BIOINFORMATIKA – CV. 6

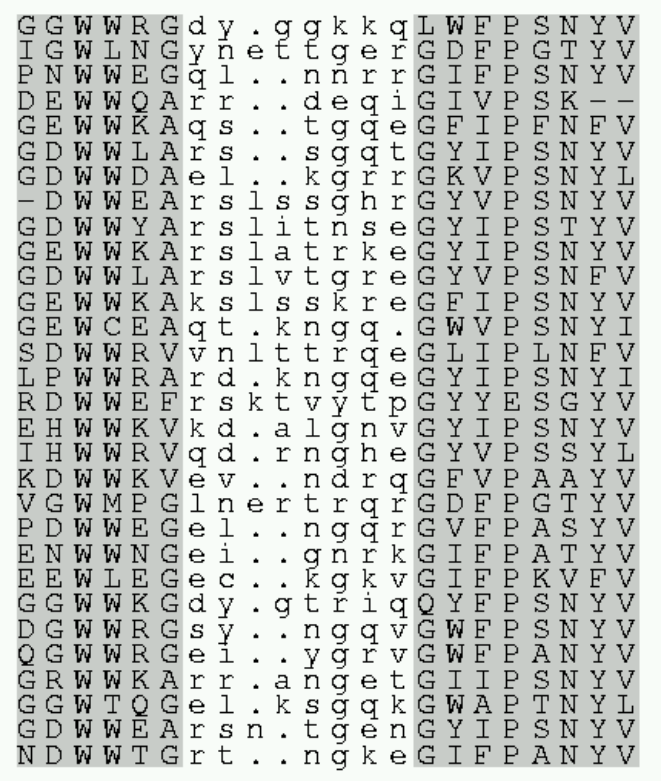
Globální zarovnávání (implementace), zarovnávání vícero sekvencí a fylogenetické stromy

Some slides are courtesy of Mark Craven

Zarovnávání vícero sekvencí

(založeno na slajdech M. Cravena)

- Máme více než 2 sekvence a chceme je zarovnat tak, abychom maximalizovali danou „scoring function“



Biologická motivace

- Vstupní data pro výpočet „vzdáleností“ pro konstrukci fylogenetických stromů
- Charakterizování proteinů majících stejnou nebo podobnou funkci (pro prediktivní klasifikaci)
- Hledání evolučně konzervovaných motivů

„Scoring function“

- Typický tvar „scoring function“ je následující:

$$Score(m) = G + \sum_i S(m_i)$$

gap function

score of i^{th} column

(předpokládá nezávislost jednotlivých sloupců)

Pro lineární „gap“ penaltu má jednodušší tvar (k mezerám se můžeme chovat jako k jinému typu aminokyselinového residua):

$$Score(m) = \sum_i S(m_i)$$

„Scoring function“: Sum of Pairs

- Sečteme skóre pro match/mismatch pro všechny dvojice sekvencí, které máme zarovnat:

$$S(m_i) = \sum_{k<l} s(m_i^k, m_i^l)$$

$m_i^k =$ znak k-té sekvence v i-tém sloupci

$S =$ substituční matice (například BLOSUM)

„Scoring function“: Minimální entropie

- Základní myšlenka: minimalizujeme entropii v každém sloupci

$$S(M_i) = -\sum_a c_{ia} \cdot \log p_{ia}$$

zde c_{ia} je počet aminokyselinových residuí typu *a* v *i*-tém *sloupci*, p_{ia} je pravděpodobnost výskytu aminokyselinového residua typu *a* v *i*-tém sloupci (můžeme ji buď získat jako relativní frekvencí, ale v praxi častěji pomocí Laplaceova odhadu)

Jak počítat zarovnávání vícero sekvencí?

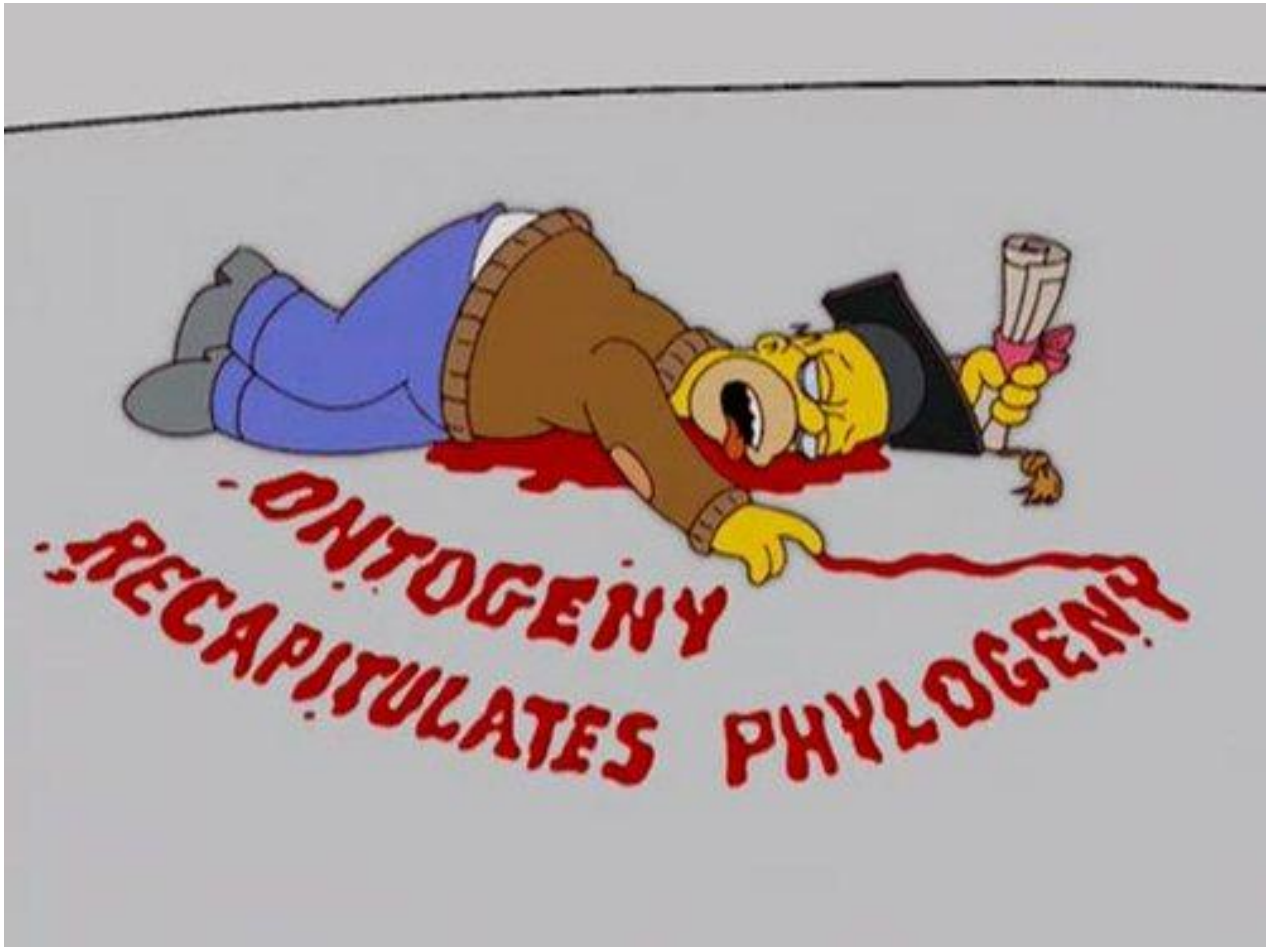
- Bohužel jde o NP-těžký úkol
- Lze zobecnit postup pro výpočet zarovnání dvou sekvencí pomocí DP pro více sekvencí. Nicméně výpočetní náročnost roste exponenciálně s počtem sekvencí.
- Nejčastěji se používají heuristické přístupy (např. CLUSTAL W).

Příklad:

- Na stránce http://mafft.cbrc.jp/alignment/server/ naleznete online nástroj pro zarovnávání sekvencí.
- Zarovnejte pomocí něho následující sekvence (budeme to potřebovat pro konstrukci fylogenetického stromu):

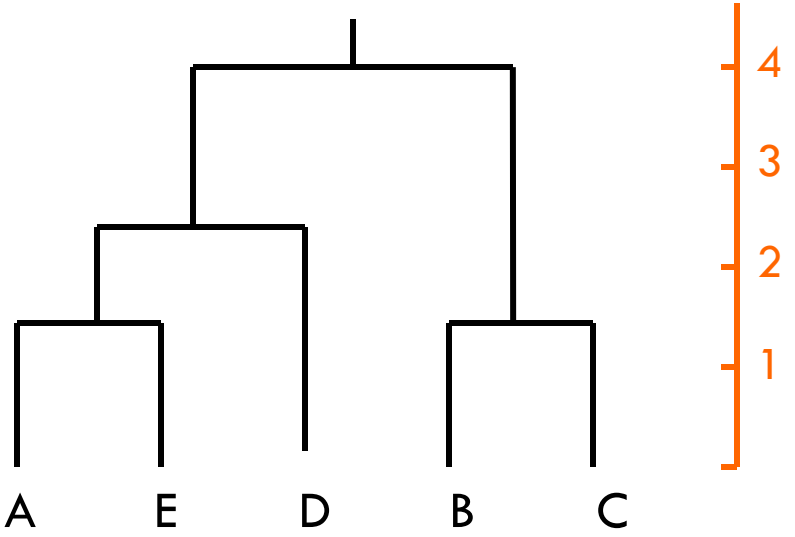
1. KVMAHMK
2. KLMAHMK
3. KLMLHMK
4. KLMAHMKK
5. RVMAHMK
6. RVMAMK

Fylogenetické stromy

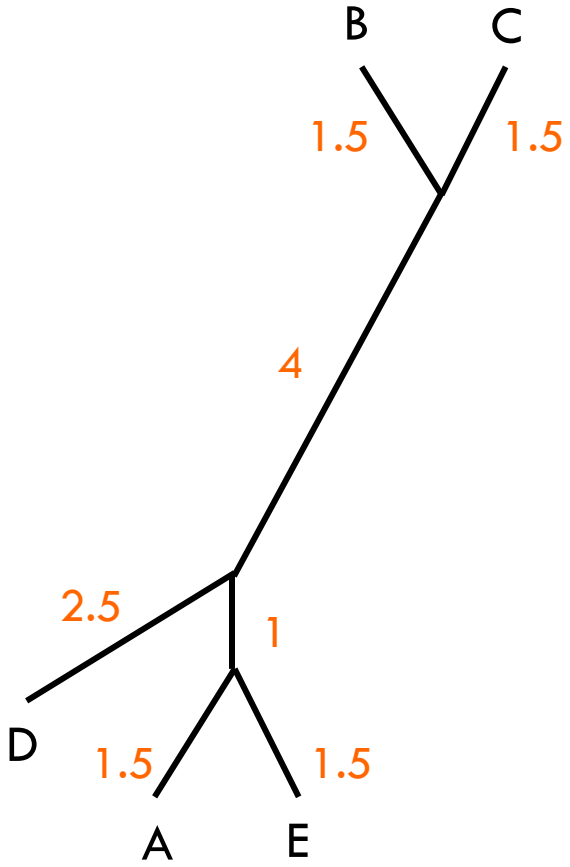


Fylogenetické stromy

NA DNEŠNÍM CVIČENÍ
(metoda UPGMA)



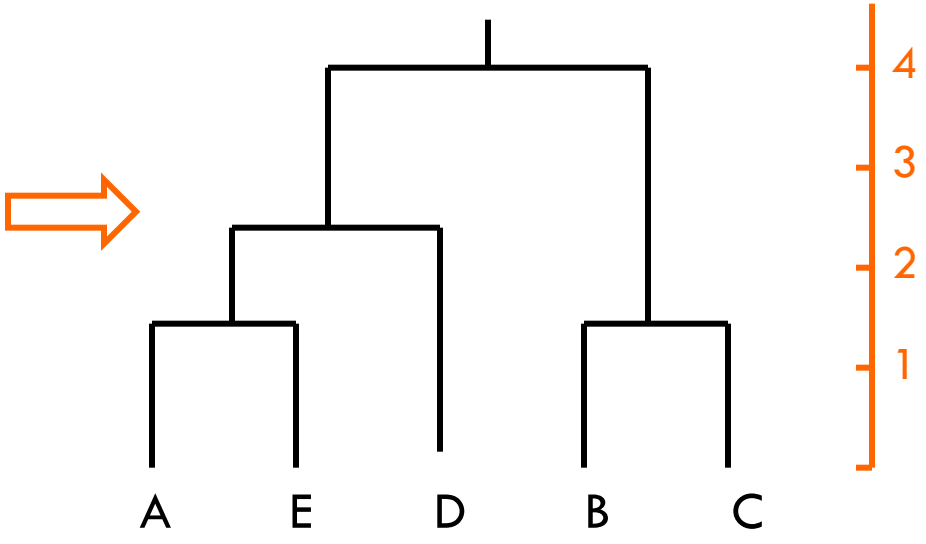
Vzdálenosti reprezentovány součtem výšek hran k potřebných k dosažení společného předka



Vzdálenosti reprezentovány součtem délek hran ke společnému předkovi

Distance-based approaches

	A	B	C	D	E
A	0	8	8	5	3
B		0	3	8	8
C			0	8	8
D				0	5
E					0



Jak budeme počítat vzdálenosti?

- Nejjednodušší varianta:

$$dist_{ij} = \frac{\#\text{mismatches}}{\#\text{matches} + \#\text{mismatches}}$$

kde *#matches* a *#mismatches* jsou vzhledem k zarovnání sekvencí, pro něž konstruujeme strom (existují i lepší varianty)

- **Zkonstruujte matici vzdáleností pro sekvence:** KVMAHMK, KLMAHMK, KLMLHMK, KLMAHMKK, RVMAHMK, RVMAMK

Metoda UPGMA

- Zkonstruujte na základě spočítané matice vzdáleností fylogenetický strom pomocí metody UPGMA (viz přednáška)

UPGMA

assign each taxon to its own cluster

define one leaf for each taxon; place it at height 0

while more than two clusters

- determine two clusters i, j with smallest d_{ij}
- define a new cluster $C_k = C_i \cup C_j$
- define a node k with children i and j; place it at height $d_{ij}/2$
- replace clusters i and j with k
- compute distance between k and other clusters

join last two clusters, i and j, by root at height $d_{ij}/2$

Metoda Neighbour Joining

- Zkonstruujte na základě spočítané matice vzdáleností fylogenetický strom pomocí metody Neighbour Joining

Neighbour Joining

define the tree T = set of leaf nodes

$L = T$

while more than two subtrees in T

- pick the pair i, j in L with minimal D_{ij}
- add to T a new node k joining i and j
- determine new distances

$$d_{ik} = \frac{1}{2}\left(d_{ij} + r_i - r_j\right)$$

$$d_{jk} = d_{ij} - d_{ik}$$

$$d_{km} = \frac{1}{2}\left(d_{im} + d_{jm} - d_{ij}\right) \text{ for all other } m \text{ in } L$$

- remove i and j from L and insert k (treat it like a leaf)

- join two remaining subtrees, i and j with edge of length

$$D_{ij} = d_{ij} - (r_i + r_j)$$

$$r_i = \frac{1}{|L| - 2} \sum_{k \in L} d_{ik}$$