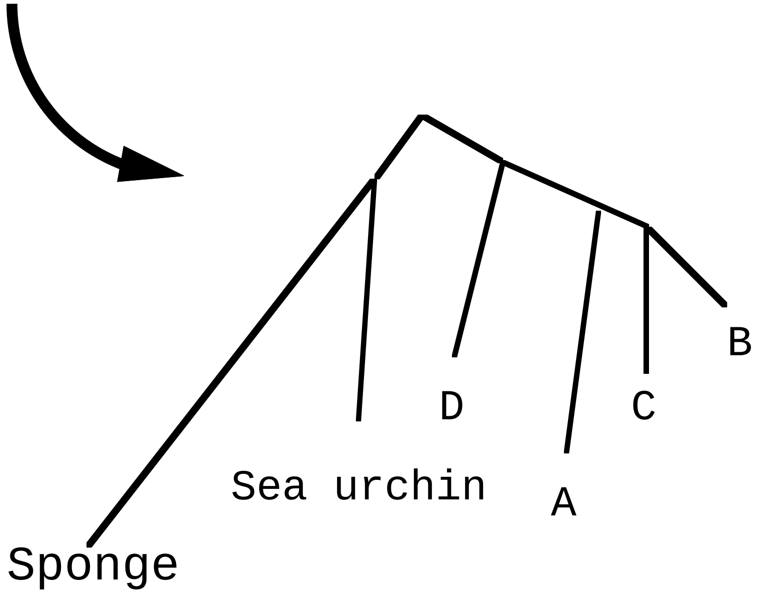
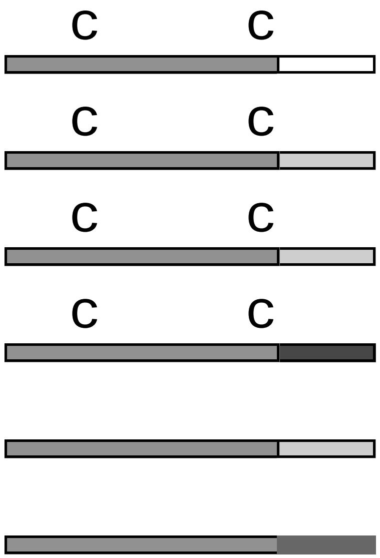
Evolutionary Trees

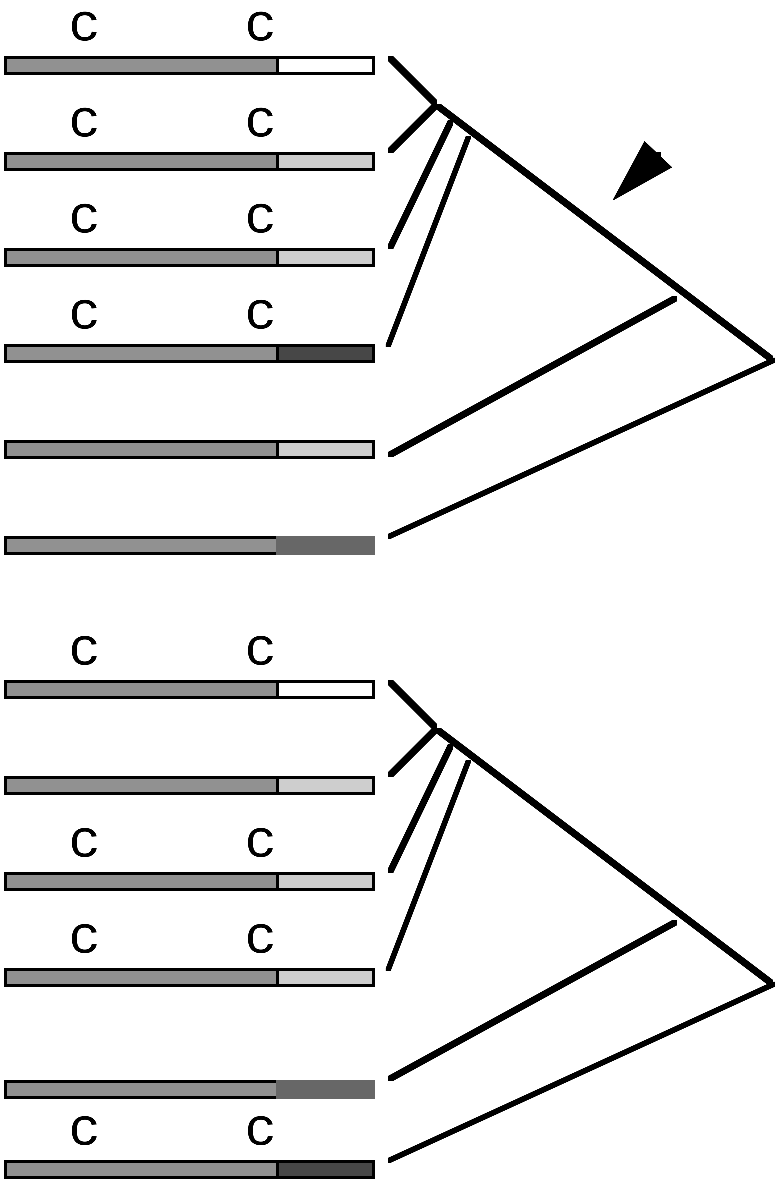
- Terminology
- Biological Assumptions
- Estimation Principles
- Algorithmic Structure
- Statistical Properties

The problem:

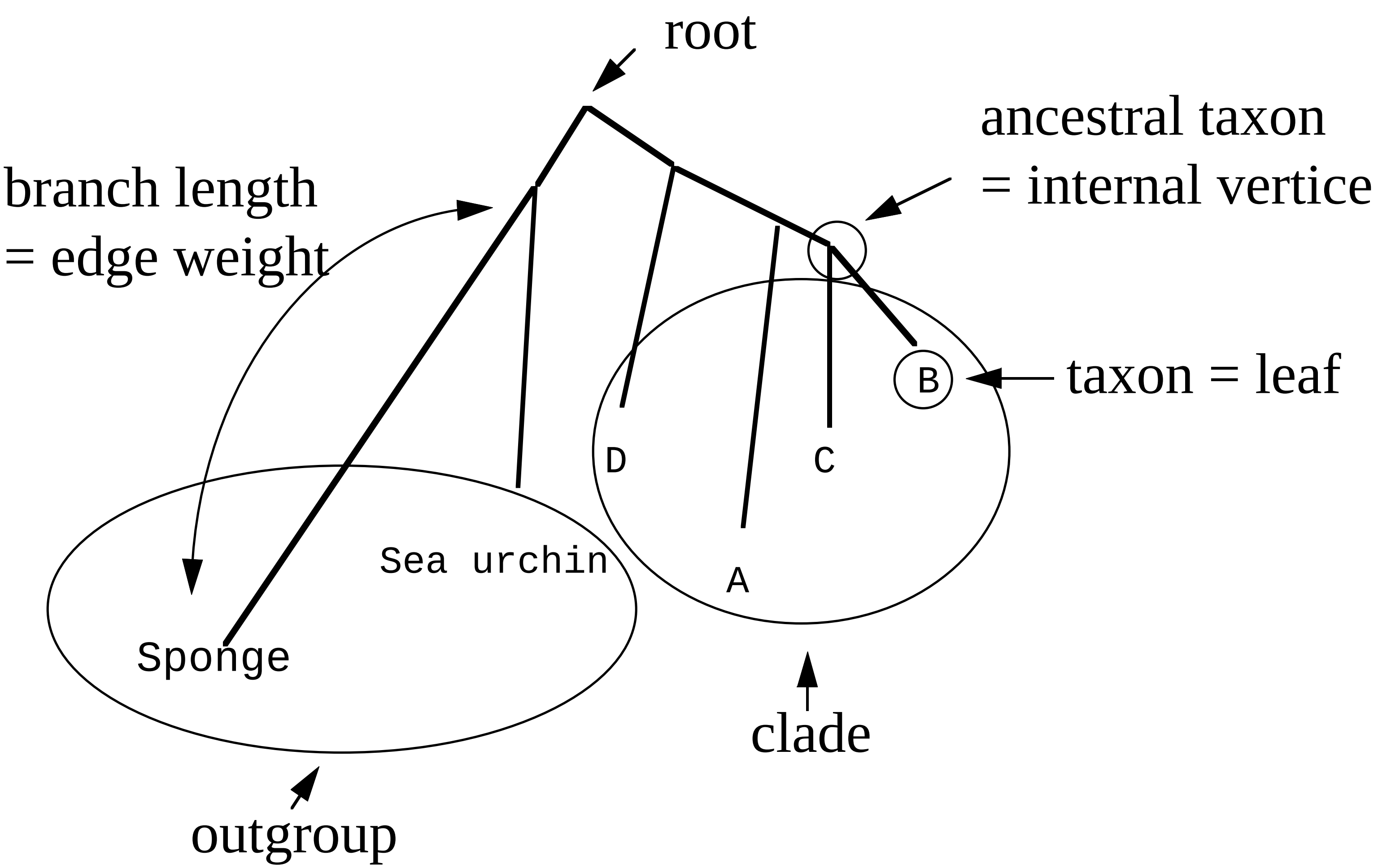
```
E. strong GGACCTCAAG GTTTGACTGG ACCATCCGGA CCTTCAGGAG AGACTGGACC
Sponge    ATGCCGCCTG GCTTCTTCGA CCCCAAAGGG CCCGCTCCTG AGCTTGGACC
HSA1A1    GGTCCCCAAG GCTTCCAAGG TCCCCCTGGT GAGCCTGGCG AGCCTGGAGC
HSA1A2    GGCCCTCAAG GTTTCCAAGG ACCTGCTGGT GAGCCTGGTG AACCTGGTCA
HSA2A1    GGTGCTCCTG GGCCTCAAGG ATTTCAAGGC AATCCTGGTG AACCTGGTGT
HSA3A1    GGTCATCCTG GTTCCCCTGG ATCTCCAGGA TACCAAGGGC AAGCTGGTCC
```

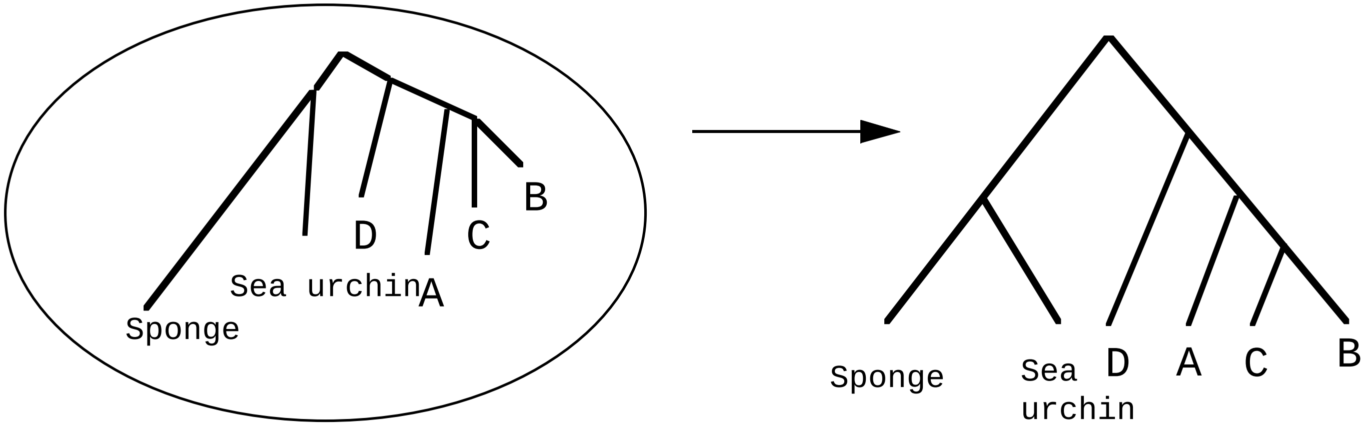



C
C
C
C
C
C
C
C


C C
C C
C C
C C
C C
C C
C C
C C

Terminology

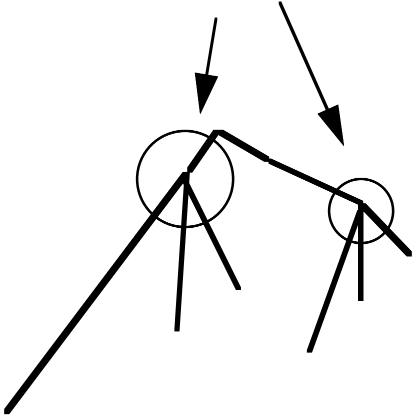


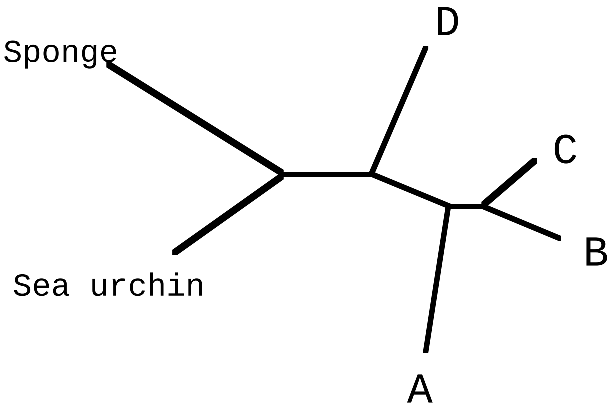


Ultrametric tree

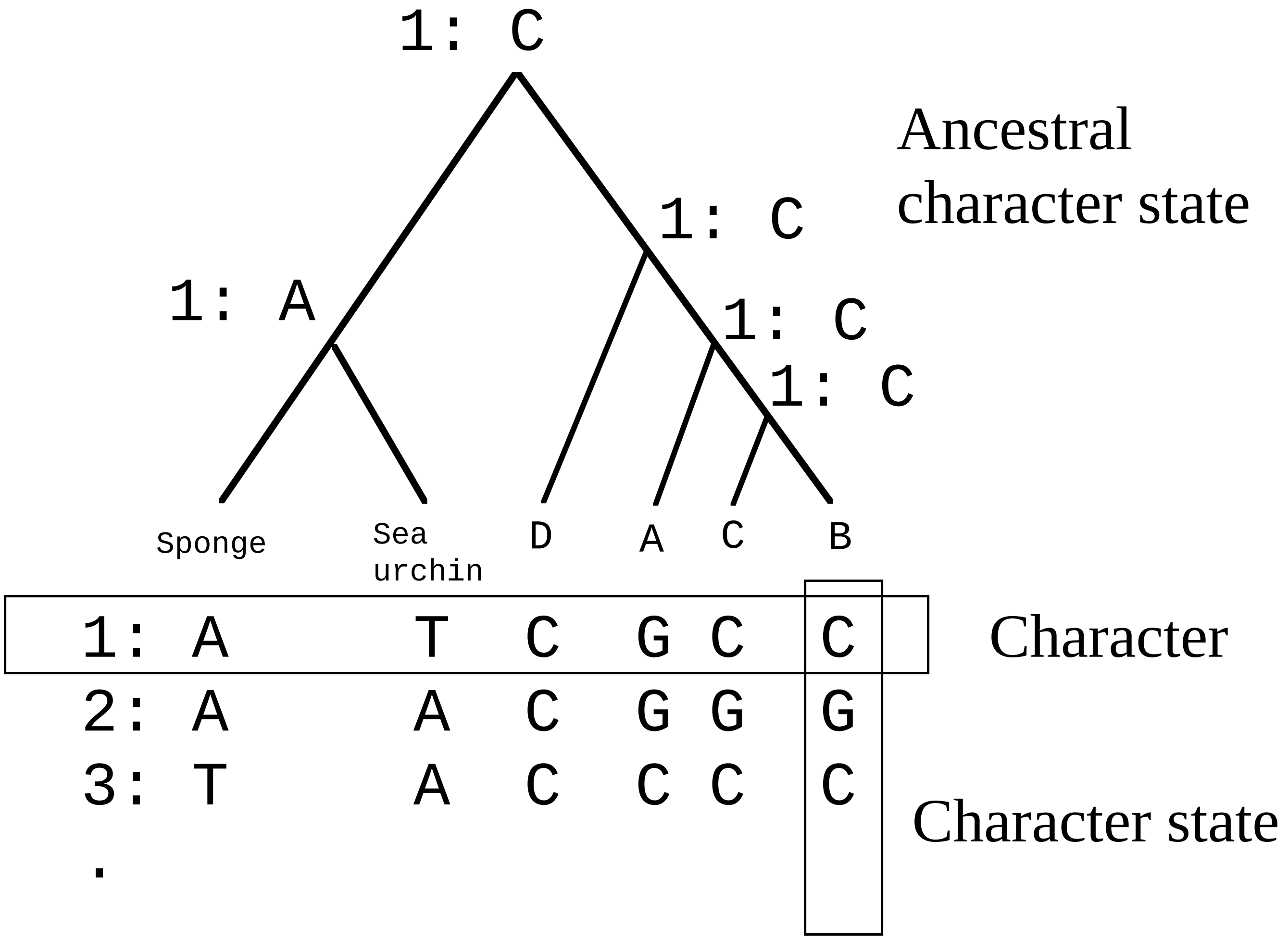
Tree topology → ((Sponge,Sea urchin),(D(A(C,B))))

Polytomy





Unrooted tree



Ancestral character state

	Sponge	Sea urchin	D	A	C	B
1:	A	T	C	G	C	C
2:	A	A	C	G	G	G
3:	T	A	C	C	C	C

Character

Character state

.
.
.

{A, C, T, G} : 4-state

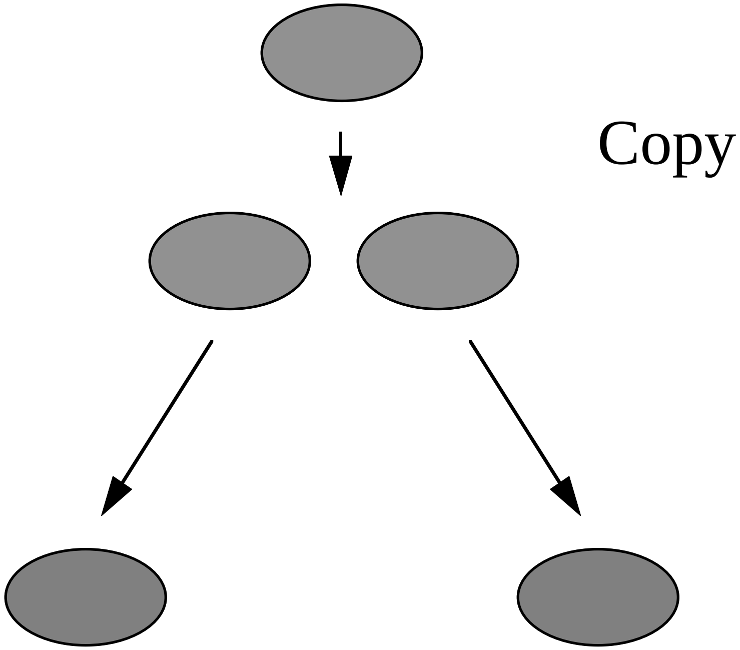
{0, 1} : Binary state

{Met, Ala, His,...,Trp} : 20-state

Biological Assumptions

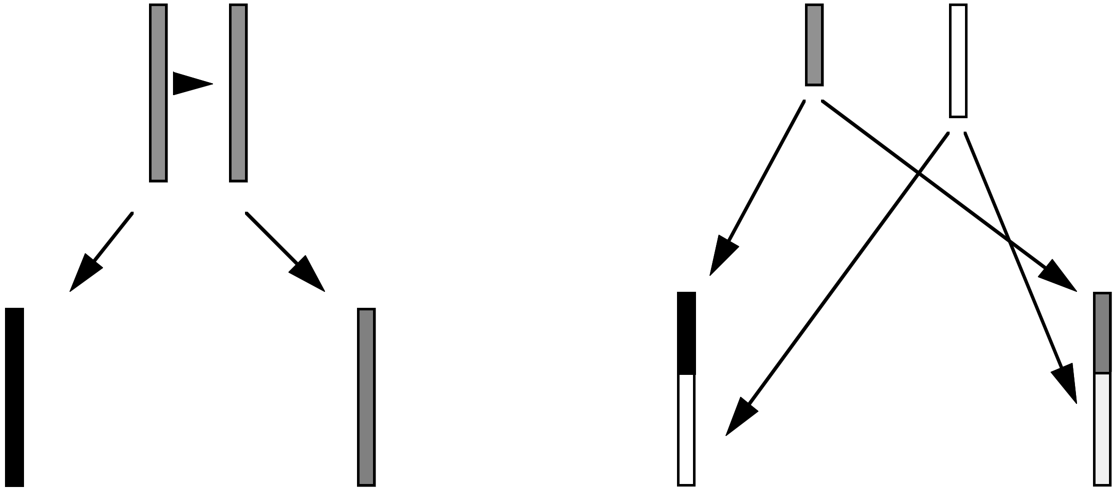
- Character Homology
- Bifurcating (multi-furcating) Descent

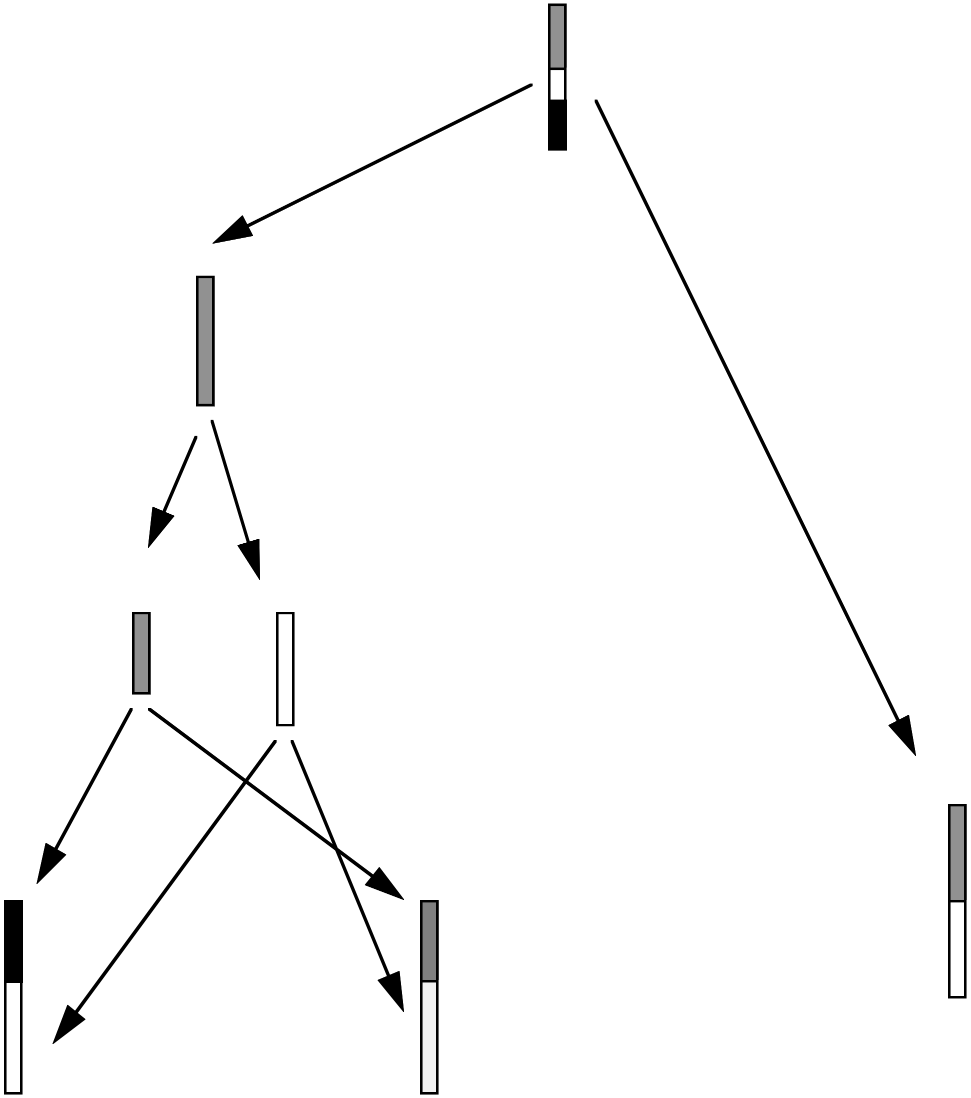
Character Homology



Two characters are considered homologous if they are descendents of an ancestral character with similar function

Bifurcating descent

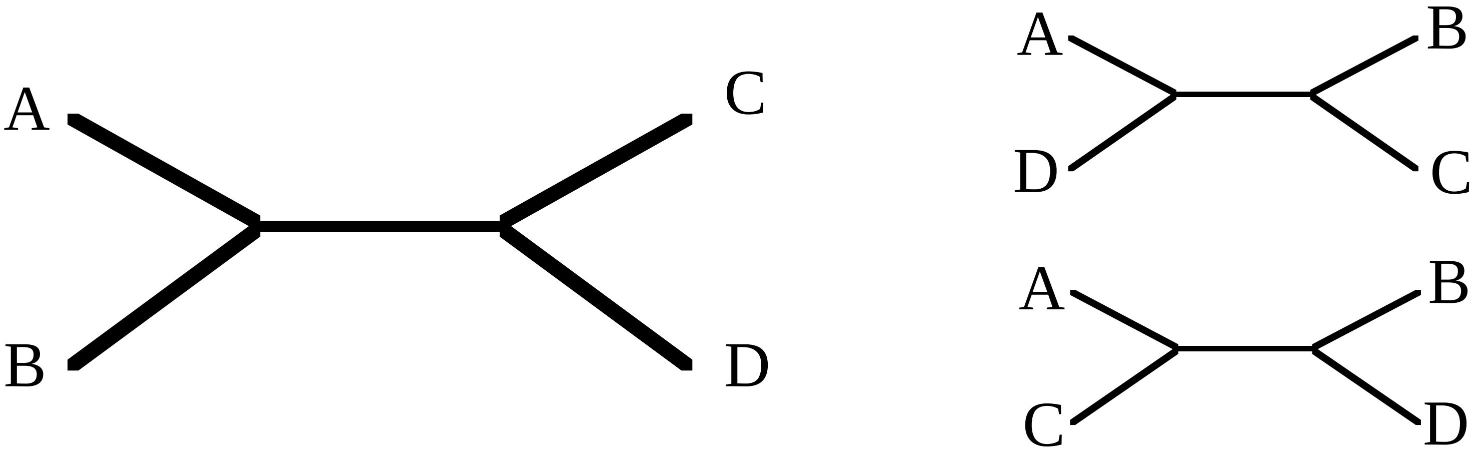




Estimation Principles

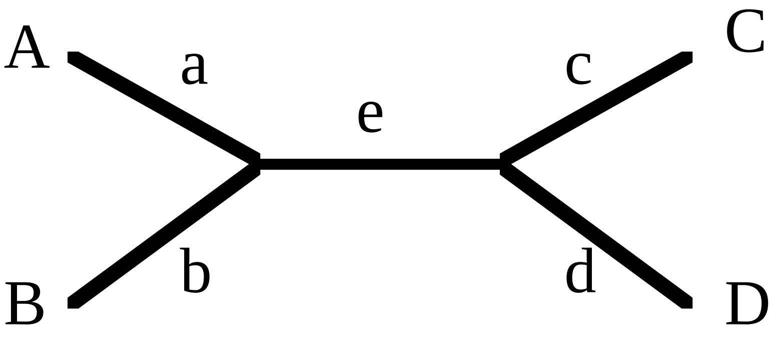
- Distance relationships
- Nested character states
- Likelihood

Distance relationships



A and B are more closely related to each other than they are to any other taxa

-> Clustering algorithms : UPGMA, WPGMA, Sattah and Tversky, Split decomposition



d(A,B) = a + b

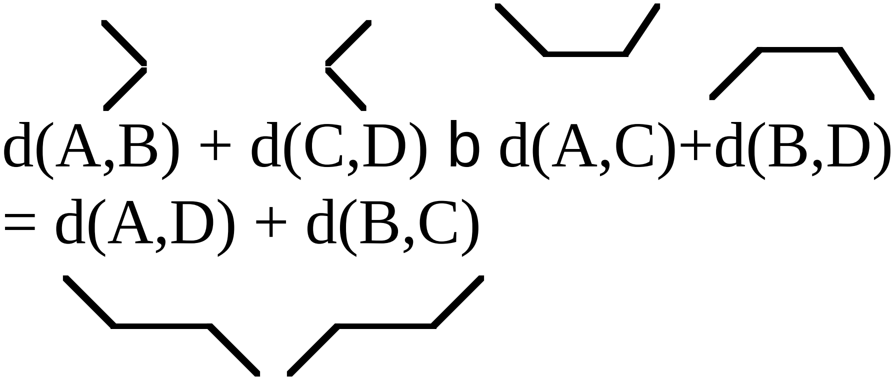
d(A,C) = a + e + c

d(A,D) = a + e + d

d(B,C) = b + e + c

d(B,D) = b + e + d

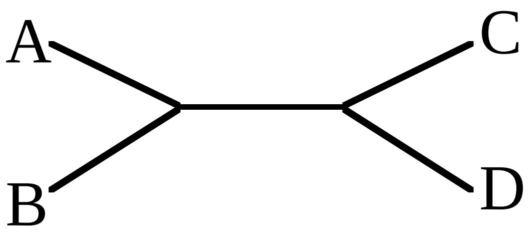
d(C,D) = c + d



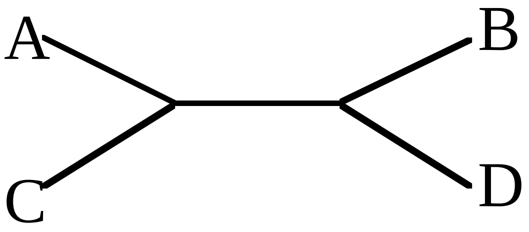
d(A,B) + d(C,D) b d(A,C)+d(B,D)
= d(A,D) + d(B,C)

	A	B	C	D
A	0	3	2	1
B	3	0	1	2
C	2	1	0	3
D	1	2	3	0

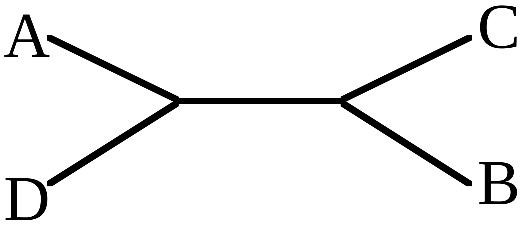
d(A,B) + d(C,D) = 6
d(A,C) + d(B,D) = 4
d(A,D) + d(B,C) = 2



d(A,B)+d(C,D) b d(A,C)+d(B,D) = d(A,D) +d(B,C)



d(A,C)+d(B,D) b d(A,B)+d(C,D) = d(A,D) +d(B,C)

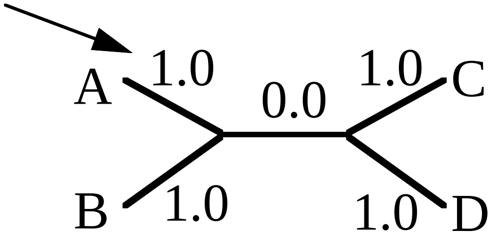


d(A,D)+d(B,C) b d(A,C)+d(B,D) = d(A,B) +d(C,D)

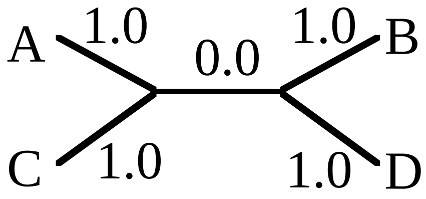
	A	B	C	D
A	0	3	2	1
B	3	0	1	2
C	2	1	0	3
D	1	2	3	0

L^2-additive tree

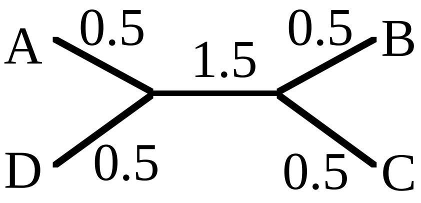
Numeric optimization



$$(3-2)^2+(2-2)^2+(1-2)^2+(1-2)^2+(2-2)^2+(3-2)^2 = 4$$

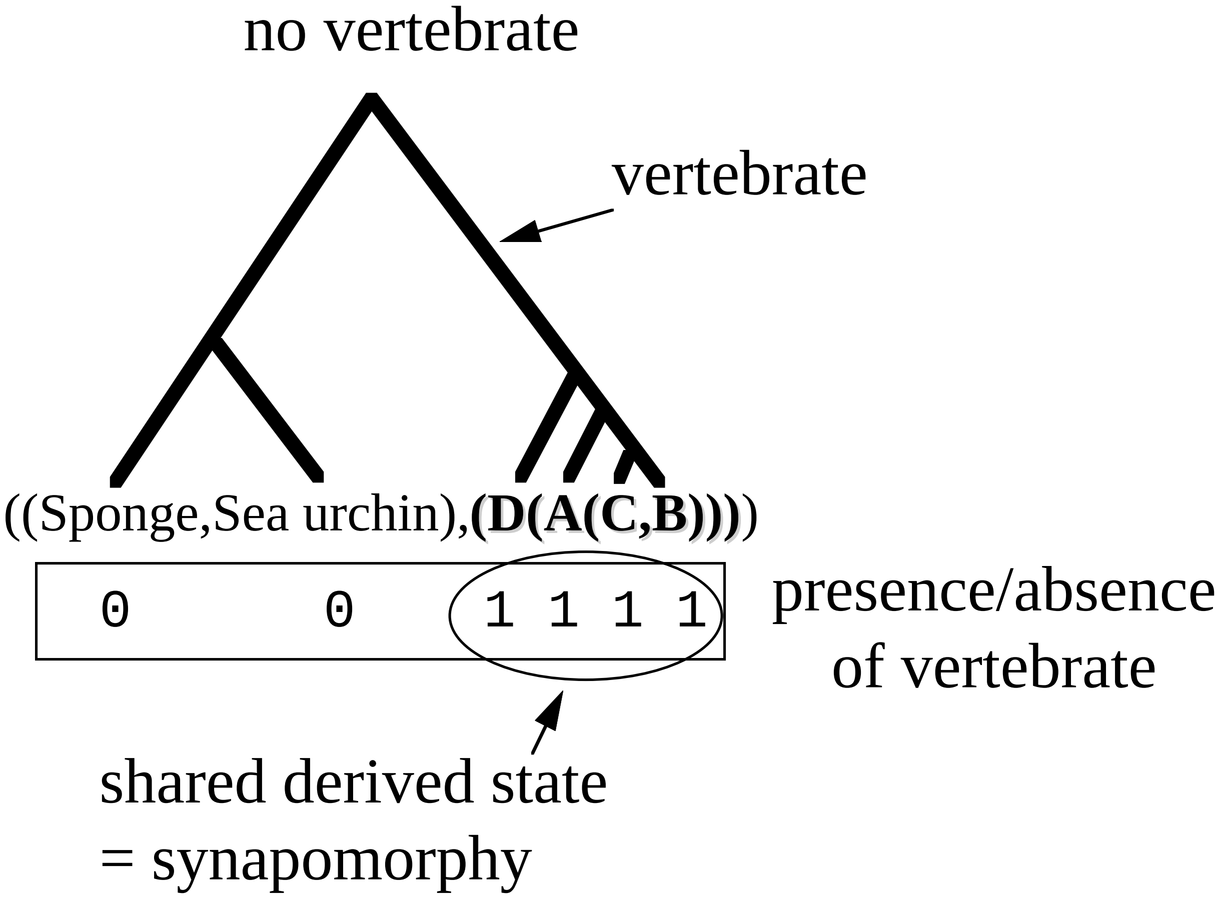


$$(3-2)^2+(2-2)^2+(1-2)^2+(1-2)^2+(2-2)^2+(3-2)^2 = 4$$



$$(3-2.5)^2+(2-2.5)^2+(1-1)^2+(1-1)^2+(2-2.5)^2+(3-2.5)^2 =1$$

Nested character-states



no vertebrate

vertebrate

((Sponge,Sea urchin),**(D(A(C,B)))**)

0 0 1 1 1 1

presence/absence of vertebrate

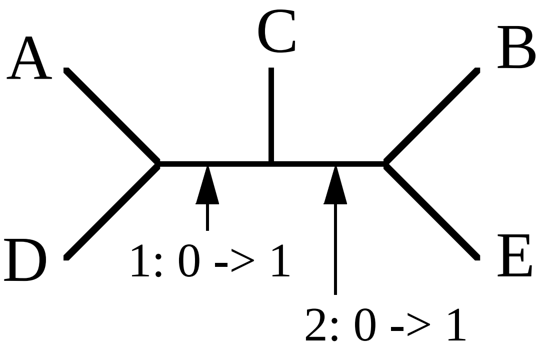
shared derived state
= synapomorphy

```
   A B C D E
1: 0 1 1 0 1
```

```
   A B C D E
2: 0 1 0 0 1
```

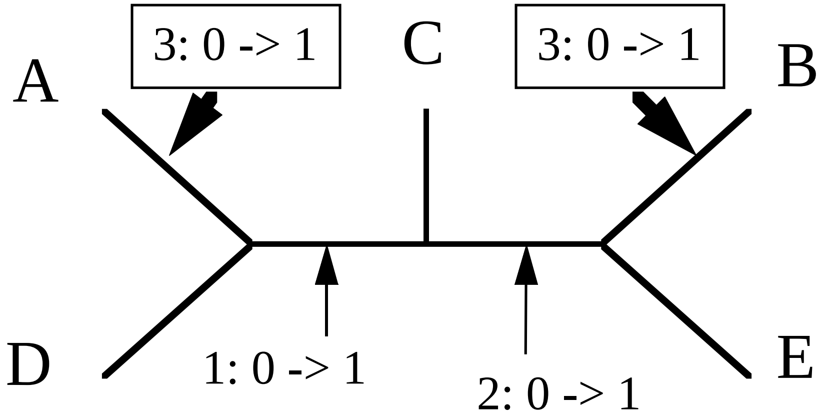
(A,D),(B,C,E) + (A,C,D),(B,E)

((A,D),(C,(B,E))) = (((A,D),C),(B,E))

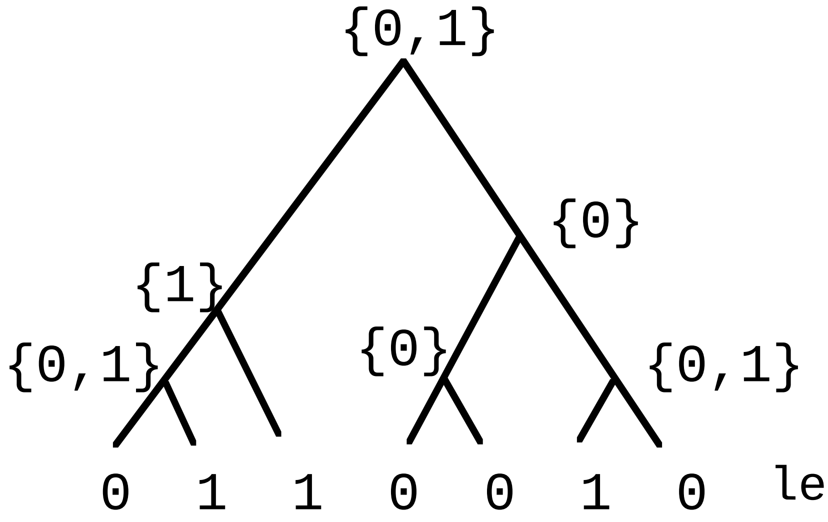


	A	B	C	D	E		Parsimony length
1:	0	1	1	0	1	(A,D)(B,C,E)	1
2:	0	1	0	0	1	(A,C,D)(B,E)	1
3:	0	0	1	1	1	(A,B)(C,D,E)	2
4:	0	0	1	1	0	(A,B,E)(C,D)	2
5:	0	1	1	0	1	(A,D)(B,C,E)	1

□ Compatible characters

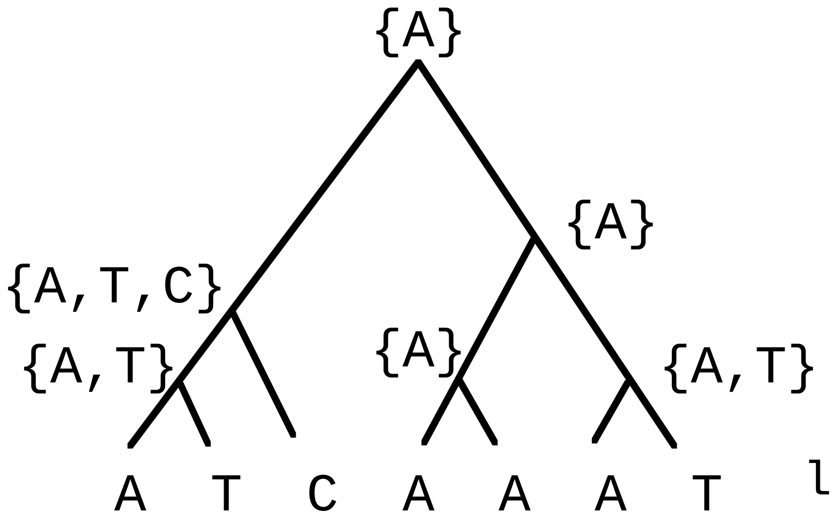


Compatibility Tree, Maximum Parsimony Tree



length = 3

Fitch-Hartigan Algorithm



length = 3

parsimony length of n characters = $\sum_{i=1}^{n} l_i$

l_i = parsimony length of *i*th character

Likelihood

$$\boxed{P(M \backslash D) = \frac{P(MD)}{P(D)} = \frac{P(D \backslash M)P(M)}{P(D)}}$$

P(M \ D) ~ P(D \ M)

therefore, P(D \ M) is called the likelihood

Suppose we see “AAAT”, what is the probability of drawing a base “A”?

Don’t know!

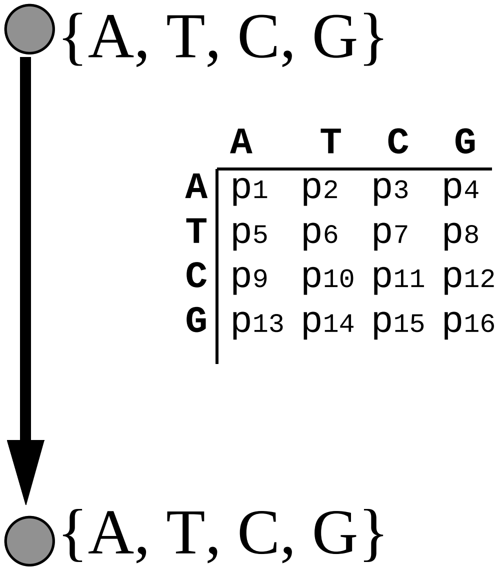
Likelihood = P(D \ M)

Likelihood = $4p^3(1-p)$

Find p such that the likelihood is maximized -> p = 3/4

Log(Likelihood) =
$\text{Log}(4p^3(1-p)) = \text{Log}(4) + 3\ \text{Log}(p) + \text{Log}(1-p)$

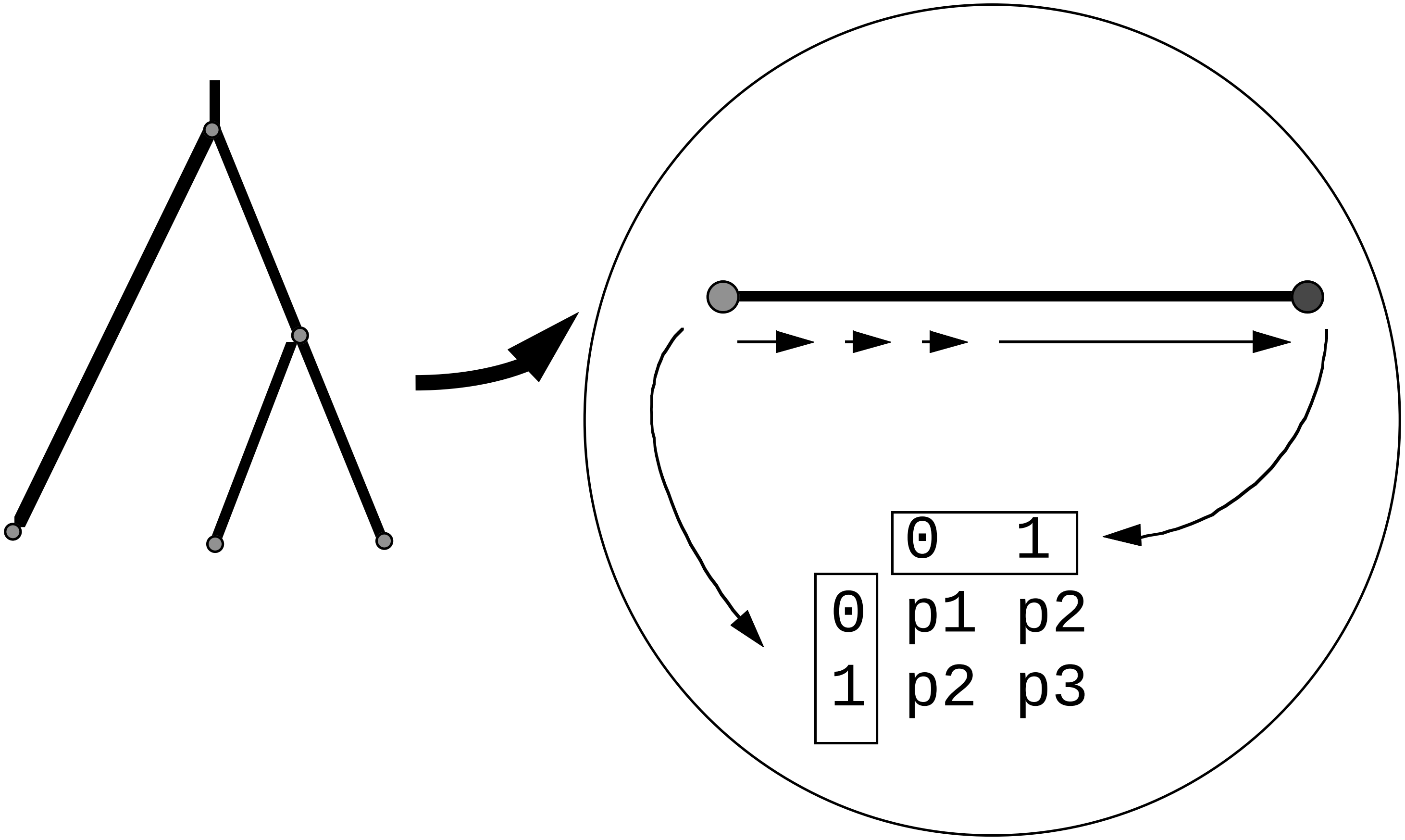
Markov chain model of character evolution



	A	T	C	G
A	p_1	p_2	p_3	p_4
T	p_5	p_6	p_7	p_8
C	p_9	p_{10}	p_{11}	p_{12}
G	p_{13}	p_{14}	p_{15}	p_{16}

- Finite number of states (e.g., {A, T, C, G})
- Transition matrix : probability of observing state *i* given state *j*

Transition probability is specified from node to node



	0	1
0	p1	p2
1	p2	p3

The character evolution model is determined by the form of the constraints on the transition matrix

$$\begin{array}{c|cccc} & A & T & G & C \\ \hline A & Q_1 & p_1 & p_2 & p_3 \\ T & p_4 & Q_2 & p_5 & p_6 \\ C & p_7 & p_8 & Q_3 & p_9 \\ G & p_{10} & p_{11} & p_{12} & Q_4 \end{array}$$

T(12)

$$\begin{array}{c|cccc} & A & T & G & C \\ \hline A & Q_1 & p_1 & p_2 & p_1 \\ T & p_1 & Q_2 & p_1 & p_2 \\ C & p_2 & p_1 & Q_3 & p_1 \\ G & p_1 & p_2 & p_1 & Q_4 \end{array}$$

T(2)

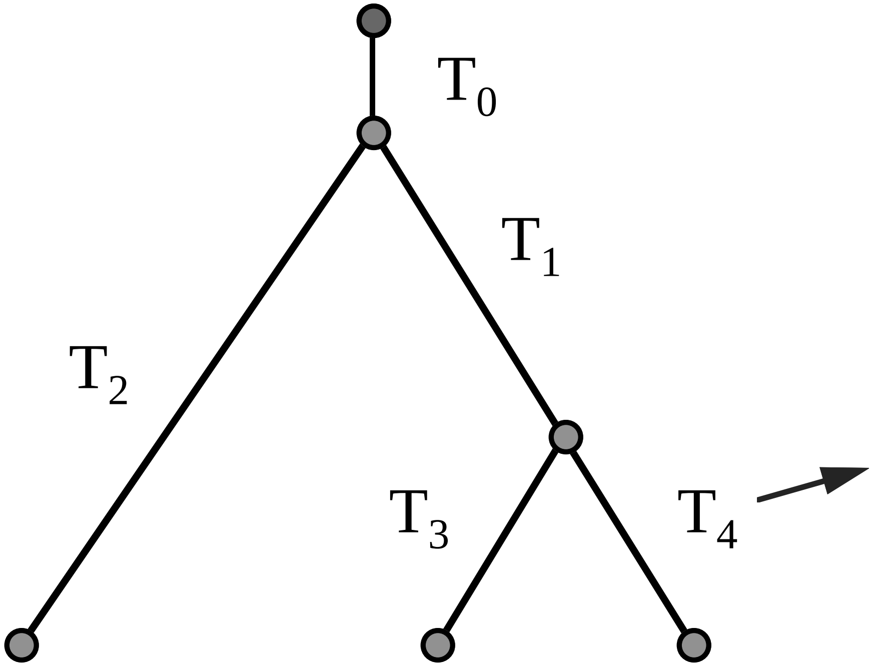
$$\begin{array}{c|cccc} & A & T & G & C \\ \hline A & Q_1 & p_1 & p_2 & p_3 \\ T & p_1 & Q_2 & p_4 & p_5 \\ C & p_2 & p_4 & Q_3 & p_6 \\ G & p_3 & p_5 & p_6 & Q_4 \end{array}$$

T(6)

$$\begin{array}{c|cccc} & A & T & G & C \\ \hline A & Q_1 & p_1 & p_1 & p_1 \\ T & p_1 & Q_2 & p_1 & p_1 \\ C & p_1 & p_1 & Q_3 & p_1 \\ G & p_1 & p_1 & p_1 & Q_4 \end{array}$$

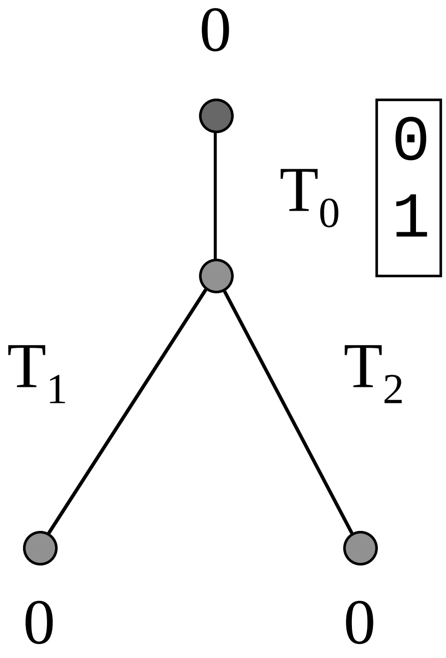
T(1)

The model is specified by the **branching order** of the tree, the **initial state** at the common ancestor, and a **transition matrix** for each branch

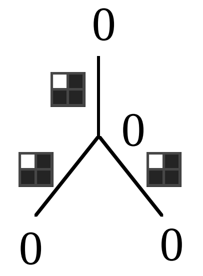


$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & p1 & p2 \\ 1 & p3 & p4 \end{array}$$

Given the model, the probability of any character pattern at the tips of the tree can be computed

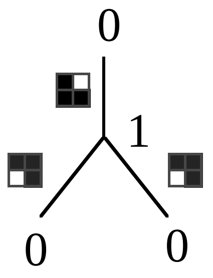


$$T_0 \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & p_1^0 & p_2^0 \\ 1 & p_3^0 & p_4^0 \end{array}$$



$p_1^0\ p_1^1\ p_1^2$

$+$

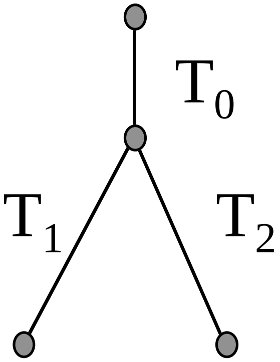


$p_2^0\ p_3^1\ p_3^2$

$\text{Prob}\{0, 0 \mid 0\} = p_1^0\ p_1^1\ p_1^2 + p_2^0\ p_3^1\ p_3^2$

Prob{0, 0} =
Prob{0, 0 | 0} Prob{Anc = 0} + Prob{0, 0 | 1} Prob{Anc = 1}

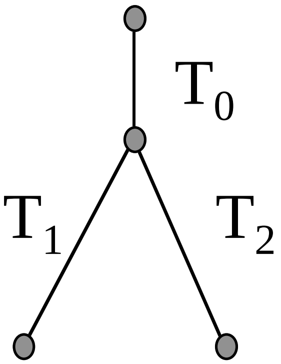
For *t* number of taxa and *n*-state characters there are n^t number of character patterns at the tips of the tree



0	0	Prob{0, 0} = π_1
0	1	Prob{0, 1} = π_2
1	0	Prob{1, 0} = π_3
1	1	Prob{1, 1} = π_4

$$\sum \pi_\iota = 1$$

-> Joint probability distribution of the character pattern



5: 0	0	Prob{0, 0} = π_1
6: 0	1	Prob{0, 1} = π_2
2: 1	0	Prob{1, 0} = π_3
3: 1	1	Prob{1, 1} = π_4

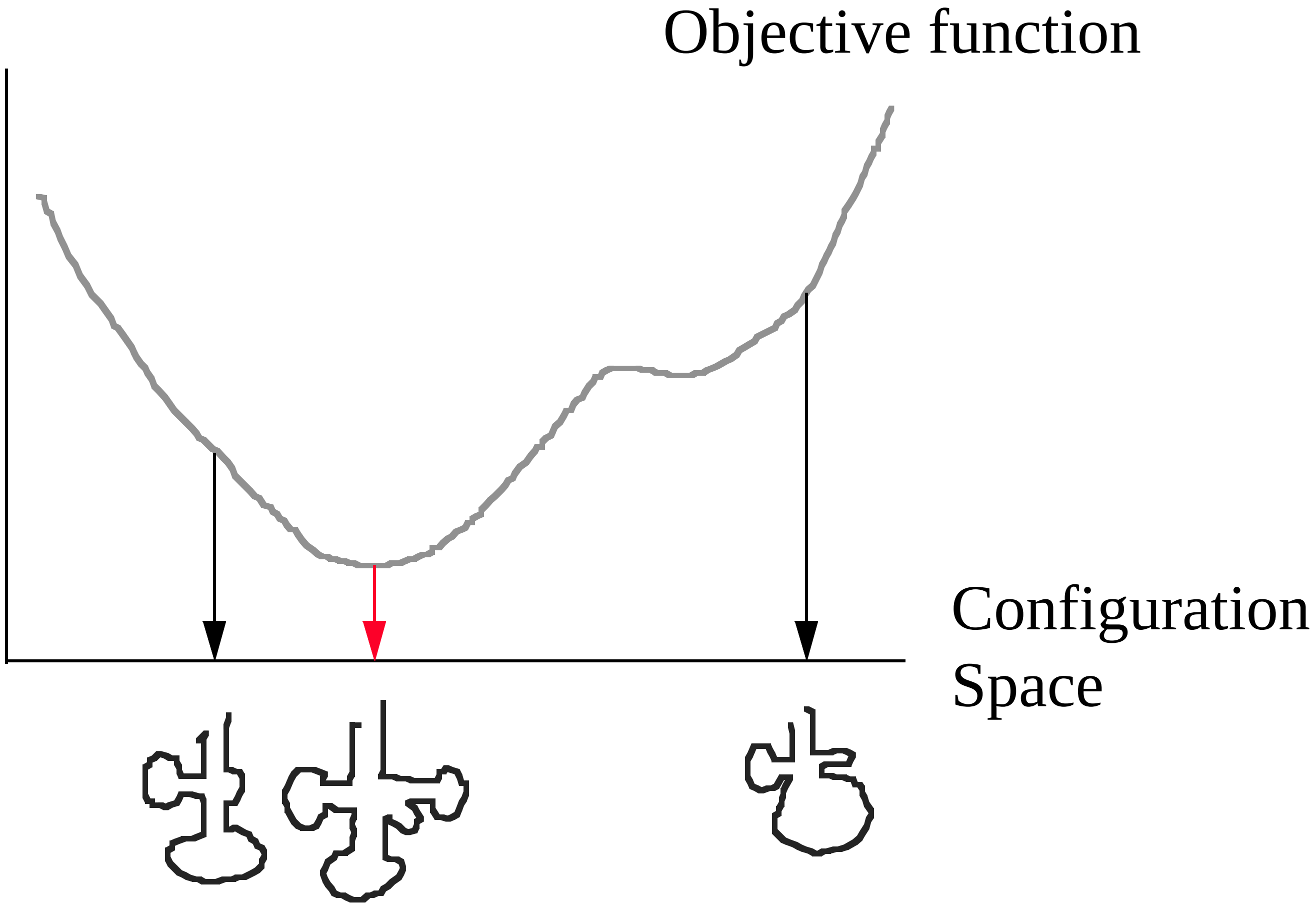
Likelihood ~ $\pi_1{}^5 \, \pi_2{}^6 \, \pi_3{}^2 \, \pi_4{}^3$

Maximum Likelihood Tree: Find the tree and the transition matrix values such that the likelihood is maximized

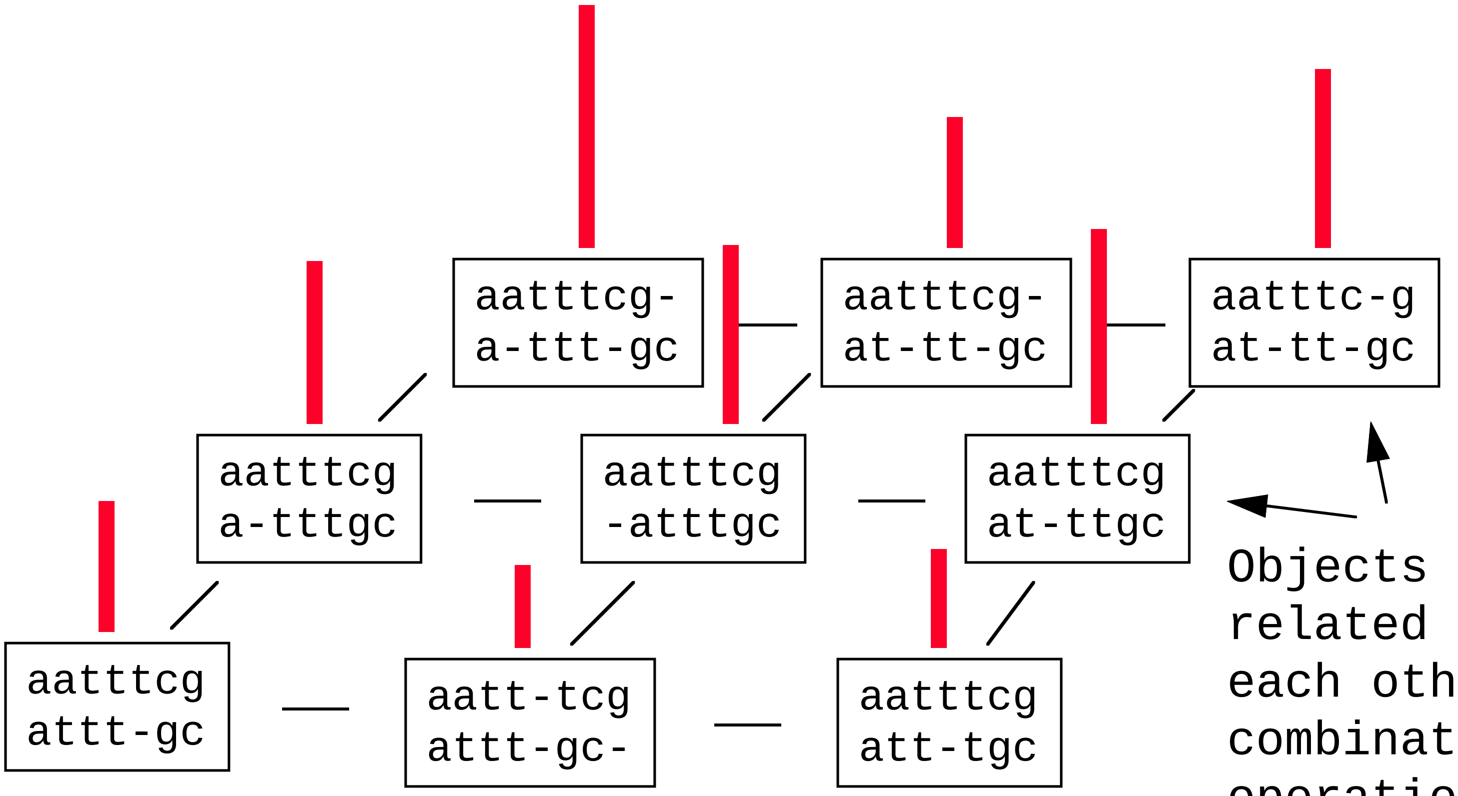
Algorithmic Structure

- Evolutionary tree estimation as a combinatorial optimization problem
- Overview of combinatorial optimization
- A taxonomy of evolutionary tree estimation algorithms

Optimization

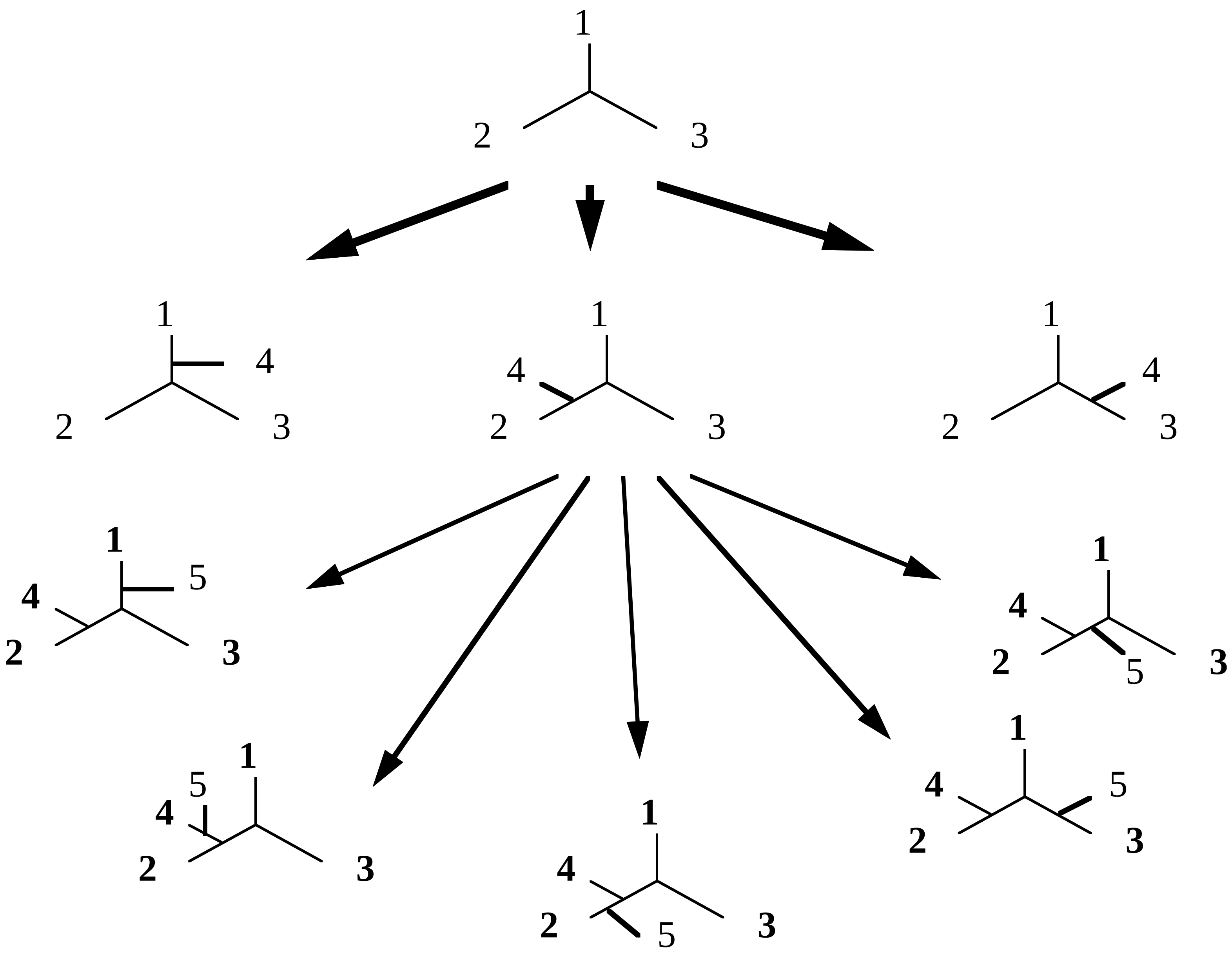


Combinatorial Optimization



```
aatttcg-
a-ttt-gc

aatttcg-
at-tt-gc

aatttc-g
at-tt-gc

aatttcg
a-tttgc

aatttcg
-atttgc

aatttcg
at-ttgc

aatttcg
attt-gc

aatt-tcg
attt-gc-

aatttcg
att-tgc
```

Objects are related to each other by combinatoric operations


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

Number of possible unrooted binary trees with n-taxa

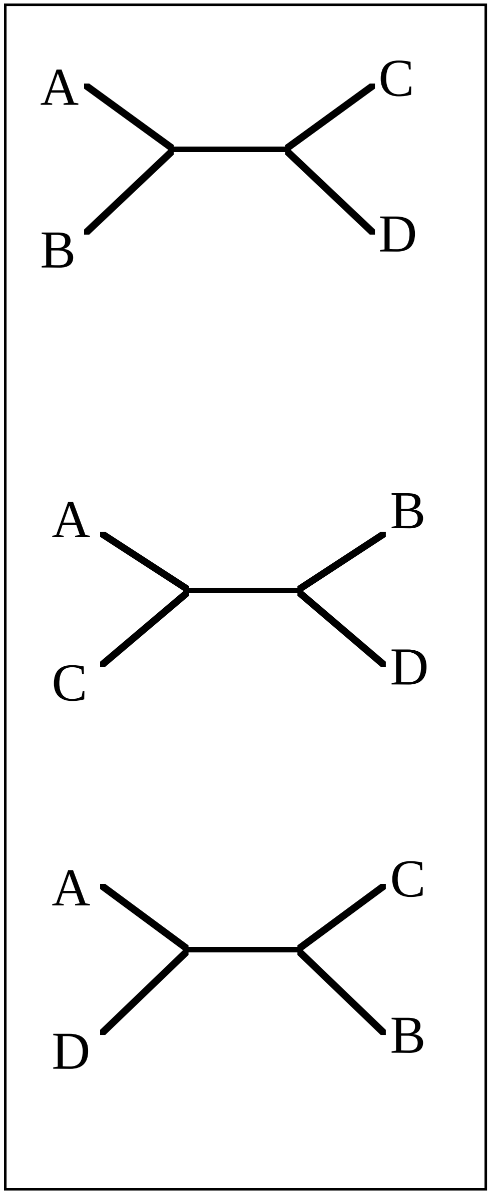
#taxa	#trees
3	1
4	1x3 = 3
5	1x3x5 = 15
6	1x3x5x7 = 105
7	1x3x5x7x9 = 945
.	
20	2.2 x 10^{20}
.	
100	1.7 x 10^{182}
.	
n	1x3x5x..x(2n-5)

$$\sqrt{\frac{2n-5}{n-3}}\,2^{3-n}e^{2-n}(2n-5)^{2n-5}(n-3)^{3-n}$$

```
aatcttacggtagtgt
aactgtacggaagtct
atctgtaccgaagcct
.
.
.
```

+

Configuration Space



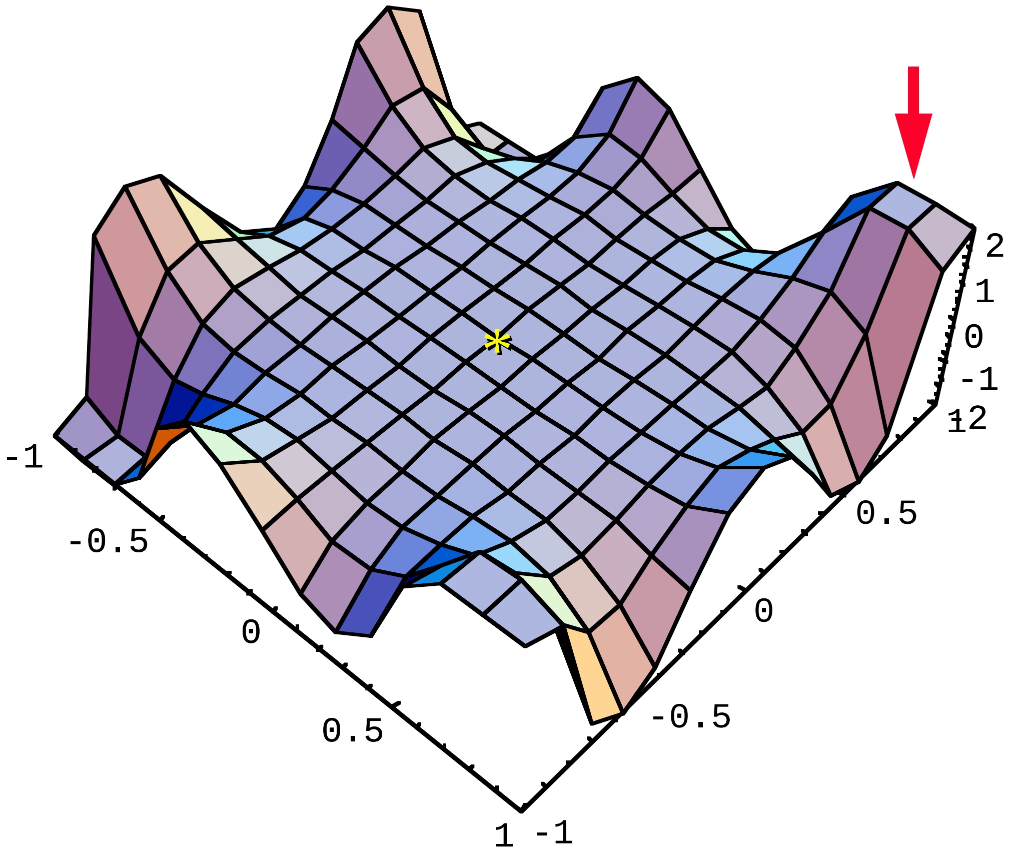
Objective Function

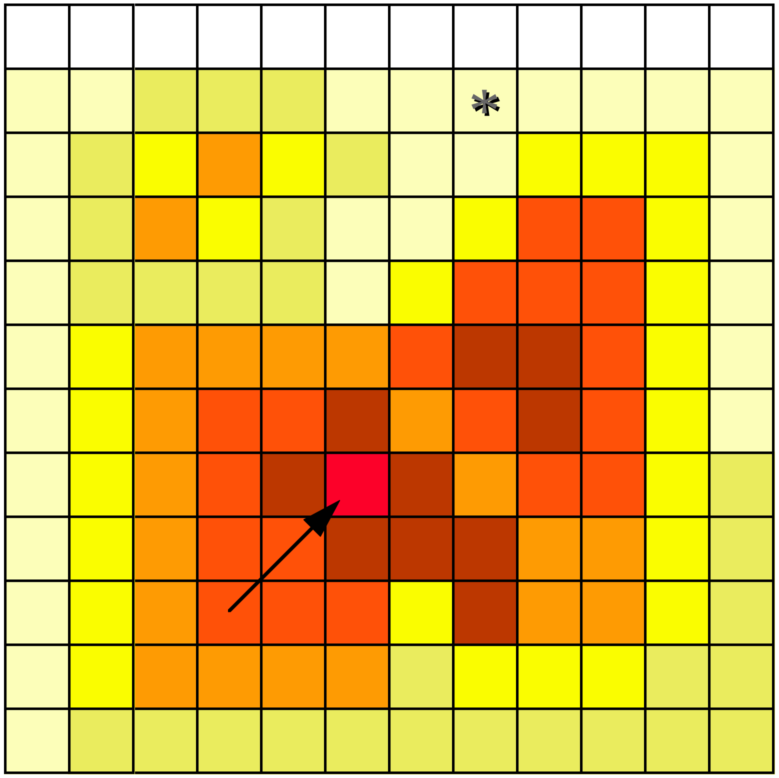
$MP(D,T_1) = x$

$MP(D,T_2) = y$

$MP(D,T_3) = z$

The Problem

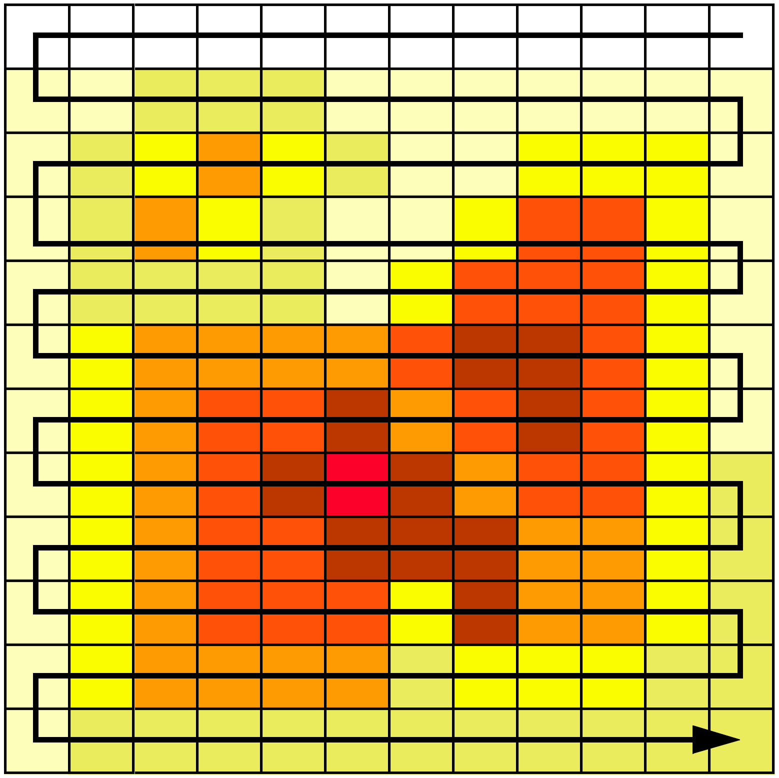




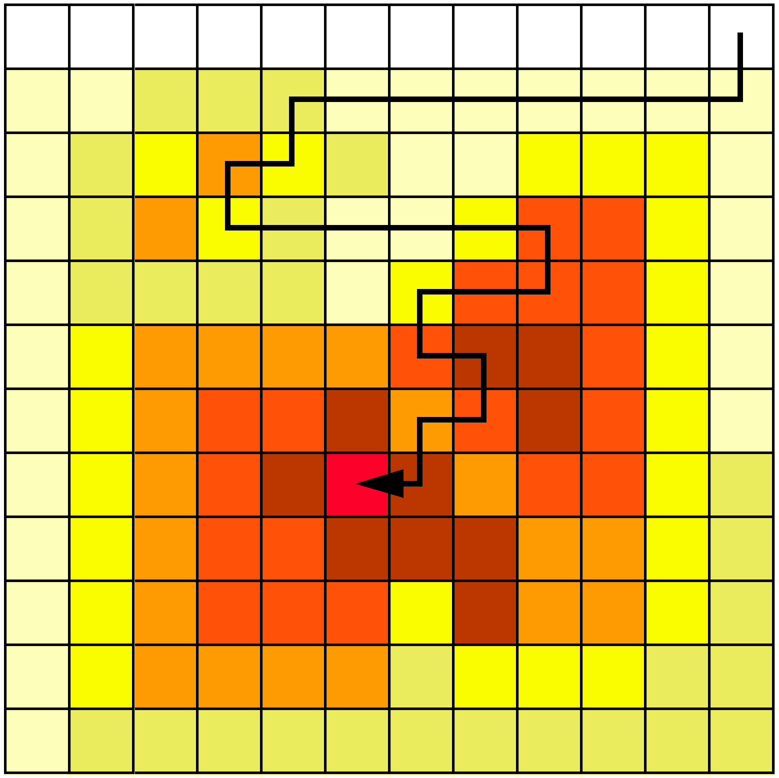
Solutions

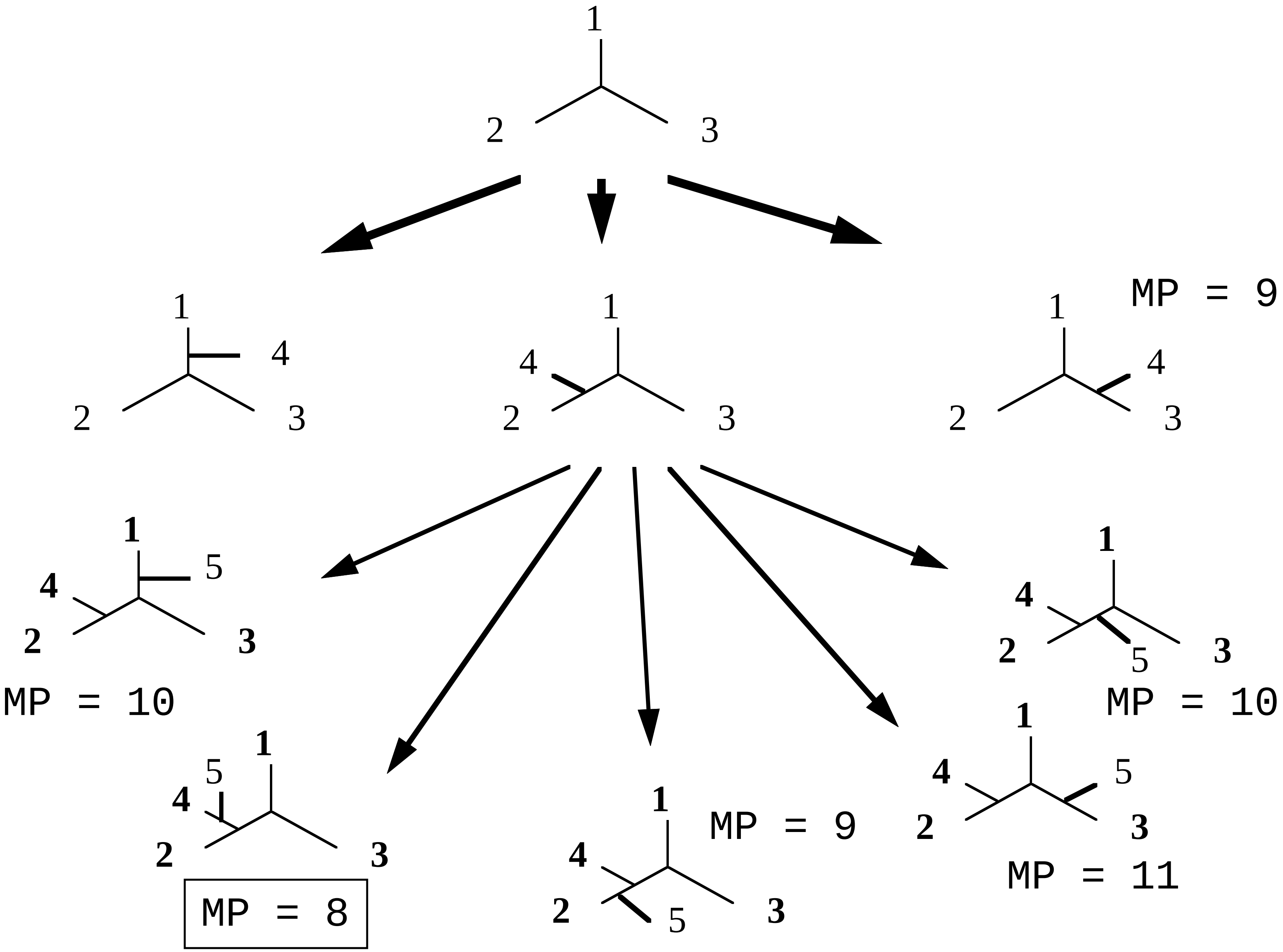
- Exact Solutions
  - Exhaustive search
  - Branch-and-bound search
  - Divide-and-conquer
  - Dynamic programming
- Heuristic Solutions
  - Greedy search
  - Stochastic search
  - Super-duper clever search

Exhaustive search

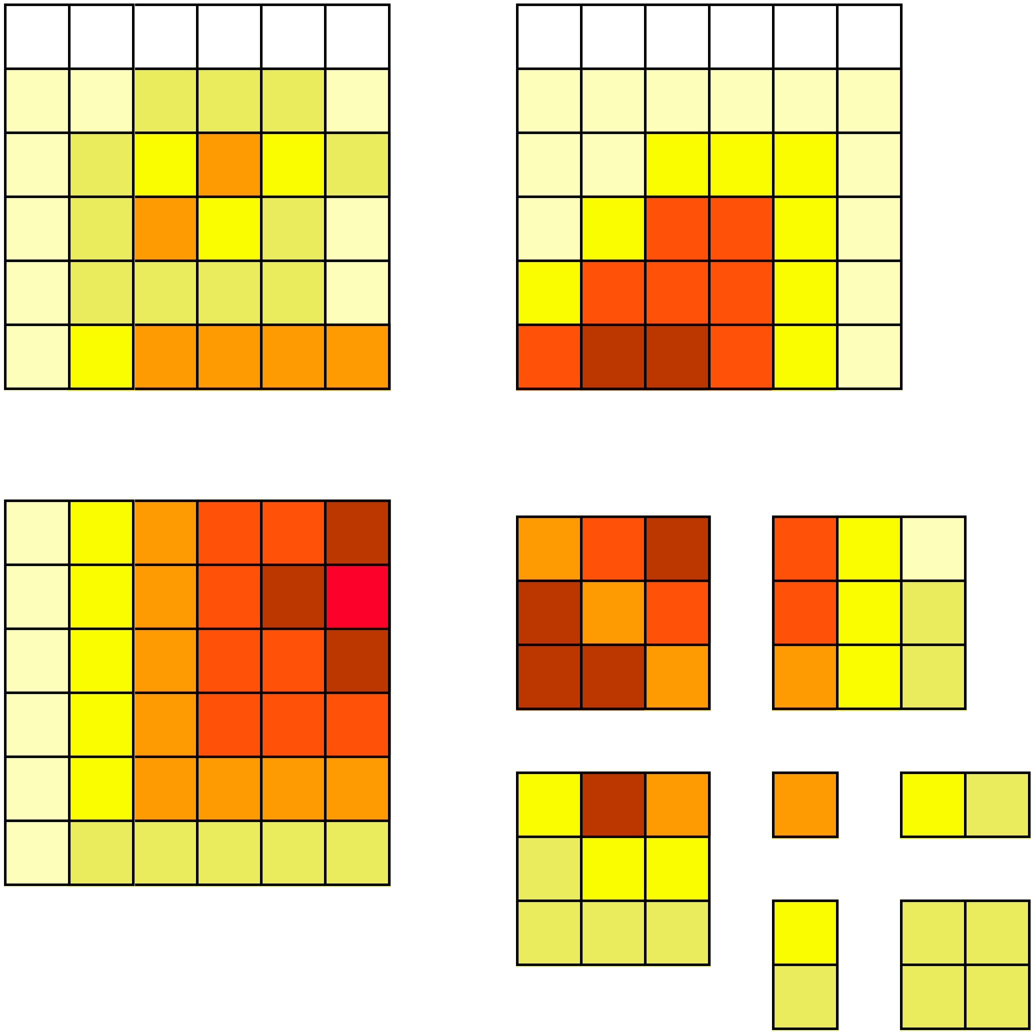


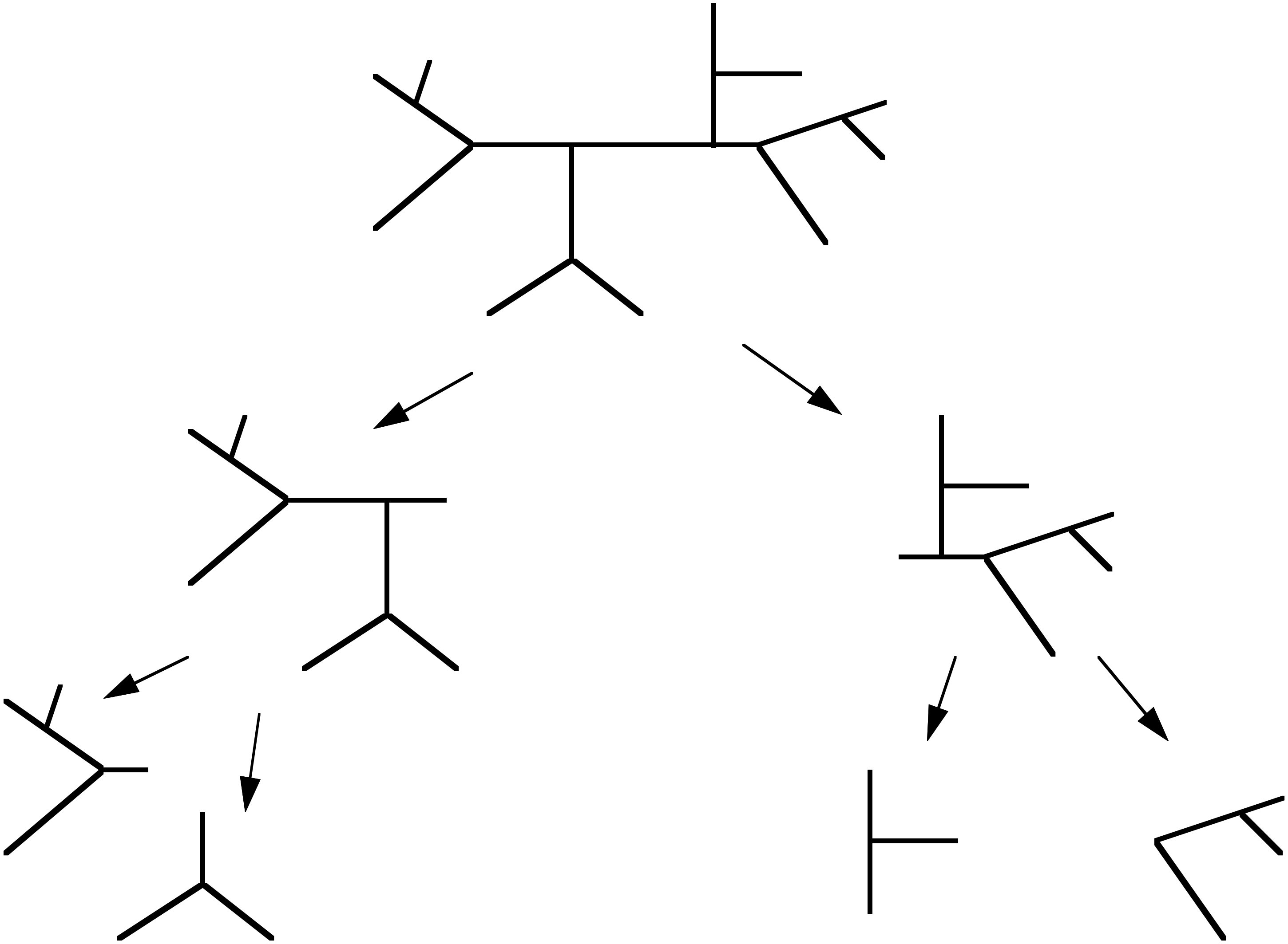
Branch-and-bound search




1
2
3
1
4
2
3
1
4
2
3
MP = 9
1
4
2
3
1
5
4
2
3
MP = 10
1
4
2
5
3
MP = 10
1
5
4
2
3
MP = 8
1
4
2
5
3
MP = 9
1
4
5
2
3
MP = 11

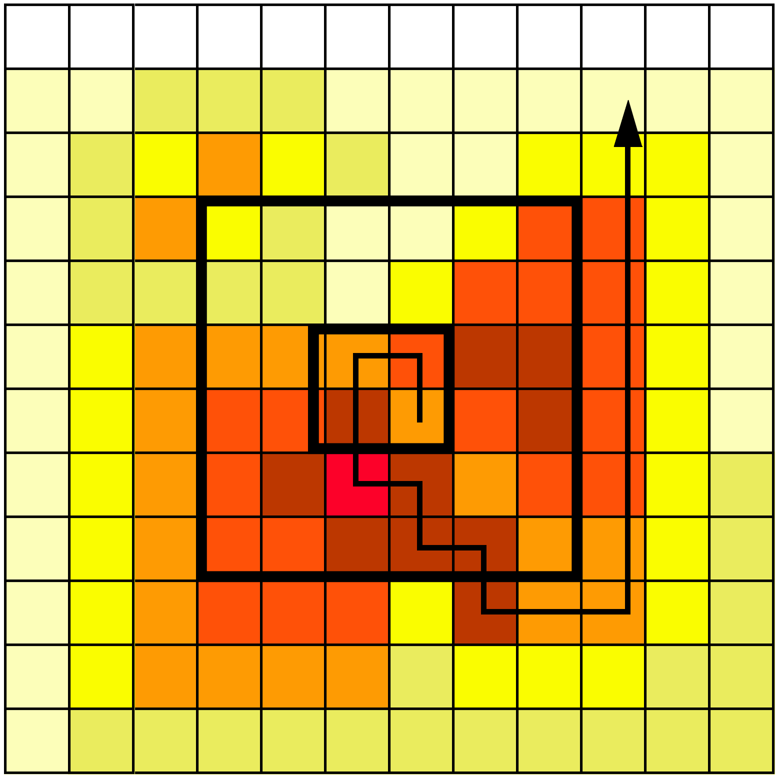
Divide-and-conquer





Quartet based methods

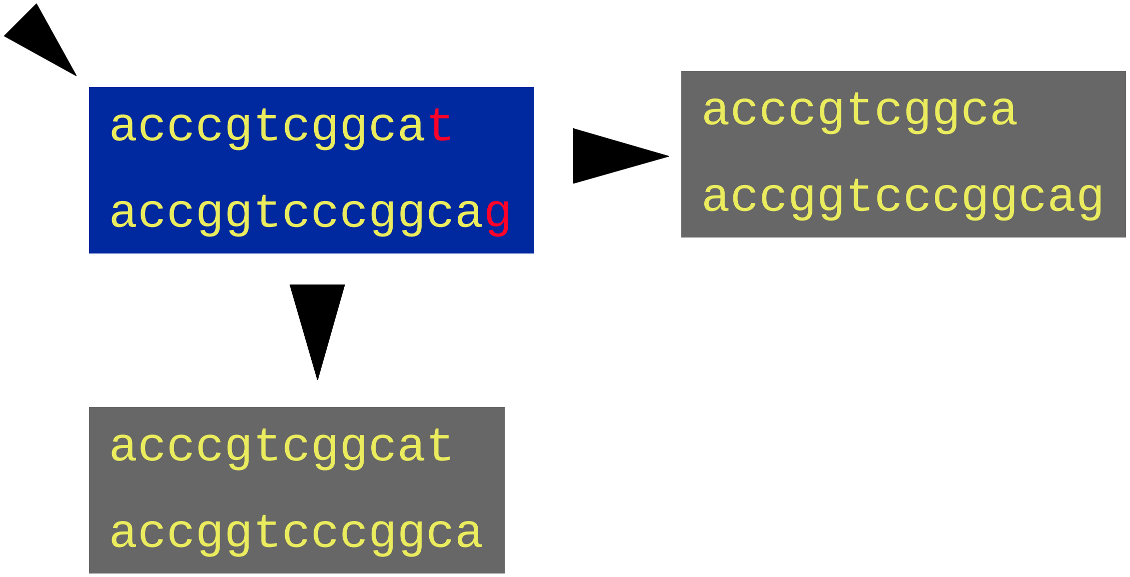
Dynamic Programming



Alignment by dynamic programming

acccgtcggcatg

accggtcccggcagg



acccgtcggcat

accggtcccggcag

acccgtcggca

accggtcccggcag

acccgtcggcat

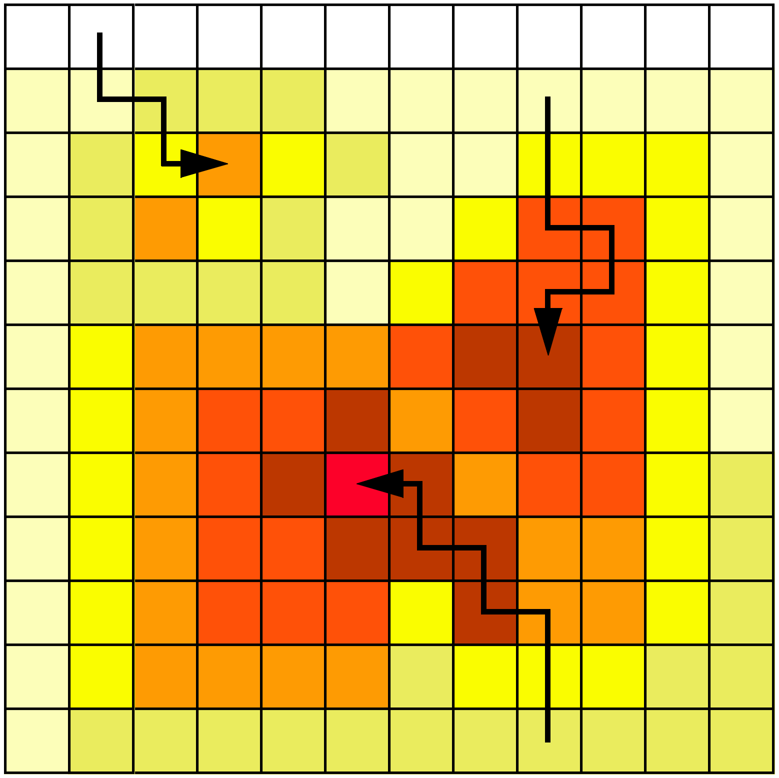
accggtcccggca

Divide-and-conquer and dynamic programming requires that the optimzation of subproblems leads to optimization of the global problem

-> Divide-and-conquer does not yield exact solutions for tree estimation problems

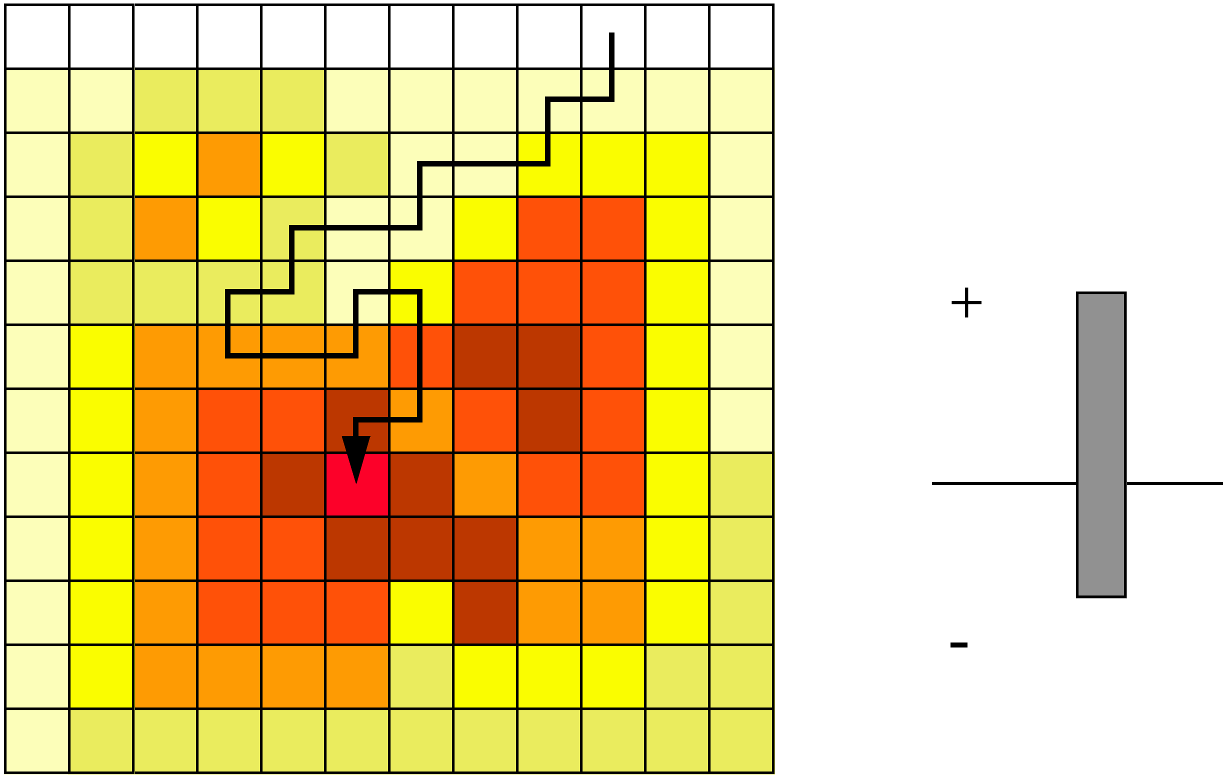
Heuristics

Greedy search

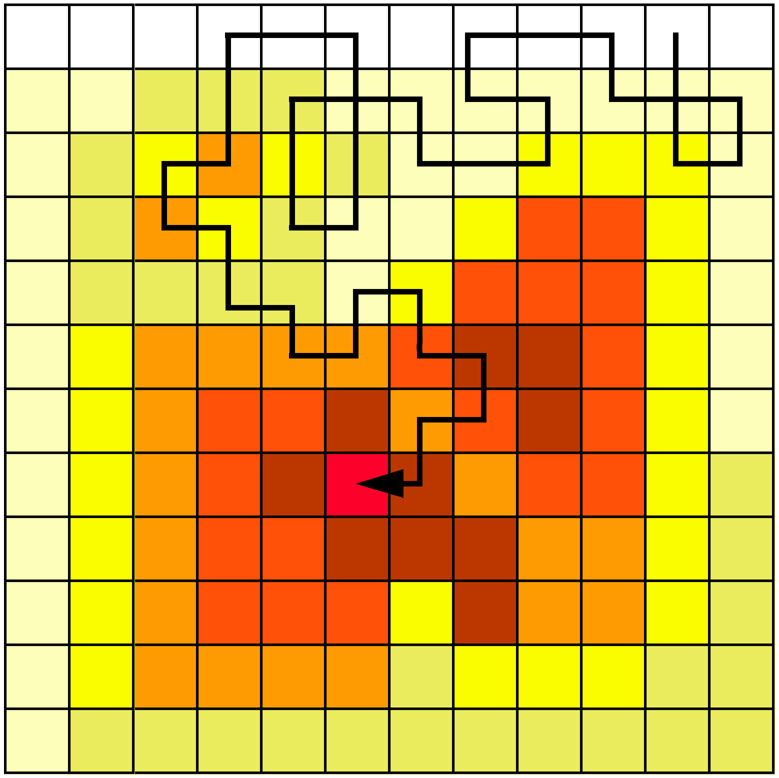


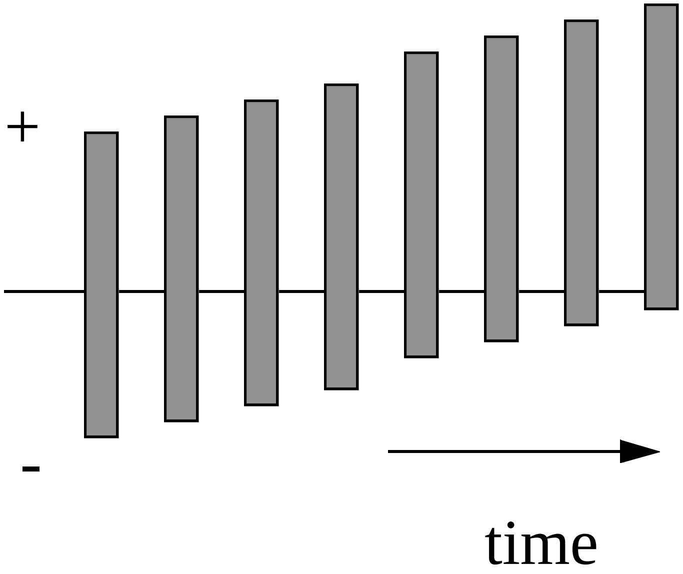
Neighbor-joining, Clustering, Maximum Parsimony, Maximum Likelihood

Stochastic search



Simulated annealing



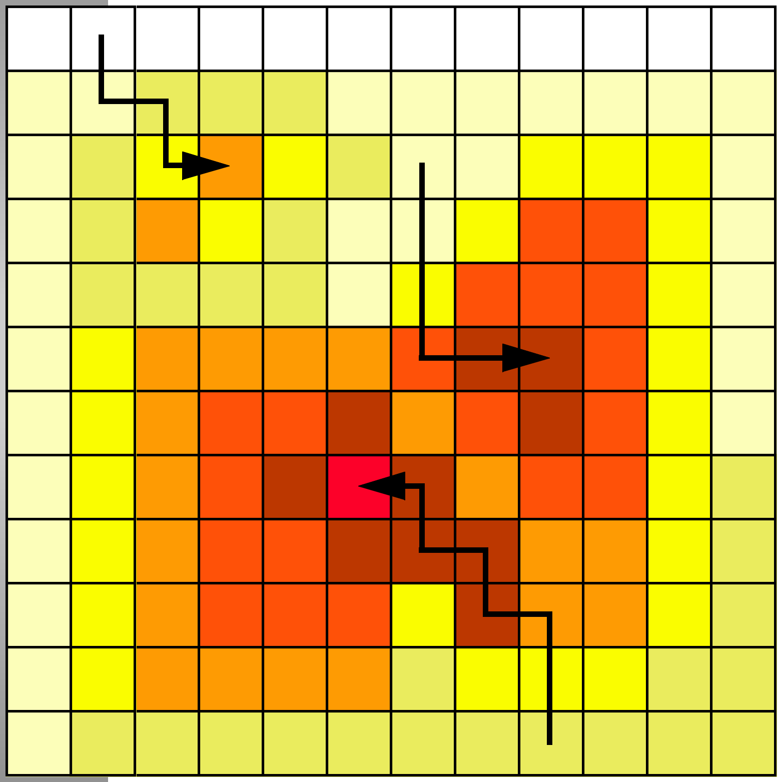


Super-duper clever search

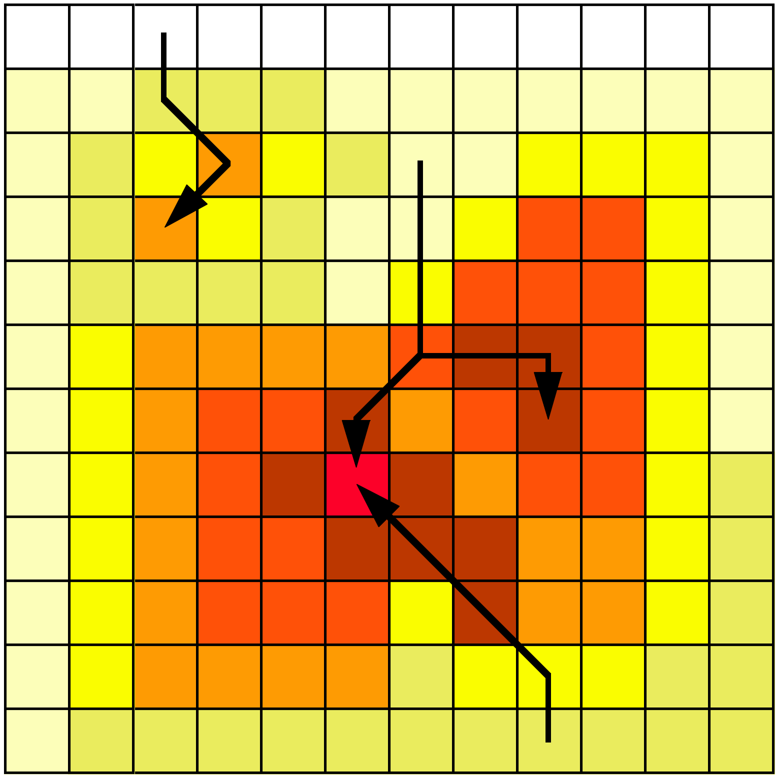
- Genetic algorithms
- Human perception
- Expert knowledge
- ?

Heuristic solutions are dependent on ...

- Initial position
- Configuration space



Starting tree options in PAUP

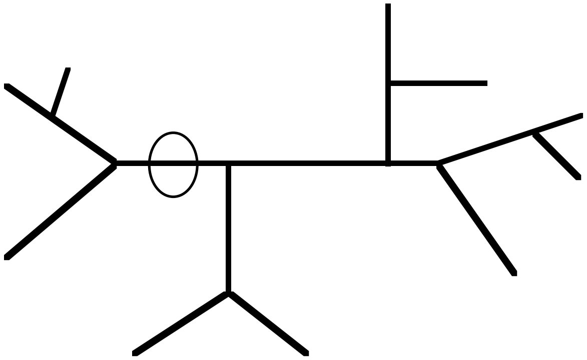


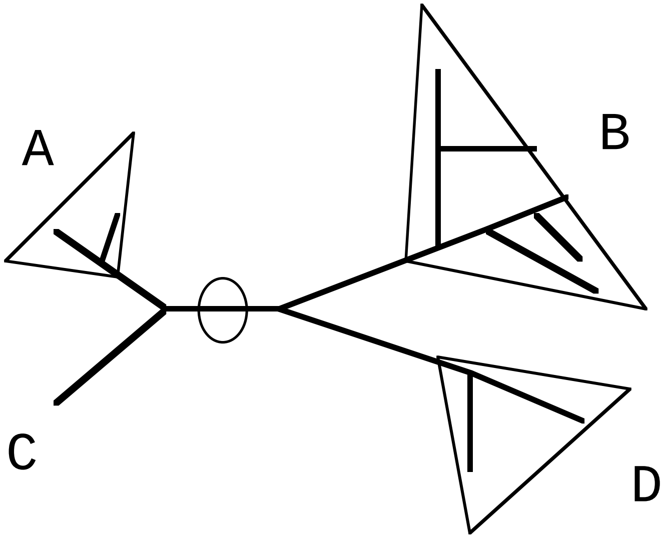
Keep option, Steepest descent option in PAUP

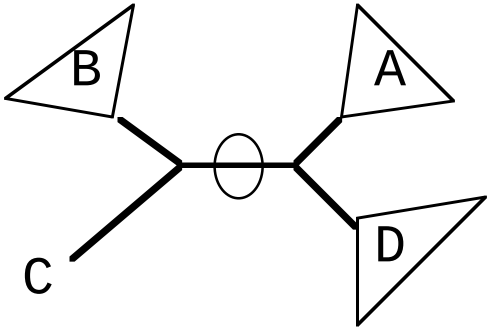
Neighbor relations of trees

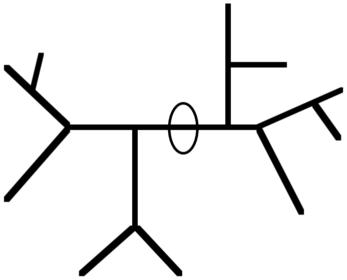
- Nearest-neighbor exchange (NNI)
- Subtree Prune and Regraft (SPR)
- Tree Bisection and Reconnection (TBR)

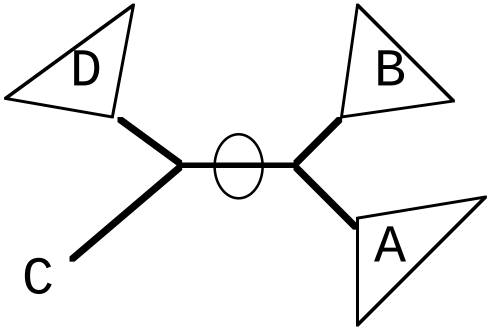
NNI configuration





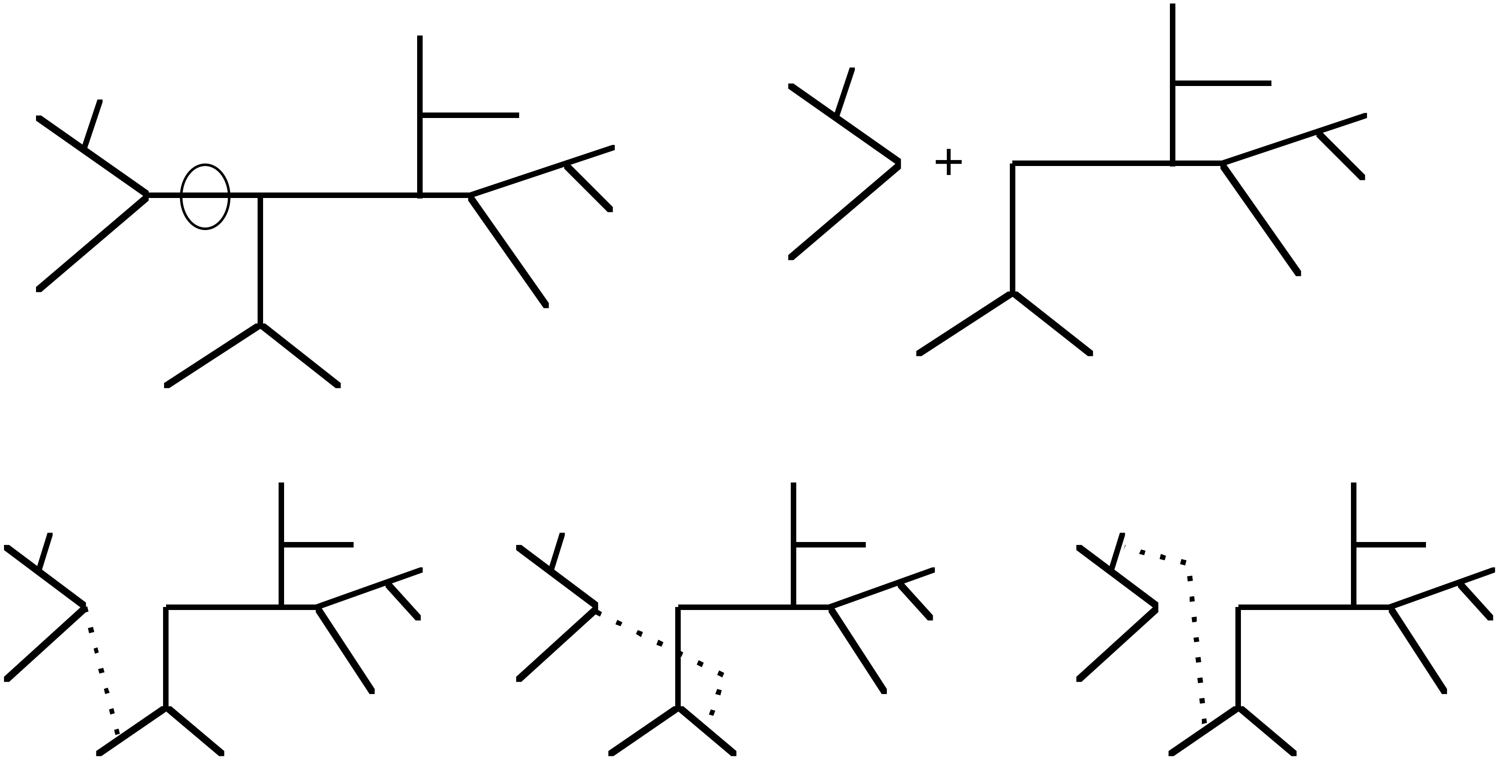




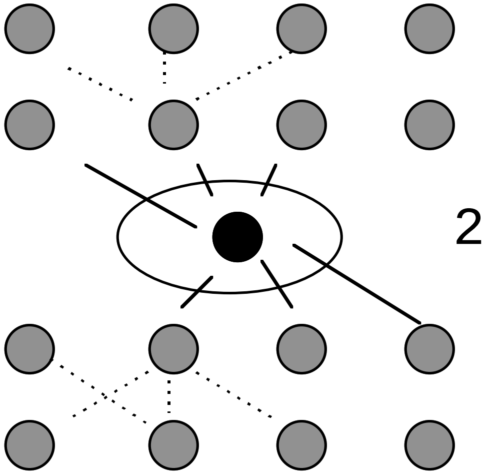


2n-6 Nearst-Neighbor Interchange trees for every tree

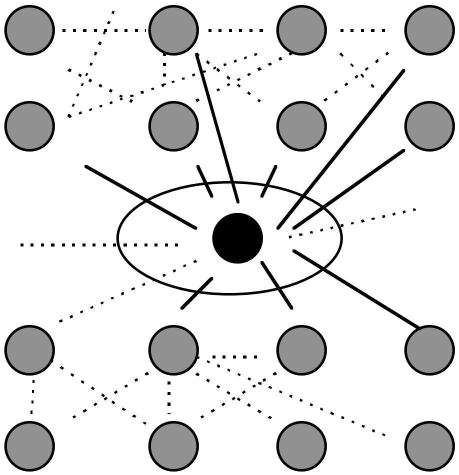
TBR Configuration



~(n-3)n^2 TBR trees for every tree



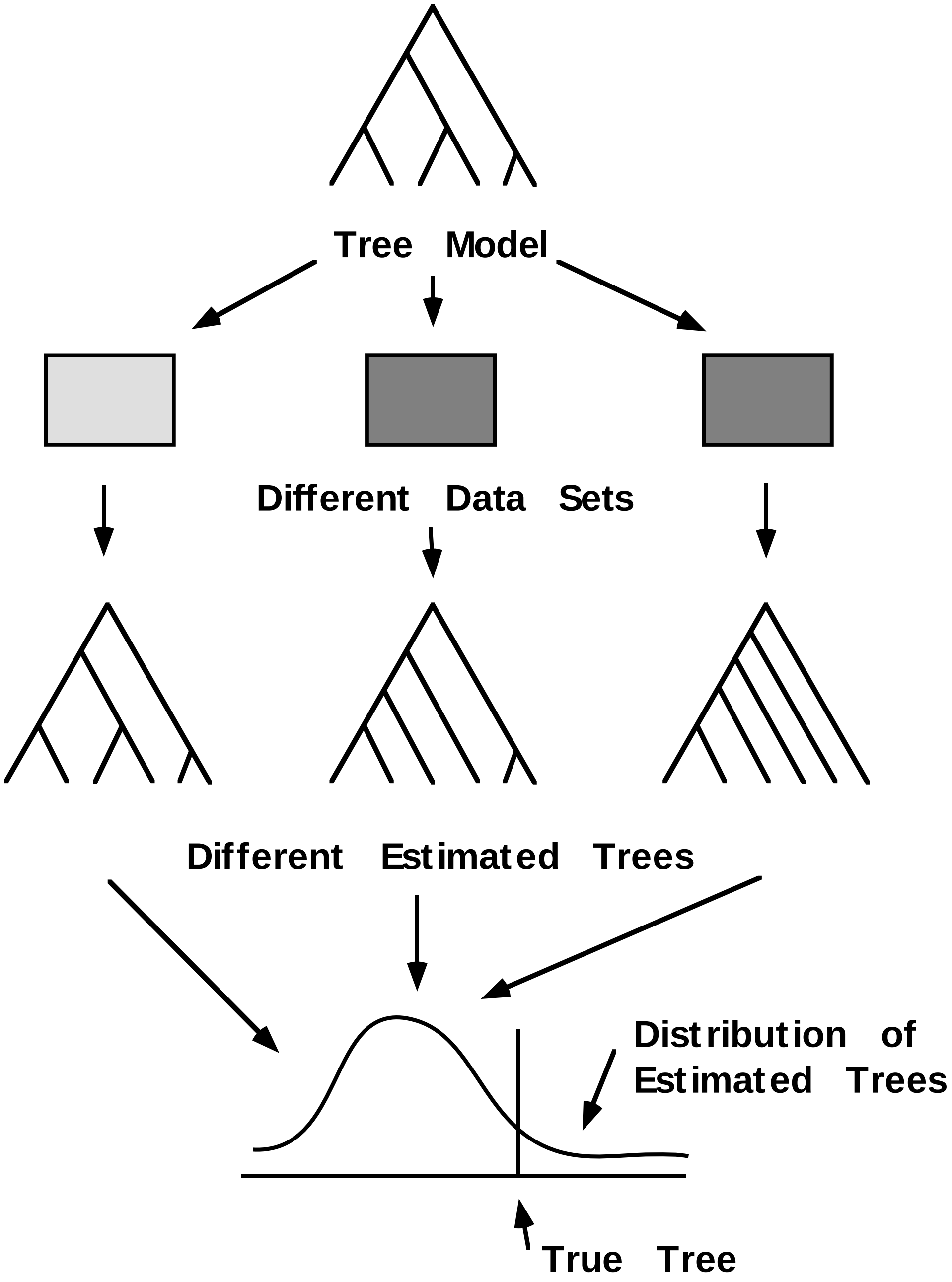
2n-6 connections



$(n-3)n^2$ connections

Statistical Properties

- Accuracy
- Measures of Tree Deviation
- Power and Error
- Confidence Limits



Tree Model

Different Data Sets

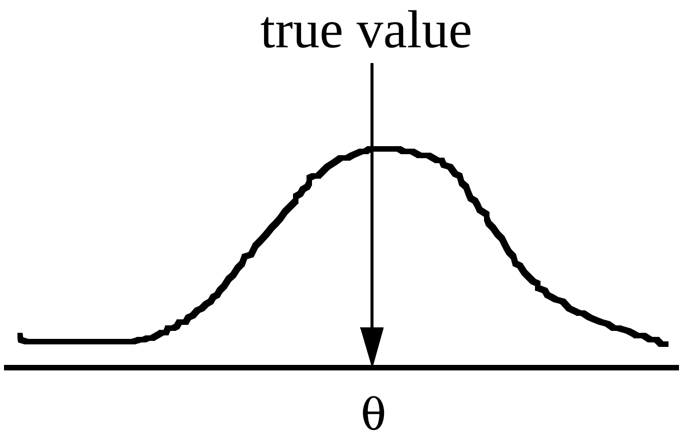
Different Estimated Trees

Distribution of Estimated Trees

True Tree

An estimator has a distribution

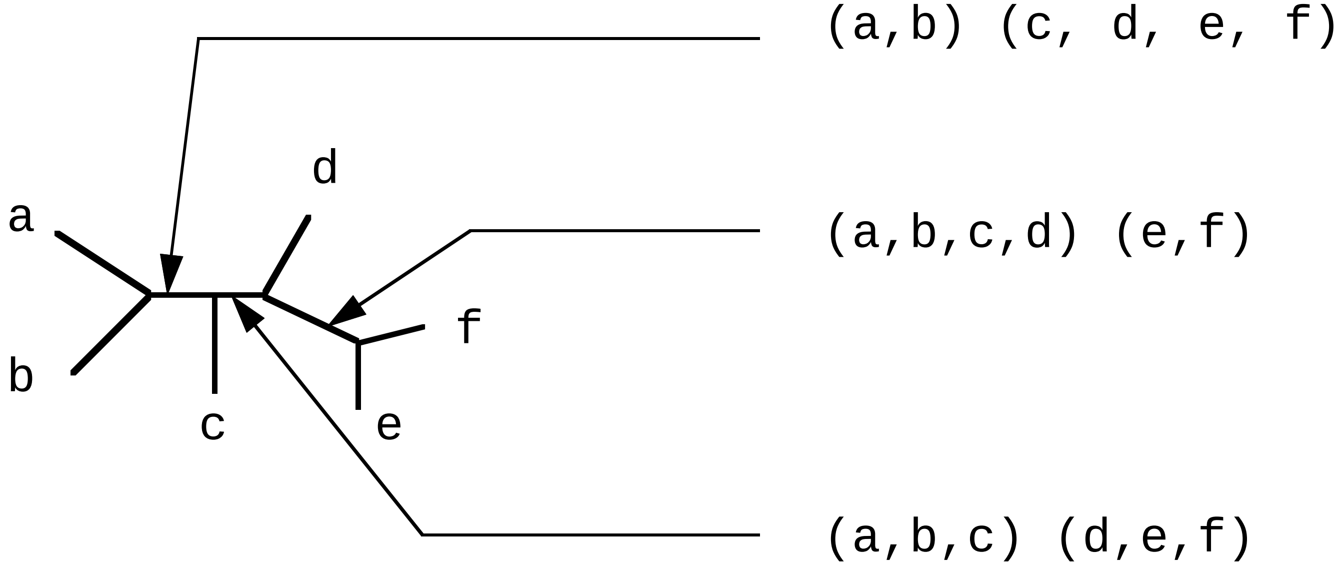
Accuracy is some measurement of the dispersal of the estimator distribution around the "true" value



"Loss function": L(θ, E)

e.g. $(\theta - E)^2$, $|\theta - E|$,
L = 1 if θ = E else L = 0

“θ – E” : We need a way of measuring deviation between trees

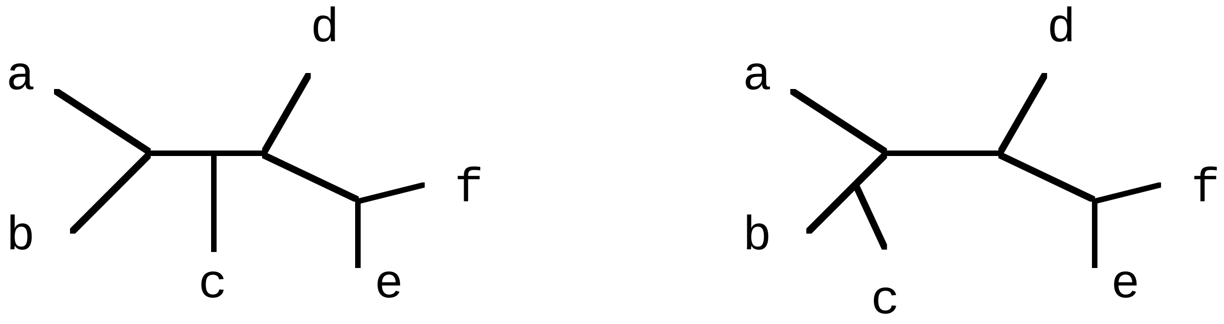


(a,b) (c, d, e, f)

(a,b,c,d) (e,f)

(a,b,c) (d,e,f)

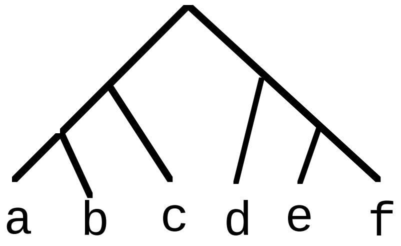
Partition metric

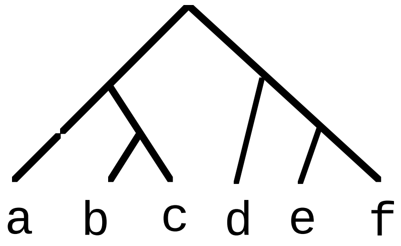


(a,b) (c,d,e,f)	≠	(b,c) (a,d,e,f)
(a,b,c) (d,e,f)	=	(a,b,c) (d,e,f)
(a,b,c,d) (e,f)	=	(a,b,c,d) (e,f)

1 - 2/3 = 0.3333...

Consensus

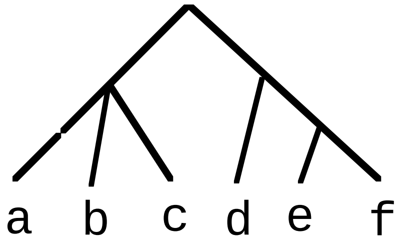




(a,b)	≠	(b,c)
(a,b,c)	=	(a,b,c)
(e,f)	=	(e,f)
(d,e,f)	=	(d,e,f)

↓

(a,b,c),(e,f),(d,e,f)



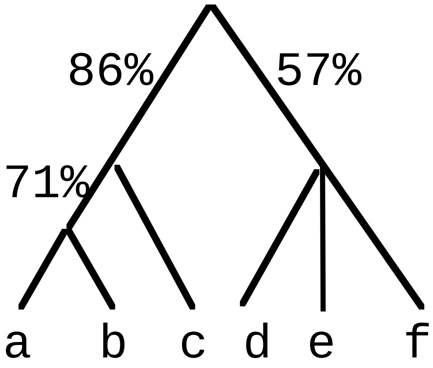
Majority-rule consensus

(a,b)	(a,b)	(a,b)	(a,c)	(a,b)	(a,b)	(a,c)
(a,b,c)	(a,b,c)	(c,d)	(a,b,c)	(a,b,c)	(a,b,c)	(a,b,c)
(e,f)	(a,b,c,d)	(a,b,c,d)	(e,f)	(d,f)	(a,b,c,d)	(d,f)
(d,e,f)	(a,b,c,d,e)	(e,f)	(d,e,f)	(d,e,f)	(a,b,c,d,e)	(d,e,f)

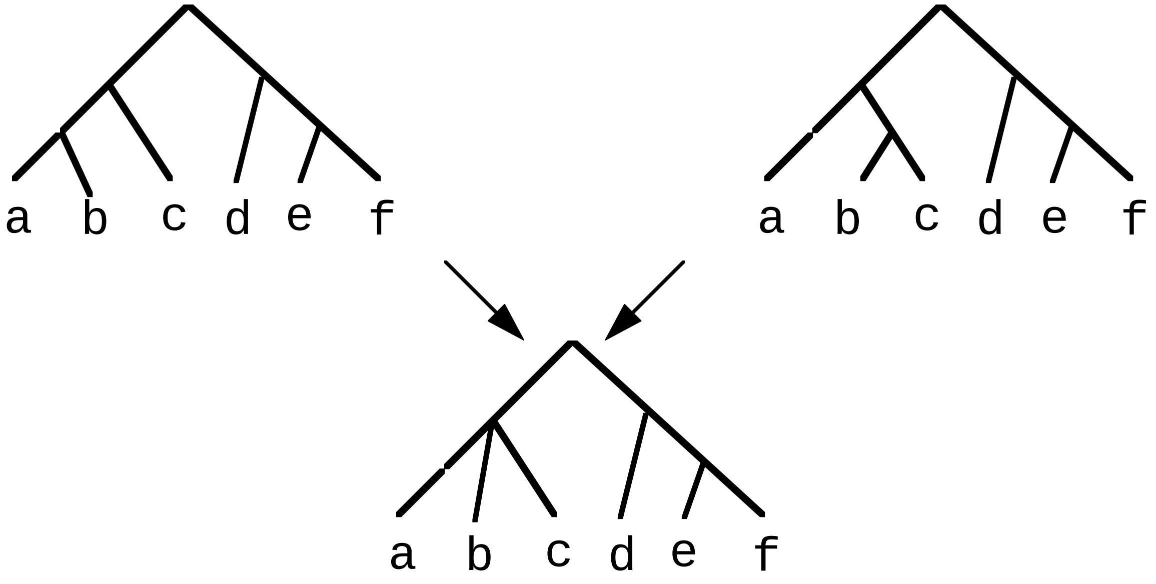
(a,b): 5 (a,c): 2 (c,d): 1 (d,f): 2 (e,f): 3
(a,b,c): 6 (d,e,f): 4 (a,b,c,d): 2 (a,b,c,d,e): 2

> 3.5

(a,b)
(a,b,c)
(d,e,f)



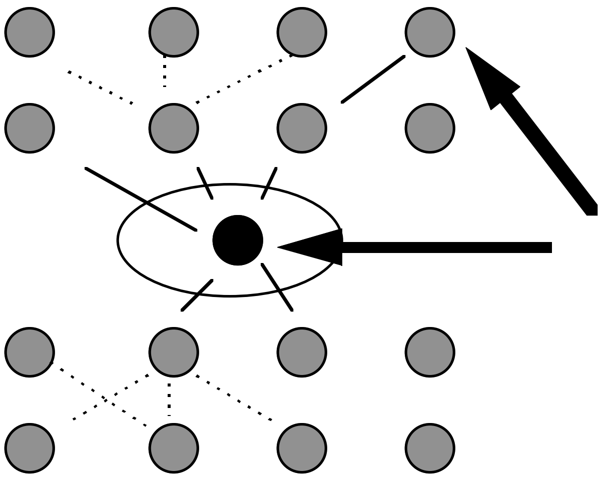
Consensus tree can be used to define a deviation measure



3/4 clades resolved
therefore, distance = 1 - 3/4 = 1/4

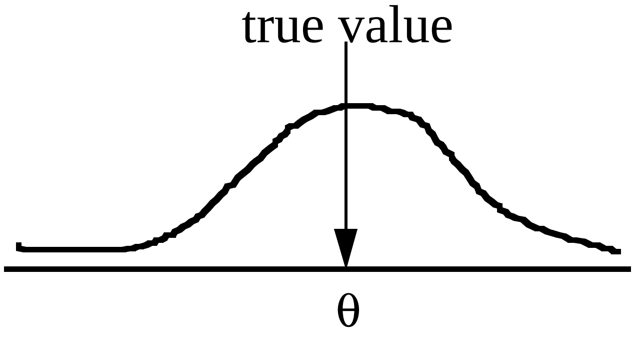
Tree neighbor relations can be used to define deviation

Nearst-Neighbor Interchange (NNI)
configuration



Related by 2 consecutive
NNI operations.
Therefore, distance = 2

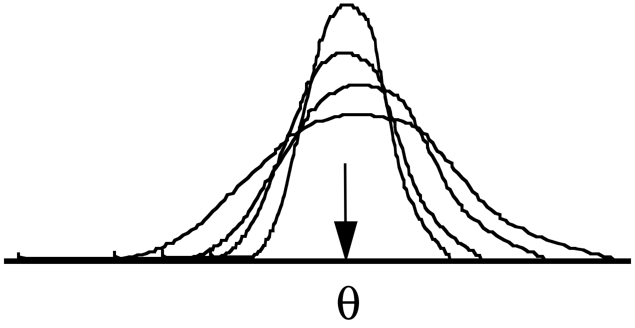
Once we settle on a suitable deviation measure, we can compute the expectation of the loss function as a measure of accuracy

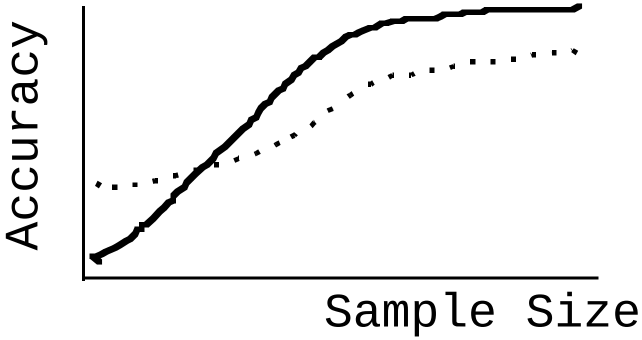


$$\int L(\theta, E)\, dE \xrightarrow[e.g.]{} \int (\theta - E)^2\, dE$$

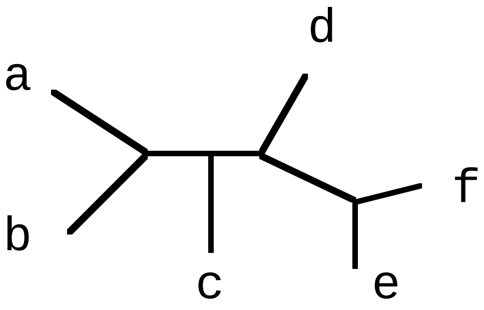
But...

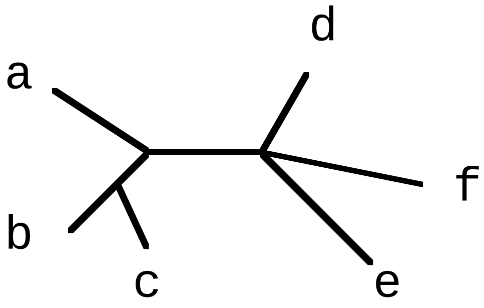
there is a different distribution for every different sample size (number of characters) ... therefore, accuracy is a function of the sample size





Power and Error



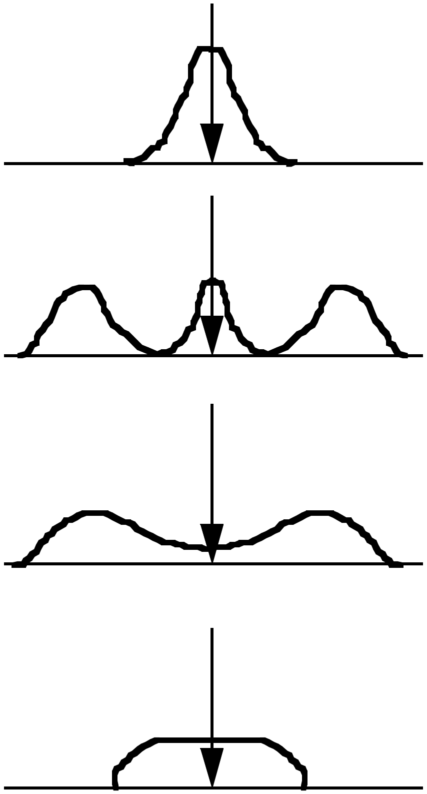


False Negatives: Branches (or clades) in the true tree not in the estimated tree -> Power

False Positives: Branches (or clades) in the estimated tree not in the true tree -> Error

Power, Error, and Accuracy are not necessarily related to each other

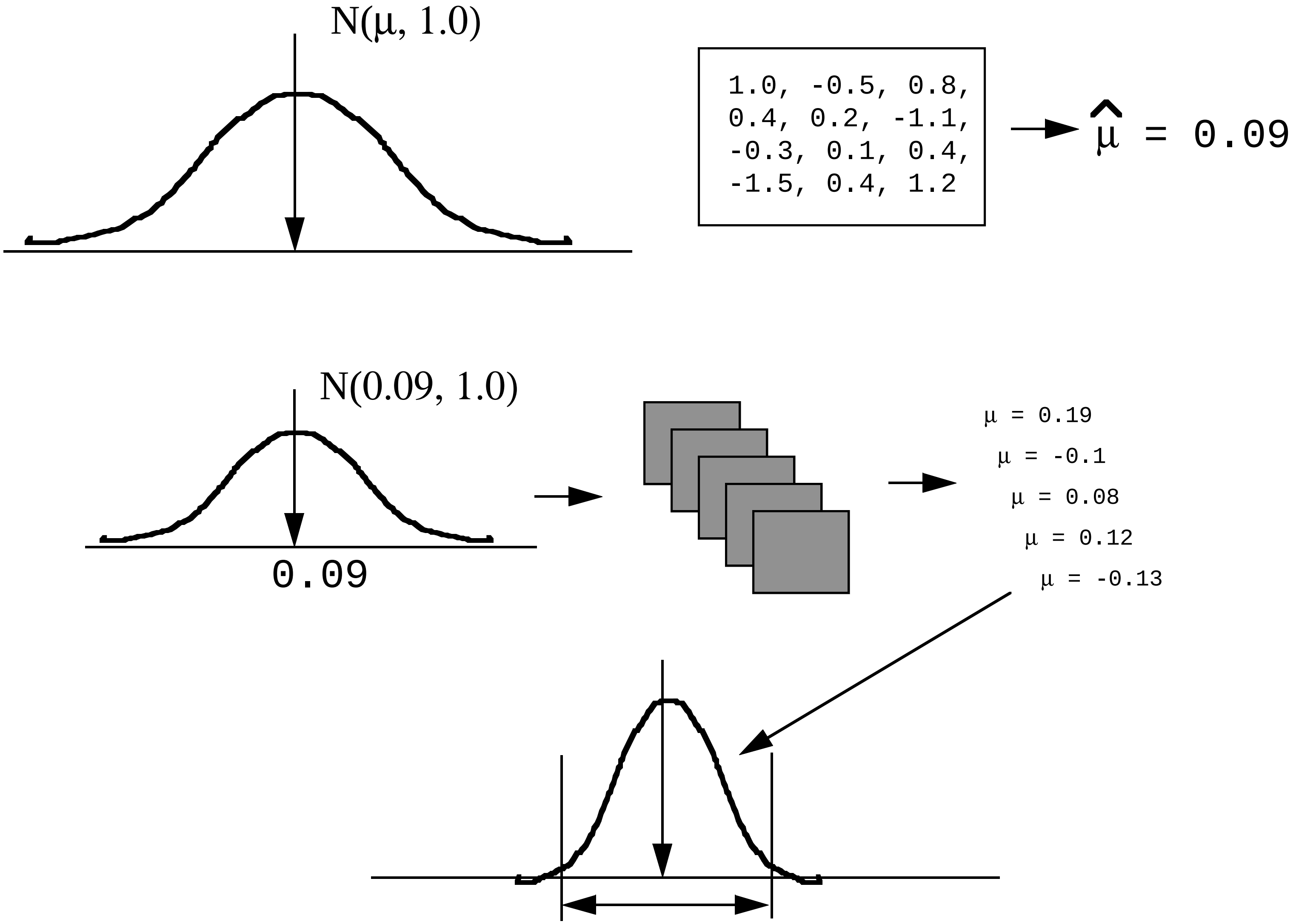
Power	Error	Accuracy
High	Low	High
High	High	Low
Low	High	Low
Low	Low	Low



Confidence Limits

What we would like to say: Given some data and an estimate of the model, the probability that the estimate is correct

What we can say: Given some estimate of the model and **assuming that the estimate is correct**, what is the probability distribution of the estimator



N(μ, 1.0)

1.0, -0.5, 0.8,
0.4, 0.2, -1.1,
-0.3, 0.1, 0.4,
-1.5, 0.4, 1.2

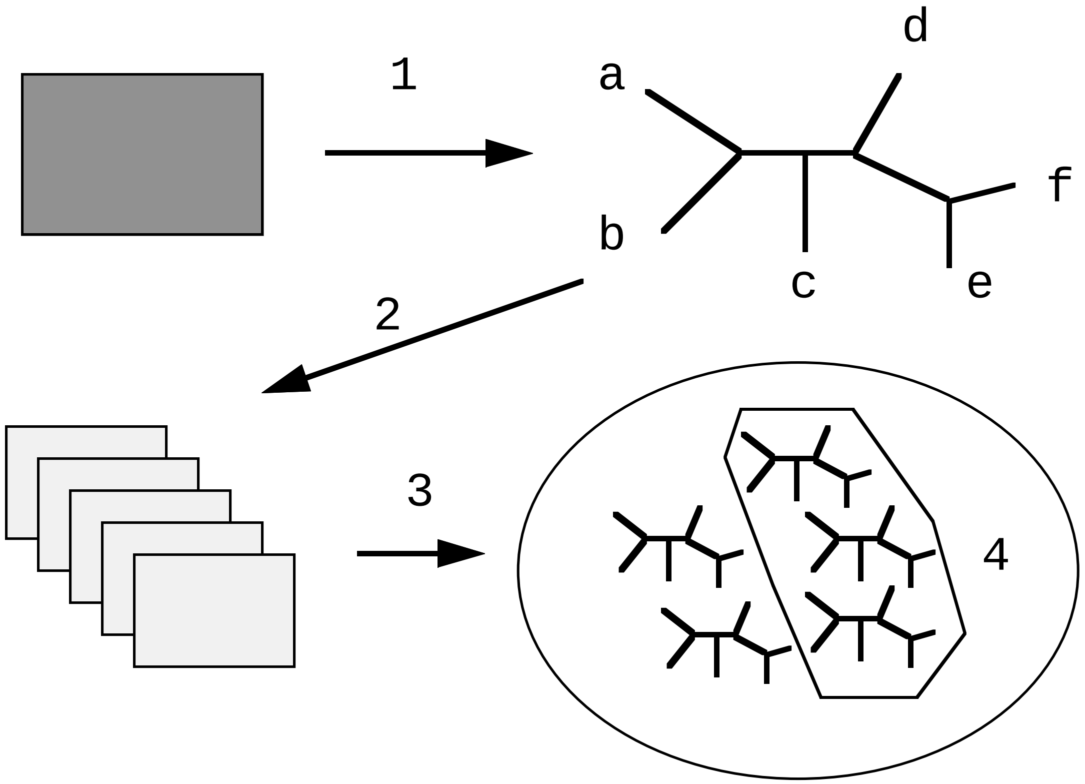
$\hat{\mu} = 0.09$

N(0.09, 1.0)

0.09

μ = 0.19
μ = -0.1
μ = 0.08
μ = 0.12
μ = -0.13

95% Confidence Limit of the estimator



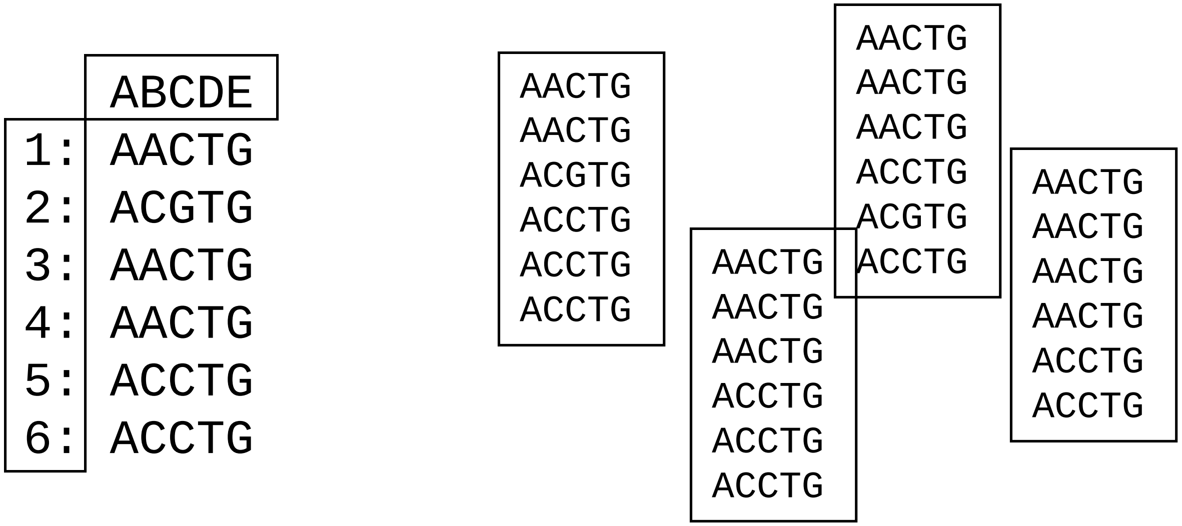
1. Estimate

2. Assume the estimate is correct and generate replicate samples

3. Estimate from the replicate samples

4. Decide on a confidence set

Bootstrap resampling as a means of generating replicate samples (step 2)



```
   ABCDE
1: AACTG
2: ACGTG
3: AACTG
4: AACTG
5: ACCTG
6: ACCTG
```

```
AACTG
AACTG
ACGTG
ACCTG
ACCTG
ACCTG
```

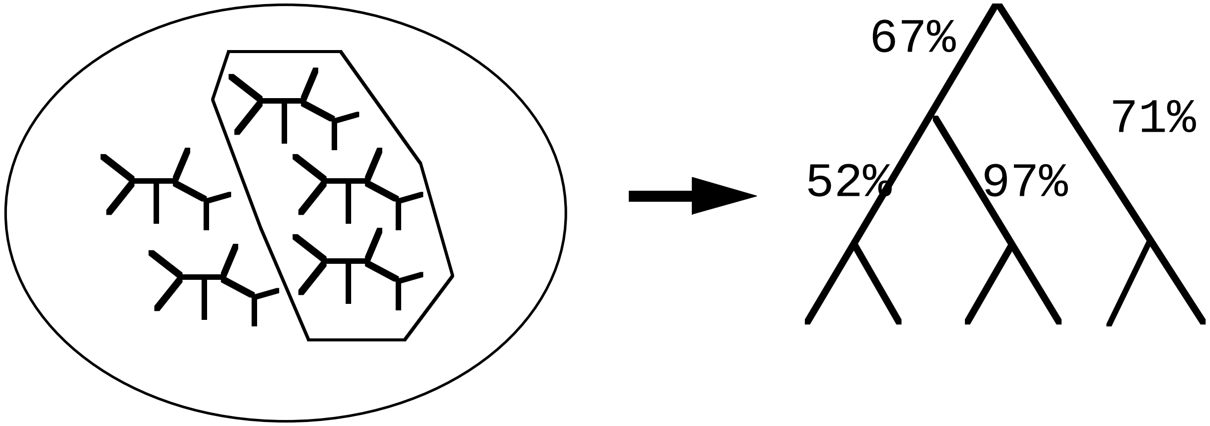
```
AACTG
AACTG
AACTG
ACCTG
ACCTG
ACCTG
```

```
AACTG
AACTG
AACTG
ACCTG
ACGTG
ACCTG
```

```
AACTG
AACTG
AACTG
AACTG
ACCTG
ACCTG
```

Samples are generated by “drawing” characters with probability proportional to their observed frequency -> We assume the observed frequency to be the “true” probability of drawing characters.

Majority-rule consensus trees can be used to select confidence sets (step 4)



Misc. confidence limites

- Skewness index
- Decay index
- T-PTP test
- Parametric bootstrapping
- Whatever...