Organizačné poznámky

- Domáca úloha 1 do stredy 12.11. 22:00

  Otázky k zadaniu emailom

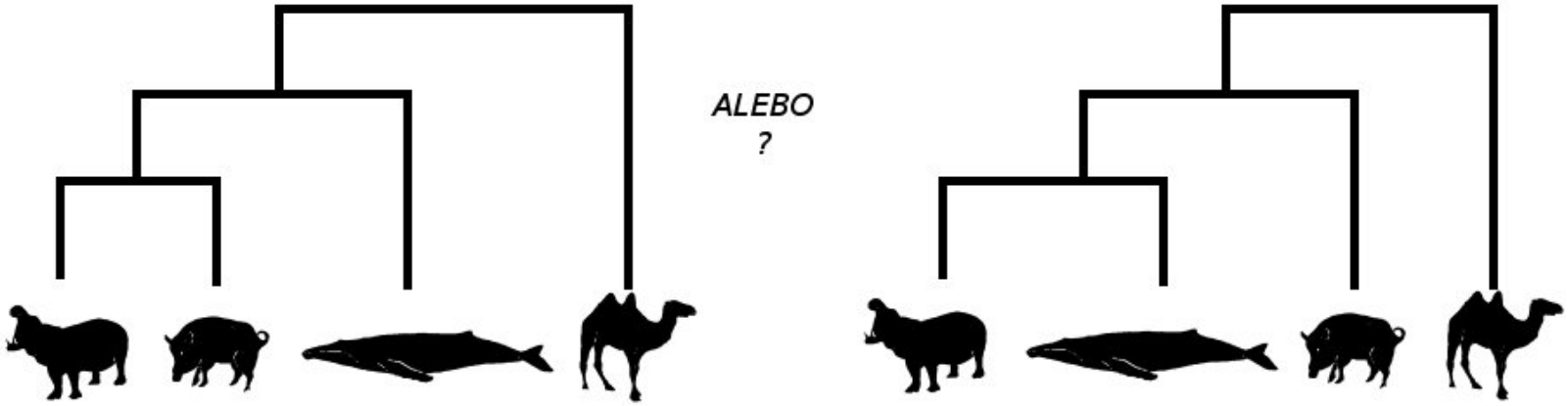
- Pracujte na journal clube

  (prečítajte si článok, naplánujte si stretnutie pred 21.11.)

Evolučné modely a stromy

Broňa Brejová

30.10.2025



Rekonštrukcia fylogenetických stromov

Vstup:

m **zarovnaných** sekvencií,

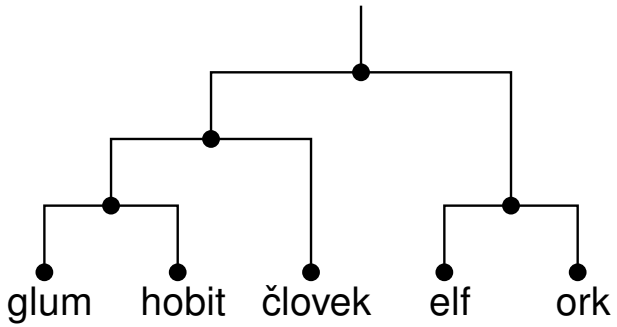
každá dĺžky n

človek	C	A	G	T	T	A
elf	A	A	T	A	G	A
Glum	C	C	G	A	G	A
hobit	C	C	G	T	T	C
ork	A	A	T	T	T	A

Výstup:

strom predstavujúci

ich evolučnú históriu

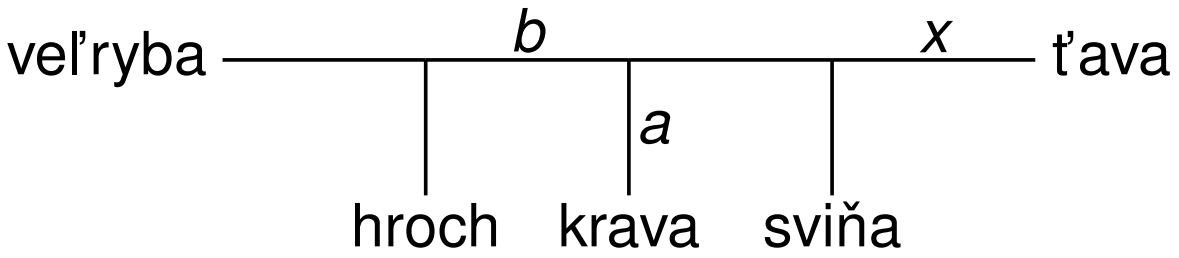


Newick format:

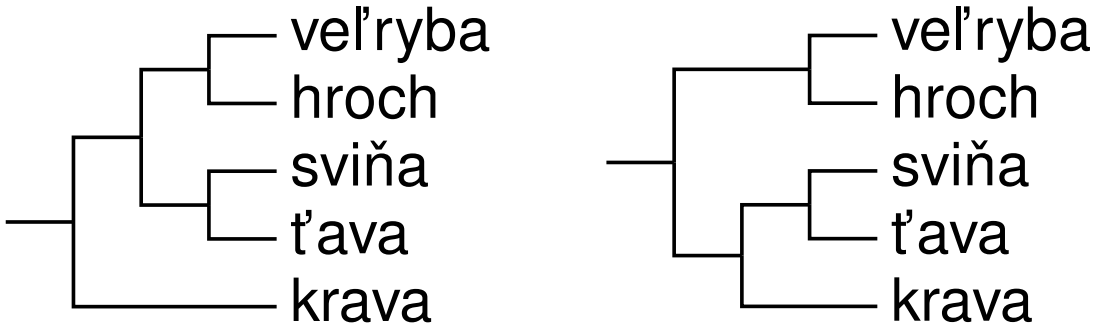
(((glum,hobit),človek),(elf,ork))

Zakorenené a nezakorenené stromy

Nezakorenený strom (unrooted tree)



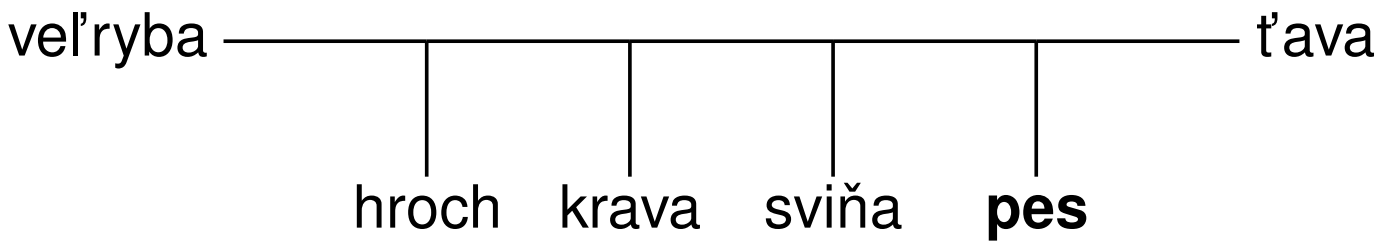
Dve zo siedmich zakorenených verzií stromu

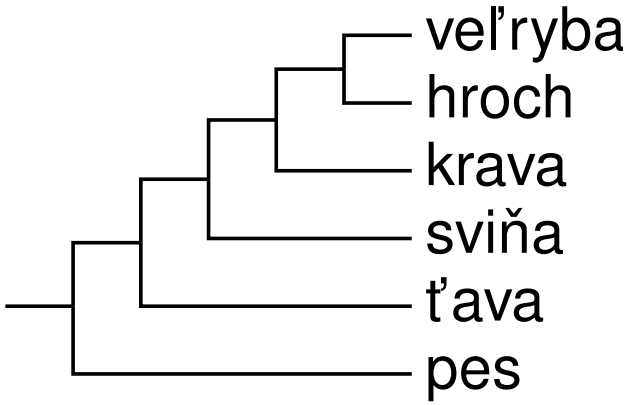


Väčšina metód rekonštruuje nezakorenené stromy

Zakorenenie stormu pomocou vonkajšej skupiny

Do nezakoreneného stromu pridáme psa, **vonkajšiu skupinu (outgroup)**





Maximum parsimony (úsporné stromy)

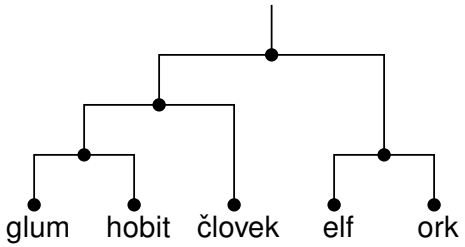
Úloha: Dané sú zarovnané sekvencie súčasných organizmov.

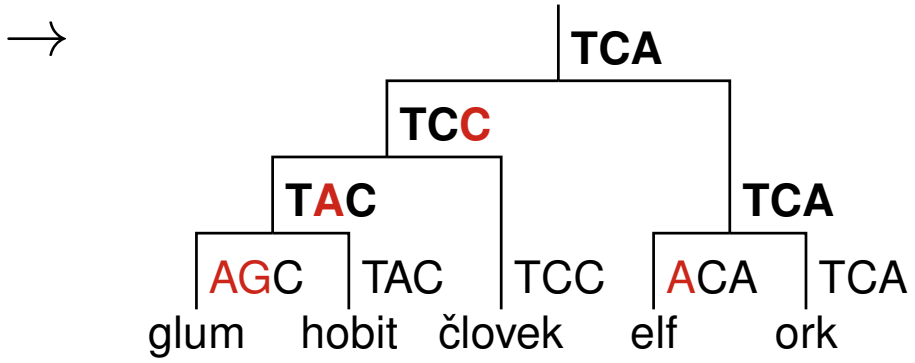
Chceme nájsť fylogenetický strom, ktorý vyžaduje **minimálny počet evolučných zmien**.

Evolučná zmena = mutácia jednej bázy na inú bázu

Podotázka: Pre daný fylogenetický strom, doplniť **ancestrálne sekvencie** tak, aby bol potrebný najmenší počet zmien.

glum	AGC
hobit	TAC
človek	TCC
elf	ACA
ork	TCA

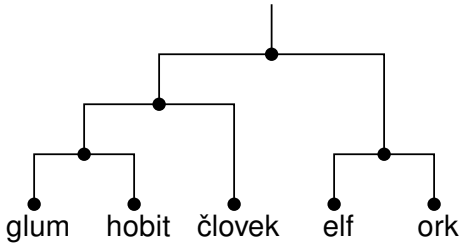


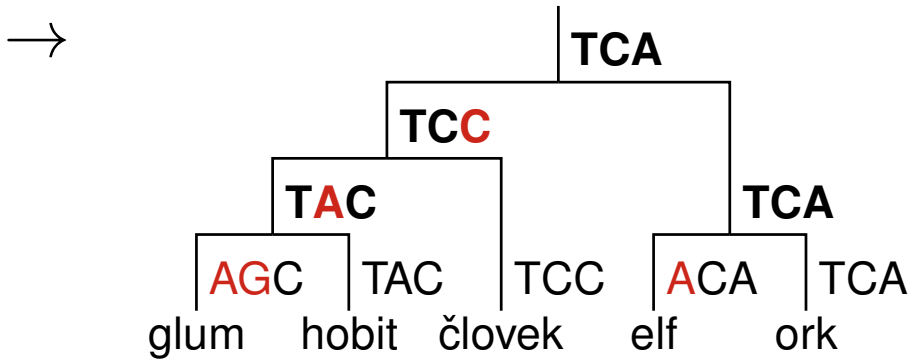


5 zmien

Podotázka: Výpočet ceny konkrétneho stromu

glum	AGC
hobit	TAC
človek	TCC
elf	ACA
ork	TCA





5 zmien

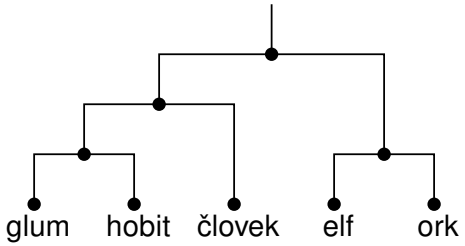
Môžeme rátať **dynamickým programovaním** pre každý stĺpec zarovnania zvlášť (cvičenia informatici).

Časová zložitosť: $O(m)$, lineárna

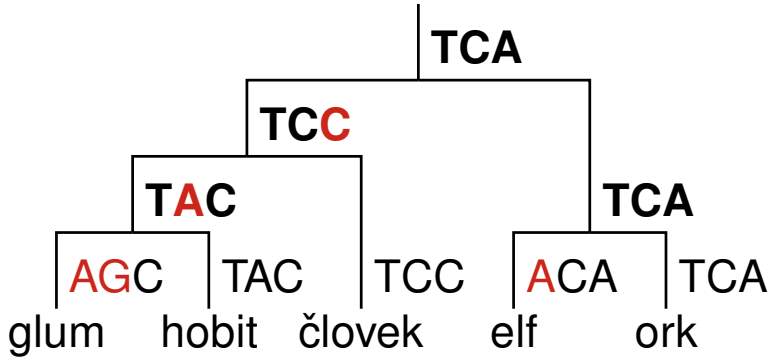
Zopakujeme pre každý stĺpec zarovnania: $O(mn)$

Vieme: Výpočet ceny konkrétneho stromu

glum	AGC
hobit	TAC
človek	TCC
elf	ACA
ork	TCA



→

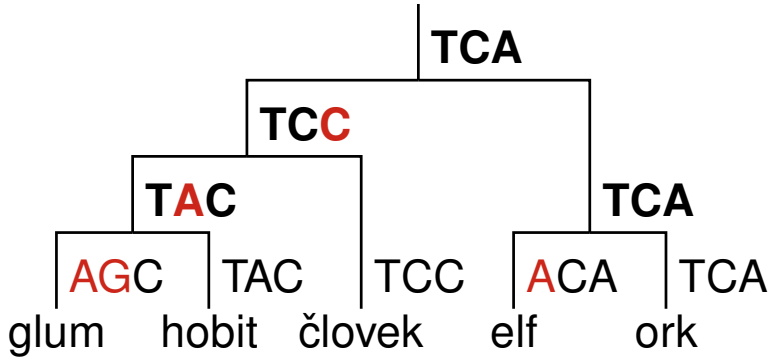


5 zmien

Chceme: Nájsť strom s najmenšou cenou

glum	AGC
hobit	TAC
človek	TCC
elf	ACA
ork	TCA

→



Hľadanie najúspornejšieho stromu

NP-ťažký problém

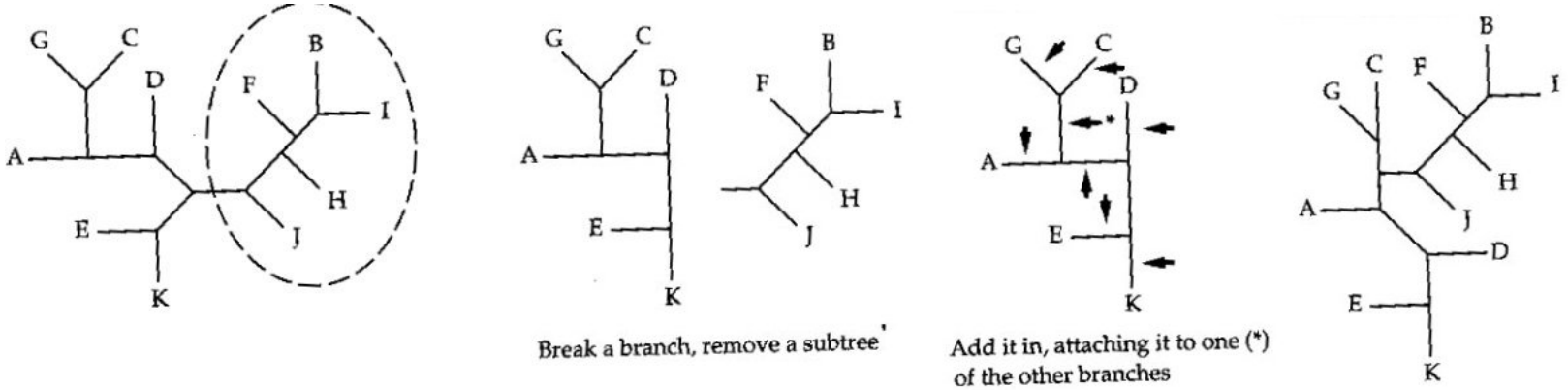
Triviálny algoritmus: vyskúšaj všetky možné stromy.

Pre m druhov $1 \cdot 3 \cdot 5 \cdots (2m-5) = (2m-5)!!$

Napr. pre 10 druhov cca 2 milióny, pre 20 druhov $2 \cdot 10^{20}$

Heuristické prehľadávanie:

- Začneme s “rozumným” stromom
- Pomocou stanovených operácií prehľadávame “podobné” stromy; napr. “subtree pruning and regraft”:



Neighbor Joining (Metóda spájania susedov)

- Nevyužívame detaily rozdielov medzi sekvenciami
- Zosumarizujeme ich pomocou **matice vzdialeností** (D_{ij})

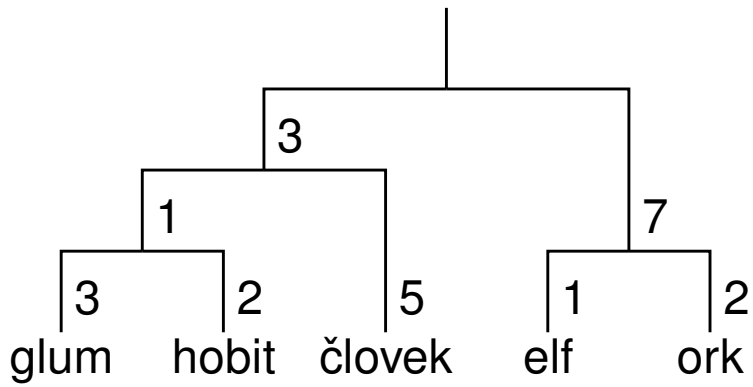
Jednoduchý príklad:

človek	C	A	G	T	T	A
elf	A	A	T	A	G	A
Glum	C	C	G	A	G	A
hobit	C	C	G	T	T	C
ork	A	A	T	T	T	A

	Č	E	G	H	O
človek	0	4	3	2	2
elf	4	0	3	6	2
Glum	3	3	0	3	5
hobit	2	6	3	0	4
ork	2	2	5	4	0

Idea spájania susedov

- Predpokladáme, že vzdialenosti $D_{i,j}$ skutočne zodpovedajú vzdialenostiam v strome (**aditivita**)



	glum	hobit	človek	elf	ork
glum	0	5	9	15	16
hobit	5	0	8	14	15
človek	9	8	0	16	17
elf	15	14	16	0	3
ork	16	15	17	3	0

$D_{\text{hobit,človek}} = 2 + 1 + 5 = 8$

Idea spájania susedov

- Predpokladáme, že vzdialenosti $D_{i,j}$ skutočne zodpovedajú vzdialenostiam v strome (**aditivita**)
- Nájdeme dva listy i a j, o ktorých vieme **s určitosťou povedať**, že majú vo výslednom strome spoločného rodiča
- i a j spojíme a nahradíme ich ich rodičom k s novými vzdialenosťami:

$$D_{k,\ell} = \frac{D_{i,\ell} + D_{j,\ell} - D_{i,j}}{2}$$

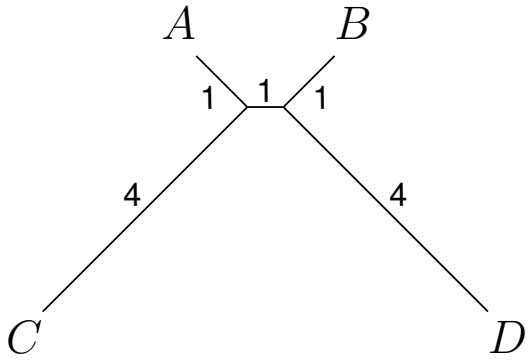
	g	h	č	e	o
g	0	5	9	15	16
h	5	0	8	14	15
č	9	8	0	16	17
e	15	14	16	0	3
o	16	15	17	3	0

Spojíme e a o
$\longrightarrow$

	g	h	č	eo
g	0	5	9	14
h	5	0	8	13
č	9	8	0	15
eo	14	13	15	0

Ako určiť dva listy na spájanie?

Prečo nie dva najbližšie?



D:

	A	B	C	D
A	-	3	5	6
B	3	-	6	5
C	5	6	-	9
D	6	5	9	-

L:

	A	B	C	D
A	-	-22	-24	-22
B	-22	-	-22	-24
C	-24	-22	-	-22
D	-22	-24	-22	-

Vyber listy i, j, ktoré **minimalizujú** nasledujúci výraz:

$$L_{i,j} = (m-2)D_{i,j} - \underbrace{\sum_k D_{i,k}}_{r_i} - \underbrace{\sum_k D_{j,k}}_{r_j}$$

m: počet listov

r_i: súčet riadku resp. stĺpca i

Spájame listy i, j, ktoré majú najnižšiu hodnotu v matici L

$$L_{i,j} = (m-2)D_{i,j} - \underbrace{\sum_k D_{i,k}}_{r_i} - \underbrace{\sum_k D_{j,k}}_{r_j}$$

D

	g	h	č	e	o	r_i
g	0	5	9	15	16	45
h	5	0	8	14	15	42
č	9	8	0	16	17	50
e	15	14	16	0	3	48
o	16	15	17	3	0	51

L

	g	h	č	e	o
g	.	-72	-68	-58	-48
h	-72	.	-68	-48	-48
č	-68	-68	.	-50	-50
e	-58	-48	-50	.	**-90**
o	-48	-48	-50	**-90**	.

nové D

	g	h	č	eo
g	0	5	9	14
h	5	0	8	13
č	9	8	0	15
eo	14	13	15	0

Časová zložitosť spájania susedov: $O(m^3)$ (m: počet listov)

V roku 2009 Elias a Lagergren vynašli algoritmus so zložitosťou $O(m^2)$

Spájanie susedov: zhrnutie

- Ak je vstupná matica aditívna a zodpovedá skutočným evolučným vzdialenostiam, spájanie susedov nám dá správny strom
- Čím dlhšie sekvencie, tým spoľahlivejší odhad vzdialenosti a tým väčšia šanca dostať správny strom
- Ako však prejdeme od sekvencií k odhadu vzdialenosti?
  Len počítanie rozdielov nestačí

človek	C	A	G	T	T	A
elf	A	A	T	A	G	A
Glum	C	C	G	A	G	A
hobit	C	C	G	T	T	C
ork	A	A	T	T	T	A

	Č	E	G	H	O
človek	0	4	3	2	2
elf	4	0	3	6	2
Glum	3	3	0	3	5
hobit	2	6	3	0	4
ork	2	2	5	4	0

Problém so vzdialenosťami

- Počas evolúcie sa môže stať, že tá istá báza zmutuje **viackrát** (trebárs aj späť na pôvodnú bázu)
- Pri počítaní rozdielov ale vidíme nanajvýš jednu zmenu na každej pozícii $\Rightarrow$ odhad vzdialenosti menší ako v skutočnosti
- Chceme korekciu na odhadovaný počet mutácií, ktoré sa naozaj stali

Jukesov-Cantorov model evolúcie

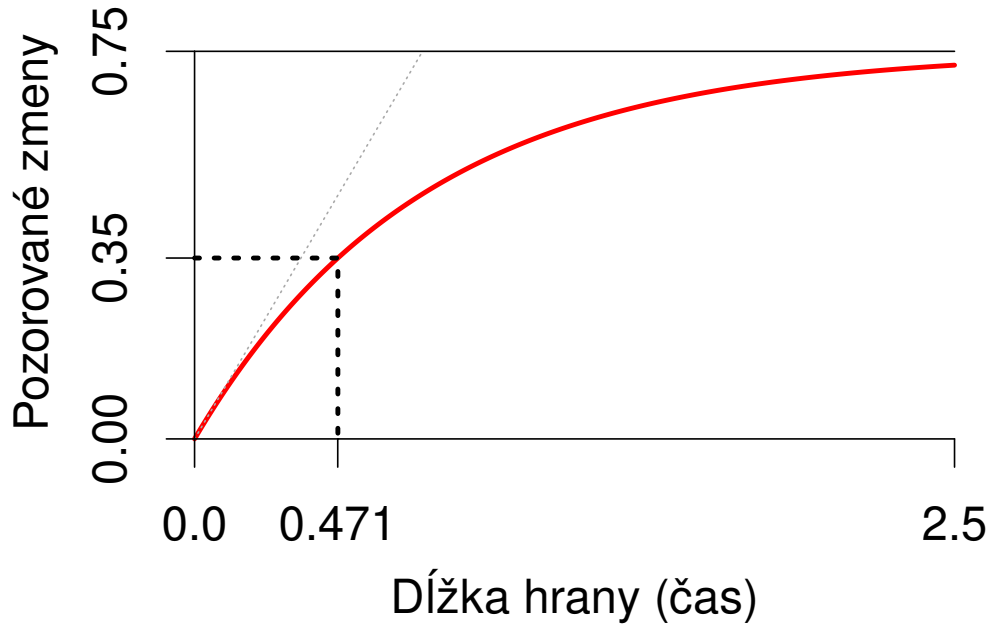
Pravdepodobnosť zmeny bázy na inú:

$$\Pr(X_{t_0+t} = C \mid X_{t_0} = A) = \frac{1}{4}\left(1 - e^{-\frac{4}{3}\alpha t}\right)$$

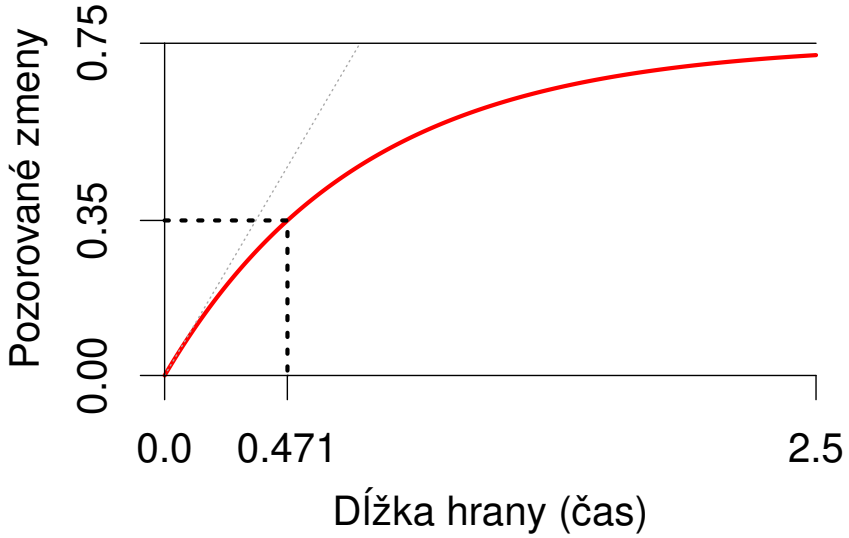
α: rýchlosť evolúcie (počet substitúcií na jednotku času)

Očakávaný počet pozorovaných zmien na bázu za čas t:

$$D(t) = \Pr(X_{t_0+t} \neq X_{t_0}) = \frac{3}{4}\left(1 - e^{-\frac{4}{3}\alpha t}\right)$$



Späť ku spájaniu susedov (Neighbor Joining)



- Podľa takéhoto modelu môžeme korigovať pozorované vzdialenosti

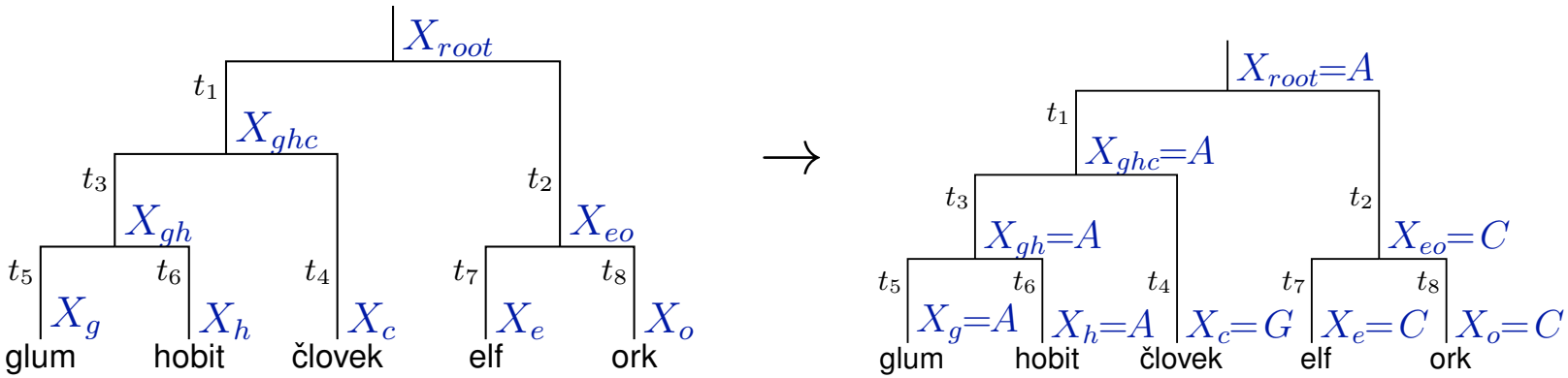
$$D = \frac{3}{4}\left(1 - e^{-\frac{4}{3}\alpha t}\right) \qquad \Rightarrow \qquad \alpha t = -\frac{3}{4}\ln\left(1 - \frac{4}{3}D\right)$$

- Nabudúce / na cvičeniach uvidíme aj zložitejšie modely evolúcie

Najvierohodnejšie stromy (Maximum likelihood)

Strom s danými dĺžkami hrán môžeme chápať

ako **jednoduchý generatívny model**

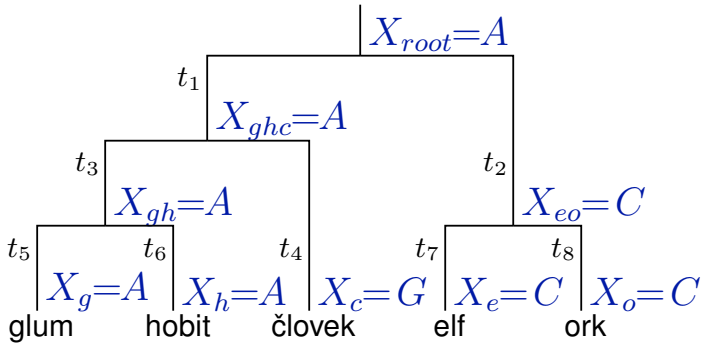


Pravdepodobnosť, že vygeneruje konkrétne bázy vo vrcholoch:

$$\Pr(X_g = A, X_h = A, X_c = G, X_e = C, X_o = C, X_{gh} = A,$$
$$X_{ghc} = A, X_{eo} = C, X_{root} = A)$$
$$= \Pr(X_{root} = A) \cdot \Pr(A \xrightarrow{t_1} A) \cdot \Pr(A \xrightarrow{t_2} C) \cdot \Pr(A \xrightarrow{t_3} A) \cdot \Pr(A \xrightarrow{t_4}$$
$$G) \cdot \Pr(A \xrightarrow{t_5} A) \cdot \Pr(A \xrightarrow{t_6} A) \cdot \Pr(C \xrightarrow{t_7} C) \cdot \Pr(C \xrightarrow{t_8} C)$$

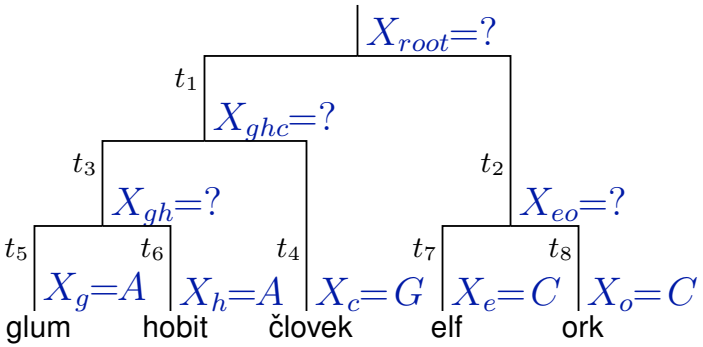
$\Pr(A \xrightarrow{t_2} C)$ je skratka z $\Pr(X_{t+t_2} = C \mid X_t = A)$ t.j. $\Pr(X_{eo} = C \mid X_{root} = A)$

Vieme počítať (súčin):



Chceme počítať

vierohodnosť stromu:



Vierohodnosť (likelihood) stromu:

$\Pr(X_g = A, X_h = A, X_c = G, X_e = C, X_o = C)$

sčítame pravdepodobnosti pre všetky kombinácie písmen v predkoch X_{gh}, X_{ghc}, X_{eo}, X_{root}

Rátame pomocou **Felsensteinovho algoritmu**

(jednoduché dynamické programovanie, podobne ako pre úspornosť)

Pre dané zarovnanie, strom a dĺžky hrán

spočíta vierohodnosť v čase $O(nm)$

Ako nájsť najvierohodnejší strom?

- Problém je NP-ťažký ;

  navyše komplikovaný tým, že na výpočet vierohodnosti **potrebujeme aj dĺžky hrán**

- Opäť použijeme heuristické vyhľadávanie:
  - Začneme s “rozumným” stromom
  - Vypočítame vierohodnosť tohto stromu:
    - Začneme s “rozumnými” dĺžkami hrán
    - Vypočítame vierohodnosť stromu s dĺžkami
    - Mierne zmeníme dĺžky tak, aby sa zlepšila vierohodnosť a opakujeme
  - Pomocou stanovených operácií (ako v prípade parsimony) skúšame “podobné” stromy, až kým nevieme zlepšiť

Konzistentnosť fylogenetických algoritmov

- "Rozumne" správajúce sa algoritmy: ak dĺžka sekvencií n rastie, ich odpoveď by sa mala približovať ku správnej odpovedi.
- Uvažujme dáta generované z modelu podľa nejakého stromu (t.j. nie reálne biologické dáta, ale simulované)
- Hovoríme, že algoritmus pre hľadanie fylogenetického stromu je **konzistentný**, ak pre n idúce do nekonečna pravdepodobnosť správneho stromu konverguje k 1.

Porovnanie algoritmov

	Zložitosť	**Konzistentný**	**Využitie dát**
Parsimony (úspornosť)	NP-ťažký	NIE	celé sekvencie
Neighbor Joining	$O(m^2)$	ÁNO	iba vzdialenosti
Likelihood (vierohodnosť)	NP-ťažký	ÁNO	celé sekvencie

Odkiaľ zohnať dáta pre fylogenetiku?

Často sa používajú špeciálne sekvencie
(napr. gény ribozomálnej RNA, mitochondriálny genóm)

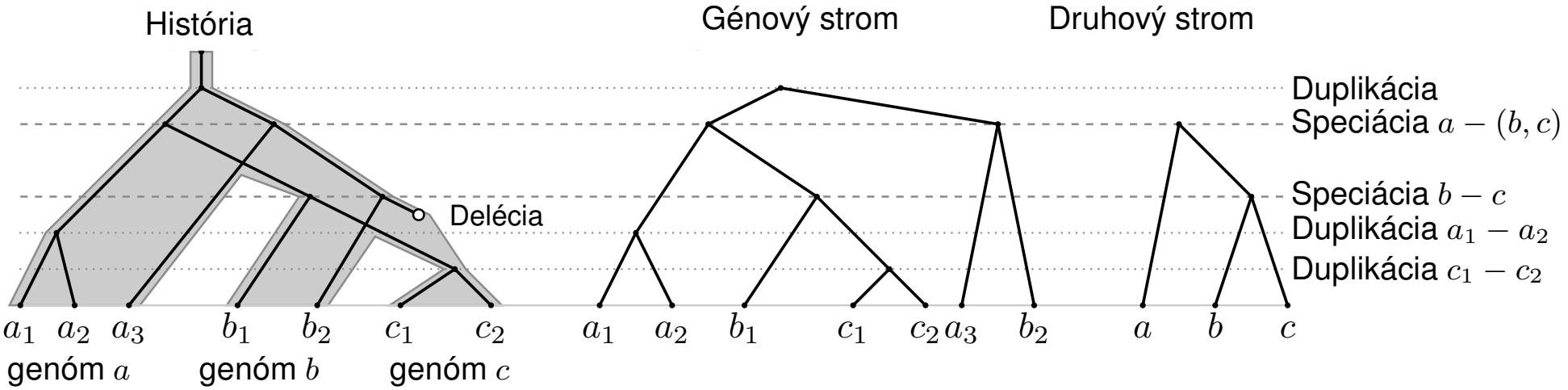
Chceme využiť aj dalšie časti genómu. Čo tak:

- Vybrať si sympatický gén
- Nájsť jeho homológy v iných genómoch
- Použiť tieto na konštrukciu fylogenetického stromu
  (DNA sekvencie alebo proteíny)

Problém: počas evolúcie sa časť genómu s vybraným génom mohla duplikovať

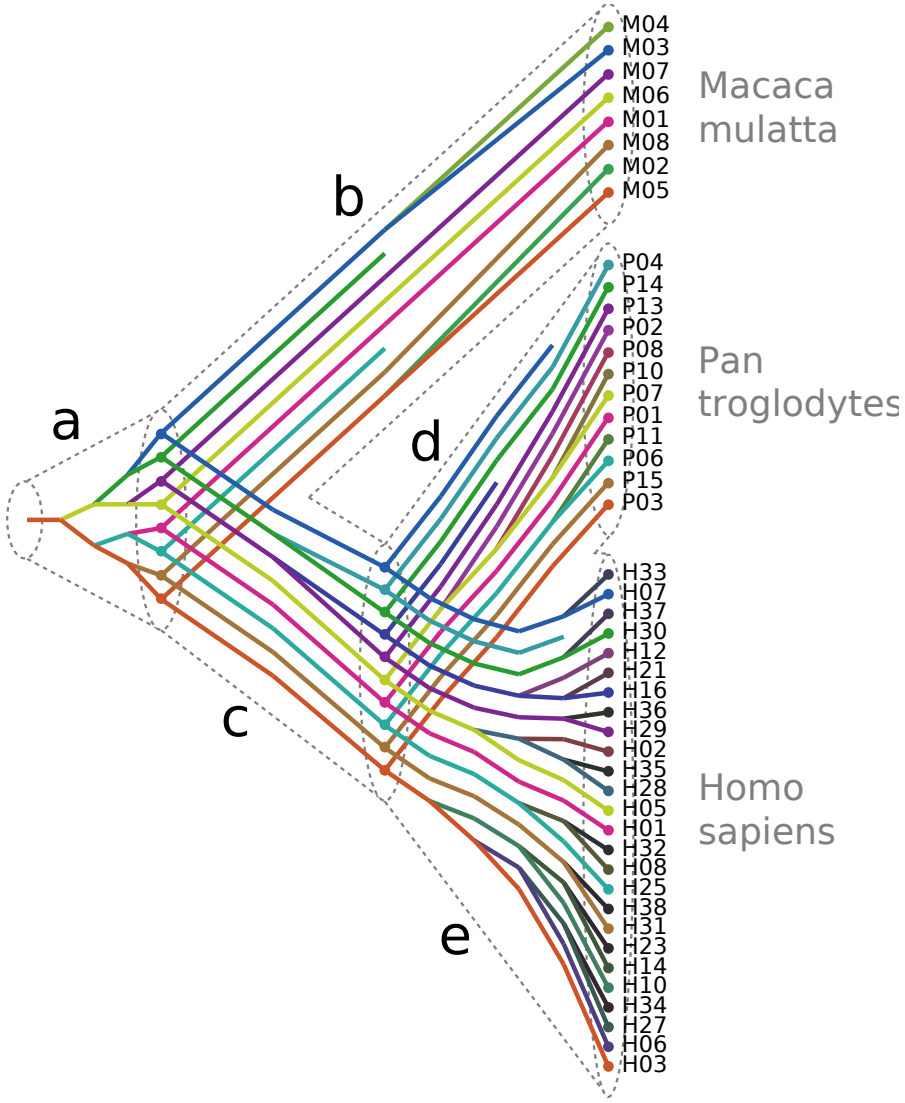
História duplikovaného génu

Príklad: organizmy a, b, c, gény $a_1, a_2, a_3, b_1, b_2, c_1, c_2$



- **Homológ:** vyvinuli sa zo spoločného predka, podobná sekvencia
- **Ortológ:** najbližší spoločný predok je speciácia
  (napr. dvojice génov $a_1 - b_1$, $a_2 - b_1$)
- **Paralóg:** najbližší spoločný predok je duplikácia
  (napr. dvojice génov $a_1 - a_2$, $a_1 - b_2$)

Zložitejší príklad duplikácie génu:



Zhrnutie

- Modely evolúcie nukleotidov nám dávajú možnosť:
  - Odhadovať skutočnú evolučnú vzdialenosť (počet substitúcií) z počtu pozorovaných zmien medzi sekvenciami
  - Počítať pravdepodobnosť, že uvidíme zmenu nukleotidu za určitý čas t
- Tri metódy na vytváranie evolučných stromov:
  - Úsporné stromy (parsimony)
  - Spájanie susedov (neighbour joining)
  - Vierohodnosť stromov (maximum likelihood)
- Praktické komplikácie: génové a druhové stromy, hľadanie ortológov, zakoreňovanie stromu
- Moderné trendy: efektívne algoritmy na spracovanie veľkých dát (veľa génov a organizmov naraz)