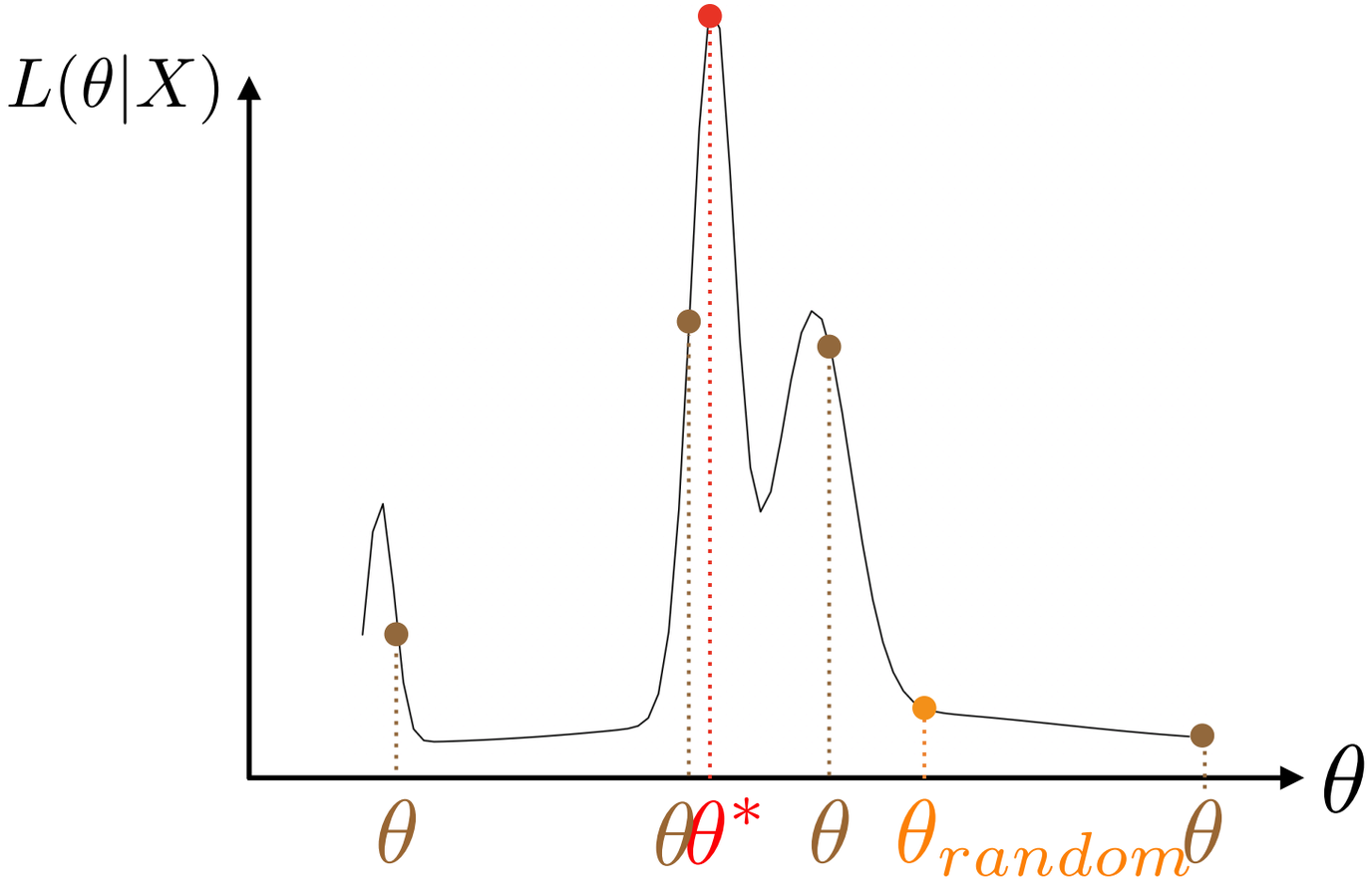


Example: maximum likelihood

θ — Parameter of interest ← Tree

X — Data ← Sequences

$L(\theta|X)$ — Likelihood: function of parameter given data ← Substitution Model

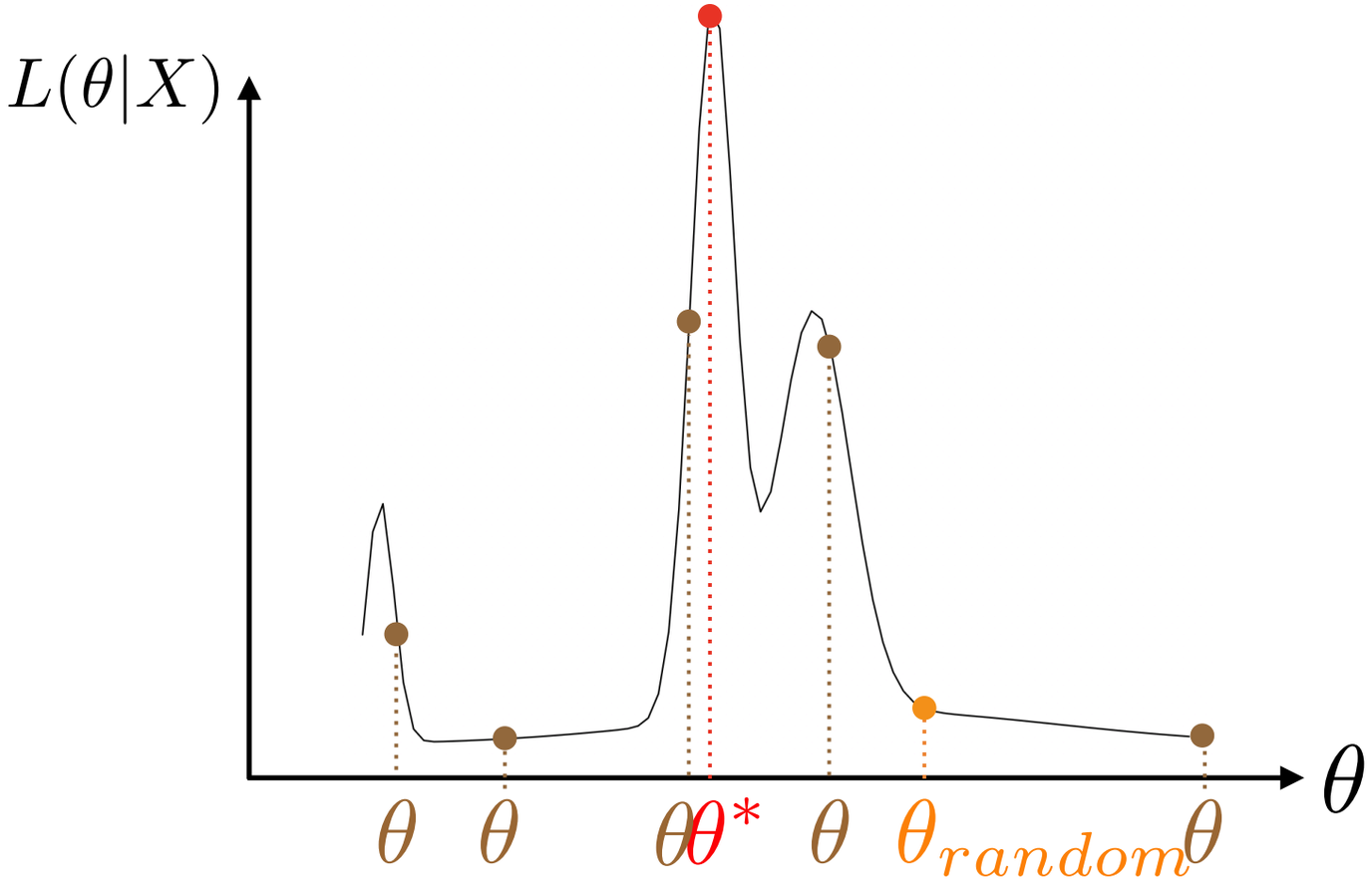


Example: maximum likelihood

θ — Parameter of interest ← Tree

X — Data ← Sequences

$L(\theta|X)$ — Likelihood: function of parameter given data ← Substitution Model

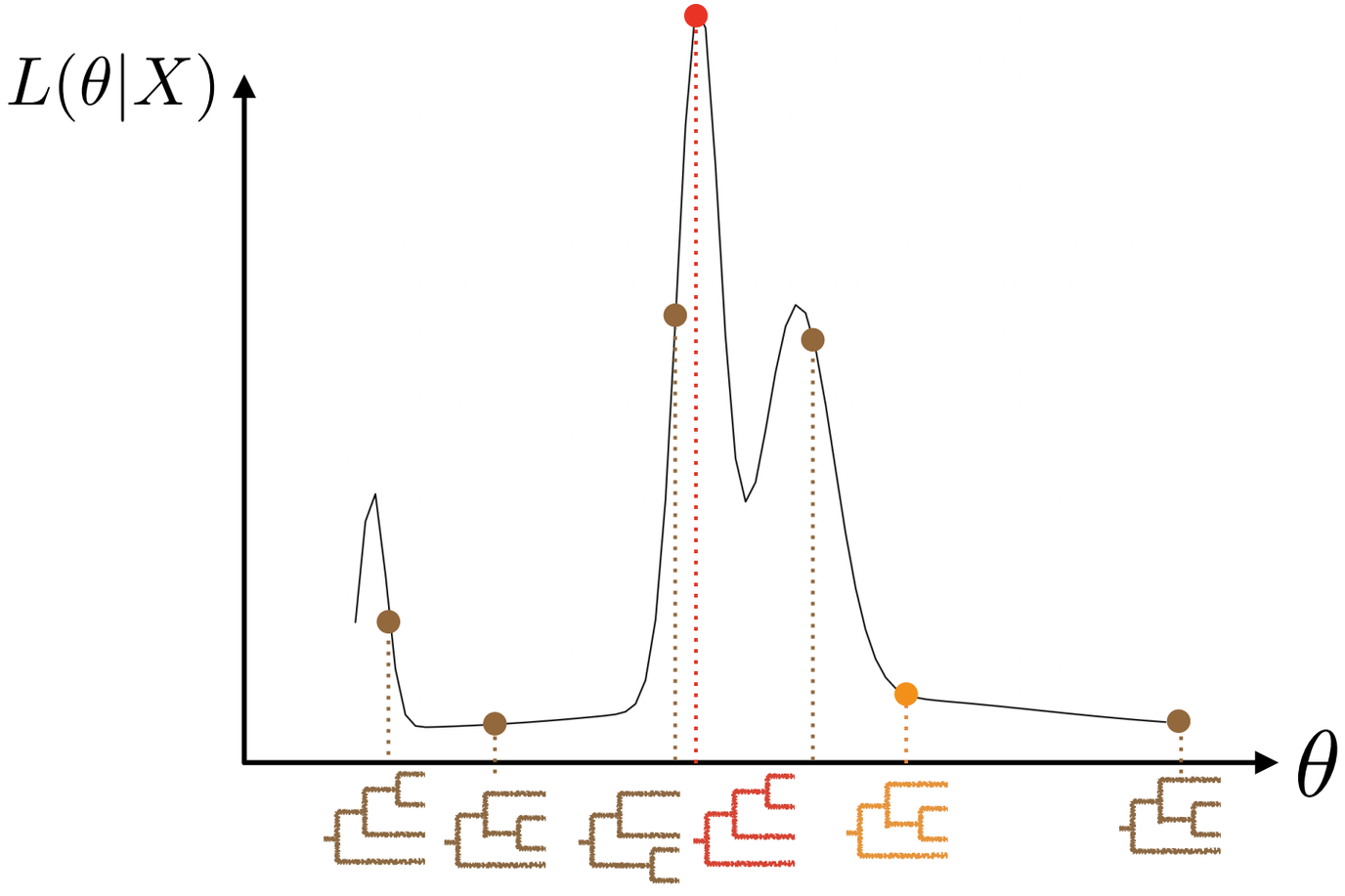


Recap: maximum likelihood

θ — Parameter of interest ← Tree

X — Data ← Sequences

$L(\theta|X)$ — Likelihood: function of parameter given data ← Substitution Model

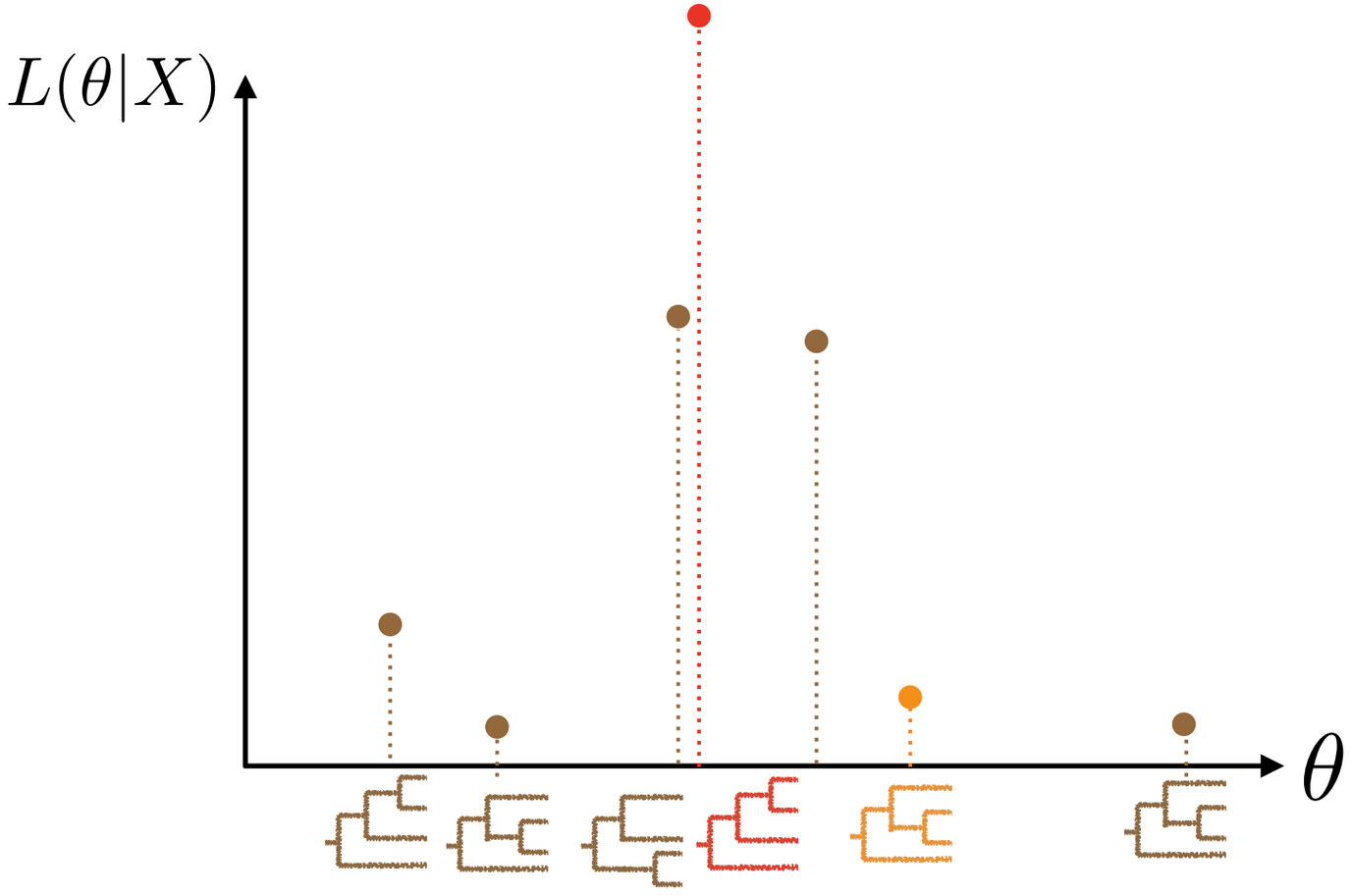


Recap: maximum likelihood

θ — Parameter of interest ← Tree

X — Data ← Sequences

$L(\theta|X)$ — Likelihood: function of parameter given data ← Substitution Model



Maximum likelihood recipe

Input: DNA sequences
Output: ML tree

1. Choose random starting tree: T_0
2. Calculate the likelihood of T_0 given the data: $L(T_0|X)$
3. Propose new tree T_1
4. Calculate the likelihood of new tree: $L(T_1|X)$
5. If $L(T_1|X) > L(T_0|X)$, keep T_1; otherwise, keep T_0
6. Repeat

Maximum parsimony recipe

Input: DNA sequences
Output: MP tree

1. Choose random starting tree: T_0
2. Calculate the parsimony of T_0 given the data: $L(T_0|X)$
3. Propose new tree T_1
4. Calculate the parsimony of new tree:$L(T_1|X)$
5. If $L(T_1|X) > L(T_0|X)$, keep T_1; otherwise, keep T_0
6. Repeat

distance

Minimum evolution recipe

Input: DNA sequences
Output: ME tree

1. Choose random starting tree: T_0
2. Calculate the length of T_0 given the data:
3. Propose new tree $L(T_0|X)$
4. Calculate the length of new tree: T_1
5. If $L(T_1|X) < L(T_0|X)$, keep T_1 ; $L(T_1|X)$ otherwise, keep T_0
6. Repeat

Phylogenetic inference

Step 1: Choose the criterion to use: distances, parsimony, likelihood

Step 2: Search the space of trees until you find the optimum

Phylogenetic inference

Step 1: Choose the criterion to use: distances, parsimony, likelihood

Step 2: Search the space of trees until you find the optimum

Weaknesses?

θ

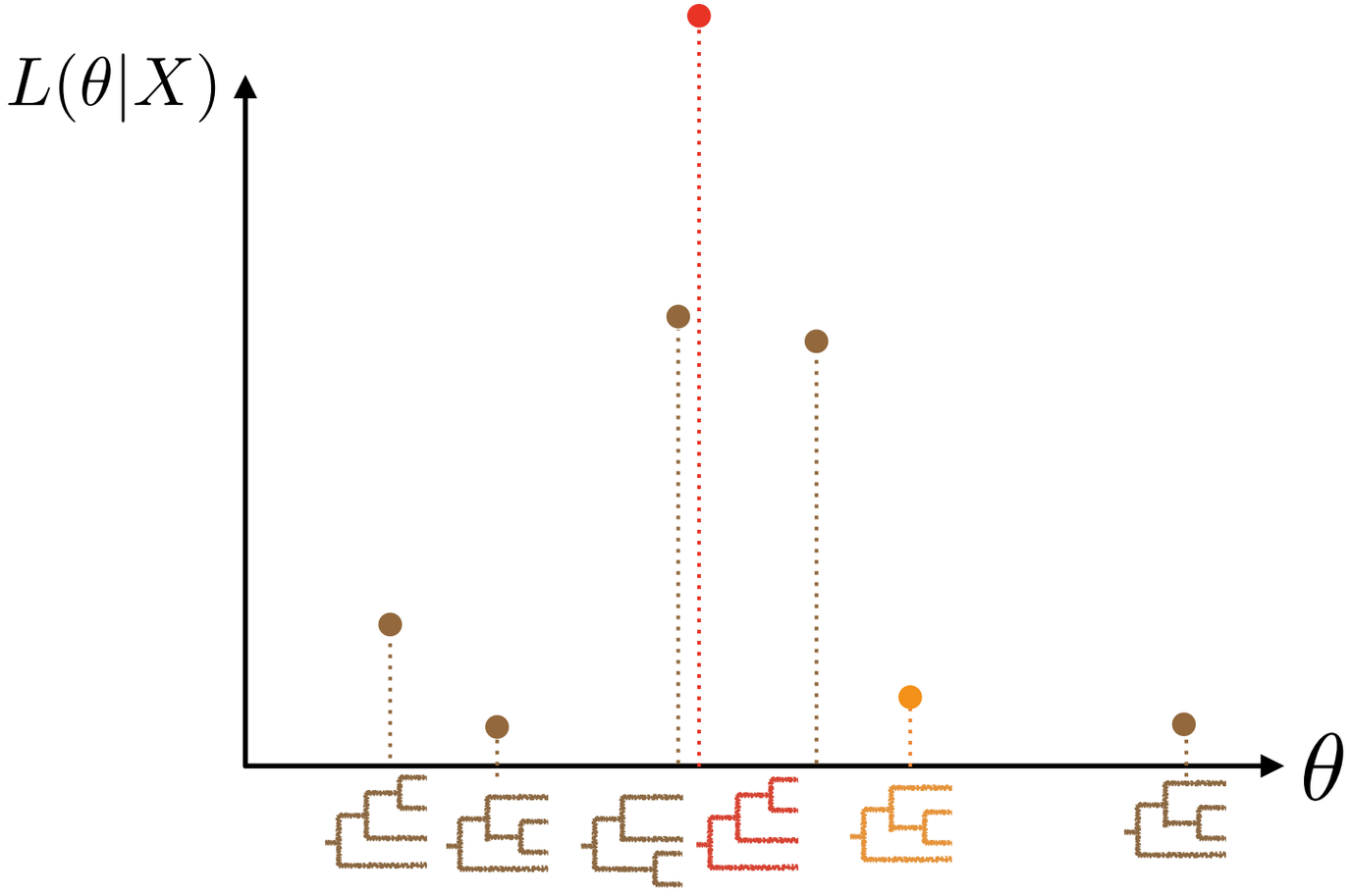
Parameter of interest ← Tree

X

Data ← Sequences

$L(\theta|X)$

Likelihood: function of parameter given data ← Substitution Model



θ

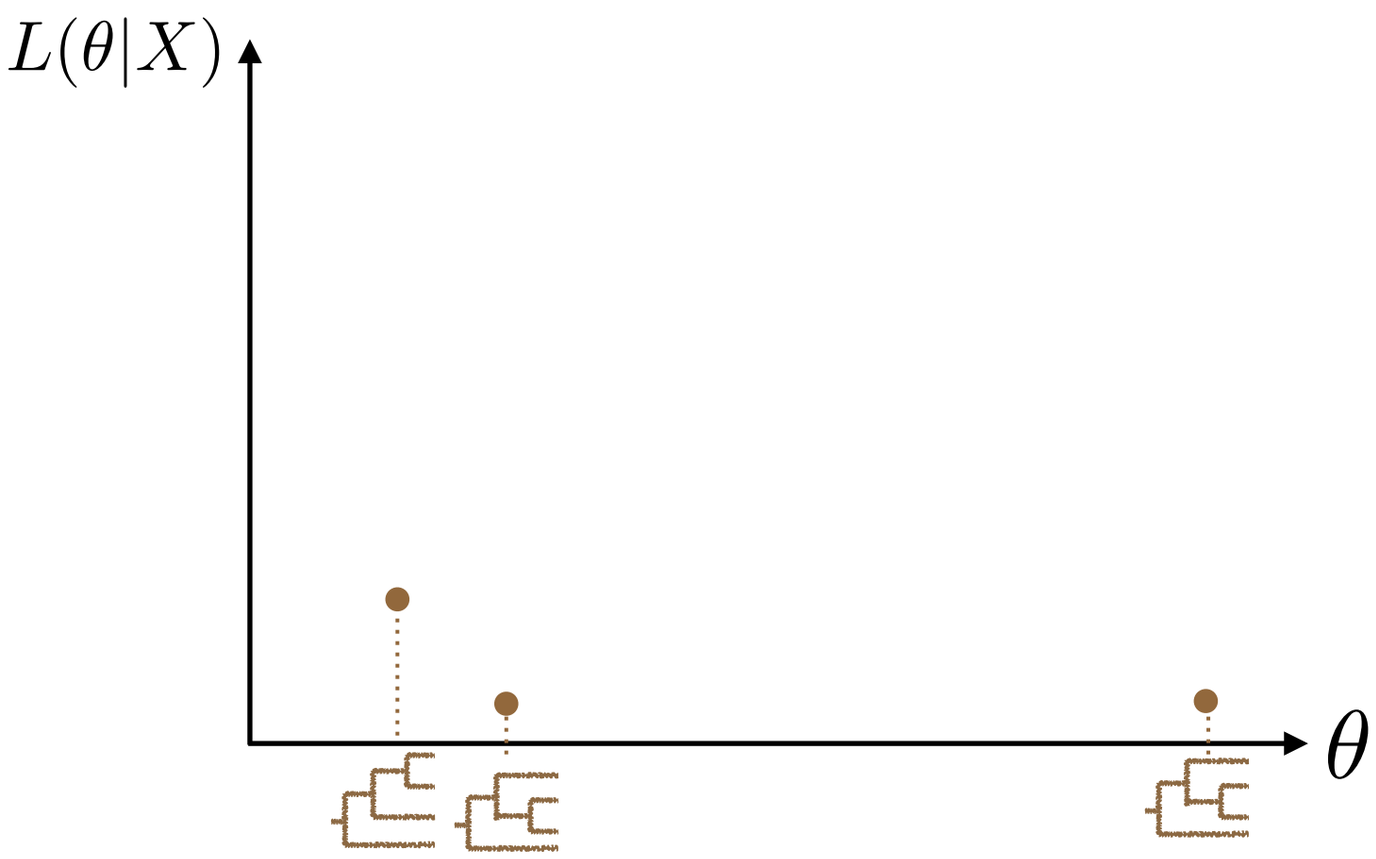
Parameter of interest ← Tree

X

Data ← Sequences

$L(\theta|X)$

Likelihood: function of parameter given data ← Substitution Model



θ

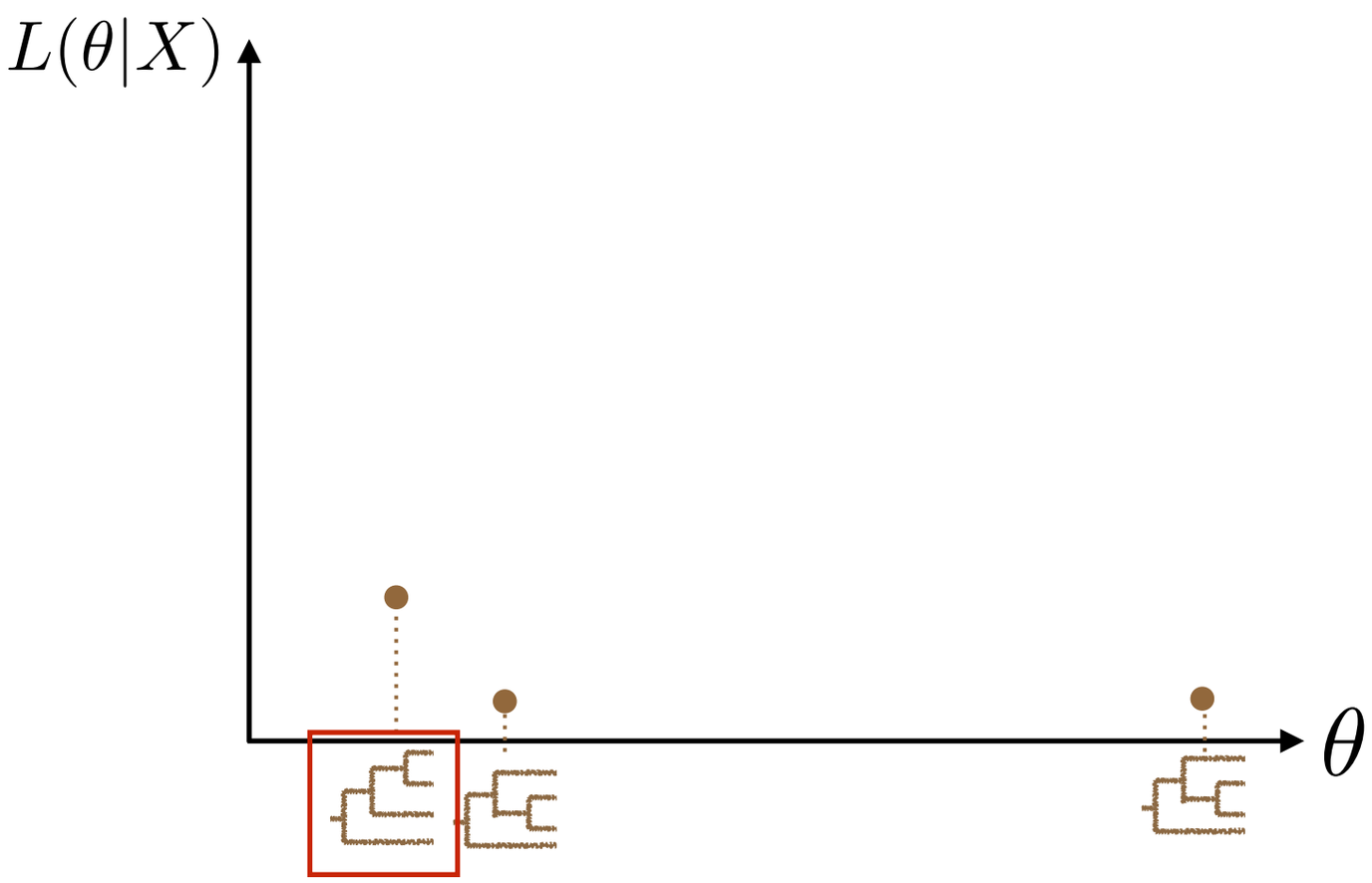
Parameter of interest ← Tree

X

Data ← Sequences

$L(\theta|X)$

Likelihood: function of parameter given data ← Substitution Model



# Species	# Unrooted trees	# Rooted trees
1	1	1
2	1	1
3	1	3
4	3	15
5	15	105
6	105	945
7	945	10395
8	10,395	135,135
9	135,135	2,027,025
10	2,027,025	34,459,425
11	34,459,425	654,729,075
12	654,729,075	13,749,310,575
13	13,749,310,575	316,234,143,225
⋮	⋮	⋮
52	$>$ # atoms in universe	

θ

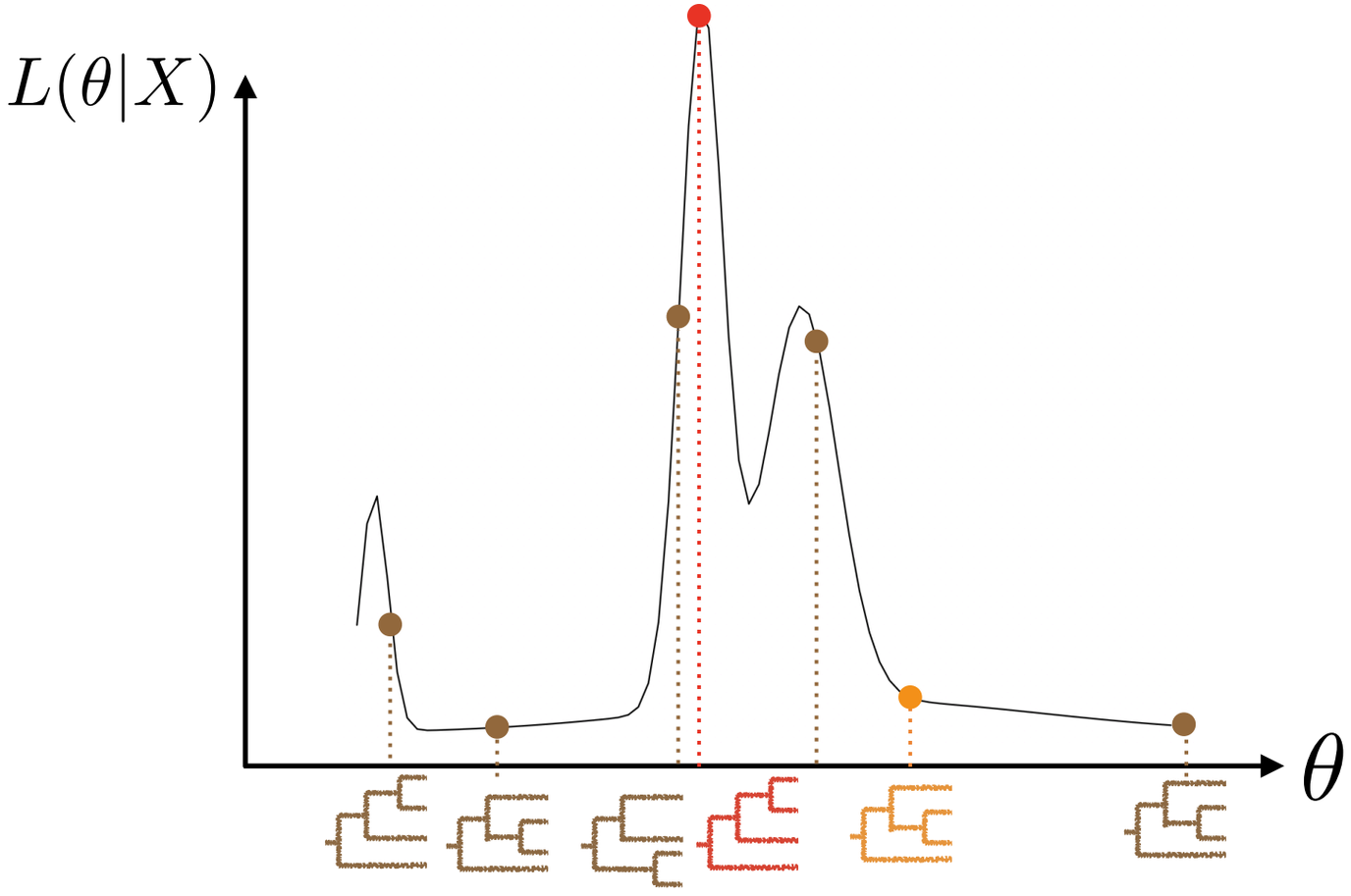
Parameter of interest ← Tree

X

Data ← Sequences

$L(\theta|X)$

Likelihood: function of parameter given data ← Substitution Model



θ

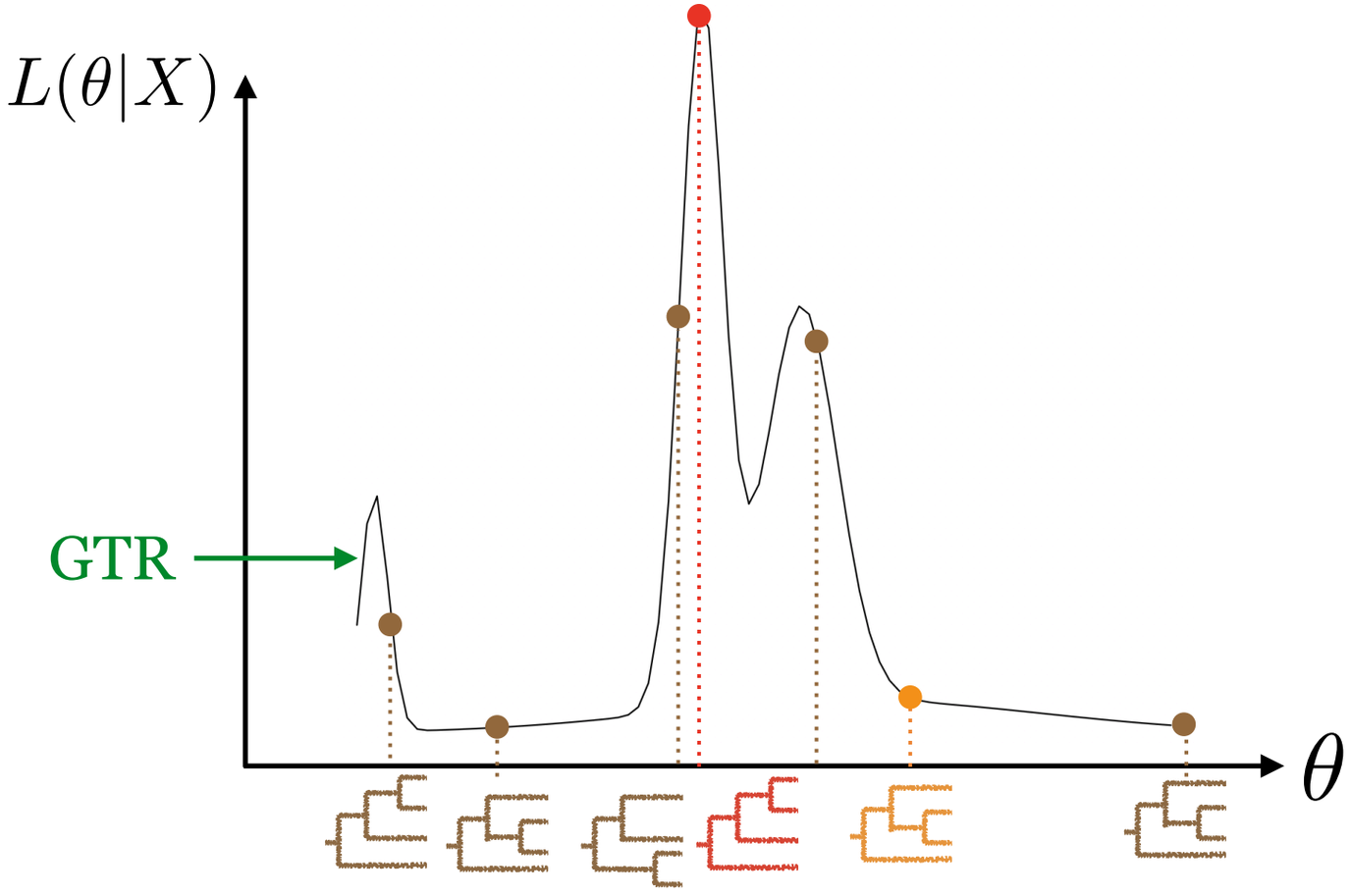
Parameter of interest ← Tree

X

Data ← Sequences

$L(\theta|X)$

Likelihood: function of parameter given data ← Substitution Model



θ

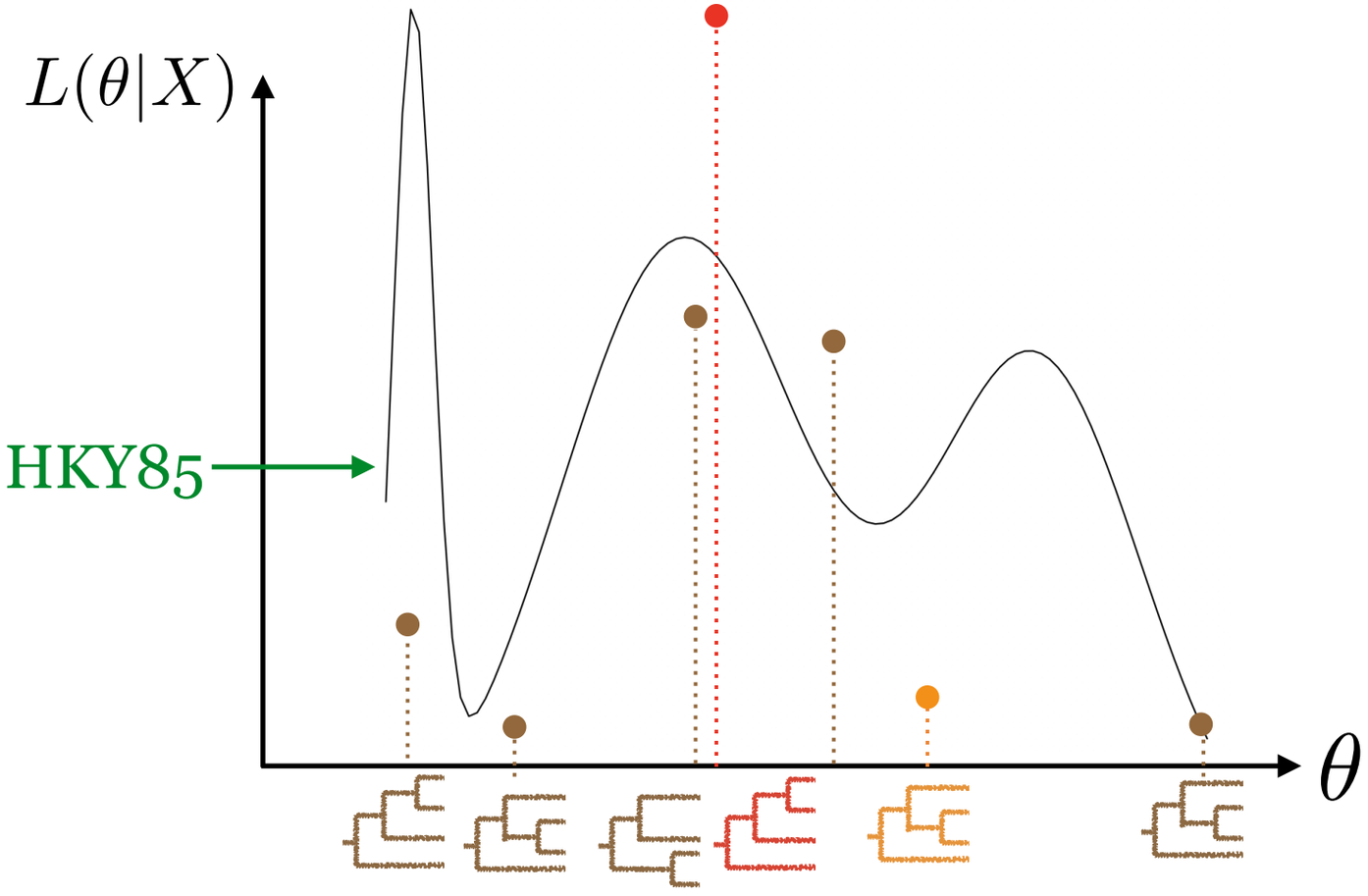
X

$L(\theta|X)$

Parameter of interest ← Tree

Data ← Sequences

Likelihood: function of parameter given data ← Substitution Model



θ

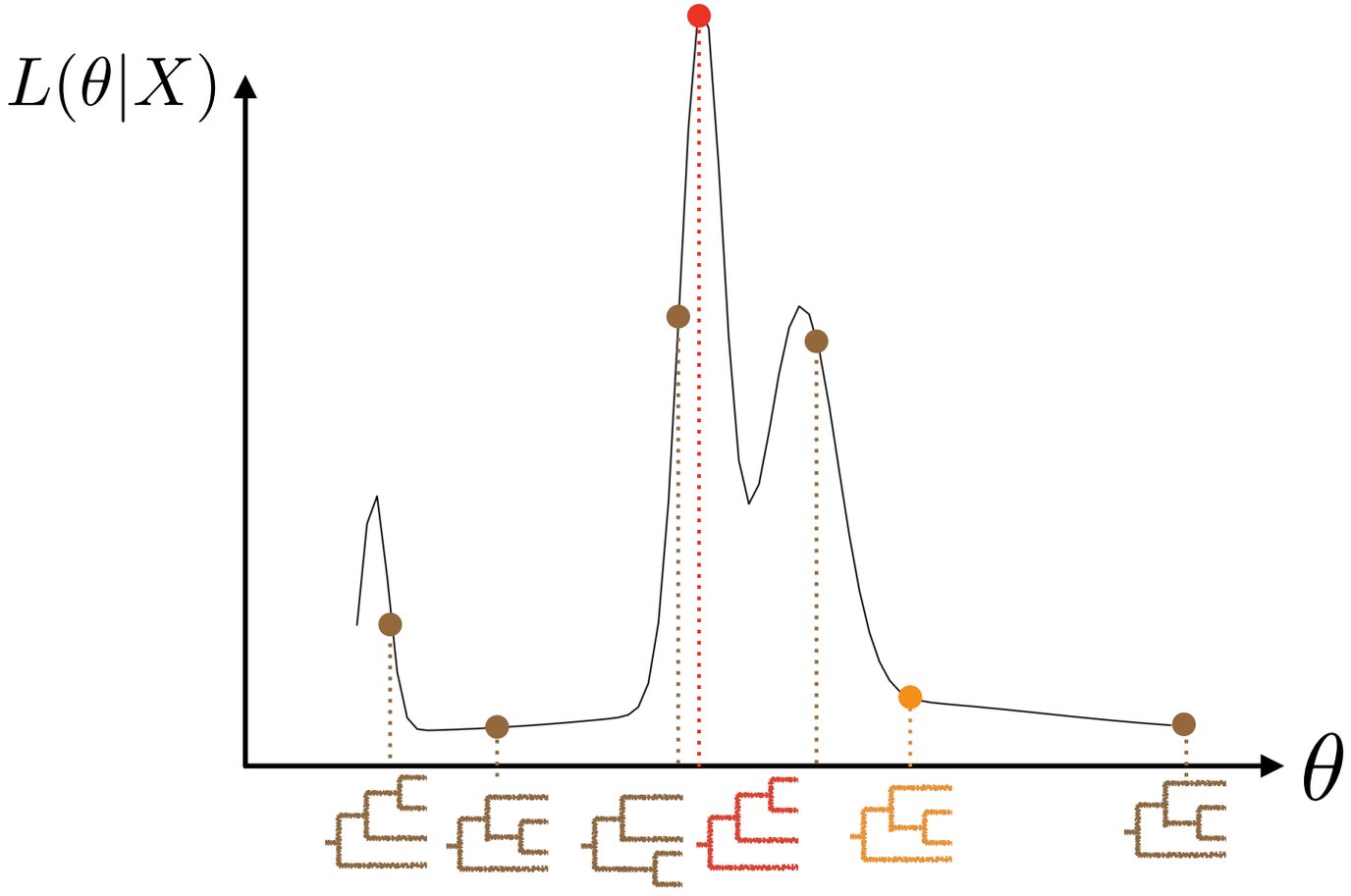
Parameter of interest ← Tree

X

Data ← Sequences

$L(\theta|X)$

Likelihood: function of parameter given data ← Substitution Model



θ

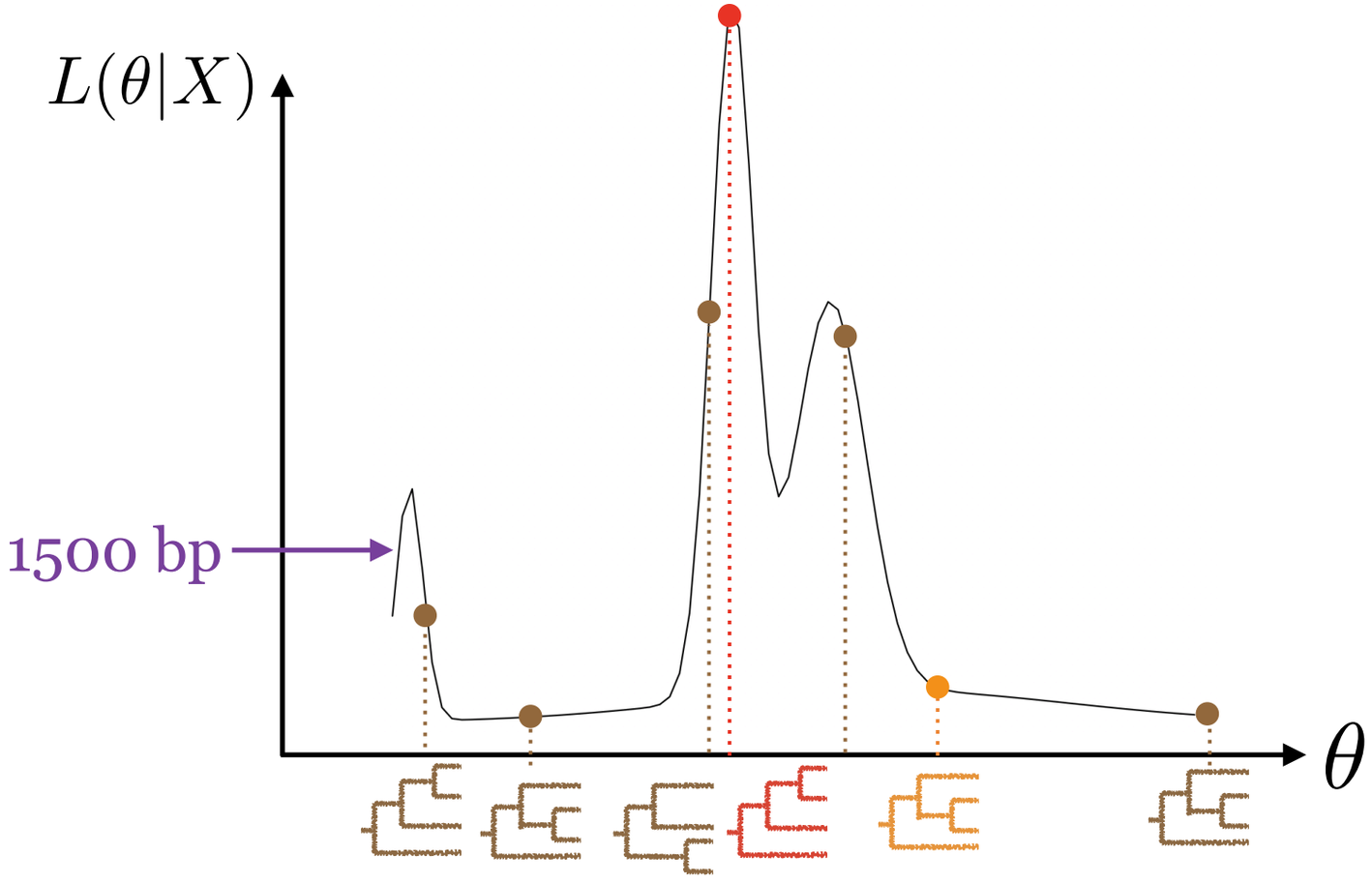
X

$L(\theta|X)$

Parameter of interest ← Tree

Data ← Sequences

Likelihood: function of parameter given data ← Substitution Model



θ

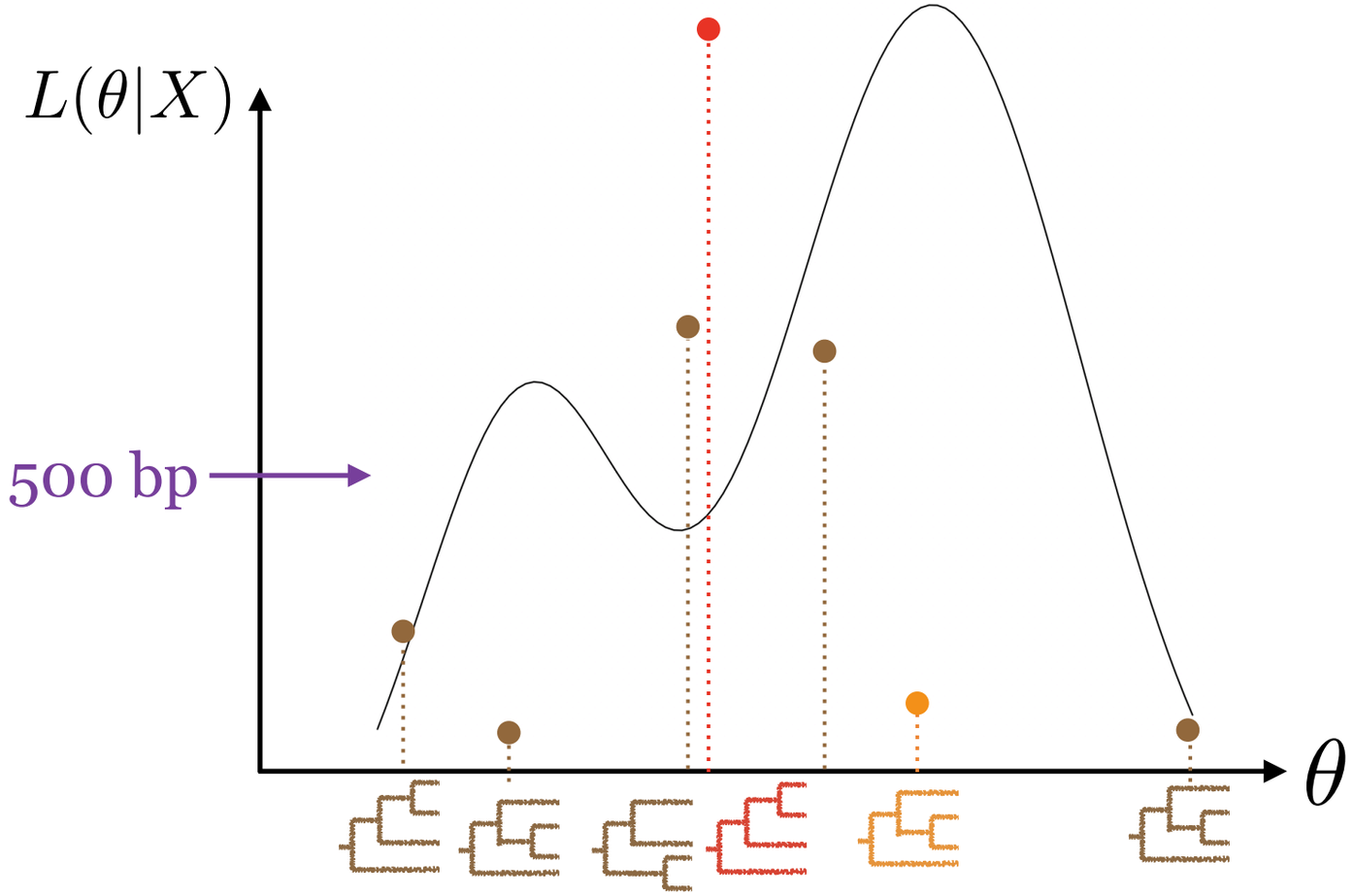
Parameter of interest ← Tree

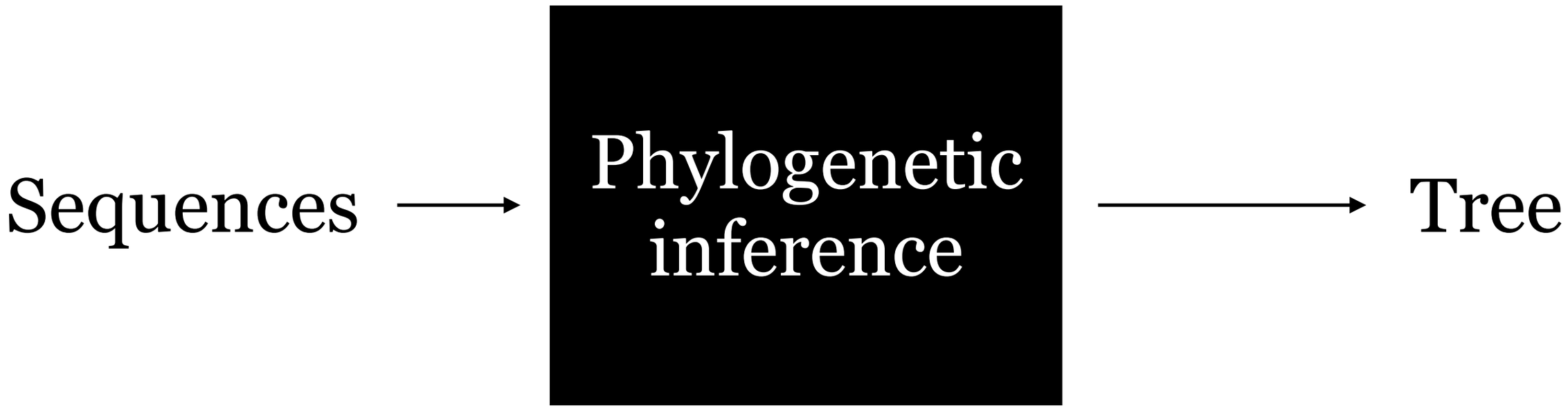
X

Data ← Sequences

$L(\theta|X)$

Likelihood: function of parameter given data ← Substitution Model

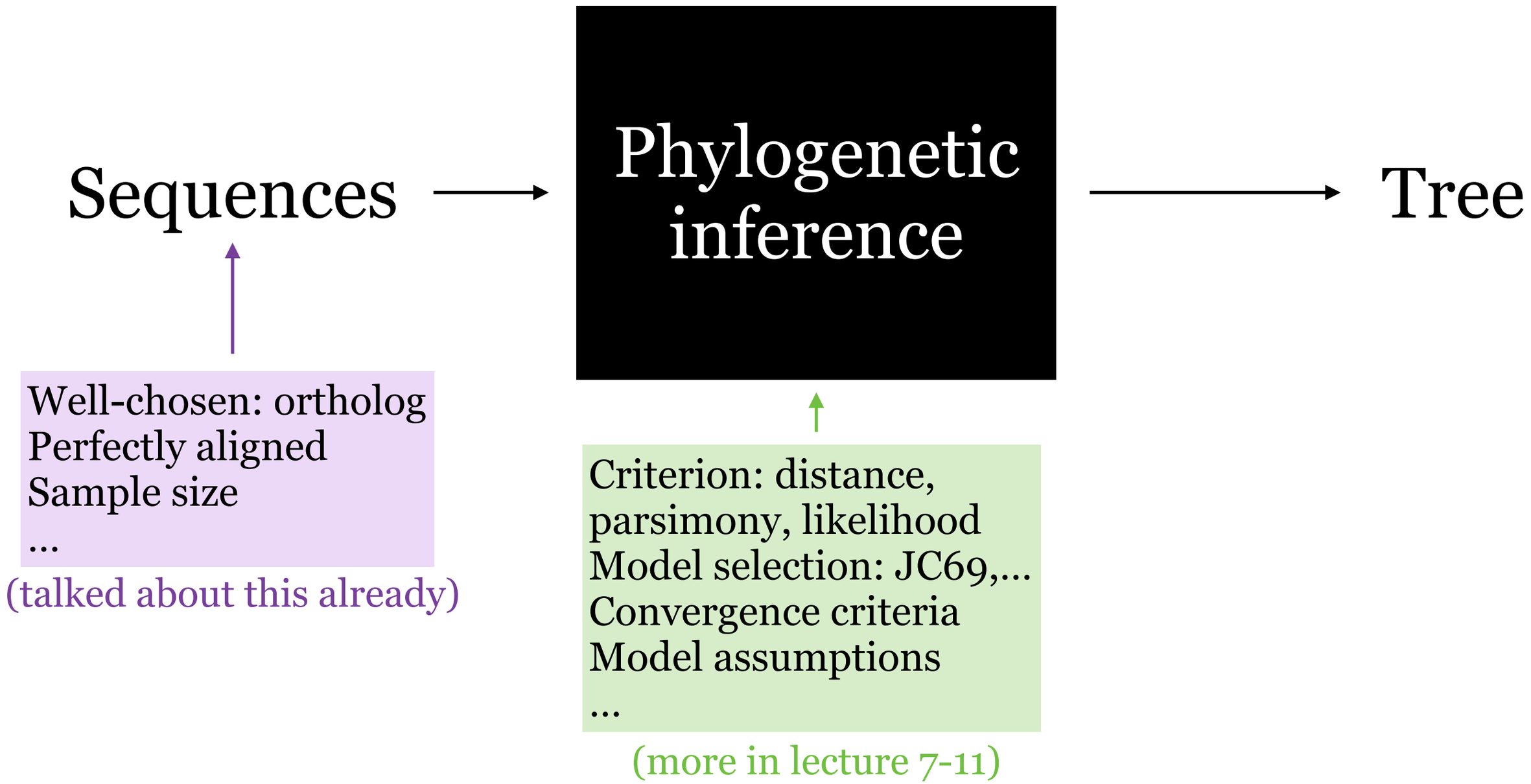



Sequences
Phylogenetic inference
Tree

Sequences → Phylogenetic inference → Tree

Well-chosen: ortholog
Perfectly aligned
Sample size
…

(talked about this already)


Sequences
Phylogenetic inference
Tree
Well-chosen: ortholog
Perfectly aligned
Sample size
…
(talked about this already)
Criterion: distance, parsimony, likelihood
Model selection: JC69,…
Convergence criteria
Model assumptions
…
(more in lecture 7-11)

Sequences

Well-chosen: ortholog
Perfectly aligned
Sample size
…

(talked about this already)

Phylogenetic inference

Criterion: distance, parsimony, likelihood
Model selection: JC69,…
Convergence criteria
Model assumptions
…

(more in lecture 7-11)

Tree

Measures of confidence

(more in lecture 14)

RAxML

(Stamatakis, 2006)

Summary: RAxML-VI-HPC (randomized axelerated maximum likelihood for high performance computing) is a sequential and parallel program for inference of large phylogenies with maximum likelihood (ML). Low-level technical optimizations, a modification of the search algorithm, and the use of the GTR+CAT approximation as replacement for GTR+Γ yield a program that is between 2.7 and 52 times faster than the previous version of RAxML. A large-scale performance comparison with GARLI, PHYML, IQPNNI and MrBayes on real data containing 1000 up to 6722 taxa shows that RAxML requires at least 5.6 times less main memory and yields better trees in similar times than the best competing program (GARLI) on datasets up to 2500 taxa. On datasets ≥4000 taxa it also runs 2–3 times faster than GARLI. RAxML has been parallelized with MPI to conduct parallel multiple bootstraps and inferences on distinct starting trees. The program has been used to compute ML trees on two of the largest alignments to date containing 25 057 (1463 bp) and 2182 (51 089 bp) taxa, respectively.

RAxML

(Stamatakis, 2006)

Summary: RAxML-VI-HPC (randomized axelerated maximum likelihood for high performance computing) is a sequential and parallel program for inference of large phylogenies with maximum likelihood (ML). Low-level technical optimizations, a modification of the search algorithm, and the use of the GTR+CAT approximation as replacement for GTR+Γ yield a program that is between 2.7 and 52 times faster than the previous version of RAxML. A large-scale performance comparison with GARLI, PHYML, IQPNNI and MrBayes on real data containing 1000 up to 6722 taxa shows that RAxML requires at least 5.6 times less main memory and yields better trees in similar times than the best competing program (GARLI) on datasets up to 2500 taxa. On datasets ≥4000 taxa it also runs 2–3 times faster than GARLI. RAxML has been parallelized with MPI to conduct parallel multiple bootstraps and inferences on distinct starting trees. The program has been used to compute ML trees on two of the largest alignments to date containing 25 057 (1463 bp) and 2182 (51 089 bp) taxa, respectively.

RAxML

(Stamatakis, 2006)

# Species	# Unrooted trees	# Rooted trees
1	1	1
2	1	1
3	1	3
4	3	15
5	15	105
6	105	945
7	945	10395
8	10,395	135,135
9	135,135	2,027,025
10	2,027,025	34,459,425
11	34,459,425	654,729,075
12	654,729,075	13,749,310,575
13	13,749,310,575	316,234,143,225
⋮	⋮	⋮
52	> # atoms in universe	

Summary: RAxML-VI-HPC (randomized axelerated maximum likelihood for high performance computing) is a sequential and parallel program for inference of large phylogenies with maximum likelihood (ML). Low-level technical optimizations, a modification of the search algorithm, and the use of the GTR+CAT approximation as replacement for GTR+Γ yield a program that is between 2.7 and 52 times faster than the previous version of RAxML. A large-scale performance comparison with GARLI, PHYML, IQPNNI and MrBayes on real data containing 1000 up to 6722 taxa shows that RAxML requires at least 5.6 times less main memory and yields better trees in similar times than the best competing program (GARLI) on datasets up to 2500 taxa. On datasets ≥4000 taxa it also runs 2–3 times faster than GARLI. RAxML has been parallelized with MPI to conduct parallel multiple bootstraps and inferences on distinct starting trees. The program has been used to compute ML trees on two of the largest alignments to date containing 25 057 (1463 bp) and 2182 (51 089 bp) taxa, respectively.

RAxML

(Stamatakis, 2006)

2 OPTIMIZATIONS OF RAxML

A detailed description of the optimizations listed below is provided in the on-line supplement. The main improvements cover:

- An efficient mechanism to store and re-store topologies and branch lengths via rearrangement descriptors.
- A consequent re-use of partial likelihood vectors.
- A dynamic adaptation of the rearrangement distance.
- Low-level optimization of the GTR+CAT and GTR+Γ likelihood functions.
- An efficient re-implementation of Maximum Parsimony starting tree computations.

RAxML

(Stamatakis, 2006)

2 OPTIMIZATIONS OF RAxML

A detailed description of the optimizations listed below is provided in the on-line supplement. The main improvements cover:

- An efficient mechanism to store and re-store topologies and branch lengths via rearrangement descriptors.
- A consequent re-use of partial likelihood vectors.
- A dynamic adaptation of the rearrangement distance.
- Low-level optimization of the GTR+CAT and GTR+Γ likelihood functions.
- An efficient re-implementation of Maximum Parsimony starting tree computations.

RAxML

(Stamatakis, 2006)

2 OPTIMIZATIONS OF RAxML

A detailed description of the optimizations listed below is provided in the on-line supplement. The main improvements cover:

- An efficient mechanism to store and re-store topologies and branch lengths via rearrangement descriptors.
- A consequent re-use of partial likelihood vectors.
- A dynamic adaptation of the rearrangement distance.
- Low-level optimization of the GTR+CAT and GTR+Γ likelihood functions.
- An efficient re-implementation of Maximum Parsimony starting tree computations.

The datasets are described and available for public download (if permission has been granted by the authors) at diwww.epfl.ch/~stamatak (material frame). For each dataset 5 randomized MP starting trees have been computed with RAxML-VI-HPC.

RAxML

(Stamatakis, 2006)

2 OPTIMIZATIONS OF RAxML

A detailed description of the optimizations listed below is provided in the on-line supplement. The main improvements cover:

- An efficient mechanism to store and re-store topologies and branch lengths via rearrangement descriptors.
- A consequent re-use of partial likelihood vectors.
- A dynamic adaptation of the rearrangement distance.
- Low-level optimization of the GTR+CAT and GTR+Γ likelihood functions.
- An efficient re-implementation of Maximum Parsimony starting tree computations.

The datasets are described and available for public download (if permission has been granted by the authors) at diwww.epfl.ch/~stamatak (material frame). For each dataset 5 randomized MP starting trees have been computed with RAxML-VI-HPC.

RAxML

(Stamatakis, 2006)

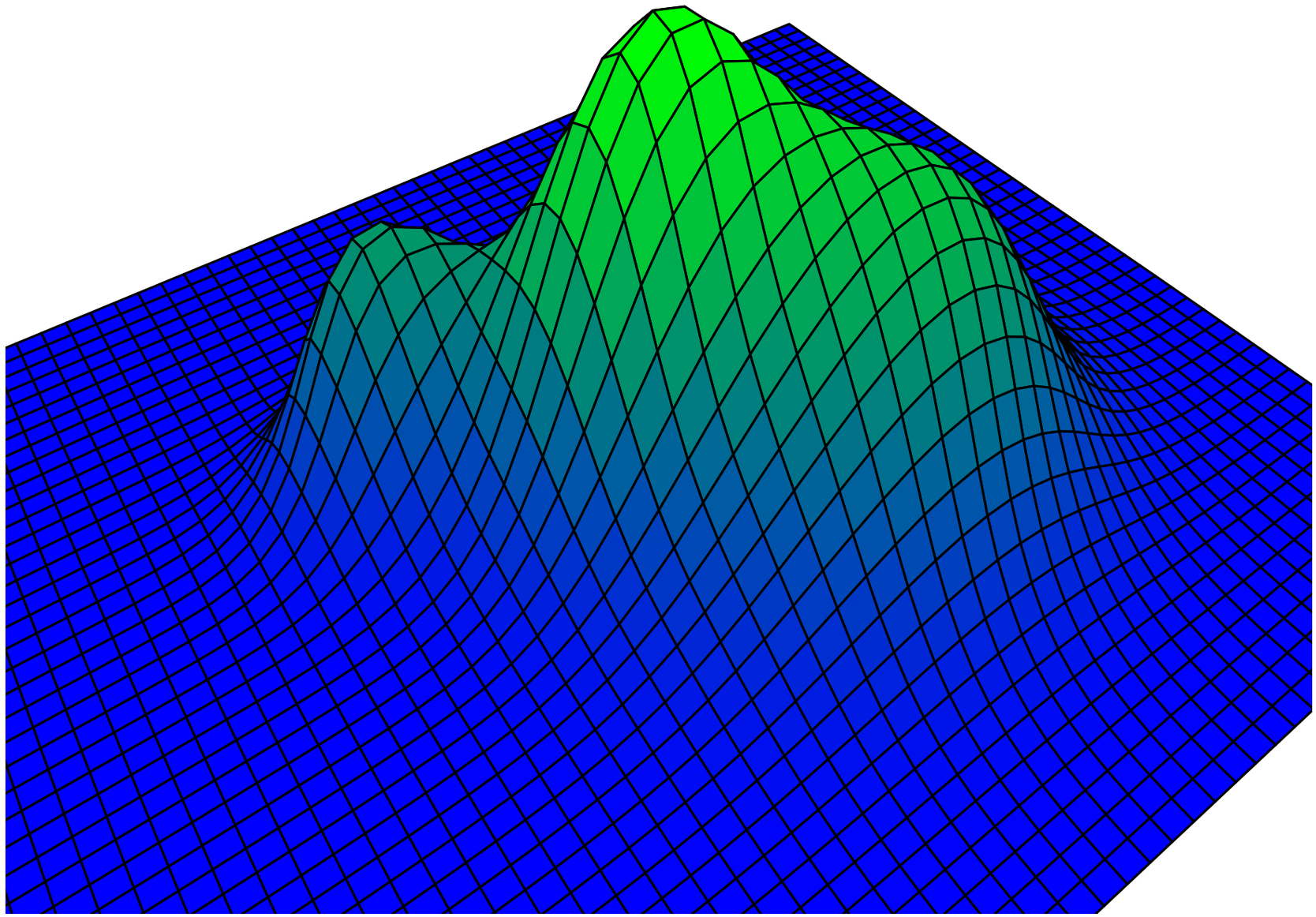
Table 1. Alignment lengths in bp and number of distinct patterns/columns

# Taxa	# bp	# patterns	# Taxa	# bp	# patterns
1000	5547	3364	1497	1241	1241
1663	1577	1576	1728	1276	1275
2000	1251	1251	2560	1232	1232
4114	1263	1263	6722	1122	1122
7769	851	851	8780	1217	1217

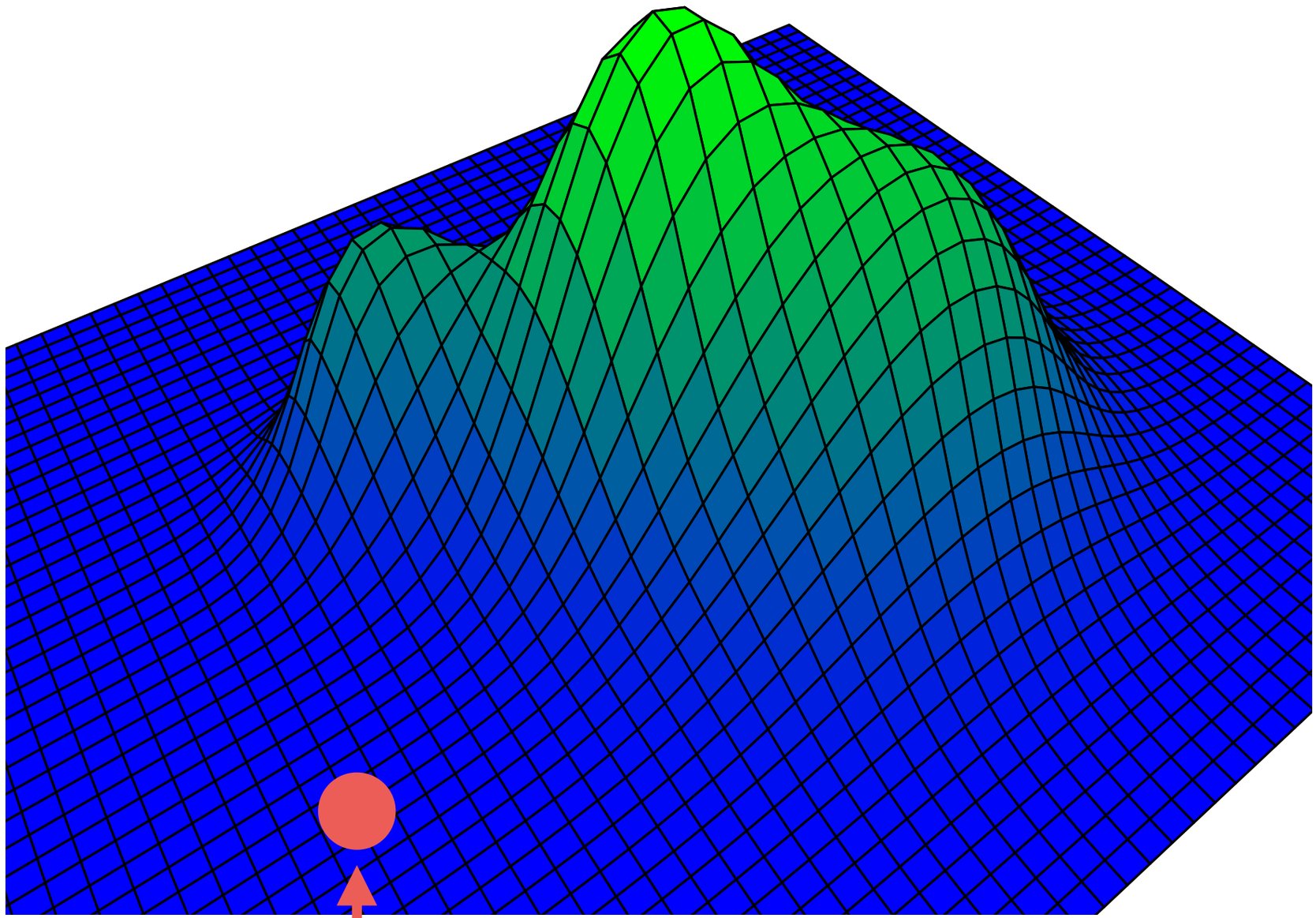
How do we navigate tree space?

Traverse tree space

Traverse tree space

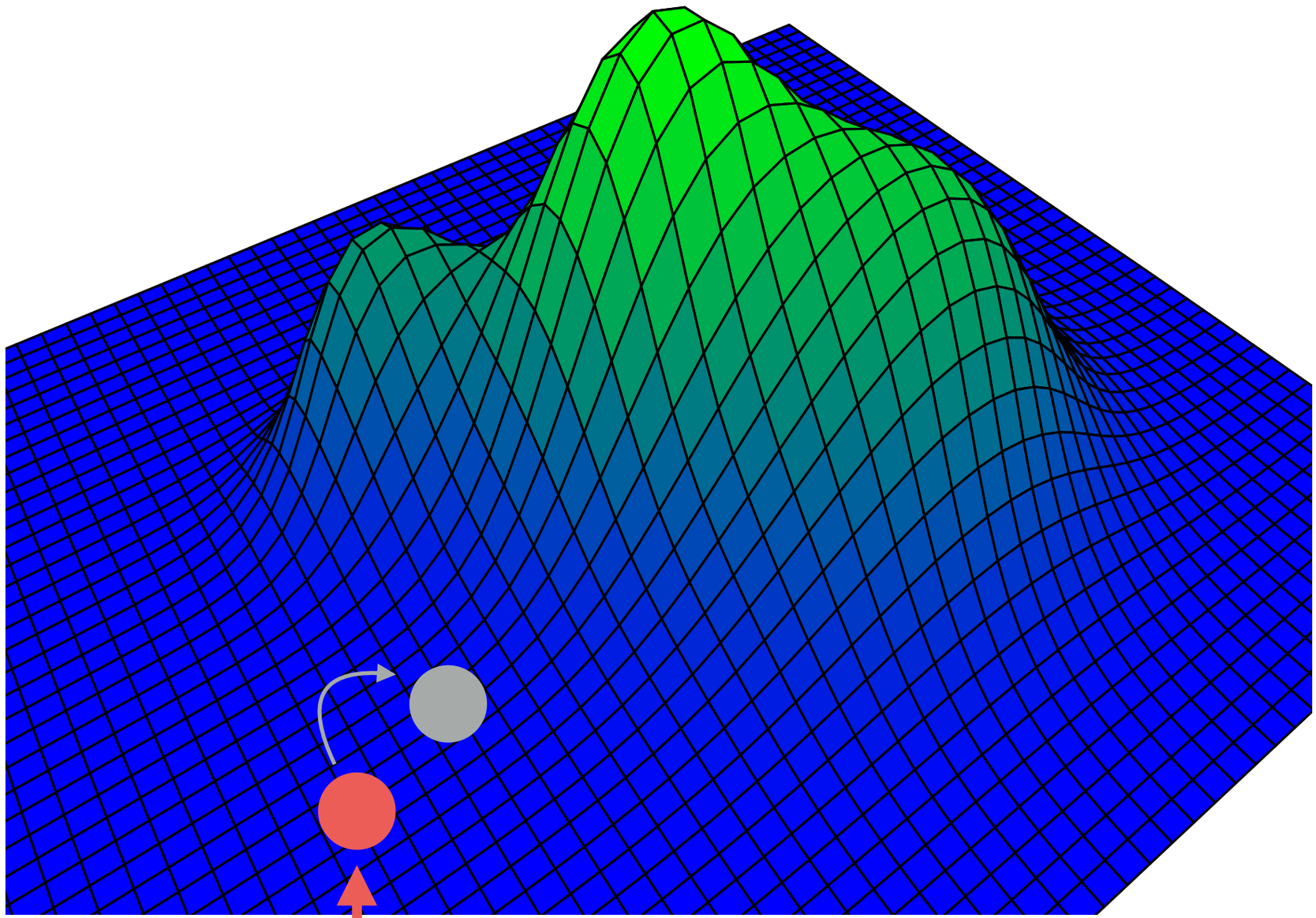


Traverse tree space



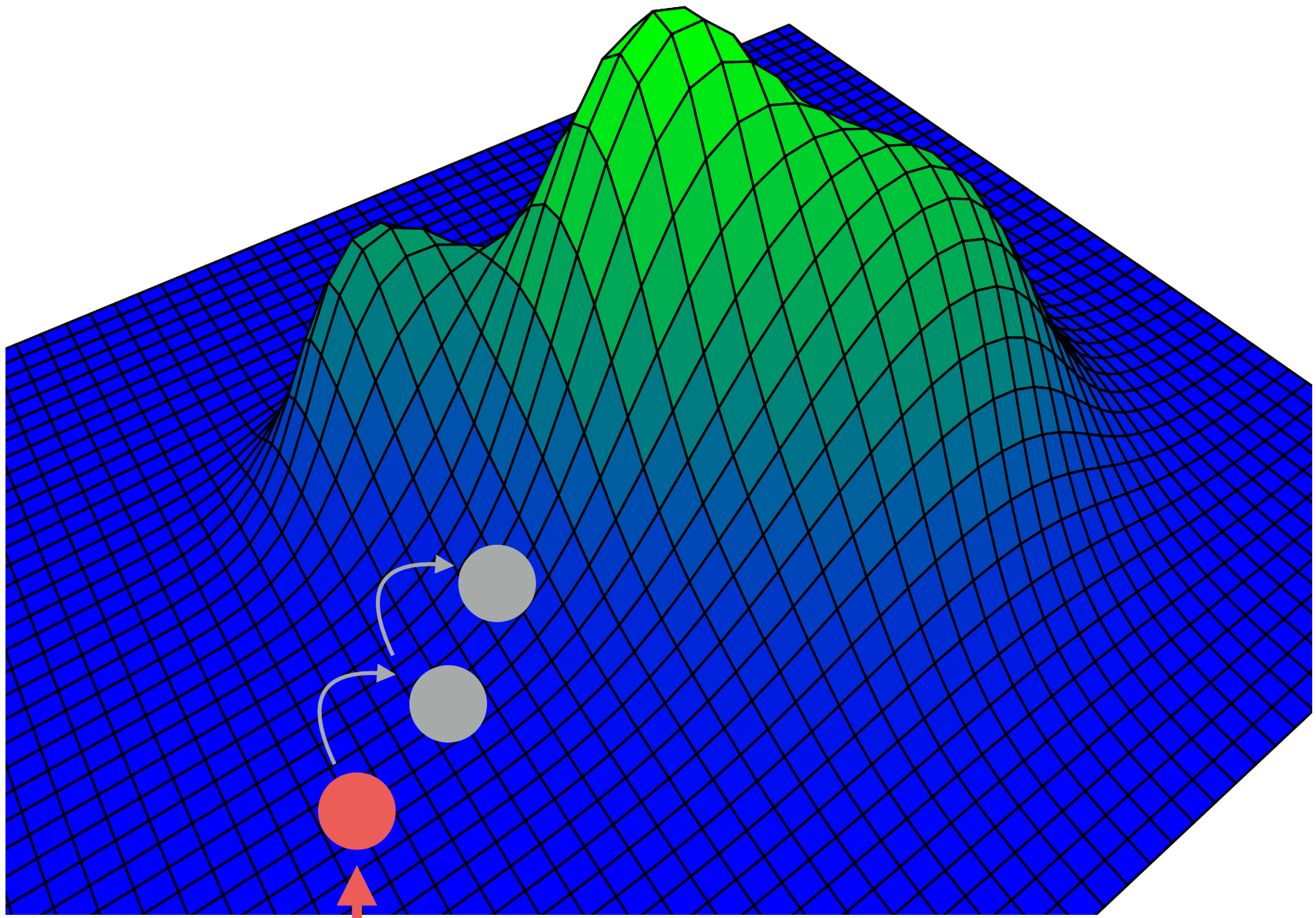
Starting tree

Traverse tree space



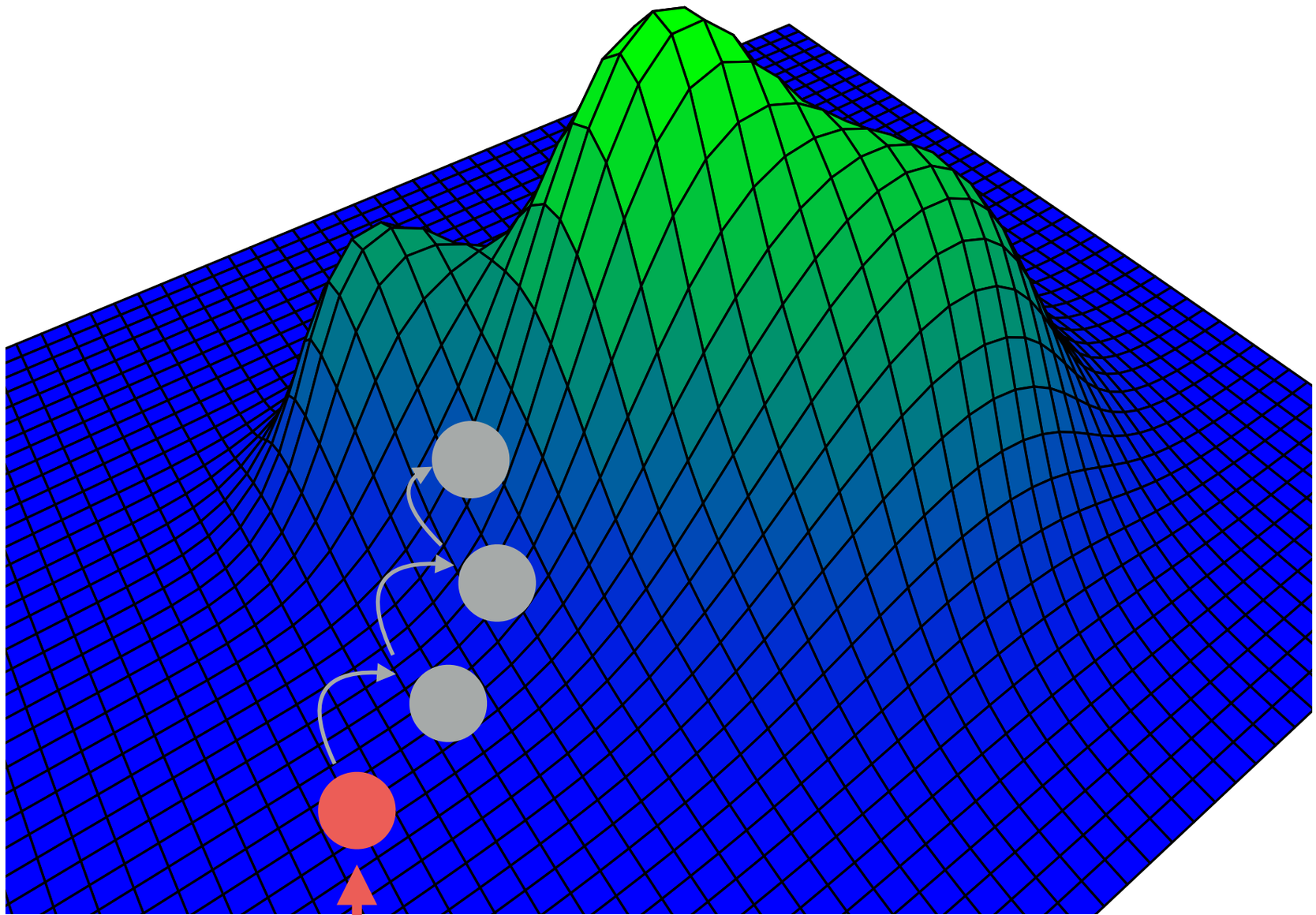
Starting tree

Traverse tree space



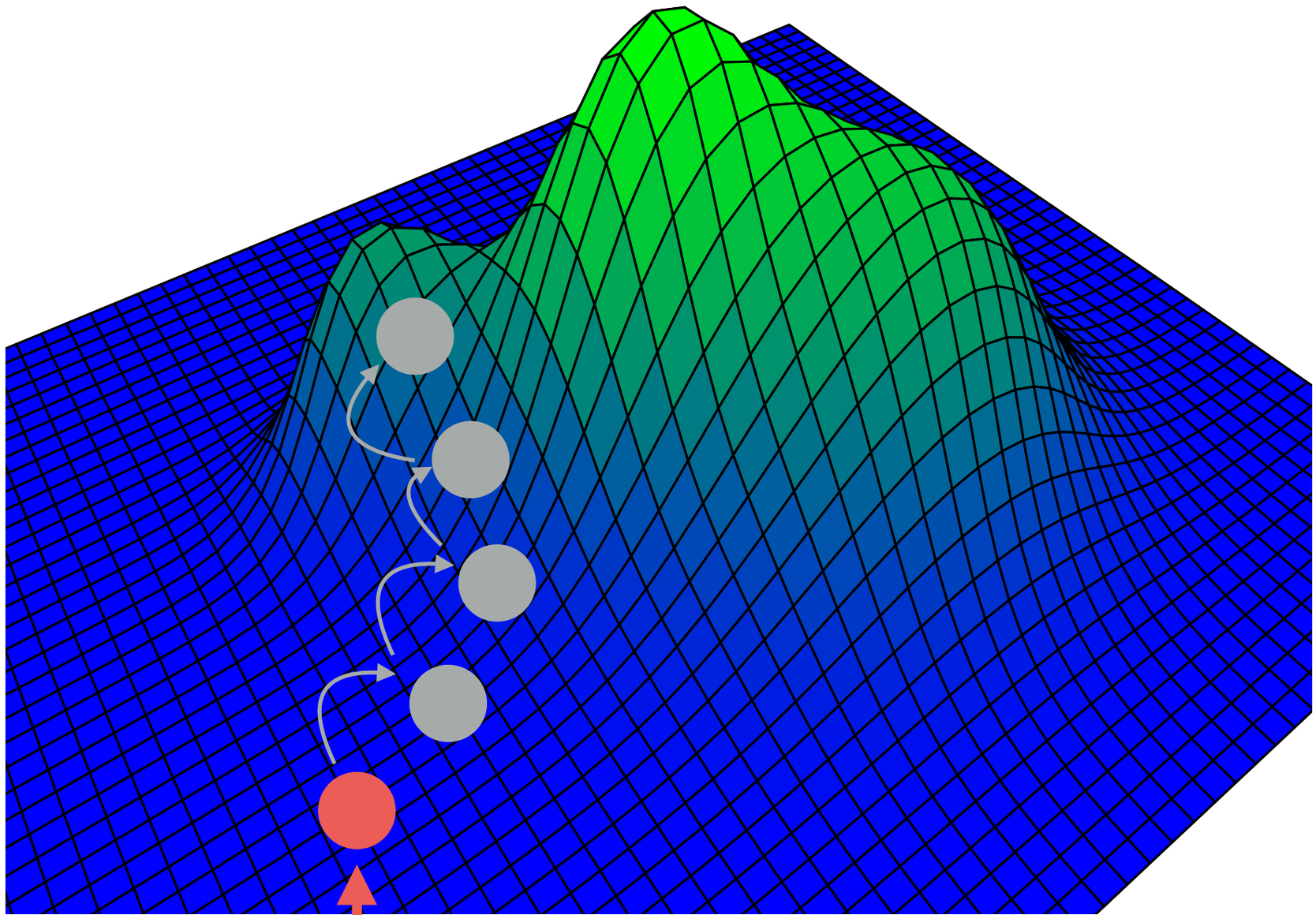
Starting tree

Traverse tree space



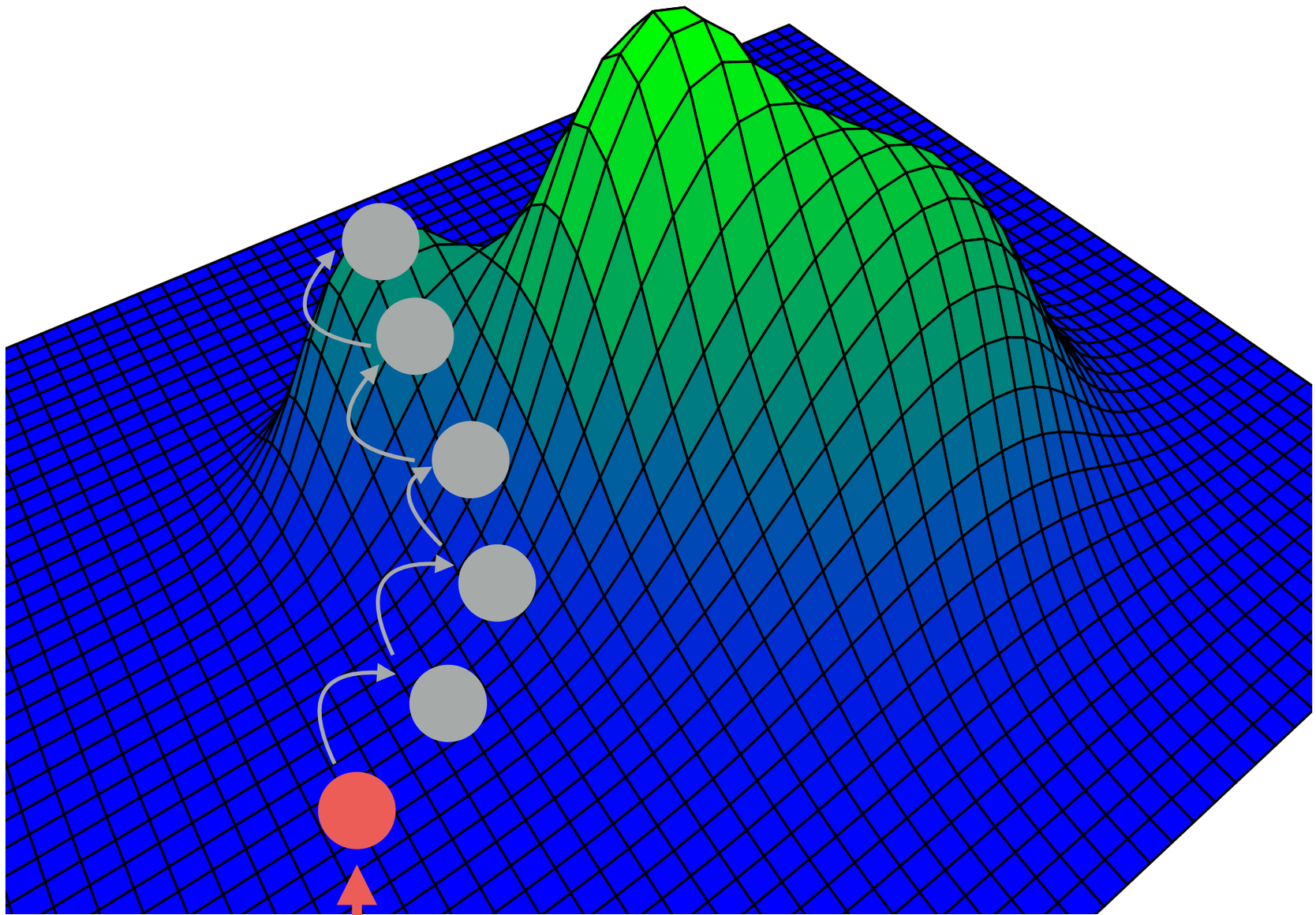
Starting tree

Traverse tree space



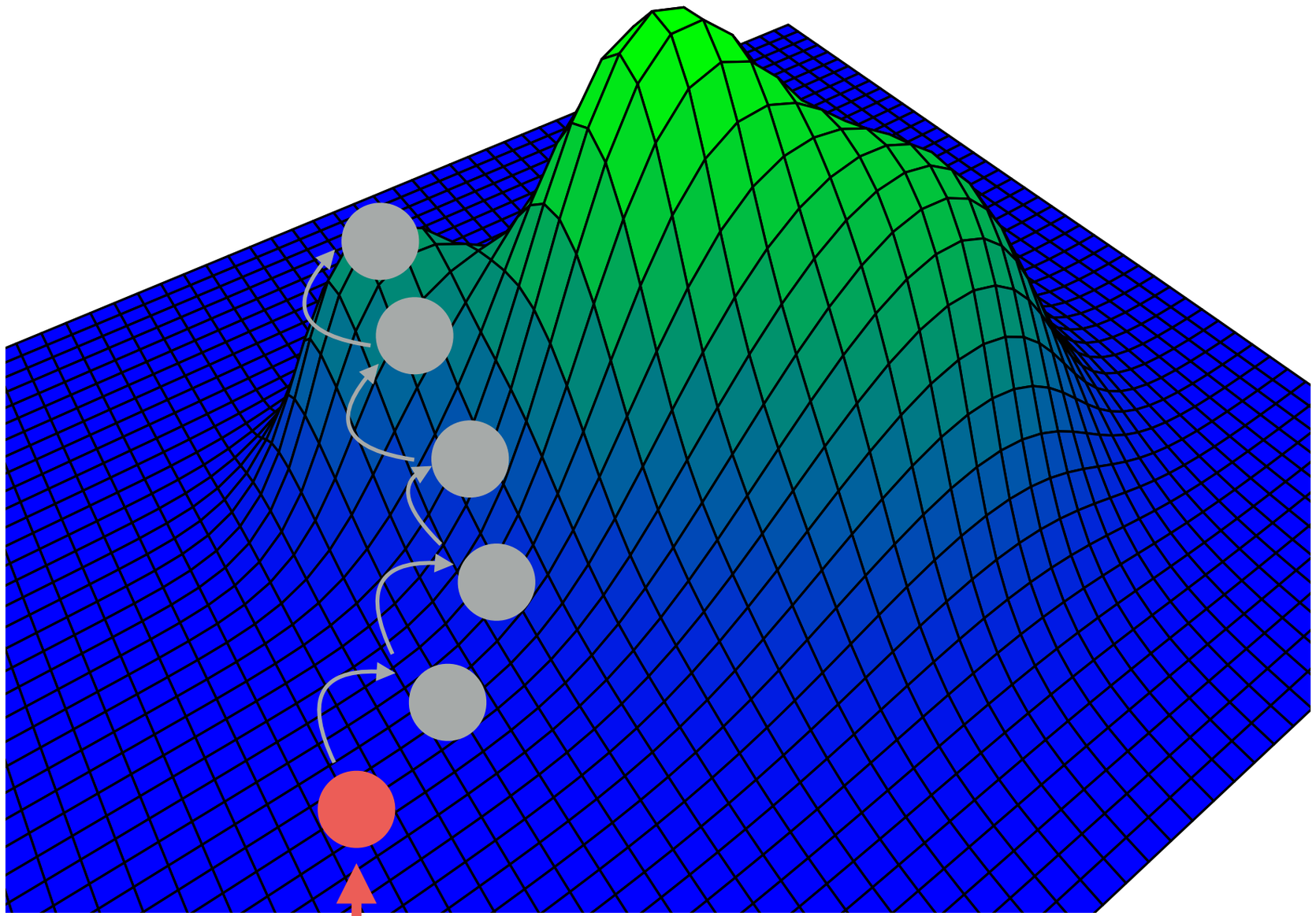
Starting tree

Traverse tree space



Starting tree

Traverse tree space



Starting tree

Nearest Neighbor Interchange (NNI)
Subtree Pruning and Regrafting (SPR)
Tree Bisection and Reconnection (TBR)

NNI

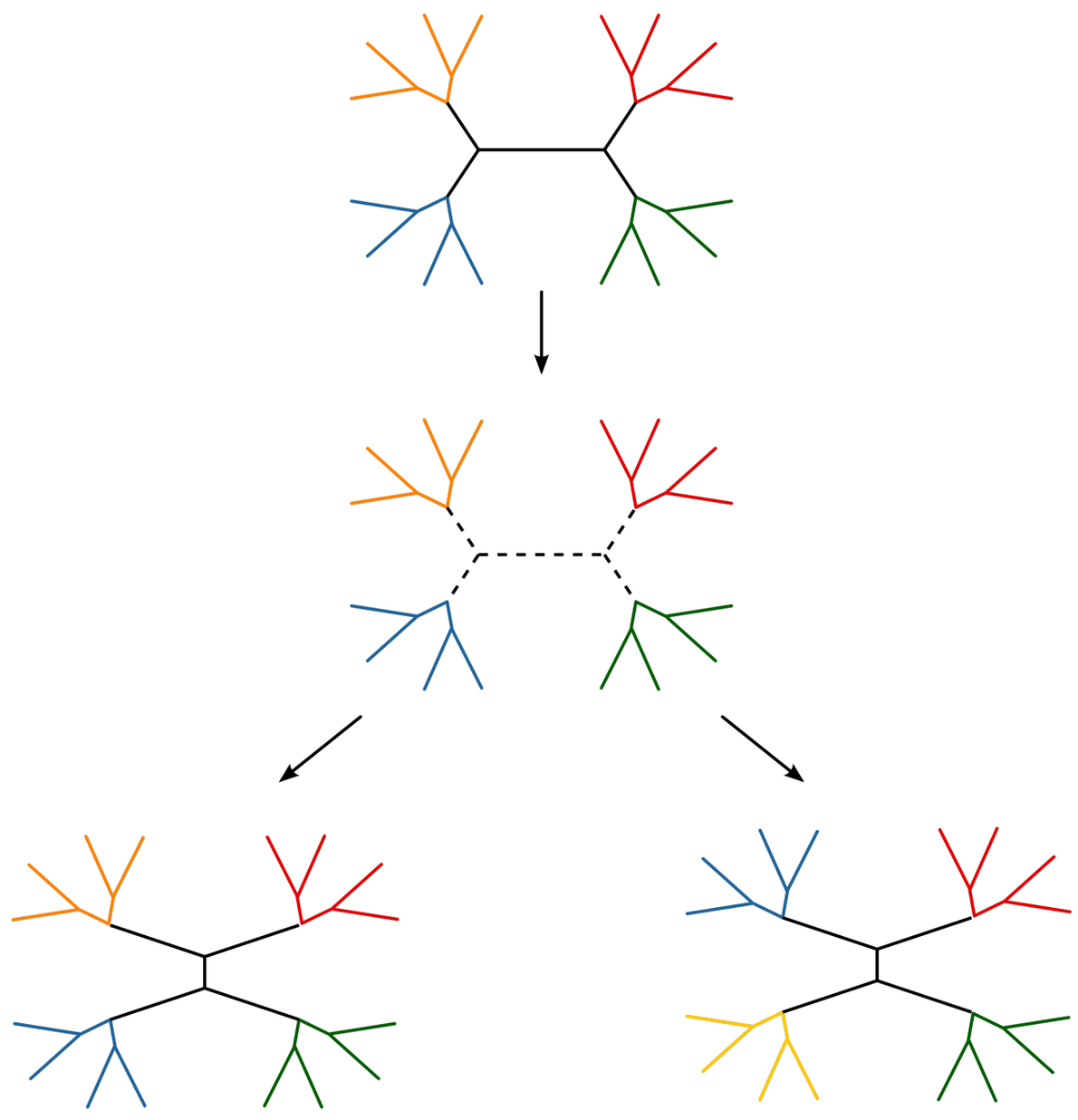


Image: Wikipedia

SPR

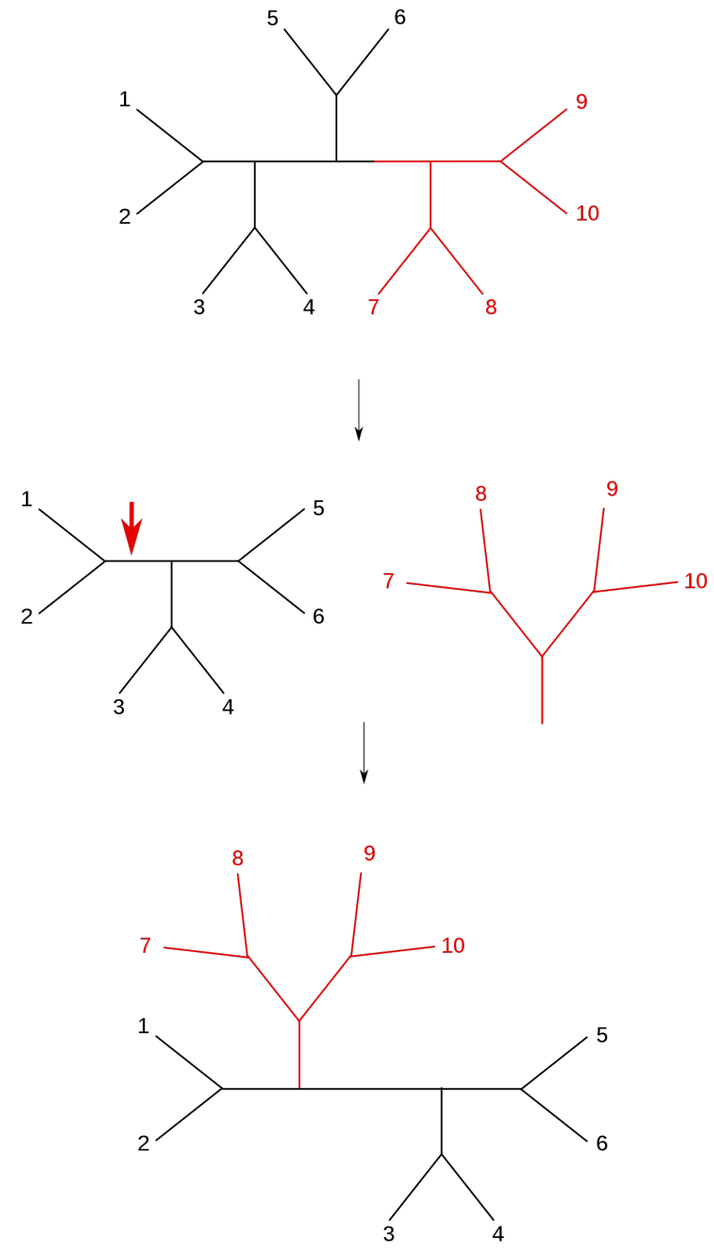
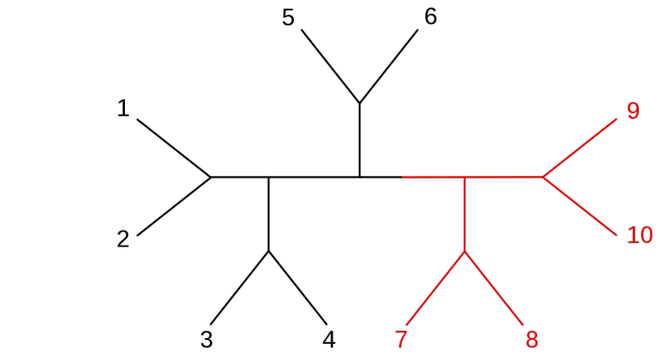


Image: Wikipedia

TBR



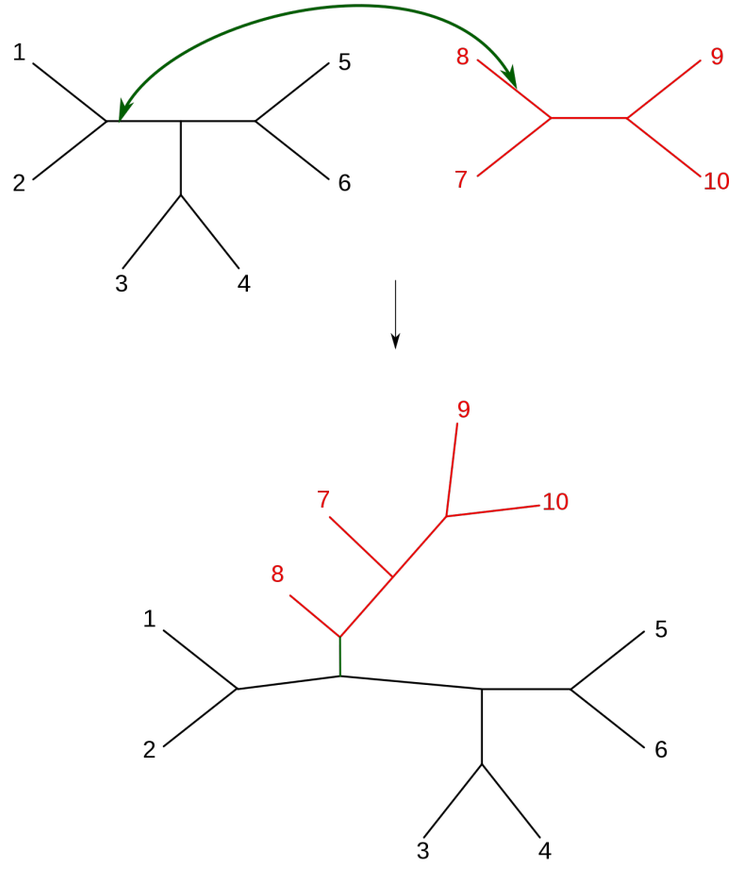


Image: Wikipedia