Filogeneza: problem konstrukcji grafu (drzewa) zależności pomiędzy gatunkami.

3. *D* $\rightarrow$ *T(D)* **poprzez algorytm łączenia sąsiadów**

D $\rightarrow$ *D* :* macierz łącząca sąsiadów

Niech $TotDist(i) = \sum_{k=1}^{n} D_{i,k}$

Definiujemy D* następująco:

$$D^*_{i,i} = 0$$

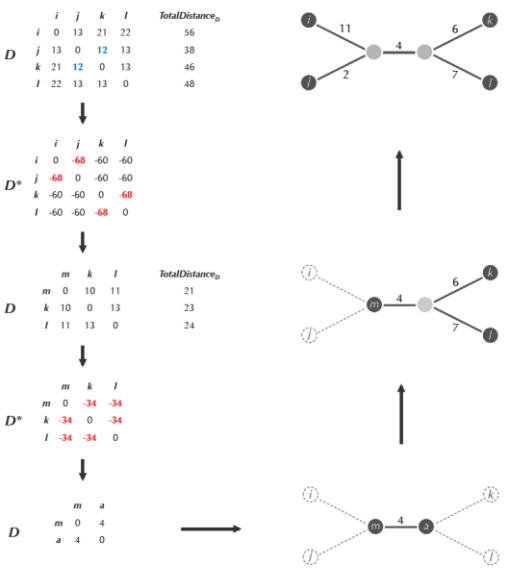
$$D^*_{i,j} = (n-2)D_{i,j} - TotDist(i) - TotDist(i)$$

Twierdzenie:
Dla danej macierzy addytywnej *D*, najmniejszy element $D^*_{i,j}$ macierzy łączącej sąsiadów odpowiada parze sąsiadujących liści i oraz j w *T(D)*

3. $D \rightarrow T(D)$ poprzez algorytm łączenia sąsiadów

NEIGHBORJOINING(D, n)
  if $n = 2$
    $T \leftarrow$ the tree consisting of a single edge of length $D_{1,2}$
    return T
  $D^* \leftarrow$ the neighbor-joining matrix constructed from the distance matrix D
  find elements i and j such that $D^*_{i,j}$ is a minimum non-diagonal element of D^*
  $\Delta \leftarrow (\text{TOTALDISTANCE}_D(i) - \text{TOTALDISTANCE}_D(j))/(n-2)$
  $limbLength_i \leftarrow \frac{1}{2}(D_{i,j} + \Delta)$
  $limbLength_j \leftarrow \frac{1}{2}(D_{i,j} - \Delta)$
  add a new row/column m to D so that $D_{k,m} = D_{m,k} = \frac{1}{2}(D_{k,i} + D_{k,j} - D_{i,j})$
    for any k
  remove rows i and j from D
  remove columns i and j from D
  $T \leftarrow$ NEIGHBORJOINING(D, $n-1$)
  add two new limbs (connecting node m with leaves i and j) to the tree T
  assign length $limbLength_i$ to LIMB(i)
  assign length $limbLength_j$ to LIMB(j)
  return T

3. $D \rightarrow T(D)$ poprzez algorytm łączenia sąsiadów



Skąd pochodzi SARS?

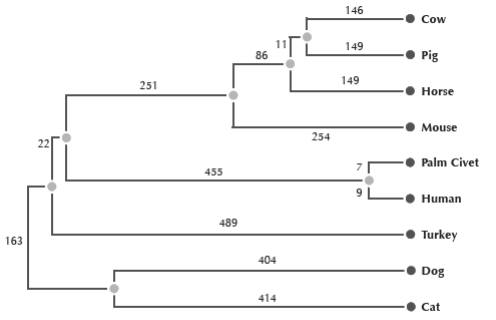


FIGURE 7.19 The neighbor-joining tree of coronaviruses isolated from different animals, based on the non-additive distance matrix in Figure 7.13 (top).

FIGURE 7.14 The palm civet.

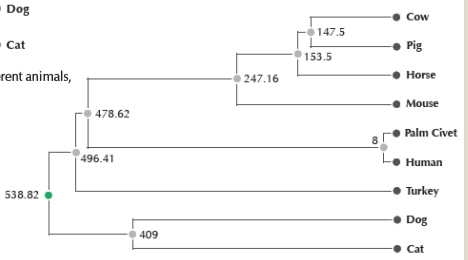
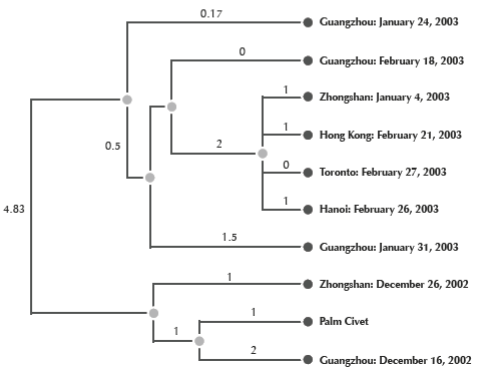


FIGURE 7.17 The ultrametric tree of coronaviruses created by UPGMA using the distance matrix in Figure 7.13 (top). The root is shown in green.

Skąd pochodzi SARS?

Macierz odległości wyznaczono w oparciu dopasowanie dla poszczególnych par białka Spike pobranego z wirusa SARS-CoV uzyskanego od różnych pacjentów. Jest także wirus od palm civet.

	Guangzhou Dec. 16, 2002	Zhongshan Dec. 16, 2002	Guangzhou Jan. 24, 2003	Guangzhou Jan. 31, 2003	Guangzhou Feb. 18, 2003	Hong Kong Feb. 18, 2003	Hanoi Feb. 26, 2003	Toronto Feb. 17, 2003	Hong Kong Mar. 15, 2003	Palm Civet
Guangzhou	0	4	12	8	9	9	12	12	11	3
Zhongshan	4	0	10	6	7	7	10	10	9	3
Guangzhou	12	10	0	4	5	3	2	2	1	11
Guangzhou	8	6	4	0	3	1	4	4	3	7
Guangzhou	9	7	5	3	0	2	5	5	4	8
Hong Kong	9	7	3	1	2	0	3	3	2	8
Hanoi	12	10	2	4	5	3	0	2	1	11
Toronto	12	10	2	4	5	3	2	0	1	11
Hong Kong	11	9	1	3	4	2	1	1	0	10
Palm Civet	3	3	11	7	8	8	11	11	10	0



Drzewo filogenetyczne bazujące na symbolach

	wings	legs
winged stick insect	Yes	6
wingless stick insect	No	6
giant centipede	No	42

FIGURE 7.21 (Top panel) Winged (left) and wingless (middle) stick insects, each having six legs, and the giant centipede (right), which has 42 legs. (Bottom panel) A 3 × 2 character table describes two characters (wings and legs) in these three invertebrates.

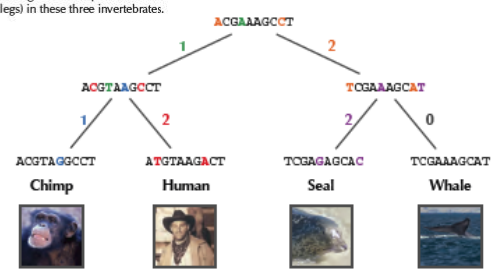


FIGURE 7.23 An evolutionary tree with parsimony score 8 whose leaves are the DNA strings in the multiple alignment from Figure 7.3. Colored letters indicate mismatches in strings connected by an edge.

Metody dyskretne rekonstrukcji drzewa ewolucji

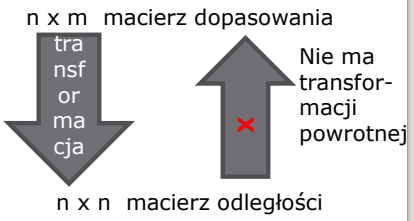
Dane jest n sekwencji DNA o długości m każda.

Mamy zatem *macierz dopasowania* w rozmiarze *n x m*.

```
Species A   ATGGCTATTCTTATAGTACG
Species B   ATCGCTAGTCTTATATTACA
Species C   TTCACTAGACCTGTGGTCCA
Species D   TTGACCAGACCTGTGGTCCG
Species E   TTGACCAGTTCTCTAGTTCG
```

Można ją przetransformować na *macierz odległości*, ale nigdy w drugą stronę.

Informacja o dopasowaniu jest bezpowrotnie tracona przy tej transformacji



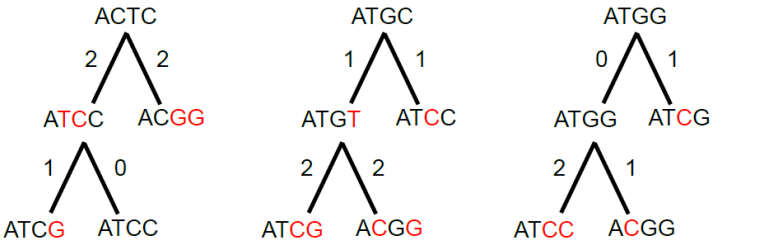
Lepsza technika:
algorytm rekonstrukcji drzewa bazujący na symbolach umożliwia badanie ewolucji dla każdego znaku.

Parsymonia w rekonstrukcji drzewa filogenetycznego

Parsymonia (oszczędność): kryterium optymalizacyjne - szukamy takiego drzewa, które wyznacza najmniejszą liczbę zdarzeń ewolucyjnych (podstawienia, zamiany, itp.)

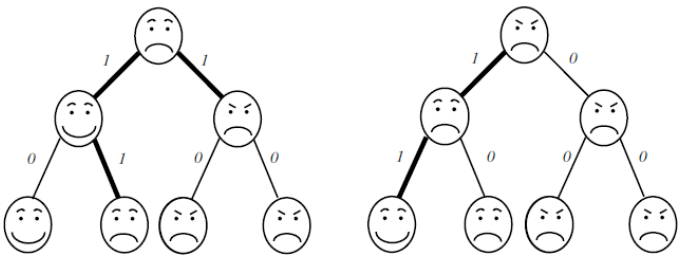
Brzytwa Ockhama

Przykład: Szukaj najprostszego wyjaśnienia dla danych { ATCG, ATCC, ACGG}



Parsimony score: 5 Parsimony score: 6 Parsimony score: 4

Problem parsymonii inaczej



(a) *Parsimony Score=3* (b) *Parsimony Score=2*

Znaki naszego drzewa to brwi i usta. Każdy z nich może być w dwóch stanach.
Dobierz etykiety węzłów wewnętrznych tak by wynik parsymonii był najmniejszy.

Drzewo filogenetyczne bazujące na symbolach

Dwie klasy problemów:
małej parsymonii : zakładamy, że struktura drzewa jest dana
wielkiej parsymonii : struktura drzewa jest dowolna.

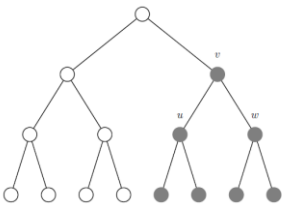
Small Parsimony Problem:
Find the most parsimonious labeling of the internal vertices in an evolutionary tree.

Input: Tree T with each leaf labeled by an m-character string.

Output: Labeling of internal vertices of the tree T minimizing the parsimony score.

Drzewo filogenetyczne bazujące na symbolach

Każdy wierzchołek *v* z drzewa T wyznacza poddrzewo o korzeniu: wierzchołków osiągalnych z *v*. Etykieta *v* ma zbierać własności dzieci wierzchołka v.



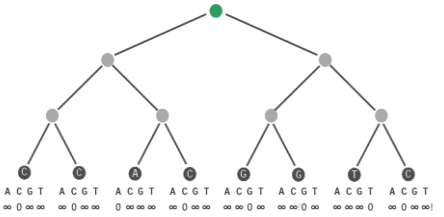
Niech $s_k(v)$ to wynik parsymonii dla poddrzewa *v* uzyskany przy założeniu, ze w *v* umieszczono znak *k*, czyli

Algorytm dynamiczny

$$s_k(v) = \min_{all\ symbols\ i}(s_i(\text{Daughter}(v)) + \delta_{i,k}) + \min_{all\ symbols\ j}(s_j(\text{Son}(v)) + \delta_{j,k})$$

Warunki początkowe algorytmu:

$$s_k(v) = \begin{cases} 0 \text{ dla } v = k \\ \infty \text{ dla } v \neq k \end{cases}$$



Algorytm Sankoffa

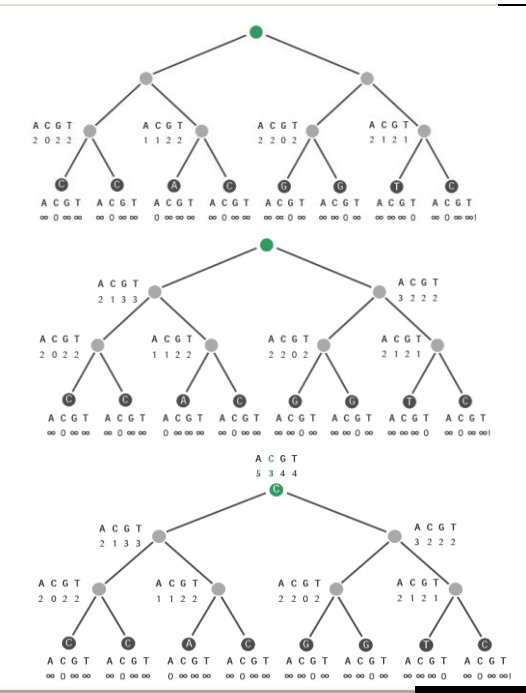
```
SMALLPARSIMONY(T, CHARACTER)
    for each node v in tree T
        TAG(v) ← 0
        if v is a leaf
            TAG(v) ← 1
            for each symbol k in the alphabet
                if CHARACTER(v) = k
                    s_k(v) ← 0
                else
                    s_k(v) ← ∞
    while there exist ripe nodes in T
        v ← a ripe node in T
        TAG(v) ← 1
        for each symbol k in the alphabet
            s_k(v) ← min_{all symbols i} {s_i(DAUGHTER(v)) + δ_{i,k}} + min_{all symbols j} {s_j(SON(v)) + δ_{j,k}}
    return min_{all symbols k} s_k(v)
```

Character: wektor dostępnych znaków (u nas ACGT)

Tag: znacznik, czy węzeł był już obsłużony

Wartości początkowe

ripe: dojrzały węzeł ma Tag równy 0, ale jego dzieci mają Tag równe 1

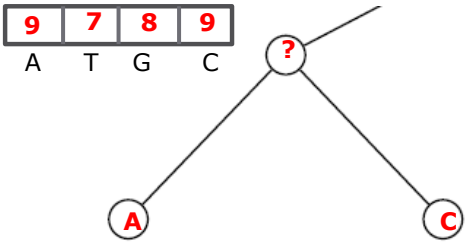


Algorytm Sankoffa z inną punktacją

$s_t(v) = \min_i \{s_i(u) + \delta_{i,t}\} + \min_j \{s_j(w) + \delta_{j,t}\}$

$s_A(v) = 0 + \min_j \{s_j(w) + \delta_{j,A}\}$

$s_A(v) = 0 + 9 = \mathbf{9}$

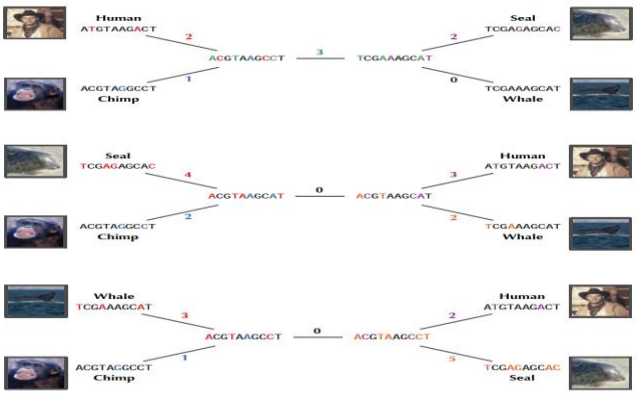


δ	A	T	G	C
A	0	3	4	9
T	3	0	2	4
G	4	2	0	4
C	9	4	4	0

Mała parsymonia na drzewie nieukorzenionym

Small Parsimony in an Unrooted Tree Problem:
Find the most parsimonious labeling of the internal nodes of an unrooted tree.

Input: An unrooted binary tree with each leaf labeled by a string of length m.
Output: A labeling of all other nodes of the tree by strings of length m that minimizes the tree's parsimony score.



Problem wielkiej parsymonii

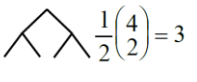
Large Parsimony Problem:
Find a tree with n leaves having the minimal parsimony score.

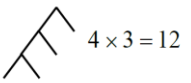
Input: An $n \times m$ matrix M describing n species, each represented by an m-character string.

Output: A tree T with n leaves labeled by the n rows of matrix M, and a labeling of the internal vertices of this tree such that the parsimony score is minimized over all possible trees and over all possible labelings of internal vertices.

Problem NP-zupełny

Przykłady drzew o 4 liściach





Ilość drzew ukorzenionych o n liściach :

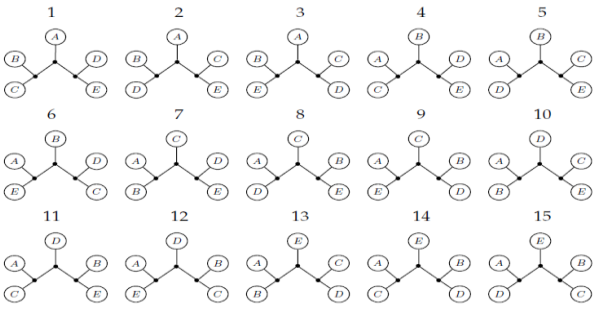
$$T(n) = \frac{(2n-3)!}{2^{n-2}(n-2)!}$$

T(n) dla n = 2, 3, 4, 5, 6, 7, 8, 9, 10, …. to

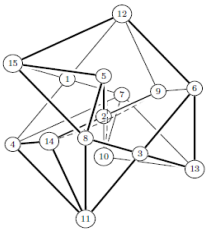
$$1, 3, 15, 105, 945, 10395, 135135, 2027025, 34459425\ldots$$

Problem przeszukiwania przestrzeni drzew

Wszystkie drzewa swobodne o pięciu liściach.



Drzewa sąsiadujące (poprzez transformację zamiany najbliższych sąsiadów) są połączone krawędzią



Algorytm zachłanny Monte Carlo przeszukuje przestrzeń drzew