An exact and polynomial distance-based algorithm to reconstruct single copy duplication trees

Olivier Elemento (1) and Olivier Gascuel (2)

(1) Princeton University, Tavazoie group, http ://www.genomics.princeton.edu

(2) LIRMM, Methods and Algorithms in Bioinformatics group, http ://www.lirmm.fr

Tandemly repeated sequences

- two or more adjacent and approximate copies of a same DNA sequence
- the successive duplications form a tandem duplication tree
- we seek to reconstruct the duplication tree of tandemly repeated sequences

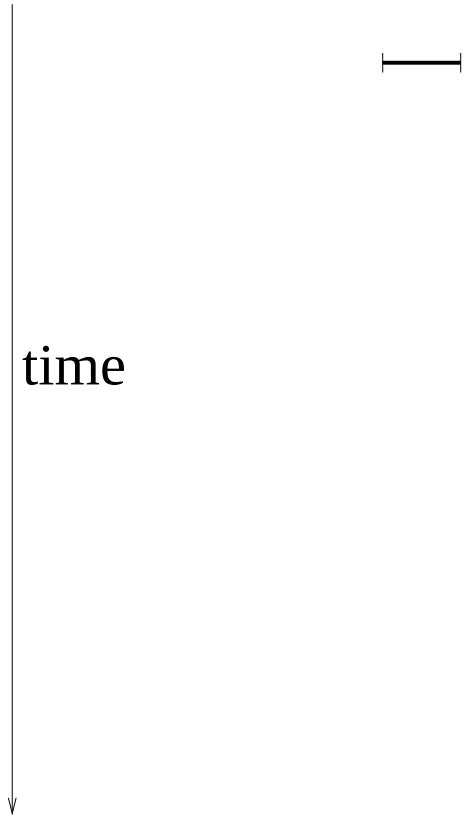
Reconstructing single copy tandem duplication trees

- Jaitly et al, 2002 : the problem of reconstructing single copy duplication trees is NP-hard, in a parsimony framework
- Benson et al, 1999 : heuristic algorithms using the parsimony criterion
- Tang et al, 2002 ; Jaitly et al, 2002 : PTAS using DP in a parsimony framework

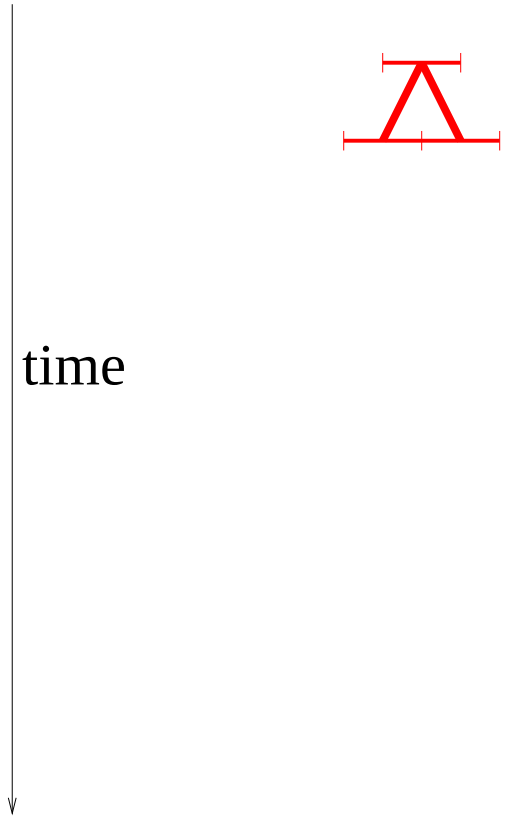
Outline

- Tandem duplication trees (DT)
- Minimum Evolution principle
- recurrence equation for estimating the length of a DT given a matrix of pairwise distances using Ordinary Least Squares
- use this equation in a DP framework to find the optimal DT (the tree with shortest length)

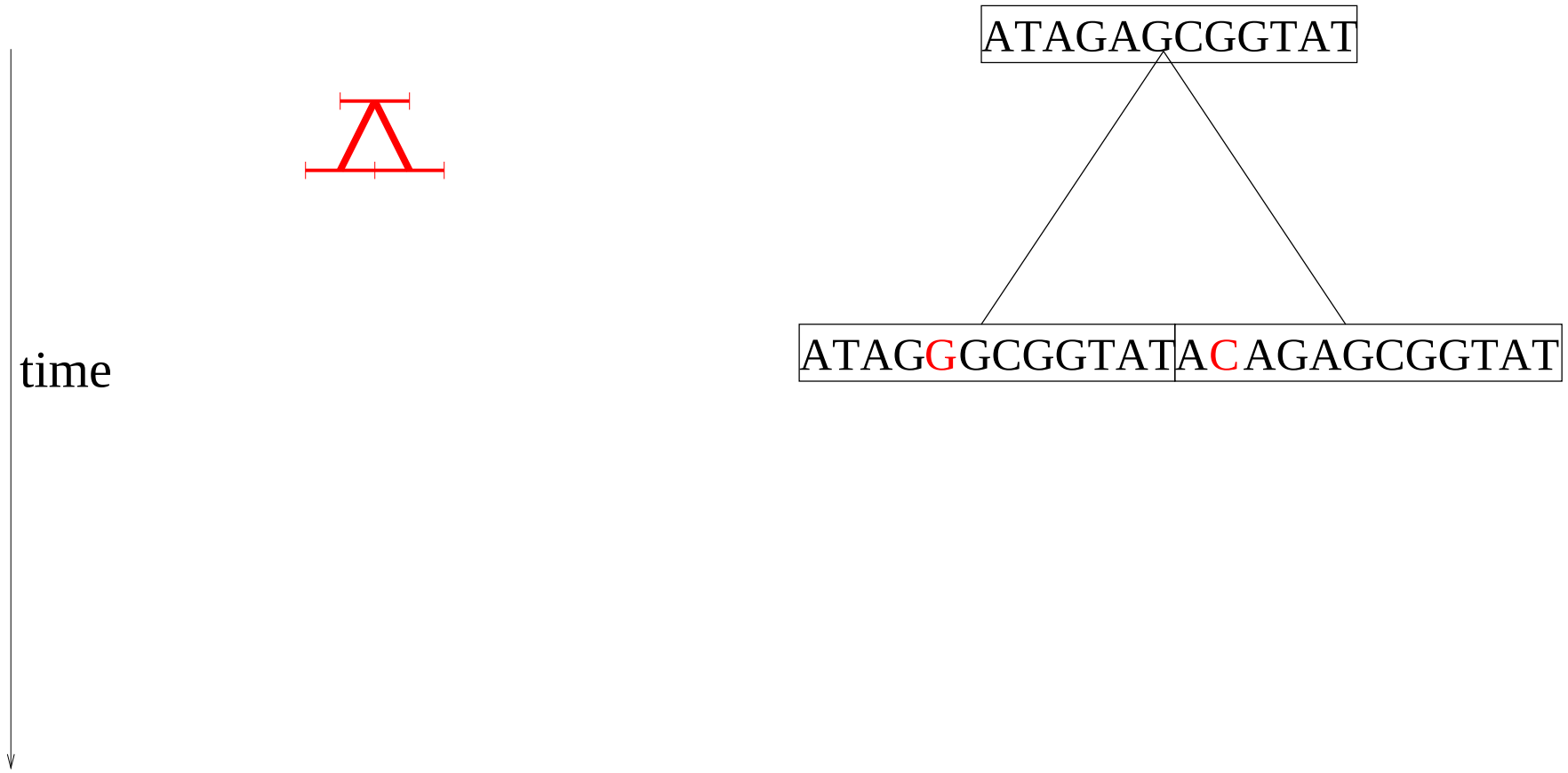
Tandem duplication trees



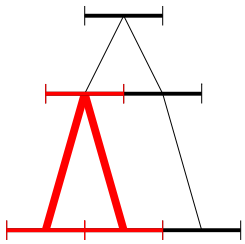
Tandem duplication trees



Tandem duplication trees



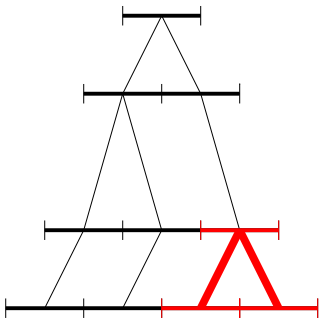
Tandem duplication trees



time

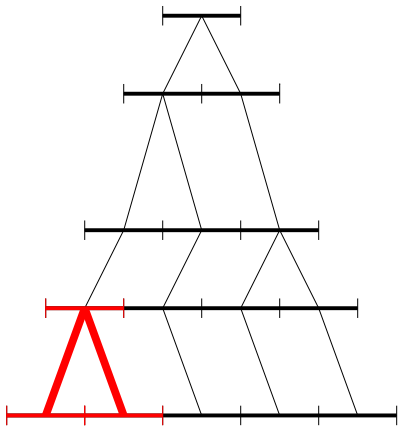
Tandem duplication trees

time

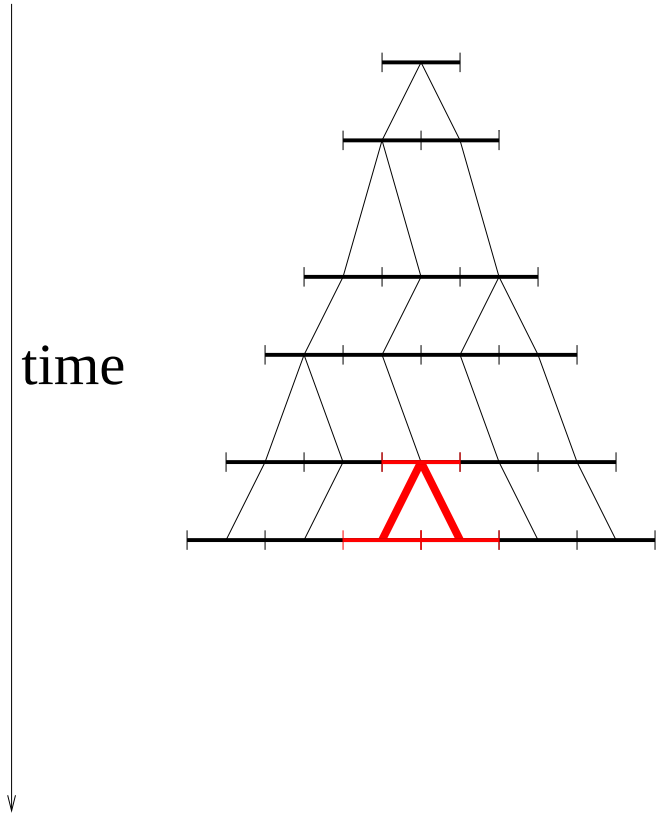


Tandem duplication trees

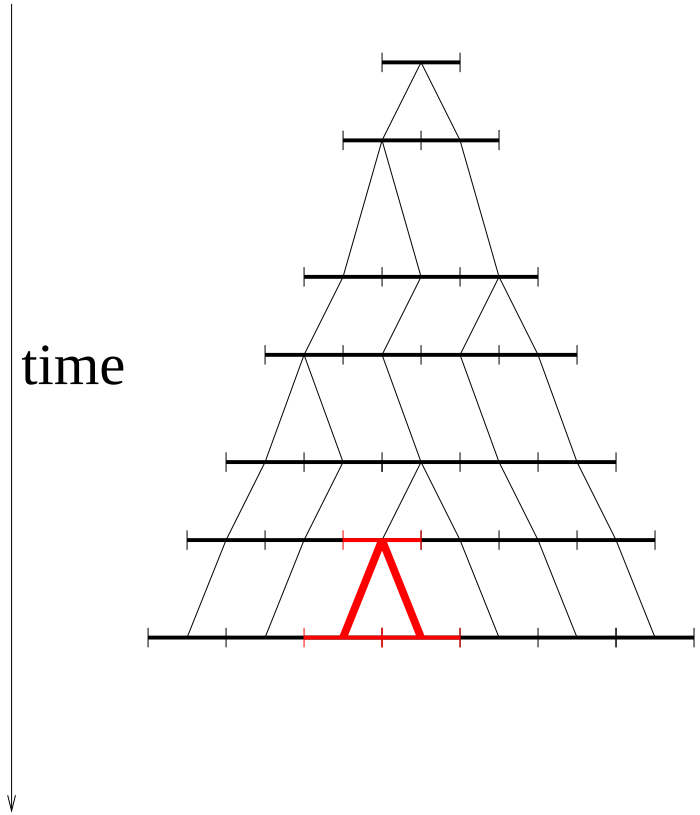
time



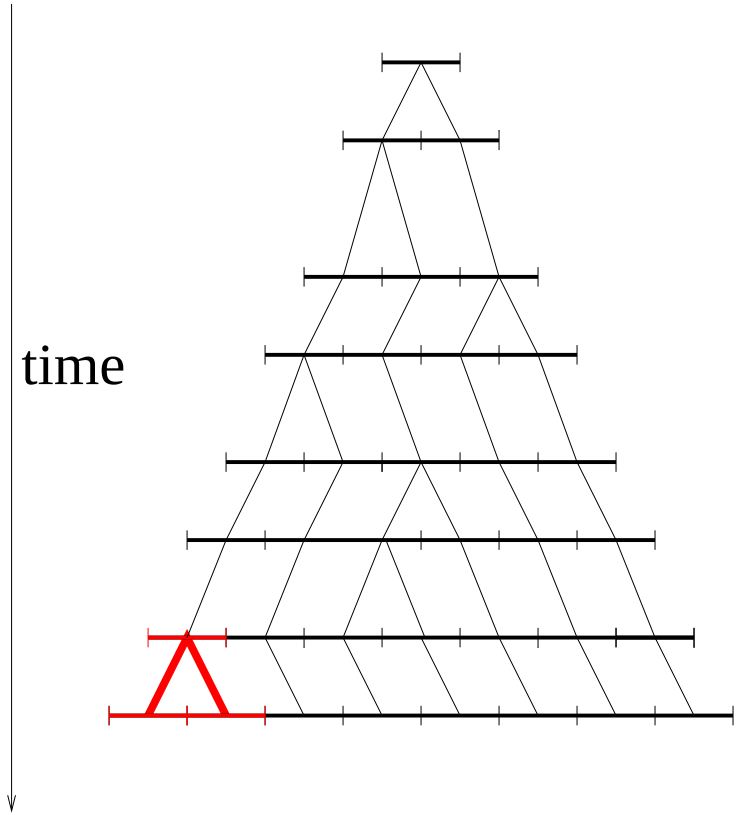
Tandem duplication trees



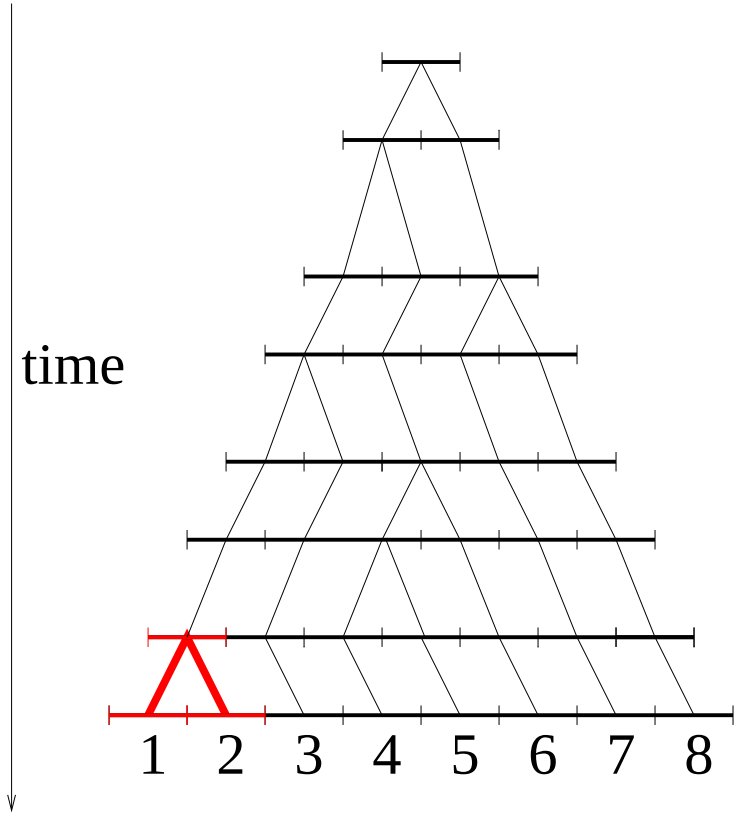
Tandem duplication trees



Tandem duplication trees

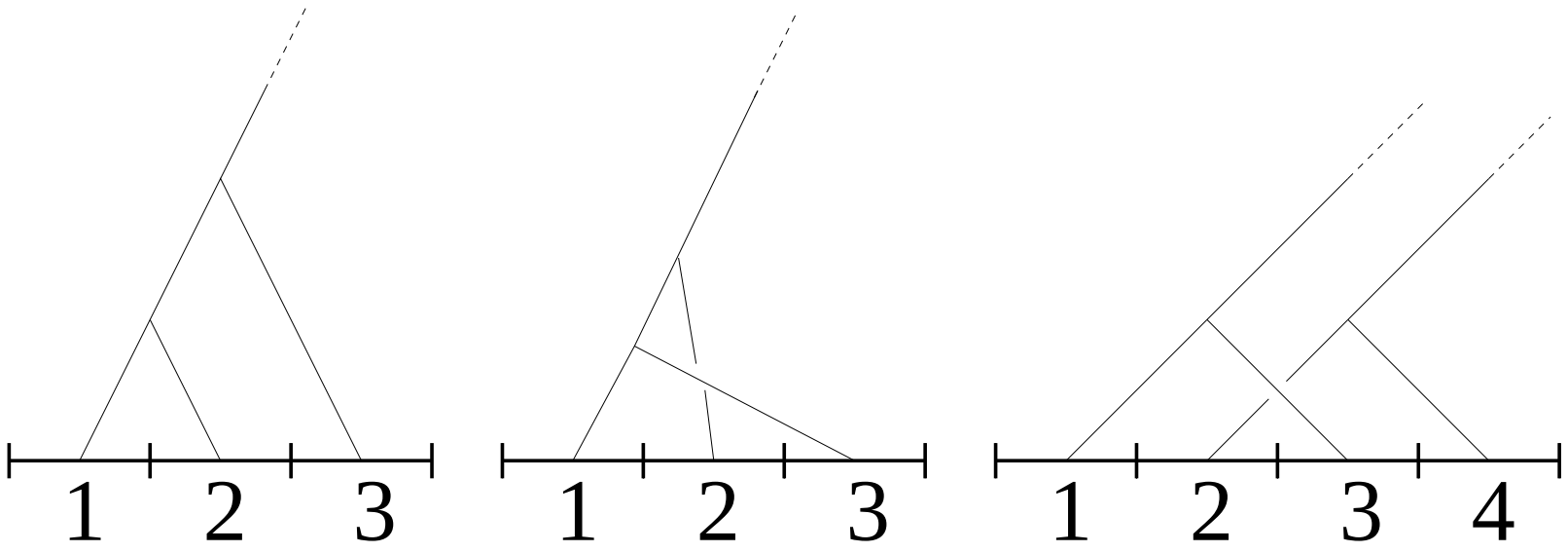


Tandem duplication trees

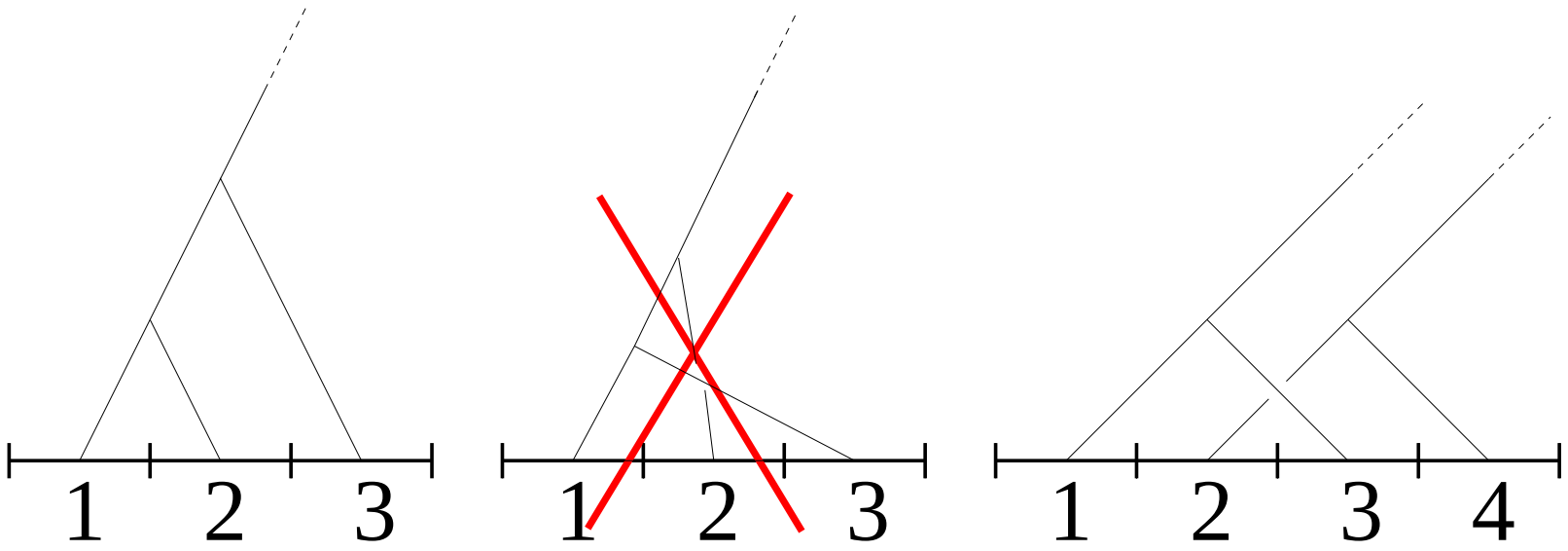


- induces a phylogeny
- order between the leaves
- not all duplications are allowed
- only single copy tandem duplications

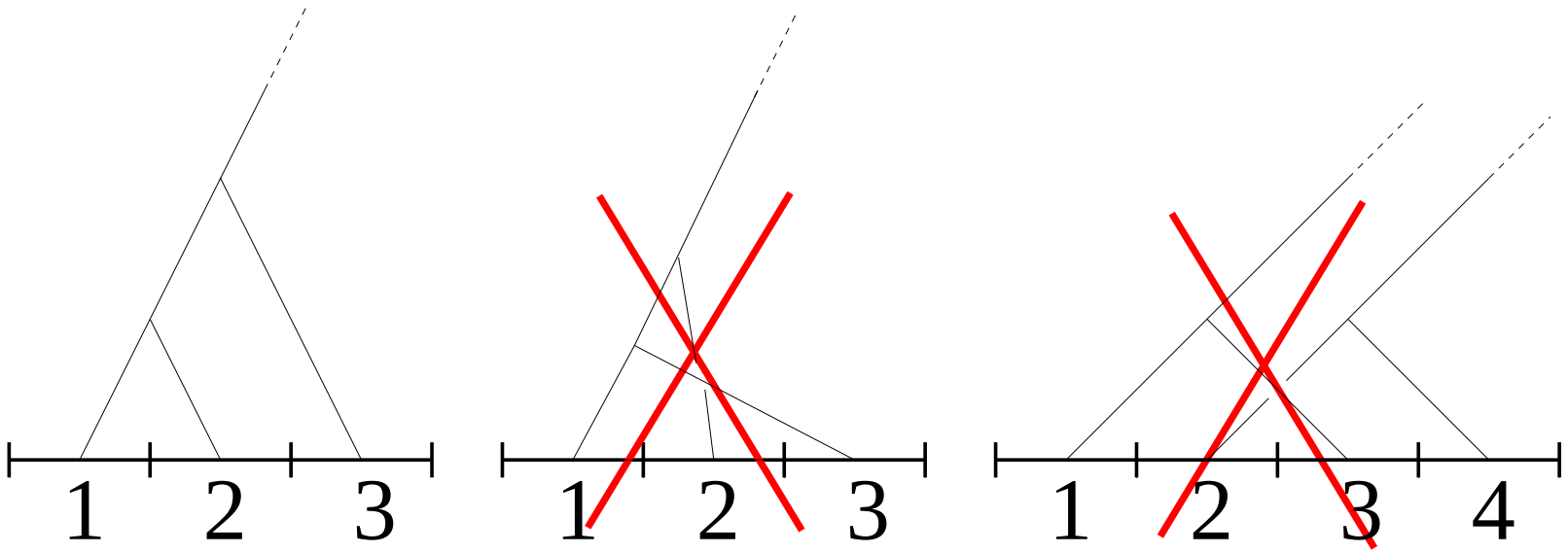
Tandem duplication trees



Tandem duplication trees

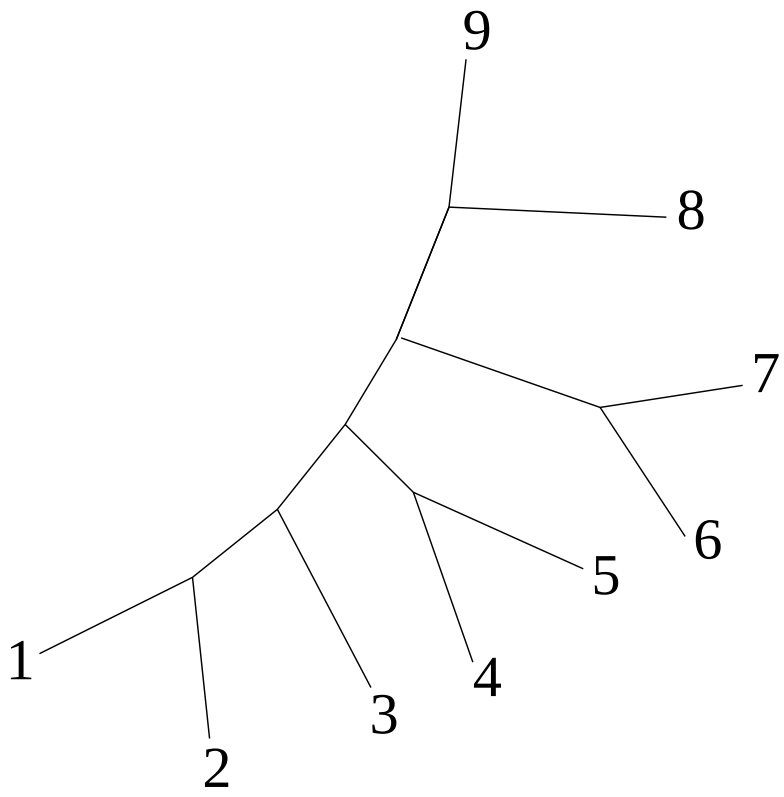


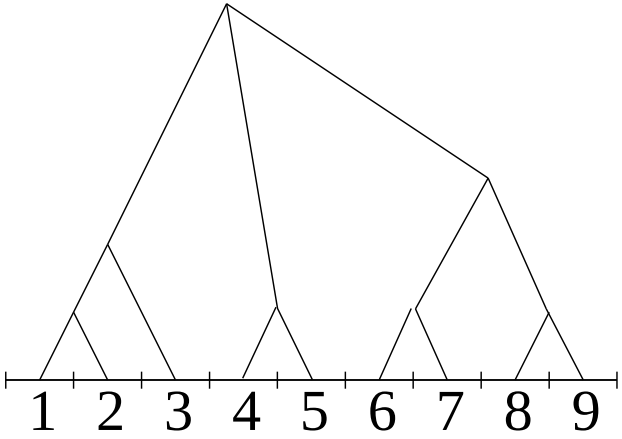
Tandem duplication trees



Tandem duplication trees

- we can only recover an unrooted DT (no molecular clock)





Counting single copy duplication trees

- the Catalan recursion (binary search trees)

$$C_n = \sum_{k=1}^{n-1} C_k C_{n-k} = \frac{(2n)!}{n!(n+1)!} \sim \frac{4^n}{\sqrt{\pi} n^{3/2}}.$$

- we cannot search exhaustively for the best single copy duplication tree

Minimum Evolution principle

- search for the tree with shortest length (sum of individual branch lengths)
- example : Neighbor-Joining (Saitou and Nei, 1987), heuristic, agglomerative method ($O(n^3)$)

Minimum Evolution principle and OLS tree length estimation

- ME with Ordinary Least Squares tree length estimation is statistically consistent (Rzhetsky and Nei, 1993 ; Denis and Gascuel, 2003)

Minimum Evolution principle and OLS tree length estimation

- Given a topology T and a distance matrix Δ, the OLS branch length estimation of T (valued tree with topology T, inducing a tree distance Δ^T) is obtained by minimizing :

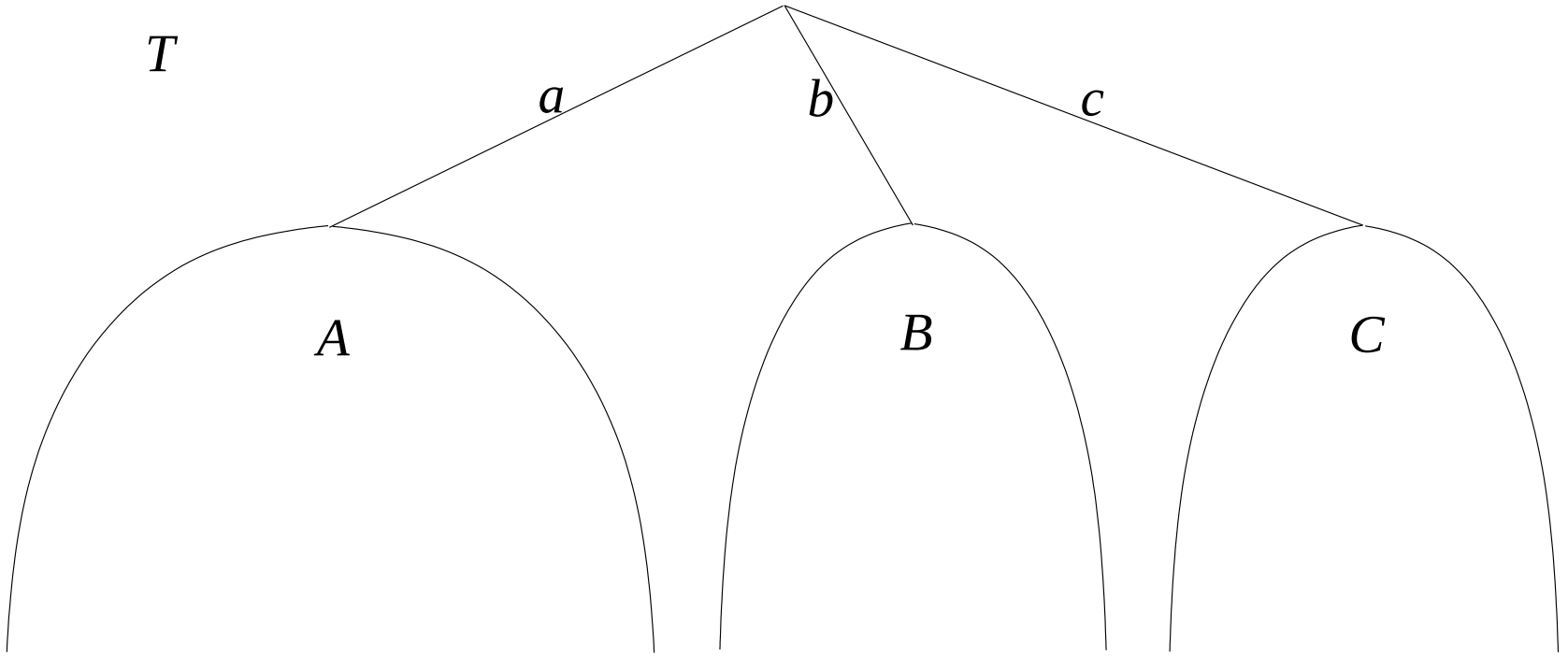
$$\sum_{i,j \in T} (\delta_{ij}^T - \delta_{ij})^2.$$

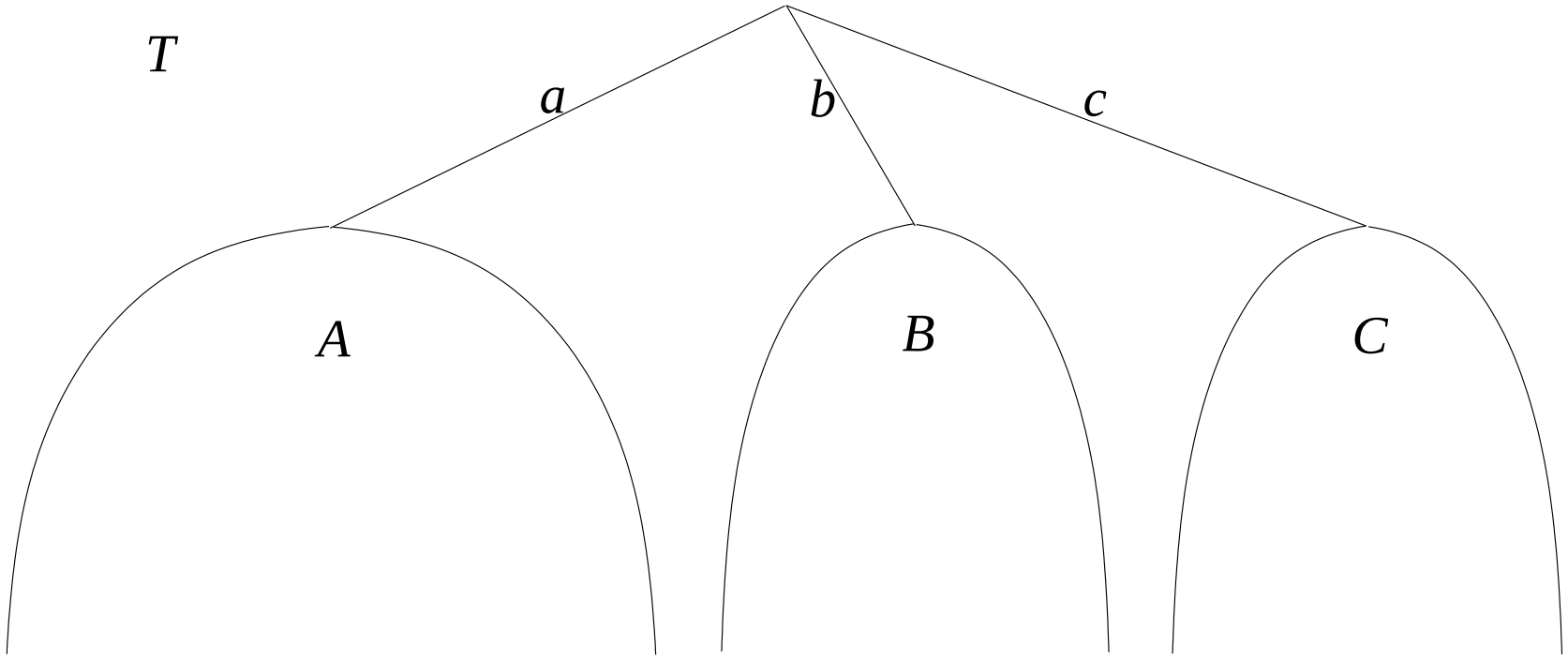
OLS estimation of tree length

- recurrence equation for calculating the length of a tree topology using OLS, given a matrix of pairwise evolutionary distances

Δ	matrix of evolutionary distances (eg K2P)
δ_{ij}	distance between copy i and copy j
T	(unrooted) tree topology
T	valued tree with topology T
n	number of leaves in T
Δ^T	tree distance induced by T
δ_{ij}^T	distance between copy i and copy j
$L(T)$	sum of branch lengths of T

OLS estimation of tree length

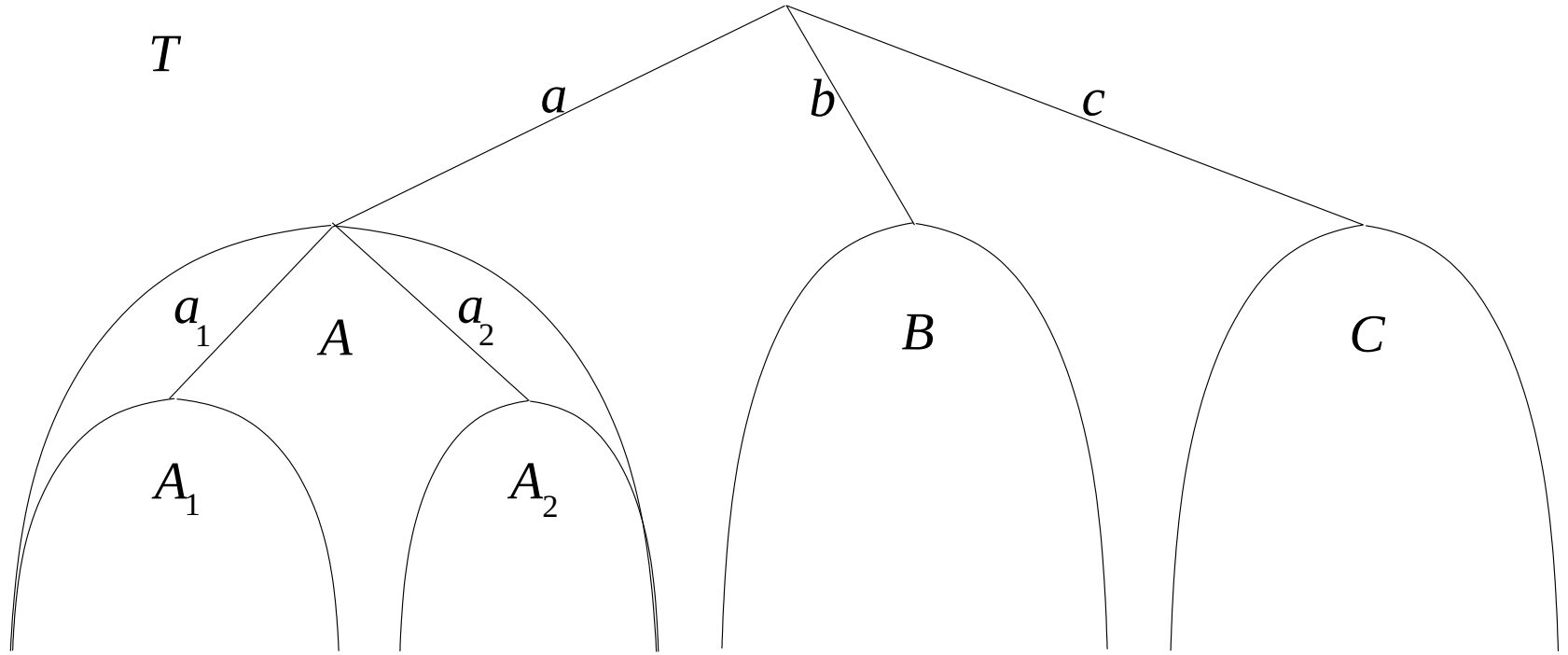




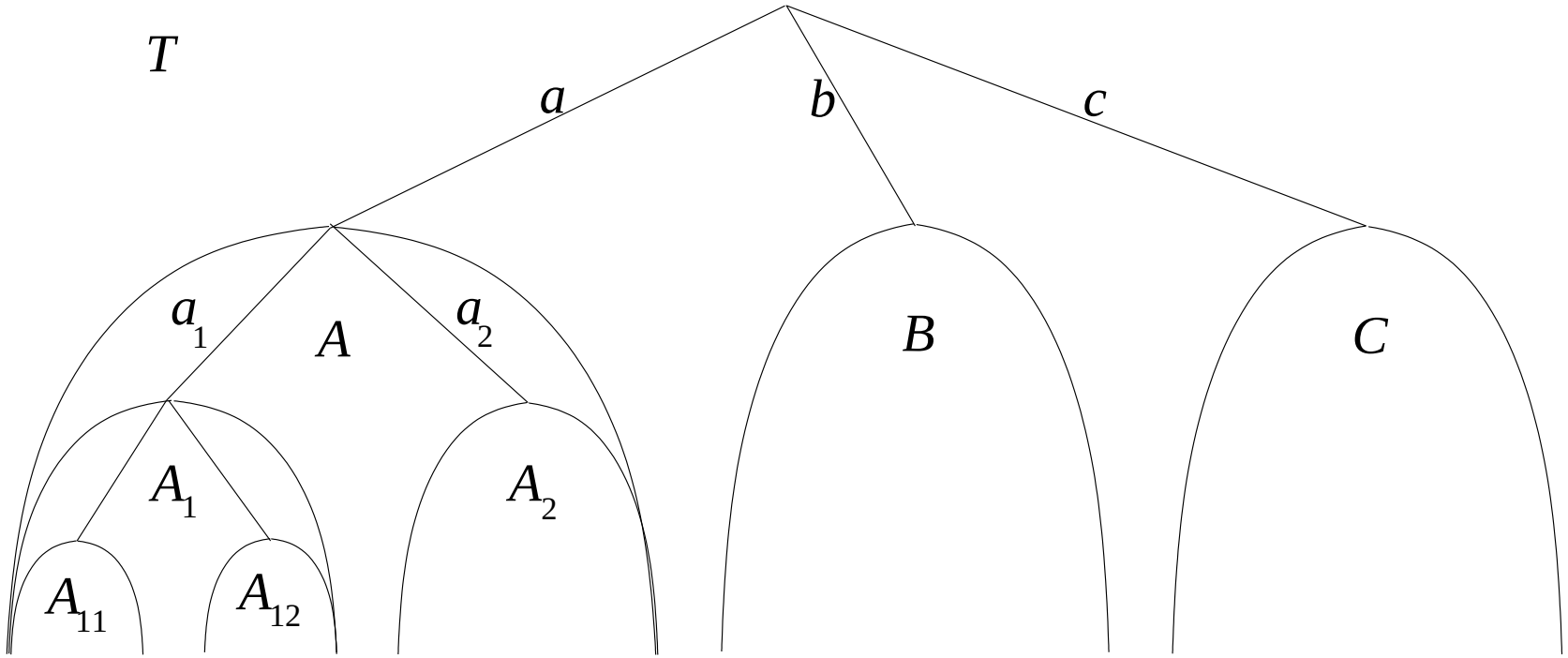
R is the subset of leaves which do not belong to A

$(R = B \cup C)$

OLS estimation of tree length

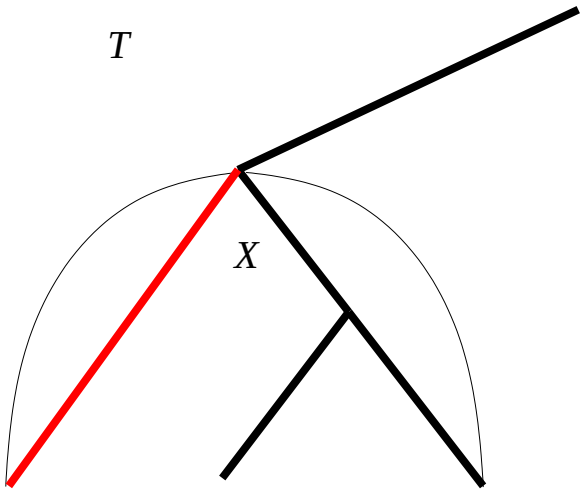


OLS estimation of tree length



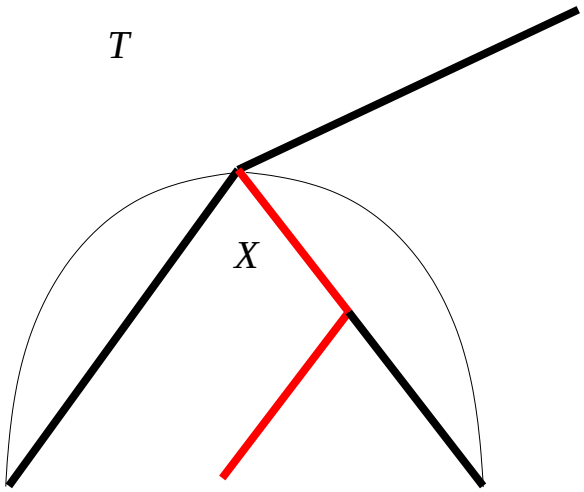
Notations

- X any subtree of T
- $\overline{X}$ average distance in T between the root of X and its leaves



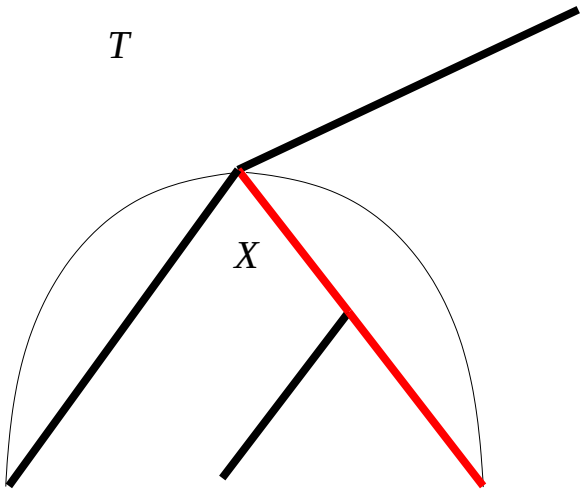
Notations

- X any subtree of T
- $\overline{X}$ average distance in T between the root of X and its leaves



Notations

- X any subtree of T
- $\overline{X}$ average distance in T between the root of X and its leaves



Notations

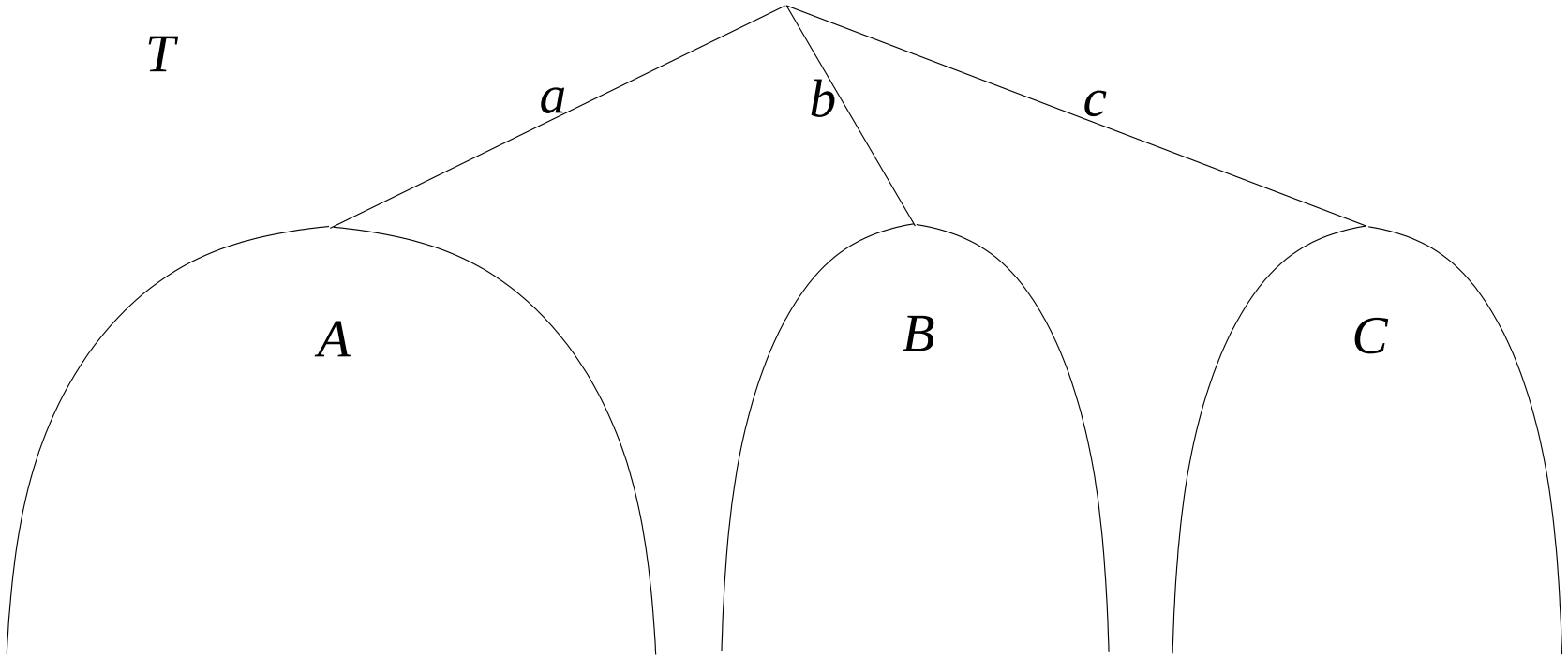
- X, Y two non-intersecting subtrees
- Δ_{XY} average distance in Δ between the leaves of X and Y

$$\Delta_{XY} = \frac{1}{|X|\,|Y|} \sum_{i \in X, j \in Y} \delta_{ij}$$

- Δ^T_{XY} average distance in T (or Δ^T) between the leaves of X and Y

$$\Delta^T_{XY} = \frac{1}{|X|\,|Y|} \sum_{i \in X, j \in Y} \delta^T_{ij}$$

OLS estimation of tree length



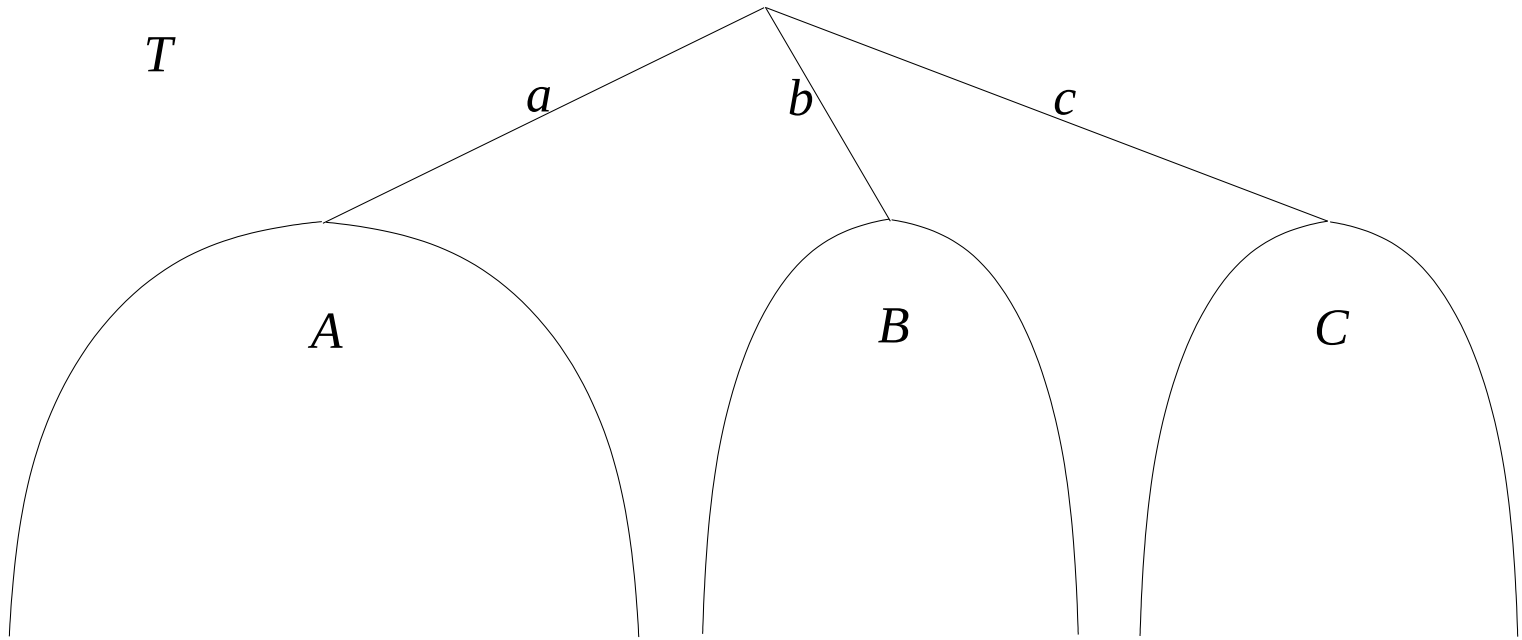
$$L(T) = L(A) + L(B) + L(C) + a + b + c$$

OLS property (Vach, 1989 ; Gascuel, 1997)

- the average distance between (non-intersecting subtrees) X and Y is preserved between Δ and Δ^T, when X and Y are adjacent to a common ternary node

$$\Delta_{XY} = \Delta^T_{XY}$$

OLS estimation of tree length



$$\Delta_{AB} = \Delta^{T}_{AB} = \overline{A} + a + b + \overline{B}$$

$$L(T) = L(A) + L(B) + L(C) + a + b + c$$

$$(i) \quad \Delta_{AB} = \overline{A} + a + b + \overline{B},$$

$$(ii) \quad \Delta_{AC} = \overline{A} + a + c + \overline{C},$$

$$(iii) \quad \Delta_{BC} = \overline{B} + b + c + \overline{C},$$

$$L(T) = L(A) + L(B) + L(C) + a + b + c$$

$$(i) \quad \Delta_{AB} = \overline{A} + a + b + \overline{B},$$

$$(ii) \quad \Delta_{AC} = \overline{A} + a + c + \overline{C},$$

$$(iii) \quad \Delta_{BC} = \overline{B} + b + c + \overline{C},$$

$$L(T) = (L(A) - \overline{A}) + (L(B) - \overline{B}) + (L(C) - \overline{C}) + \frac{1}{2}(\Delta_{AB} + \Delta_{AC} + \Delta_{BC})$$

$$L(T) = L(A) + L(B) + L(C) + a + b + c$$

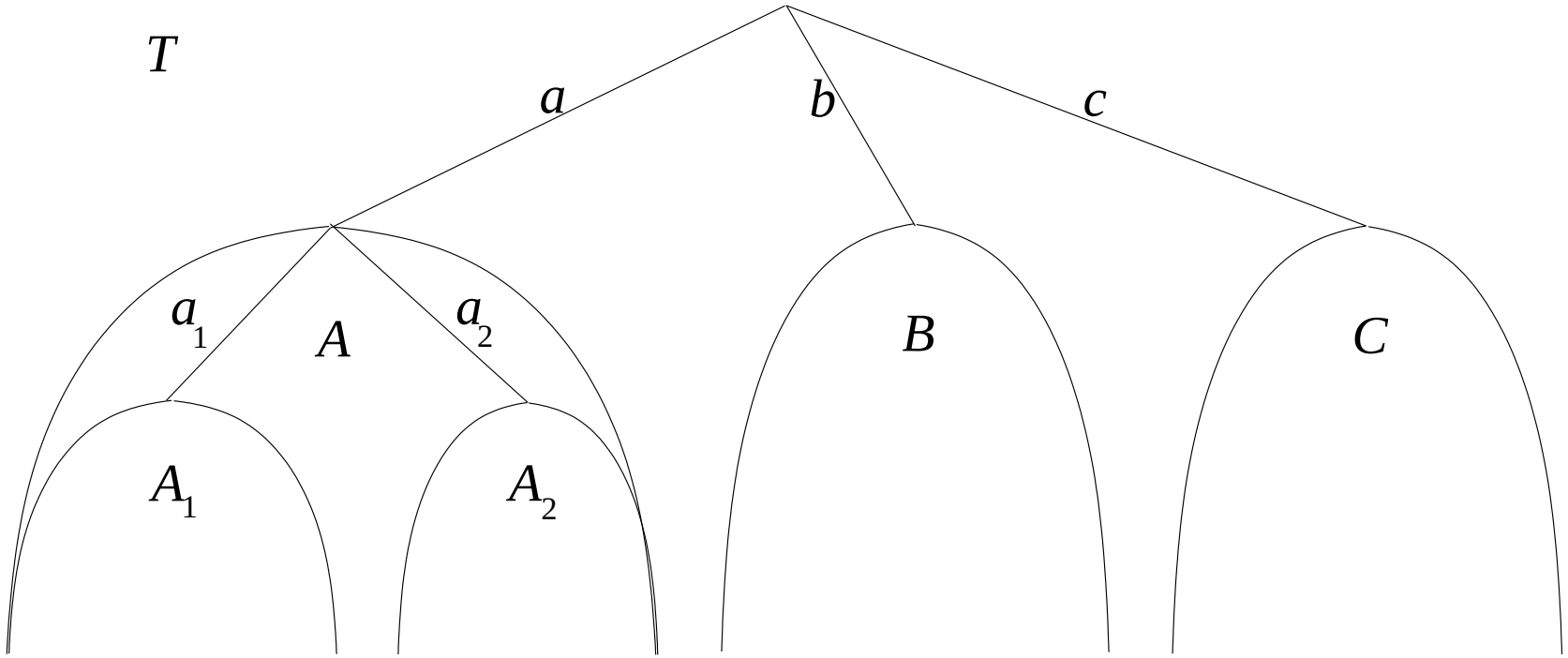
$$(i) \quad \Delta_{AB} = \overline{A} + a + b + \overline{B},$$

$$(ii) \quad \Delta_{AC} = \overline{A} + a + c + \overline{C},$$

$$(iii) \quad \Delta_{BC} = \overline{B} + b + c + \overline{C},$$

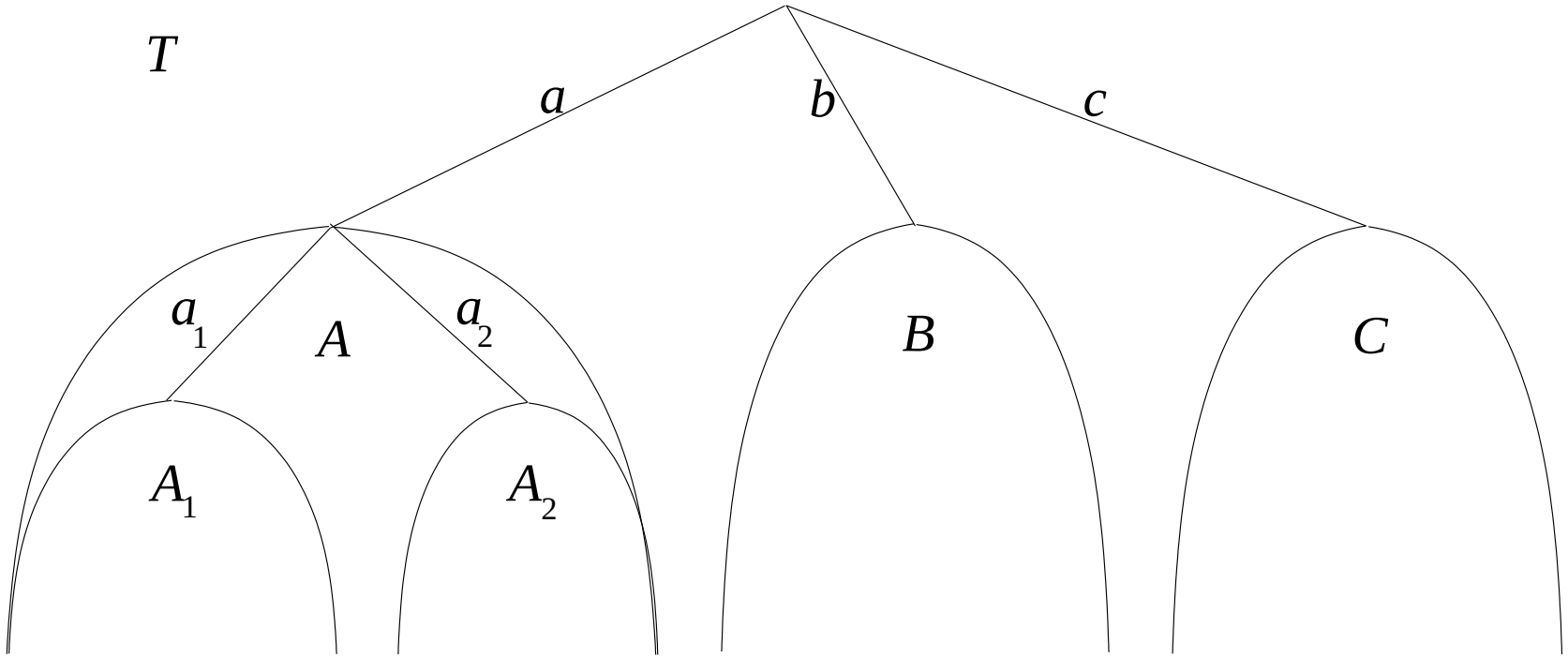
$$L(T) = (L(A) - \overline{A}) + (L(B) - \overline{B}) + (L(C) - \overline{C}) + \frac{1}{2}(\Delta_{AB} + \Delta_{AC} + \Delta_{BC})$$

OLS estimation of tree length



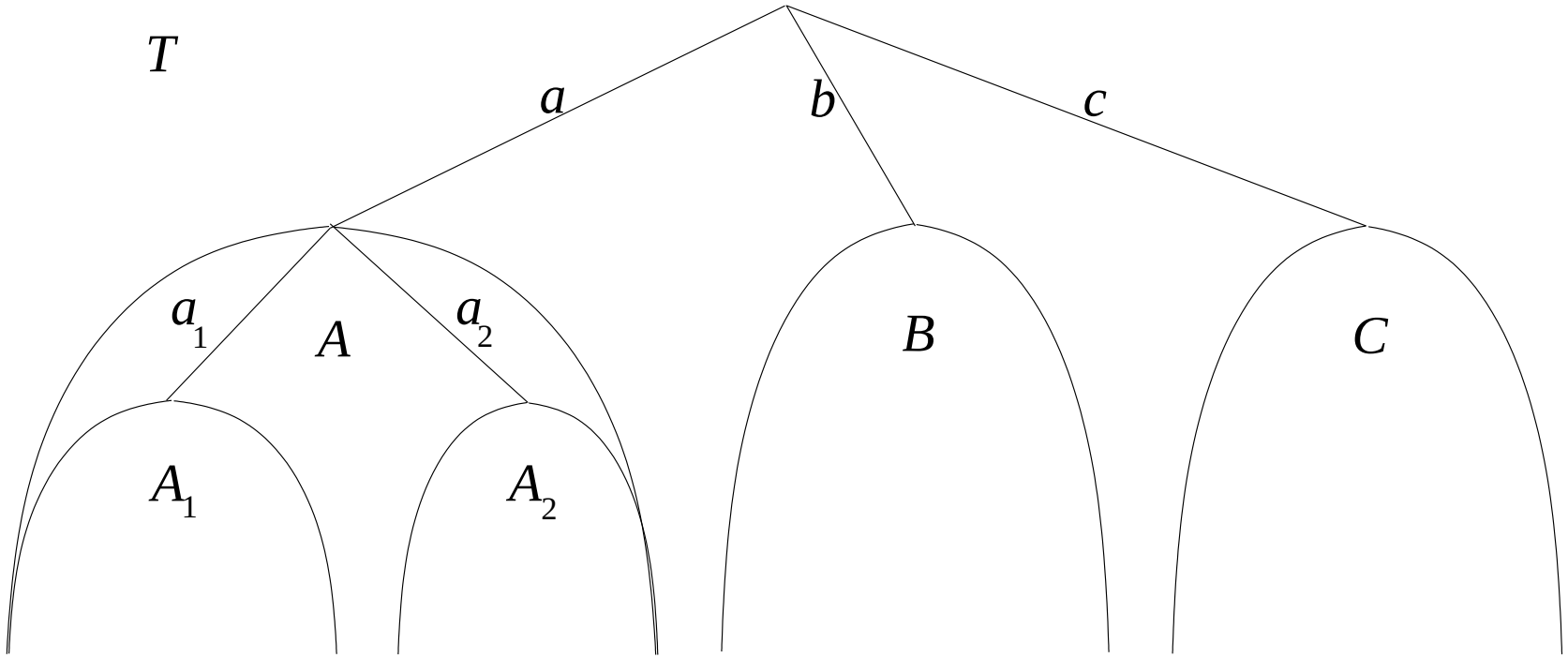
$$L(A) = L(A_1) + L(A_2) + a_1 + a_2$$

OLS estimation of tree length



$$\overline{A} = \frac{|A_1|}{|A|}(a_1 + \overline{A_1}) + \frac{|A_2|}{|A|}(a_2 + \overline{A_2})$$

OLS estimation of tree length



$$\Delta_{A_1A_2} = \Delta^T_{A_1A_2} = \overline{A_1} + a_1 + a_2 + \overline{A_2}$$

$$
\begin{aligned}
(i)' \quad & \Delta_{A_1 A_2} = \overline{A_1} + a_1 + a_2 + \overline{A_2}, \\
(ii)' \quad & \Delta_{A_1 R} = \overline{A_1} + a_1 + a + \overline{R}, \\
(iii)' \quad & \Delta_{A_2 R} = \overline{A_2} + a_2 + a + \overline{R},
\end{aligned}
$$

- solve to get an expression for a_1 and a_2

$$a1 = \frac{1}{2}\Delta_{A_1A_2} + \frac{1}{2}\Delta_{A_1R} - \frac{1}{2}\Delta_{A_2R} - \overline{A_1}$$

$$a2 = \frac{1}{2}\Delta_{A_1A_2} + \frac{1}{2}\Delta_{A_2R} - \frac{1}{2}\Delta_{A_1R} - \overline{A_2}$$

$$a1 = \frac{1}{2}\Delta_{A_1A_2} + \frac{1}{2}\Delta_{A_1R} - \frac{1}{2}\Delta_{A_2R} - \overline{A_1}$$

$$a2 = \frac{1}{2}\Delta_{A_1A_2} + \frac{1}{2}\Delta_{A_2R} - \frac{1}{2}\Delta_{A_1R} - \overline{A_2}$$

$$\overline{A} = \frac{|A_1|}{|A|}(a_1 + \overline{A_1}) + \frac{|A_2|}{|A|}(a_2 + \overline{A_2})$$

$$a1 = \frac{1}{2}\Delta_{A_1A_2} + \frac{1}{2}\Delta_{A_1R} - \frac{1}{2}\Delta_{A_2R} - \overline{A_1}$$

$$a2 = \frac{1}{2}\Delta_{A_1A_2} + \frac{1}{2}\Delta_{A_2R} - \frac{1}{2}\Delta_{A_1R} - \overline{A_2}$$

$$\overline{A} = \frac{|A_1|}{|A|}(a_1 + \overline{A_1}) + \frac{|A_2|}{|A|}(a_2 + \overline{A_2})$$

$$L(A) = L(A_1) + L(A_2) + a_1 + a_2$$

- combine to get an expression for $(L(A) - \overline{A})$

$$
\begin{aligned}
(L(A)-\overline{A}) = \; & (L(A_1) - \overline{A_1}) + (L(A_2) - \overline{A_2}) \\
& + \tfrac{1}{2}\Delta_{A_1 A_2} \\
& + \tfrac{1}{2}\left(\tfrac{|A_2|-|A_1|}{|A|}\right)\Delta_{A_1 R} + \tfrac{1}{2}\left(\tfrac{|A_1|-|A_2|}{|A|}\right)\Delta_{A_2 R}
\end{aligned}
$$

$$L(T) = (L(A) - \overline{A}) + (L(B) - \overline{B}) + (L(C) - \overline{C}) + \frac{1}{2}(\Delta_{AB} + \Delta_{AC} + \Delta_{BC}),$$

$$(L(A) - \overline{A}) = (L(A_1) - \overline{A_1}) + (L(A_2) - \overline{A_2}) + \frac{1}{2}\Delta_{A_1 A_2} + \frac{1}{2}\left(\frac{|A_2| - |A_1|}{|A|}\right)\Delta_{A_1 R} + \frac{1}{2}\left(\frac{|A_1| - |A_2|}{|A|}\right)\Delta_{A_2 R},$$

$(L(A) - \overline{A}) = 0$ if A is a leaf.

$$L(T) = (L(A) - \overline{A}) + (L(B) - \overline{B}) + (L(C) - \overline{C}) + \frac{1}{2}(\Delta_{AB} + \Delta_{AC} + \Delta_{BC}).$$

- $(L(A) - \overline{A})$ only depends on the structure of subtree A (not on the structure of $R = B \cup C$), idem for $(L(B) - \overline{B})$, $(L(C) - \overline{C})$
- to compute $L(T)$, we only have to compute (independently) $(L(A) - \overline{A})$, $(L(B) - \overline{B})$ and $(L(C) - \overline{C})$, then to apply the above equation.

$$
\begin{aligned}
(L(A)-\overline{A}) = \; & (L(A_1) - \overline{A_1}) + (L(A_2) - \overline{A_2}) \\
& + \tfrac{1}{2}\Delta_{A_1 A_2} \\
& + \tfrac{1}{2}\left(\tfrac{|A_2|-|A_1|}{|A|}\right)\Delta_{A_1 R} + \tfrac{1}{2}\left(\tfrac{|A_1|-|A_2|}{|A|}\right)\Delta_{A_2 R},
\end{aligned}
$$

- $(L(A_1) - \overline{A_1})$ only depends on the structure of A_1, not on the structure of $R' = R \cup A_2$
- to compute $(L(A) - \overline{A})$, we only have to compute (independently) $(L(A_1) - \overline{A_1})$, $(L(A_2) - \overline{A_2})$ then to apply the above equation.

$$L(T) = (L(A) - \overline{A}) + (L(B) - \overline{B}) + (L(C) - \overline{C}) + \frac{1}{2}(\Delta_{AB} + \Delta_{AC} + \Delta_{BC}),$$

$$(L(A) - \overline{A}) = (L(A_1) - \overline{A_1}) + (L(A_2) - \overline{A_2}) + \frac{1}{2}\Delta_{A_1 A_2} + \frac{1}{2}\left(\frac{|A_2| - |A_1|}{|A|}\right)\Delta_{A_1 R} + \frac{1}{2}\left(\frac{|A_1| - |A_2|}{|A|}\right)\Delta_{A_2 R},$$

$(L(A) - \overline{A}) = 0$ if A is a leaf.

Reconstructing the optimal single duplication tree under the ME principle

- we want to find the tree whose length is minimum among all possible duplication trees, when calculated with this equation

$$L(T) = (L(A) - \overline{A}) + (L(B) - \overline{B}) + (L(C) - \overline{C}) + \frac{1}{2}(\Delta_{AB} + \Delta_{AC} + \Delta_{BC})$$

- we divide the n copies into 3 adjacent subsets (intervals) A, B and C
- we then (independently) compute the 3 subtree structures which minimize $(L(A) - \overline{A})$, $(L(B) - \overline{B})$, $(L(C) - \overline{C})$
- we then apply the above equation
- we consider all possible combinations of A, B, C

$$L(T) = (L(A) - \overline{A}) + (L(B) - \overline{B}) + (L(C) - \overline{C}) + \frac{1}{2}(\Delta_{AB} + \Delta_{AC} + \Delta_{BC})$$

- we divide the n copies into 3 adjacent subsets (intervals) A, B and C
- we then (independently) compute the 3 subtree structures which minimize $(L(A) - \overline{A})$, $(L(B) - \overline{B})$, $(L(C) - \overline{C})$
- we then apply the above equation
- we consider all possible combinations of A, B, C

$$
\begin{aligned}
(L(A)-\overline{A}) = \; & (L(A_1) - \overline{A_1}) + (L(A_2) - \overline{A_2}) \\
& + \tfrac{1}{2}\Delta_{A_1 A_2} \\
& + \tfrac{1}{2}\left(\tfrac{|A_2|-|A_1|}{|A|}\right)\Delta_{A_1 R} + \tfrac{1}{2}\left(\tfrac{|A_1|-|A_2|}{|A|}\right)\Delta_{A_2 R},
\end{aligned}
$$

- we divide subset A into 2 adjacent sets A_1 and A_2
- we then (independently) compute the 2 subtree structures which minimize $(L(A_1) - \overline{A_1})$, $(L(A_2) - \overline{A_2})$
- we then apply the above equation

- we consider all possible combinations of A_1, A_2

Reconstruction algorithm

$S \leftarrow n \times n$ matrix
$M \leftarrow n \times n$ matrix
for l from 1 to $n-3$ **do**
  for i from 1 to $n-l$ **do**
    compute $S_{i,i+l}$ and $M_{i,i+l}$
  end for
end for
$L^*(T) \leftarrow \infty$
for m_1 from 1 to $n-2$ **do**
  for m_2 from m_1+1 to $n-1$ **do**
    compute $L(T)$ for X_{1,m_1}, X_{m_1+1,m_2}, $X_{m_2+1,n}$
    if $L(T) < L^*(T)$ **then**
      $L^*(T) \leftarrow L(T)$, $m_1^* \leftarrow m_1$, $m_2^* \leftarrow m2$
    end if
  end for
end for
recreate T using M, starting from M_{1,m_1^*}, $M_{m_1^*+1,m_2^*}$ and $M_{m_2^*+1,n}$

First part

- consider an interval $X_{p,q}$
- $S_{p,q}$ stores the minimum value of $(L(X_{p,q}) - \overline{X_{p,q}})$ (calculated with the recurrence equation)
- computing $S_{p,q}$ requires evaluating all combinations $X_{p,m}$ and $X_{m+1,q}$ (using values in S from smaller interval sizes)
- $M_{p,q}$ stores the value of m for which $(L(X_{p,q}) - \overline{X_{p,q}})$ is minimum

Reconstruction algorithm

$S \leftarrow n \times n$ matrix
$M \leftarrow n \times n$ matrix
for l from 1 to $n-3$ **do**
  for i from 1 to $n-l$ **do**
    compute $S_{i,i+l}$ and $M_{i,i+l}$
  end for
end for
$L^*(T) \leftarrow \infty$
for m_1 from 1 to $n-2$ **do**
  for m_2 from m_1+1 to $n-1$ **do**
    compute $L(T)$ for X_{1,m_1}, X_{m_1+1,m_2}, $X_{m_2+1,n}$
    if $L(T) < L^*(T)$ **then**
      $L^*(T) \leftarrow L(T)$, $m_1^* \leftarrow m_1$, $m_2^* \leftarrow m2$
    end if
  end for
end for
recreate T using M, starting from M_{1,m_1^*}, $M_{m_1^*+1,m_2^*}$ and $M_{m_2^*+1,n}$

Complexity of the first part

- $O(n^2)$ intervals to consider in the first step
- each interval requires the evaluation of $O(n)$ combinations of adjacent sub-intervals
- each combination requires the 3 average distances to be calculated, takes $O(n^2)$ operations
- the time complexity is in $O(n^5)$, $O(n^3)$ with refinements, and $O(n^2)$ space

Complexity of the second part

- $O(n^2)$ combinations of 3 adjacent intervals to consider

- each combination requires the 3 average distances to be calculated, takes $O(n^2)$ operations

- the complexity is in $O(n^4)$, $O(n^2)$ with refinements

Complexity of the third part

- depth first tree traversal through the best intervals, takes $O(n)$ time

Algorithmic refinements

- calculating $(L(A) - \overline{A})$ requires 3 average distances to be calculated
- we consider growing intervals
- most of the work has already been done at previous interval sizes
- we only need to "update" average distances for new intervals
- reduces the complexity from $O(n^5)$ to $O(n^3)$

Summary

- single copy tandem duplication trees
- new recurrence equation to calculate the length of a given tree using OLS
- $O(n^3)$ time, $O(n^2)$ space exact algorithm using a DP framework to reconstruct the shortest single copy duplication tree

The future

- compare the topological accuracy with NJ, DTSCORE (Elemento and Gascuel, 2002)

- extend to multiple copy duplications

- use other distance criteria (eg weighted least-squares)