

## Metódy v bioinformatike, 1-BIN-301/2-AIN-501

### Vyučujú:

Broňa Brejová, M-163, brejova@fmph.uniba.sk

Tomáš Vinař, M-163, vinar@fmph.uniba.sk

Vladimír Boža, M-25, boza@fmph.uniba.sk

**Web:** <http://compbio.fmph.uniba.sk/vyuka/mbi/>

**Diskusné fórum a oznamy:** Facebook

### Literatúra:

I-INF-D-23: Durbin, Eddy, Krogh, Mitchison: Biological sequence analysis. Cambridge University Press 1998.

I-INF-Z-2: Zvelebil, Baum: Understanding Bioinformatics. Taylor&Francis 2008.

Skriptá k predmetu a poznámky na webstránke.

## Ciele predmetu

- **Všetci:** Prehľad základných metód na výpočtovú analýzu biologických sekvencií a ďalších dát v molekulárnej biológii.
- **Informatici:** Algoritmy a dátové štruktúry, strojové učenie, pravdepodobnosť. Ako prejsť od problému v reálnom svete k matematickej abstrakcii.
- **Biológovia:** Matematické modely tvoriace základ populárnych bioinformatických nástrojov, používanie nástrojov, interpretácia výsledkov.
- **Všetci:** Skúsenosť s interdisciplinárnou spoluprácou.

## Známkovanie

3 domáce úlohy 30% (10% každá)

Journal club 10%

Skúška 60%

**Hodnotenie:** A: 90+, B: 80+, C: 70+, D: 60+, E: 50+

- Dve verzie otázok: biologická a informatická
- Journal club: čítanie 1 článku v skupine a správa (prípadne nepovinná prezentácia)
- Na skúške povolený ťahák 2 listy A4
- Neopisovať!

## Časy a miestnosti

- Prednáška štvrtok 15:40-17:10 F1-328
- Cvičenia informatici štvrtok 14:00-15:30 F1-328
- Cvičenia biológovia štvrtok 17:20-18:50 F1-328 a M-217  
(ak prednáška skončí skôr, cvičenia začnú po krátkej prestávke)

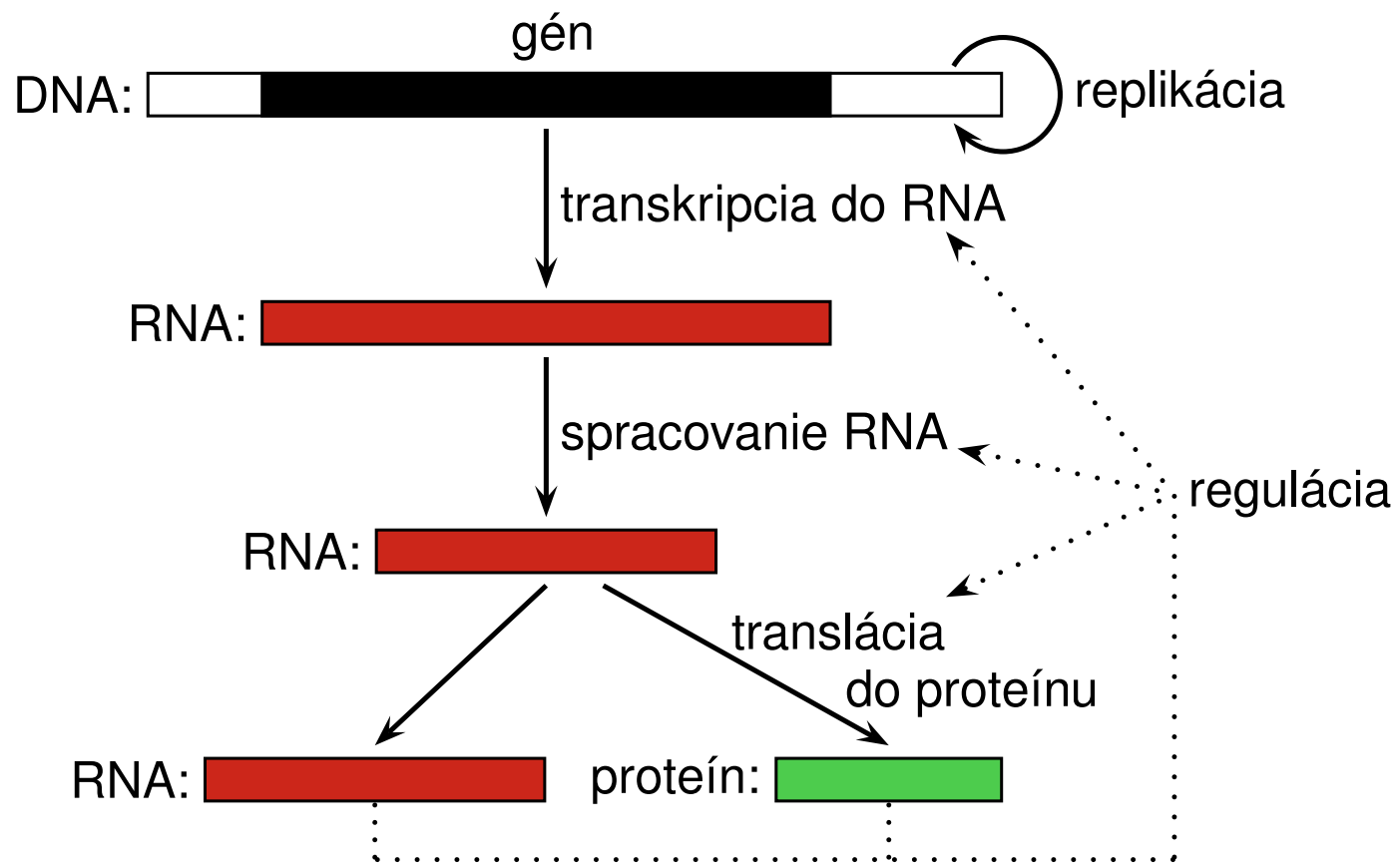
## Čo nás v tomto predmete čaká

### Typická prednáška

- Biologické pozadie problému
- Formulácia ako informatický problém
- Idea algoritmu (riešenia problému)

### Typické cvičenia

- Informatici: ďalšie detaily algoritmov, potrebné poznatky z biológie
- Biológovia: aplikácia na konkrétne dáta, význam rôznych parametrov, potrebné poznatky z informatiky



## Príklad z nášho výskumu

Kosmáč bielofúzy

(common marmoset, *Callithrix jacchus*, štvrt' kila, 18cm)



Genóm osekvenovaný 2007

(Washington University St. Louis a Baylor College of Medicine, USA)

Analýza publikovaná v roku 2014

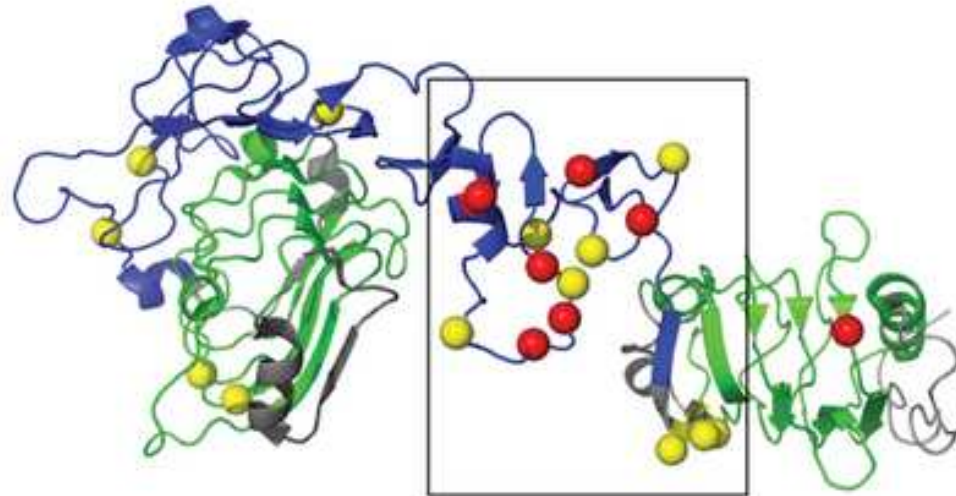
# IGF1R: Insulin-like growth factor 1 receptor

Proteín prechádza cez cytoplazmatickú membránu na povrchu bunky

Po naviazaní hormónov IGF1, IGF2 signalizuje dovnútra bunky

Súvisí s rastom a delením bunky, rastom organizmu, rakovinou

human	RDFCANILSAESSDSEGFVIHDGECMQECPSSGFIRNGSQSMYCIPCEGPCPKVC-EEEKKT
chimp	RDFCANILSAESSDSEGFVIHDGECMQECPSSGFIRNGSQSMYCIPCEGPCPKVC-EEEKKT
orang	RDFCANILSAESSDSEGFVIHDGECMQECPSSGFIRNGSQSMYCIPCEGPCPKVC-EEEKKT
macaque	RDFCANILSAESSDSEGFVIHDGECMQECPSSGFIRNGSQSMYCIPCEGPCPKVC-EEEKKT
marmoset	RQFCASIVSSSENENKGFVIHDGECMQDCPSGFIRDTHSMQCIPCKGPCPKVC-D-EQMAK
mouse	RDFCANIPNAESSDSDGFVIHDDEECMQECPSSGFIRNSTQSMYCIPCEGPCPKVCGDEEKKTK
rat	RDFCANIPNAESSDSDGFVIHDGECMQECPSSGFIRNSTQSMYCIPCEGPCPKVCGDEEKKTK
dog	RDFCANIPSAESSDSEGFVIHDGECMQECPSSGFIRNGSQSMYCIPCEGPCPKVC-EEEKKT





## **Aké bioinformatické nástroje boli potrebné k tomuto výsledku?**

1. Zostavenie genómu
2. Nájdenie zarovnaní s inými genómami
3. Hľadanie génov kódujúcich proteíny
4. Hľadanie génov s pozitívnym výberom
5. Určovanie štruktúry a funkcie proteínov

## 1. Zostavenie genómu

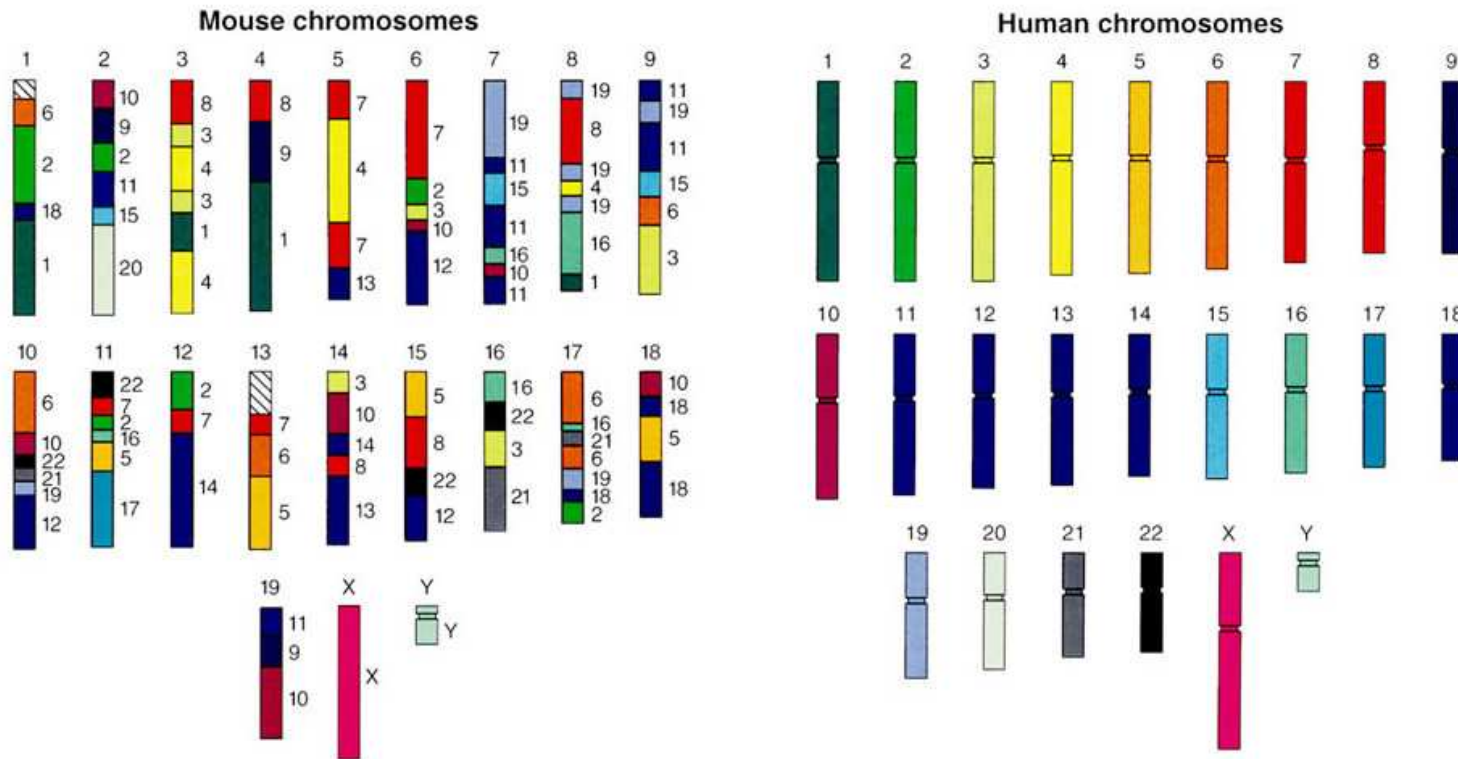
- Pri sekvenovaní DNA vieme čítať len krátke kúsky (napr. dĺžky 1000)
- Každé miesto v genóme prečítame viackrát (u kosmáča priemer  $6\times$ )



- Čítania “zliepame” na základe prekryvov
- Veľmi veľa dát  $\Rightarrow$  potreba veľmi efektívnych programov

## 2. Nájdenie zarovnaní s inými genómami

Ku každému miestu v genóme kosmáča chceme nájsť zodpovedajúce časti iných genómov (napr. človek, šimpanz, myš, ...)



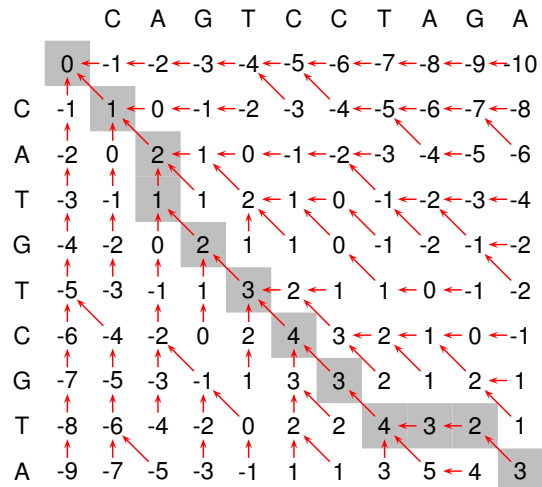
## 2. Nájdenie zarovnaní s inými genómami

- Hľadáme podobnosti medzi DNA sekvenciami

```

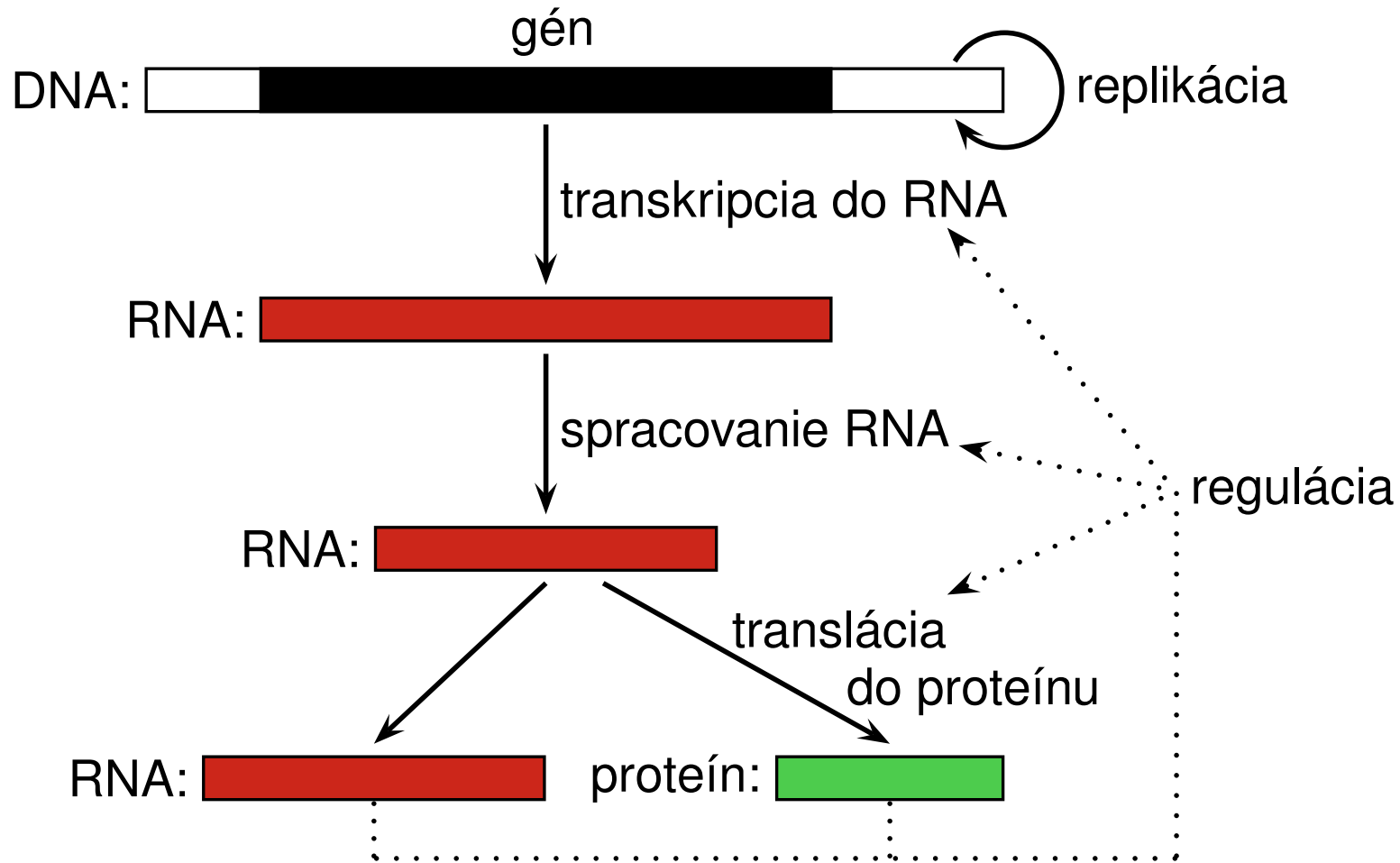
Human AGTGGCTGCCAGGCTG---GGATGCTGAGGCCTTGTTTGCAGGGA
Rhesus AGTGGCTGCCAGGCTG---GGTTGCTGAGGCCTTGTTTGCCGGGA
Mouse  GGTGGCTGCCGGGCTG---GGTGGCTGAGGCCTTGTTGGTGGGGT
Dog    AGTGGCTGCCCGGCTG---GGTGGCTGAGGCCTTATTTGCAGGGA
Chicken AGTGGCTGCCAGTCTGCGCCGTGGCCGACGTCTTGCTCGGGGGAA
  
```

- Základ je technika dynamického programovania, ktorá veľký problém rozkladá na veľa malých podproblémov



- Tabuľka je veľmi veľká, v praxi treba pridať veľa vylepšení

### 3. Hľadanie génov kódujúcich proteíny



Ktoré časti osekvenovaného genómu kódujú proteíny?

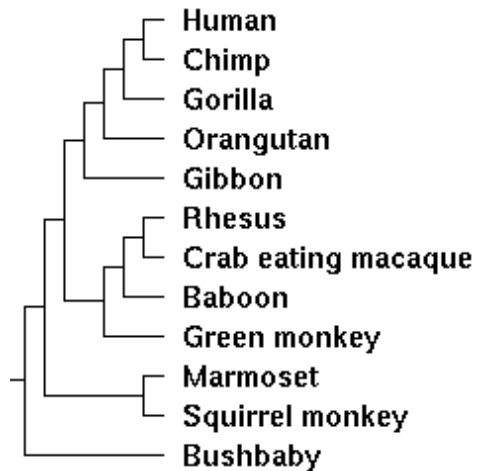
### 3. Hľadanie génov kódujúcich proteíny

- Hľadanie ihly v kope sena: iba cca 1% ľudského genómu kóduje proteíny
- Kód pre jeden proteín rozbitý do veľa krátkych exónov
- Napr. IGF1R zaberá 315 569nt, z toho kóduje 4101nt v 21 exónoch



- Zoberieme známe gény, spravíme rôzne štatistiky potom hľadáme iné oblasti s podobným štatistickým profilom

## 4. Hľadanie génov s pozitívnym výberom



- Štúdium evolučných procesov
- V DNA vznikajú mutácie, tie však podliehajú prirodzenému výberu
- Väčšina náhodných zmien v proteíne je škodlivých, preto sa proteíny menia pomerne pomaly

## 4. Hľadanie génov s pozitívnym výberom

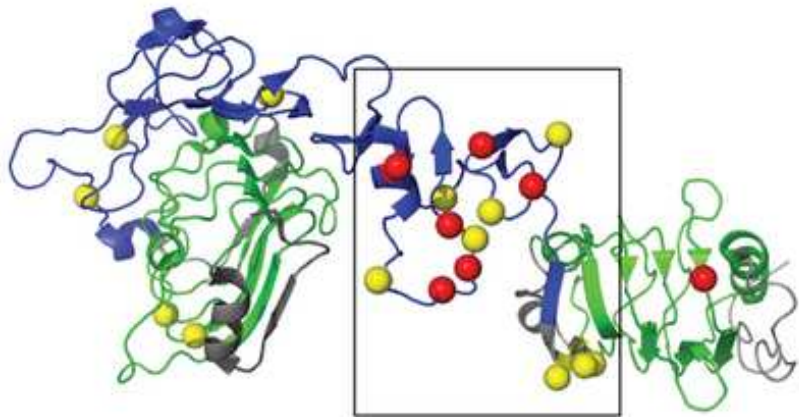
- Niekedy sa však proteín mení rýchlejšie, nakoľko náhodné zmeny sú užitočné (pozitívny výber)
- Veľké množstvo zmien v proteíne môže znamenať zmeny vo funkcii

human	R	D	F	C	A	N	I	L	S	A	E	S	S	D	S	E	G	F	V	I	H	D	G	E	C	M	Q	E	C	P	S	G	F	I	R	N	G	S	Q	S	M	Y	C	I	P	C	E	G	P	C	P	K	V	C	-	E	E	E	K	K	T	K
chimp	R	D	F	C	A	N	I	L	S	A	E	S	S	D	S	E	G	F	V	I	H	D	G	E	C	M	Q	E	C	P	S	G	F	I	R	N	G	S	Q	S	M	Y	C	I	P	C	E	G	P	C	P	K	V	C	-	E	E	E	K	K	T	K
orang	R	D	F	C	A	N	I	L	S	A	E	S	S	D	S	E	G	F	V	I	H	D	G	E	C	M	Q	E	C	P	S	G	F	I	R	N	G	S	Q	S	M	Y	C	I	P	C	E	G	P	C	P	K	V	C	-	E	E	E	K	K	T	K
macaque	R	D	F	C	A	N	I	L	S	A	E	S	S	D	S	E	G	F	V	I	H	D	G	E	C	M	Q	E	C	P	S	G	F	I	R	N	G	S	Q	S	M	Y	C	I	P	C	E	G	P	C	P	K	V	C	-	E	E	E	K	K	T	K
marmoset	R	Q	F	C	A	S	I	V	S	S	E	N	S	E	N	N	K	F	V	I	H	D	G	E	C	M	Q	D	C	P	S	G	F	I	R	D	T	T	H	S	M	Q	C	I	P	C	K	G	P	C	P	K	V	C	-	D	-	E	Q	M	A	K
mouse	R	D	F	C	A	N	I	P	N	A	E	S	S	D	S	D	G	F	V	I	H	D	D	E	C	M	Q	E	C	P	S	G	F	I	R	N	S	T	Q	S	M	Y	C	I	P	C	E	G	P	C	P	K	V	C	G	D	E	E	K	K	T	K
rat	R	D	F	C	A	N	I	P	N	A	E	S	S	D	S	D	G	F	V	I	H	D	G	E	C	M	Q	E	C	P	S	G	F	I	R	N	S	T	Q	S	M	Y	C	I	P	C	E	G	P	C	P	K	V	C	G	D	E	E	K	K	T	K
dog	R	D	F	C	A	N	I	P	S	A	E	S	S	D	S	E	G	F	V	I	H	D	G	E	C	M	Q	E	C	P	S	G	F	I	R	N	G	S	Q	S	M	Y	C	I	P	C	E	G	P	C	P	K	V	C	-	E	E	E	K	K	T	K



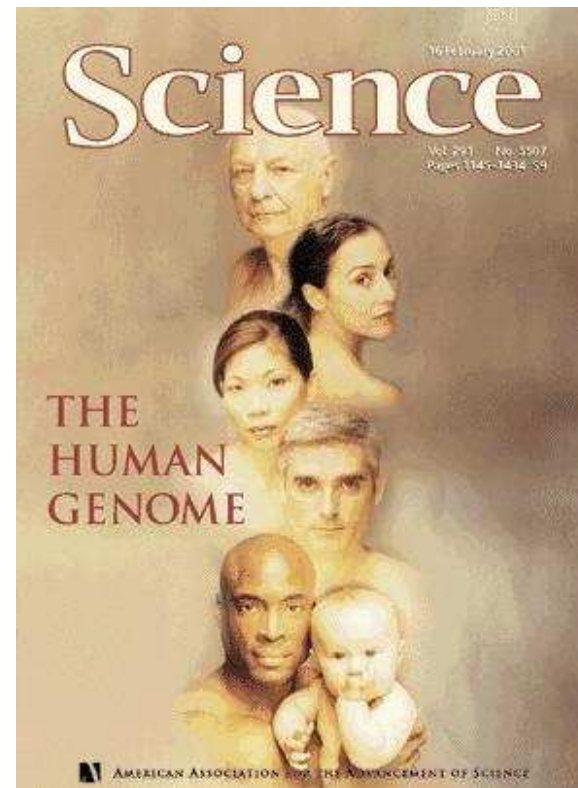
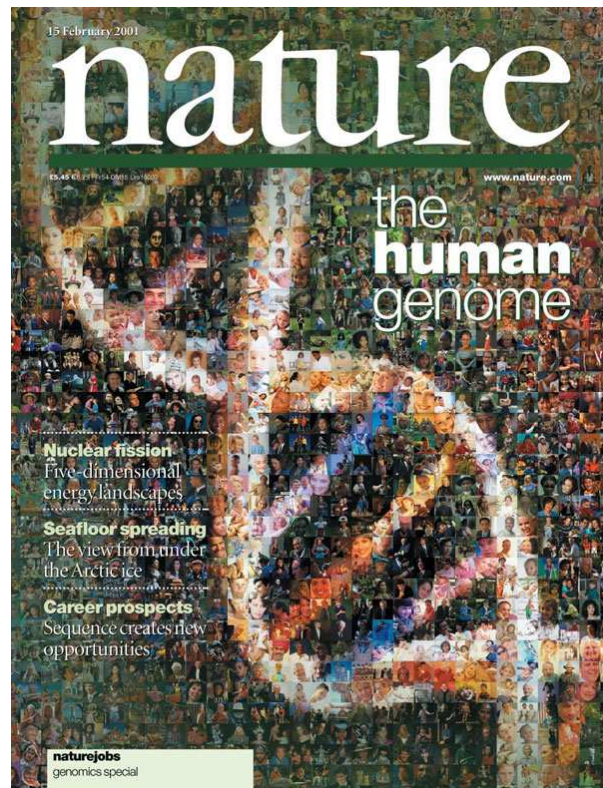
## 5. Určovanie štruktúry a funkcie proteínov

- Spravili sme kroky 1-4 a dostali sme zoznam 37 génov pod vplyvom pozitívneho výberu v kosmáčovi
- Čo tie gény robia, ktoré by mohli súvisieť s veľkosťou?
- Aký má daný proteín tvar, kde sú pozície, ktoré sa v evolúcii zmenili?
- Štruktúra (tvar) proteínov sa dá určovať experimentálne je to drahé, namiesto toho predikcia 3D štruktúry



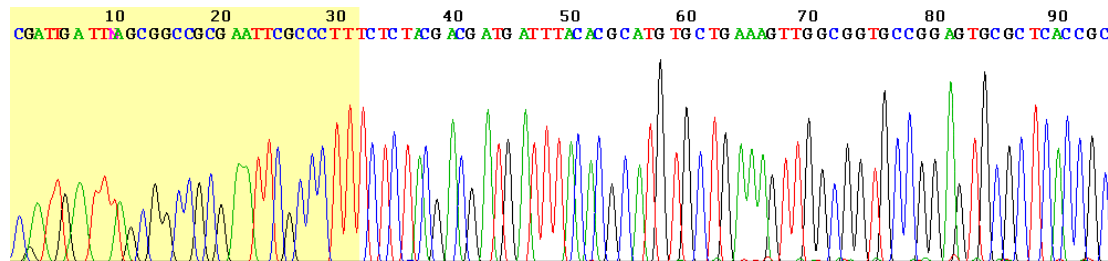
# Sekvenovanie a zostavovanie genómov (genome sequencing and assembly)

Tomáš Vinar̃  
27.9.2018



## Sangerovo sekvenovanie

- Výsledok: sekvenovací profil (trace)



- Ďalej sa spracuje pomocou programu PHRED:
  - Na každej pozícii (kde sa dá) určí bázu (A,C,G,T)
  - Pre každú bázu odhadne kvalitu  $q$   
( $10^{-q/10}$  je pravdepodobnosť chyby,  
t.j. bázy s kvalitou  $q > 40$  sú správne na 99.99%)
- Sangerovo sekvenovanie produkuje čítania (reads)  
dlhé 500-1000 bp
- Ako osekvenovať dlhú DNA sekvenciu?

## Typický priebeh sekvenovania

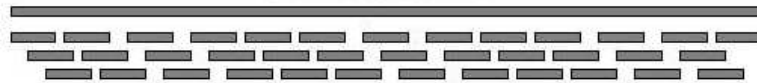
1. Chromozómy náhodne rozsekáme na menšie kúsky  
(napr. pomocou sonikácie)
2. Menšie kúsky namnožíme  
(napr. pomocou PCR, bakteriálneho klonovania a pod.)
3. Konce týchto kúskov osekvenujeme niektorou zo sekvenovacích technológií  $\Rightarrow$  mnoho krátkych reťazcov, ktoré nazývame **čítania**
4. Čítania **výpočtovo zostavíme** späť do chromozómov

## Prehľad sekvenovacích technológií

Technológia	Dĺžka čítania	Chybovosť	Za deň	Cena za MB
<b>1. generácia</b>				
Sanger	do 1000 bp	< 2%	3 MB	\$4000
<b>2. (next) generácia (cca 2004)</b>				
Illumina	2× 150bp	< 1%	30 GB	< \$1
<b>3. generácia (práve nabieha)</b>				
PacBio	6-25 Kbp	10%	1 GB	\$2
Oxford Nanopore	5-100+ kbp	10%	1 GB	\$1

## Bioinformatický problém: Zostavenie genómu (sequence assembly)

- **Vstup:** krátke čítania sekvenovanej DNA
- **Cieľ:** zostaviť pôvodnú DNA
  - riadime sa zhodou v prekrývajúcich častiach čítaní
- Dôležité faktory:
  - **dĺžka genómu**
  - **pokrytie** (coverage) – koľko krát čítania pokrývajú genóm?



## Formulácia problému

### Najkratšie spoločné nadslovo (shortest common superstring)

**Úloha:** Daných je niekoľko reťazcov (čítaní), nájdite **najkratší** reťazec, ktorý obsahuje **všetky** vstupné reťazce ako (súvislé) **podreťazce**.

Motivácia: čo najviac využiť prekryvy medzi čítaniami

### Príklad:

Vstup: GCCAAC, CCTGCC, ACCTTC

Výstup: CCTGCCAACCTTC (najkratšie možné)

## Najkratšie spoločné nadslovo

- **Problém je NP ťažký**  
takže nepoznáme rýchly algoritmus, ktorý vždy nájde najlepšie riešenie
- **Jednoduchá heuristika:** opakovane nájdí dva reťazce, ktoré sa prekrývajú najviac a zlúč ich do jedného reťazca
- Príklad: CATATAT, TATATA, ATATATC  
Optimum: CATATATATC, dĺžka 10  
Heuristika: CATATATCTATATA, dĺžka 14
- V skutočnosti táto heuristika **aproximačný algoritmus:**  
Nájdene riešenie je najviac  $3,5\times$  horšie ako optimálne  
T.j. je to 3,5-aproximačný algoritmus  
(možno aj 2-aproximačný, otvorený problém)
- Existuje aj 2,5-aproximačný algoritmus



## Najkratsie spoločné nadslovo: Čo sme nezahrnuli do formulácie

- V sekvenovaní sa vyskytujú chyby (cca 1 zo 100 báz)
- Polymorfizmus
- Orientácia čítaní (vlákno, strand)
- Kontaminácia cudzou sekvenciou (napr. baktérie, v ktorých sa segmenty DNA klonovali), chiméry
- Viac chromozómov, neúplné pokrytie čítaniami
- Repetitívna sekvencia (sequence repeats, opakovania)  
cca 50% ľudského genómu  
Príklad: 10xTTAATA, 10xATATTA, 3xTTAGCT  
TTAATATTAGCT?  
TTAATATTAATATTAATATTAATATTAGCT?  
TTAATATTA + ATATTAGCT?

## Najkratšie spoločné nadslovo: Zľahčujúce faktory

**Prídavná informácia:** spárované čítania (Pair-end reads)



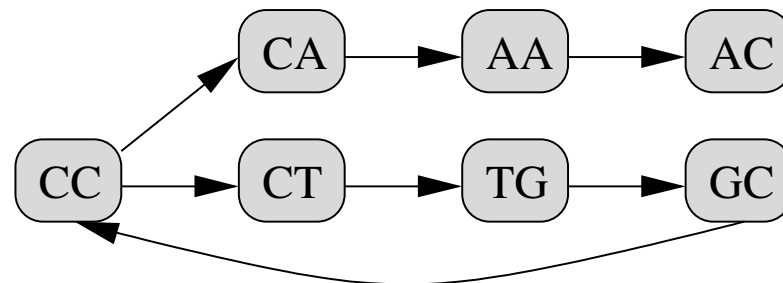
**Zjednodušenie:** nemusíme spojiť všetko do jedného reťazca, spájame len časti spojené viacerými čítaniami  $\Rightarrow$  konzervatívny prístup (radšej menej pospájať, ale nerobiť chyby)

## Najkratšie spoločné nadslovo: Zhrnutie

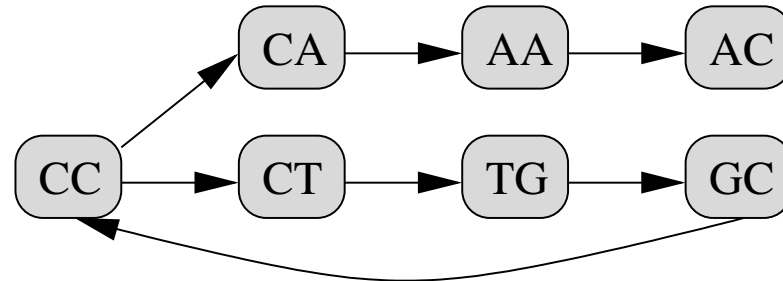
- Nerealistická formulácia, ťažký výpočtový problém
- Ale teoretický problém môže poskytnúť nejaký posun k pochopeniu skutočného problému
- Overlap-Layout-Consensus prístup motivovaný greedy algoritmami pre najkratšie spoločné nadslovo

## Skladanie krátkych čítaní: de Bruijnové grafy

- Nasekajme čítania na (prekrývajúce sa) kúsky dĺžky  $k$
- Zostavme z nich **de Bruijnov graf**
  - **vrcholy**: podreťazce dĺžky  $k$  všetkých čítaní
  - **hrany**: nadväzujúce  $k$ -tice v rámci každého čítania (s prekryvom  $k - 1$ )
  - Graf je orientovaný (hrany majú smer)
- **Príklad**:  $k = 2$ , čítania: CCTGCC, GCCAAC



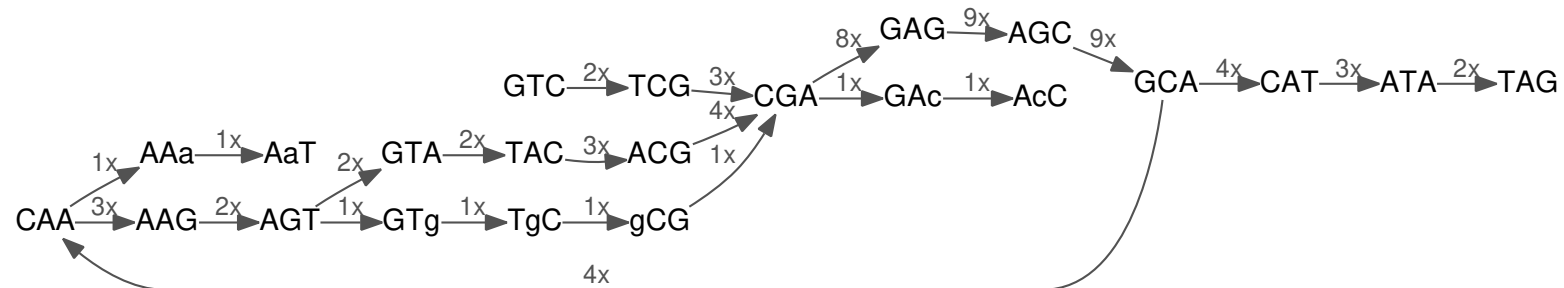
## Ako použiť de Bruijnovu grafy?



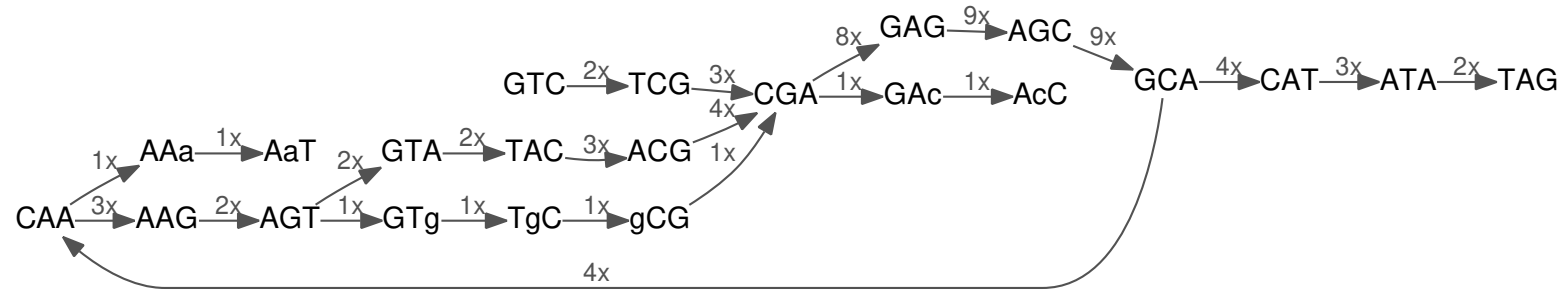
- jediný chromozóm a žiadne “nejednoznačné”  $k$ -tice  
⇒ zostavenie = **Eulerovská cesta**  
(cesta v grafe, ktorá použije každú hranu práve raz)
- Eulerovskú cestu možno nájsť v čase  $O(m + n)$
- v realistickom prípade:  
zostavenie genómu zodpovedá niekoľkým  
**pochôdkam v de Bruijnovom grafe** (nazývame **kontigy**),  
ktoré dohromady pokrývajú veľkú časť hrán

## Príklad: sada čítaní a zodpovedajúci deBruijnov graf

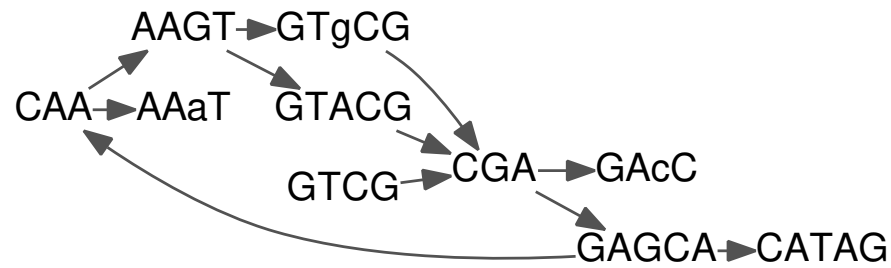
GTCGAGCAAGTACGAGCATAG  
 TCGAGCA AGCATAG  
 AGCAAaT AGCATAG  
 GTCGAcC GTACGAG  
 GTCGAGC TACGAGC  
 CGAGCAA ACGAGCA  
 AGTgCGA  
 CAAGTAC  
 GCAAGTA GAGCAT  
 GAGCAAG GAGCATA  
 TACGAGC



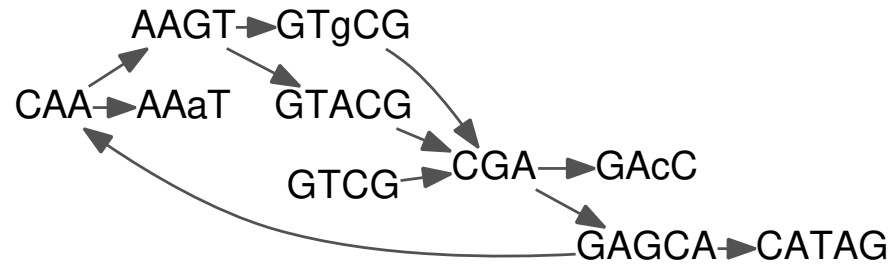
## Príklad: zjednodušovanie de Bruijnovho grafu



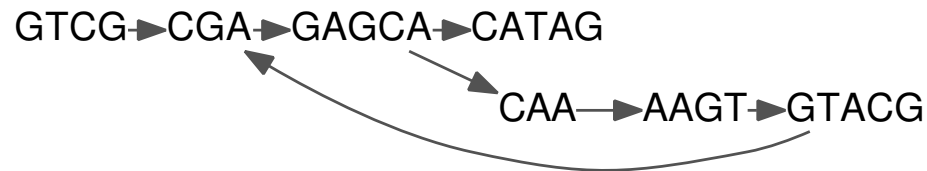
Spojíme jednoznačné cesty do vrcholov



## Príklad: odstraňovanie chýb z de Bruijnovho grafu



Odstránenie chýb (výbežkov a bublín s nízkym pokrytím)



Spájaním dostaneme 4 kontigy (pôv. GTCGAGCAAAGTACGAGCCATAG)





## Typické výsledky zostavovania genómov

- Veľa **kratších kontigov**, ktoré možno ďalej kombinovať do väčších celkov (**scaffolds**) pomocou ďalšej informácie (napr. spárované čítania, čítania 3. generácie)
- Niektoré časti nemožno jednoznačne zostaviť z dôvodu **dlhých opakujúcich sa sekvencií**

**Príklad:** človek chr14, 88 Mbp, 70× pokrytie

Metóda	Počet kontigov	Chýb	N50 po korekcii
Velvet (základný de Bruijn)	>45000	4910	2.1 kbp
Velvet (scaffolding)	3565	9156	27 kbp
AllPaths-LG	225	45	4.7 Mbp

N50: kontigy s touto alebo dlhšou dĺžkou pokrývajú 50% genómu

korekcia: rozsekne všetky zle spojené kontigy

## História sekvenovania genómov

- 1976 MS2 (RNA vírus) 40 kB
- 1988 projekt sekvenovania ľudského genómu (15 rokov)
- 1995 baktéria H. influenzae 2 MB, shotgun (TIGR)
- 1996 S. cerevisiae 10 MB, BAC-by-BAC (Belgicko, Británia)
- 1998 C. elegans 100 MB, BAC-by-BAC (Wellcome Trust)
- 1998 Celera: ľudský genóm do troch rokov!
- 2000 D. melanogaster 180 MB, shotgun (Celera, Berkeley)
- 2001 2x ľudský genóm 3 GB (NIH, Celera)
- po 2001 Myš, potkan, kura, šimpanz, pes, makak,...
- 2007 Watsonov a Venterov genóm (454)
- po 2012 1000 ľudských genómov
- čoskoro 10k genómov stavovcov, sekvenovanie ako diagnostický nástroj

## Zhrnutie

- Sekvenovanie genómu je zložitý proces, v ktorom hrá bioinformatika dôležitú úlohu
- V súčasnosti nízka cena, krátke čítania nastupuje 3. generácia s extra dlhými ale chybovými čítaniami
- Problém zostavovania genómu, najkratšie spoločné nadslovo
- Praktické riešenie pre krátke čítania: de Bruijnove grafy
- V zostavenej sekvencii môžu byť chyby, medzery, viaceré kontigy
- Na budúce: ako sa vysporiadať s dlhými čítaniami 3. generácie?
- Pokrytie genómu a veľkosť čítania hrajú najdôležitejšiu úlohu pri tom, ako fragmentovaný bude výsledok:
  - pre Sanger: 7-10× pokrytie
  - pre NGS: 40-70× pokrytie
  - pre 3. generáciu: 30× pokrytie

## Použitie NGS: Populačná genetika

- Sekvenujeme krátke čítania z genómu určitého človeka
- Ako sa môj vlastný genóm líši od genómu “priemerného” človeka?
- Ako jednoduché genetické rozdiely ovplyvňujú fenotyp?
- Personalizovaná medicína
- Populačná štruktúra, história ľudstva
- Etické otázky

## Problémy:

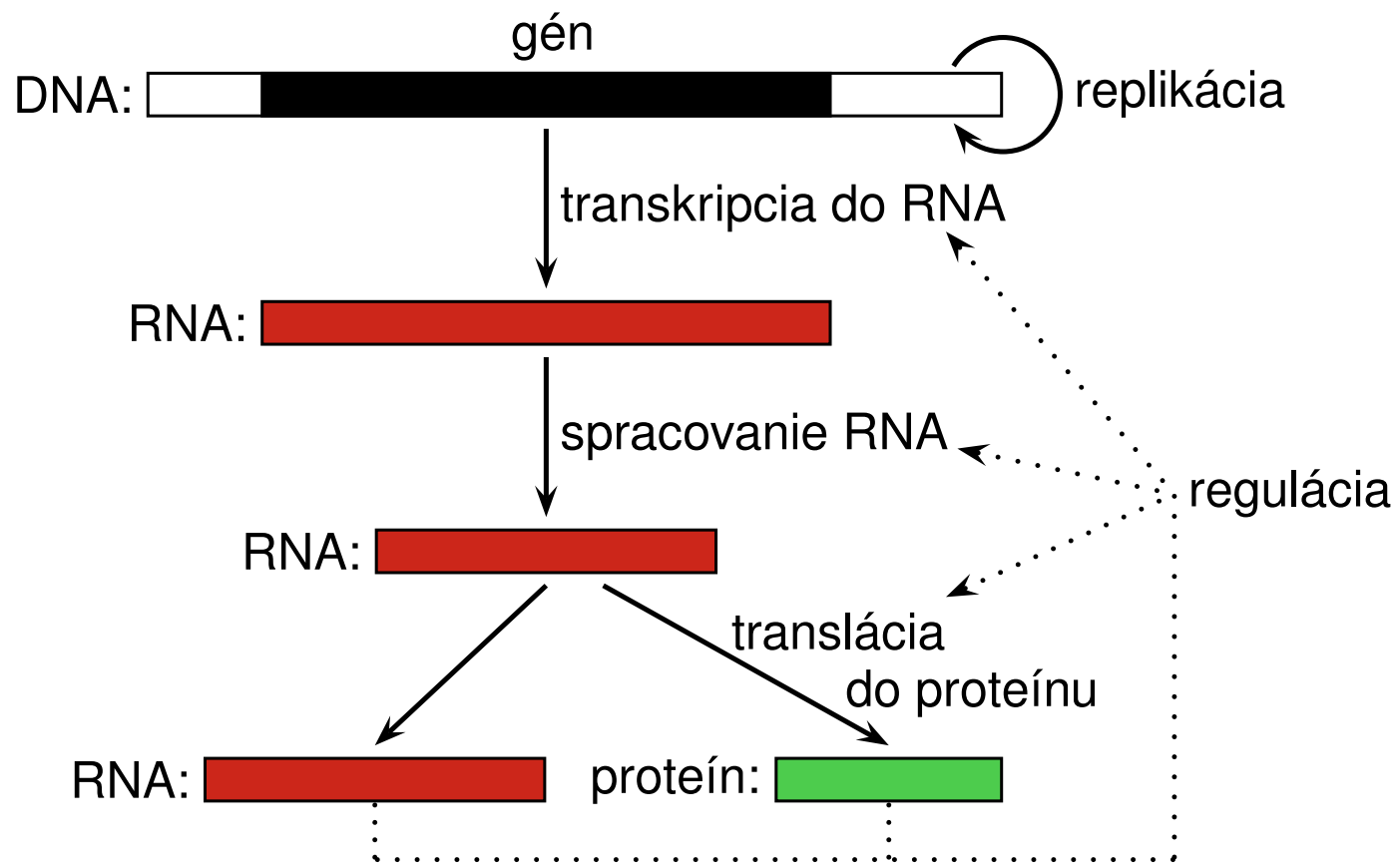
- Mapovanie krátkych čítaní na referenčnú sekvenciu
- Identifikácia rozdielov (malých a väčších)

## Použitie NGS: Environmentálne sekvenovanie – Metagenomika

- Aké mikroorganizmy žijú v našich telách?  
črevná a žalúdočná flóra, ústna dutina, koža, ...
- Diverzita mikroorganizmov v rôznych ekosystémoch
- Ťažké izolovať jednotlivé organizmy
- Sekvenujeme zmes čítania z rôznych genómov
- Snažíme sa zostaviť aspoň krátke kontigy

### Problémy:

- Oddelenie čítaní/kontigov patriacich do rôznych genómov



## Použitie NGS: Hľadanie génov, väzobných miest,...

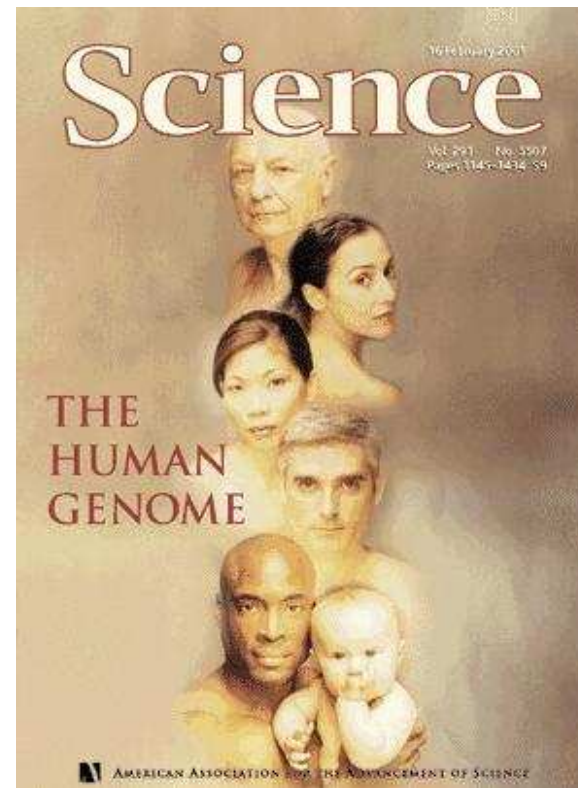
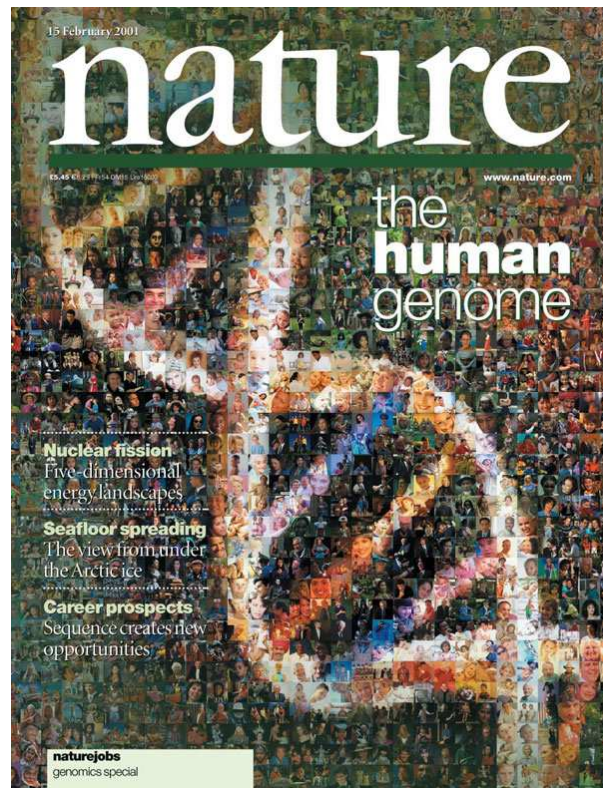
- Sekvenovať môžeme aj RNA, dostávame gény v genóme
- Chip-Seq: vyfiltrujeme kúsky DNA, na ktoré je naviazaný určitý proteín, sekvenujeme, mapujeme na genóm

### Problémy:

- Identifikácia miest zotrihu
- Identifikácia väzobných miest podľa hĺbky pokrytia

# Sekvenovanie a zostavovanie genómov (časť 2 - dlhé čítania)

Tomáš Vinar  
4.10.2018





## Prehľad sekvenovacích technológií

Technológia	Dĺžka čítania	Chybovosť	Za deň	Cena za MB
<b>1. generácia</b>				
Sanger	do 1000 bp	< 2%	3 MB	\$4000
<b>2. (next) generácia (cca 2004)</b>				
Illumina	2× 150bp	< 1%	30 GB	< \$1
<b>3. generácia (práve nabieha)</b>				
PacBio	6-25 Kbp	10%	1 GB	\$2
Oxford Nanopore	5-100+ kbp	10%	1 GB	\$1

## Na minulej prednáške

- Genóm je potrebné zostaviť zo sekvenačných čítaní
- Zostavovanie genómov pomocou de Bruijnových grafov
- Nie je vhodné pre najnovšie technológie s dlhými a chybovými čítaniami
  - Rozklad na  $k$ -mery zahadzuje príliš veľa informácie (dĺžka čítania 10000,  $k$  obvykle medzi 30 a 70)
  - Chybovosť okolo 10% robí de Bruijnov graf neprehľadným (pre  $k = 31$ , **každý**  $k$ -mer v priemere 3 chyby)

## Prístup Overlap–Layout–Consensus

- **Overlap:** Nájdi prekryvy medzi čítaniami a zostav tzv. **graf prekryvov**
- **Layout:** Zjednoduš graf prekryvov a nájdi v ňom cesty, ktoré budú zodpovedať **kontigom**
- **Consensus:** Ku každému kontigu zostav sekvenciu, ktorá je konsenzom sekvencií čítaní, ktoré kontig tvoria (opravovanie lokálnych chýb)

## Overlap: hľadanie prekryvov

CATCTCTAGGCCAGC

|||||| |

TAGGCCTGCTTCTTG

- špeciálny prípad zarovnávanie sekvencií (nasledujúca prednáška)
- prekryvy **budú obsahovať chyby**  
(v našom prípade cca 1 chyba na 10 báz prekryvu)
- **čítaní je veľa:**  $30\times$  pokrytie ľudského genómu  
 $\Rightarrow$  cca 9 mil. čítaní dĺžky 10000  
**nemôžeme porovnávať každé čítanie s každým**
- praktický prístup:
  - rýchle predfiltrovanie **vhodných kandidátskych párov čítaní**  
(napríklad musia obsahovať dosť dlhý spoločný  $k$ -mer)
  - pomalšie zarovnávanie len pre kandidátske páry

## Zostavenie grafu prekryvov

- Výsledok predchádzajúcej fázy:  
CATCTCTAGGCCAGC / TAGGCCTGCTTCTTG, prekryv 9 báz  
...
- Zostavíme **graf prekryvov**:  
vrcholy: čítania      ohodnotené hrany: prekryvy s dĺžkami

Príklad:

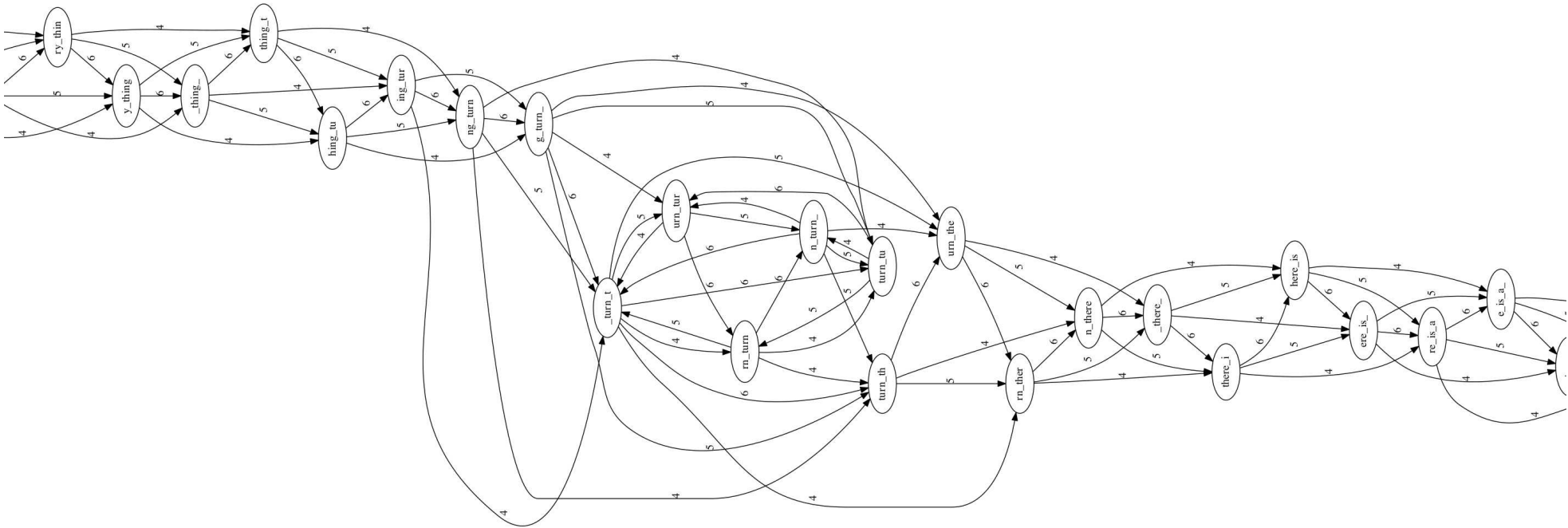
`to_every_thing_turn_turn_turn_there_is_a_season`

čítania dĺžky 7, minimálny prekryv 4

Príklad:

to\_every\_thing\_turn\_turn\_turn\_there\_is\_a\_season

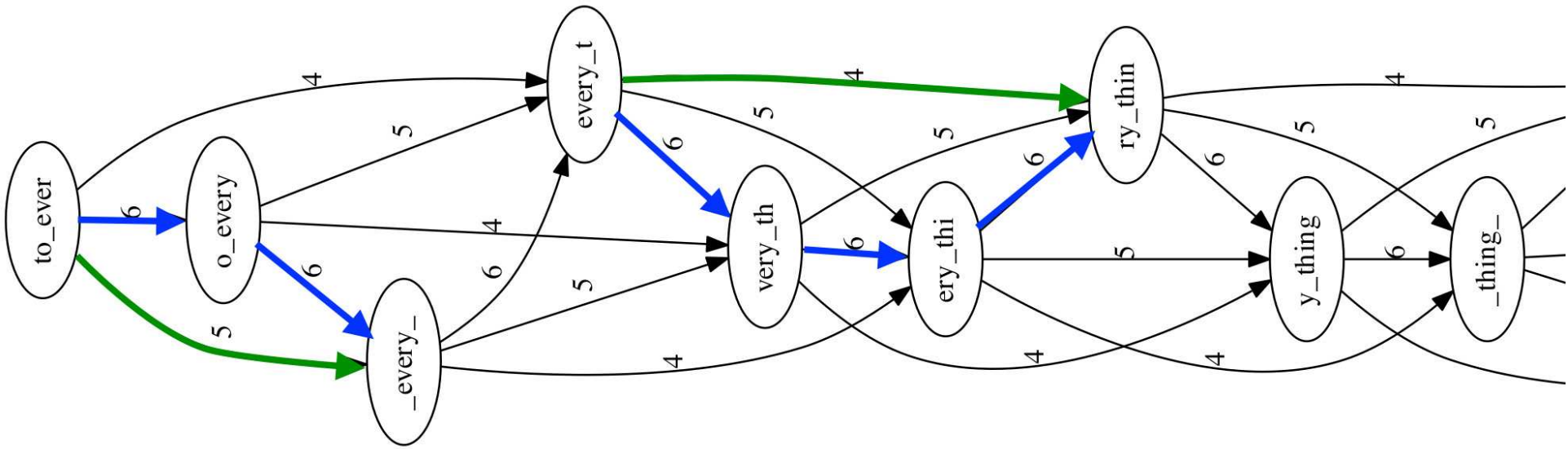
čítania dĺžky 7, minimálny prekryv 4



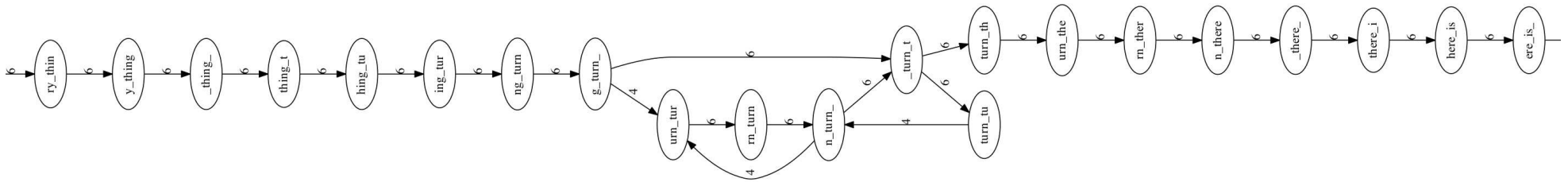
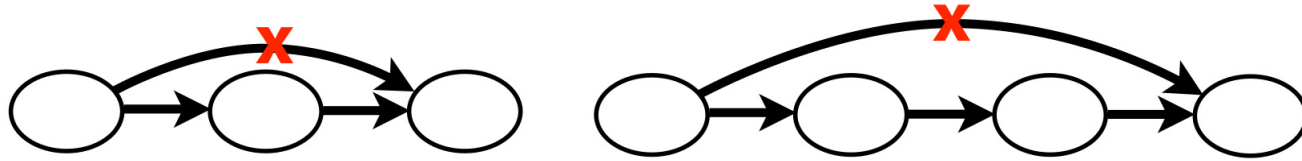
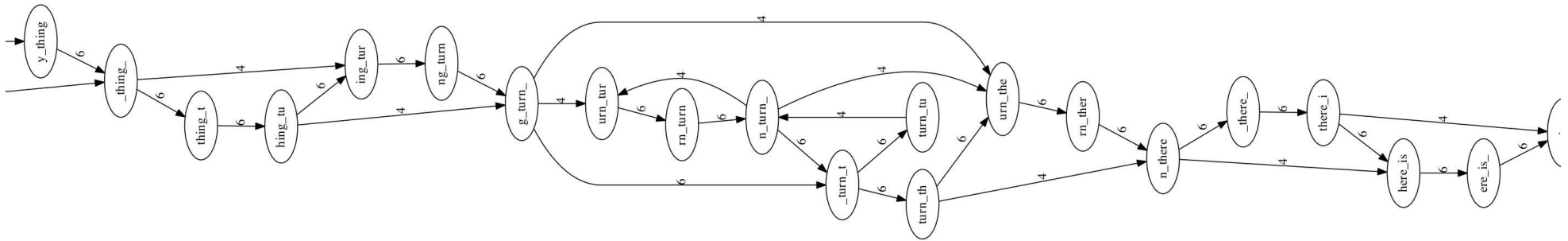
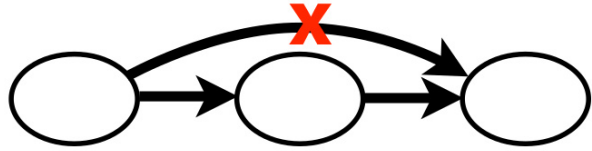
Príklad a obrázky Ben Langmead

## Layout: Tranzitívne hrany

- Niektoré hrany sú nadbytočné, lebo hovoria to isté ako cesty z iných hrán



# Layout: Odstránenie tranzitívnych hrán



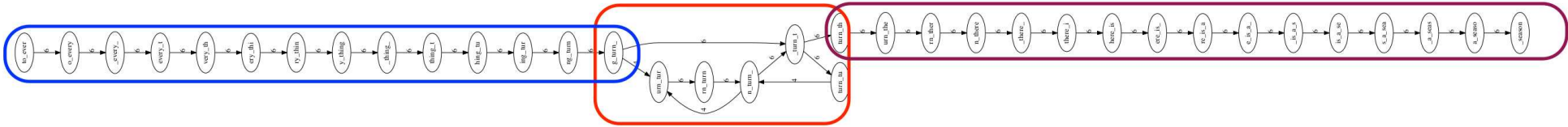


## Layout: Rozdelenie na kontigy

### Pôvodná sekvencia:

to\_every\_thing\_turn\_turn\_turn\_there\_is\_a\_season

### Nerozvetvujúce sa cesty reprezentujú kontigy



### Výsledok:

Contig 1

to\_every\_thing\_turn\_

Contig 2

turn\_there\_is\_a\_season



Unresolvable repeat

## Consensus: Získanie finálnej sekvencie

TAGATTACACAGATTACTGA TTGATGGCGTAA CTA  
TAGATTACACAGATTACTGACTTTGATGGCGTAAACTA  
TAG TTACACAGATTATTGACTTCATGGCGTAA CTA  
TAGATTACACAGATTACTGACTTTGATGGCGTAA CTA  
TAGATTACACAGATTACTGACTTTGATGGCGTAA CTA

↓ ↓ ↓ ↓ ↓  
TAGATTACACAGATTACTGACTTTGATGGCGTAA CTA

Take reads that make up a contig and line them up

Take *consensus*, i.e. majority vote

## Ako sa líši de Bruijnov graf od grafu prekryvov?

### de Bruijnov graf

- fixná dĺžka prekryvov
- zahadzujeme informáciu o kontinuite presahujúcej  $k$  znakov
- cesty reprezentujú genóm
- chyby  $\Rightarrow$  bubliny a výbežky
- riešia sa v predspracovaní
- kontigy pokrývajú takmer všetky hrany

### Graf prekryvov

- variabilná dĺžka prekryvov
- maximálne využitie informácie o prekryvoch
- cesty reprezentujú genóm
- chyby sú zväčša “schované”
- riešia sa dodatočne (consensus)
- treba odstraňovať tranzitívne hrany

## Príklad: Skladanie genómu *Magnusiomyces capitatus*

(dĺžka genómu 19.6 Mbp, 4 chromozómy + mtDNA)

Technológia	Pokrytie	# kontigov	najväčší	avg	N50
Illumina / Spades	250x	1102	172.6 Kbp	17.6 Kbp	62.0 Kbp
PacBio / Canu	37x	17	4.7 Mbp	1.2 Mbp	1.7 Mbp
PacBio + MinION	65x	11	4.4 Mbp	1.8 Mbp	2.0 Mbp

## Zhrnutie

- Dlhé čítania nám umožňujú poskladať genómy do podstatne menej fragmentovanej podoby ako krátke čítania
- Na hľadanie prekryvov medzi čítaniami sú potrebné rýchle algoritmy (niektoré si ukážeme o dve prednášky)
- Grafy prekryvov a de Bruijnové grafy sa podobajú, existujú snahy o zjednotenie týchto dvoch konceptov

## Použitie NGS: Populačná genetika

- Sekvenujeme krátke čítania z genómu určitého človeka
- Ako sa môj vlastný genóm líši od genómu “priemerného” človeka?
- Ako jednoduché genetické rozdiely ovplyvňujú fenotyp?
- Personalizovaná medicína
- Populačná štruktúra, história ľudstva
- Etické otázky

## Problémy:

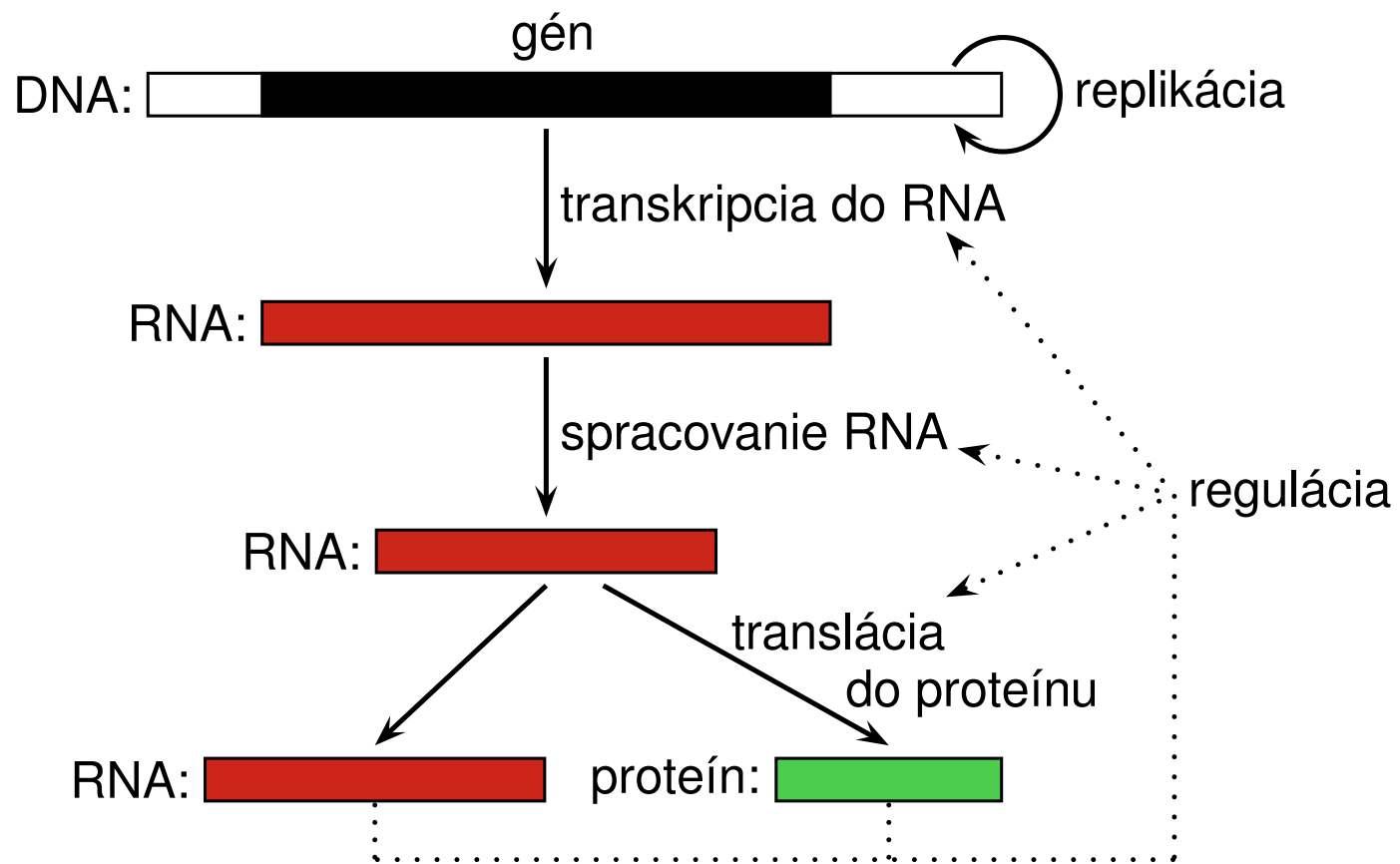
- Mapovanie krátkych čítaní na referenčnú sekvenciu
- Identifikácia rozdielov (malých a väčších)

## Použitie NGS: Environmentálne sekvenovanie – Metagenomika

- Aké mikroorganizmy žijú v našich telách?  
črevná a žalúdočná flóra, ústna dutina, koža, ...
- Diverzita mikroorganizmov v rôznych ekosystémoch
- Ťažké izolovať jednotlivé organizmy
- Sekvenujeme zmes čítania z rôznych genómov
- Snažíme sa zostaviť aspoň krátke kontigy

### Problémy:

- Oddelenie čítaní/kontigov patriacich do rôznych genómov





## Použitie NGS: Hľadanie génov, väzobných miest,...

- Sekvenovať môžeme aj RNA, dostávame gény v genóme
- Chip-Seq: vyfiltrujeme kúsky DNA, na ktoré je naviazaný určitý proteín, sekvenujeme, mapujeme na genóm

### Problémy:

- Identifikácia miest zotrihu
- Identifikácia väzobných miest podľa hĺbky pokrytia

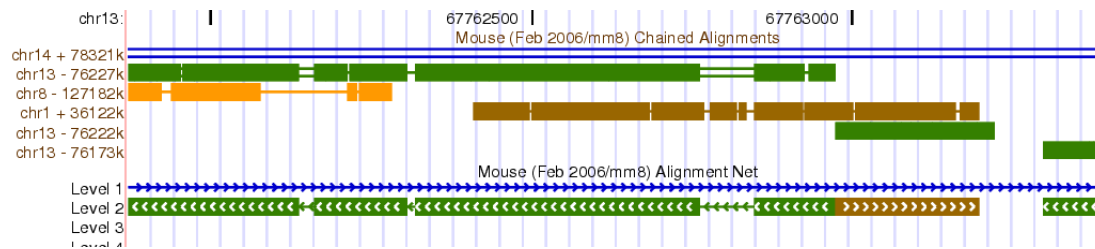
## Oznamy

- Výber článku na journal club formulárom na stránke do utorka 23.10.
- Prvá domáca úloha: zadanie na budúci týždeň, čas na vypracovanie cca 2 týždne (detaily nabudúce)
- Ďalšie info <http://compbio.fmph.uniba.sk/vyuka/mbi/>

# Zarovnávane sekvencií (sequence alignment) 1/2

Tomás Vinar

11.10.2018



[Durbin et al., 1998, kapitola 2]

## Problém: Lokálne zarovnávanie (local alignment)

ggcccttggagttgactgtcctgctgctccttgagg  
ccattctcagagagaggaagtggcctcattttaatc  
cgcttcccacagccttgtcctttccagacccatggg  
agagggaggggctgaggggtgtggctgagcccacca  
agtcacgcgctcactctgcaggtccctctccccaag  
gccgtggccttgggagcccgtggatcccagtgagtg  
acgcctccacccccgcccactcgggcagtttaac  
ccttgttgttcacttgcagacatcgtgaacacggcc  
cggcccgacgagaaggccataatgacctatgtgtcc  
agcttctaccatgccttttcaggagcgcagaaggta  
ccgagcagggccaggcaggccctcctcgccgccacc  
gcgcaatgccgcccgtgcctctcgctcccgtgctc  
acctcatttctcttgcagacggcagtggcctctctc  
caactggaagccacccccagctccct...

tgatgccgaggatgtgttcgctcgagcatccggacga  
gaagtcacacctacgtggtcacactactatcacta  
cttagcaaaactcaagcaggagacgggtgcagggcat  
aagcgtatcggtaaggtggctcggcattgccatggag  
aacgacaaaatgggtccacgactacgagaacttcaca  
agcgatctgctcaagtggatcgaaacgacccatccag  
tcgctgggagcgggagttcgaaaactcgctggcc  
ggcgtccaagggcagttggcccagttctccaactac  
cgcacccatcgagaagccgcccagtttgtggaaaag  
ggcaacctcgaggtgctccttttcaccctgcagtc  
aagatgcgggccaacaaccagaagccctacacacc  
aaagagggcaagatgatttcggacatcaacaaggcc  
tgggagcgtctggagaaggccgagcacgaacgcgaa  
ttggccctgcgcgaggagctcatccg...

**Vstup:** dve sekvencie

## Problém: Lokálne zarovnávanie (local alignment)

ggccttggagttgactgtcctgctgctccttgagg  
ccattctcagagagaggaagtggcctcattttaatc  
cgcttcccacagccttgtcctttccagacccatggg  
agagggaggggctgagggtgtggctgagcccacca  
agtcacgcgtcactctgcaggtccctctccccaag  
gccgtggccttgggagcccgtggatcccagtgagtg  
acgcctccacccccgcctactcgggcagtttaac  
ccttgttgttacttgcagacatcgtgaacacggcc  
cggcccgacgagaaggccataatgacctatgtgtcc  
agcttctaccatgccttttcaggagcgcagaaggta  
ccgagcagggccaggcaggccctcctcgccgccacc  
gcgcaatgccgccgctgcctctgcctcccgtgctc  
acctcatttctcttgcagacggcagtggcctctctc  
caactggaagccacccccagctccct...

tgatgccgaggatgtgttcgtcgagcatcggacga  
gaagtccatcacctacgtggtcacctactatcacta  
cttagcaaaactcaagcaggagacggtgcagggcat  
aagcgtatcggtaagggtggcggcattgccatggag  
aacgacaaaatgggtccacgactacgagaacttcaca  
agcgatctgctcaagtggatcgaacgaccatccag  
tcgctgggagcagcgggagttcgaaaactcgctggcc  
ggcgtccaagggcagttggcccagttctccaactac  
cgcaccatcgagaagccgccaagtttgtggaaaag  
ggcaacctcgaggtgctccttttcacctgcagtcc  
aagatgcgggccaacaaccagaagccctacacacc  
aaagagggcaagatgatttcggacatcaacaaggcc  
tgggagcgtctggagaaggccgagcacgaacgcgaa  
ttggcctgcgcgaggagctcatccg...

**Výstup:** podobné úseky (zarovnaná, alignments).

```
CCCGACGAGAAGGCCATAATGACCTATGTGTCCAGCTTCTACCATGCCTTT
|| ||| ||| ||| ||| ||| ||| ||| ||| ||| ||| ||| |||
CCGGACGAGAAGTCCAT---CACCTACGTGGTCACCTACTATCACTACTTT
```

Vlož pomlčky (medzery, gaps) tak, aby rovnaké bázy boli pod sebou.  
Dobré zarovnanie má veľa zarovnaných rovnakých báz, málo medzier.

## Na čo sú dobré zarovnanie?

- **Orientácia v obrovských databázach.**

Genbank WGS má vyše 3 TB sekvencií.

Napr. z ktorého genómu (a odkiaľ) pochádza daná sekvencia?

- **Určovanie funkcie (napr. proteínu).**

Podobné sekvencie často majú rovnakú/podobnú funkciu.

- **Štúdium evolúcie.**

Hľadáme homológy: sekvencie, ktoré sa vyvinuli z toho istého spoločného predka.

V ideálnom prípade medzery zodpovedajú inzerciam a deléciám, zarovnané bázy zachovaným bázam a substitúciám.

- **Hľadanie génov a iných funkčných prvkov.**

Menia sa pomalšie ako ostatné sekvencie.

## Formulácia problému

**Skórovanie zarovnaní:** napr. zhoda +1, nezhoda -1, medzera -1.

```
GAGAAGGCCATAATGACCTATGTGTCCAGCT
|||||  |||  ||||  ||  ||  ||
GAGAAGTCCAT---CACCTACGTGGTCACCT
```

22 zhôd, 6 nezhôd, 3 medzery → skóre 13.

V praxi zložitejšie skórovanie. Chceme nastaviť tak, aby homológy mali vysoké skóre, náhodné zarovnanie nízke.

### Problém 1: globálne zarovnanie (global alignment)

Vstup: sekvencie  $X = x_1x_2 \dots x_n$  a  $Y = y_1y_2 \dots y_m$ .

Výstup: zarovnanie  $X$  a  $Y$  s najvyšším skóre.

### Problém 2: lokálne zarovnanie (local alignment)

Vstup: sekvencie  $X = x_1x_2 \dots x_n$  a  $Y = y_1y_2 \dots y_m$ .

Výstup: zarovnanie podreťazcov  $x_i \dots x_j$  a  $y_k \dots y_l$  s najvyšším skóre.

## Dynamické programovanie pre globálne zarovnanie (Needleman, Wunsch 1970)

**Podproblém:**  $A[i, j]$ : najvyššie skóre globálneho zarovnaní reťazcov  $x_1x_2 \dots x_i$  a  $y_1y_2 \dots y_j$ .

**Jeden z reťazcov dĺžky 0:** druhý reťazec je zarovnaný s medzerou.

$$A[0, j] = -j, \quad A[i, 0] = -i.$$

**Všeobecný prípad,  $i > 0, j > 0$ :**

ak  $x_i = y_j$  a sú zarovnané  $A[i, j] = A[i - 1, j - 1] + 1$ ,

ak  $x_i \neq y_j$  a sú zarovnané  $A[i, j] = A[i - 1, j - 1] - 1$ ,

ak  $x_i$  je zarovnané s medzerou  $A[i, j] = A[i - 1, j] - 1$ ,

ak  $y_j$  je zarovnané s medzerou  $A[i, j] = A[i, j - 1] - 1$ .



## Dynamické programovanie pre globálne zarovnanie

**Podproblém:**  $A[i, j]$ : najvyššie skóre globálneho zarovnanania reťazcov  $x_1x_2 \dots x_i$  a  $y_1y_2 \dots y_j$ .

**Všeobecný prípad,  $i > 0, j > 0$ :**

ak  $x_i$  a  $y_j$  sú zarovnané  $A[i, j] = A[i - 1, j - 1] + s(x_i, y_j)$

ak  $x_i$  je zarovnané s medzerou  $A[i, j] = A[i - 1, j] - 1$

ak  $y_j$  je zarovnané s medzerou  $A[i, j] = A[i, j - 1] - 1$

kde  $s(x, y) = 1$  ak  $x = y$  a  $s(x, y) = -1$  ak  $x \neq y$

**Rekurencia:**

$$A[i, j] = \max \begin{cases} A[i - 1, j - 1] + s(x_i, y_j), \\ A[i - 1, j] - 1, \\ A[i, j - 1] - 1 \end{cases}$$

## Príklad globálneho zarovnaní

CATGTCATA vs CAGTCCTAGA

		C	A	G	T	C	C	T	A	G	A
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
C	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
A	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
T	-3	-1	1	1	?						
G	-4										
T	-5										
C	-6										
G	-7										
T	-8										
A	-9										

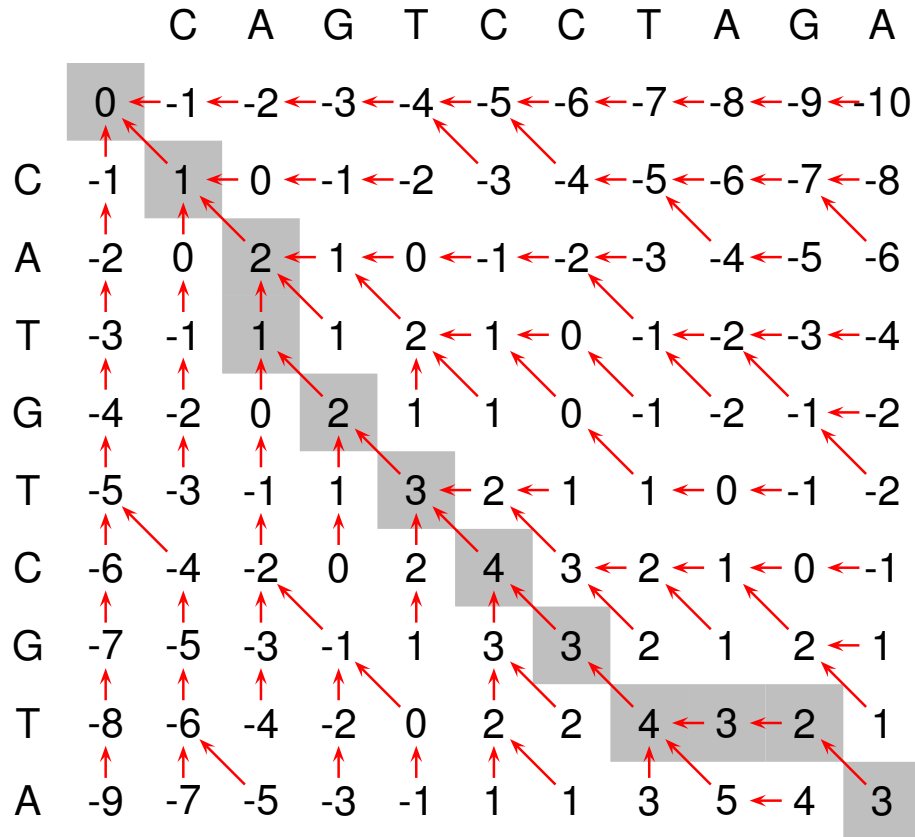
$$A[i, j] = \max \begin{cases} A[i - 1, j - 1] + s(x_i, y_j), \\ A[i - 1, j] - 1, \\ A[i, j - 1] - 1 \end{cases}$$

## Príklad globálneho zarovnania

CATGTCATA vs CAGTCCTAGA

		C	A	G	T	C	C	T	A	G	A
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
C	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
A	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
T	-3	-1	1	1	2	1	0	-1	-2	-3	-4
G	-4	-2	0	2	1	1	0	-1	-2	-1	-2
T	-5	-3	-1	1	3	2	1	1	0	-1	-2
C	-6	-4	-2	0	2	4	3	2	1	0	-1
G	-7	-5	-3	-1	1	3	3	2	1	2	1
T	-8	-6	-4	-2	0	2	2	4	3	2	1
A	-9	-7	-5	-3	-1	1	1	3	5	4	3

## Ako získať zarovnanie?



CA-GTCCTAGA

CATGTCAT--A

## Dynamické programovanie pre lokálne zarovnanie (Smith, Waterman 1981)

**Podproblém:**  $A[i, j]$ : najvyššie skóre lokálneho zarovnanie reťazcov  $x_1x_2 \dots x_i$  a  $y_1y_2 \dots y_j$ , ktoré obsahuje bázy  $x_i$  a  $y_j$ , alebo je prázdne.

**Jeden z reťazcov dĺžky 0:** prázdne zarovnanie  $A[0, j] = A[i, 0] = 0$

**Všeobecný prípad,  $i > 0, j > 0$ :**

ak  $x_i$  a  $y_j$  sú zarovnané  $A[i, j] = A[i - 1, j - 1] + s(x_i, y_j)$

ak  $x_i$  je zarovnané s medzerou  $A[i, j] = A[i - 1, j] - 1$

ak  $y_j$  je zarovnané s medzerou  $A[i, j] = A[i, j - 1] - 1$

ak  $x_i$  a  $y_j$  nie sú časťou zarovnaní s kladným skóre  $A[i, j] = 0$

**Rekurencia:**  $A[i, j] = \max \left\{ \begin{array}{l} 0, \\ A[i - 1, j - 1] + s(x_i, y_j), \\ A[i - 1, j] - 1, \\ A[i, j - 1] - 1 \end{array} \right.$

## Príklad lokálneho zarovnania

		C	A	G	T	C	C	T	A	G	A
	0	0	0	0	0	0	0	0	0	0	0
C	0	1	0	0	0	1	1	0	0	0	0
A	0	0	2	1	0	0	0	0	1	0	1
T	0	0	1	1	2	1	0	1	0	0	0
G	0	0	0	2	1	1	0	0	0	1	0
T	0	0	0	1	3	2	1	1	0	0	0
C	0	1	0	0	2	4	3	2	1	0	0
G	0	0	0	1	1	3	3	2	1	2	1
T	0	0	0	0	2	2	2	4	3	2	1
A	0	0	1	0	1	1	1	3	5	4	3

CA-GTCCTA

CATGTCATA

## Zložitejšie skórovanie

### Problémy +1, -1 skórovania:

- Je skutočne jedna nezhoda alebo medzera až taká zlá v porovnaní s jednou zhodou?
- Čo urobíme pre zarovnávanie proteínov?  
(20 prvková abeceda  $\approx$  200 parametrov)

### Úloha skórovacej schémy:

- Chceme vedieť rozlíšiť **lepšie zarovnanie** od **horších zarovnaní**:
  - Ktoré usporiadania pomlčiek dávajú väčší zmysel
- Chceme vedieť, či dané zarovnanie **má biologický význam**:
  - Ide o homológy, alebo sekvencie nesúvisia?

## Zložitejšie skórovanie: prvý pokus

Nech  $X$  a  $Y$  sú **správne zarovnané homológy**

$a$  = pravdepodobnosť, že sa dve bázy **zhodujú**

$b$  = pravdepodobnosť, že sa **nezhodujú**

$c$  = pravdepodobnosť, že báza je **zarovnaná s medzerou**

$$a + b + c = 1$$

### Pravdepodobnosť zarovnania $A$ :

```
GAGAAGGCCATAATGACCTATGTGTCCAGCT
|||||  |||  ||||  ||  ||  ||
GAGAAGTCCAT---CACCTACGTGGTCACCT
```

$$\Pr(A) = a^{22}b^6c^3$$

### Ktoré je pravdepodobnejšie?

```
CACA
|  |
CCAA
```

$$\Pr(A) = a^2b^2$$

```
CACA-
|  ||
C-CAA
```

$$\Pr(A) = a^3c^2$$



## Zložitejšie skórovanie: prvý pokus

Zlogaritmujeme: násobenie sa zmení na sčítavanie  
môžeme použiť S.-W. alebo N.-W. dyn. prog. algoritmy

$$\Pr(A) = a^{22}b^6c^3$$

$$\log \Pr(A) = 22 \log a + 6 \log b + 3 \log c$$

**Skóre:** Zhoda:  $\log a$     Nezhoda:  $\log b$     Medzera:  $\log c$

### Nevýhody takejto schémy:

- Vždy záporné skóre  $\Rightarrow$  čo s lokálnymi zarovnaniami?
- Neužitočné pre porovnávanie rôznych párov sekvencií

## Zložitejšie skórovanie: dva pravdepodobnostné modely

(Pre jednoduchosť teraz neuvažujme medzery)

**Model H:** Sekvencie  $X$  a  $Y$  sú **správne zarovnané homológy**

$$\Pr(X, Y | H) = \prod_{i=1}^n p(x_i, y_i)$$

$p(x_i, y_i)$  : pravdepodobnosť, že vidíme zarovnané práve bázy  $x_i$  a  $y_i$

**Model R:** Sekvencie  $X$  a  $Y$  nijako spolu nesúvisia

$$\Pr(X, Y | R) = \left(\prod_{i=1}^n p(x_i)\right) \left(\prod_{i=1}^n p(y_i)\right)$$

$p(x_i)$  : pravdepodobnosť výskytu bázy  $x_i$

**Porovnanie modelov H a R:** “log likelihood”

$$\log \frac{\Pr(X, Y | H)}{\Pr(X, Y | R)}$$

## Zložitejšie skórovanie: dva pravdepodobnostné modely

Porovnanie modelov  $H$  a  $R$ : “log likelihood”

$$\log \frac{\Pr(X, Y | H)}{\Pr(X, Y | R)}$$

- Dve sekvencie sú **homológy**
  - ⇒ pomer pravdepodobností je oveľa väčší ako 1
  - ⇒ **veľmi kladné skóre**
- Dve sekvencie **nesúvisia**
  - ⇒ pomer pravdepodobností je oveľa menší ako 1
  - ⇒ **veľmi zaporné skóre**

$$\log \frac{\Pr(X, Y | H)}{\Pr(X, Y | R)} = \log \frac{\prod_{i=1}^n p(x_i, y_i)}{(\prod_{i=1}^n p(x_i)) (\prod_{i=1}^n p(y_i))} = \sum_{i=1}^n \log \frac{p(x_i, y_i)}{p(x_i)p(y_i)}$$

## BLOSUM62 skórovacia matica pre proteíny

BLOcks of aminoacid **S**Ubstitution **M**atrix; Henikoff, Henikoff 1992

	A	R	N	D	C	Q	E	G	H	I	L	...
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	
N	-2	0	6	1	-3	0	0	0	1	-3	-3	
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	
...												

- Vyber **biologicky relevantné zarovnanie** proteínov (BLOCKS)
- Páry s nanajvyš 62% identitou
- $p(x, y)$  : ako často vidíme aminokyseliny  $x$  a  $y$  zarovnané
- $p(x)$  : ako často sa vyskytuje aminokyselina  $x$

- **skóre pre dvojicu aminokyselín  $x$  a  $y$** :  $\log \frac{p(x, y)}{p(x)p(y)}$
- prenásobíme konštantou a zaokrúhlime:
  - aby sme neurobili príliš veľkú chybu
  - aby sa s číslami lepšie počítalo

## Zložitejšie skórovanie: afínne skóre medzier

```
CCCGACGAGAAGGCCATAATGACCTATGTGTCCAGCTTCTACCATGCCTTT
|| ||| ||| ||| ||| ||| ||| ||| ||| ||| ||| ||| |||
CCGGACGAGAAGTCCAT---CACCTACGTGGTCACCTACTATCACTACTTT
```

Niekoľko medzier za sebou asi nevzniklo nezávisle, možno jedna mutácia.

Penalta za začatie medzery (gap opening cost)  $o$ ,

Penalta za rozšírenie medzery o jedna (gap extension cost)  $e$ .

Medzera dĺžky  $g$  má penaltu  $o + e(g - 1)$ .

Zvolíme  $o < e$  (t.j.  $|o| > |e|$ ).

Základné nastavenia blastn: zhoda +2, nezhoda -3,  $o = -5$ ,  $e = -2$ .

Príklad vyššie: 22 zhôd, 6 nezhôd, 1 medzera dĺžky 3

→ skóre  $2 \cdot 22 - 3 \cdot 6 - 5 - 2 \cdot 2 = 16$ .

## Zhrnutie

- Globálne a lokálne zarovania
- Needleman-Wunschov a Smith-Watermanov algoritmus
- Skórovanie zarovnaní pomocou porovnávaní modelov
- Proteínové BLOSUM matice
- Afínne skórovanie medzier

## Problémy na zamyslenie

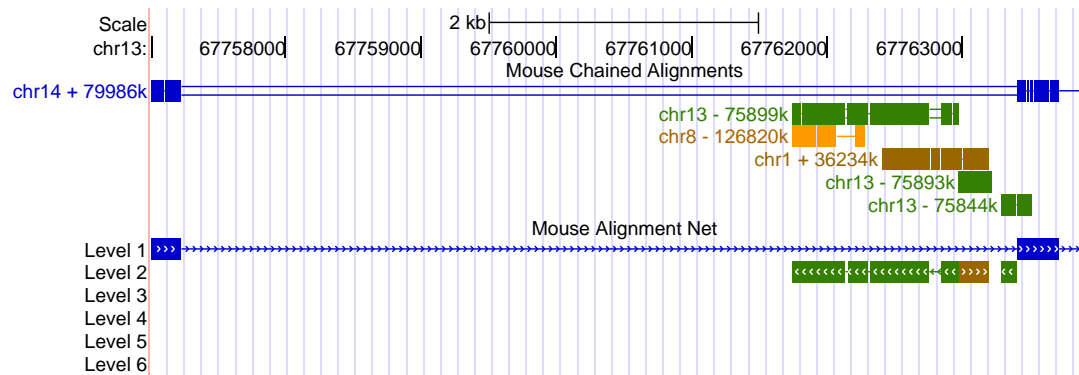
1. **Časová zložitosť Smith-Waterman:**  $O(nm)$   
 $n$  - veľkosť prvej sekvencie  
 $m$  - veľkosť druhej sekvencie  
**Čo robiť ak chceme porovnať ľudský genóm s myšacím genómom?**
2. Povedzme, že nájdeme zarovnanie so skóre 14  
**Je toto skóre dobré, alebo ide o niečo, čo vidíme náhodou?**

## Oznamy

- Výber článku na journal club formulárom na stránke do budúceho utorka 23.10. Zloženie skupín oznámime na budúcej prednáške.
- Domáca úloha 1 je zverejnená na stránke, odovzdávajúce do štvrtka 8.11.
- Na domácich úlohách neopisujte. Môžete sa rozprávať, ale nerobte si pritom poznámky, neukazujte si navzájom svoje riešenia. Každý by mal napísať riešenie samostatne.
- Otázky k zadaniam a všeobecnú diskusiu k predmetu píšete do Facebookovej skupiny (môžete spolužiakom aj odpovedať), otázky k vašim riešeniam posielajte vyučujúcim e-mailom.

# Zarovňavanie sekvencií 2/2 (sequence alignment)

Tomáš Vinar̄  
18.10.2018





## Zhrnutie z minulej prednášky

- **Problém globálneho a lokálneho zarovnania**

**Vstup:** sekvencie  $X = x_1x_2 \dots x_n$  a  $Y = y_1y_2 \dots y_m$ .

**Výstup:** zarovnanie  $X$  a  $Y$  s najvyšším skóre

resp. zarovnania podreťazcov  $x_i \dots x_j$  a  $y_k \dots y_\ell$  s najvyšším skóre.

- **Správny algoritmus na riešenie**

dynamické programovanie

- **Realistické skórovacie schémy**

## Máme správny algoritmus na zarovnávanie, čo viac nám chýba?

**Časová zložitosť:**  $O(nm)$  na sekvenciách dĺžky  $n$  a  $m$ .

### Koľko je to času v skutočnosti?

(jednoduchá implementácia, náhodné sekvencie dĺžky  $n$ , bežný počítač)

$n$	čas výpočtu
100	0.0008s
1,000	0.08s
10,000	8s
100,000	13 minút (*)
1,000,000	22 hodín (*)
10,000,000	3 mesiace (*)
100,000,000	25 rokov (*)

### Potrebujeme efektívnejší algoritmus,

najmä ak chceme pracovať s celými genómami

## Heuristické lokálne zarovnávanie

- Nie je zaručené, že nájdeme najlepšie zarovnanie, ale program pobeží rýchlejšie.
- Prehľadá iba “sľubné” časti dyn. prog. matice.

**Napríklad:** BLASTN [Altschul et al., 1990],  
FASTA [Pearson and Lipman, 1988]

- Nájdí krátke zhodujúce sa úseky dĺžky  $w$  (**jadrá zarovnaní**).
- Rozšír každé jadro pozdĺž uhlopriečky na zarovnanie bez medzier.
- Spoj zarovnaní na neďalekých uhlopriečkach medzerami.
- Lokálne vylepši zarovnanie dynamickým programovaním (možno vynechať).

## Ako nájdeme zhodujúce sa úseky?

- Vybudujeme “slovník” úsekov dĺžky  $w$  z prvej sekvencie.
- Nájdeme každý úsek z druhej sekvencie v slovníku.

**Príklad:** CAGTCCTAGA vs CATGTCATA

### Slovník:

AG 2, 8

CA 1

CC 5

CT 6

GA 9

GT 3

TA 7

TC 4

### Hľadaj:

CA → 1

AT → -

TG → -

GT → 3

TC → 4

CA → 1

AT → -

TA → 7

## Heuristické lokálne zarovnávanie

**Príklad:** začíname z jadier dĺžky  $w = 2$   
(V praxi sa používa  $w = 10$  a viac.)

		C	A	G	T	C	C	T	A	G	A
	0	0	0	0	0	0	0	0	0	0	0
C	0	1	0	0	0	1	1	0	0	0	0
A	0	0	2	1	0	0	0	0	1	0	0
T	0	0	0	1	2	1	0	1	0	0	0
G	0	0	0	0	1	0	0	0	0	1	0
T	0	0	0	0	2	2	1	1	0	0	0
C	0	1	0	0	0	4	3	0	0	0	0
A	0	0	2	1	0	3	3	2	1	0	1
T	0	0	1	1	2	2	2	4	3	2	1
A	0	0	1	0	1	1	1	3	5	4	3

1. nájdi zhodné úseky
2. rozšír bez medzier
3. spoj medzerami

## Rýchlosť heuristického algoritmu

### Algoritmus:

- Nájdi jadrá zarovnaní (krátke zhodujúce sa úseky dĺžky  $w$ ).
- **Drahý krok:** Rozširovanie/spájanie jadier do väčších zarovnaní.

**Náhodné zhody dĺžky  $w$ :** nie sú častou zarovnaní s vysokým skóre. Vyfiltrujeme ich pri rozširovaní, ale spomaľujú program.

### Koľko náhodných zhôd?

Dva nukleotidy sa zhodujú s pravdepodobnosťou  $1/4$ .

$w$  zhôd za sebou s pravdepodobnosťou  $4^{-w}$ .

Stredná hodnota počtu zhôd  $nm4^{-w}$ .

Zvýšenie  $w$  o 1 zníži počet zhôd cca 4 krát.

## Senzitivita heuristického algoritmu

### Algoritmus:

- Nájdi jadrá zarovnaní (krátke zhodujúce sa úseky dĺžky  $w$ ).
- **Drahý krok:** Rozširovanie/spájanie jadier do väčších zarovnaní.

**Nenájdené zarovnanie:** vysoké skóre, ale **nemajú jadro dĺžky  $w$**

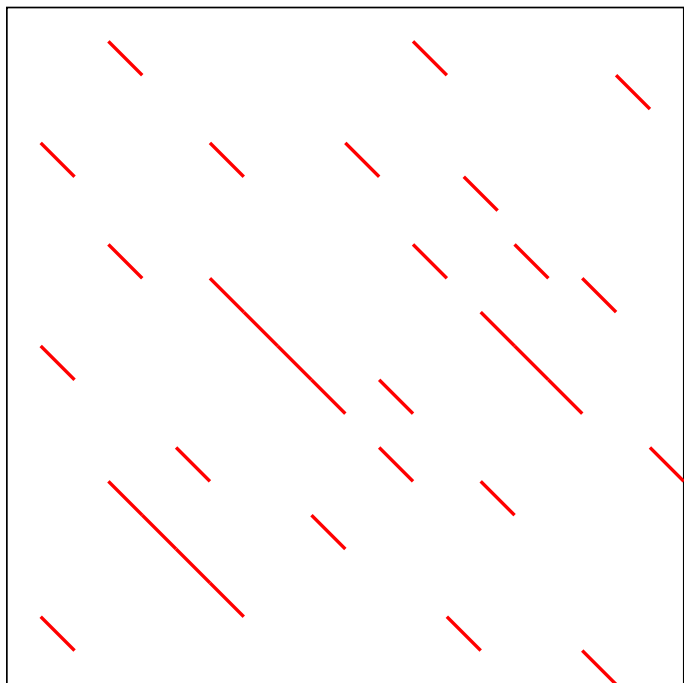
**Príklad:** CA-GTCCTA                      nenájdeme pre  $w \geq 4$   
                  CATGTCATA

**Senzitivita:** aká časť **skutočných zarovnaní** obsahuje zhodu dĺžky  $w$

## Rýchlosť vs. senzitivita

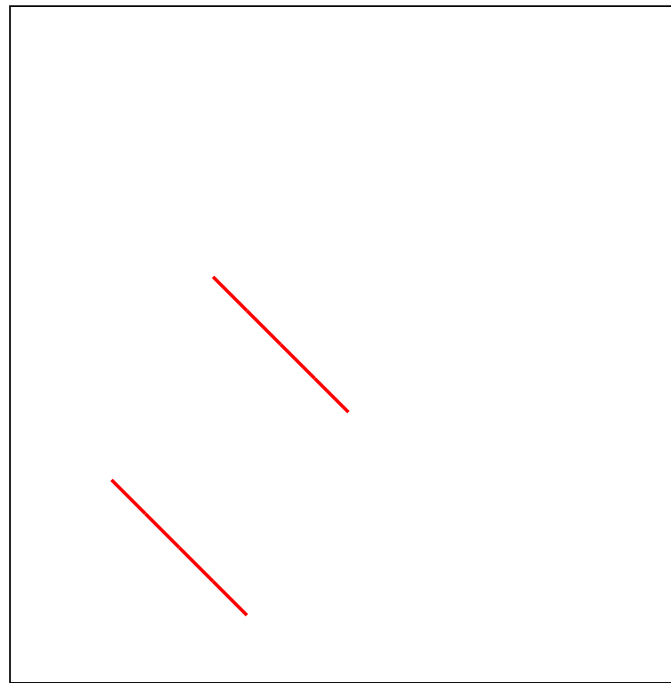
**Malé**  $w$

veľa náhodných zhôd, pomalé



**Veľké**  $w$

nenájdeme veľa zarovnaní





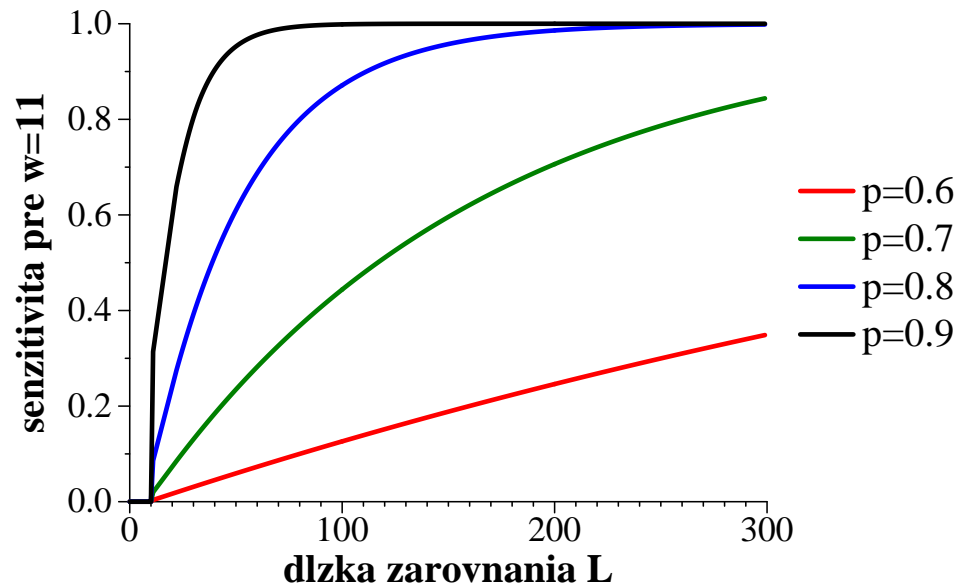
## Senzitivita heuristického algoritmu

### Odhad senzitivity:

Predpokladáme zarovnanie bez medzier, dĺžky  $L$

Každá pozícia je zhoda s pravdepodobnosťou  $p$

$f(L, p) = \Pr(\text{zarovnanie obsahuje } w \text{ zhôd za sebou})$



(človek-myš:  $p \approx 0.7$ )

## BLAST algoritmus pre proteíny

### BLOSUM62 skórovacia matica pre proteíny

	A	R	N	D	C	Q	E	G	H	I	...
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	
R	-1	5	0	-2	-3	1	0	-2	0	-3	
N	-2	0	6	1	-3	0	0	0	1	-3	
D	-2	-2	1	6	-3	0	2	-1	-1	-3	
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	
Q	-1	1	0	0	-3	5	2	-2	0	-3	
E	-1	0	0	2	-4	2	5	-2	0	-3	
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	
H	-2	0	1	-1	-3	0	0	-2	8	-3	
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	

Proteínový BLAST namiesto zhody dĺžky  $w$  vyžaduje 3 aminokyseliny so skóre aspoň 13

Áno: N I R  
N L R  
 $6+2+5=13$

Nie: A I L  
A I L  
 $4+4+4=12$

## Príklady programov

**NCBI BLAST:** `blastn` pre DNA/RNA, `blastp` pre proteíny, `tblastx` preloží DNA do proteínu a použije `blastp`  
[Altschul et al., 1990, Altschul et al., 1997]

**UCSC Blat:** veľmi rýchle vyhľadávanie veľmi podobných sekvencií, napr. mRNA ku genómu [Kent, 2002].

- používa veľké  $w$
- vie rozdeliť mRNA na exóny

**PSI-BLAST:** [Altschul et al., 1997]

- Pre dotaz nájdeme zarovnaná cez `blastp`.
- Vidíme, ktoré pozície mutujú viac a ktoré menej.
- Nezhoda na zachovanej pozícii stojí viac.

⇒ nájde vzdialenejšie homológy.

Sequences producing significant alignments:		Score (Bits)	E Value	
<a href="#">ref XP_002345317.1 </a>	PREDICTED: similar to protein tyrosine ph...	<a href="#">28.2</a>	108	<b>UG</b>
<a href="#">ref XP_001726210.1 </a>	PREDICTED: similar to protein tyrosine ph...	<a href="#">28.2</a>	108	<b>G</b>
<a href="#">ref ZP_03264973.1 </a>	isocitrate dehydrogenase, NADP-dependent [...	<a href="#">27.4</a>	194	
<a href="#">ref XP_001225150.1 </a>	hypothetical protein CHGG_07494 [Chaetomi...	<a href="#">27.4</a>	194	<b>G</b>
<a href="#">ref YP_002967336.1 </a>	hypothetical protein MexAM1_META2p1254 [M...	<a href="#">26.9</a>	261	<b>G</b>
<a href="#">ref ZP_03013307.1 </a>	hypothetical protein BACINT_00864 [Bactero...	<a href="#">26.9</a>	261	
<a href="#">ref YP_001834672.1 </a>	phospholipid/glycerol acyltransferase [Be...	<a href="#">26.9</a>	261	<b>G</b>
<a href="#">ref ZP_04426281.1 </a>	NADH dehydrogenase subunit L [Planctomyces...	<a href="#">26.1</a>	469	
<a href="#">ref YP_003129642.1 </a>	putative exonuclease RecJ [Halorhabdus ut...	<a href="#">26.1</a>	469	<b>G</b>
<a href="#">ref ZP_02926313.1 </a>	multidrug efflux pump, AcrB/AcrD/AcrF fami...	<a href="#">26.1</a>	469	
<a href="#">ref ZP_02044690.1 </a>	hypothetical protein ACTODO_01565 [Actinom...	<a href="#">26.1</a>	469	
<a href="#">ref XP_001153320.1 </a>	PREDICTED: similar to tyrosine phosphatas...	<a href="#">26.1</a>	469	<b>G</b>
<a href="#">ref YP_001958968.1 </a>	inner-membrane translocator [Chlorobium p...	<a href="#">26.1</a>	469	<b>GG</b>
<a href="#">ref YP_003133865.1 </a>	hypothetical protein Svir_20200 [Saccharo...	<a href="#">25.7</a>	630	<b>G</b>

<http://blast.ncbi.nlm.nih.gov/Blast.cgi>

[Most Visited](#)
[Smart Bookmarks](#)
[Getting Started](#)
[Latest BBC Head...](#)
[Gmail](#)
[Entrez PubMed](#)

**Alignments**
[Select All](#)
[Get selected sequences](#)
[Distance tree of results](#)
[Multiple alignment](#) NEW

> [ref|XP\\_002345317.1|](#) **UG** PREDICTED: similar to protein tyrosine phosphatase 4a1 isoform 2 [Homo sapiens]  
 Length=139

[GENE ID: 730167 LOC730167](#) | similar to protein tyrosine phosphatase 4a1 [Homo sapiens]

Score = 28.2 bits (59), Expect = 108  
 Identities = 9/10 (90%), Positives = 10/10 (100%), Gaps = 0/10 (0%)

Query	1	VIVALASVEG	10
		V+VALASVEG	
Sbjct	79	VLVALASVEG	88

> [ref|XP\\_001726210.1|](#) **G** PREDICTED: similar to protein tyrosine phosphatase 4a1 isoform 1 [Homo sapiens]  
 Length=170

[GENE ID: 730167 LOC730167](#) | similar to protein tyrosine phosphatase 4a1 [Homo sapiens]

Score = 28.2 bits (59), Expect = 108  
 Identities = 9/10 (90%), Positives = 10/10 (100%), Gaps = 0/10 (0%)

Query	1	VIVALASVEG	10
		V+VALASVEG	
Sbjct	110	VLVALASVEG	119

## Ako rozlíšiť, či ide o významné zarovnanie?

Zarovnanie so skóre  $S$ .

Dĺžka dotazu  $m$ . Veľkosť databázy  $n$ .

**$P$ -hodnota:** Pravdepodobnosť, že pre náhodný dotaz dĺžky  $m$  v náhodnej databáze dĺžky  $n$  nájdeme zarovnanie so skóre aspoň  $S$ .

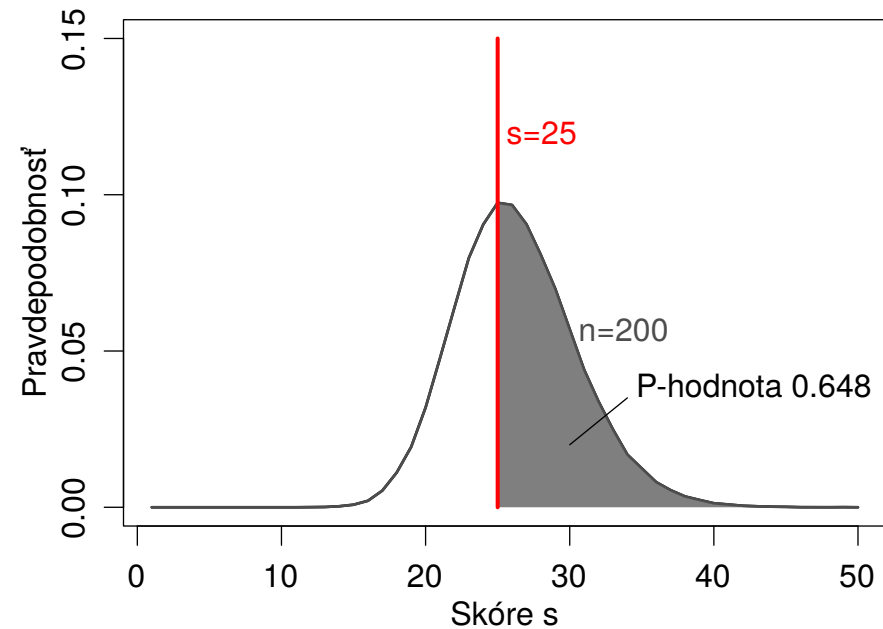
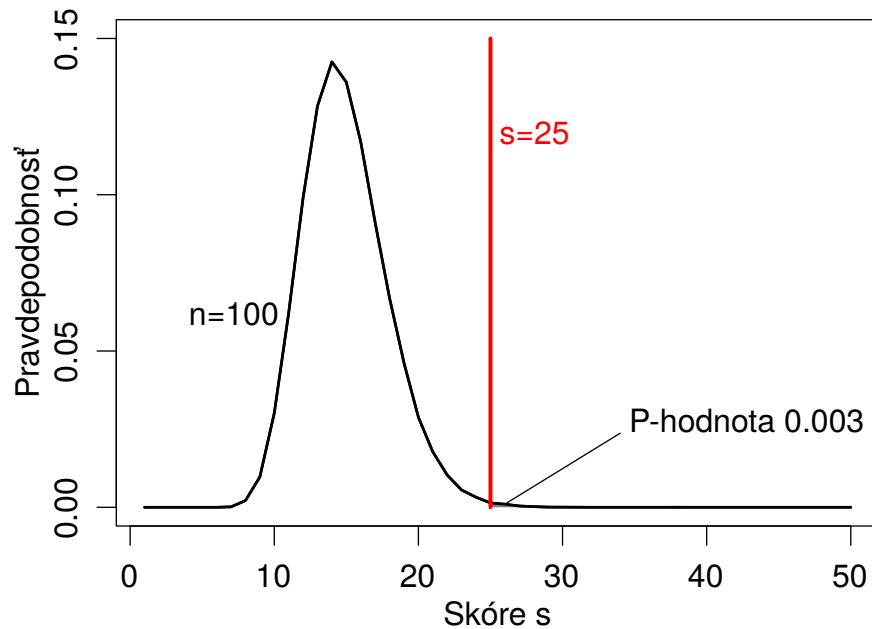
**$E$ -hodnota:** Očakávaný počet zarovnaní so skóre aspoň  $S$  nájdených pre náhodný dotaz dĺžky  $m$  v náhodnej databáze dĺžky  $n$ .

Pri veľmi malých hodnotách sú  $E$ -value a  $P$ -value takmer identické.

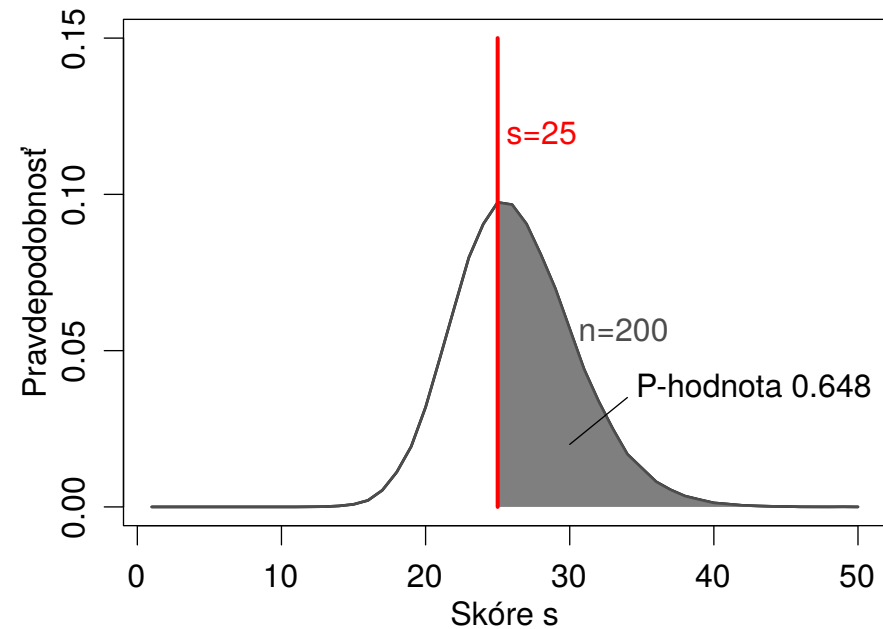
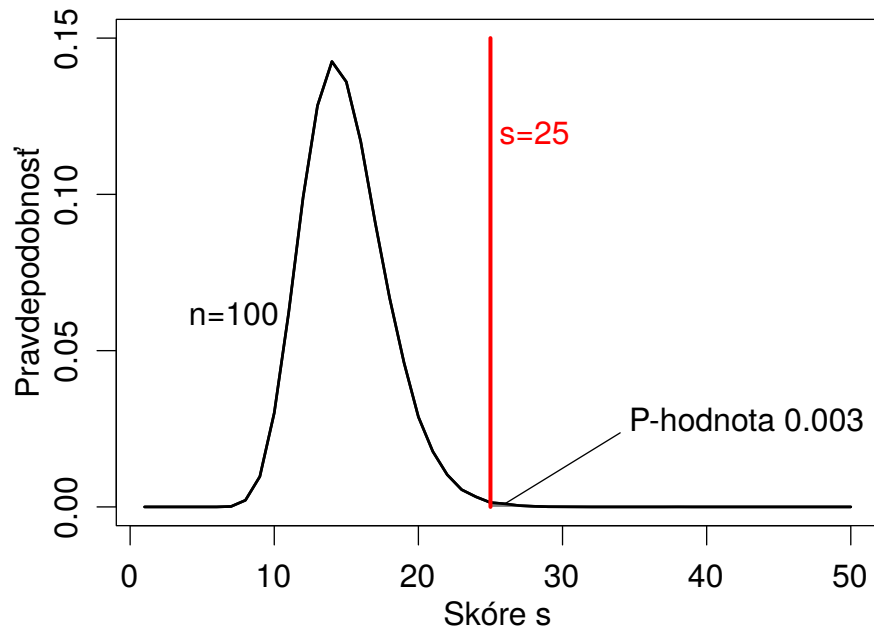
[Karlin and Altschul, 1990, Dembo et al., 1994]

## Výpočet P-hodnoty simuláciou

- Vygenerujeme náhodne dve sekvencie dĺžky  $n$
- Spočítame ich najlepšie lokálne zarovnanie (schéma +1/-1)
- Zaznamenáme si výsledné skóre
- Opakujeme veľa krát



## Výpočet P-hodnoty simuláciou (pokr.)



### P-hodnota pre skóre 25:

Aká časť zarovnaní má skóre 25 alebo vyššie?

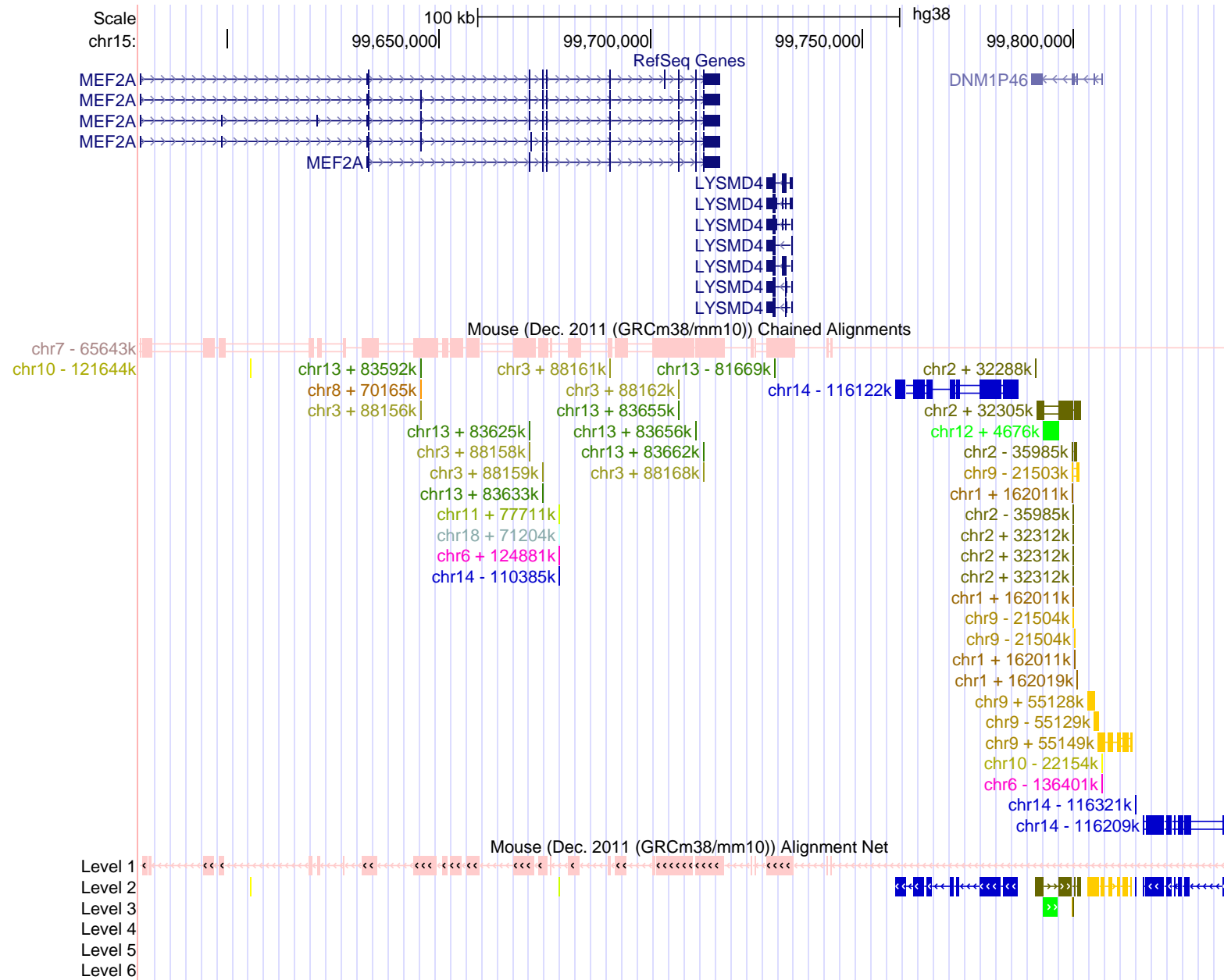
(V praxi je simulácia pomalá, existujú odhady rozdelenia.)



## Genomické zarovnanie (whole-genome alignments)

Ku každému úseku ľudského genómu nájsť zodpovedajúcu časť z myši, psa, sliepky, atď. (predpočítané v UCSC browseri) [Kent et al., 2003]

- Lokálne zarovnanie nájdu exóny a iné zachované časti, sú však úseky, ktoré sa príliš zmenili.
- Pri duplikovaných úsekoch nevieme rozhodnúť, ktoré dvojice úsekov patria k sebe.
- **Synténia (synteny)**: lokálne zarovnanie, ktoré sa nachádzajú v dvoch genómoch v tom istom poradí a orientácii.  
Pomáha nám určiť, ktoré dvojice úsekov vznikli z tej istej oblasti v spoločnom predkovi (ortológ)



## Viacnásobné zarovnanie, multiple sequence alignment

Zarovnaj viacero sekvencií.

**Zložitosť:**  $O(2^k n^k)$  pre  $k$  sekvencií dĺžky  $n$

Pre všeobecné  $k$  NP-ťažké.

```
Human ctccatagcaatgt-cagagatagggcagagcggat-----ggtggtgac
Rhesus ctccatggcaatgt-cagagatagggcagagcggat-----gctggtgac
Mouse ttt--tgacaaca--tagagac-tgagatagaaaat-----atgctgac
Dog -tccccgctaatgtacaaagatggggcag-gaaga--a----tgtgctgaa
Horse -tccacggcaatac-tggagatggggcagagcaga--agat-ggtgatgaa
Armadillo ctgcatagaaatct-cagagatgggggaaagcaga-----agacattcat
Opossum atccatggaaacat-cagaagtgggagaaatagaaga----tggcaatga-
Platypus acccggggaagggg-aagaggaagggccggccg-----
```

Heuristické algoritmy, napr. CLUSTAL-W [Higgins et al., 1996], MUSCLE [Edgar, 2004] a TBA [Blanchette et al., 2004].

## Zhrnutie

- Zarovnávanie (alignment) je základný nástroj bioinformatiky
- Formulácia problému: voľba skórovacej schémy
- Riešenie problému: presné ale pomalé algoritmy a rýchlejšie heuristiky, ktoré nie vždy nájdu všetko
- Špecializované programy na rôzne úlohy súvisiace so zarovnávaním

## Organizačné poznámky

- DÚ1 je zverejnená, odovzdávanie do štvrtka 8.11. 14:00 (inf) resp. 15:40 (bio)
- Dnes na konci prednášky zverejníme rozdelenie skupín na journal club

# Hľadanie génov

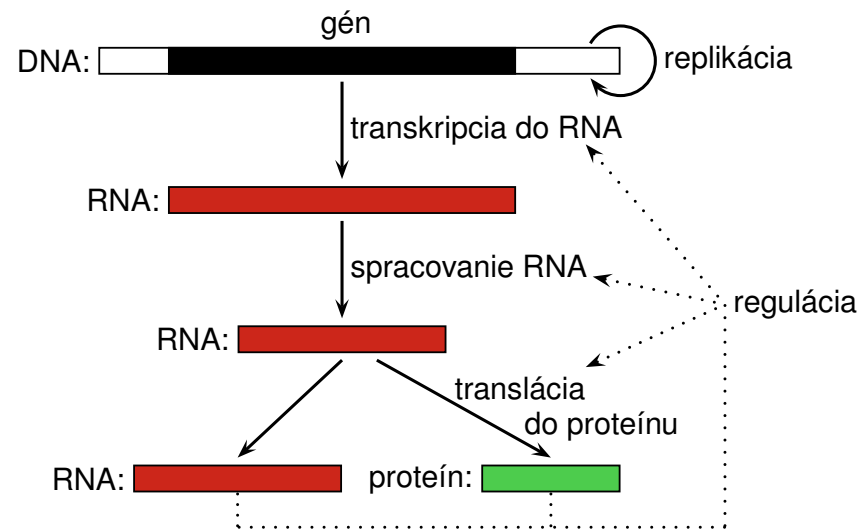
**Broňa Brejová**

**25.10.2018**

## Čo s osekvenovanými genómami?

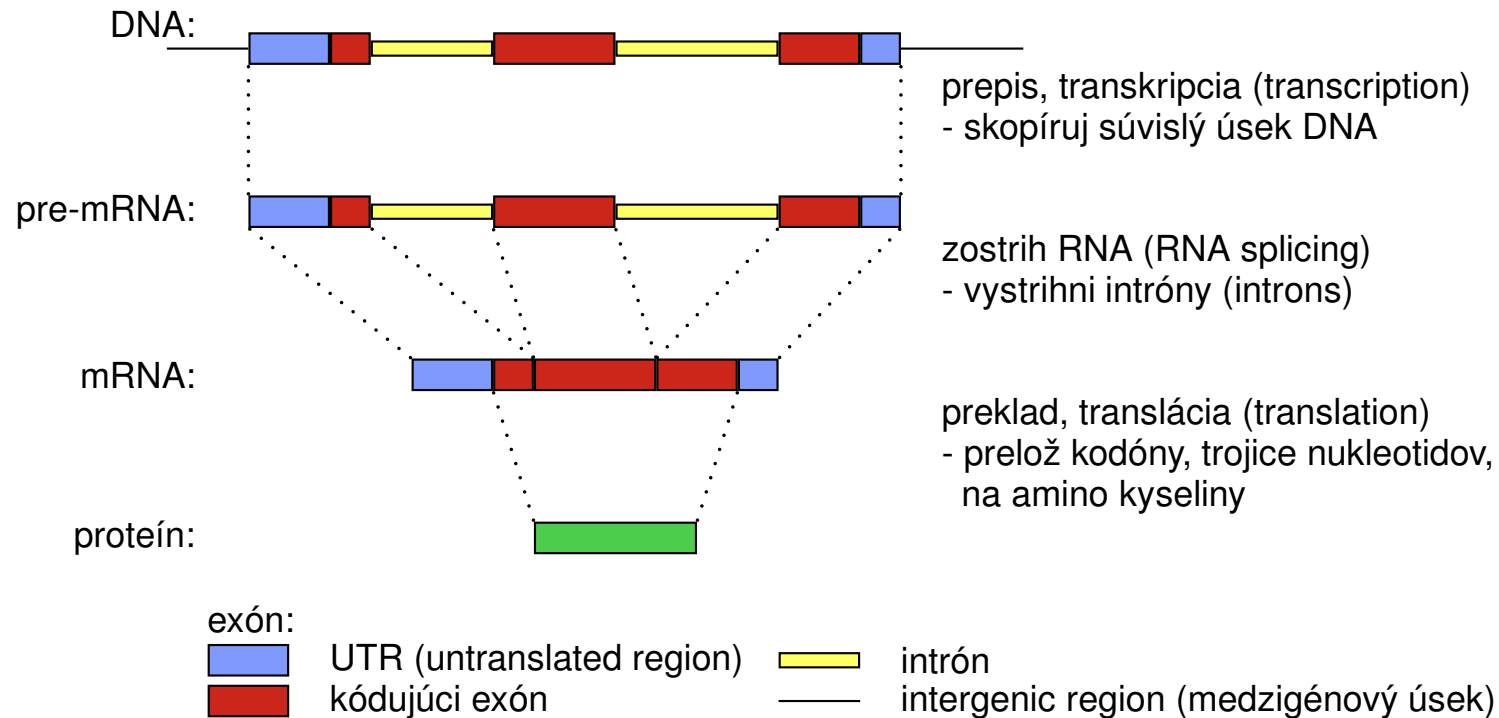
Chceme vedieť, čo genóm kóduje, hľadáme zaujímavé prvky, ako:

- gény kódujúce proteíny (dnešná prednáška)
- RNA gény
- signály pre reguláciu transkripcie, zostrihu, atď
- pseudogény (nefunkčné kópie génov)
- repetitívne sekvencie, opakovania (sequence repeats)

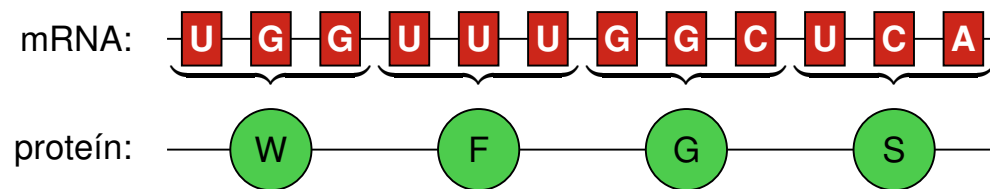


# Štruktúra eukaryotických génov

## Proces syntézy proteínov:



## Translácia: tri bázy mRNA (kodón) → aminokyselina proteínu





## Ľudský genóm

- gény kódujúce proteíny
  - cca 20,000, pokrývajú 40% genómu
  - cca 10 exónov v géne
  - exóny pokrývajú 2% genómu
  - kódujúce exóny 1.2% genómu
- repetitívne sekvencie
  - pokrývajú 49% genómu

## Bioinformatický problém: hľadanie génov

**Cieľ:** nájsť všetky gény kódujúce proteíny v genóme.

Tým získame katalóg všetkých proteínov.

### Zjednodušená:

- neuvažujeme alternatívny zostrih, prekrývajúce sa gény
- nehľadáme neprekladané oblasti (UTRs) na začiatku a konci génu

## Bioinformatický problém: hľadanie génov

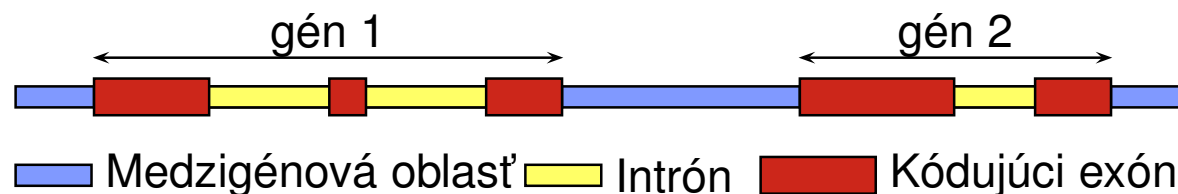
### Vstup: sekvencia DNA

```
cggtgaaactgcacgattggtgctggcttaaagatagaccaatcagagtgtgtaacgtca  
tatttagcgtcttctatcatccaatcactgcactttacacactataaatagagcagctca  
tgggcgtatattgcgctagtgttgggtgttccgctgtgctgtttttccgtcatggctcgca  
ctaagcaaactgctcgggaagtctactggtggcaaggcgccacgcaaacagttggccacta  
aggcagcccgcaaaagcgctccggccaccggcggcgtgaaaaagccccaccgctaccggc  
cgggcaccgtggctctgcgcgagatccgccgttatcagaagtccactgaactgcttattc  
gtaaactacctttccagcgcctggtgcgcgagattgcgcaggactttaaacagacctgc  
gtttccagagctccgctgtgatggctctgcaggaggcgtgcgaggcctacttggtagggc  
tatttgaggacactaacctgtgcgccatccacgccaagcgcgtcactatcatgccaagg  
acatccagctcgcccgccatccgcggagagagggcgtgattactgtggtctctctgac
```

## Bioinformatický problém: Hľadanie génov

**Cieľ:** označ každú bázu ako intrón/exón/medzigénovú oblasť

```
cgggtgaaactgcacgattggtgctggcttaaagatagaccaatcagagtgtgtaacgtca  
tatttagcgtcttctatcatccaatcactgcactttacacactataaatagagcagctca  
tgggcgtatTTTgCGctagtgttgggtgttccgctgtgctgtttttccgtcatggctcgca  
ctaagcaaactgctcggaaGTctactggtggcaaggCGccacgcaaacagttggccacta  
aggcagCCCGcaaaagcgctccggccaccggCGgctgaaaaagccccaccgctaccggc  
cgggcaccgtggctctgCGcgagatccgCCgttatcagaagtccactgaactgcttattc  
gtaaactacctttccagCGcctggtgCGcgagattgCGcaggactttaaaacagacctgc  
gtttccagagctccgctgtgatggctctgCaggaggCGtgcgaggcctacttggtagggc  
tatttgaggacactaacctgtgCGccatccacgccaagCGcgtcactatcatgccaagg  
acatccagctCGcccgcgcatccgCGgagagaggCGgtgattactgtggtctctctgac
```



## Bioinformatický problém: hľadanie génov

**Vstup:** sekvencia DNA

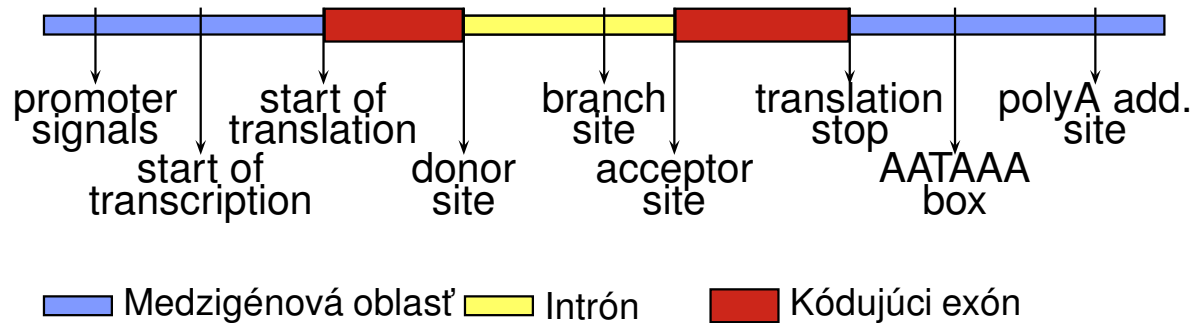
**Cieľ:** označ každú bázu ako intrón/exón/medzigénovú oblasť  
(anotácia)

- Toto nie je dobre definovaný problém!  
Ako spoznáme, čo je gén?

## Ako spoznáme gény?

**Signály** na hraniciach exónov:

krátke reťazce, kde sa viažu komplexy zúčastňujúce sa na expresii génu



**Príklad signálu:** miesto zostrihu

**Exón**                      **Intrón**

```
ccatcccctatatatttatggcagGTgaggaaaggggtgggggctgggg
attcatcatcatgggtgcatcgGTgagtatctccaggccccaatc
agaagatctacccaccatctgGTAagtgtgtcccaccactgcccc
acagagtgagcccttcttcaagGTgggtggtgtcagggcctcccc
acgagtcctgcatgagccagatGTAaggcttgccgttgccctcct
tgcagaacctcatgggtgctgagGTggggccaagcctgggcccggggg
tcgatgaatttgggatcatccgGTgagagctcttctctctctctgg
agatgacgtccgtgatgagaagGTaggggggtgcaccccagtccca
gtggagaatgagaggtgggatgGTaggtgatgccttcgaggccag
tttcttgtggcctattttaaaagGTAattcatggagaaatagaaaa
```

## Ako spoznáme gény?

### Zloženie sekvencie:

- iná frekvencia  $k$ -tic báz v kódujúcich a nekódujúcich oblastiach,
- kódujúce oblasti sú 3-periodické,
- stop kodóny (TAA, TGA, TAG) len na konci posledného kódujúceho exónu.

**Príklad:** ak uvažujeme len jednotlivé bázy, exóny majú viac C a G (ľudský genóm)

		a	c	g	t
kódujúci exón	<b>0</b>	0.26	0.26	0.32	0.16
	<b>1</b>	0.30	0.24	0.20	0.26
	<b>2</b>	0.17	0.32	0.31	0.20
intrón		0.26	0.22	0.22	0.30
medzig.		0.27	0.23	0.23	0.27

## Bioinformatický problém: hľadanie génov

**Vstup:** sekvencia DNA

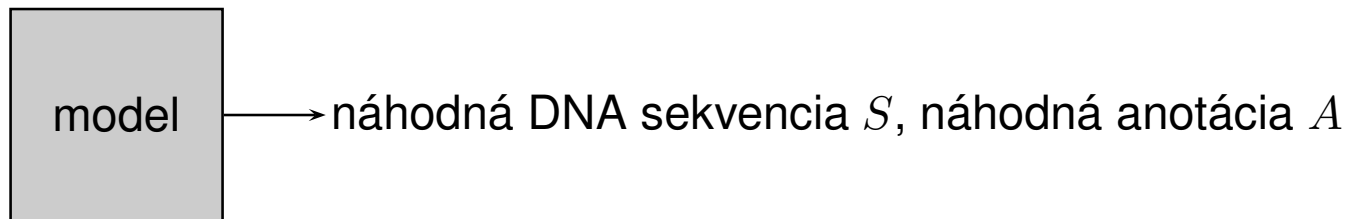
**Cieľ:** označ každú bázu ako intrón/exón/medzigénovú oblasť (anotácia)

- Toto nie je dobre definovaný problém!  
Ako spoznáme, čo je gén?
- Žiadna informácia nám neumožňuje jednoznačne určiť, čo je gén.
- Chceme **skórovací systém**, ktorý povie, ako dobre potenciálna anotácia zodpovedá našim znalostiam.
- Potom hľadáme anotáciu (sadu neprekrývajúcich sa génov) **s maximálnym skóre.**
- Na definíciu skórovacieho systému použijeme **pravdepodobnostné modely.**



## Pravdepodobnostný model génov

Žiadna informácia nám neumožňuje jednoznačne určiť, čo je gén.  
Skombinujeme dostupnú informáciu pravdepodobnostným modelom.



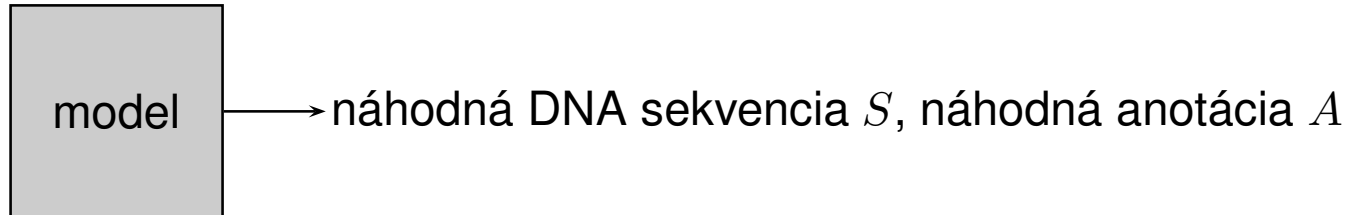
$\Pr(S, A)$  – pravdepodobnosť, že model vygeneruje pár  $(S, A)$ .

Model zostavíme tak, aby páry s vlastnosťami podobnými skutočným génom mali veľkú pravdepodobnosť.

**Použitie:** pre novú sekvenciu  $S$  nájdí najpravdepodobnejšiu anotáciu

$$A = \arg \max_A \Pr(A|S)$$

## Pravdepodobnostný model génov



**Použitie:** pre sekvenciu  $S$  nájdí najpravdepodobnejšiu anotáciu  $A$

**Hračkársky príklad modelu:** sekvencie dĺžky 2

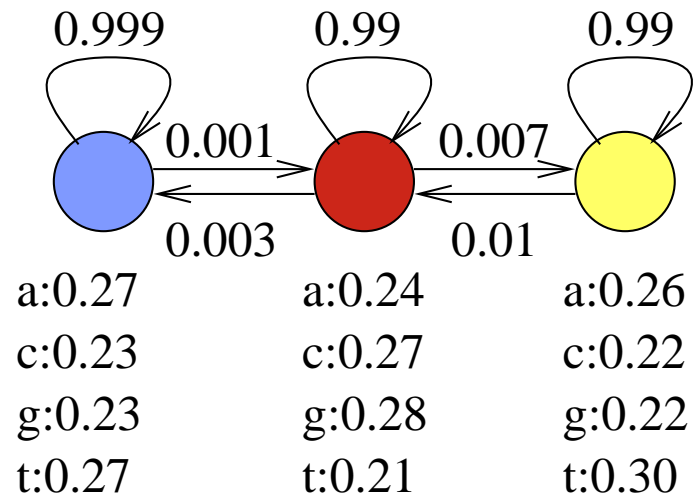
Tabuľka pravdepodobností pre 16 sekvencií, 9 anotácií (súčet 1)

Najpravdepodobnejšia anotácia pre  $S = aa$  je **aa**.

<b>aa</b>	0.008	<b>ac</b>	0.009	<b>ag</b>	0.0085	...
<b>aa</b>	0	<b>ac</b>	0	...		
<b>aa</b>	0.011	...				
<b>aa</b>	0					
<b>aa</b>	0.009					
<b>aa</b>	0					
<b>aa</b>	0.007					
<b>aa</b>	0					
<b>aa</b>	0.010					

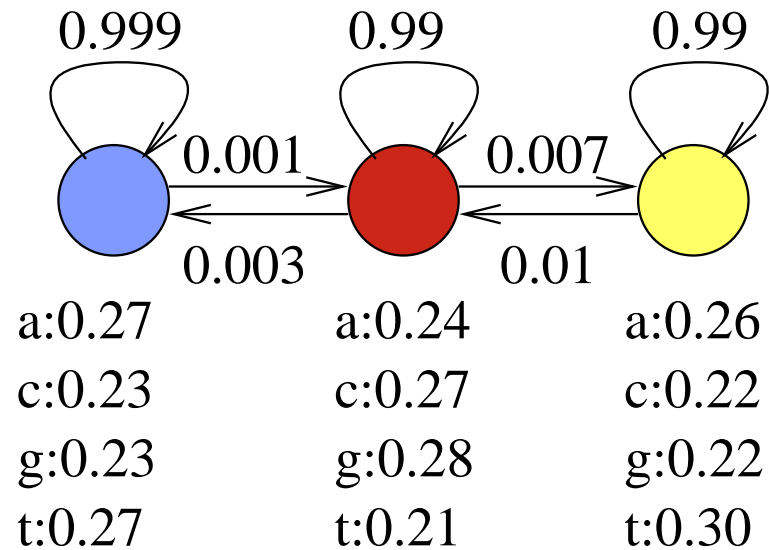
## Skrytý Markovov model, hidden Markov model (HMM)

Spôsob, ako zdefinovať model pre dlhšie sekvencie.



- Konečný automat, stavy napr. exón, intrón, medzigénová oblasť
- Sekvenciu aj anotáciu generuje bázu po báze
- V každom kroku je v jednom stave a náhodne vygeneruje jednu bázu podľa tabuľky v stave
- Potom sa presunie do ďalšieho stavu podľa pravdepodobností na hranách

## Skrytý Markovov model (HMM)



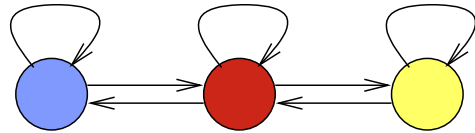
Predpokladajme, že model vždy začína v modrom stave.

### Príklad:

$$\Pr(\text{a} \color{red}{\text{c}} \text{a}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 = 0.000017$$

$$\Pr(\text{a} \color{blue}{\text{a}} \text{a}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 = 0.017$$

## Matematické označenie



Sekvencia  $S_1, \dots, S_n$










Anotácia  $A_1, \dots, A_n$

## Parametre modelu:

Prechodová pravdepodobnosť  $a(u, v) = \Pr(A_{i+1} = v | A_i = u)$ ,

Emisná pravdepodobnosť  $e(u, x) = \Pr(S_i = x | A_i = u)$ ,

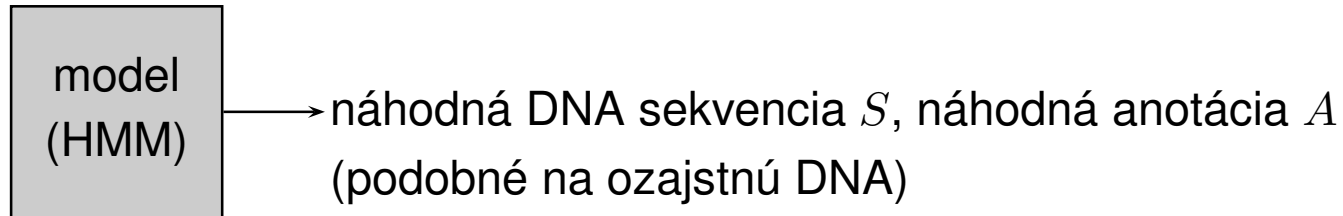
Počiatočná pravdepodobnosť  $\pi(u) = \Pr(A_1 = u)$ .

$a$				$e$	a	c	g	t
	0.99	0.007	0.003		0.24	0.27	0.28	0.21
	0.01	0.99	0		0.26	0.22	0.22	0.30
	0.001	0	0.999		0.27	0.23	0.23	0.27

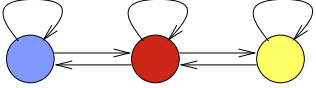
**Výsledná pravdepodobnosť:**  $\Pr(A_1, \dots, A_n, S_1, \dots, S_n) =$

$$\pi(A_1)e(A_1, S_1) \prod_{i=2}^n a(A_{i-1}, A_i)e(A_i, S_i)$$

## Hľadanie génov s HMM



$\Pr(S, A)$  – pravdepodobnosť, že model vygeneruje pár  $(S, A)$ .

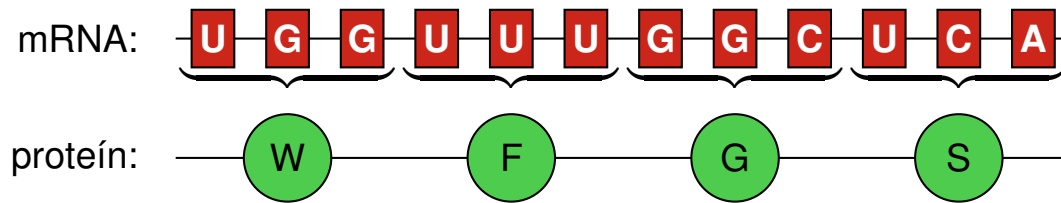
- **Určenie stavov a prechodov v modeli:** ručne, na základe poznatkov o štruktúre génu. 

- **Trénovanie parametrov:** emisné a prechodové pravdepodobnosti určíme na základe sekvencií so známymi génmi (**trénovacia množina**).

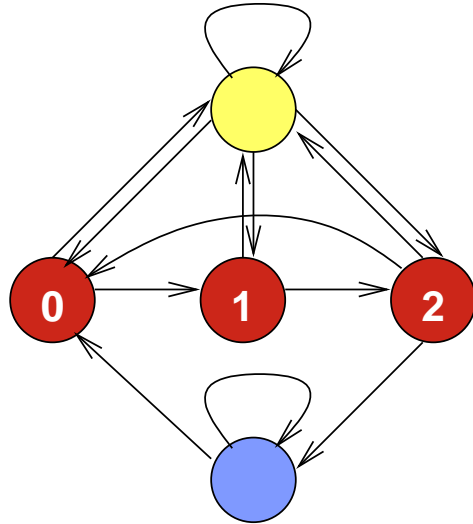
- **Použitie:** pre novú sekvenciu  $S$  nájdí najpravdepodobnejšiu anotáciu  $A = \arg \max_A \Pr(A|S)$   
Viterbiho algoritmus v čase  $O(nm^2)$  (dynamické programovanie)

## HMM na hľadanie génov: 3-periodické exóny

Kodón (trojica báz) → jedna aminokyselina



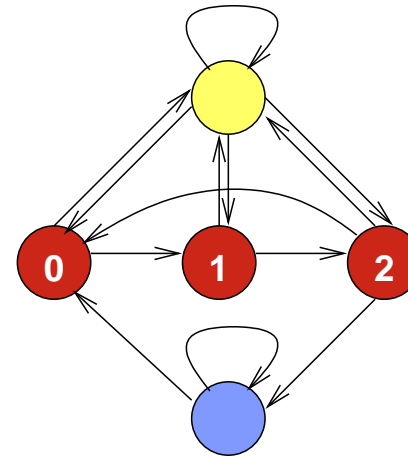
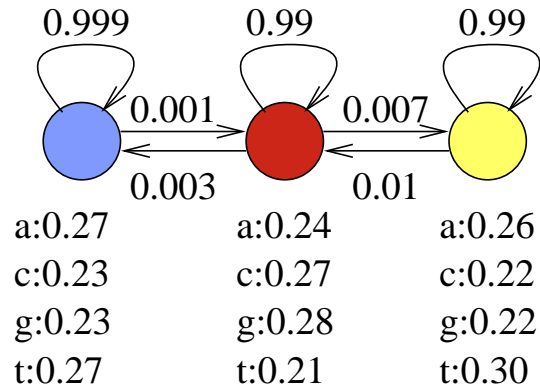
Namiesto jedného stavu pre exón použijeme tri stavy v cykle.



$a$	<b>0</b>	<b>1</b>	<b>2</b>	Yellow	Blue
<b>0</b>	0		0		0
<b>1</b>	0	0			0
<b>2</b>		0	0		
Yellow					0
Blue		0	0	0	

$\Pr(A_i|A_{i-1})$

## Nové stavy mají odlišné emisné pravdepodobnosti



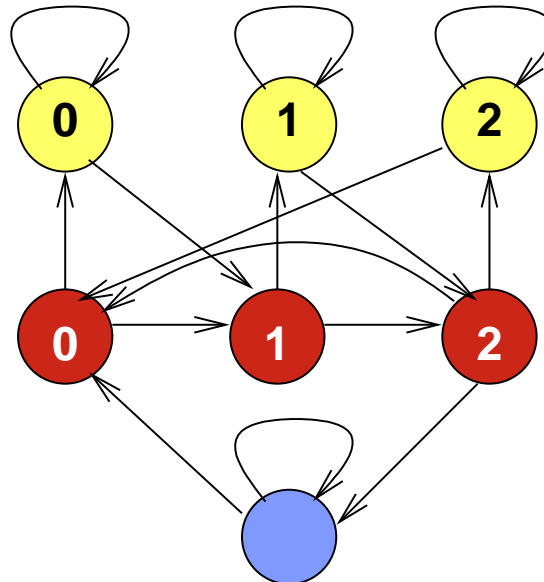
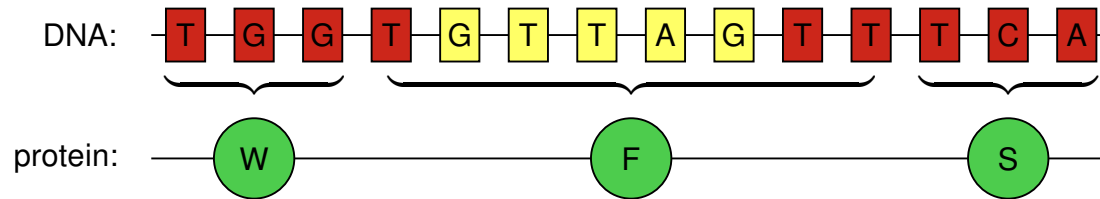
<i>e</i>	a	c	g	t
■	0.24	0.27	0.28	0.21
■	0.26	0.22	0.22	0.30
■	0.27	0.23	0.23	0.27

<i>e</i>	a	c	g	t
0	0.26	0.26	0.32	0.16
1	0.30	0.24	0.20	0.26
2	0.17	0.32	0.31	0.20
■	0.26	0.22	0.22	0.30
■	0.27	0.23	0.23	0.27



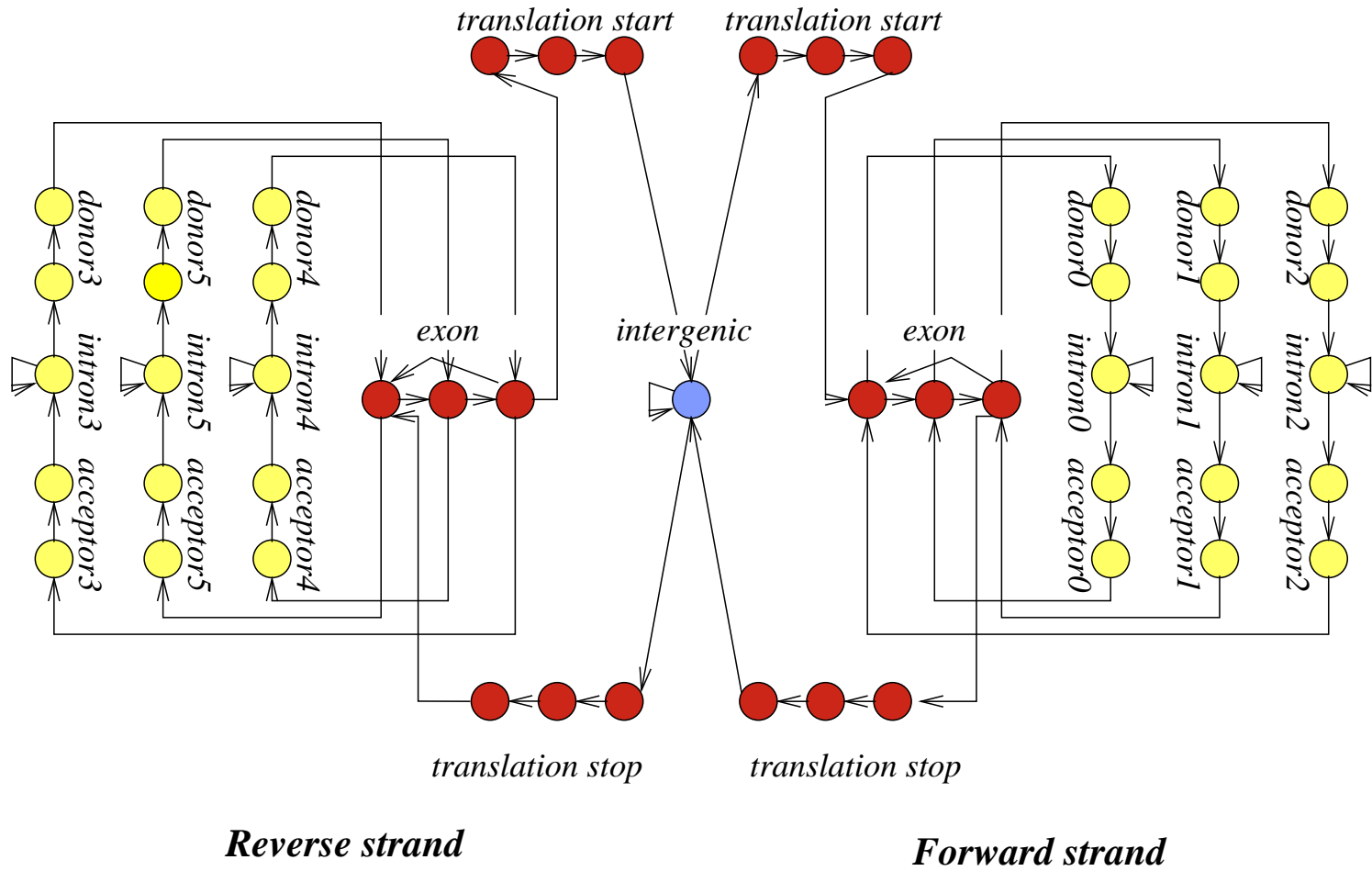
## HMM na hľadanie génov: konzistentné kodóny

Intrón môže prerušiť kodón uprostred, chceme pokračovať, kde sme prestali.







# HMM na hľadanie génov: celkový model



## Stavy vyšších rádov

**Rád 0:** emisná tabuľka  $e$  určuje  $\Pr(S_i|A_i)$

**Rád 1:**  $e$  určuje  $\Pr(S_i|A_i, S_{i-1})$

$A_i$	$S_{i-1}$	a	c	g	t
	a	0.24	0.23	0.34	0.19
	c	0.30	0.31	0.13	0.26
	g	0.27	0.28	0.28	0.17
	t	0.13	0.28	0.38	0.21
	a	0.30	0.18	0.27	0.25
	c	0.32	0.28	0.06	0.35
	g	0.27	0.22	0.27	0.24
	t	0.20	0.21	0.26	0.33

...

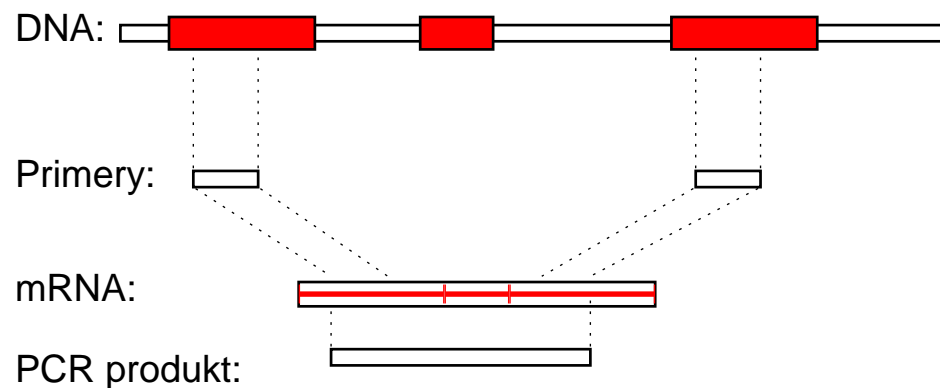
Na charakterizovanie exónov, intrónov atď používame rád 4-5.

## Experimentálne overovanie génov

### Overenie transkripcie a zstrihu

- RNA-Seq: sekvenovanie častí mRNA extrahovaných z bunky. Nie je cieleňé na konkrétny gén.
- RT PCR: cielene over konkrétny predpovedaný gén pomocou špecifických primerov.

Problémy: ťažko nájsť gény s expresiou iba za zvláštnych podmienok, napr. v embryu, kontaminácia genómovou DNA, nejednoznačné namapovanie na genóm.



## Experimentálne overovanie génov

### Overenie translácie, prítomnosti proteínu

- Hmotnostná spektrometria (mass spectrometry) dokáže detegovať prítomnosť proteínu izolovaného napr. z 2D gélu.
- Metódy založené na protilátkach (antibody), prípadne špecifické techniky podľa typu proteínu.

## Príklady programov na hľadanie génov

### Len na základe sekvencie DNA:

HMMGene [Krogh, 1997] (autor je priekopníkom HMM v bioinf.),  
Genscan [Burge and Karlin, 1997] (po mnohé roky štandard),  
GeneZilla [Majoros et al., 2004], ExonHunter [Brejová et al., 2005],  
Augustus [Stanke and Waack, 2003] (novšie programy založené na  
zovšeobecnených HMM).

CONTRAST [Gross et al., 2007], CONRAD [DeCaprio et al., 2007]  
(programy založené na conditional random fields, obmena HMM)

### Prokaryotické genómy:

GeneMark [Lukashin and Borodovsky, 1998], Glimmer  
[Delcher et al., 1999] a ďalšie.

## Vybrané programy na hľadanie génov

### Porovnávaním viacerých sekvencií:

Twinscan [Korf et al., 2001]

(prvý úspešný gene finder s dvoma genómami),

Exoniphy [Siepel and Haussler, 2004]

(viacero genómov, nehľadá celé gény),

N-SCAN [Gross and Brent, 2006]

(rozšírenie Twinscanu na viacero genómov).

**Iná informácia:** (napr. RNA-seq, príbuzné proteíny a pod.)

ExonHunter [Brejová et al., 2005], Augustus [Stanke et al., 2006],

Jigsaw [Allen and Salzberg, 2005],

Fgenesh++ [Solovyev et al., 2006].



## Obmedzenia hľadačov génov

- Alternatívny zostrih (alternative splicing): jeden gén môže vyprodukovať viacero mRNA molekúl. Programy väčšinou hľadajú iba jednu.

Retained intron:



Skipped exon:



Alternative donor or acceptor:

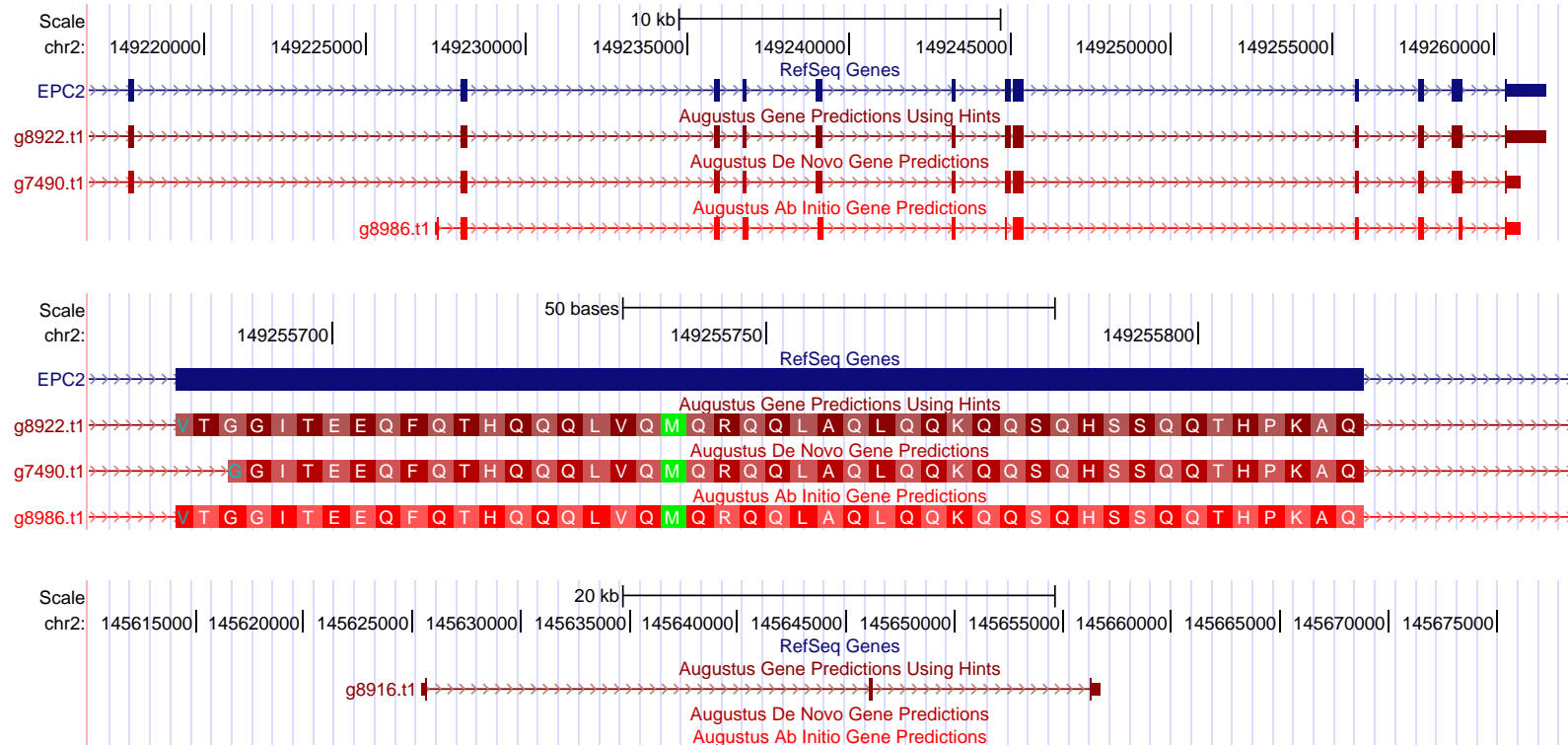


Mutually exclusive exons:



- Pretínajúce sa gény, resp. gény v intrónoch.
- Netypické gény (neobvyklé signály, veľmi krátke alebo dlhé exóny alebo intróny atď.)
- Hľadanie UTR a začiatku/konca transkripcie.

# Hľadáče génov robia často chyby



Najlepšie metódy v 2005 na ľudskom genóme: [Guigo et al 2006]

20% génov, 60% exónov správne iba na základe DNA

35% génov, 65% exónov správne komparatívne

70% génov, 85% exónov správne s ďalšou informáciou

## Koľko g3nov m3 3lovek?

**Do 2001:** R3zne odhady: **50 000–140 000** g3nov

**2001:** predbeŹn3 verzia 3udsk3ho gen3mu: **30 000–40 000** g3nov

**2004:** sekvencia 3udsk3ho gen3mu: **20 000–25 000** g3nov

**2007:** v katal3goch Ensembl, RefSeq a VEGA spolu **24 500** g3nov

[Clamp a kol. 2007] tvrdia, Źe iba **20 500** z nich je spr3vn3ch

Ale s3 g3ny, o ktor3ch eŹte nevieme?

**2010:** RefSeq m3 **22 333** g3nov

St3le neistota  $\pm 1000$  [Pertea, Salzberg 2010]

R3zni 3udia sa m3Źu l3iŹ v desiatkach g3nov

**2012:** Projekt ENCODE odhaduje **20 687** g3nov k3duj3cich prote3ny,

v priemere 6 altern3vn3ch transkriptov na g3n,

plus 8 800 kr3tk3ch a 9 600 dlh3ch RNA g3nov

## Zhrnutie

- Novo osekvenované genómy treba anotovať:  
určovať funkcie jednotlivým oblastiam sekvencie
- Príkladom anotácie je hľadanie génov kódujúcich proteíny
- Na hľadanie génov sa hodia skryté Markovove modely
- Modely robia veľa chýb, ale dajú nám základnú predstavu o polohe a počte génov, môžeme študovať ich funkciu

## Journal club

- Vyhlásime rozdelenie do skupín, každá skupiny sa zoznámte, vymeňte si e-maily.
- Každý si najprv prečíta článok, potom sa koná stretnutie, kde o článku diskutujete, vysvetlíte si navzájom nejasnosti, plánujete písanie správy
- Prvé stretnutie skupiny najneskôr 23.11. (na FMFI alebo PriFUK), čas a miesto oznámte aspoň 2 dni vopred na facebookovej skupine predmetu
- Po stretnutí pošlite e-mail B.Brejovej s krátkou správou zo stretnutia
- Ak treba, dohodnite si s nami konzultácie

## Správa zo journal clubu

- Vlastnými slovami hlavné metódy a výsledky článku
- Pochopiteľná pre študentov tohto predmetu (inf aj bio)
- Netreba pokryť všetko a naopak, môžete využiť aj iné zdroje
- Skúste vložiť vlastný pohľad na tému
- Rozsah cca 1-2 strany na osobu, jeden ucelený text
- V správe vymenujte členov skupiny, ktorí sa podieľali na jej spísaní, dostanú rovnako bodov

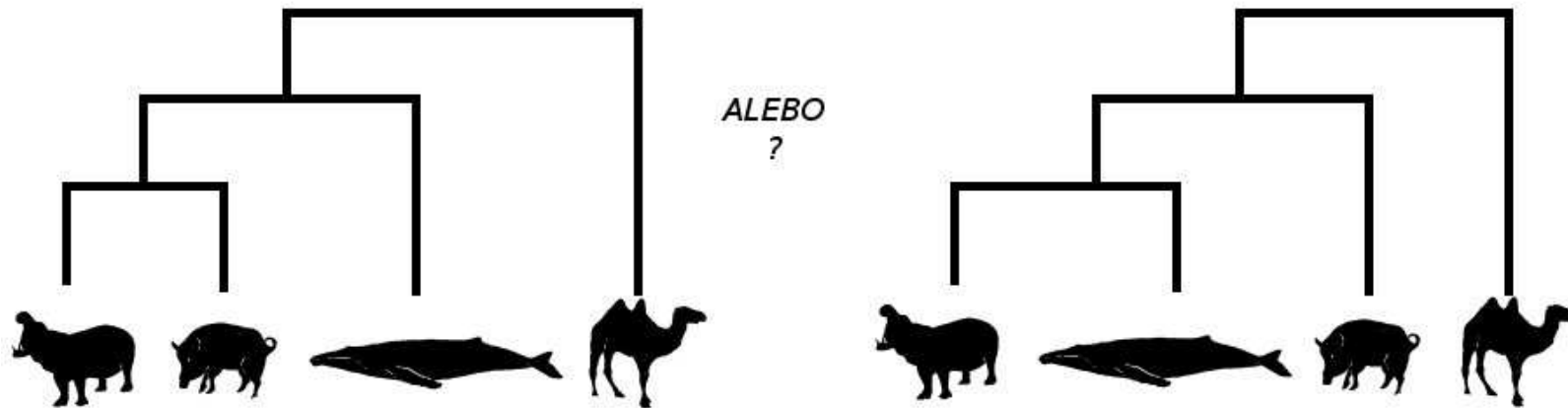
## Organizačné poznámky

- Domáca úloha 1 bola do dnes
- Domáca úloha 2 zadania o dva týždne
- Pracujte na journal clube (prečítajte si článok, naplánujte si stretnutie)

# Evoluční modely a stromy

Broňa Brejová

8.11.2018





## Rekonštrukcia fylogenetických stromov

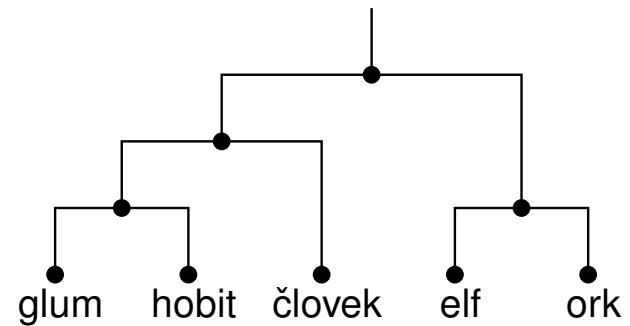
### Vstup:

$m$  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

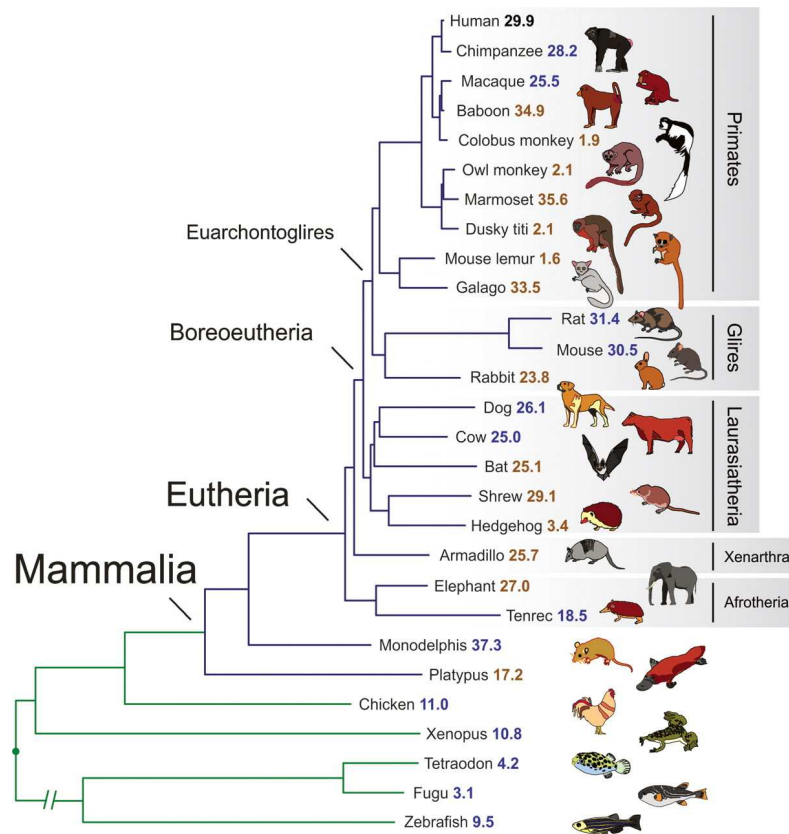


Newick format:

```
((glum,hobit),človek),(elf,ork))
```

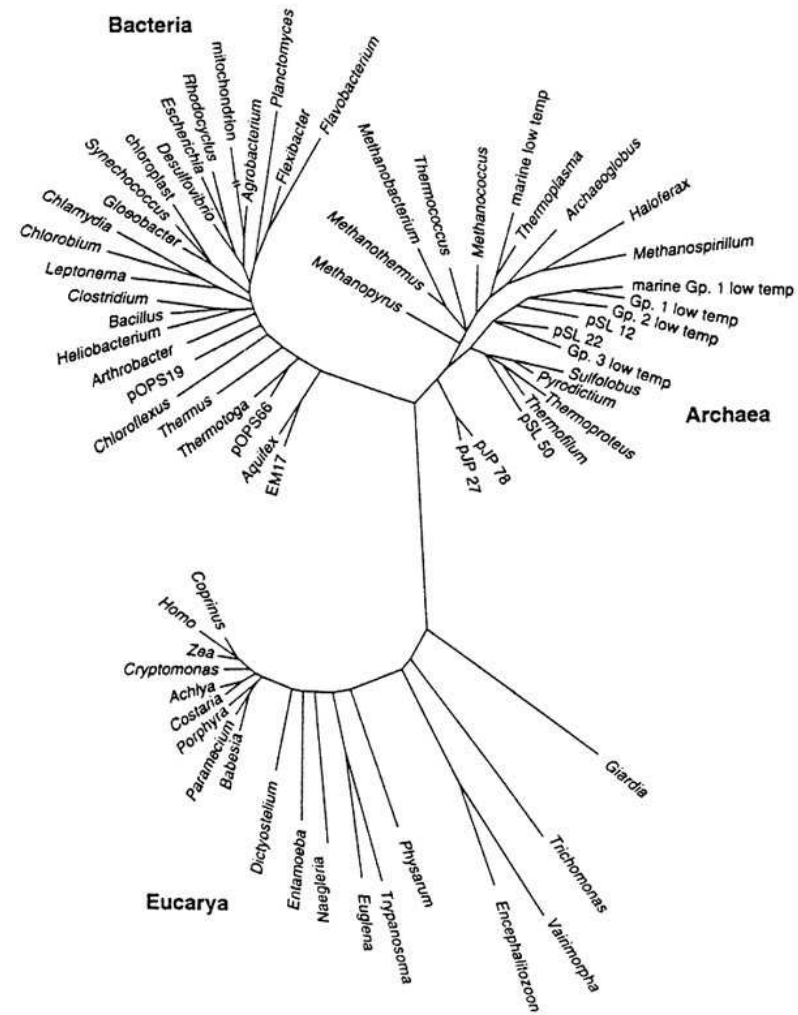
# Zakorenené a nezakorenené stromy

[Margulies et al., 2007]



zakorenený pomocou  
“outgroup”

[Pace, 1997]



## Maximum parsimony (úsporné stromy)

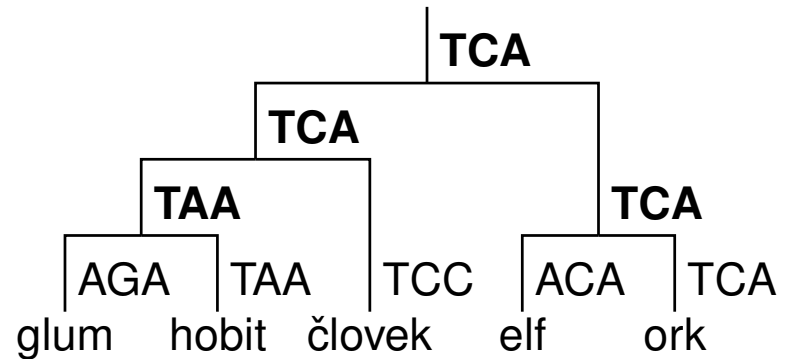
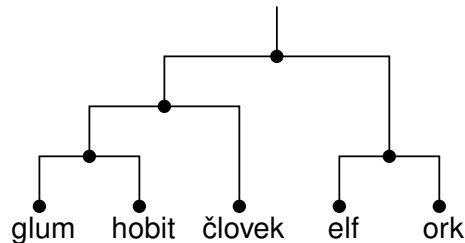
**Úloha:** Dané sú zarovnané sekvencie súčasných organizmov.

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	AGA
hobit	TAA
človek	TCC
elf	ACA
ork	TCA



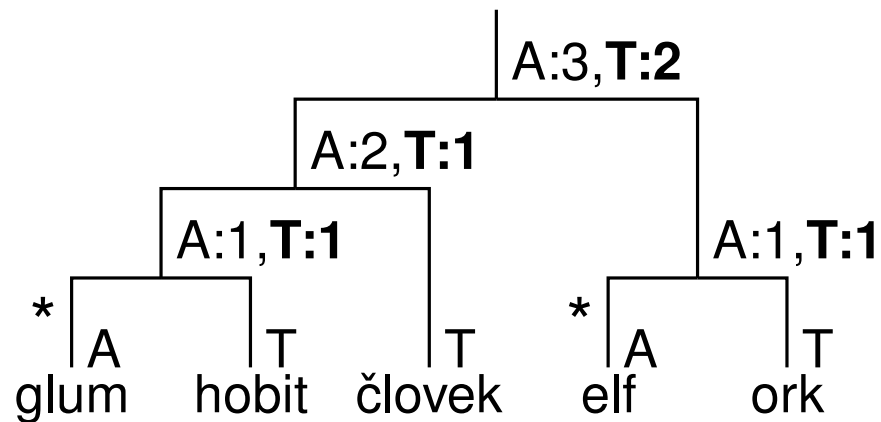
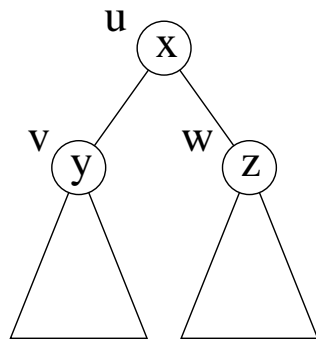
## Výpočet ceny konkrétného stromu

Môžeme rátať **dynamickým programovaním** pre každý stĺpec zarovnania zvlášť.

Pre každý vnútorný vrchol  $u$  a symbol  $x$ :

$N_{u,x}$ : koľko zmien treba v podstrome pod  $u$ , ak v  $u$  bude symbol  $x$ ?

$$N_{u,x} = \min_y \{N_{v,y} + [x \neq y]\} + \min_z \{N_{w,z} + [x \neq z]\}$$



**Časová zložitosť:**  $O(m)$ , lineárna

# Hľadanie najúspornejšieho stromu

## NP-ťažký problém

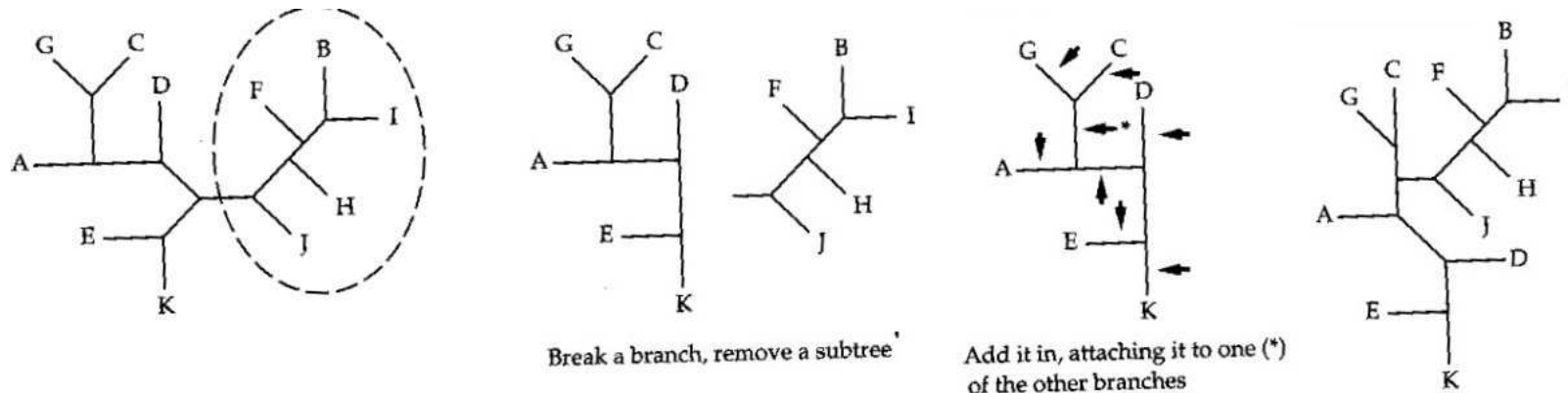
**Triviálny algoritmus:** vyskúšaj všetky možné stromy.

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)

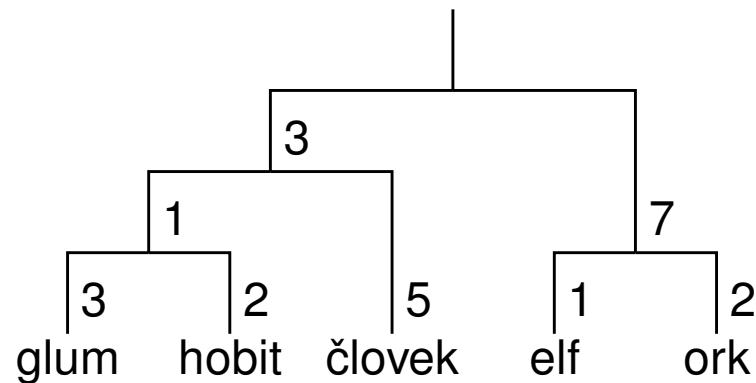
- Nevyužívame detaily rozdielov medzi sekvenciami
- Zosumarizujeme ich pomocou **matice vzdialeností** ( $D_{ij}$ )

### Jednoduchý príklad:

človek	C	A	G	T	T	A		Č	E	G	H	O
elf	A	A	T	A	G	A	človek	0	4	3	2	2
Glum	C	C	G	A	G	A	elf	4	0	3	6	2
hobit	C	C	G	T	T	C	Glum	3	3	0	3	5
ork	A	A	T	T	T	A	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**)



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

	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

## Idea spájania susedov

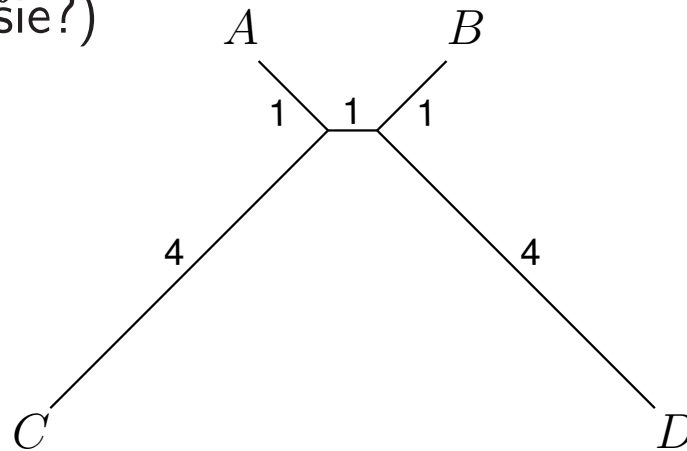
- Predpokladáme, že vzdialenosti  $D_{i,j}$  skutočne zodpovedajú vzdialenostiam v strome (**aditivita**)
- Nájdeme dva listy  $i$  a  $j$ , o ktorých vieme **s určitosťou povedať**, že majú vo výslednom strome spoločného otca
- $i$  a  $j$  spojíme a nahradíme ich ich otcom  $k$  s novými vzdialenosťami:

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



## Ako určiť dva listy na spájanie?

(Prečo nie dva najbližšie?)



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

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

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

$m$ : počet listov

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

## Ako určiť dva listy na spájanie?

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

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

$D$	$L$	nové $D$																																																																																																							
<table style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <th style="width: 5%;"></th> <th style="width: 10%;">g</th> <th style="width: 10%;">h</th> <th style="width: 10%;">č</th> <th style="width: 10%;">e</th> <th style="width: 10%;">o</th> <th style="width: 15%;"><math>r_i</math></th> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">g</th> <td>0</td> <td>5</td> <td>9</td> <td>15</td> <td>16</td> <td>45</td> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">h</th> <td>5</td> <td>0</td> <td>8</td> <td>14</td> <td>15</td> <td>42</td> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">č</th> <td>9</td> <td>8</td> <td>0</td> <td>16</td> <td>17</td> <td>50</td> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">e</th> <td>15</td> <td>14</td> <td>16</td> <td>0</td> <td>3</td> <td>48</td> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">o</th> <td>16</td> <td>15</td> <td>17</td> <td>3</td> <td>0</td> <td>51</td> </tr> </table>		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	<table style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <th style="width: 5%;"></th> <th style="width: 10%;">g</th> <th style="width: 10%;">h</th> <th style="width: 10%;">č</th> <th style="width: 10%;">e</th> <th style="width: 15%;"><math>r_j</math></th> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">g</th> <td>.</td> <td>-72</td> <td>-68</td> <td>-58</td> <td>-48</td> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">h</th> <td>-72</td> <td>.</td> <td>-68</td> <td>-48</td> <td>-48</td> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">č</th> <td>-68</td> <td>-68</td> <td>.</td> <td>-50</td> <td>-50</td> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">e</th> <td>-58</td> <td>-48</td> <td>-50</td> <td>.</td> <td><b>-90</b></td> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">o</th> <td>-48</td> <td>-48</td> <td>-50</td> <td><b>-90</b></td> <td>.</td> </tr> </table>		g	h	č	e	$r_j$	g	.	-72	-68	-58	-48	h	-72	.	-68	-48	-48	č	-68	-68	.	-50	-50	e	-58	-48	-50	.	<b>-90</b>	o	-48	-48	-50	<b>-90</b>	.	<table style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <th style="width: 5%;"></th> <th style="width: 10%;">g</th> <th style="width: 10%;">h</th> <th style="width: 10%;">č</th> <th style="width: 15%;">eo</th> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">g</th> <td>0</td> <td>5</td> <td>9</td> <td>14</td> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">h</th> <td>5</td> <td>0</td> <td>8</td> <td>13</td> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">č</th> <td>9</td> <td>8</td> <td>0</td> <td>15</td> </tr> <tr> <th style="border-top: 1px solid black; border-bottom: 1px solid black;">eo</th> <td>14</td> <td>13</td> <td>15</td> <td>0</td> </tr> </table>		g	h	č	eo	g	0	5	9	14	h	5	0	8	13	č	9	8	0	15	eo	14	13	15	0
	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																																																																																																			
	g	h	č	e	$r_j$																																																																																																				
g	.	-72	-68	-58	-48																																																																																																				
h	-72	.	-68	-48	-48																																																																																																				
č	-68	-68	.	-50	-50																																																																																																				
e	-58	-48	-50	.	<b>-90</b>																																																																																																				
o	-48	-48	-50	<b>-90</b>	.																																																																																																				
	g	h	č	eo																																																																																																					
g	0	5	9	14																																																																																																					
h	5	0	8	13																																																																																																					
č	9	8	0	15																																																																																																					
eo	14	13	15	0																																																																																																					

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

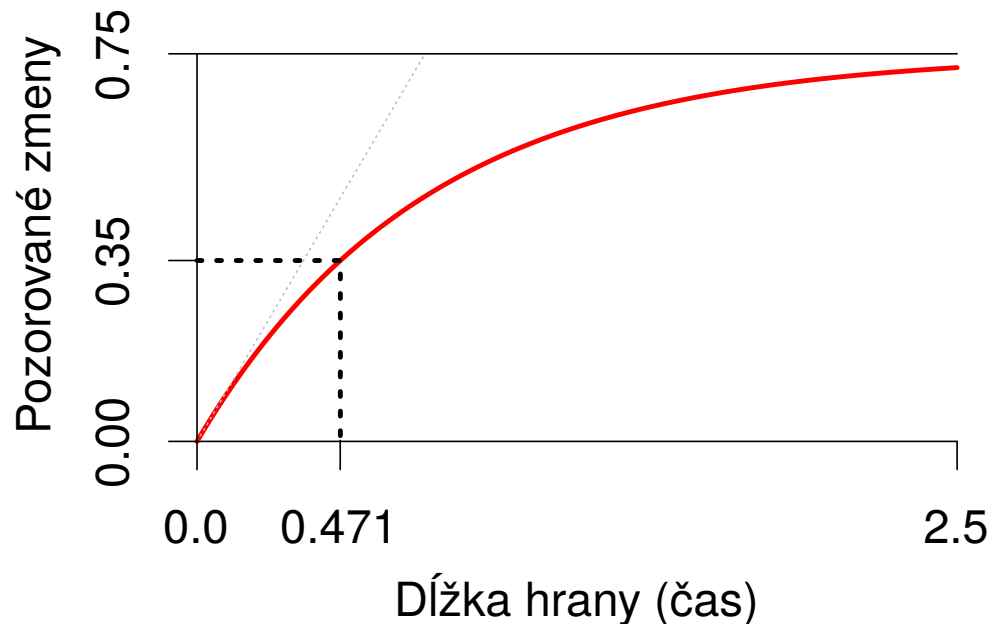
**Pravdepodobnosť zmeny bázy na inú:**

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

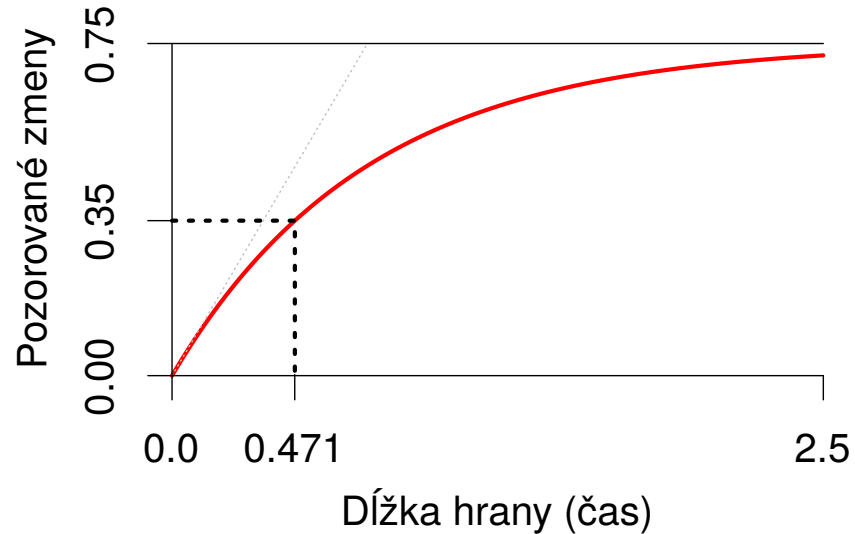
$\alpha$ : rýchlosť evolúcie (počet substitúcií na jednotku času)

**Očakávaný počet pozorovaných zmien na bázu za čas  $t$ :**

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



## Späť ku spájaniu susedov (Neighbor Joining)



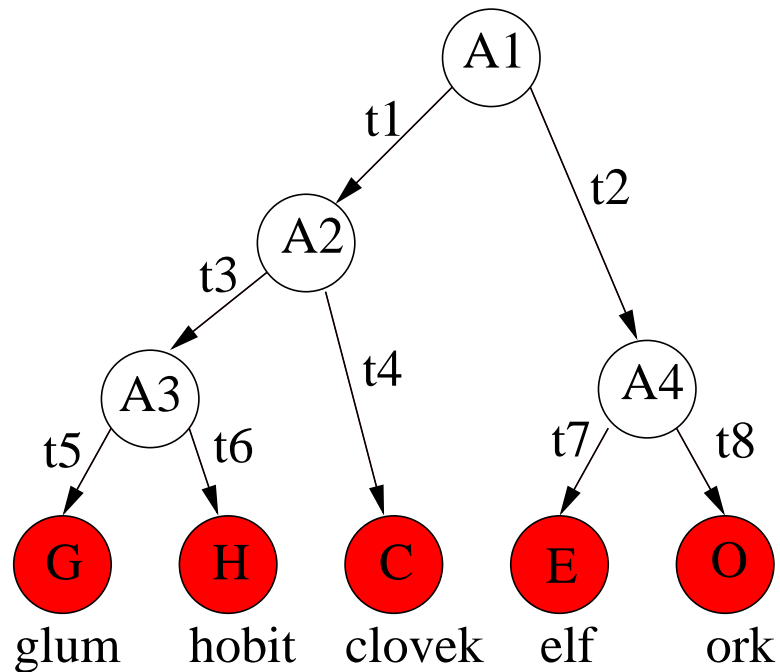
- Podľa takéhoto modelu môžeme korigovať pozorované vzdialenosti

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

- Často lepšie zložitejšie modely, ktoré zahŕňajú rôzne frekvencie báz, pomer tranzícií a transverzií, variabilnú rýchlosť evolúcie na rôznych pozíciách

## Najvierohodnejšie stromy (Maximum likelihood)

- Strom môžeme chápať ako **jednoduchý generatívny model**



- $\Pr(G, H, C, E, O, A1, \dots, A4) = \Pr(A1) \cdot \Pr(A2 | A1, t_1) \cdot \Pr(A4 | A1, t_2) \cdot \Pr(A3 | A2, t_3) \cdot \Pr(G | A3, t_5) \cdot \Pr(H | A3, t_6) \cdot \Pr(C | A2, t_4) \cdot \Pr(E | A4, t_7) \cdot \Pr(O | A4, t_8)$

- $\Pr(G, H, C, E, O, A_1, \dots, A_4) = \Pr(A_1) \cdot \Pr(A_2 | A_1, t_1) \cdot \Pr(A_4 | A_1, t_2) \cdot \Pr(A_3 | A_2, t_3) \cdot \Pr(G | A_3, t_5) \cdot \Pr(H | A_3, t_6) \cdot \Pr(C | A_2, t_4) \cdot \Pr(E | A_4, t_7) \cdot \Pr(O | A_4, t_8)$
- Pre **daný strom** a **dané dĺžky hrán** možno jednotlivé pravdepodobnosti spočítať použitím evolučného modelu (napr. Jukes-Cantor)

- **Vierohodnosť (likelihood) stromu:**

$$\Pr(G, H, C, E, O) = \sum_{A_1, \dots, A_4} \Pr(G, H, C, E, O, A_1, \dots, A_4)$$

- Rátame pomocou **Felsensteinovho algoritmu** (jednoduché dynamické programovanie, podobne ako pre parsimony)
- $\Rightarrow$  Pre daný strom a dĺžky hrán vieme spočítať vierohodnosť v čase  $O(m)$



## Ako nájsť najvieryhodnejší 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) skúsime “podobné” stromy, až kým nevieme zlepšiť

## “Správnosť” fylogenetických algoritmov: Konzistentnosť

- “Rozumne” správajúce sa algoritmy: ak množstvo dát ( $n$ ) rastie, ich odpoveď by sa mala približovať ku správnej odpovedi.
- Hovoríme, že algoritmus pre hľadanie fylogenetického stromu je **konzistentný**, ak v prípade, že  $n$  ide do nekonečna, pravdepodobnosť správneho stromu konverguje k 1.

## Porovnanie algoritmov

	Zložitosť	Konzistentný	Využitie dát
Parsimony (úspornosť)	NP-ťažký	NIE	celé sekvencie
Neighbor Joining	$O(m^3)$	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)

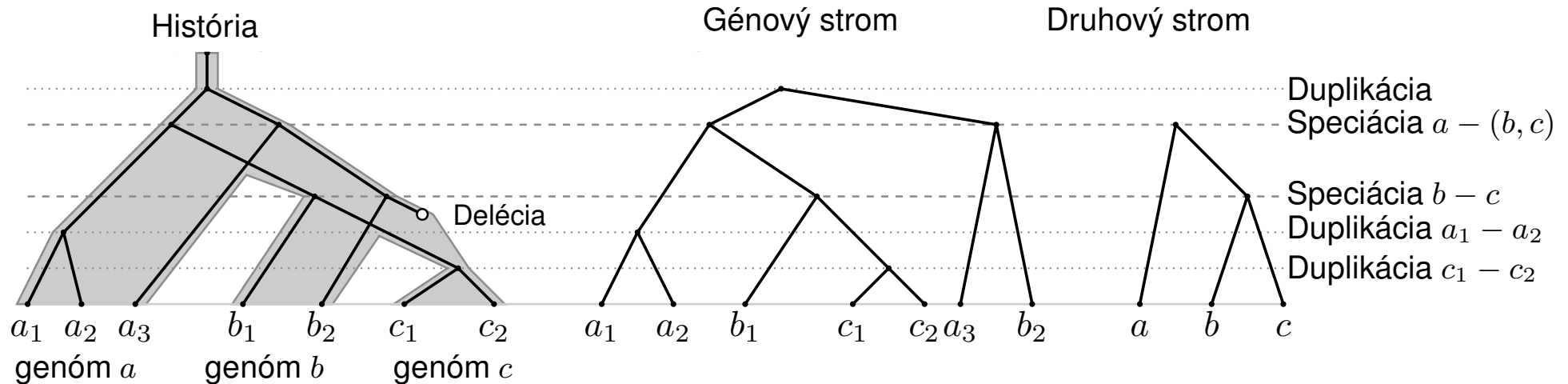
**Chceme využiť aj ďalšie časti genómu.** Čo tak:

- 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

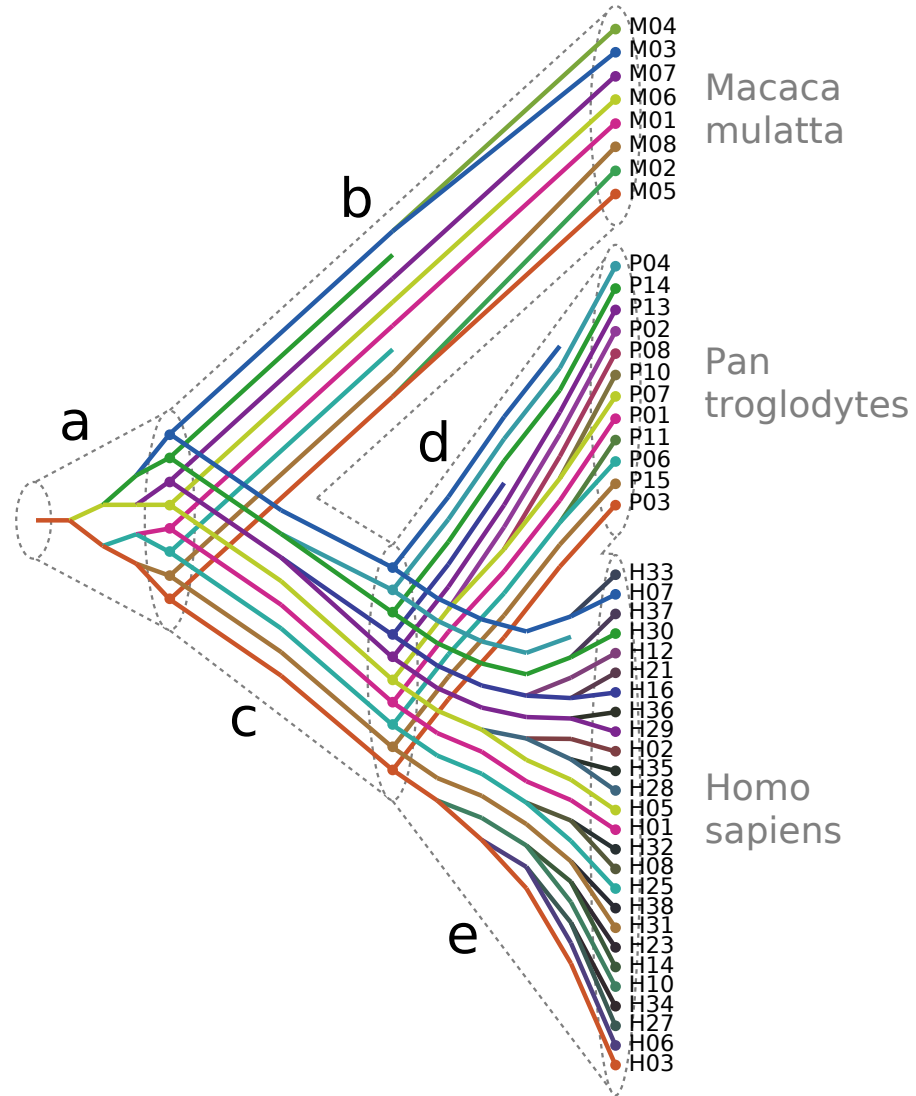
## História duplikovaného génu

**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)
- Génové a druhové stromy; komplikácie pri vytváraní stromov

## Organizačné poznámky

- **Domáca úloha 2**

bude na stránke budúci týždeň

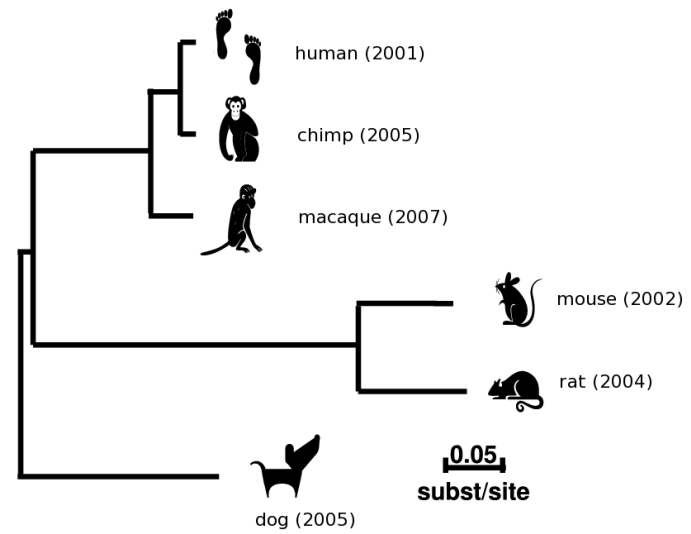
- Nezabudnite na prvé stretnutie ohľadom journal clubu!!!  
(najneskôr 24.11.2018, nezabudnite oznámiť termín na facebooku)



# Komparatívna genomika

Tomás Vinar

15.11.2018



## Komparatívna genomika

- Štúdium evolúcie genómov
  - Mutácie jednotlivých báz DNA (táto prednáška)
  - Krátke inzercie a delécie
  - Väčšie udalosti: prestavby genómu, duplikácie
- Typy mutácií:
  - Neutrálne
  - Škodlivé (deleterious)
    - ⇒ **Purifikačný výber (purifying selection)**
  - Prospešné (advantageous)
    - ⇒ **Pozitívny výber (positive selection)**
- Na základe porovnávania genómov chceme nájsť oblasti s nezvyčajnou evolučnou históriou (zachovávanie dôležitých funkcií, vývoj nových funkcií)

## Komparatívna genomika

- Zostavíme viacnásobné zarovnanie genómov  
(zarovnané miesta by mali pochádzať z tej istej sekvencie spoločného predka)

```
Human  AGTGGCTGCCAGGCTG---GGATGCTGAGGCCTTGTTTGCAGGGAGGT
Rhesus AGTGGCTGCCAGGCTG---GGTTGCTGAGGCCTTGTTTGCCGGGAGGT
Mouse  GGTGGCTGCCGGGCTG---GGTGGCTGAGGCCTTGTTGGTGGGGTGGT
Dog     AGTGGCTGCCCGGCTG---GGTGGCTGAGGCCTTATTTGCAGGGAGGT
Horse  GATGGCTGCCGGGCTG---GGCTGCCGAGGCCTTGTTTCGTGGGGAGGT
Armadillo AGTGGCTGCCGGGCTG---GGAGGCCAAGGCCTTGTTTCGCGGGCAGGT
Chicken AGTGGCTGCCAGTCTGCGCCGTGGCCGACGTCTTGCTCGGGGAAGGT
X. tropicalis AATGGCTTCCATTTTGTGCCGCTGCTGAGGTCTTGTTCTGGGAAGAT
```

- **Metódy:** Kombinujeme techniky na anotáciu (HMM) a pravdepodobnostné modely evolúcie

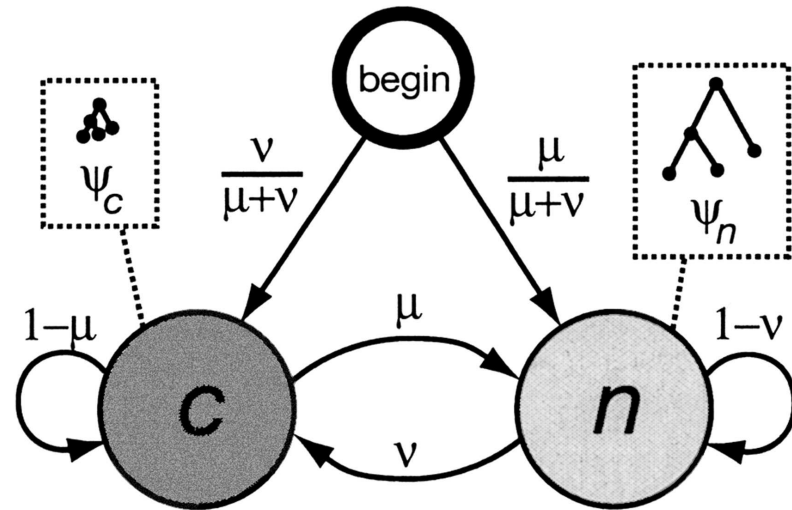
## Príklad 1: Hľadanie funkčných oblastí sekvencií

### Dôsledky purifikačného výberu:

- Funkčné časti sekvencie zostávajú zachované, menia sa pomalšie
- Nefunkčné sekvencie sa vyvíjajú rýchlejším tempom
- **Príklad:** gény kódujúce proteíny, porovnanie človek myš
  - kódujúce časti: 85% zhoda (zarovnanie na 98% dĺžky)
  - intróny: 69% zhoda (zarovnanie na 48% dĺžky)
- **Úloha:** Hľadáme **nadmerne dobre zachované sekvencie**
- Veľká časť bude zodpovedať známym funkčným elementom (kódujúce gény, regulačné regióny, a pod.)
- Zachované sekvencie ktoré sa neprekrývajú so známymi funkčnými elementami — zaujímavé objekty pre výskum

## PhastCons: detekcia dobre zachovaných sekvencií

**Fylogenetické HMM:** kombinácia HMM a fylogenetického stromu.



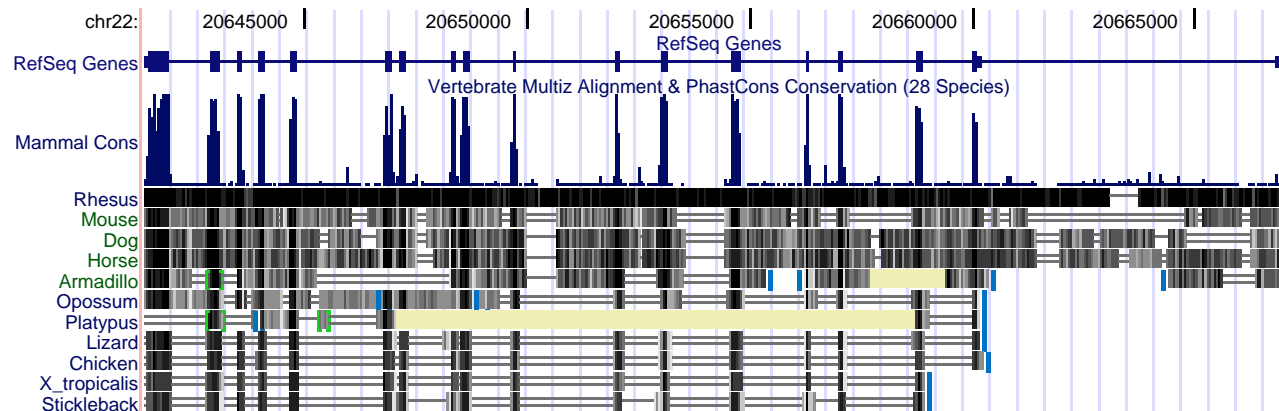
**x** = 

TCGCGAC	A	T	A	C	G	A	...				
TTGGGG	C	A	T	G	T	G	G	G	T	...	
AGCAG	A	C	G	T	C	C	G	C	A	A	...

- Dva stavy: zachovaná sekv., neutrálna sekv.
- V každom stave generujeme celý stĺpec zarovnania
- Zachovaná sekvencia má kratšie hrany stromu

## Použitie fylogenetického HMM

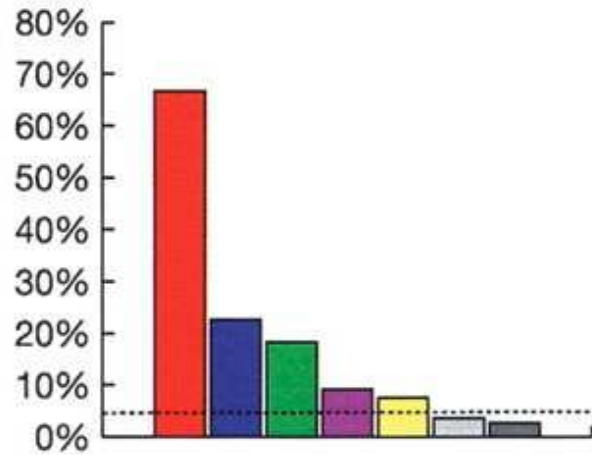
- Model určuje rozdelenie pravdepodobnosti cez zarovnanie a anotácie  
(tu: anotácia = označenie zachovaných sekvencií)
- Pre dané zarovnanie hľadáme najpravdepodobnejšiu anotáciu
- Kombinácia Viterbiho a Felsensteinovho algoritmu



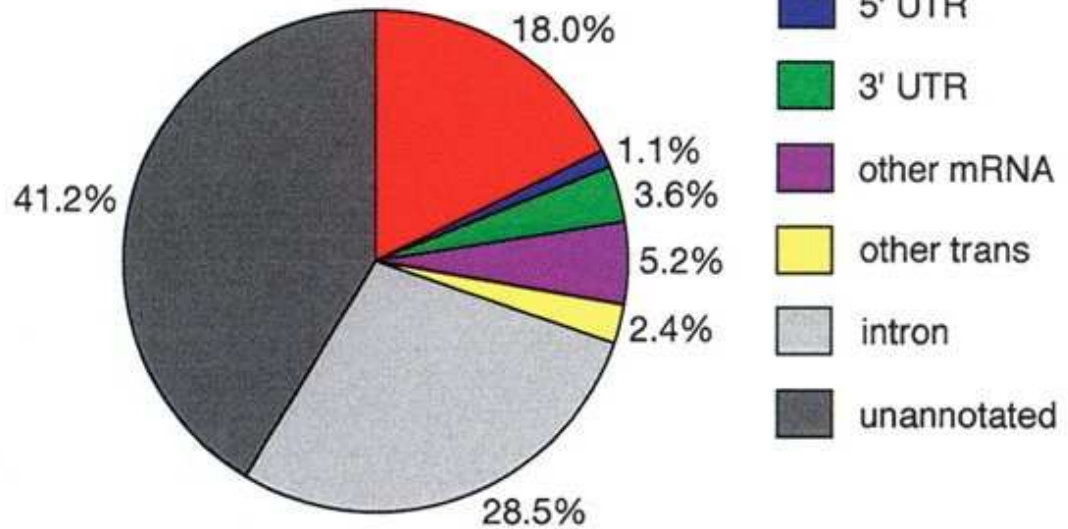
# Výsledky celogenómovej aplikácie PhastCons-u

Zarovnania genómov človeka, myši, sliepky, fugu

Coverage of Annotation Types by Conserved Elements



Composition of Conserved Elements by Annotation Type



## Fylogenetické HMM pre hľadanie génov

- Použijeme stavy z hľadača génov
- Pre každý stav máme evolučný model (maticu rýchlostí, dĺžky hrán)
- Trojperiodickosť frekvencií mutácií pomáha nájsť gény

### Ako veľmi pomôžu zarovnaná zlepšiť presnosť

Program	Exóny		Gény	
	sn	sp	sn	sp
AUGUSTUS (1 genóm)	52%	63%	24%	17%
NSCAN (zarovnanie)	68%	82%	35%	37%

Guigo et al 2006, evaluácia na 1% ľudského genómu



## Genetický kód

Ala / A	GCT, GCC, GCA, GCG	Leu / L	TTA, TTG, CTT, CTC, CTA, CTG
Arg / R	CGT, CGC, CGA, CGG, AGA, AGG	Lys / K	AAA, AAG
Asn / N	AAT, AAC	Met / M	ATG
Asp / D	GAT, GAC	Phe / F	TTT, TTC
Cys / C	TGT, TGC	Pro / P	CCT, CCC, CCA, CCG
Gln / Q	CAA, CAG	Ser / S	TCT, TCC, TCA, TCG, AGT, AGC
Glu / E	GAA, GAG	Thr / T	ACT, ACC, ACA, ACG
Gly / G	GGT, GGC, GGA, GGG	Trp / W	TGG
His / H	CAT, CAC	Tyr / Y	TAT, TAC
Ile / I	ATT, ATC, ATA	Val / V	GTT, GTC, GTA, GTG
START	ATG	STOP	TAA, TGA, TAG

## Príklad 2: Hľadanie génov pod vplyvom pozitívneho výberu

- **Pozitívny výber** = proces, ktorým sa v genóme ustália **prospešné mutácie**
- Neobvykle vysoké množstvo mutácií, ktoré by mohli súvisieť so zmenou funkcie
- V rámci génov, ktoré kódujú proteíny:
  - **Synonymné mutácie** nemenia zakódovanú aminokyselinu  
napr. ACA (Thr)  $\Rightarrow$  ACT (Thr)
  - **Nesynonymné mutácie** menia zakódovanú aminokyselinu  
napr. ACA (Thr)  $\Rightarrow$  AAA (Lys)
- Vytvoríme pravdepodobnostný model evolúcie, ktorý bude rozlišovať synonymné a nesynonymné mutácie  $\Rightarrow$  identifikácia sekvencií s neobvykle vysokým podielom nesynonymných mutácií

## Od Jukes-Cantorovho modelu ku všeobecnejším modelom mutácií

- Jukes-Cantor predpokladá, že každá mutácia rovnako pravdepodobná
- Vo všeobecnosti zavedieme  $\mu_{xy}$  – **rýchlosť substitúcie** z bázy  $x$  na bázu  $y$
- Matica rýchlostí (substitution rate matrix)

$$\begin{pmatrix} -\mu_A & \mu_{AC} & \mu_{AG} & \mu_{AT} \\ \mu_{CA} & -\mu_C & \mu_{CG} & \mu_{CT} \\ \mu_{GA} & \mu_{GC} & -\mu_G & \mu_{GT} \\ \mu_{TA} & \mu_{TC} & \mu_{TG} & -\mu_T \end{pmatrix}$$

$$\begin{pmatrix} -\mu_A & \mu_{AC} & \mu_{AG} & \mu_{AT} \\ \mu_{CA} & -\mu_C & \mu_{CG} & \mu_{CT} \\ \mu_{GA} & \mu_{GC} & -\mu_G & \mu_{GT} \\ \mu_{TA} & \mu_{TC} & \mu_{TG} & -\mu_T \end{pmatrix}$$

- **Rovnovážny stav:** frekvencie  $\pi_A, \pi_C, \pi_G, \pi_T$  nemení sa v čase
- Pre daný čas  $t$ , môžeme vypočítať pravdepodobnosť každej substitúcie (**transition probabilities**):

$$\Pr(X = C \mid Y = A, t)$$

## Znižovanie počtu parametrov — HKY matica

Hasegawa, Kishino a Yano

$$\begin{pmatrix} -\mu_A & \beta\pi_C & \alpha\pi_G & \beta\pi_T \\ \beta\pi_A & -\mu_C & \beta\pi_G & \alpha\pi_T \\ \alpha\pi_A & \beta\pi_C & -\mu_G & \beta\pi_T \\ \beta\pi_A & \alpha\pi_C & \beta\pi_G & -\mu_T \end{pmatrix} \quad \mu_{x,y} = \begin{cases} \alpha\pi_y & \text{ak } x \Leftrightarrow y \text{ je tranzícia} \\ \beta\pi_y & \text{ak } x \Leftrightarrow y \text{ je tranzverzia} \end{cases}$$

- **rýchlosť tranzícií (transition rate)**  $\alpha$ :  $C \Leftrightarrow T, A \Leftrightarrow G$
- **rýchlosť tranzverzií (transversion rate)**  $\beta$ :  $\{C, T\} \Leftrightarrow \{A, G\}$
- Máme iba štyri parametre:  $\pi_A, \pi_C, \pi_G, \kappa = \alpha/\beta$

## Substitučný model pre kodóny

Namiesto jednotlivých báz uvažujeme trojice

Rýchosť zmeny z kodónu  $i$  na kodón  $j$ :

$$\mu_{i,j} = \begin{cases} 0, & \text{ak sa } i, j \text{ líšia na } > 1 \text{ pozíciách,} \\ \alpha\pi_j, & \text{synonymné tranzície,} \\ \beta\pi_j, & \text{synonymné transverzie,} \\ \omega\alpha\pi_j, & \text{nesynonymné tranzície,} \\ \omega\beta\pi_j, & \text{nesynonymné transverzie.} \end{cases}$$

**Príklad:**  $\mu_{AAC,GGC} = 0$ ,  $\mu_{CTA,CTT} = \beta\pi_{CTT}$ ,

$\mu_{CTA,CCA} = \omega\alpha\pi_{CCA}$

**Parametre:** Frekvencie kodónov  $\pi_j$ ,  $\omega$ ,  $\kappa = \alpha/\beta$

**Prirodzený výber:** neutrálna evolúcia  $\omega = 1$ , pozitívny výber  $\omega > 1$ ,  
purifikačný výber  $\omega < 1$

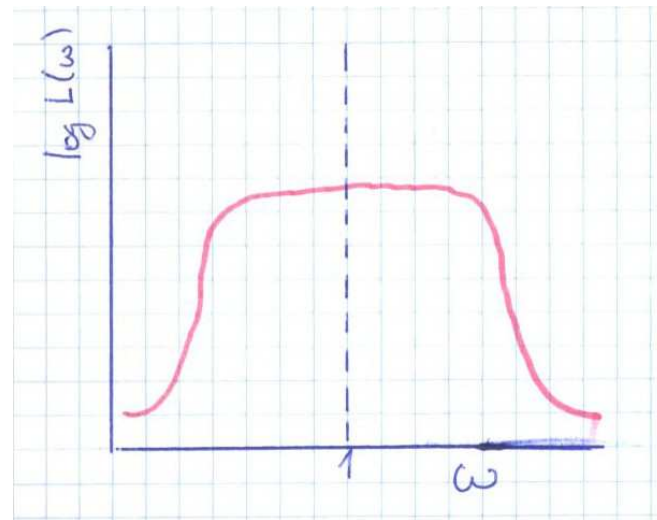
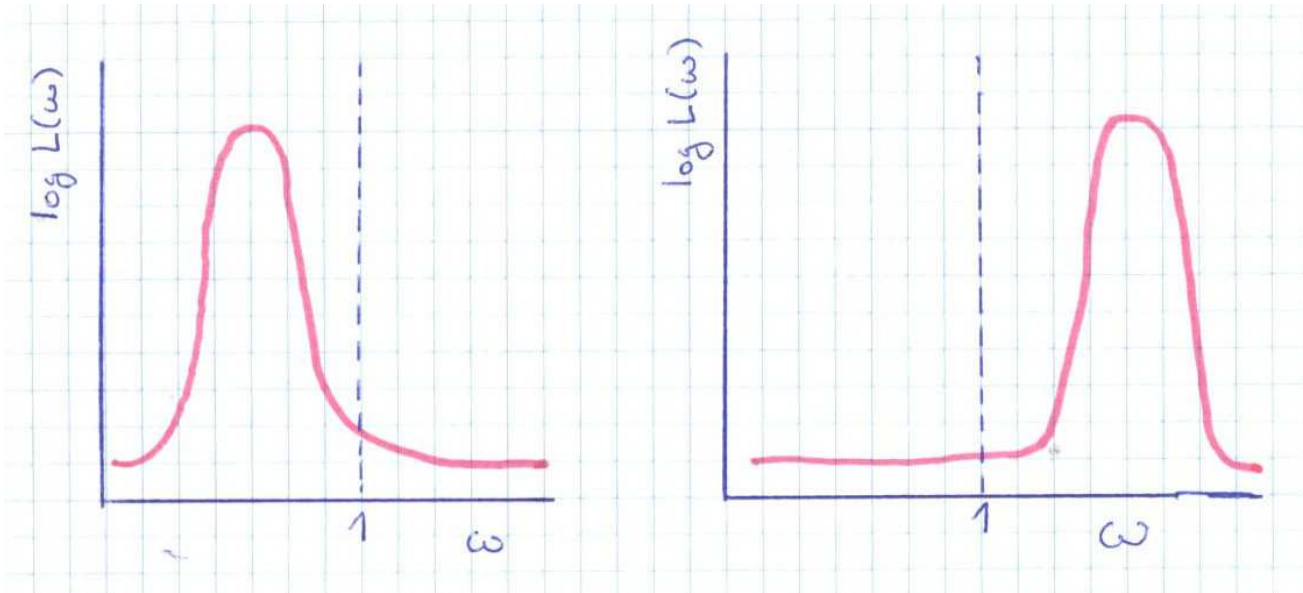
## Aplikácia kodónového substitučního modelu

	F	V	I	H	D	S	E	G	D	G	E	C	M	Q	E
človek	TTT	GTG	ATC	CAC	GAC	TCC	GAG	GGG	GAC	GGC	GAG	TGC	ATG	CAG	GAG
kosmák	TTT	GTG	ATC	CAC	GAG	AAC	AAC	AAG	GAC	GGC	GAG	TGC	ATG	CAG	GAT
	F	V	I	H	E	N	N	K	D	G	E	C	M	Q	D

- Na základe celých genómov môžeme odhadnúť základné parametre modelu  $\pi_A, \pi_C, \pi_G, \pi_T, \kappa$
- Pre dané  $\omega$  a  $t$  vieme spočítať vierohodnosť

$$L(\omega, t) = \Pr(C, K \mid \omega, t)$$

- Sledujeme, ako sa mení  $L(\omega) = \max_t L(\omega, t)$  pre rôzne hodnoty  $\omega$





## Test pomerov vierohodností (Likelihood-ratio test)

- $L(\omega)$  môže byť najväčšie pre  $\omega > 1$ ,  
ale môže to byť spôsobené len štatistickou variáciou v dátach  
 $\Rightarrow$  potrebujeme štatistický test
- Spočítame vierohodnosť  $L_A = \max_{\omega < 1} L(\omega)$
- Spočítame vierohodnosť  $L_B = \max_{\omega} L(\omega)$  (bez obmedzenia  $\omega$ )
- Vždy platí  $L_B \geq L_A$
- Ak skutočné  $\omega < 1$ ,  $L_A \approx L_B$  (nulová hypotéza)  
nás zaujímajú prípady  $L_B \gg L_A$   
 $\Rightarrow$  gén pod vplyvom pozitívneho výberu (alt. hypotéza)

Za predpokladu, že  $\omega < 1$ , platí  $2 \log(L_B/L_A) \approx \chi_1^2$   
 $\Rightarrow$  možno priradiť P-hodnotu nulovej hypotéze  $\omega < 1$

## Hľadanie génov pod vplyvom pozitívneho výberu: Zhrnutie

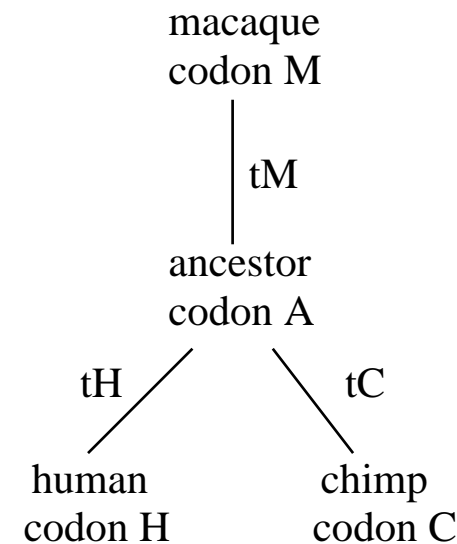
- Nájdem zariadenie toho istého génu z dvoch organizmov (na úrovni kodónov)
- Odhadneme základné parametre kodónového modelu na základe porovnania celých genómov
- Parameter  $\omega$  modeluje selekciu
- Spočítame vierohodnosť  $L_A = \max_{\omega < 1} L(\omega)$   
a vierohodnosť  $L_B = \max_{\omega} L(\omega)$
- Na základe štatistiky  $2 \log(L_B/L_A)$  priradíme P-hodnotu nulovej hypotéze  $\omega < 1$
- Gény s malou P-hodnotou sú pod vplyvom pozitívneho výberu

## “Jednoducho” rozšíriteľné na porovnanie viacerých organizmov

$$\Pr(A, H, C, M | \omega, t_H, t_C, t_M) = \pi_A \cdot \Pr(H | A, t_H) \cdot \Pr(C | A, t_C) \cdot \Pr(M | A, t_M)$$

Zbavíme sa ancestrálnych sekvencií:

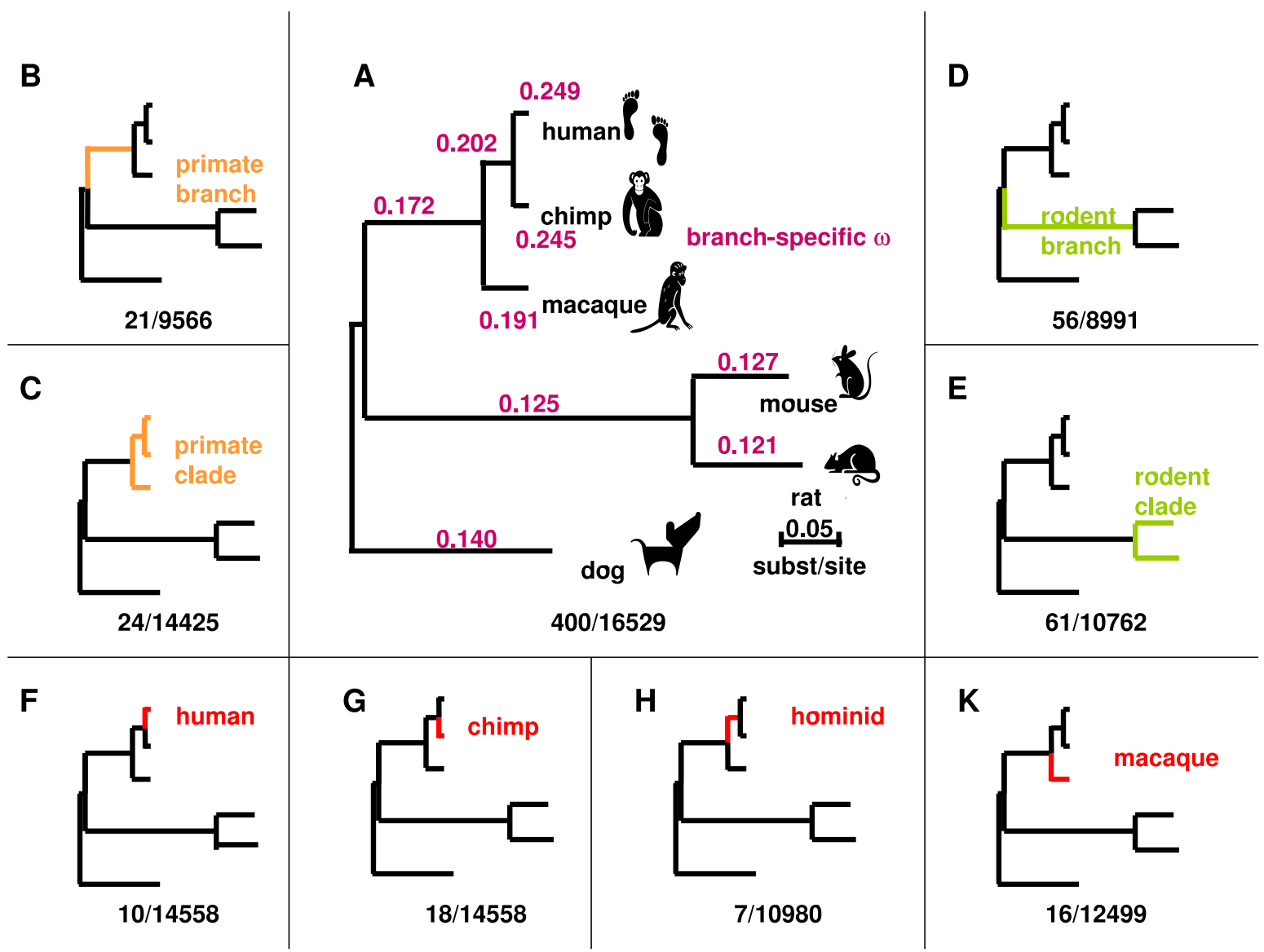
$$\Pr(H, C, M | \omega, t_H, t_C, t_M) = \sum_A \Pr(A, H, C, M | \omega, t_H, t_C, t_M)$$



### Vierohodnosť $\omega$ :

$$L(\omega) = \max_{t_H, t_C, t_M} \Pr(H, C, M | \omega, t_H, t_C, t_M)$$

- Existuje program PAML, ktorý takúto vierohodnosť počíta
- K dispozícii zložitejšie modely, napr. s meniacim sa  $\omega$  v rámci génu



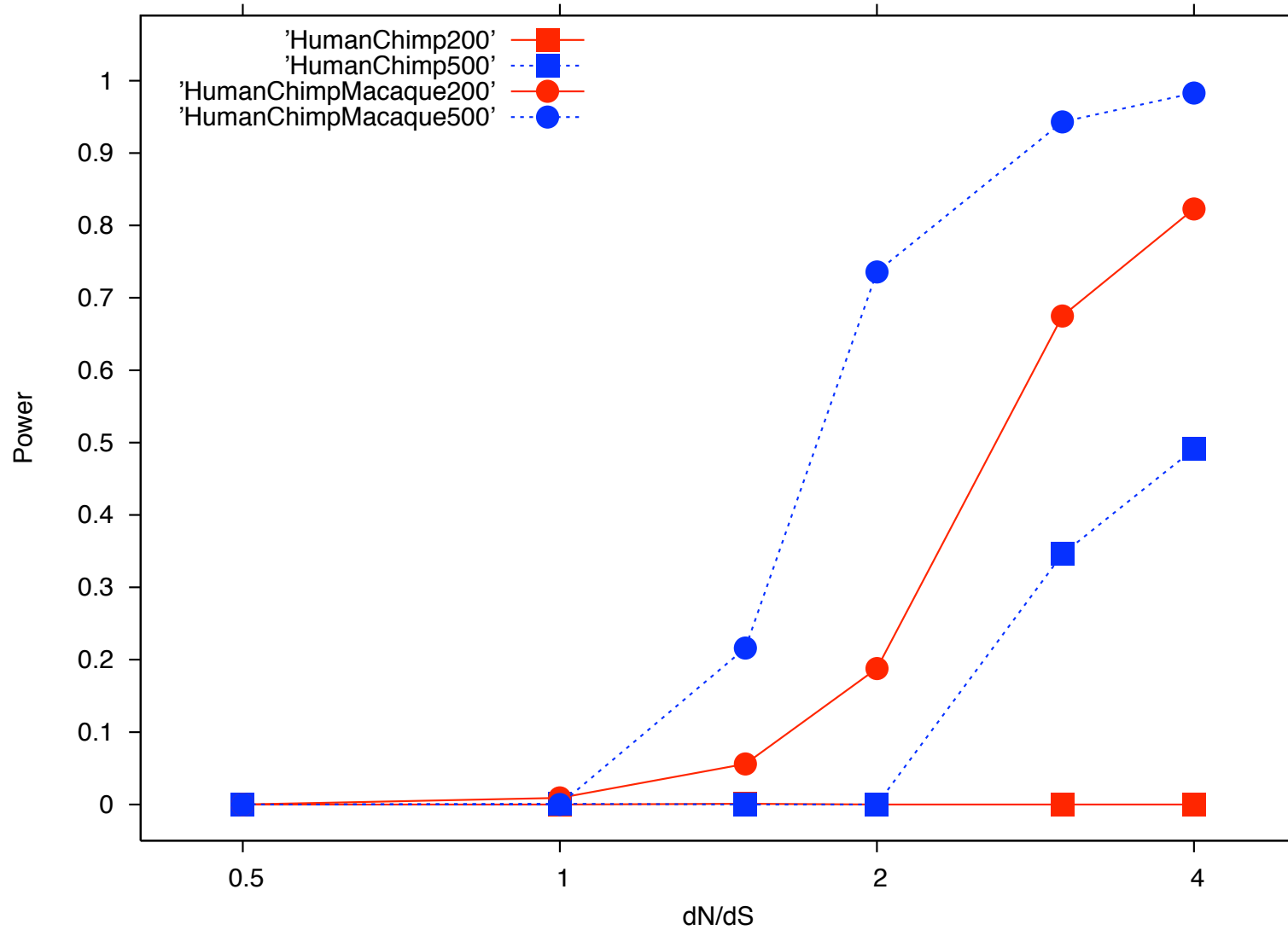
## Funkčné kategórie obohatené o gény s pozitívnym výberom

**Defense:** cellular defense response, antigen processing and presentation, response to virus, response to bacterium

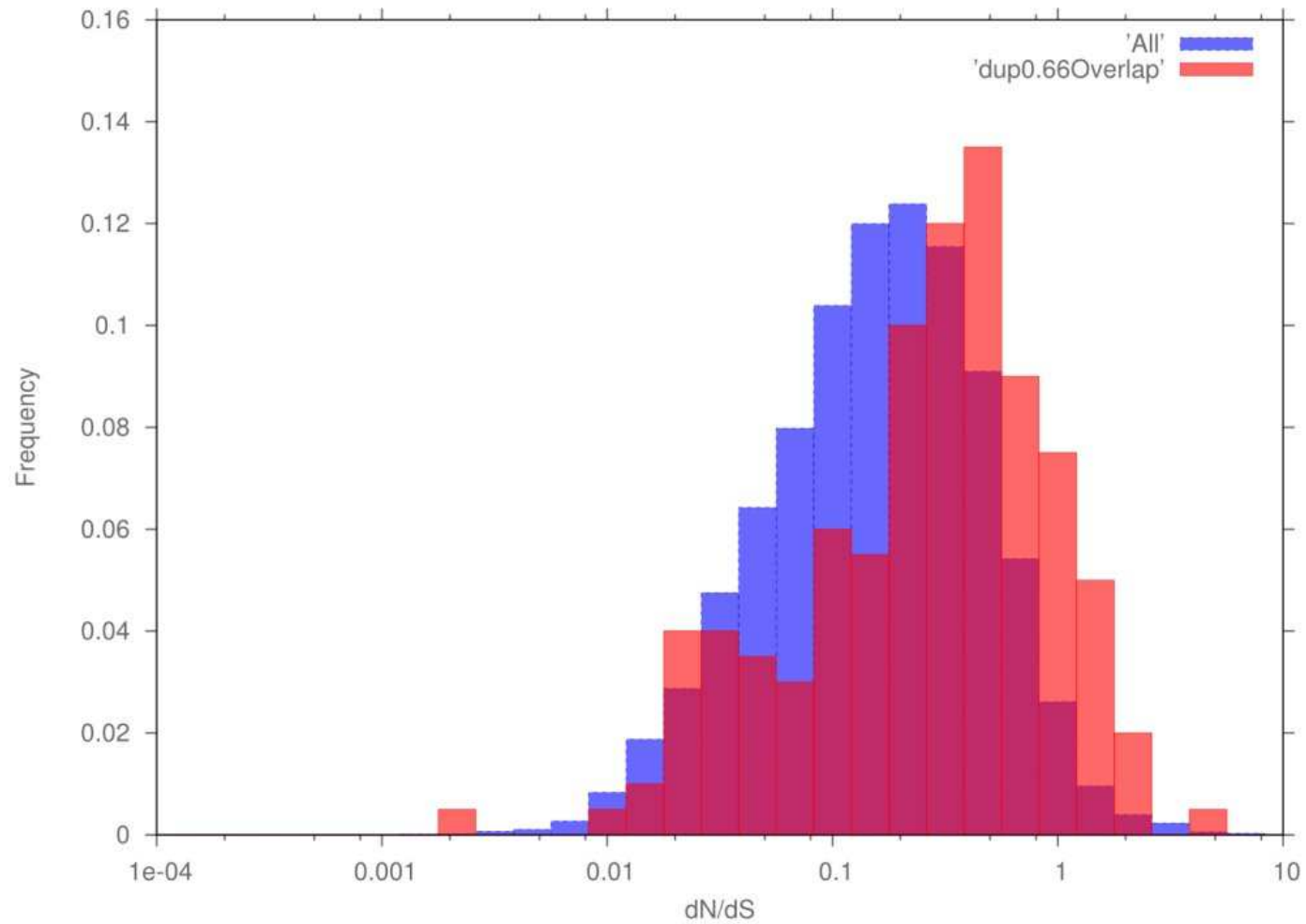
**Immunity:** adaptive immune response, adaptive immune response somatic recomb, lymphocyte mediated immunity, immunoglobulin mediated immune response, B cell mediated immunity, innate immune response, complement activation alternative pathway, regulation of immune system process, positive regulation of immune response, humoral immune response, complement activation classical pathway, humoral immune response circulating immunoglob, complement activation, activation of plasma proteins mute inflam resp, akute inflammatory response, response to wounding

**Sensory perception:** sensory perception of taste, G-protein coupled receptor protein signaling pathway, neurological process, sensory perception of chemical stimulus, sensory perception of smell

## Viacej genómov pomáha vylepšiť účinnosť testov



## Pozitívny výber v duplikovaných génoch



## Zhrnutie

- Prirodzený výber má významnú úlohu v evolúcii
- **Purifikačný výber:**
  - Zachované regióny majú s veľkou pravdepodobnosťou nejakú funkciu
  - Pri hľadaní génov berieme do úvahy aj typické mutácie kodónov
- **Pozitívny výber:**
  - Pozitívny výber v génoch sa prejavuje veľkým pomerom nesynonymných zmien (evolúcia na proteínovej úrovni)
  - Zduplikované gény sú častejšie pod vplyvom pozitívneho výberu
  - Poľovačka pokračuje: hľadáme gény spôsobujúce charakteristické črty človeka
- **Metódy:** evolučné modely, fylogenetické HMM, test pomerov vierohodností



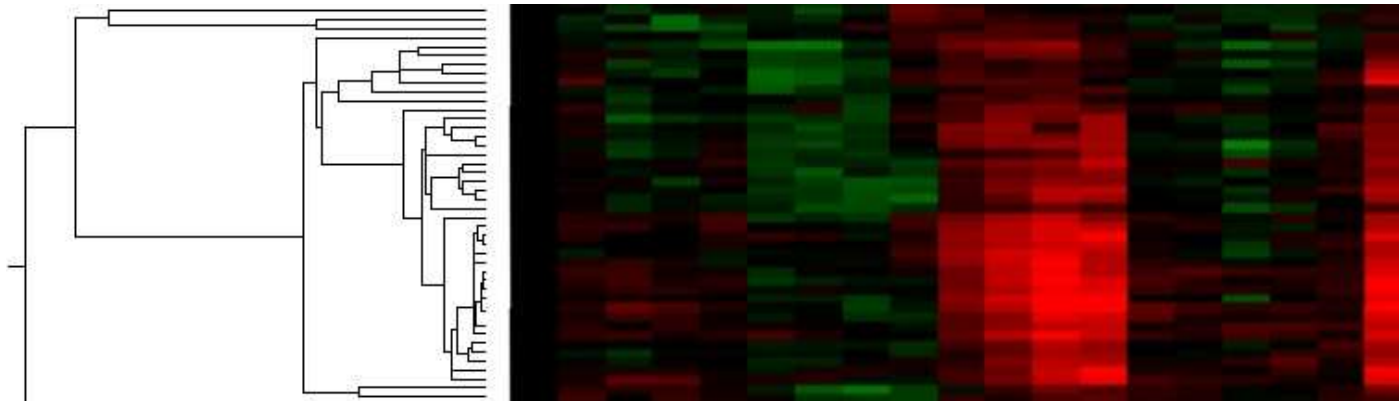
## Oznamy

- DÚ 2 na stránke, odovzdať do 6.12.  
(na začiatku cvičení, resp. prednášky)  
Informatici zdrojový kód odovzdávajú v systéme Moodle
- Ďakujem za správy o stretnutiach journal clubu,  
dajte vedieť, ak potrebujete konzultácie

# Regulácia génovej expresie

Broňa Brejová

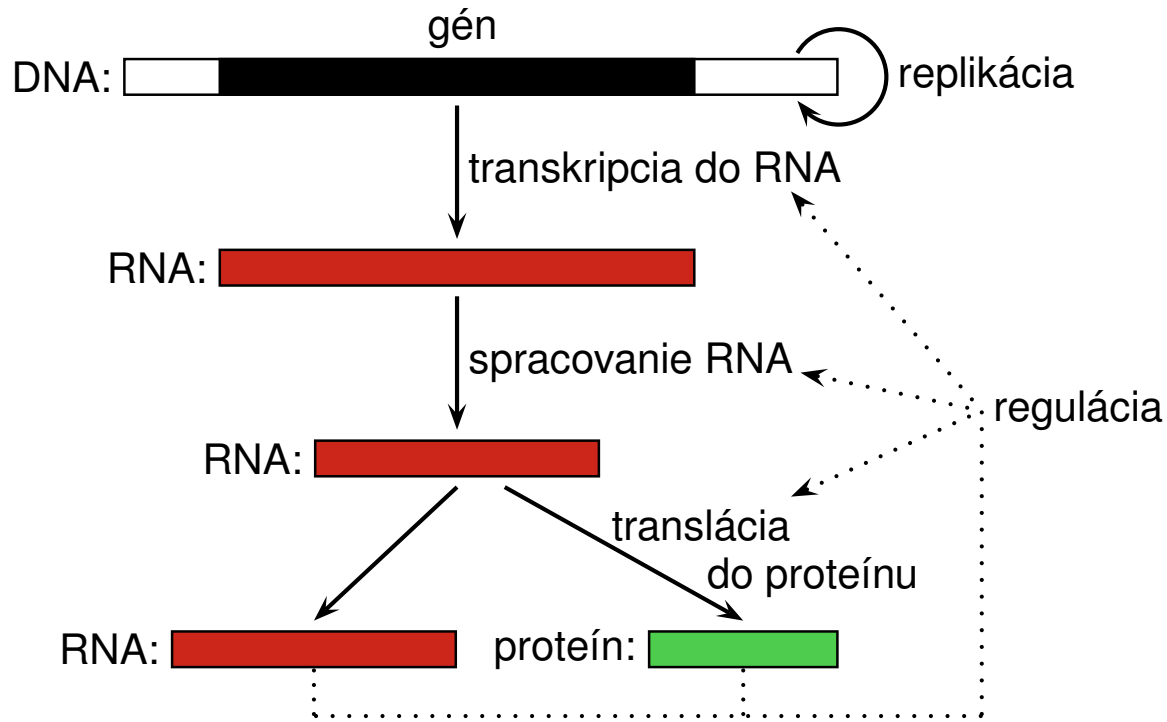
22.11.2018



## Aká informácia je uložená v DNA?

**Gény:** Predpisy na tvorbu proteínov a funkčných RNA molekúl.

**Riadenie ich expresie:** kedy a koľko sa má tvoriť.

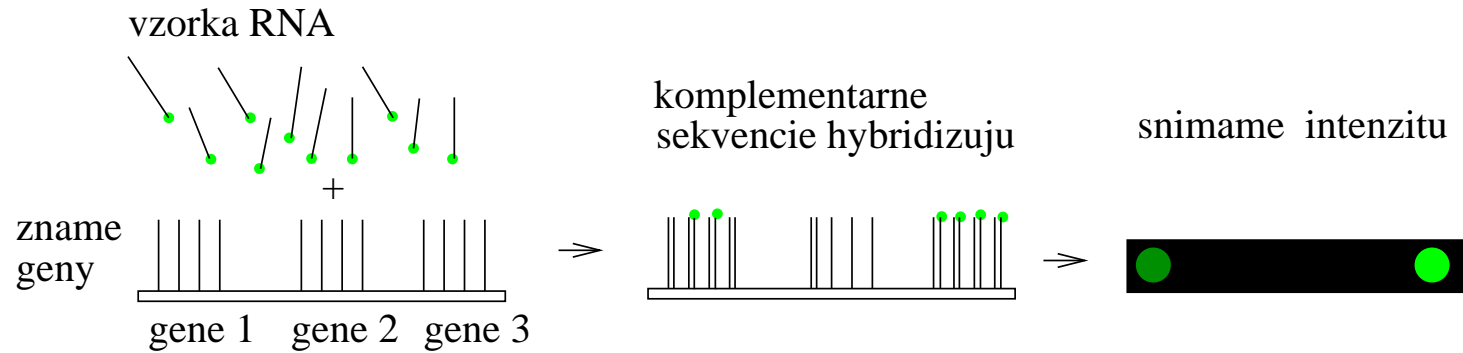


Regulácia na úrovni transkripcie, spracovania, translácie, posttranslačných modifikácií, ...

## Ciele

- Zistiť, za akých podmienok je daný gén exprimovaný (súvisí s funkciou génu)
- Ktoré gény ho regulujú
- Detaily regulačného mechanizmu (väzobné miesta, zmeny v množstve expresie, ...)

## Technológia: expression array, microarray



Meranie množstva mRNA prítomnej v bunke pre **veľa génov** naraz.  
Zopakujeme za rôznych podmienok.

# Technológia: RNA-seq

sekvenujeme RNA extrahovanú z bunky NGS technológiami, mapujeme na genóm, hĺbka pokrytia zodpovedá úrovni expresie



## Príklad microarray dát

Pomer expresie génu v meranej a kontrolnej vzorke fg/bg

	15min	30min	1hod	2hod	4hod	...
W95909	0.72	0.1	0.57	1.08	0.66	
AA045003	1.58	1.05	1.15	1.22	0.54	
AA044605	1.1	0.97	1	0.9	0.67	
W88572	0.97	1	0.85	0.84	0.72	
AA029909	1.21	1.29	1.08	0.89	0.88	
AA059077	1.45	1.44	1.12	1.1	1.15	

...

Iyer et al 1999 The Transcriptional Program in the Response of Human Fibroblasts to Serum

Fibroblast: bunky generujúce zložky medzibunkovej hmoty

pre delenie potrebujú rastové faktory dodávané ako "fetal bovine serum"

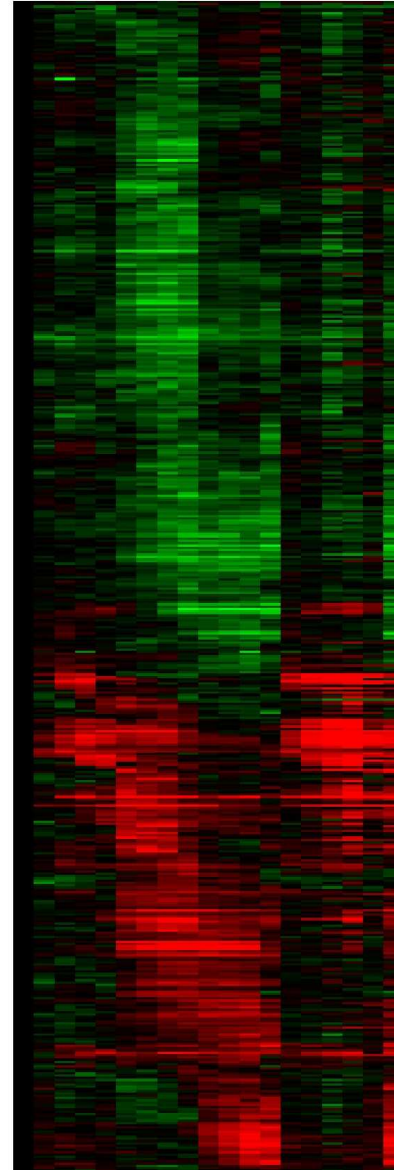
## Vizualizácia

Červená:  $fg > bg$

Zelená:  $fg < bg$

517 génov (z 8600)

19 experimentov





## Dnes: iný typ dát

- tabuľka čísel
- typické dáta v štatistike
- možno použiť všeobecné metódy štatistiky, strojového učenia

## Všetky ostatné prednášky: pracujeme so sekvenciami

- zostavovanie genómov
- zarovnávanie sekvencií
- hľadanie génov
- fylogenetické stromy, populačná a komparatívna genomika
- štruktúra a funkcia proteínov a RNA

## Prvá sada problémov: predspracovanie dát

- Zo scanovaných obrázkov určiť intenzitu, odhaliť zlé merania
- Agregácia dát z viacerých meraní pre jeden gén
- Použitie kontrolných meraní
- Normalizácia, aby sme mali porovnateľné výsledky z rôznych experimentov

Merania expresie nie veľmi presné, veľa šumu, rôzne zdroje chýb

### Jednoduchý výsledok:

zoznam výrazne podexprimovaných/nadexprimovaných génov

napr.  $fg/bg > 2$ , resp.  $fg/bg < 0.5$

často na ďalšiu analýzu používame iba tieto

## Zhlukovanie (clustering)

**Ciel:** nájsť skupiny génov s podobným profilom expresie.

Ak veľa génov v skupine má rovnakú funkciu,  
ďalšie gény asi robia to isté

**Meranie podobnosti profilov:** napr. Pearsonov korelačný koeficient

Profil génu 1:  $x_1, x_2, \dots, x_n$ , priemer  $\bar{x}$

Profil génu 2:  $y_1, y_2, \dots, y_n$ , priemer  $\bar{y}$

$$C(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

Číslo od -1 do 1, 1 pre lineárne korelované dáta

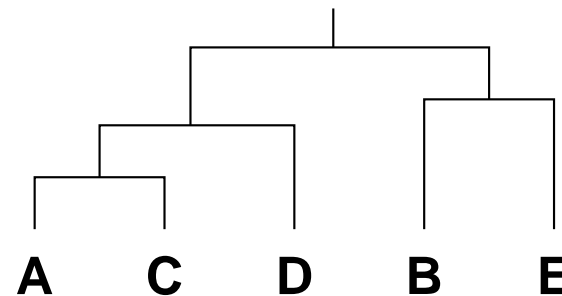
Vzdialenosť  $d(x, y) = 1 - C(x, y)$

Aj iné možnosti, napr. Euklidovská vzdialenosť

## Hierarchické zhlukovanie

- Podobné na metódu spájania susedov vo fylogenetických stromoch
- Začneme s každým génom v samostatnej skupinke
- Nájdeme dve najbližšie skupinky a spojíme ich do jednej
- Opakujeme, kým nie sú všetky gény spolu
- Vzdialenosť skupiniek: napr. vzdialenosť najbližších génov z jednej a druhej, alebo priemer vzdialeností cez všetky páry
- Výsledkom je strom zobrazujúci postupnosť spájania

	A	B	C	D	E
gén A	0	0.6	0.1	0.3	0.7
gén B	0.6	0	0.5	0.5	0.4
gén C	0.1	0.5	0	0.6	0.6
gén D	0.3	0.5	0.6	0	0.8
gén E	0.7	0.4	0.6	0.8	0



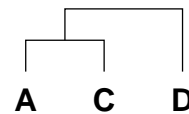
## Hierarchické zhlukovanie - príklad

Vzdialenosť skupiniek ako vzdialenosť najbližších génov z jednej a druhej (single linkage clustering)

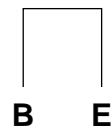
	A	B	C	D	E
gén A	0	0.6	0.1	0.3	0.7
gén B	0.6	0	0.5	0.5	0.4
gén C	0.1	0.5	0	0.6	0.6
gén D	0.3	0.5	0.6	0	0.8
gén E	0.7	0.4	0.6	0.8	0



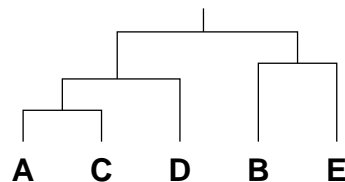
	A+C	B	D	E
A+C	0	0.5	0.3	0.6
B	0.5	0	0.5	0.4
D	0.3	0.5	0	0.8
E	0.6	0.4	0.8	0



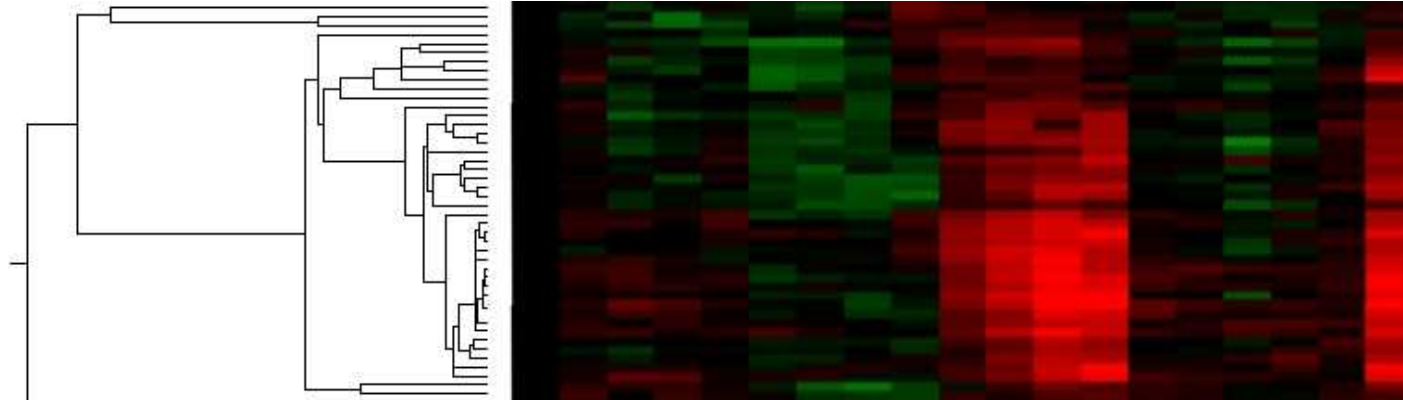
	A+C+D	B	E
A+C+D	0	0.5	0.6
B	0.5	0	0.4
E	0.6	0.4	0



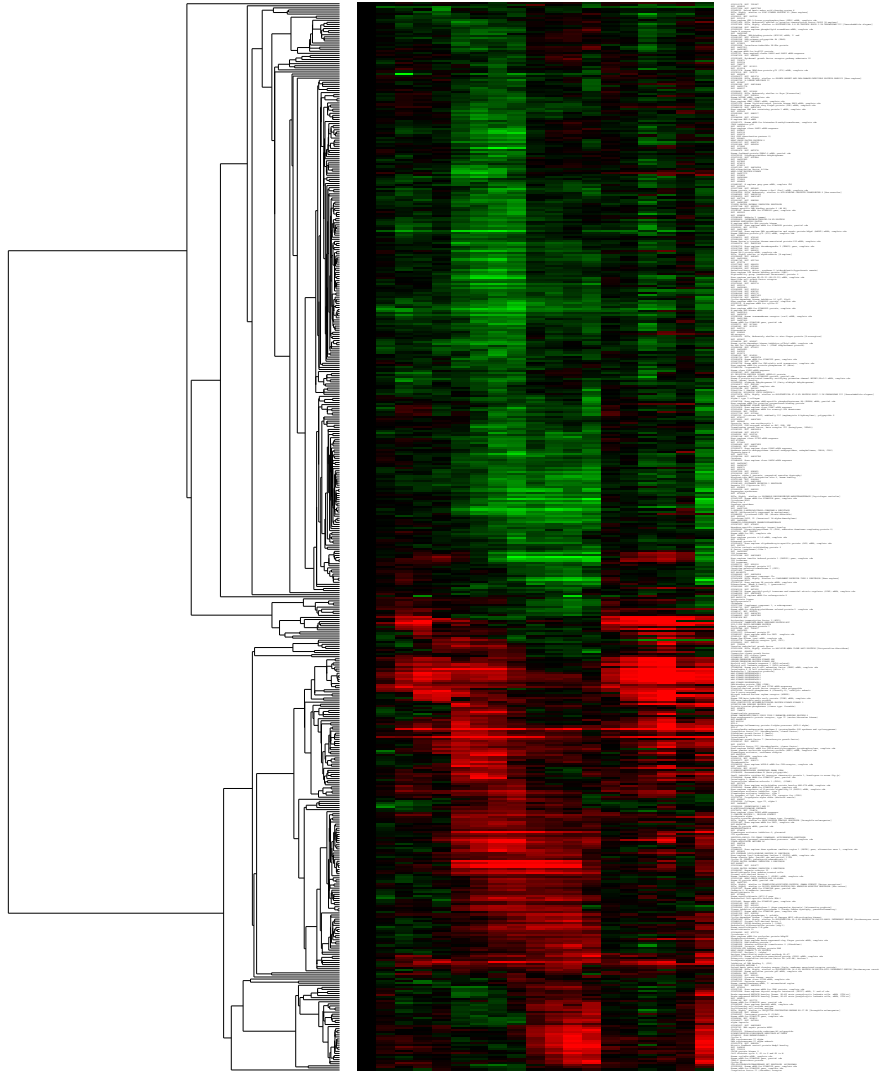
	A+C+D	B+E
A+C+D	0	0.5
B+E	0.5	0



## Príklad: časť mikroarray dát



Zhlukovanie tiež pomáha vizualizácii dát,  
podobné gény sa dostanú ku sebe



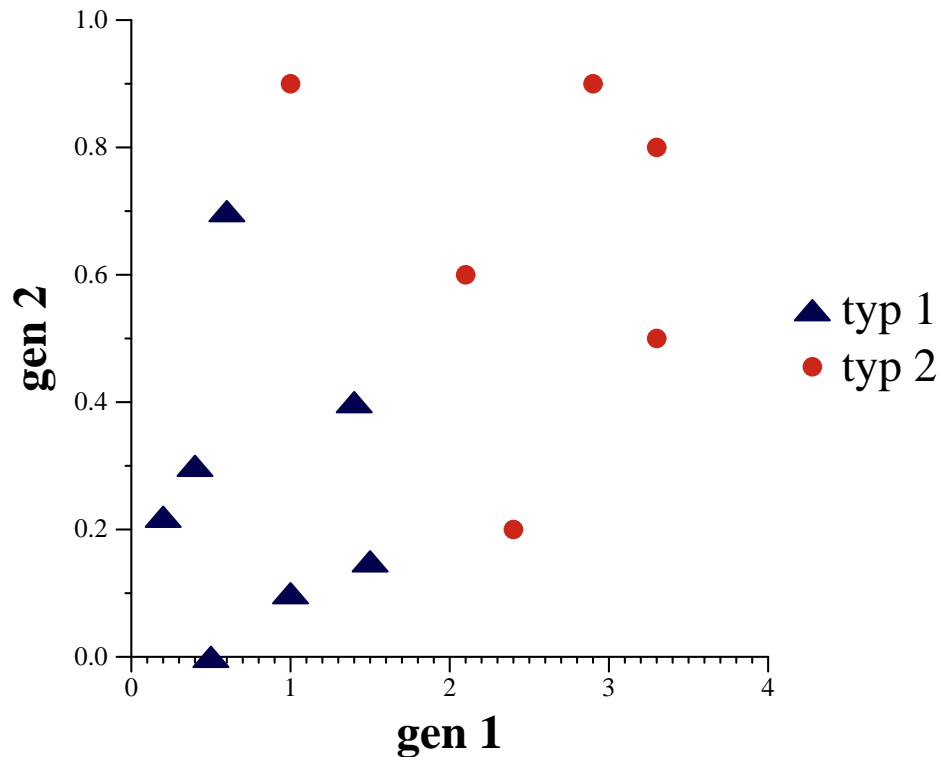
## Klasifikácia

- Typický problém v strojovom učení
- Chceme odlíšiť napr. rôzne typy tumorov podľa expresie génov
- Máme nejaké príklady, kde vieme expresiu aj typ tumoru
- Chceme napr. nájsť vzorec, ktorý nám z expresie vyráta záporné číslo pre typ 1, kladné číslo pre typ 2.
- Vopred si vyberieme si typ vzorca s neznámymi parametrami (trieda hypotéz)
- Na tréningových dátach hľadáme hodnoty parametrov, pre ktoré vzorec najlepšie funguje
- Fungovanie vzorca testujeme na testovacích dátach (nepoužité na tréning)
- Hotový vzorec použijeme na dáta s neznámym typom



## Jednoduchý príklad: expresia 2 génov

Trénovacie dáta so známym typom:



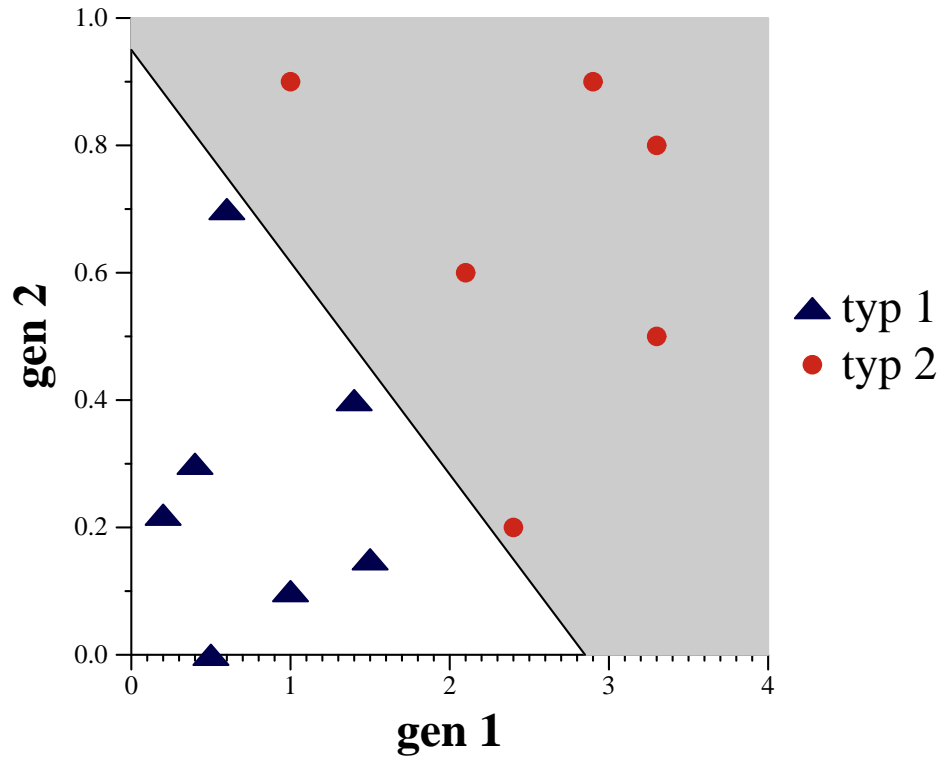
Typ vzorca: lineárne funkcie (lineárny diskriminant)

tumor typu 1 ak  $ax + by + c < 0$

Hľadáme  $a, b, c$  také, aby na trénovacích dátach predpovedal dobre

## Jednoduchý příklad: expresia 2 génov

### Výsledný vzorec:



$$a = 1, b = 3, c = -2.85$$

tumor typu 1 ak  $x + 3y - 2.85 < 0$

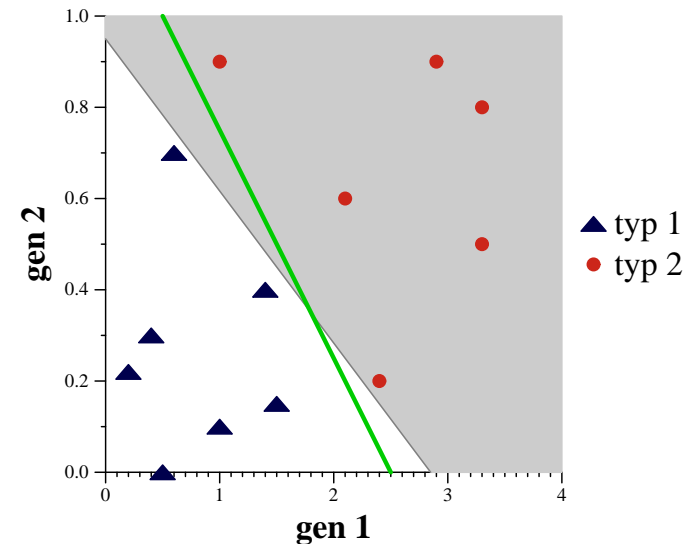
## Populárne techniky na klasifikáciu

### Logistic regression, logistická regresia:

lineárny diskriminátor, vracia pravdepodobnosť jednotlivých tried, dobre známa štatistická metóda.

### Support vector machines

**(SVM):** hľadanie lineárneho diskriminátora s nulovou tréningovou chybou, ktorý je najďalej od všetkých tréningových dát.



Dá sa zovšeobecniť na nelineárne funkcie priemetom vektorov do väčšieho priestoru.

## Populárne techniky na klasifikáciu

### Neurónové siete:

“neuróny” poprepájané “synapsami”,  
každý neurón na výstupe váhovaný priemer vstupov.

### Bayesovské siete:

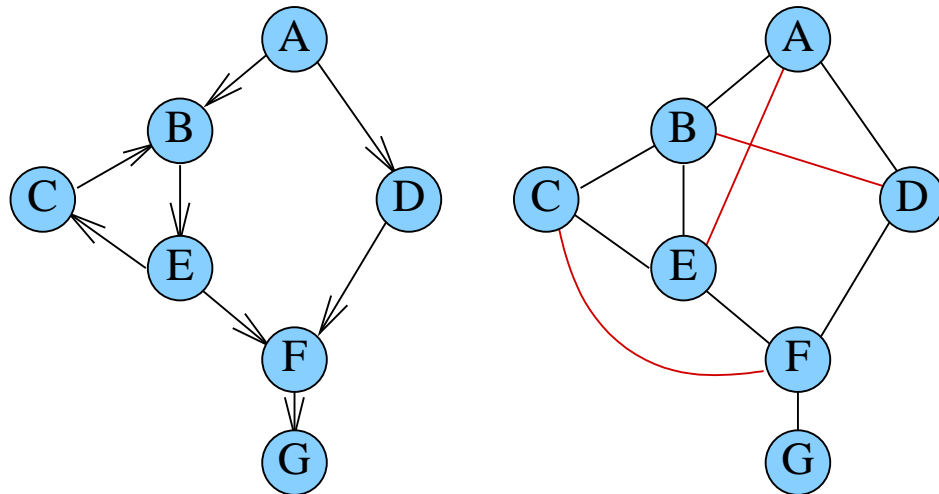
pravdepodobnostný model generujúci náhodné expresie  
typ tumoru je tiež náhodná premenná, ktorej hodnotu nepoznáme  
podobne ako stav v HMM

## Regulačné siete z profilov expresie

**Vstup:** Profily expresie génov (napr. séria microarray/RNA-seq experimentov),  
možno so známymi podmienkami (časové rady, delečný mutant)

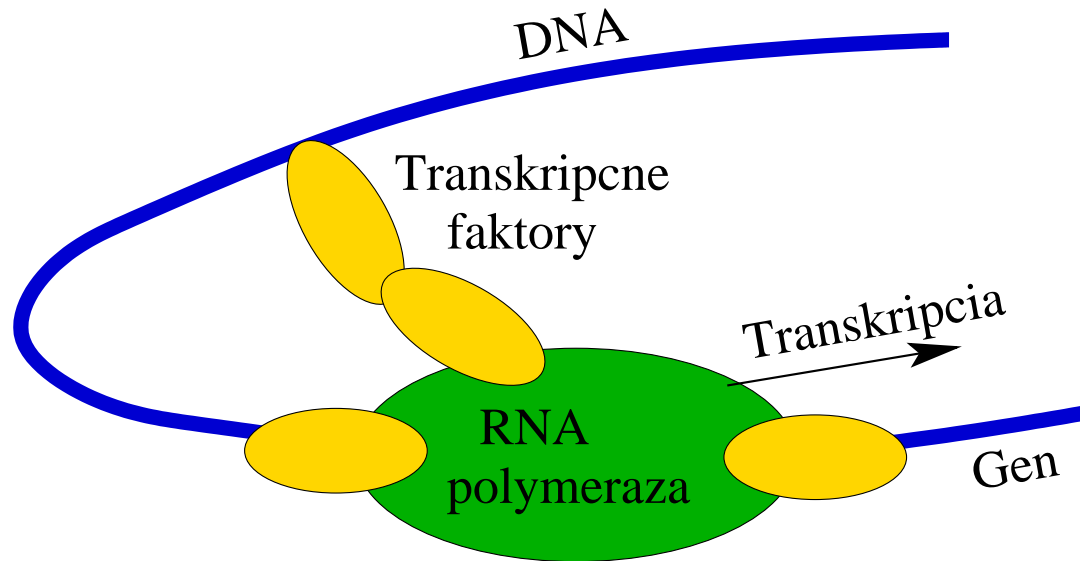
**Výstup:** regulačná sieť, vrcholy sú gény,  
orientovaná hrana  $A \rightarrow B$ , ak  $A$  reguluje  $B$

Podobnosť profilov expresie nám môže dať neorientované hrany.  
Chceme vylúčiť hrany, ktoré vznikli tranzitivitou  
a správne orientovať hrany (ťažký problém)



## Transkripčné faktory (TF)

Regulácia začatia transkripcie pomocou transkripčných faktorov: proteíny viažúce DNA, pomáhajú pritiahnúť RNA polymerázu

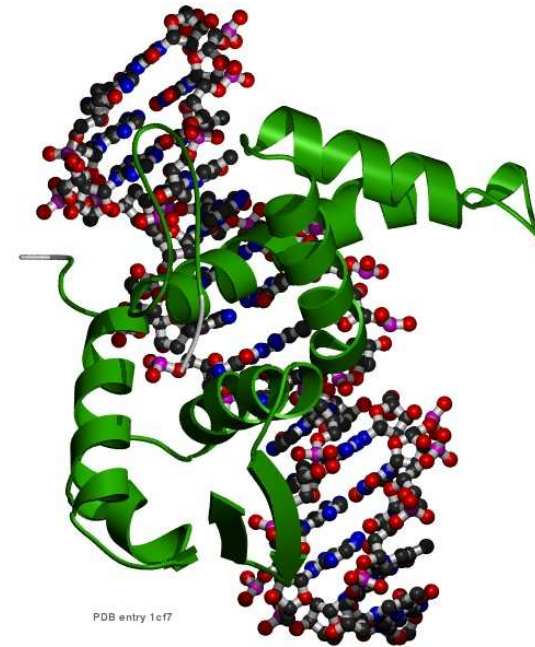
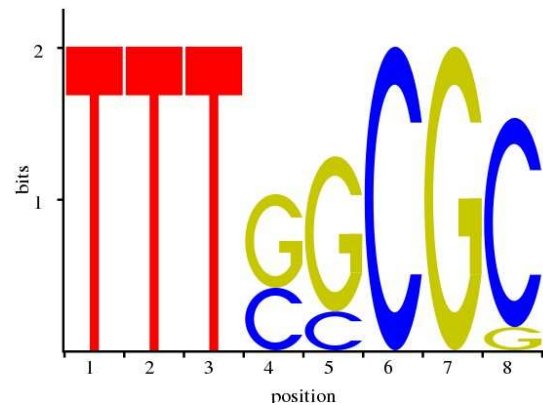


Človek má vyše 2000 TF-ov

Môžu zvyšovať alebo znižovať mieru expresie,  
fungovať v skupinách

## Príklad: transkripčný faktor E2F1

- Reguluje bunkový cyklus
- Viaže TTTC<sup>CC</sup>GC alebo TTTC<sup>CG</sup>GC, prípadne ďalšie varianty



- Sekvencie DNA, na ktoré sa viaže určitý TF chceme **reprezentovať** ako sekvenčný **motív** a hľadať **ďalšie výskyty** v genóme

## Reprezentácia väzobných motívov

### Reťazec s nezhodami (konsenzus):

motív je reťazec, výskyty môžu mať vopred ohraničený počet nezhôd

**Príklad:** motív TTTGGCGC + 1 nezhoda

TTTGGCGC, TT**A**GGCGC, TTTG**C**CGC sú výskyty motívu

TTT**C**CGC nie je výskyt

**Zostavenie motívu:** napr. vezmi najčastejšie písmeno na každej pozícii

A	0	0	0	0	0	0	0	0
C	0	0	0	4	2	10	0	9
G	0	0	0	6	8	0	10	1
T	10	10	10	0	0	0	0	0



## Reprezentácia väzobných motívov 2

### Regulárny výraz:

niektoré pozície motívu dovoľujú výber z viacej možností

[GC] znamená pozíciu, na ktorej môže byť G alebo C

N znamená hociktorú bázu

**Príklad:** motív TTT[CG][CG]CGC

TTTGGCGC, TTT**CC**CGC, TTTG**C**CGC sú výskyty motívu

TT**A**GGCGC nie je výskyt

**Zostavenie motívu:** povol' najčastejšie bázy na každej pozícii

A	0	0	0	0	0	0	0	0
C	0	0	0	4	2	10	0	9
G	0	0	0	6	8	0	10	1
T	10	10	10	0	0	0	0	0

## Reprezentácia väzobných motívov 3

### Position specific scoring matrix (PSSM, PWM):

skórovacia matica, skóre pre každú bázu na každej pozícii

Výskyty dosahujú skóre väčšie ako číslo  $T$

**Príklad:**  $T = 8$

A	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0
C	-1.6	-1.6	-1.6	0.6	0.0	1.5	-1.6	1.4
G	-1.6	-1.6	-1.6	1.0	1.3	-1.6	1.5	-0.5
T	1.1	1.1	1.1	-2.0	-2.0	-2.0	-2.0	-2.0

TTT**CC**CGC je výskyt:  $1.1+1.1+1.1+0.6+0.0+1.5+1.5+1.4=8.3$

TTTGG**C**GG je výskyt:  $1.1+1.1+1.1+1.0+1.3+1.5+1.5-0.5=8.1$

TT**A**GGCGC nie je:  $1.1+1.1-2.0+1.0+1.3+1.5+1.5+1.4=6.4$

Zostavenie matice z frekvencií: budúca prednáška

## Hľadanie výskytov v genóme

- Hľadanie motívu v genóme: skús každú pozíciu, či je výskytom
- Väčšinou veľa falošných výskytov
- Vieme spočítať E-hodnotu: koľko výskytov očakávame v náhodnej sekvencii
- Napr. TTT[CG][CG]CGC sa vyskytuje v priemere raz za 30 000 báz
- Na zlepšenie špecificity hľadáme
  - zhluky väzobných miest,
  - miesta podporené experimentálne,
  - evolučne zachované
- Databázy motívov, napr. TRANSFAC, JASPAR

## Ako nájsť väzobné miesta experimentálne?

### Chromatin immunoprecipitation (ChIP)

Pomocou protilátky (antibody) na špecifický transkripčný faktor zistí, kde približne sa tento faktor viaže.

- Väzba medzi TF a DNA sa spevní formaldehydom
- DNA sa naseká na kusy
- Kusy, na ktorých je TF, sa zachytia na protilátke
- DNA sa izoluje a sekvenuje pomocou NGS (**ChIP-seq**) alebo deteguje pomocou microarray (**ChIP-chip**)

**Problém:** zistíme len približnú polohu väzobného miesta

## Ako nájsť motívy výpočtovými metódami?

... ak nemáme niekoľko príkladov väzobného miesta

- Máme skupinu sekvencií, kde každá obsahuje väzobné miesto toho istého TF, ale väzobné preferencie TF nie sú známe
- Snažíme sa nájsť **čo najšpecifickejší** motív, ktorý sa vyskytuje vo všetkých týchto sekvenciách resp. sa vyskytuje častejšie, ako by sme očakávali.
- **Pôvodne:** zoberieme skupinu génov s podobným profilom expresie a teda možno regulovaných tým istým TF, hľadáme motív v oblastiach pred týmito génmi
- **V súčasnosti:** zoberieme oblasti detegované pomocou ChIP-seq okolo väzobných miest, nájdený motív použijeme na presnejšie určenie polohy väzby TF

## Príklad: Consensus Pattern Problem (CPP)

Jednoduchá formulácia problému hľadania motívov

**Vstup:** dĺžka motívu  $L$ , reťazce (sekvencie)  $S_1, S_2, \dots, S_k$

**Výstup:** motív (reťazec)  $M$  dĺžky  $L$

a výskyt motívu v každom  $S_i$  (reťazec  $s_i$  dĺžky  $L$ )

také, že celkový počet nezhôd medzi  $M$  a  $s_i$  je najmenší možný

### Príklad:

Vstup: CAAACAT, AGTAGC, TAACCA, TCTCCTC,  $L = 4$

Výstup: motív TAAC

výskyty a nezhody AAAC 1, TAGC 1, TAAC 0, TCTC 2

celkový počet nezhôd 4

## Riešenie CPP

NP-ťažký problém

- **Idea 1:** Vyskúšaj všetky možné motívy dĺžky  $L$

**Problém:** Nepraktické — prečo?

- **Idea 2:** Vyskúšaj všetky možné podreťazce dĺžky  $L$  reťazcov  $S_1, \dots, S_k$

**Problém:** Nemusí fungovať — prečo?

Ale dá sa dokázať, že cena riešenia bude najviac dvojnásobok optima (2-aproximačný algoritmus)

- **Ďalšie vylepšenie:** Skúšame všetky konsenzus sekvencie  $\ell$  podreťazcov. PTAS (polynomial-time approximation scheme)

## Praktickejší prístup k hľadaniu motívov

**Pravdepodobnostný model** generujúci sekvenciu  $S$  pomocou matice frekvencií báz v motíve  $W$  a frekvencie báz  $q$  mimo motívu

A	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
C	0.01	0.01	0.01	0.39	0.19	0.97	0.01	0.01	0.89
G	0.01	0.01	0.01	0.59	0.79	0.01	0.97	0.97	0.09
T	0.97	0.97	0.97	0.01	0.01	0.01	0.01	0.01	0.01

$$q(A) = 0.3, q(C) = 0.2, q(G) = 0.2, q(T) = 0.3$$

Pozícia motívu v  $S$  sa zvolí náhodne, každá báza sa vygeneruje z  $q$  alebo z jedného stĺpca  $W$

Tento model definuje rozdelenie  $\Pr(S | W)$ .



## Hľadanie motívov cez pravdepodobnostné modely

**Vstup:** dĺžka motívu  $L$ , sekvencie  $S_1, S_2, \dots, S_k$ , frekvencie  $q$

**Výstup:** spoločný motív ako matica frekvencií  $W$  maximalizujúca vierohodnosť dát  $\Pr(S_1|W) \cdot \dots \cdot \Pr(S_k|W)$

- Ťažký problém, používajú sa heuristické algoritmy
- Napríklad EM (expectation maximization)
- Lokálna optimalizácia, ktorá konverguje k lokálnemu maximu vierohodnosti
- Softvér: MEME

## Schéma algoritmu EM

- **Inicializácia:**

Zvoľ si počiatočnú maticu  $W$

(napr. zostavenú podľa jedného okna dĺžky  $L$ )

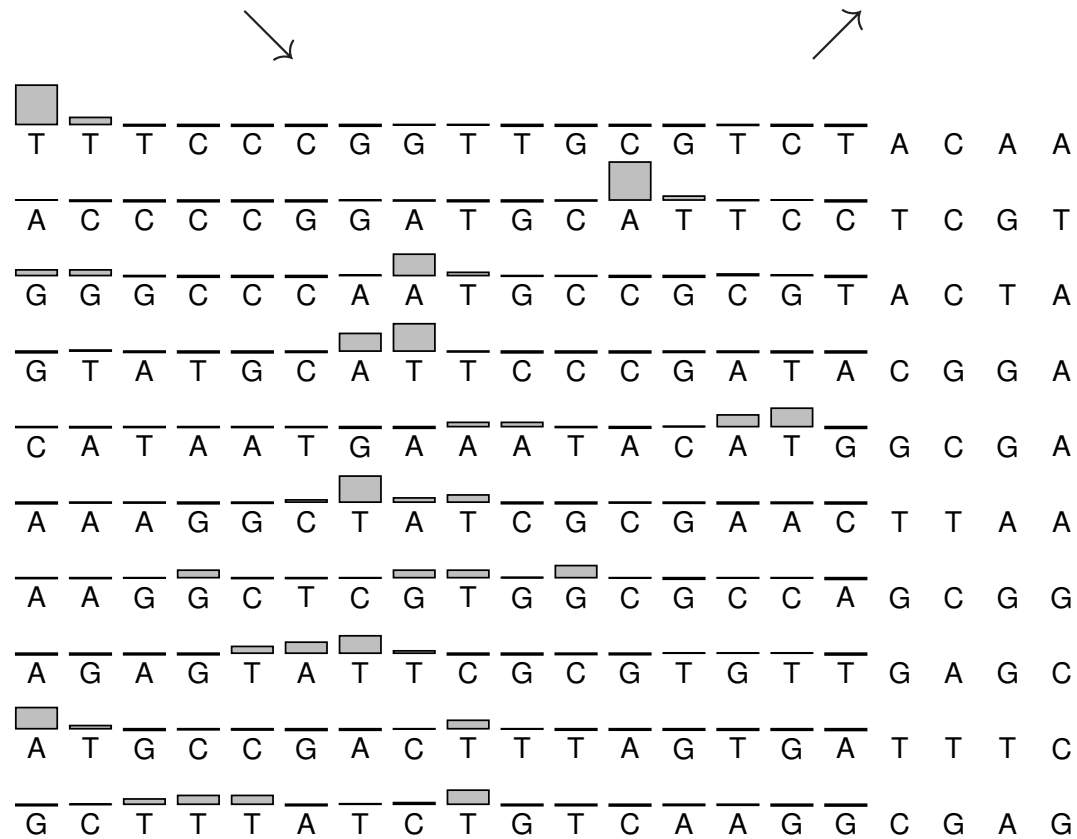
- **Iterácia:**

1. Prirad' každej pozícii  $j$  v sekvencii  $S_i$  váhu  $p_{i,j}$ , ktorá zodpovedá pravdepodobnosti, že na pozícii  $S_i[j]$  začína výskyt motívu  $W$
2. Spočítaj  $W$  zo všetkých možných výskytov v  $S_1, \dots, S_k$  váhovaných podľa  $p_{i,j}$

Iterácie zvyšujú vierohodnosť dát, kým nedôjde ku konvergencii.  
Skúšame veľa krát z rôznych počiatočných  $W$

## Príklad algoritmu EM

A	0.10	0.10	0.10	0.10	0.10	A	0.31	0.14	0.06	0.07	0.07
C	0.10	0.10	0.10	0.70	0.70	C	0.06	0.10	0.19	0.71	0.61
G	0.10	0.10	0.10	0.10	0.10	G	0.12	0.17	0.29	0.14	0.25
T	0.70	0.70	0.70	0.10	0.10	T	0.51	0.60	0.46	0.08	0.07



## Príklad algoritmu EM: ďalšia iterácia

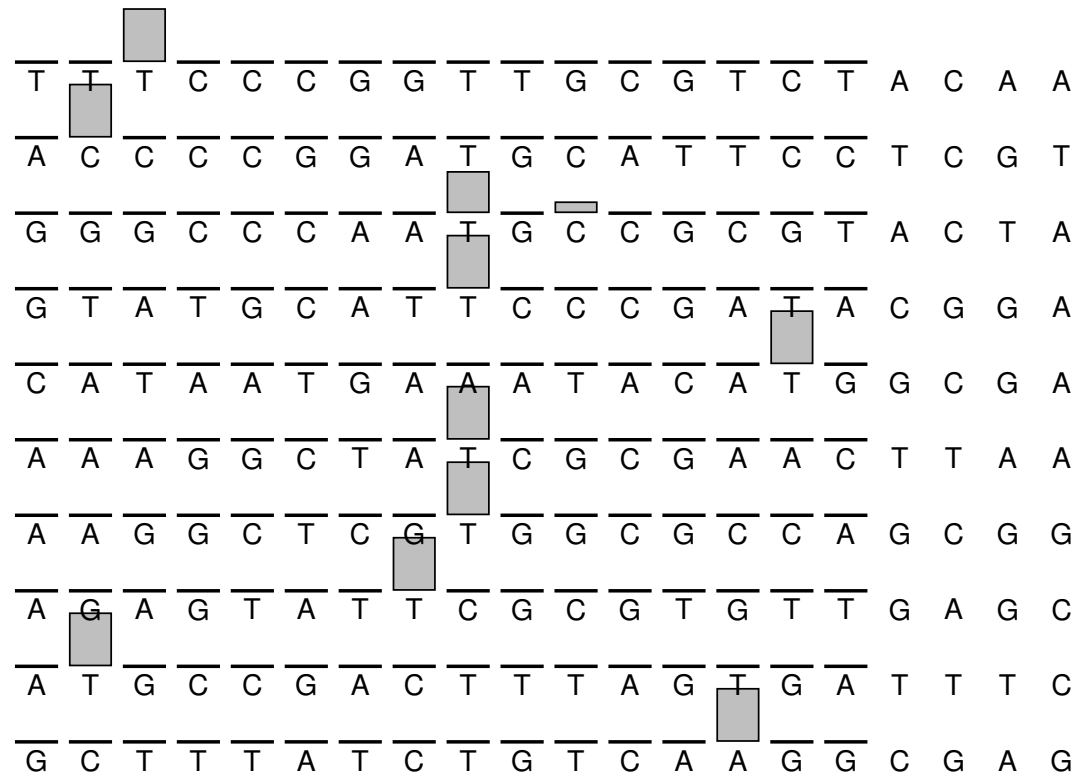
A	0.31	0.14	0.06	0.07	0.07	A	0.47	0.09	0.01	0.02	0.03
C	0.06	0.10	0.19	0.71	0.61	C	0.02	0.11	0.20	0.80	0.58
G	0.12	0.17	0.29	0.14	0.25	G	0.08	0.22	0.48	0.15	0.35
T	0.51	0.60	0.46	0.08	0.07	T	0.42	0.58	0.30	0.03	0.03



T T T C C C G G T T G C G T C T A C A A  
 A C C C C G G A T G C A T T C C T C G T  
 G G G C C C A A T G C C G C G T A C T A  
 G T A T G C A T T C C C G A T A C G G A  
 C A T A A T G A A A T A C A T G G C G A  
 A A A G G C T A T C G C G A A C T T A A  
 A A G G C T C G T G G C G C C A G C G G  
 A G A G T A T T C G C G T G T T G A G C  
 A T G C C G A C T T T A G T G A T T T C  
 G C T T T A T C T G T C A A G G C G A G

## Príklad algoritmu EM: po 20 iteráciách

A	0.10	$\epsilon$	$\epsilon$	$\epsilon$	$\epsilon$
C	0.12	0.52	0.48	$1 - 3\epsilon$	$\epsilon$
G	$\epsilon$	0.48	0.52	$\epsilon$	$1 - 3\epsilon$
T	0.78	$\epsilon$	$\epsilon$	$\epsilon$	$\epsilon$



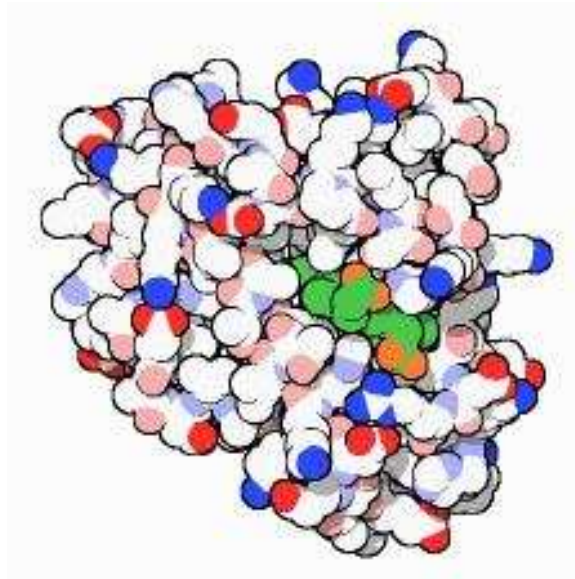
## Zhrnutie

- Microarray alebo RNA-seq merajú úroveň expresie pre veľa génov naraz, ale v dátach veľa šumu
- Zhlukovanie (clustering) nájde podobné gény, nepotrebujeme o dátach vopred nič vedieť (unsupervised learning)
- Klasifikácia môže rozlišovať napr. choroby podľa expresie, potrebuje dáta so známou odpoveďou (supervised learning)
- Dáta o expresii pomáhajú zostaviť regulačné siete
- Väzobné motívy môžeme reprezentovať rôznym spôsobom (reťazec, regulárny výraz, skórovacia matica)
- Tieto motívy nie sú dosť špecifické, preto sa ťažko rozpoznávajú ich výskyty v genóme
- EM algoritmus na hľadanie nových motívov v sekvenciách

# Štruktúra a funkcia proteínov

Tomáš Vinař

29.11.2018



## Proteíny

Reťazce 20 rôznych aminokyselín s rôznymi chemickými vlastnosťami:

Aminokyselina	Postranný reťazec	Jeho vlastnosti
Alanín (A)	-CH <sub>3</sub>	hydrofóbný
Arginín (R)	-(CH <sub>2</sub> ) <sub>3</sub> NH-C(NH)NH <sub>2</sub>	bázický
Asparagín (N)	-CH <sub>2</sub> CONH <sub>2</sub>	hydrofilný
Kyselina asparágová (D)	-CH <sub>2</sub> COOH	kyslý
Cysteín (C)	-CH <sub>2</sub> SH	hydrofóbný
Kyselina glutámová (E)	-CH <sub>2</sub> CH <sub>2</sub> COOH	kyslý
Glutamín (Q)	-CH <sub>2</sub> CH <sub>2</sub> CONH <sub>2</sub>	hydrofilný
Glycín (G)	-H	hydrofilný
Histidín (H)	-CH <sub>2</sub> -C <sub>3</sub> H <sub>3</sub> N <sub>2</sub>	bázický
Izoleucín (I)	-CH(CH <sub>3</sub> )CH <sub>2</sub> CH <sub>3</sub>	hydrofóbný
Leucín (L)	-CH <sub>2</sub> CH(CH <sub>3</sub> ) <sub>2</sub>	hydrofóbný
Lyzín (K)	-(CH <sub>2</sub> ) <sub>4</sub> NH <sub>2</sub>	bázický
Metionín (M)	-CH <sub>2</sub> CH <sub>2</sub> SCH <sub>3</sub>	hydrofóbný
Fenylalanín (F)	-CH <sub>2</sub> C <sub>6</sub> H <sub>5</sub>	hydrofóbný
Prolín (P)	-CH <sub>2</sub> CH <sub>2</sub> CH <sub>2</sub> -	hydrofóbný
Serín (S)	-CH <sub>2</sub> OH	hydrofilný
Treonín (T)	-CH(OH)CH <sub>3</sub>	hydrofilný
Tryptofán (W)	-CH <sub>2</sub> C <sub>8</sub> H <sub>6</sub> N	hydrofóbný
Tyrozín (Y)	-CH <sub>2</sub> -C <sub>6</sub> H <sub>4</sub> OH	hydrofóbný
Valín (V)	-CH(CH <sub>3</sub> ) <sub>2</sub>	hydrofóbný

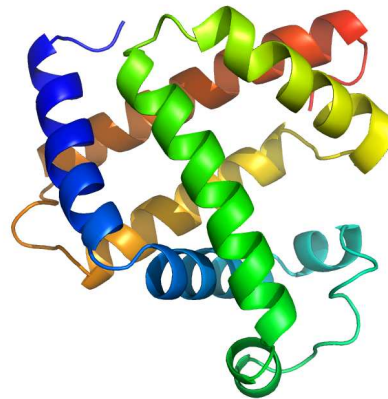
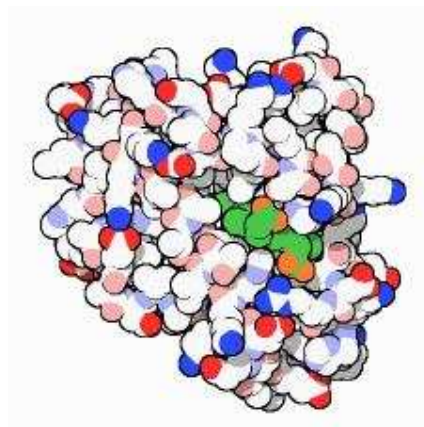


## Štruktúra proteínov

- **Primárna štruktúra:** sekvencia aminokyselín
- **Sekundárna štruktúra:** pravidelné útvary alfa-hélix, beta-skladaný list (beta sheet)
- **Terciálna štruktúra:** presné 3D rozloženie atómov
- **Kvartérna štruktúra:** interakcia viacerých proteínov v komplexe



Myoglobín, prvý proteín so známou štruktúrou [Kendrew et al 1958]



## Experimentálne určovanie štruktúry

- RTG kryštalografia (X-ray crystallography)  
vyžaduje proteín v kryštalickej forme
- NMR (nuclear magnetic resonance spectroscopy)  
hlavne používaná na kratšie proteíny
- Náročný a drahý proces
- Databáza štruktúr PDB  
146 000 proteínových štruktúr  
(UniProt má 134 miliónov sekvencií)

## Určovanie štruktúry proteínov ako bioinformatický problém

(protein structure prediction, protein folding)

**Vstup:** sekvencia proteínu

**Výstup:** 3D pozície atómov alebo aminokyselín

### Ab initio metódy

- Nájdi štruktúru s najnižšou voľnou energiou
- Vzorce na približný výpočet energie založené na fyzike
  - sily medzi atómami v proteíne a okolitom roztoku
- Štatistické vzorce merajúce typické vzialenosti medzi aminokyselinami na známych štruktúrach
- V oboch prípadoch veľmi ťažký výpočtový problém
  - simulácia molekulárnej dynamiky
  - optimalizačné metódy, napr. simulované žihanie
- Používané na malé proteíny a zlepšenie približných štruktúr

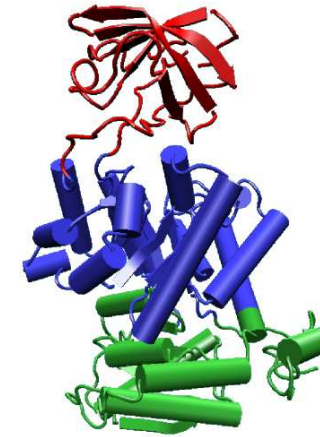
## Protein threading

- Aj proteíny s pomerne odlišnou sekvenciou môžu mať podobnú štruktúru
- Môžeme skúsiť “napasovať” proteín na každú známu štruktúru
- Určitý typ zarovnania, ale pri skórovaní uvažujeme aj interakcie medzi amino kyselinami blízko v štruktúre
- Výpočtovo ťažký problém

## Proteínové domény a rodiny

### Doména (domain)

- Časť proteínu s nezávislou štruktúrou
- Veľa proteínov sa skladá z viacerých domén
- Domény sa tiež v proteínoch preskupujú počas evolúcie



### Rodina (family)

- Skupina proteínov/domén s podobnou sekvenciou, štruktúrou, funkciou
- Ak poznáme štruktúru jedného člena rodiny, môžeme predpokladať, že ostatné majú podobnú

## Proteíny ako skladačka domén

### Databáza Pfam

Domény v proteínoch rozdelené do rodín

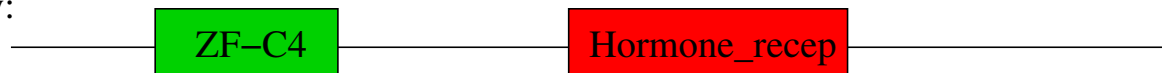
80% proteínov aspoň jedna známa doména

58% proteínových sekvencií pokrývajú známe domény

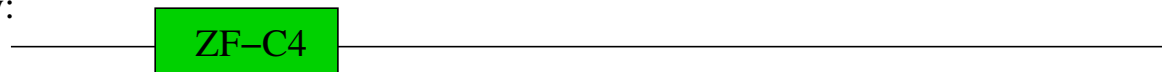
### Príklad:

Rôzne architektúry obsahujúce doménu Zinc finger, C4 type (Pfam)

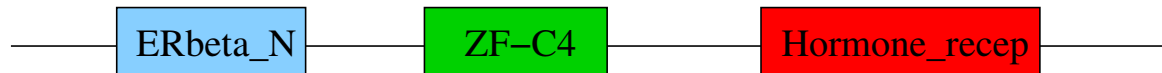
5124 proteínov:



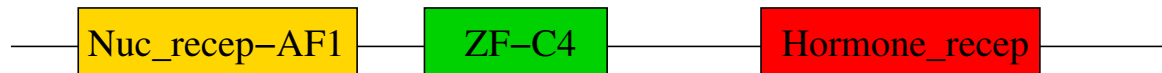
1220 proteínov:



208 proteínov:



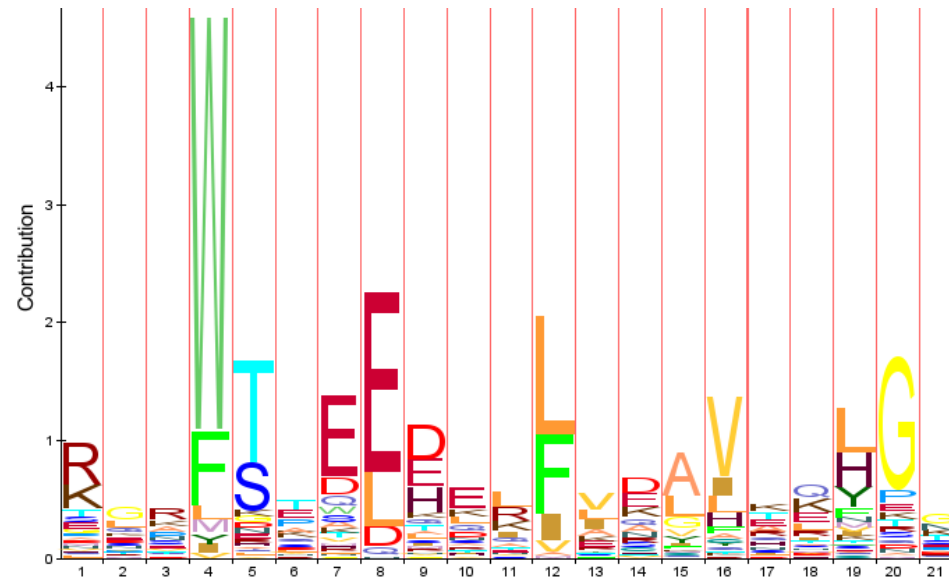
170 proteínov:



## Charakterizácia rodín proteínov

- Zarovnanie medzi známymi prvkami rodiny a novým proteínom nemusia nájsť vzdialených členov
- Viacnásobné zarovnanie rodiny ukáže dôležité zachované pozície

```
MEEW SASEANLFEEALEKYGKDF  
PDEWTVEDKVLFEQAFSFHGKT.  
GTKWTAENKKFENALAFYDKDT  
SKNWSEDDLQLLIKAVNLFPA GT  
EKPwSNQETLLLLLEAIETYGDD.  
AREWTDQETLLLLLEGLEMHKDD.  
KPEwSDKEILLLEAVMHY GDD.  
DDTWTAQELVLLSEGVEMYS...  
KKNwSDQEMLLLLLEGIEMYE...  
DENwSKEDLQKLLKGIQEF GAD.  
EDDWSQAEQKAFETALQKYPKGT  
EEAWTQSQQKLELALQQYPKGA  
EDVWSATEQKTLEDAIKKHKSSD  
AMSwTHEDEFELLKAAHKFKMG.
```



## Pravdepodobnostný profil rodiny

(profile, position specific score matrix PSSM)

- V zarovnaní spočítaj  $e_i(x)$ : frekvencia výskytu písmena  $x$  v stĺpci  $i$
- Dostaneme model, ktorý generuje sekvenciu  $x_1, x_2, \dots, x_n$  s pravdepodobnosťou

$$e_1(x_1) \cdot e_2(x_2) \cdots e_n(x_n)$$

- Nulová hypotéza: sekvencia bola vygenerovaná náhodne, kde písmeno  $x$  má frekvenciu  $q(x)$
- Skóre: logaritmus pomeru pravdepodobností v dvoch modeloch

$$\log \frac{\prod_{i=1}^n e_i(x_i)}{\prod_{i=1}^n q(x_i)} = \sum_{i=1}^n \log \frac{e_i(x_i)}{q(x_i)} = \sum_{i=1}^n s_i(x_i)$$



## Hračkářský příklad PSSM

- Uvažujme len leucín L a alanín A
- Majme zarovnanie 10 sekvencií s nasledujúcimi počtami

	1	2	3	4
A	2	6	9	1
L	8	4	1	9

- Nulová hypotéza  $q(A) = 30\%$ ,  $q(L) = 70\%$
- Sekvencia LAAL má v profile pravdepodobnosť  
 $0.8 \cdot 0.6 \cdot 0.9 \cdot 0.9 = 0.3888$ ,  
v nulovom modeli  $0.7 \cdot 0.3 \cdot 0.3 \cdot 0.7 = 0.0441$
- Skóre  $\log_2(0.3888/0.0441) = 3.14$

## Hračkářsky příklad PSSM

- Majme zarovnanie 10 sekvencií s nasledujúcimi počtami

	1	2	3	4
A	2	6	9	1
L	8	4	1	9

- Nulová hypotéza  $q(A) = 30\%$ ,  $q(L) = 70\%$
- Skóre alanínu v prvom stĺpci  $s_1(A) = \log_2(0.2/0.3) = -0.58$   
skóre leucínu v prvom stĺpci  $s_1(L) = \log_2(0.8/0.7) = 0.19$
- Dostávame tabuľku skór

	1	2	3	4
A	-0.58	1.00	1.58	-1.58
L	0.19	-0.81	-2.81	0.36

- Skóre LAAL je  $0.19 + 1 + 1.58 + 0.36 = 3.13$   
Skóre ALAL je  $-0.58 - 0.81 + 1.58 + 0.36 = 0.55$

## Pseudocounts

Ak na niektorej pozícii určitá amino kyselina nebola pozorovaná, mala by v modeli pravdepodobnosť 0

	1	2	3	4
A	2	6	9	0
L	8	4	1	10

Aby sme sa vyhli tomuto problému, pridáme ku každému políčku najskôr nejakú malú hodnotu, **pseudocount**, napr. 0,5:

	1	2	3	4
A	2.5	6.5	9.5	0.5
L	8.5	4.5	1.5	10.5

Potom postupujeme ako predtým

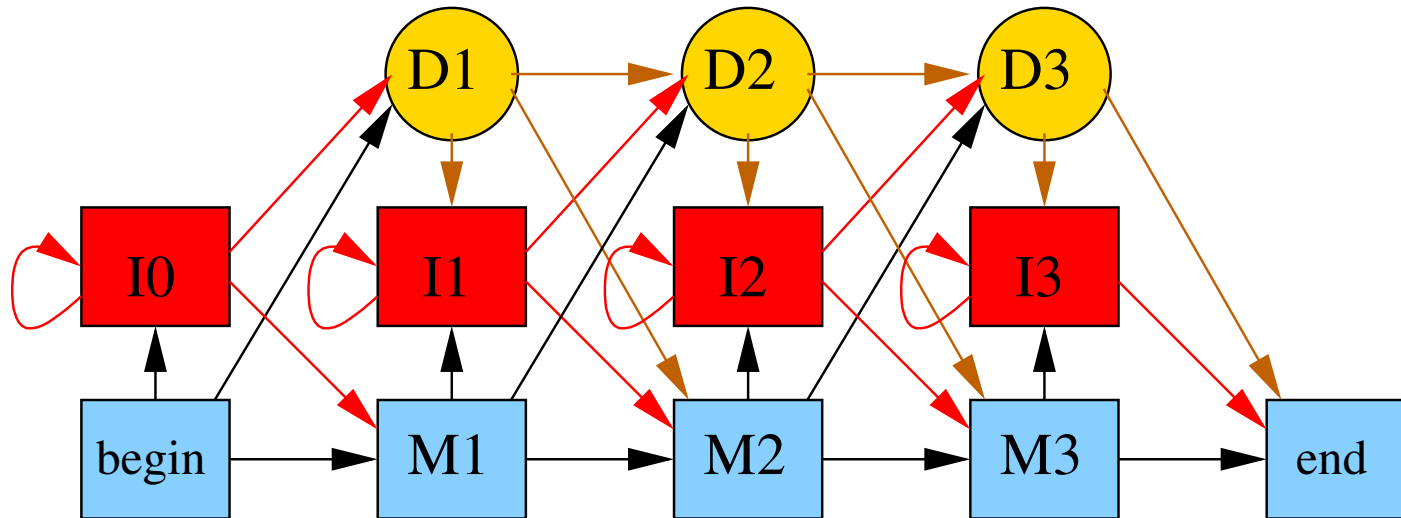
## Profilové HMM

Rozšíř profil o inzercie a delécie

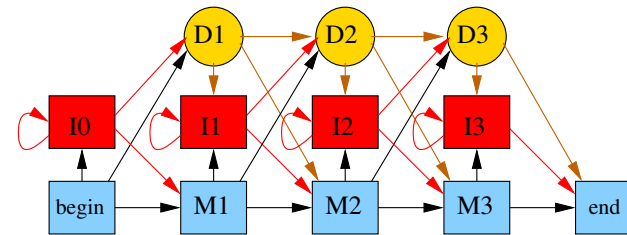
**PSSM profil ako HMM:**



**Profilové HMM:** match state, insert state, delete state



## Konštrukcia profilového HMM



- Začneme z viacnásobného zarovnaní
- Stĺpcom s málo medzerami priradíme match stavy, ostatné budú v insert stavoch
- V každom stĺpci zrátame  $E_i(a)$ : počet výskytov  $a$
- Pravdepodobnosť emisie  $e_i(a) = \frac{E_i(a)}{\sum_b E_i(b)}$
- Pridáme “pseudocounts”, aby sme nemali nulové položky  
$$e_i(a) = \frac{E_i(a)+c}{\sum_b (E_i(b)+c)}$$
- Pravdepodobnosti prechodu nastavíme podľa medzier v zarovnaní
- Veľmi podobné sekvencie môžeme použiť s menšou váhou

## Použitie profilov a profilových HMM

### Odkiaľ vziať profily/profilové HMM?

- PSI-Blast: PSSM iteratívne zo skupiny podobných proteínov
- Databáza Pfam: rodiny domén reprezentované ako profilové HMM
- PSSM sa používajú aj na reprezentáciu motívov v DNA (napr. väzobné miesta transkripčných faktorov)

### Nájdí výskyty profilu v proteínovej sekvencii

- Podobné problému lokálneho zarovnania
- PSSM profily: dynamické programovanie, penalta za medzery
- Profilové HMM: Viterbiho algoritmus (mierne modifikovaný)

### Patrí proteín do rodiny?

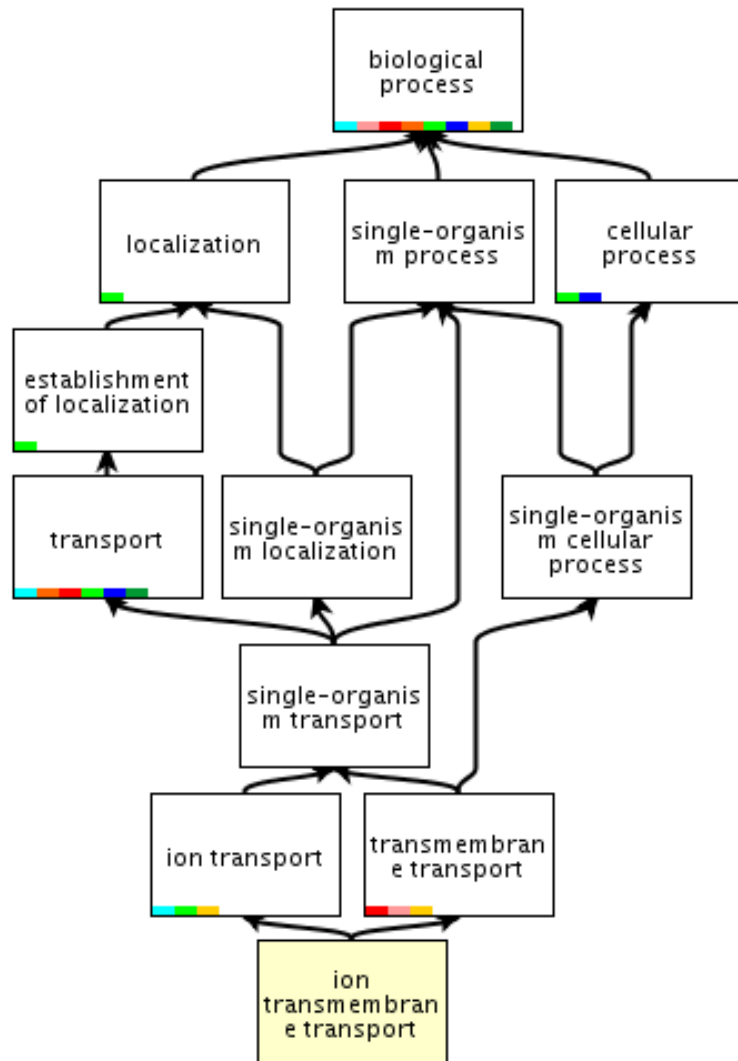
- PSSM profily: vysoko kladné skóre
- Profilové HMM: pravdepodobnosť vygenerovania sekvencie, normalizovaná na dĺžku

## Funkcia proteínu

- Pre niektoré proteíny určená laboratórne
- Na ďalšie proteíny prenášame bioinformaticky pomocou podobnosti sekvencie, prítomnosti domén, polohy v genóme a ďalších dát
- Swissprot/Uniprot zhromažďuje údaje o funkcii proteínov
- Klasifikácia proteínov pomocou Gene ontology (GO)  
Príklad pojmu v GO:  
Accession: GO:0034220  
Name: ion transmembrane transport  
Ontology: biological\_process  
Definition: A process in which an ion is transported from one side of a membrane to the other by means of some agent such as a transporter or pore.  
Comment: Note that this term is not intended for use in annotating lateral movement within membranes.

# Gene ontology (GO)

Hierarchická štruktúra pojmov:





## Zhrnutie: akú štruktúru má proteín?

- Pozriem do PDB, či má známu štruktúru
- Ak nie, skúsim BLAST voči proteínom so známou štruktúrou
- Ak nič, skúsim hľadať domény so známou štruktúrou
- Ak nič, skúsim protein threading
- Pre krátke proteíny môžem skúsiť minimalizovať energiu, inak získané štruktúry doplniť/vylepšiť minimalizáciou energie

Minimalizácia energie je výpočtovo veľmi náročná

Súťaž CASP raz za dva roky

Zaujímavosti: Folding@home, Foldit

## Oznamy

- Budúci štvrtok 13.12.:
  - posledná prednáška
  - predposledné cvičenia pre informatikov
  - predposledné cvičenia pre biológov
- Štvrtok 20.12.:
  - posledné cvičenia pre informatikov
  - nepovinné prezentácie journal clubu
  - posledné cvičenia pre biológov
  - termín DÚ3
- Piatok 21.12. správy zo journal clubu
- Budúci štvrtok dohodneme:
  - či chcete prezentovať journal club (dohodnite sa v skupinách)
  - termín skúšky (doneste si termíny iných skúšok)

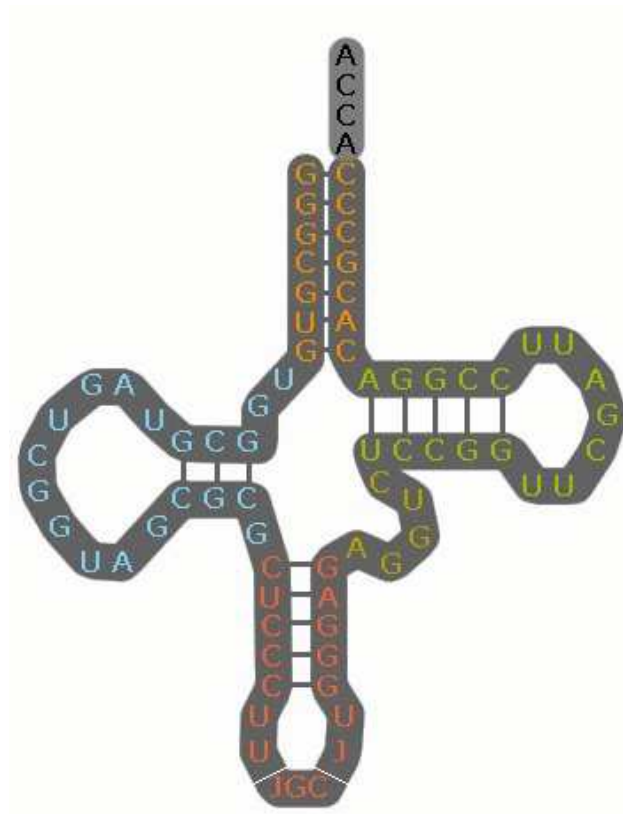
## Správa zo journal clubu

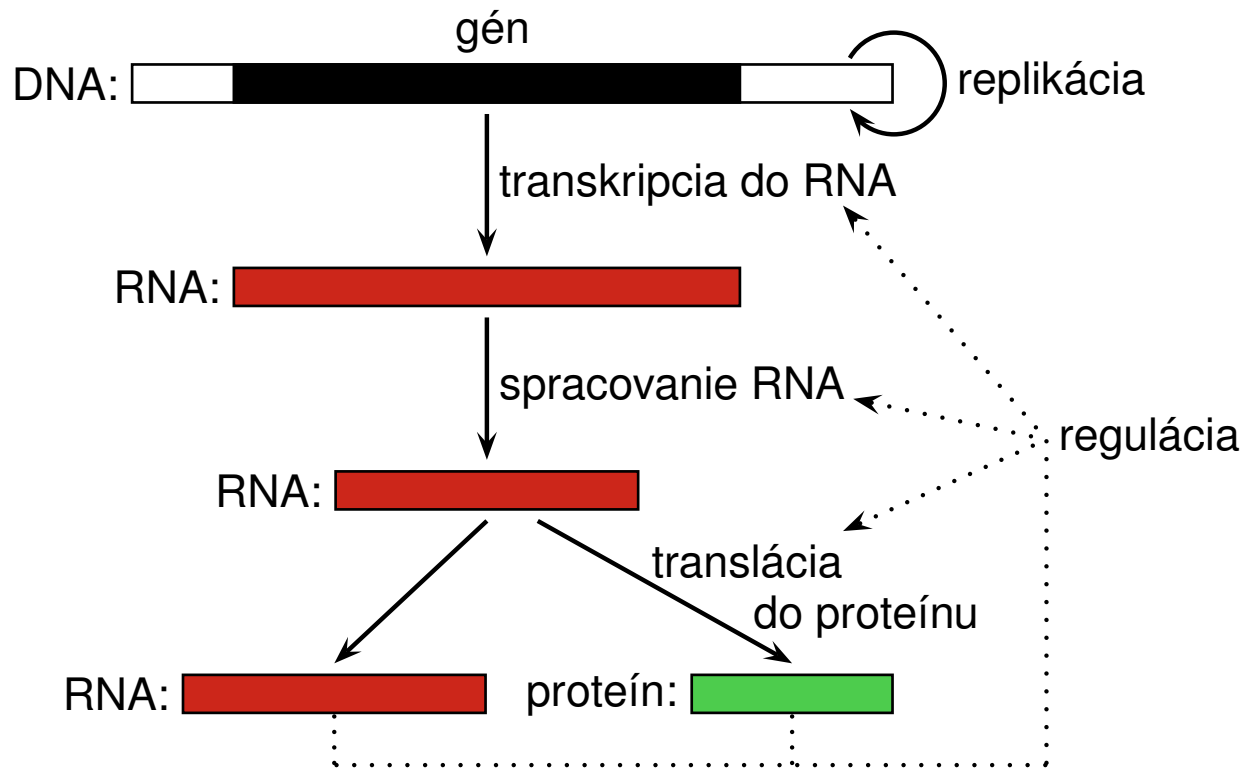
- Vlastnými slovami hlavné metódy a výsledky článku
- Pochopiteľná pre študentov tohto predmetu (inf aj bio)
- Netreba pokryť všetko a naopak, môžete využiť aj iné zdroje
- Skúste vložiť vlastný pohľad na tému
- Rozsah cca 1-2 strany na osobu, jeden ucelený text
- V správe vymenujte členov skupiny, ktorí sa podieľali na jej spísaní, dostanú rovnako bodov
- Pdf odovzdať cez Moodle (stačí 1 za skupinu)

# RNA

Broňa Brejová

6.12.2018





## Vlastnosti RNA

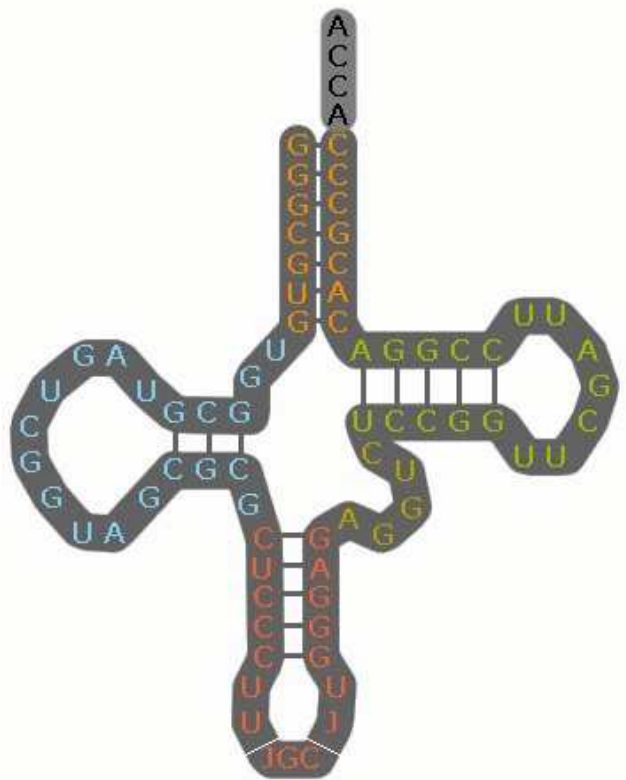
### Ako sa líši od DNA?

- obsahuje ribózu namiesto deoxyribózy
- obsahuje uracil namiesto tymínu (bázy A,C,G,U)
- jednovláknové reťazce, zvyčajne kratšie
- zložitá sekundárna štruktúra: spárované komplementárne úseky
- okrem párov A-U, C-G aj nekanonické páry (napr. G-U)

# Štruktúra RNA

## Príklad: transferová RNA (transfer RNA)

Sekundárna štruktúra  
(secondary structure):  
páry nukleotidov

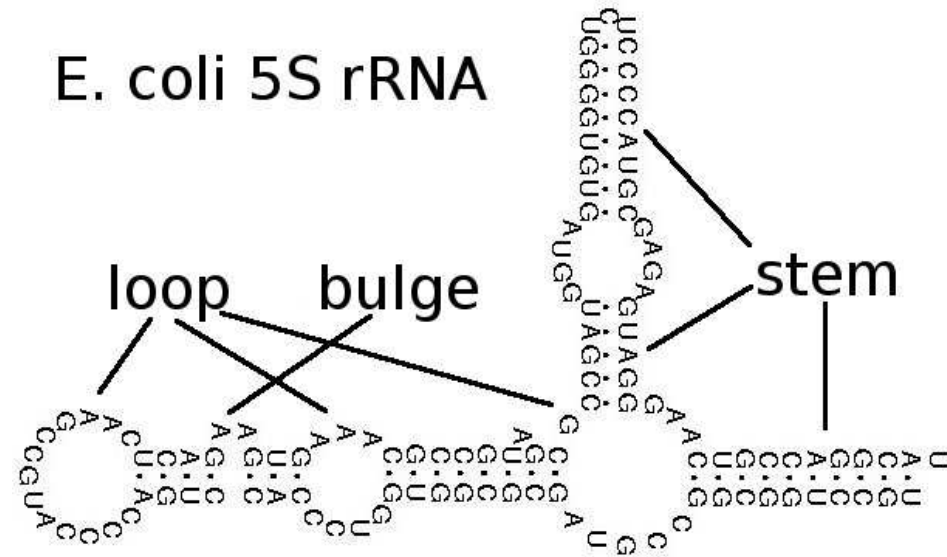


Terciárna štruktúra  
(tertiary structure):  
3D súradnice



## Sekundárna štruktúra RNA

### Prvky sekundárnej štruktúry



V tomto prípade spárované bázy tvoria **dobre uzátvorkovaný výraz**:

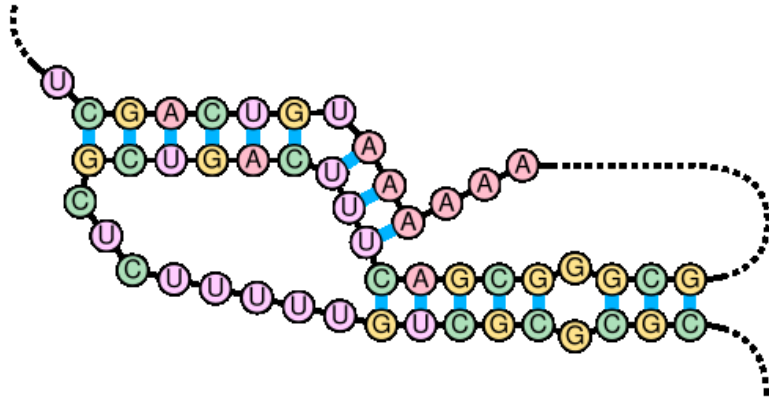
(((((((((((.....((( )..)).(( )..))))))))..)).  
 UGCCUGGCGGCCGUAGCG...UAGCGCC...GGGAACUGCCAGGCAU

t.j. ak máme páry medzi pozíciami  $i$  a  $j$  a  $i'$  a  $j'$  a  $i < i'$ , tak buď  $i < i' < j' < j$  alebo  $i < j < i' < j'$ .



## Sekundárna štruktúra RNA

### Pseudouzol: výnimka z dobrého uzátvorkovania



Mnohé algoritmy na prácu so sekundárnou štruktúrou ignorujú pseudouzly.

Zhruba 1.4% RNA nukleotidových párov v pseudouzloch.

## Problém: určovanie štruktúry RNA

**Vstup:** RNA sekvencia

**Cieľ:** nájsť spárované bázy

**Veľmi zjednodušená formulácia:** nájsi dobre uzátvorkované spárovanie s najväčším počtom komplementárnych párov A-U, C-G.

**Príklad:** ((.(((( )))((.( )))))  
GAACACAUGUAAAAUUUGUC

## Nussinovej algoritmus

### Dynamické programovanie:

Majme RNA  $X_1, \dots, X_n$ .

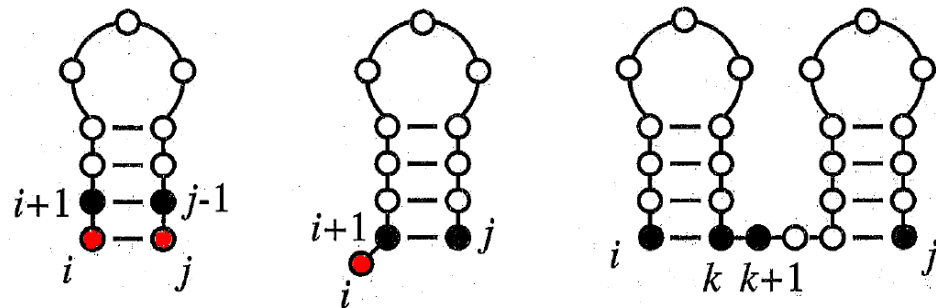
Nech  $A[i, j]$  je maximálny počet párov v podreťazci  $X_i, X_{i+1}, \dots, X_j$ .

### Rekurencia:

Podreťazce dĺžky 1: žiadne páry  $A[i, i] = 0$

Dlhšie podreťazce: 3 prípady

- $X_i$  a  $X_j$  sú pár:  $A[i, j] = A[i + 1, j - 1] + 1$
- $X_i$  je nespárované:  $A[i, j] = A[i + 1, j]$
- $X_i$  je pár s  $X_k$  pre  $i < k < j$ :  $A[i, j] = A[i, k] + A[k + 1, j]$

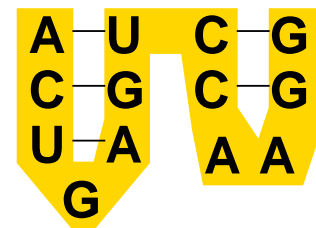


**Rekurencia:**  $A[i, j] = \max \begin{cases} A[i + 1, j - 1] + c(X_i, X_j), \\ A[i + 1, j], \\ \max_{k=i+1 \dots j-1} \{A[i, k] + A[k + 1, j]\} \end{cases}$

	A	C	U	G	A	G	U	C	C	A	A	G	G
A	0	0	1	1	1	2	3	3	3	3	3	4	5
C		0	0	1	1	2	2	2	2	3	3	4	4
U			0	0	1	1	1	2	2	3	3	3	3
G				0	0	0	1	2	2	2	2	3	3
A					0	0	1	1	1	1	1	2	3
G						0	0	1	1	1	1	2	2
U							0	0	0	1	1	1	2
C								0	0	0	0	1	2
C									0	0	0	1	1
A										0	0	0	0
A											0	0	0
G												0	0
G													0

$c(X_i, X_j) = \begin{cases} 1 & \text{ak } X_i - X_j \text{ môže byť pár} \\ 0 & \text{inak} \end{cases}$

$A[i, j] = 0$  pre  $i \geq j$



**Zložitosť:**

$O(n^3)$  čas

$O(n^2)$  pamäť

## Štruktúra s minimálnou voľnou energiou (MFE folding)

Realistickejšia formulácia problému určovania sek. štruktúry RNA.

**Predpoklad:** molekula v rovnovážnom stave s minimálnou Gibbsovou voľnou energiou (Gibbs free energy).  
Energie pre niektoré sekvencie experimentálne zmerané.

**Nearest neighbor model:** sada parametrov, energie pre dvojice susedných párov v helixoch, dĺžky slučiek atď.

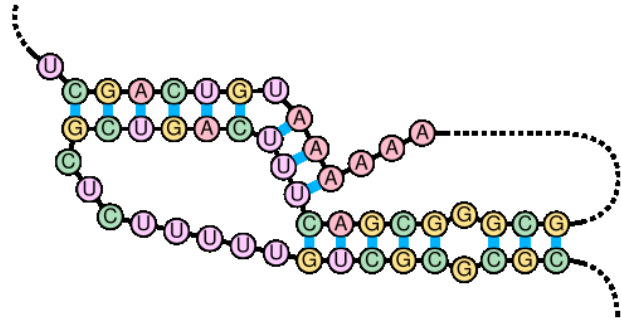
Odvodené z nameraných dát.

### Príklad:

		Y:	A	C	G	U		
5'	CX	3'	-----					
3'	GY	5'	X:A		.	.	.	-2.1
			C		.	.	-3.3	.
			G		.	-2.4	.	-1.4
			U		-2.1	.	-2.1	.

Štruktúra s minimálnou energiou sa dá nájsť podobným (ale zložitejším) dyn. programovaním [Zuker and Stiegler, 1981].

## Algoritmy dovoľujúce pseudouzly



Vo všeobecnosti NP-ťažký problém [Lyngso and Pedersen, 2000].

Pomalé dyn. programovanie  $O(n^4)$  –  $O(n^6)$  nájde niektoré typy pseudouzlov [Rivas and Eddy, 1999].

Tiež môžeme použiť heuristiky [Ren et al., 2005] (opakované vytváranie silných helixov).

## Pravdepodobnostné modely na predikciu štruktúry

Chceme: model, ktorý generuje dvojice sekvencia a sek. štruktúra

Použitie: pre danú sekvenciu nájsť najpravdepodobnejšiu štruktúru

HMM nevhodné: závislosti medzi vzdialenými spárovanými bázami.

## Stochastická bezkontextová gramatika, stochastic context free grammar (SCFG):

Rozšírenie bezkontextových gramatík

Pravidlám pridáme pravdepodobnosti

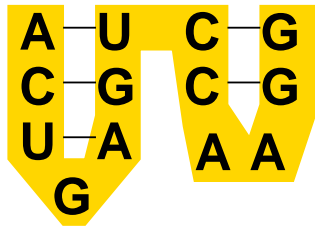




## Stochastické bezkontextové gramatiky

$S \rightarrow aSu|uSa|cSg|gSc|aS|cS|gS|uS|Sa|Sc|Sg|Su|SS|\epsilon$

$S \rightarrow SS \rightarrow aSuS \rightarrow acSguS \rightarrow acuSaguS \rightarrow acugSaguS \rightarrow$   
 $acugaguS \rightarrow acugagucSg \rightarrow acugagucgScg \rightarrow acugagucgSacg \rightarrow$   
 $acugagucgaSacg \rightarrow acugagucgaacg$



Bázy vygenerované v jednom kroku sú spárované.

**Úloha:** Nájdi najpravdepodobnejšie odvodenie danej RNA

⇒ určuje sekundárnu štruktúru

**Riešenie:** Dynamické programovanie (CYK algoritmus),  $O(n^3)$

**Trénovanie parametrov:** zo známych RNA štruktúr

## Gramatiky vs. minimalizácia energie

### Výhody gramatík:

- možno automaticky trénovať, netreba náročné experimenty.
- rozšíriteľné na modely viacerých sekvencií.

### Nevýhody gramatík:

- jednoduché gramatiky nevystihujú komplexitu problému
- nižšia presnosť ako minimalizácia energie.

### Conditional log-linear models: podobné na SCFG, ale:

- maximalizujeme podmienenú pravdepodobnosť správnej odpovede (diskriminatívne tréovanie: nemodelujeme rozdelenie pravdepodobností samotných RNA sekvencií, iba štruktúr)
- dosahujú dokonca lepšiu presnosť ako minimalizácia energie.

## Evolúcia RNA sekvencií

Často vidíme koreláciu medzi mutáciami v spárovaných bázach.  
Např. C sa zmení na A, spárované G sa súčasne zmení na U

**Príklad:** niekoľko sekvencií z D ramena tRNA

```
(((((.....))))  
GCUCAGCC.CGCG...AGAGC  
GCCUAGCC.UGGUCA.AGGGC  
GUCUAGC...GGA...AGGAU  
GAGCAGUU.CGCU...AGCUC  
GUUCAAUC..GGU...AGAAC
```

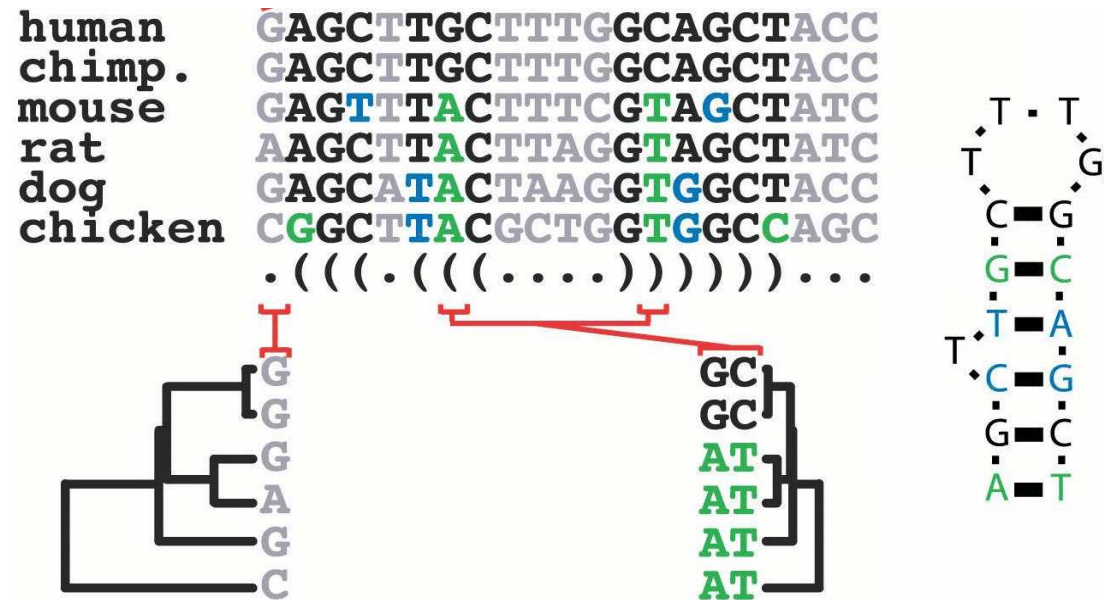
**Úloha:** daných je niekoľko (zarovnaných) sekvencií RNA  
nájdite ich spoločnú RNA štruktúru

(korelácie medzi spárovanými bázami potvrdzujú správnosť štruktúry)

# Hľadanie spoločnej štruktúry pre viacero sekvencií

## Phylo-SCFG:

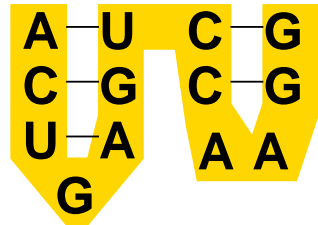
- namiesto jednotlivých báz emituje stĺpce zarovnaní podľa fylogenetického stromu.
- nespárované bázy emituje bežnou substitučnou maticou,
- spárované bázy substitučnou maticou dvojíc ( $16 \times 16$ ).



## Problém: hľadanie známych typov RNA génov v genóme

- Obdoba databázy Pfam  $\Rightarrow$  databáza Rfam  
2473 rodín podobných RNA génov
- Pre každú rodinu zarovnanie a pravdepodobnostný model
- Proteínové rodiny reprezentujeme profilovými HMM
- Pre RNA kovariančné modely (covariance model, CM):  
špeciálny typ SCFG

## Kovariančný model



$$\begin{array}{lll}
 S \rightarrow B_1 & P_1 \rightarrow aP_2u & P_4 \rightarrow cP_5g \\
 B_1 \rightarrow P_1P_4 & P_2 \rightarrow cP_3g & P_5 \rightarrow gL_2c \\
 & P_3 \rightarrow uL_1a & L_2 \rightarrow aL_3 \\
 & L_1 \rightarrow gE_1 & L_3 \rightarrow aE_2 \\
 & E_1 \rightarrow \epsilon & E_2 \rightarrow \epsilon
 \end{array}$$

- $S$  =start,  $E_i$  =end  
 $P_i$  =pár,  $L_i$  =nespárovaná báza vľavo,  $R_i$  =nespárovaná báza vpravo  
 ďalšie neterminály modelujú indely.

- terminály (bázy) sa emitujú s pravdepodobnosťami podľa príslušného stĺpca zarovnania

$$\text{Např. } P_1 \rightarrow \overbrace{aP_2u}^{0.2} \mid \overbrace{uP_2a}^{0.2} \mid \overbrace{cP_2g}^{0.4} \mid \overbrace{cP_2u}^{0.1}$$

- veľkosť gramatiky úmerná dĺžke modelovanej RNA rodiny

## Kovariančný model

### Použitie:

hľadať výskyty génu v DNA (lokálne zarovnanie),  
nájsť štruktúru nového génu z tej istej rodiny (globálne zarovnanie).

**Dynamické programovanie:** čas  $O(MND^2)$ ,

$M$  = počet neterminálov v gramatike, úmerný dĺžke zarovnania,

$N$  = dĺžka DNA sekvencie,

$D$  = max. dĺžka RNA génu v DNA (úmerná  $M$ ).

### Zrýchlenie:

nájsť sľubné úseky podobné na sekvencie v RNA rodine

(iba na základe podobnosti sekvencií)

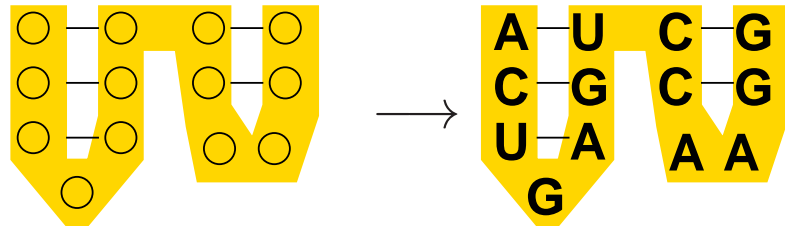
aplikuj CM iba na sľubné úseky

## Problém: dizajn RNA

Daná RNA sekundárna štruktúra (párovanie).

Nájdí sekvenciu, pre ktorú je táto štruktúra optimálna.

Nie je známy efektívny algoritmus, heuristiky často nájdu sekvenciu pomerne rýchlo.



**Použitie:** skúmanie možných RNA štruktúr, vývoj liekov (ribozymes, riboswitches), RNA pre laboratórne techniky, RNA nanoštruktúry



## Zhrnutie

- Určovanie sekundárnej štruktúry RNA:  
minimalizácia energie, pravdepodobnostné SCFG
- Lepšie výsledky, keď použijeme zarovnanie viacerých sekvencií  
(PhyloSCFG)
- Známe rodiny reprezentujeme pomocou kovariančných modelov  
v nových sekvenciách hľadáme výskyty rodín z databázy Rfam
- Väčšina problémov sa dá riešiť dynamickým programovaním, ktoré  
je pomerne pomalé a ignoruje pseudouzly.
- Ďalšie zaujímavé problémy: napr. dizajn RNA štruktúr

## Oznamy

- Dnes posledná prednáška
- Budúci štvrtok 20.12.:
  - posledné cvičenia pre informatikov
  - nepovinné prezentácie journal clubu
  - posledné cvičenia pre biológov
  - termín DÚ3 (odovzdať na papieri alebo cez Moodle)
- Piatok 21.12. správy zo journal clubu
- Body z domácich úloh a journal clubu oznámime elektronicky
  - sledujte oznamy na Facebooku

## Skúška

- Povolený ťahák na dvoch listoch A4, kalkulačka
- Na stránke ukážky jednoduchých príkladov, približne 50% bodov
  - ak niektorý nevíete riešiť, spýtajte sa na Facebooku
  - v prípade pred záujmu pred skúškou konzultačné hodiny
- Zvyšné príklady budú prekvapením, v minulosti sa vyskytli:
  - Krátke príklady na pochopenie základných pojmov
  - Pre informatikov: navrhnite/modifikujte algoritmus alebo model
  - Pre biológov: usudzovanie z konkrétnych výsledkov, výber metódy pre daný problém
- Termín?

# Polymorfizmus a populačná genetika

Broňa Brejová

13.12.2018



## Populačná genetika

- Rôzne jedince toho istého druhu nemajú identický genóm
- Tieto rozdiely vplývajú na fenotyp (výzor, správanie, choroby, ...)
- Genómy viacerých jedincov môžeme sekvenovať a porovnávať s referenčnou verziou

### Možné aplikácie populačnej genetiky:

- Úloha jednotlivých genetických rozdielov
- História a charakter populácie (podpopulácie, migrácia, historická veľkosť populácie)

## SNPy (Single Nucleotide Polymorphisms)

- SNP: jednobázová variabilita medzi jedincami ( $> 1\%$  jedincov)
- Obvykle iba dve formy: **väčšinová** a **menšinová** alela
- Aj malá zmena v DNA môže spôsobiť veľké fenotypické zmeny

### Systematické mapovanie SNPov:

Projekt 1000 ľudských genómov 2008-2015

identifikácia  $> 95\%$  SNPov s aspoň  $1\%$  frekvenciou menšinovej alely  
pomocou NGS sekvenovania

## Mapovanie asociácií (Trait/Disease Association Mapping)

- Znaky (a choroby) vznikajú kombináciou genetických a environmentálnych vplyvov
- Cieľ: Identifikovať genetické vplyvy
  - Ako fungujú choroby?
  - Aký je risk dedičného faktoru choroby?
  - Vývoj nových liekov, ich správne cielenie (farmakogenomika)

Napr. mutácie v génoch rodiny cytochrómu P450 majú vplyv na odbúravanie liekov v pečeni, ovplyvňujú veľkosť potrebnej dávky

## Odbočka: diploidné genómy

- Človek je **diploidný**: má v bunke po dva chromozómy 1...22 plus pohlavné chromozómy X,X alebo X,Y
- Jeden chromozóm z páru od matky, jeden od otca
- Pre daný SNP s alelami  $a$ ,  $A$  môže byť **homozygot** ( $aa$  alebo  $AA$ ), alebo **heterozygot** ( $aA$ )
- Ak nejaká choroba zapríčinená alelou  $a$ , tak sa môže prejaviť iba pri homozygotoch  $aa$ , alebo aj pri heterozygotoch  $aA$ , alebo môže byť pri  $aa$  silnejšia ako pri  $aA$
- **Haplotyp**: kombinácia aliel rôznych SNPov na tom istom chromozóme (zdedená od jedného rodiča)  
Diploidný jedinec má teda dva haplotypy



## Testovanie asociácie jedného SNPu

### Kontingenčná tabuľka - počet haplotypov

Veľkosť psa vs. alela na pozícii chr15:44,228,468

	pôvodná alela	odvodená alela	spolu
malý pes (< 9 kg)	14	535	549
veľký pes (> 31 kg)	339	38	377
spolu	353	573	926



[Sutter a kol. 2007]

Štatisticky testujeme či sú riadky a stĺpce nezávislé (nulová hypotéza).

Ak nevyлúčime nulovú hypotézu, nepreukázali sme súvis SNPu s veľkosťou (môže ale existovať, možno treba viac dát)

Ak ju vylúčime, našli sme asociáciu, nemusí však ísť o príčinu

## Testovanie nezávislosti v kontingenčnej tabuľke

	pôvodná alela	odvodená alela	spolu
malý pes	14	535	549
veľký pes	339	38	377
spolu	353	573	926

**Fisherov test:** (Fisher's exact test) presný výsledok z hypergeometrického rozdelenia

**Chí-kvadrát ( $\chi^2$ ) test:** obľúbený približný test, vhodný ak máme vysoké počty

Používajú sa aj zložitejšie štatistické metódy/modely (napr. diploidný genóm, príbuzenské vzťahy, ...)

## Testovanie nezávislosti v kontingenčnej tabuľke $\chi^2$ testom

	alela $A$	alela $a$	spolu
malý pes ( $m$ )	14	535	549
veľký pes ( $v$ )	339	38	377
spolu	353	573	926

V nulovej hypotéze (nezávislosť riadkov a stĺpcov) máme:

$$\Pr(A) = 353/926 = 0.381, \Pr(a) = 0.619$$

$$\Pr(m) = 549/926 = 0.593, \Pr(v) = 0.407$$

$$\Pr(A, m) = \Pr(A) \Pr(m) = 0.226$$

$$\Pr(a, m) = \Pr(a) \Pr(m) = 0.367$$

$$\Pr(A, v) = \Pr(A) \Pr(v) = 0.155$$

$$\Pr(a, v) = \Pr(a) \Pr(v) = 0.252$$

Podľa nulovej hypotézy by sme teda čakali, že 926 haplotypov bude v tabuľke rozdelených v pomeroch 0.226:0.367:0.155:0.252

## Testovanie nezávislosti v kontingenčnej tabuľke $\chi^2$ testom

Skutočná tabuľka

$O_{i,j}$  (observed):

	A	a	spolu
malý	14	535	549
veľký	339	38	377
spolu	353	573	926

Očakávané podľa nulovej hypotézy

$E_{i,j}$  (expected):

	A	a	spolu
malý	209.3	339.8	549
veľký	143.5	233.4	377
spolu	353	573	926

**Spočítame veličinu**  $\chi^2 = \sum_{i \in \{m,v\}} \sum_{j \in \{A,a\}} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$

$$\chi^2 = (14 - 209.3)^2/209.3 + (535 - 339.8)^2/339.8 + (339 - 143.5)^2/143.5 + (38 - 233.4)^2/233.4 = 724.3$$

$\chi^2$  je určitá miera rozdielnosti tabuliek  $O$  a  $E$ .

Platí, že  $\chi^2 \geq 0$  a  $\chi^2$  je nula, iba ak sa tabuľky úplne zhodujú.

## Testovanie nezávislosti v kontingenčnej tabuľke $\chi^2$ testom

$O_{i,j}$  (observed):

	$A$	$a$	spolu
malý	14	535	549
veľký	339	38	377
spolu	353	573	926

$E_{i,j}$  (expected):

	$A$	$a$	spolu
malý	209.3	339.8	549
veľký	143.5	233.4	377
spolu	353	573	926

Spočítame veličinu  $\chi^2 = \sum_{i \in \{m,v\}} \sum_{j \in \{A,a\}} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} = 724.3$

Ak platí nulová hypotéza,  $\chi^2$  je približne z rozdelenia  $\chi^2(1)$ ,  
t.j. **chí kvadrát s jedným stupňom voľnosti**.

1 stupeň: ak poznáme  $E$  a 1 políčko z  $O$ , zvyšok  $O$  vieme dopočítať.

Šanca, že pri nulovej hypotéze nám náhodne vyjde  $\chi^2 \geq 724.3$  je  $1.6 \cdot 10^{-159}$  (P-hodnota)

Na **odmietnutie nulovej hypotézy** často používame  
prah  $P < 0.05$ , t.j.  $\chi^2 > 3.841$

## Závislosť medzi dvoma rôznymi SNPmi

Uvažujme SNP s alelami  $p/P$  a ďalší SNP s alelami  $q/Q$ .

Nameriame počty haplotypov  $pq, PQ, pQ, Pq$

**Príklad:** 2000 haplotypov (1000 jedincov)

	Q	q	
P	474	611	$\chi^2 = 184.78$ , P-hodnota $4.4 \cdot 10^{-42}$
p	142	773	

Stĺpce a riadky teda nie sú nezávislé, medzi SNPmi je závislosť

**Príklad 2:** Podobné pomery počtov, ale iba 30 haloptypov:

	Q	q	
P	7	9	$\chi^2 = 3.0867$ , P-hodnota 0.07893
p	2	12	

Nulovú hypotézu nevyhlúčime pre prah  $P < 0.05$  ( $\chi^2 > 3.841$ )

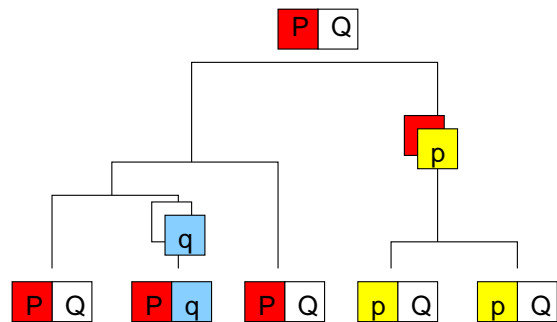
Ale pozor, pre takéto malé hodnoty  $\chi^2$  **nepresný**

## Ako vzniká závislosť medzi dvoma rôznymi SNPmi

### Na rozdielnych chromozómoch:

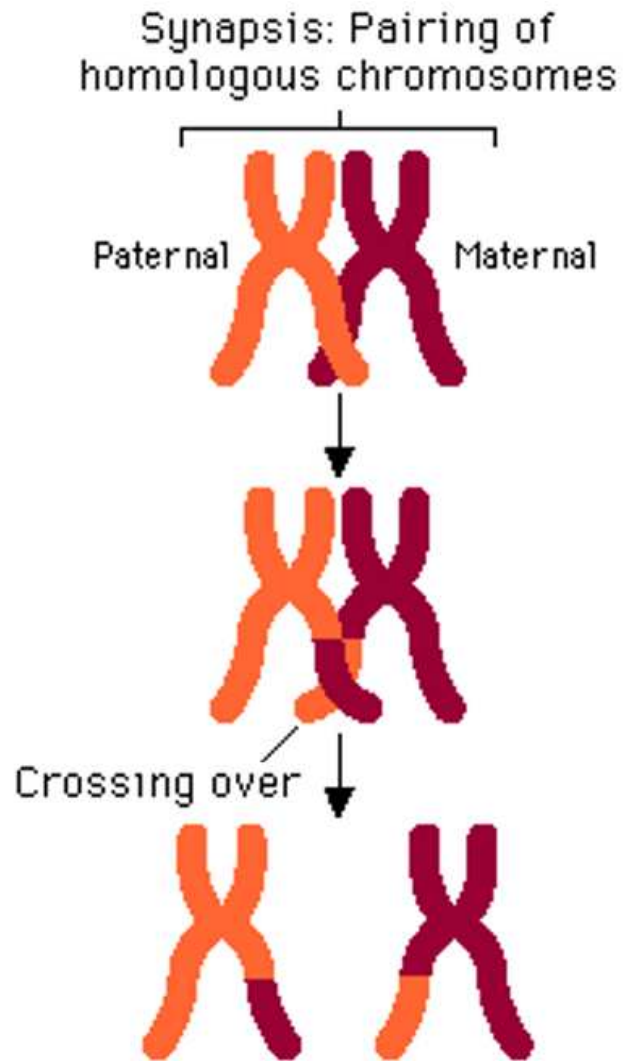
- Pravdepodobnosti výskytu jednotlivých alel sú nezávislé
- $\Pr(pq) = \Pr(p) \Pr(q)$ ,  $\Pr(PQ) = \Pr(P) \Pr(Q)$ , atď
- **väzbová rovnováha, linkage equilibrium (LE)**

### Blízko seba na tom istom chromozóme:



- Málokedy mutácia na to istom mieste 2x, zriedkavá rekombinácia
- Kombinácie nie sú úplne náhodné
- Korelácie medzi SNPmi  
⇒ **väzbová nerovnováha, linkage disequilibrium (LD)**

## Rekombinácia



Cca 1-3 **rekombinácie** v 1 ľudskom chromozóme počas meiózy (tvorba pohlavných buniek)

### Rekombinácia znižuje LD

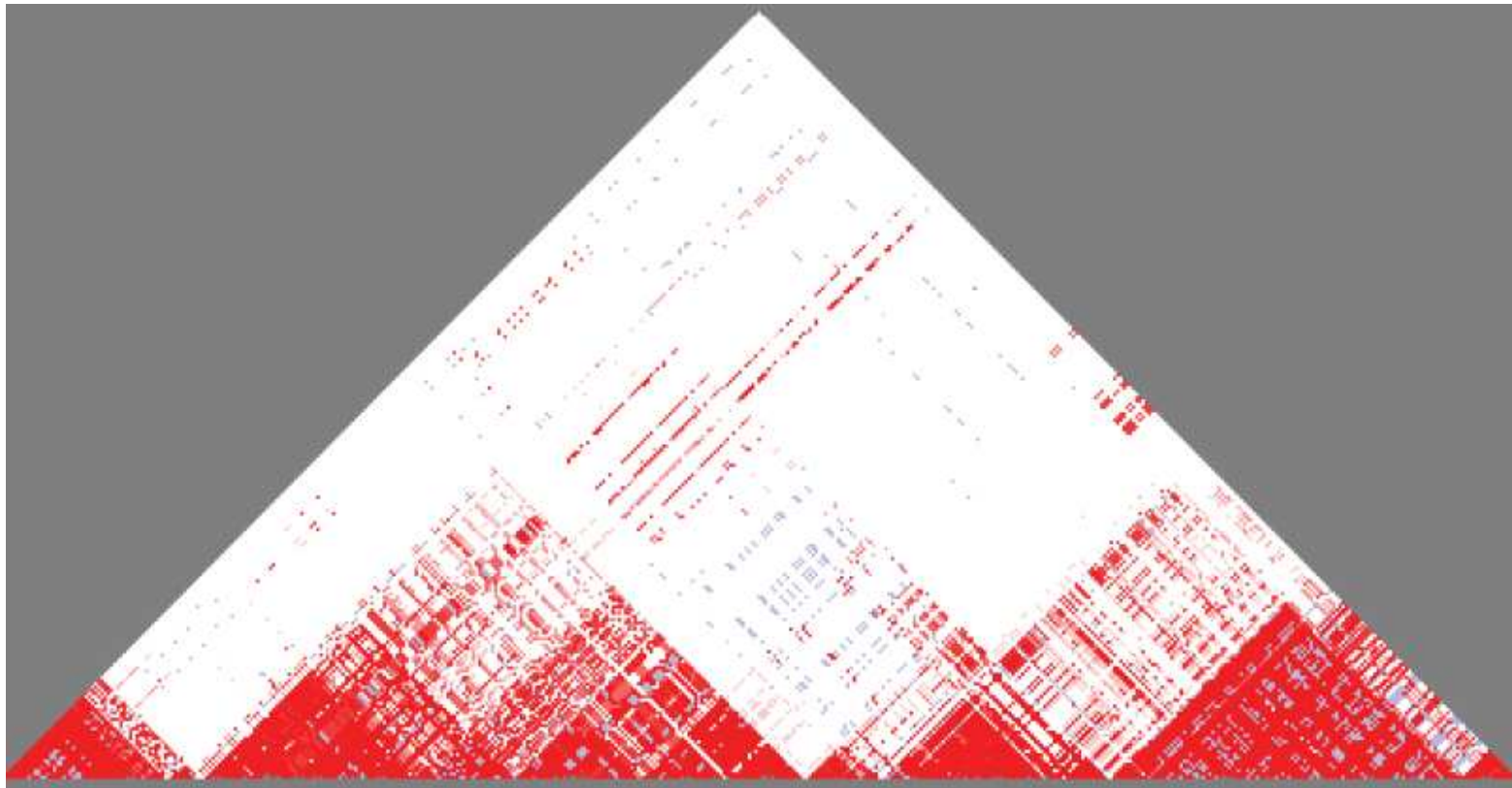
Ak predpokladáme rovnomernú rekombináciu:

- Čím vzdialenejšie SNPy, tým nižšie LD
- Čím staršie SNPy, tým nižšie LD
- Ďalšie aspekty: štruktúra populácie, prirodzený výber, rekombinačné hotspoty



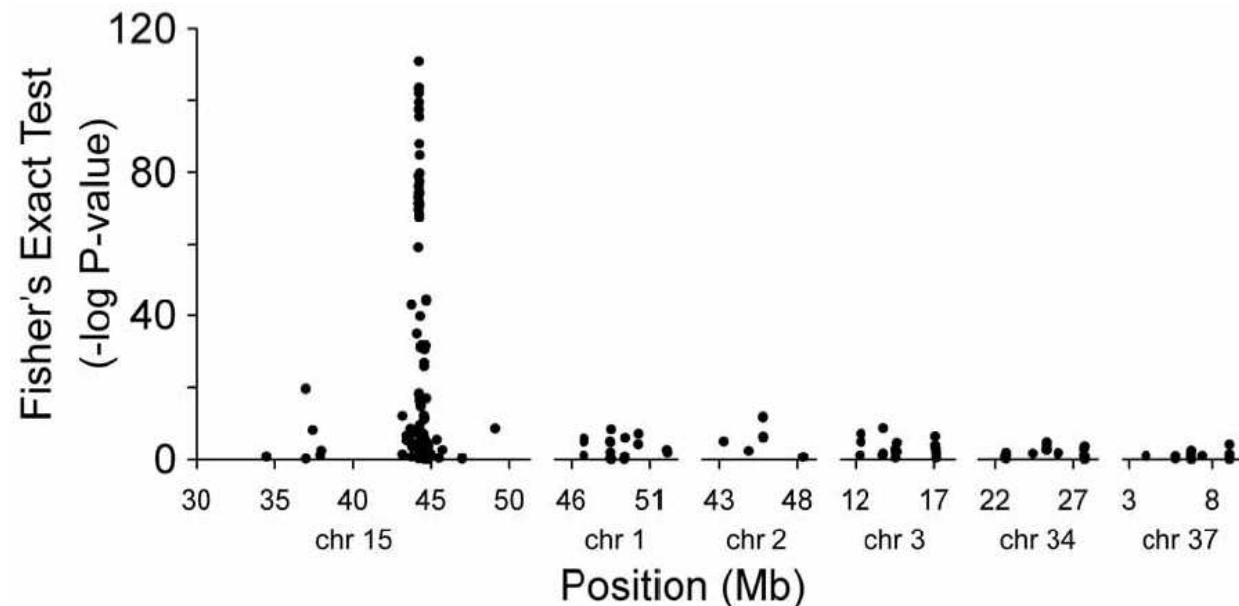
## Väzbová nerovnováha (LD) v ľudskom genóme

[The International HapMap Consortium, 2005]



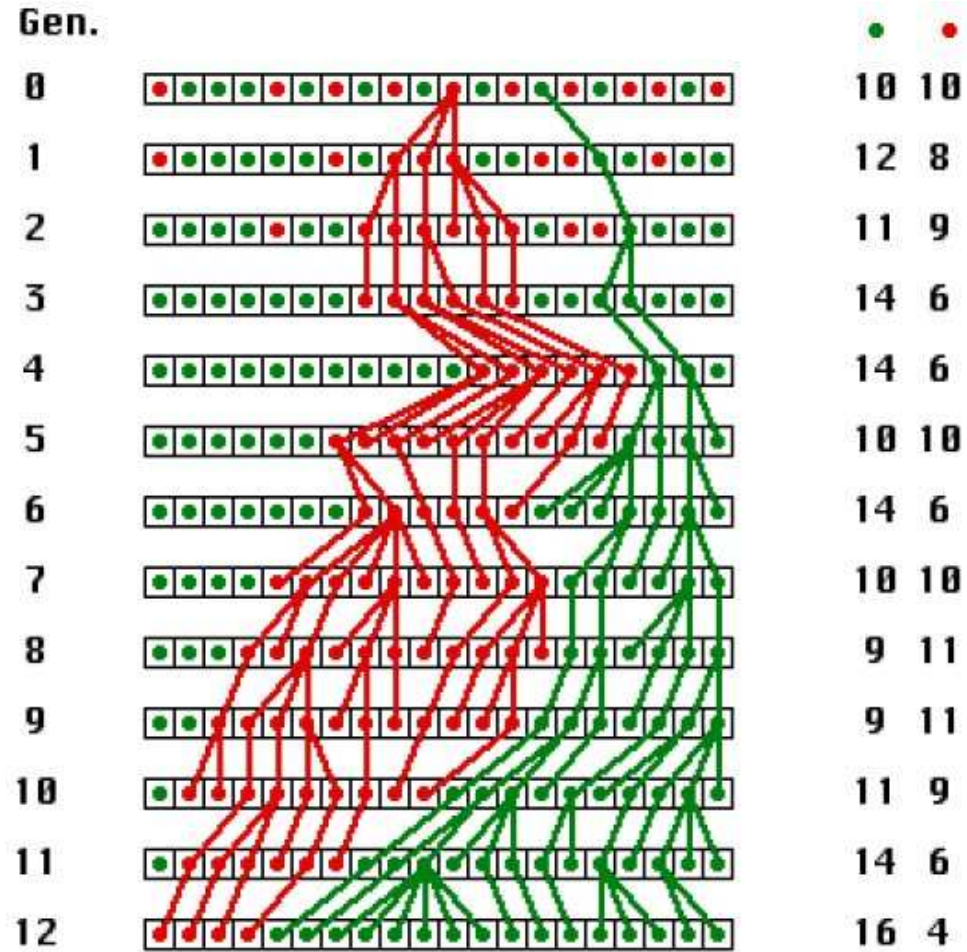
Región ENm014 (500kB, chr 7), 90 ľudí Utah

## Späť k psom: Hľadanie asociácií v celom genóme (Whole-Genome Association Scan, WGAS)



- V prípade štúdie veľkosti psov: WGAS identifikoval 84 kB región
- Pozíciu ďalej treba spresniť ďalšími experimentami
- **Malé LD bloky**  $\Rightarrow$  potreba veľkého rozlíšenia SNPov
- **Veľké LD bloky**  $\Rightarrow$  príliš veľké výsledné regióny

# Základný model populačnej genetiky: Wright-Fisherov model



## Životný cyklus SNPov vo Wright-Fisherovom modeli

- Populácia  $N$  jedincov (stabilná veľkosť)
- Jedinec = jedna alela ( $A$  or  $a$ )
- Nová generácia vzniká “skopírovaním” náhodného rodiča (random mating), bez vplyvu prirodzeného výberu
- $X_t$ : počet jedincov s alelou  $a$  v generácii  $t$
- **Markovovský reťazec** so stavmi  $X_t \in \{0, 1, \dots, N\}$

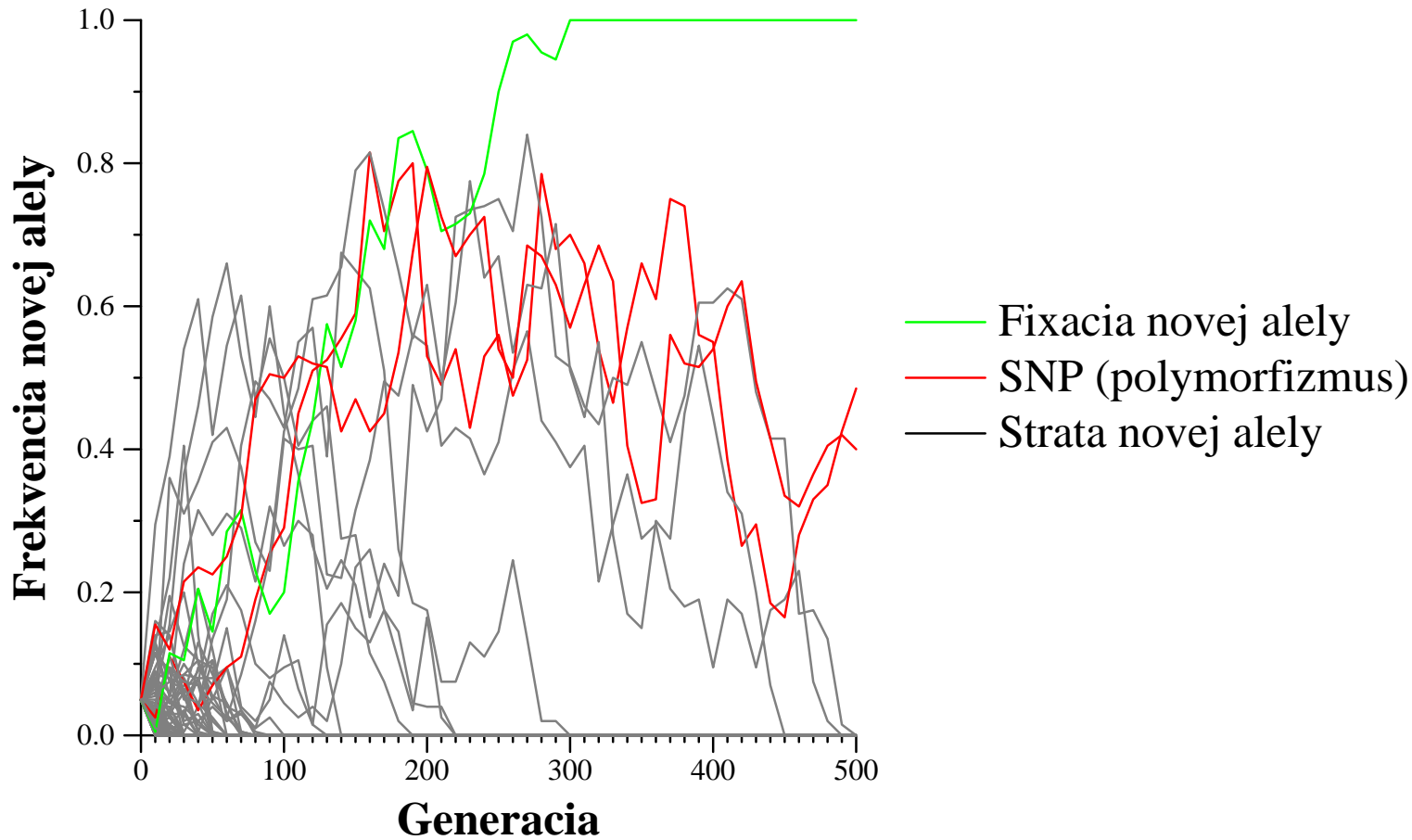
$$\Pr(X_t = j \mid X_{t-1} = i) = \left(\frac{i}{N}\right)^j \left(\frac{N-i}{N}\right)^{N-j} \binom{N}{j}$$

(Pravdepodobnosť, že v generácii  $t$  máme  $j$  kópií alely  $a$ , ak v generácii  $t - 1$  ich bolo  $i$ )

- Stavy 0 and  $N$  sú **pohlujúce**

# Náhodný genetický drift

$N = 200$ ,  $X_0 = 10$ , 500 generací



## Zložitejšie modely populácie

- **Mutácie** zavádzajú do populácie nové alely, ktoré po čase náhodným genetickým driftom zaniknú, alebo ovládnu populáciu (fixation).
- Rýchlosť procesu je ovplyvnená efektami ako **štruktúra populácie** alebo **prirodzený výber**
- $\Rightarrow$  Zložitejšie pravdepodobnostné modely

## Analýza histórie populácie na základe pravdepodobnostných modelov

### Typické parametre pravdepodobnostného modelu:

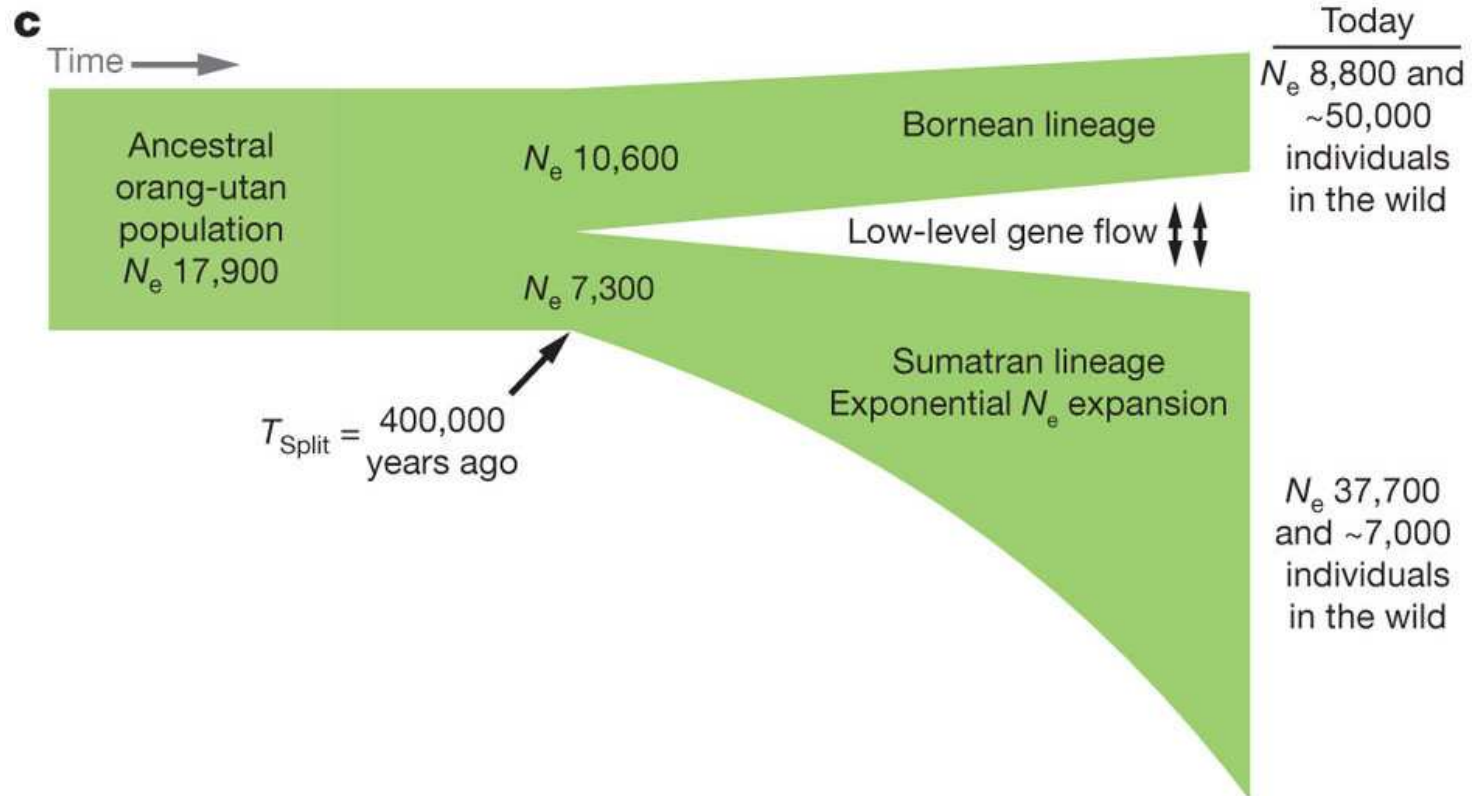
- efektívna veľkosť populácie
- frekvencia rekombinácie a mutácie

### Parametre ovplyvňujú pozorované dáta:

- Frekvencie SNPov (frekvencia menšinovej alely)
- Heterozygocita u diploidných jedincov
- Počet a veľkosť LD blokov

**Štandardný prístup:** Snažíme sa nájsť parametre modelu, ktoré najlepšie vysvetľujú pozorované dáta u osekvenovaných jedincov.

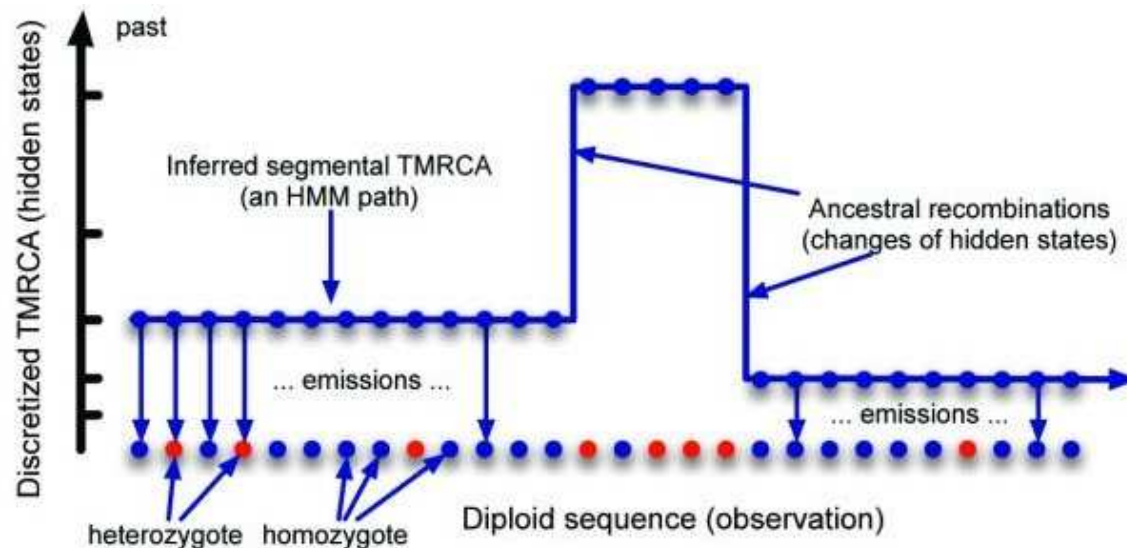
## Príklad: Populačná história orangutánov





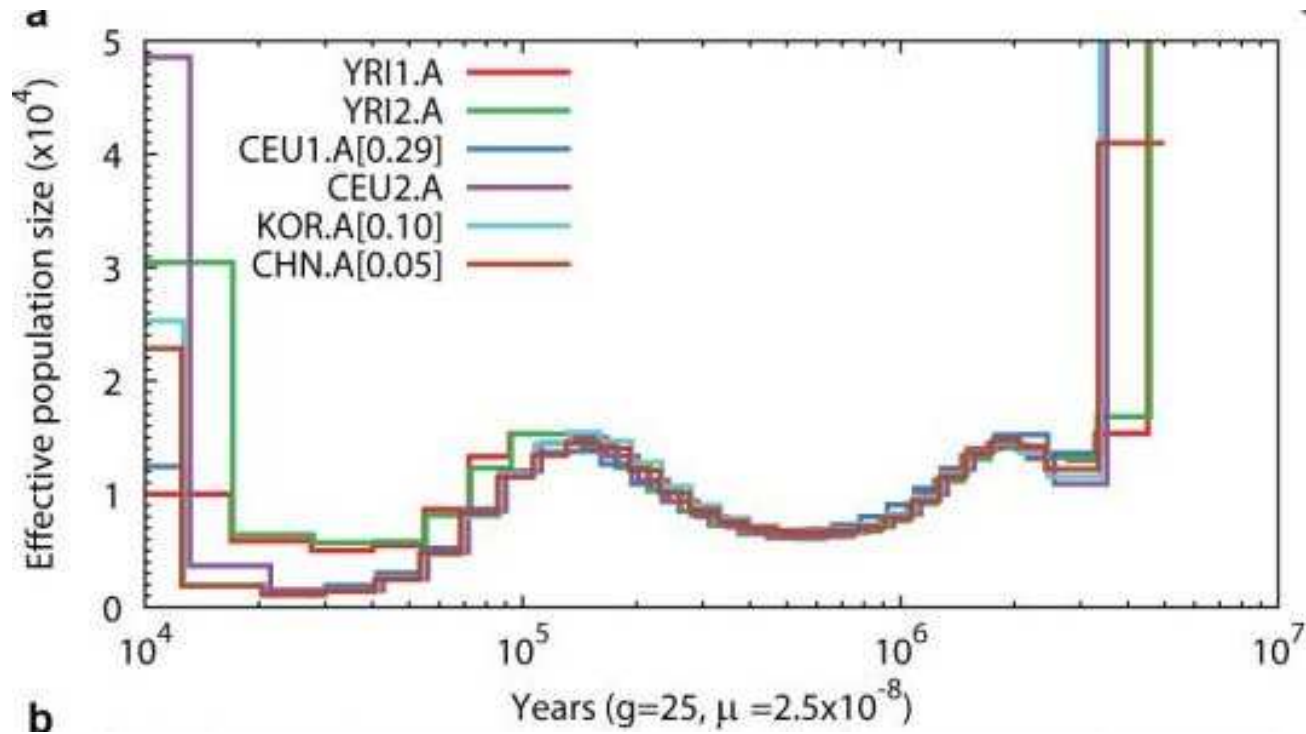
## História ľudskej populácie z genómu jedinca (Li, Durbin 2011)

- **Parametre modelu:** história vývoja efektívnej veľkosti ľudskej populácie v čase
- **Pozorované štatistiky:**
  - rozdelenie veľkostí rekombinačných blokov
  - rozdelenie časov ku najbližšiemu spoločnému predkovi (TMRCA)



## Príklad: História ľudskej populácie z genómu jedinca

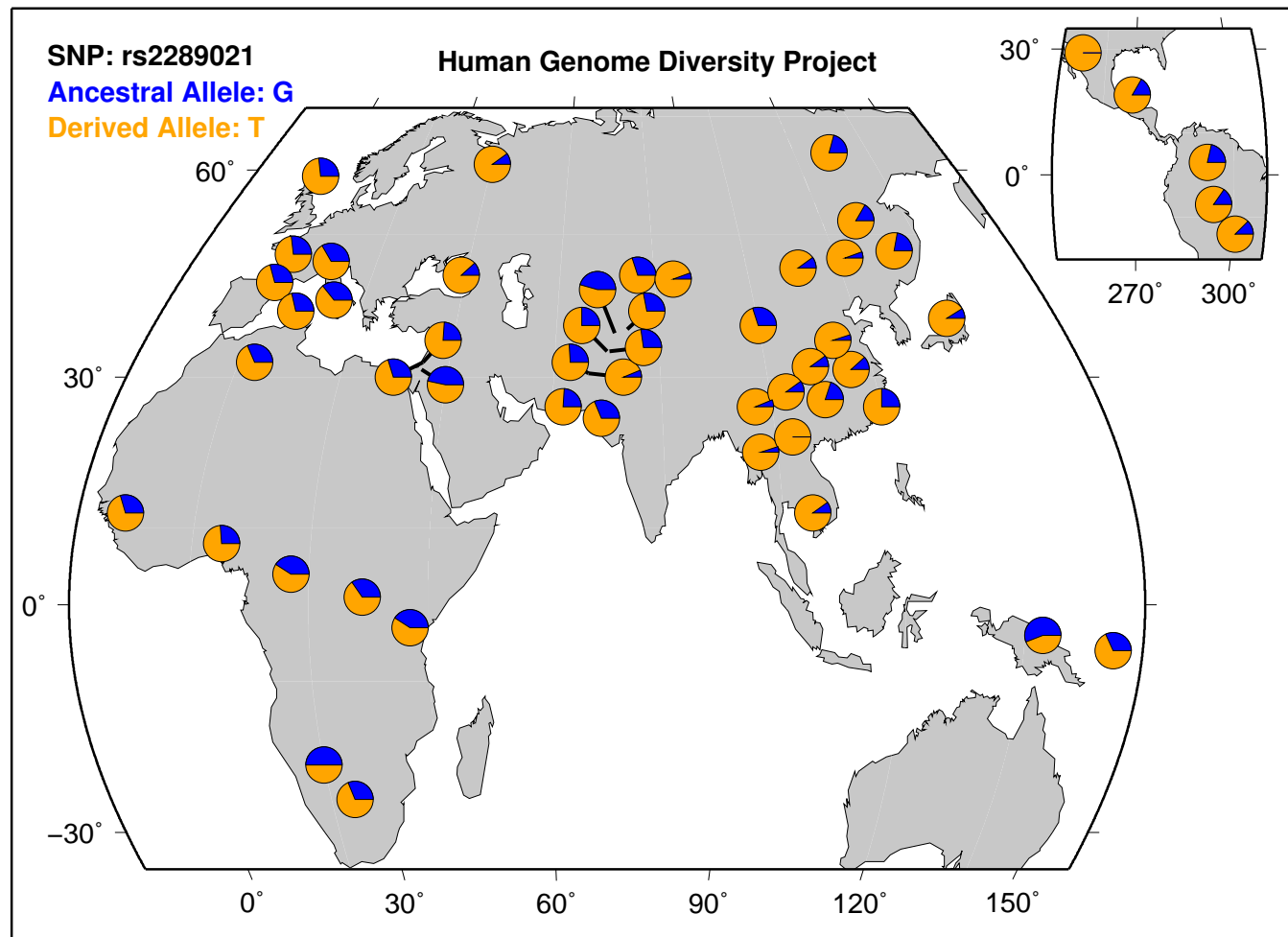
**Úloha:** Nájdi históriu vývoja efektívnej veľkosti ľudskej populácie, ktorá najlepšie vysvetľuje pozorované štatistiky



## Štruktúra populácie

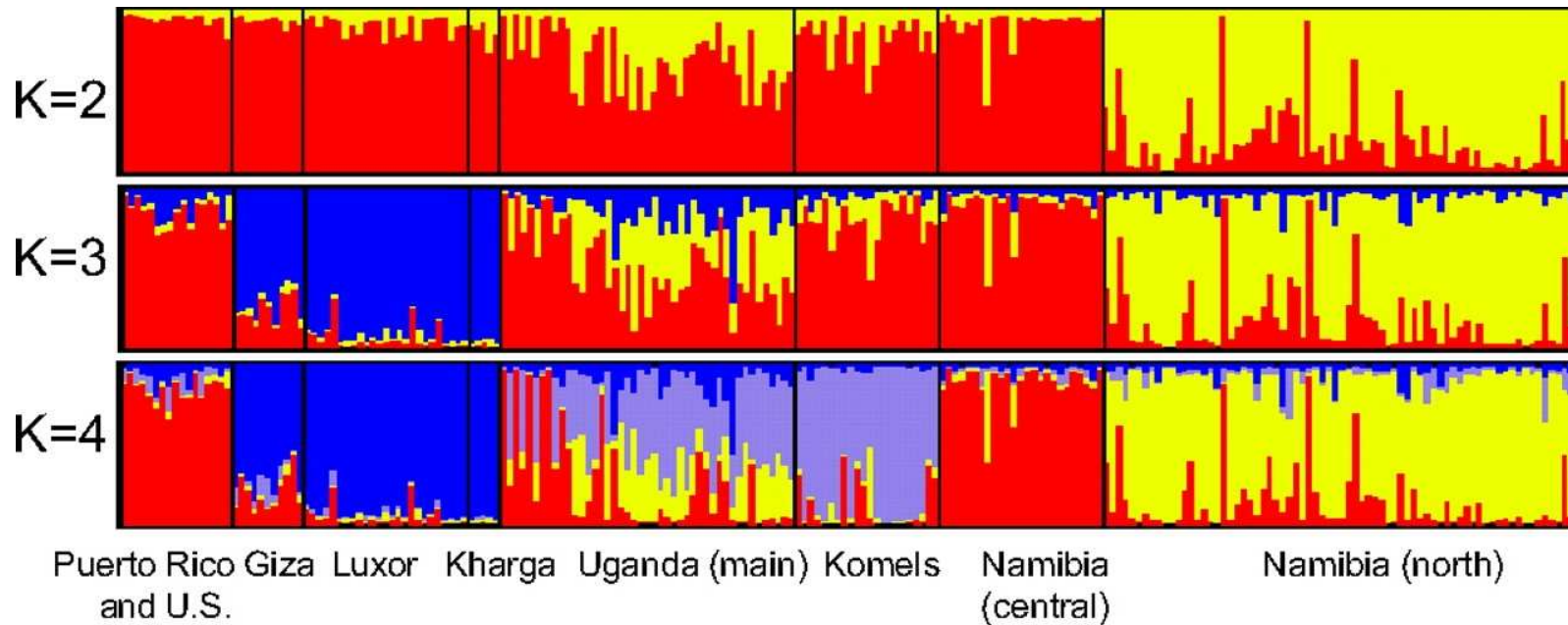
- Doteraz sme predpokladali, že nová generácia vzniká **náhodným párovaním** (random mating)
- Väčšina organizmov sa vyvíja v **subpopuláciách**, s obmedzeným prenosom genetického materiálu medzi subpopuláciami
- Frekvencie toho istého SNPu v dvoch subpopuláciách môžu byť značne odlišné
- $\Rightarrow$  “falošné” korelácie medzi SNPami (napr. aj medzi chromozómami), ak pracujeme s viacerými subpopuláciami naraz
- $\Rightarrow$  chybné výsledky pri LD a WGAS

# Príklad: frekvencie alel jedného konkrétneho SNPu u ľudí v rôznych častiach sveta



zdroj: genome.ucsc.edu

## Štruktúra populácie psov



Boyko et al. PNAS 2009; software STRUCTURE Pritchard et al. Genetics 2000

- Program STRUCTURE rozdelí populáciu na  $K$  subpopulácií (farby)
- Každý stĺpec je jedinec z populácie
- Pomer farieb zodpovedá pomeru SNPov z každej z  $K$  populácií

## Ako funguje STRUCTURE?

- **Vstup:** Vzorka haplotypov  $X$ , ktorú chceme rozdeliť do  $K$  subpopulácií
- Definujeme stochastický model s nasledujúcimi premennými:
  - $P_{i,j}$  - frekvencia SNPu  $j$  v subpopulácii  $i$
  - $Q_i$  - aká časť SNPov v haplotype  $i$  patrí ku ktorej subpopulácii
  - $Z_{i,j}$  - priradenie subpopulácie SNPu  $j$  v haplotype  $i$
- Model definuje  $\Pr[X | P, Q, Z]$  a apriórne rozdelenie pre  $P, Q$
- **Výstup:**  $E[Q | X]$

## Algoritmus Markov Chain Monte Carlo (MCMC)

- Premenné:
  - $P_{i,j}$  - frekvencia SNPu  $j$  v populácii  $i$
  - $Z_{i,j}$  - priradenie subpopulácie SNPu  $j$  v haplotype  $i$
  - $Q_i$  - aká časť SNPov v haplotype  $i$  patrí ku ktorej populácii
- Začni s hodnotami  $P^{(0)}, Z^{(0)}, Q^{(0)}$ . V každej ďalšej iterácii získame novú náhodnú vzorku:
  - Vyber náhodnú vzorku  $P^{(i)}, Q^{(i)}$  z distribúcie  $\Pr(P, Q | X, Z^{(i-1)})$
  - Vyber náhodnú vzorku  $Z^{(i)}$  z distribúcie  $\Pr(Z | X, P^{(i)}, Q^{(i)})$
- Pre vhodné  $m, c$ , priemer postupnosti

$$Q^{(m)}, Q^{(m+c)}, Q^{(m+2c)}, \dots$$

konverguje k hodnote  $E[Q | X]$

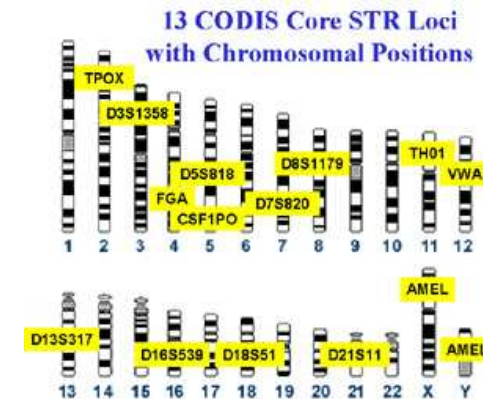
## Zhrnutie

- **SNPy (single nucleotide polymorphisms)** priebežne vznikajú a zanikajú v populáciách
- Ich frekvencia ovplyvnená navyše prirodzeným výberom
- Bez rekombinácie korelácia medzi SNPmi na tom istom chromozóme (**linkage disequilibrium**)
- Rekombinácie vytvárajú v genóme LD bloky
- Prítomnosť LD blokov vplýva na výsledky mapovania asociácií znakov (**whole-genome association mapping**)
- Pravdepodobnostné modely veľkosti LD blokov, frekvencií alel, heterozygocity a pod. nám môžu veľa prezradiť o **histórii populácie**
- Pri analýzach treba brať do úvahy **štruktúru populácie**, ktorú možno odhadnúť pomocou výpočtových metód



## Ďalšie typy polymorfizmov

- **Krátke indely**
- **Mikrosatelity a minisatelity** (jednoduché krátke opakujúce sa sekvencie)  
13 lokusov ako štandardný “odtlačok” pre porovnávanie DNA vzoriek na súdoch v USA



- **Transpozóny** (Alu, LINE, SINE)  
Alu má cca milión kópií, cca 1 nová kópia na 20 novorodencov
- **Veľké úseky s variabilnou multiplicitou** (Large scale copy number variations)