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

Vyučujú:

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

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

Jana Černíková, M-25, jana.cernikova@fmph.uniba.sk

Web: https://fmfi-compbio.github.io/mbi/

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, poznámky a videá na webstránke.

Časy a miestnosti

- Prednáška štvrtok 15:40-17:10 F1-109
- Cvičenia informatici štvrtok 14:00-15:30 F1-109
- Cvičenia biológovia štvrtok 17:20-18:50 M-217
  (ak prednáška skončí skôr, cvičenia začnú po krátkej prestávke)

“Informatici”: študenti informatiky, bioinformatiky, aplikovanej informatiky, dátovej vedy; zapíšte si 1-BIN-301

“Biológovia”: študenti z PriFUK, študenti biomedicínskej fyziky; zapíšte si 2-AIN-501

Ostatní: poraďte sa, do ktorej skupiny sa zaradiť

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%

Kvízy 10% (každý týždeň 1 bod)

INF: Skúška 50% BIO: Projekt 50%

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

INF: Zo skúšky treba aspoň polovicu bodov

BIO: Aktívna účasť na cvičeniach

- Dve verzie DÚ: biologická a informatická
- Journal club: čítanie 1 článku v skupine a správa (prípadne nepovinná prezentácia)
- Na skúške povolený ťahák 2 listy A4, môže mať ústnu časť
- Neodpisovať!

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

Týždenné kvízy

- Cca 5 krátkych otázok týkajúcich sa prednášky aj cvičení
- Vypĺňajte od štvrtka 19:00 do ďalšej stredy 22:00
- Linku na Moodle s kvízmi nájdete na stránke predmetu
- Cieľ: pripomenúť si aspoň základné pojmy z prednášky a cvičení
- **Prvý kvíz už tento týždeň**

Príklad z nášho výskumu

Kosmáč bielofúzy

(common marmoset, Callithrix jacchus, štvrť kila, 18cm)

Genóm osekvenovaný 2007

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

Analýza publikovaná v roku 2014

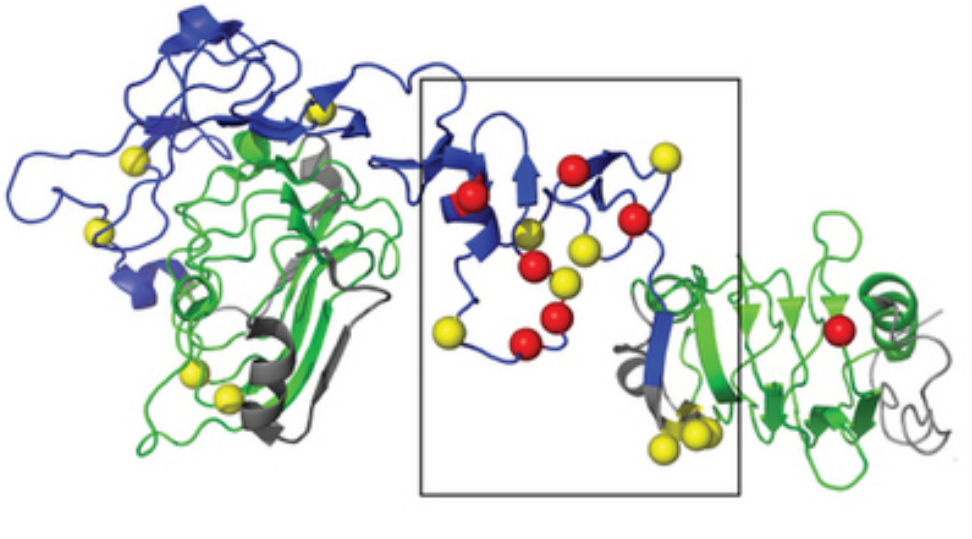
IGF1R: Insulin-like growth factor 1 receptor

Proteín prechádza cez cytoplazmatickú mebránu na povrchu bunky

Po naviazaní hormónov IGF1, IGF2 signalizuje dovnútra bunky

Súvisí s rastom a delením bunky, rastom organizmu, rakovinou

```
human     RDFCANILSAESSDSEGFVIHDGECMQECPSGFIRNGSQSMYCIPCEGPCPKVC-EEEKKTK
chimp     RDFCANILSAESSDSEGFVIHDGECMQECPSGFIRNGSQSMYCIPCEGPCPKVC-EEEKKTK
orang     RDFCANILSAESSDSEGFVIHDGECMQECPSGFIRNGSQSMYCIPCEGPCPKVC-EEEKKTK
macaque   RDFCANILSAESSDSEGFVIHDGECMQECPSGFIRNGSQSMYCIPCEGPCPKVC-EEEKKTK
marmoset  RQFCASIVSSENSENNKFVIHDGECMQDCPSGFIRDTTHSMQCIPCKGPCPKVC-D-EQMAK
mouse     RDFCANIPNAESSDSDGFVIHDDECMQECPSGFIRNSTQSMYCIPCEGPCPKVCGDEEKKTK
rat       RDFCANIPNAESSDSDGFVIHDGECMQECPSGFIRNSTQSMYCIPCEGPCPKVCGDEEKKTK
dog       RDFCANIPSAESSDSEGFVIHDGECMQECPSGFIRNGSQSMYCIPCEGPCPKVC-EEEKKTK
```



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)

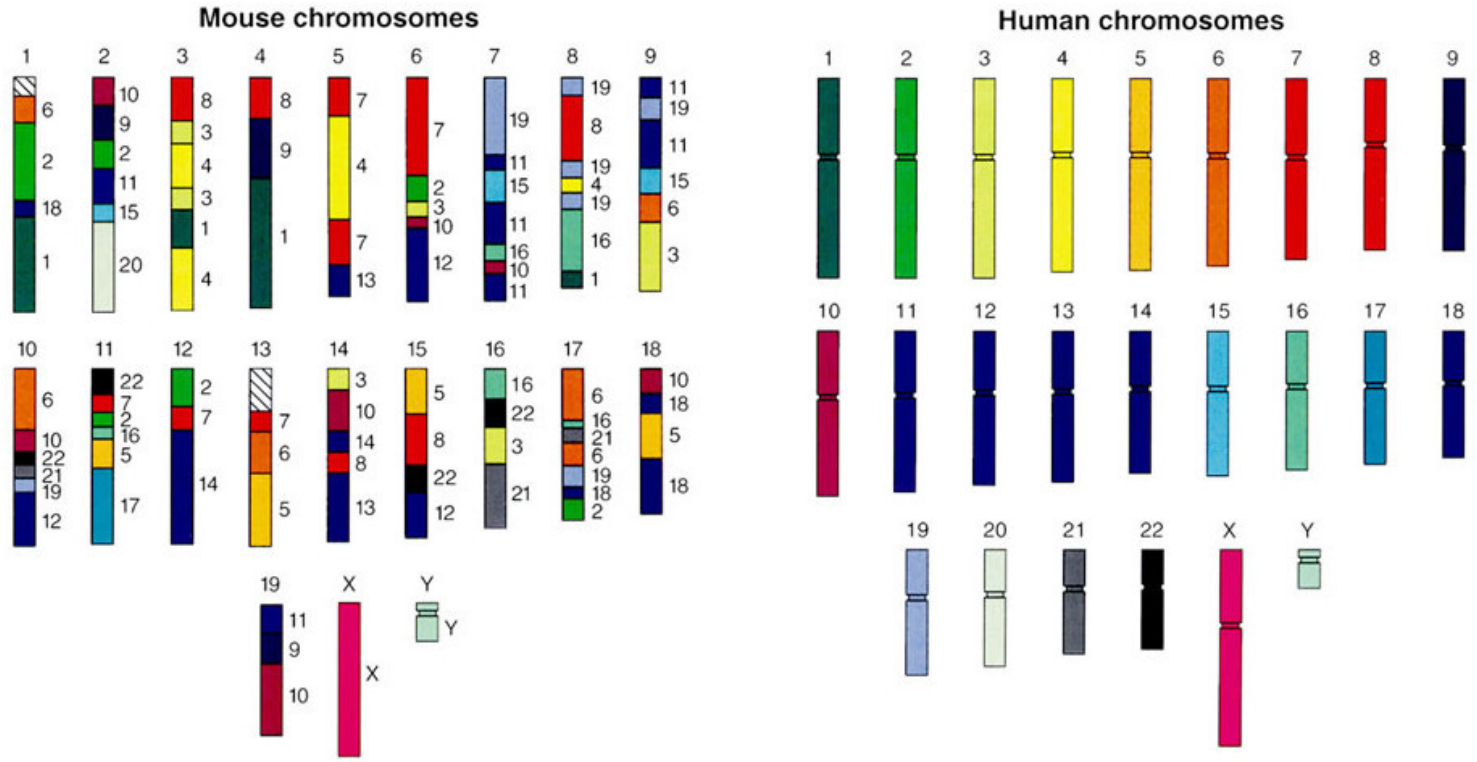
- Každé miesto v genóme prečítame viackrát (u kosmáča priemer 6×)

- 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, . . .)

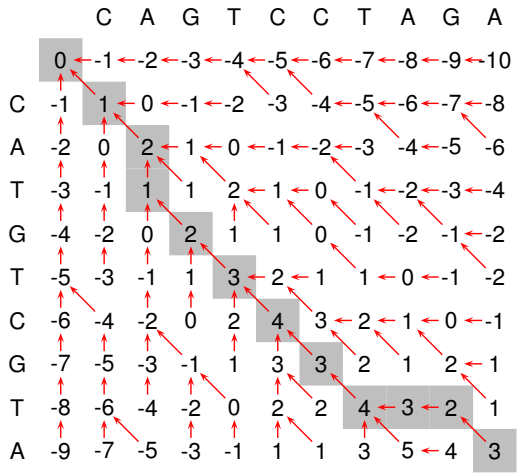


2. Nájdenie zarovnaní s inými genómami

- Hľadáme podobnosti medzi DNA sekvenciami

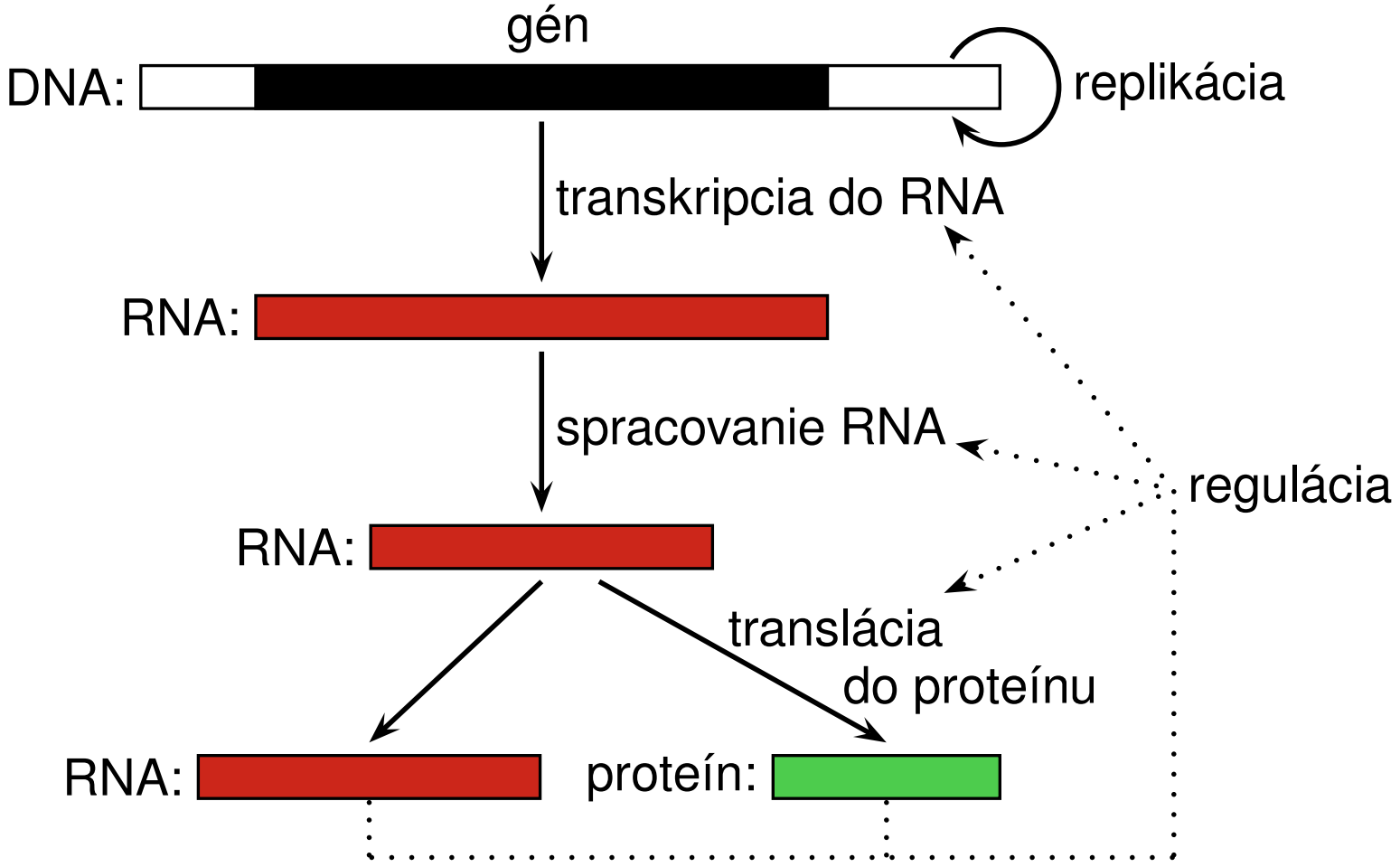
```
  Human AGTGGCTGCCAGGCTG---GGATGCTGAGGCCTTGTTTGCAGGGA
 Rhesus AGTGGCTGCCAGGCTG---GGTTGCTGAGGCCTTGTTTGCCGGGA
  Mouse GGTGGCTGCCGGGCTG---GGTGGCTGAGGCCTTGTTGGTGGGGT
    Dog AGTGGCTGCCCGGCTG---GGTGGCTGAGGCCTTATTTGCAGGGA
Chicken AGTGGCTGCCAGTCTGCGCCGTGGCCGACGTCTTGCTCGGGGGAA
```

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



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

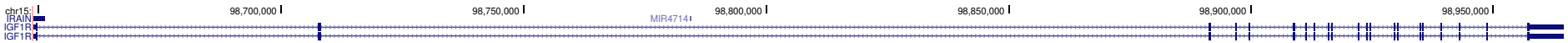
3. Hľadanie génov kódujúcich proteíny



Ktoré časti osekvenovaného genómu kódujú proteíny?

3. Hľadanie génov kódujúcich proteíny

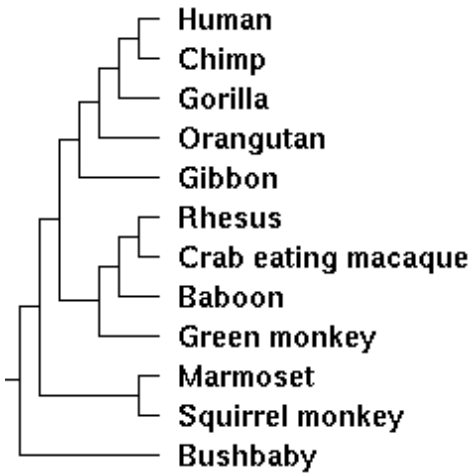
- Hľadanie ihly v kope sena: iba cca 1% ľudského genómu kóduje proteíny
- Kód pre jeden proteín rozbitý do veľa krátkych exónov
- Napr. IGF1R zaberá 315 569nt, z toho kóduje 4101nt v 21 exónoch



- Zoberieme známe gény, spravíme rôzne štatistiky

  potom hľadáme iné oblasti s podobným štatistickým profilom

4. Hľadanie génov s pozitívnym výberom



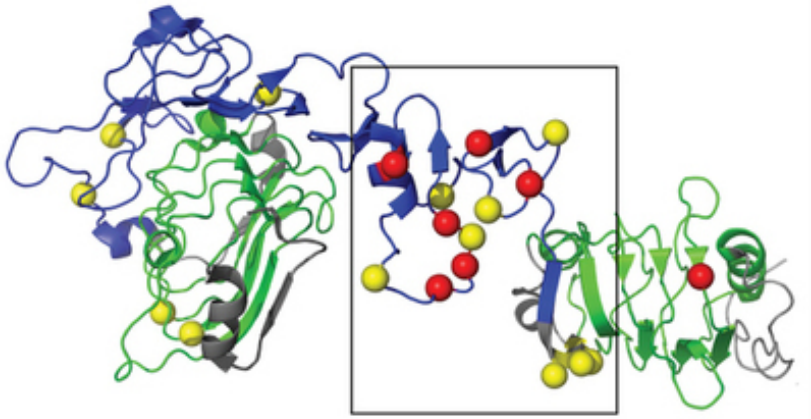
- Štúdium evolučných procesov
- V DNA vznikajú mutácie, tie však podliehajú prirodzenému výberu
- Väčšina náhodných zmien v proteíne je škodlivých, preto sa proteíny menia pomerne pomaly

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

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

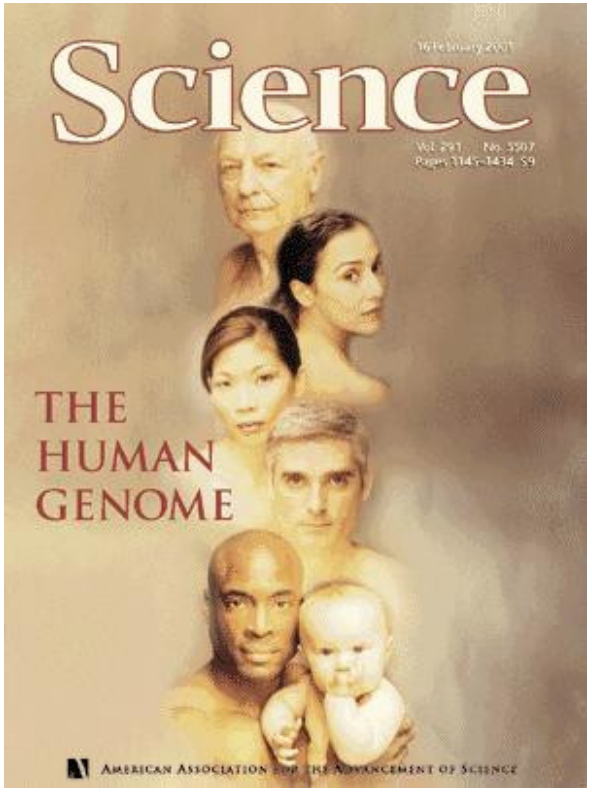


Sekvenovanie a zostavovanie genómov

(genome sequencing and assembly)

Tomáš Vinař

26.9.2024



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

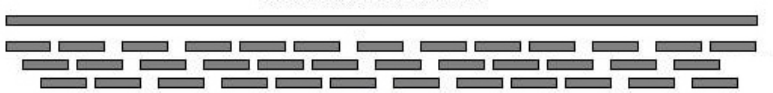
Prehľad sekvenovacích technológií

Technológia	Dĺžka čítania	Chybovosť	Za deň	Cena za GB
1. generácia				
Sanger	do 1000 bp	$< 1\%$	3 MB	$4 mil.
2. (next) generácia (cca od 2004)				
Illumina	300bp	$< 0.1\%$	2 TB	$3
3. generácia (cca od 2018)				
PacBio HiFi	cca 15 Kbp	$< 1\%$	360 GB	$15
Oxford Nanopore	5-100+kbp	$< 5\%$	50 GB	$10

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**
  - **dĺžka jednotlivých čítaní**
  - **pokrytie** (coverage) – koľko krát čítania pokrývajú genóm?



Formulácia problému (jednoduchá, ale nerealistická)

Najkratšie spoločné nadslovo (shortest common superstring)

Úloha: Daných je niekoľko reťazcov $S_1, S_2, \ldots S_n$ (čítania), nájdite **najkratší** reťazec S, ktorý obsahuje **každý** vstupný reťazec S_i ako (súvislý) **podreťazec**.

Motivácia: čo najviac využiť prekryvy medzi čítaniami

Príklad:

Vstup: $S_1 =$ GCCAAC, $S_2 =$ CCTGCC, $S_3 =$ ACCTTC

Výstup: $S =$ CCTGCCAACCTTC
(čítania spojené v poradí S_2, S_1, S_3)

Najkratšie spoločné nadslovo

- **Problém je NP ťažký**

  takže nepoznáme rýchly algoritmus, ktorý vždy nájde najlepšie riešenie

- **Jednoduchá heuristika:** opakovane nájdi 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:**

  Nájdené 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

Najkratšie spoločné nadslovo: Čo sme nezahrnuli do formulácie

- V sekvenovaní sa vyskytujú chyby
- Polymorfizmus
- Orientácia čítaní (vlákno, strand)
- Kontaminácia cudzou sekvenciou, chiméry
- Viac chromozómov, neúplné pokrytie čítaniami
- Repetitívna sekvencia (sequence repeats, opakovania)

  cca 50% ľudského genómu

  Príklad: 10xTTAATA, 10xATATTA, 3xTTAGCT

  TTAATATTAGCT?

  TTAATATTAATATTAATATTAATATTAGCT?

  TTAATATTA + ATATTAGCT?

Čo sme nezahrnuli do formulácie: kvalita báz

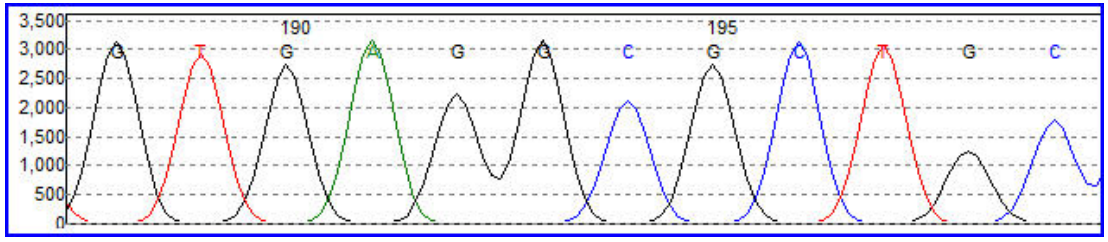
- K čítaniam máme väčšinou informáciu o **kvalite báz**

  Aká je pravdepodobnosť, že daná báza je správna?

- Báza s kvalitou $q \Rightarrow$ pravdepodobnosť chyby $10^{-q/10}$

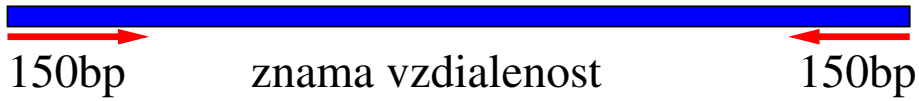
  napr. báza s $q = 40$ je správna s pr. 99.99%

Príklad výsledku Sangerovho sekvenovania (trace):



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

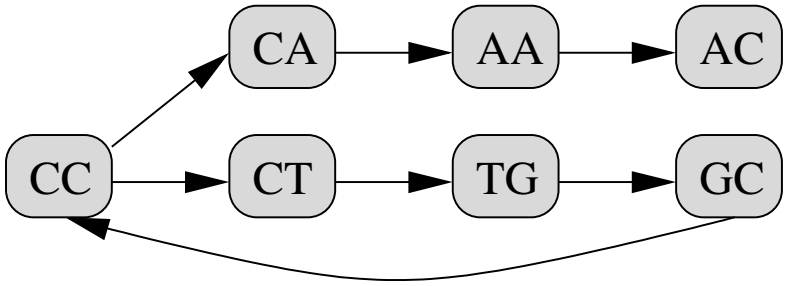
Konzervatívny prístup (radšej menej pospájať, ale nerobiť chyby)

Najkratšie spoločné nadslovo: Zhrnutie

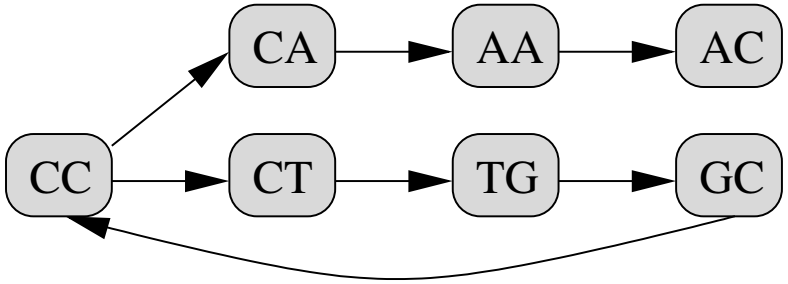
- Nerealistická formulácia, ťažký výpočtový problém
- Ale teoretický problém môže poskytnúť nejaký posun k pochopeniu skutočného problému
- Overlap-Layout-Consensus prístup
  motivovaný greedy algoritmami pre najkratšie spoločné nadslovo
  (budúci týždeň)

Skladanie krátkych čítaní: de Bruijnove 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 Bruijnove grafy?



- jediný chromozóm a žiadne "nejednoznačné" k-tice

  $\Rightarrow$ zostavenie = **Eulerovská cesta**

  (cesta v grafe, ktorá použije každú hranu práve raz)

- Eulerovskú cestu možno nájsť v čase $O(m+n)$

- v realistickom prípade:

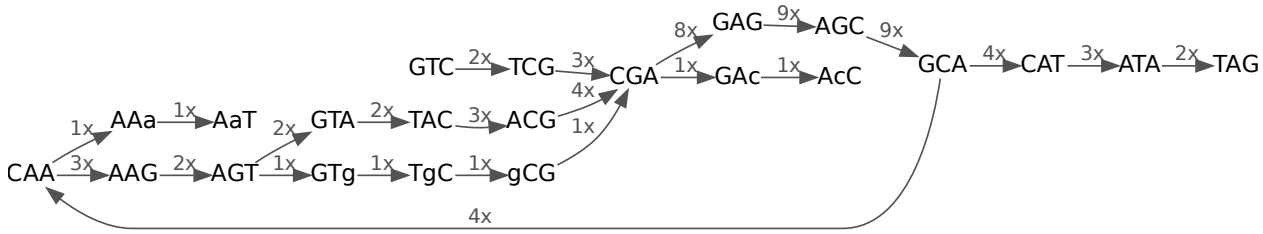
  zostavenie genómu zodpovedá niekoľkým

  pochôdzkam v de Bruijnovom grafe (nazývame **kontigy**),

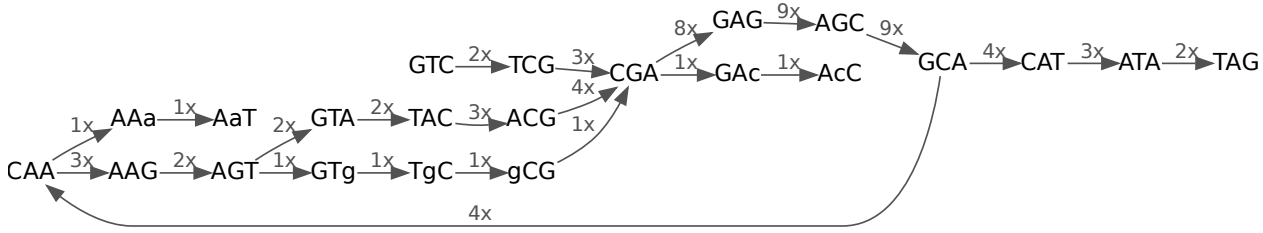
  ktoré dohromady pokrývajú veľkú časť hrán

Príklad: sada čítaní a zodpovedajúci deBruijnov graf

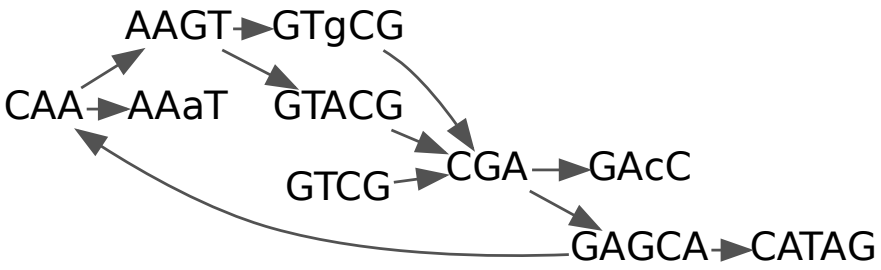
```
GTCGAGCAAGTACGAGCATAG
 TCGAGCA      AGCATAG
    AGCAAaT   AGCATAG
GTCGAcC  GTACGAG
GTCGAGC   TACGAGC
  CGAGCAA  ACGAGCA
        AGTgCGA
      CAAGTAC
     GCAAGTA GAGCAT
   GAGCAAG   GAGCATA
          TACGAGC
```



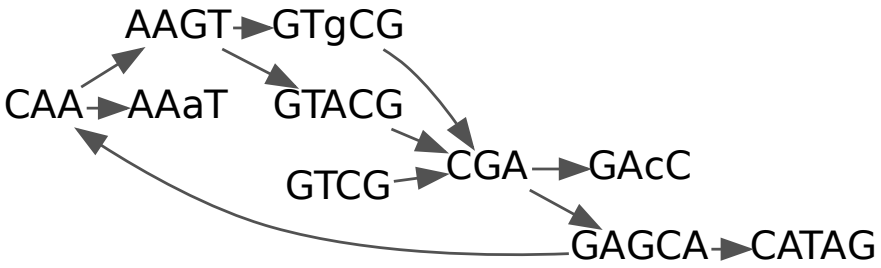
Príklad: zjednodušovanie de Bruijnovho grafu



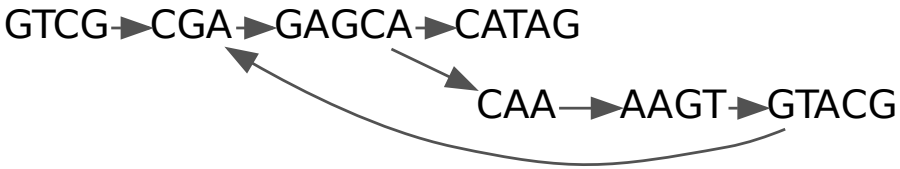
Spojíme jednoznačné cesty do vrcholov



Príklad: odstraňovanie chýb z de Bruijnovho grafu



Odstránenie chýb (výbežkov a bublín s nízkym pokrytím)



Spájaním dostaneme 4 kontigy (pôv. GTCGAGCAAGTACGAGCATAG)

Typické výsledky zostavovania genómov

- Veľa **kratších kontigov**,
  niekedy spájané 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, $70\times$ 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: rozsekneme všetky zle spojené kontigy

Zhrnutie

- Sekvenovanie genómu je zložitý proces, v ktorom hrá bioinformatika dôležitú úlohu
- Illumina nízka cena, krátke čítania
- 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 najdôležitejsiu úlohu pri tom, ako fragmentovaný bude výsledok:
  - pre Sanger: 7-10$\times$ pokrytie
  - pre NGS: 40-70$\times$ pokrytie
  - pre 3. generáciu: 30$\times$ pokrytie

História sekvenovania genómov

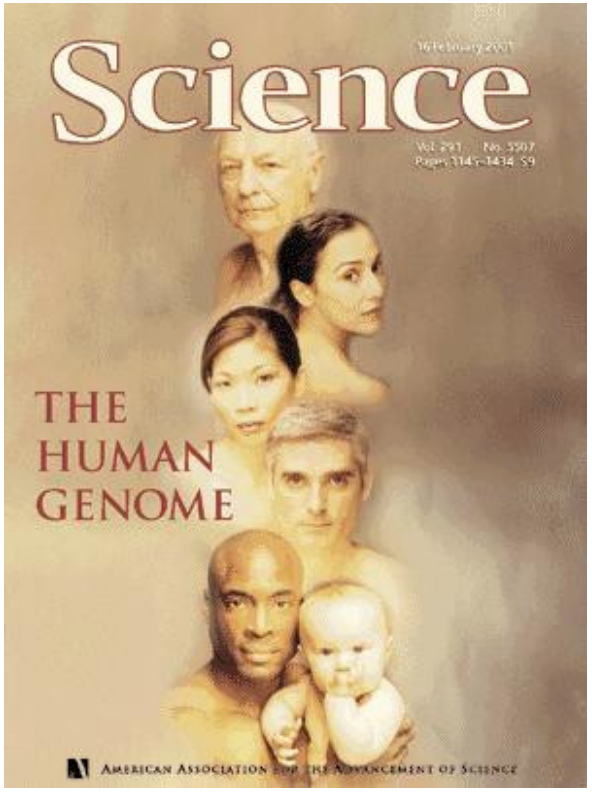
1976	MS2 (RNA vírus) 40 kB
1988	projekt sekvenovania ľudského genómu (15 rokov)
1995	baktéria *H. influenzae* 2 MB, shotgun (TIGR)
1996	*S. cerevisiae* 10 MB, BAC-by-BAC (Belgicko, Británia)
1998	*C. elegans* 100 MB, BAC-by-BAC (Wellcome Trust)
1998	Celera: ľudský genóm do troch rokov!
2000	*D. melanogaster* 180 MB, shotgun (Celera, Berkeley)
2001	2x ľudský genóm 3 GB (NIH, Celera)
po 2001	Myš, potkan, kura, šimpanz, pes,. . .
2007	Watsonov a Venterov genóm (454)
2012	1000 ľudských genómov
2021	3,5 milióna genómov SARS-CoV-2
2021	UK Biobank 200,000 ľudských genómov + veľa ďalších dát
2022	Naozaj dokončený ľudský genóm (telomere to telomere)
2024	All of US 246,000 ľudských genómov + zdravotné záznamy

Sekvenovanie a zostavovanie genómov

(časť 2 - dlhé čítania)

Tomáš Vinař

3.10.2024



Oznamy

- Nezabudnite na pravidelný kvíz v Moodli.
- Dnes kratšia prednáška, biológovia budú mať dlhšie cvičenia v F1 109.
- Budúci týždeň informatici budú mať cvičenia aj prednášku vo forme videa.
- Biológovia príďte budúci týždeň 15:40, s Jankou Černíkovou prejdete prvú polovicu prednášky a cvičení, druhé polovice video.

Prehľad sekvenovacích technológií

Technológia	Dĺžka čítania	Chybovosť	Za deň	Cena za GB
1. generácia				
Sanger	do 1000 bp	$< 1\%$	3 MB	$4 mil.
2. (next) generácia (cca od 2004)				
Illumina	300bp	$< 0.1\%$	2 TB	$3
3. generácia (cca od 2018)				
PacBio HiFi	cca 15 Kbp	$< 1\%$	360 GB	$15
Oxford Nanopore	5-100+kbp	$< 5\%$	50 GB	$10

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 5% robí de Bruijnov graf neprehľadným
    (pre $k = 31$, **každý** k-mer v priemere 1-2 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 zarovnávania sekvencií (nasledujúca prednáška)
- prekryvy **budú obsahovať chyby**
  (v našom prípade cca 1 chyba na 20 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:

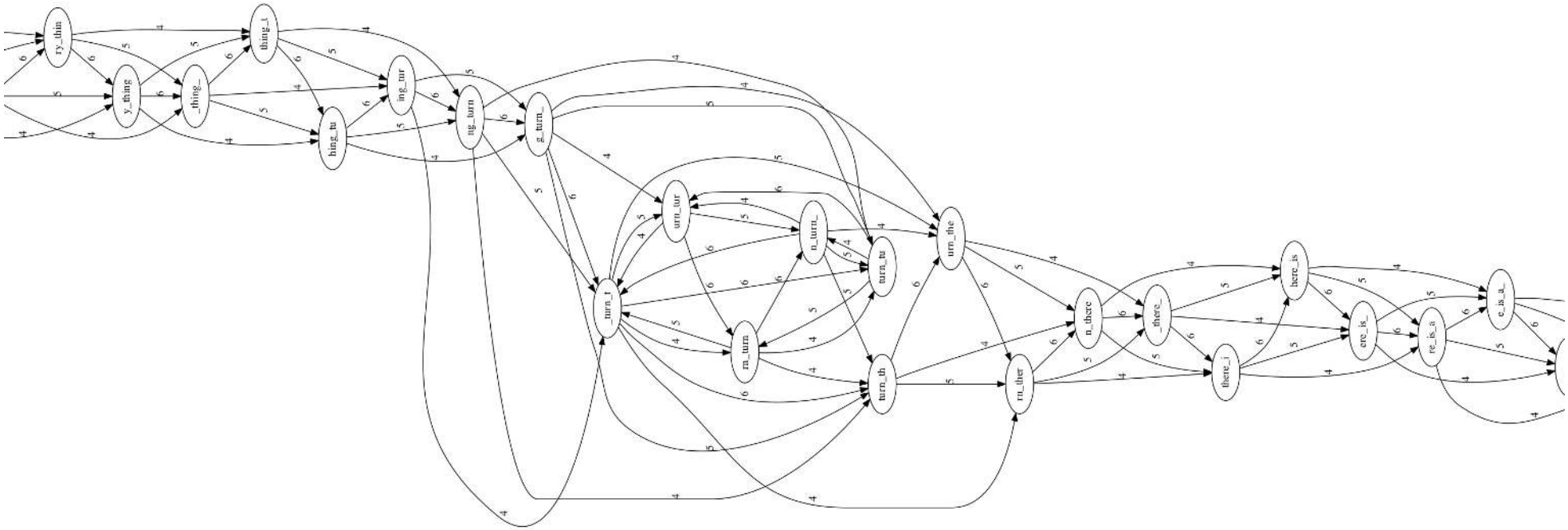
`to_every_thing_turn_turn_turn_there_is_a_season`

čítania dĺžky 7 písmen, minimálny prekryv 4

Príklad:

`to_every_thing_turn_turn_turn_there_is_a_season`

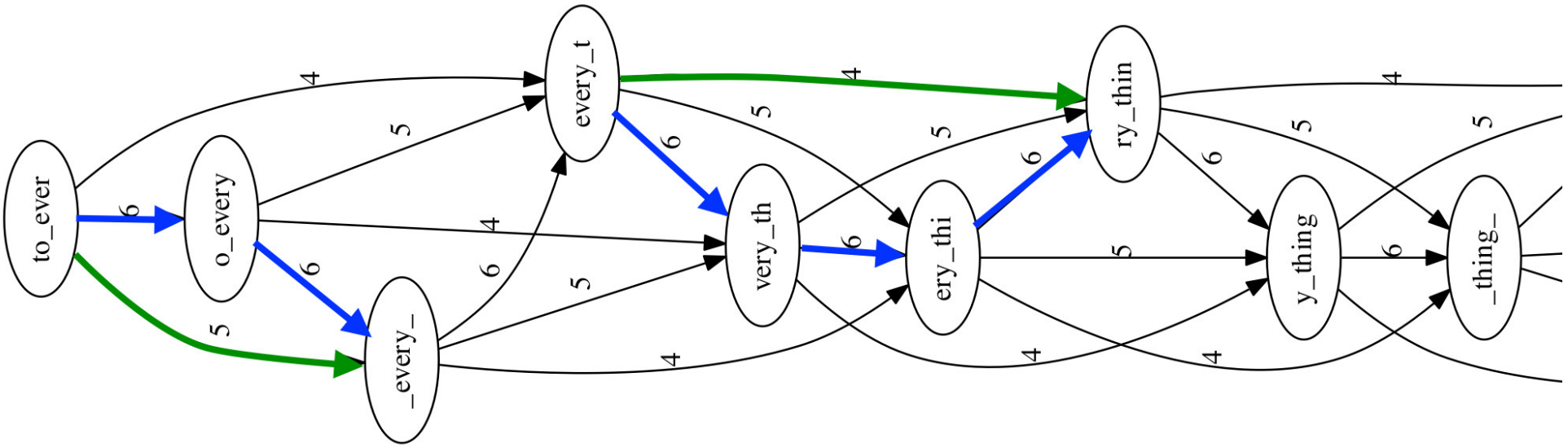
čítania dĺžky 7, minimálny prekryv 4



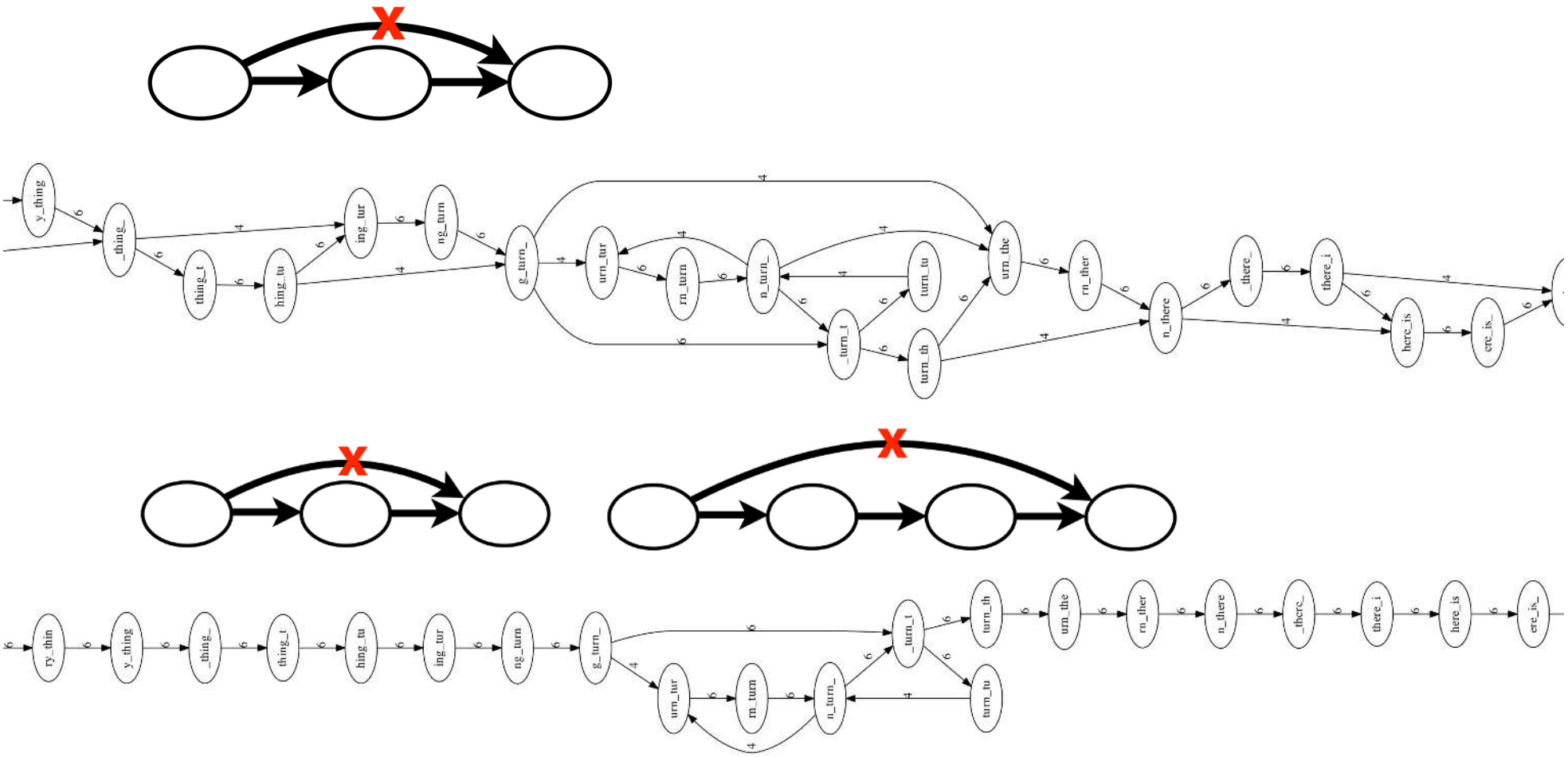
Príklad a obrázky Ben Langmead

Layout: Tranzitívne hrany

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



Layout: Odstránenie tranzitívnych hrán

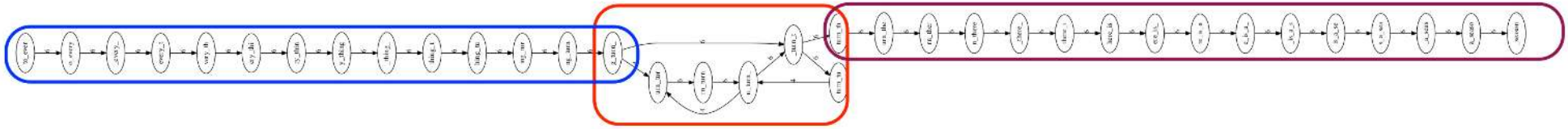


Layout: Rozdelenie na kontigy

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

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	N50
Illumina / Spades	250x	1102	172.6 Kbp	62.0 Kbp
PacBio / Canu	37x	17	4.7 Mbp	1.7 Mbp
PacBio + nanopore	65x	11	4.4 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 Bruijnove grafy sa podobajú, existujú snahy o zjednotenie týchto dvoch konceptov

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,. . .
2007	Watsonov a Venterov genóm (454)
2012	1000 ľudských genómov
2021	3,5 milióna genómov SARS-CoV-2
2021	UK Biobank 200,000 ľudských genómov + veľa ďalších dát
2022	Naozaj dokončený ľudský genóm (telomere to telomere)
2024	All of US 246,000 ľudských genómov + zdravotné záznamy

Použitie NGS: Populačná genetika

- Sekvenujeme väčšinou krátke čítania z genómu určitého človeka
- Ako sa môj vlastný genóm líši od referenčného ľudského genómu?
- Ako jednotlivé genetické rozdiely ovplyvňujú fenotyp?
- Personalizovaná medicína
- Populačná štruktúra, história ľudstva
- Etické otázky

Problémy:

- Mapovanie čítaní na referenčný genóm
- 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 čítaní z rôznych genómov
- Snažíme sa zostaviť aspoň krátke kontigy

Problémy:

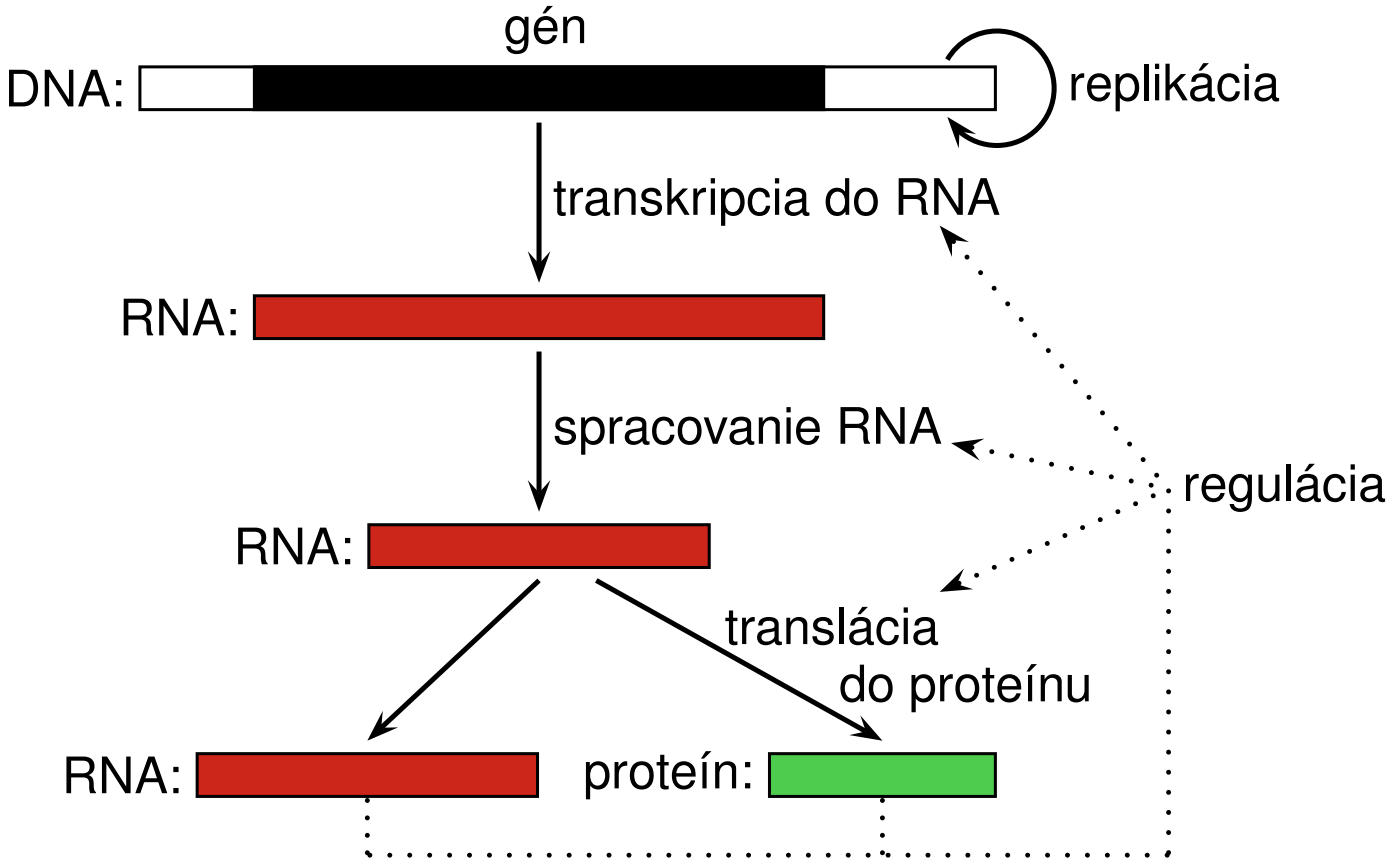
- Oddelenie čítaní/kontigov patriacich do rôznych genómov
- Porovnanie veľkého množstva čítaní s veľkou databázou známych genómov

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

- RNA-seq: Sekvenovať môžeme aj RNA, dostávame gény v genóme
- ChIP-seq: vyfiltrujeme kúsky DNA, na ktoré je naviazaný určitý proteín, sekvenujeme, mapujeme na genóm
- Veľa ďalších technológií mapujúcich pomocou sekvenovania modifikácie DNA, stav chromatínu, 3D rozmiestnenie a pod. (viď predmet Genomika)

Problémy:

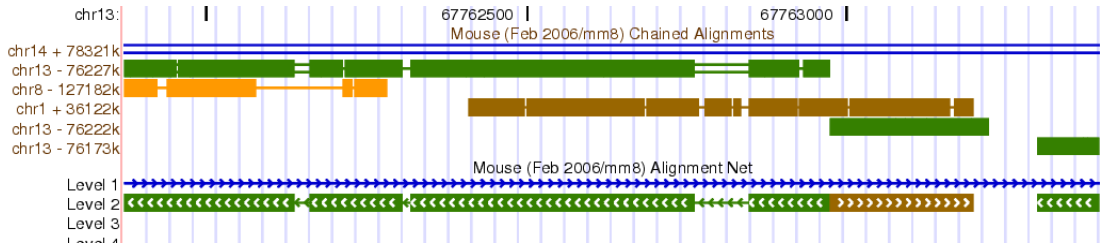
- Opäť mapovanie čítaní na referečný genóm
- Identifikácia miest zostrihu
- Identifikácia väzobných miest podľa hĺbky pokrytia


gén
DNA:
replikácia
transkripcia do RNA
RNA:
spracovanie RNA
regulácia
RNA:
translácia
do proteínu
RNA:
proteín:

Zarovnávanie sekvencií (sequence alignment) 1/2

Broňa Brejová

5.10.2023



Problém: Lokálne zarovnávanie (local alignment)

```
ggcccttggagttgactgtcctgctgctccttgagg
ccattctcagagagaggaagtggcctcattttaatc
cgcttcccacagccttgtcctttccagacccatggg
agagggaggggctgagggtgtggctgagcccaccca
agtcacgcgtcactctgcaggtccctctcccccaag
gccgtggccttgggagcccgtggatcccagtgagtg
acgcctccacccccgccctactcgggcagtttaac
ccttgttgttcacttgcagacatcgtgaacacggcc
cggcccgacgagaaggccataatgacctatgtgtcc
agcttctaccatgccttttcaggagcgcagaaggta
ccgagcagggccaggcaggccctcctcgccgccacc
gcgcaatgccgccgctgcctctcgcctcccgtgctc
acctcatttctcttgcagacggcagtggcctctctc
caactggaagccacccccagctccct...
```

```
tgatgccgaggatgtgttcgtcgagcatccggacga
gaagtccatcacctacgtggtcacctactatcacta
ctttagcaaactcaagcaggagacggtgcagggcat
aagcgtatcggtaaggtggtcggcattgccatggag
aacgacaaaatggtccacgactacgagaacttcaca
agcgatctgctcaagtggatcgaaacgaccatccag
tcgctgggcgagcgggagttcgaaaactcgctggcc
ggcgtccaagggcagttggcccagttctccaactac
cgcaccatcgagaagccgcccaagtttgtggaaaag
ggcaacctcgaggtgctcctttcaccctgcagtcc
aagatgcgggccaacaaccagaagccctacacaccc
aaagagggcaagatgatttcggacatcaacaaggcc
tgggagcgtctggagaaggccgagcacgaacgcgaa
ttggccctgcgcgaggagctcatccg...
```

Vstup: dve sekvencie

Problém: Lokálne zarovnávanie (local alignment)

```
ggcccttggagttgactgtcctgctgctccttgagg   tgatgccgaggatgtgttcgtcgagcatccggacga
ccattctcagagagaggaagtggcctcattttaatc   gaagtccatcacctacgtggtcacctactatcacta
cgcttcccacagccttgtcctttccagacccatggg   ctttagcaaactcaagcaggagacggtgcagggcat
agagggaggggctgagggtgtggctgagcccaccca   aagcgtatcggtaaggtggtcggcattgccatggag
agtcacgcgtcactctgcaggtccctctcccccaag   aacgacaaaatggtccacgactacgagaacttcaca
gccgtggccttgggagcccgtggatcccagtgagtg   agcgatctgctcaagtggatcgaaacgaccatccag
acgcctccacccccgccctactcgggcagtttaac   tcgctgggcgagcgggagttcgaaaactcgctggcc
ccttgttgttcacttgcagacatcgtgaacacggcc   ggcgtccaagggcagttggcccagttctccaactac
cggcccgacgagaaggccataatgacctatgtgtcc   cgcaccatcgagaagccgcccaagtttgtggaaaag
agcttctaccatgcctttcaggagcgcagaaggta   ggcaacctcgaggtgctcctttcaccctgcagtcc
ccgagcagggccaggcaggccctcctcgccgccacc   aagatgcgggccaacaaccagaagccctacacaccc
gcgcaatgccgccgctgcctctcgcctcccgtgctc   aaagagggcaagatgatttcggacatcaacaaggcc
acctcatttctcttgcagacggcagtggcctctctc   tgggagcgtctggagaaggccgagcacgaacgcgaa
caactggaagccacccccagctccct...          ttggccctgcgcgaggagctcatccg...
```

Výstup: podobné úseky (zarovnania, 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 zarovnania?

- **Orientácia v obrovských databázach.**

  Genbank WGS má vyše 22 TB sekvencií.

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

- **Prekryvy čítaní** pri skladaní genómov, mapovanie čítaní

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

  Podobné sekvencie často majú rovnakú/podobnú funkciu.

- **Štúdium evolúcie.**

  Hľadáme homológy: sekvencie, ktoré sa vyvinuli z tej istej sekvencie v spoločnom predkovi.

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

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

Zarovnávanie sekvencií ako optimalizačný problém

Cieľ: nájdi páry homologických sekvencií

(tých, čo pochádzajú z rovnakého spoločného predka)

Modelovacia fáza: vytvor skórovaciu schému, ktorá

– skutočným homologickým párom dáva vysoké skóre

– falošne pozitívnym párom dáva nízke skóre

Optimalizačná fáza:

pre dané dve vstupné sekvencie, nájde zarovnanie s najlepším skóre

dôležitá je výpočtová a pamäťová zložitosť algoritmu

Formulácia problému

Skórovanie zarovnania: napr. zhoda +1, nezhoda -1, medzera -1.

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

22 zhôd, 6 nezhôd, 3 medzery $\rightarrow$ skóre 13.

V praxi zložitejšie skórovanie.

Problém 1: globálne zarovnanie (global alignment)

Vstup: sekvencie $X = x_1 x_2 \ldots x_n$ a $Y = y_1 y_2 \ldots y_m$.

Výstup: zarovnanie X a Y s najvyšším skóre.

Problém 2: lokálne zarovnanie (local alignment)

Vstup: sekvencie $X = x_1 x_2 \ldots x_n$ a $Y = y_1 y_2 \ldots y_m$.

Výstup: zarovnania podreťazcov $x_i \ldots x_j$ a $y_k \ldots y_\ell$ s najvyšším skóre.

Dynamické programovanie pre globálne zarovnanie

(Needleman, Wunsch 1970)

Podproblém: $A[i, j]$: najvyššie skóre globálneho zarovnania reťazcov $x_1 x_2 \ldots x_i$ a $y_1 y_2 \ldots y_j$.

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

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

Všeobecný prípad, $i > 0$, $j > 0$:

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

ak $x_i \neq y_j$ 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 zarovnania reťazcov $x_1 x_2 \ldots x_i$ a $y_1 y_2 \ldots y_j$.

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

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

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

Rekurencia:

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

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

Príklad globálneho zarovnania

CATGTCGTA 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										

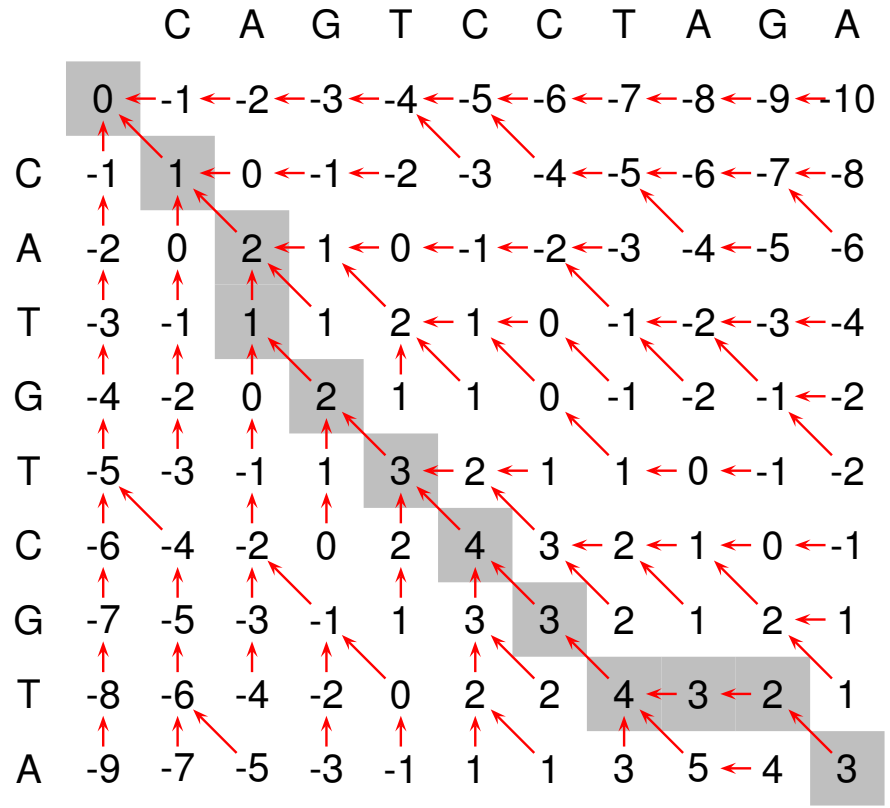
$$A[i,j] = \max \begin{cases} A[i-1,j-1] + s(x_i, y_j), \\ A[i-1,j] - 1, \\ A[i,j-1] - 1 \end{cases}$$

Príklad globálneho zarovnania

CATGTCGTA 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?




```
CATGTCGT--A
CA-GTCCTAGA
```

Časová zložitosť celého algoritmu $O(nm)$

Dynamické programovanie pre lokálne zarovnanie

(Smith, Waterman 1981)

Podproblém: $A[i,j]$: najvyššie skóre lokálneho zarovnania reťazcov $x_1x_2\ldots x_i$ a $y_1y_2\ldots y_j$, 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 $A[i,j] = A[i-1,j] - 1$

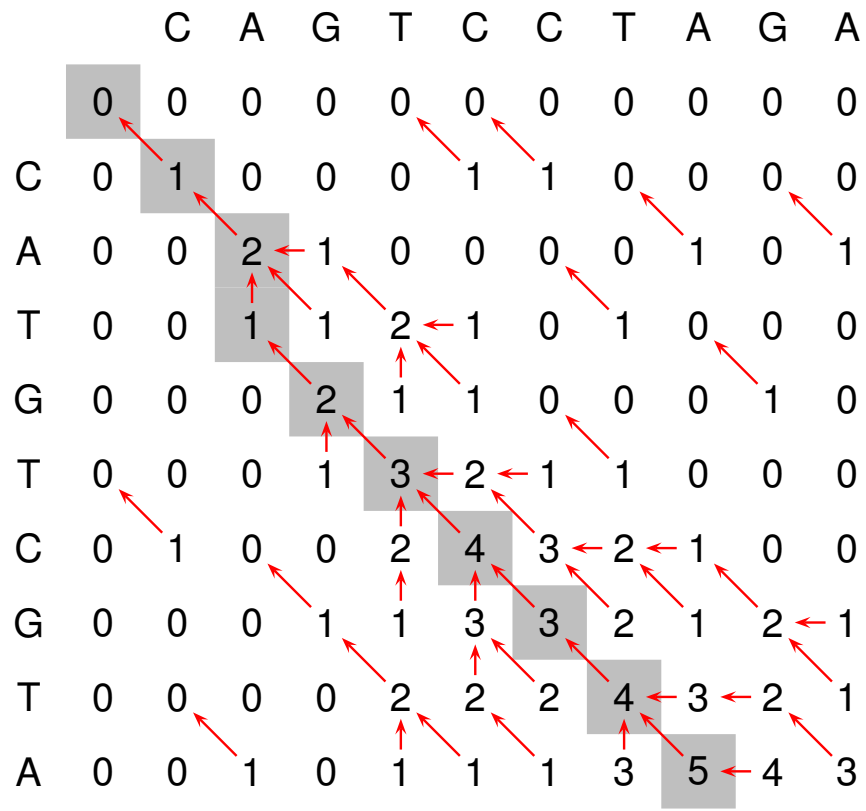
ak y_j je zarovnané s medzerou $A[i,j] = A[i,j-1] - 1$

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

Rekurencia:

$$A[i,j] = \max \begin{cases} 0, \\ A[i-1,j-1] + s(x_i, y_j), \\ A[i-1,j] - 1, \\ A[i,j-1] - 1 \end{cases}$$

Príklad lokálneho zarovnania




```
CATGTCGTA
CA-GTCCTA
```

Časová zložitosť celého algoritmu $O(nm)$

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

Skórovanie založené na dvoch pravdepodobnostných modeloch

Majme zarovnané sekvencie X a Y

Aké má zarovnanie dostať skóre?

Pre jednoduchosť teraz neuvažujme medzery v zarovnaní.

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

Model generuje nezávisle **jednotlivé stĺpce** zarovnania.

Dané pravdepodobnosti $p(x, y)$, že v stpĺci budú pod sebou x a y.

Pravdepodobnosť vygenerovania X zarovnaného s Y:

$$\Pr(X, Y \mid H) = \prod_{i=1}^{n} p(x_i, y_i) = p(x_1, y_1) \cdot p(x_2, y_2) \cdot \ldots \cdot p(x_n, y_n)$$

Príklad: Uvažujme iba bázy A, T

Model H: $p(A, A) = 0.4$, $p(A, T) = 0.15$, $p(T, A) = 0.15$, $p(T, T) = 0.3$

$$\Pr(\mathrm{AAAT}, \mathrm{ATAT} \mid H) = 0.4 \cdot 0.15 \cdot 0.4 \cdot 0.3 = 0.0072$$

Skórovanie založené na dvoch pravdepodobnostných modeloch

Majme zarovnané sekvencie X a Y

Aké má zarovnanie dostať skóre?

Pre jednoduchosť teraz neuvažujme medzery v zarovnaní.

Model R: Sekvencie X a Y **nijako spolu nesúvisia** (random)

Model generuje nezávisle **jednotlivé bázy** v obidvoch sekvenciach.

Dané pravdepodobnosti $p(x)$, že vygeneruje bázu x.

Pravdepodobnosť vygenerovania X zarovnaného s Y:

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

Príklad: Uvažujme iba bázy A, T

Model R: $p(A) = 0.55, p(T) = 0.45$

$$\Pr(\mathrm{AAAT}, \mathrm{ATAT} \mid R) = 0.55^5 \cdot 0.45^3 \approx 0.00459$$

Skórovanie založené na dvoch pravdepodobnostných modeloch

Porovnávame dva modely H a R: skóre bude logaritmus podielu

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

- Ak sú sekvencie X a Y **homológy**

  $\Rightarrow$ logaritmus podielu oveľa väčšieho ako 1 $\Rightarrow$ **veľmi pozitívne skóre**

- Ak sekvencie X a Y **nesúvisia**

  $\Rightarrow$ logaritmus podielu oveľa menšieho ako 1 $\Rightarrow$ **veľmi negatívne skóre**

Príklad:

$$\Pr(\mathrm{AAAT}, \mathrm{ATAT} \mid H) = 0.0072$$
$$\Pr(\mathrm{AAAT}, \mathrm{ATAT} \mid R) \approx 0.00459$$

$$\log_2 \frac{\Pr(X, Y \mid H)}{\Pr(X, Y \mid R)} \approx \log_2 \frac{0.0072}{0.00459} \approx \log_2 1.5699 \approx 0.6507$$

Základ logaritmu môžeme zvoliť hocijaký, tu 2.

Skórovanie založené na dvoch pravdepodobnostných modeloch

Porovnávame dva modely H a R: skóre bude logaritmus podielu

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

- Ak sú sekvencie X a Y **homológy**

  $\Rightarrow$ logaritmus podielu oveľa väčšieho ako 1 $\Rightarrow$ **veľmi pozitívne skóre**

- Ak sekvencie X a Y **nesúvisia**

  $\Rightarrow$ logaritmus podielu oveľa menšieho ako 1 $\Rightarrow$ **veľmi negatívne skóre**

Príklad:

$$\Pr(\mathrm{AAAT}, \mathrm{TTAA} \mid H) = 0.00135$$
$$\Pr(\mathrm{AAAT}, \mathrm{TTAA} \mid R) \approx 0.00459$$

$$\log_2 \frac{\Pr(X, Y \mid H)}{\Pr(X, Y \mid R)} \approx \log_2 \frac{0.000135}{0.00459} \approx \log_2 0.2944 \approx -1.7643$$

Ako použiť takéto skóre v N.-W. a S.-W. algoritmoch?

Potrebujeme ho rozbiť na **súčet** skór pre zarovnané dvojice písmen.

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

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

$$= \log \prod_{i=1}^{n} \frac{p(x_i, y_i)}{p(x_i) \cdot p(y_i)} = \sum_{i=1}^{n} \log \frac{p(x_i, y_i)}{p(x_i) \cdot p(y_i)}$$

Skóre zarovnanie bázy x a bázy y:

$$s(x, y) = \log \frac{p(x, y)}{p(x) \cdot (y)}$$

Ako použiť takéto skóre v N.-W. a S.-W. algoritmoch?

Skóre zarovnanie bázy x a bázy y:

$s(x, y) = \log \frac{p(x,y)}{p(x)p(y)}$

Príklad:

Model H: $p(A, A) = 0.4, p(A, T) = 0.15, p(T, A) = 0.15, p(T, T) = 0.3$

Model R: $p(A) = 0.55, p(T) = 0.45$

$$s(A, A) = \log_2 \frac{0.4}{0.55 \cdot 0.55} \approx 0.4031$$
$$s(A, T) = \log_2 \frac{0.15}{0.55 \cdot 0.45} \approx -0.7225$$
$$s(T, A) = \log_2 \frac{0.15}{0.45 \cdot 0.55} \approx -0.7225$$
$$s(T, T) = \log_2 \frac{0.3}{0.45 \cdot 0.45} \approx 0.5670$$

Pre AAAT, ATAT skóre $0.4031 - 0.7225 + 0.4031 + 0.5670 = 0.6507$

Pre AAAT, TTAA skóre $-0.7225 - 0.7225 + 0.4031 - 0.7225 = -1.7644$

BLOSUM62 skórovacia matica pre proteíny

BLOcks of aminoacid **SU**bstitution **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 zarovnania** proteínov (BLOCKS)
- Páry s nanajvýš 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

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

Zhrnutie

- Globálne a lokálne zarovania
- Needlemanov-Wunschov a Smithov-Watermanov algoritmus
- Skórovanie zarovnaní pomocou porovnávania 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

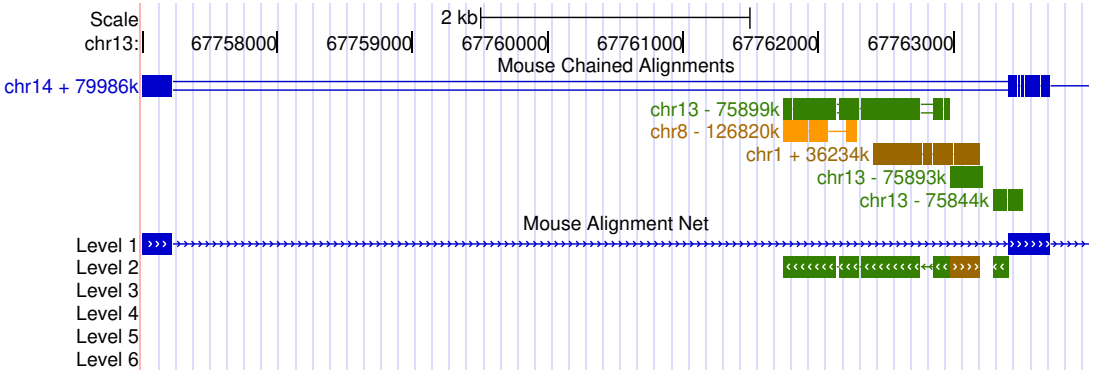
- Výber článku na journal club formulárom na stránke do budúcej stredy 23.10. 22:00.
- Domáca úloha 1 bude zverejnená budúci týždeň
- Nezabudnite na pravidelné kvízy, ak je niečo nejasné, pýtajte sa.

Zarovnávanie sekvencií 2/2

(sequence alignment)

Tomáš Vinař

17.10.2024



Zhrnutie z minulej prednášky

- **Problém globálneho a lokálneho zarovnania**

  Vstup: sekvencie $X = x_1 x_2 \ldots x_n$ a $Y = y_1 y_2 \ldots y_m$.

  Výstup:

  globálne: zarovnanie X a Y s najvyšším skóre

  lokálne: zarovnania podreťazcov $x_i \ldots x_j$ a $y_k \ldots y_\ell$ s najvyšším skóre.

- **Správny algoritmus na riešenie**

  dynamické programovanie

- **Realistické skórovacie schémy**

Máme správny algoritmus na zarovnávanie, čo viac nám chýba?

Časová zložitosť: $O(n^2)$ na dvoch sekvenciách dĺžky n.

Koľko je to času v skutočnosti?

(jednoduchá implementácia, náhodné sekvencie dĺžky n,

bežný počítač)

n	čas výpočtu
1,000	0.02s
10,000	1.5s
100,000	2.5 minúty (*)
1,000,000	4 hodiny (*)
10,000,000	17 dní (*)
100,000,000	5 rokov (*)
1,000,000,000	476 rokov (*)

Potrebujeme efektívnejší algoritmus,

najmä ak chceme pracovať s celými genómami

Pamäť: základný algoritmus $O(n^2)$, dá sa zlepšiť na $O(n)$.

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 a Lipman 1988)

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

Ako nájdeme jadrá, 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
```

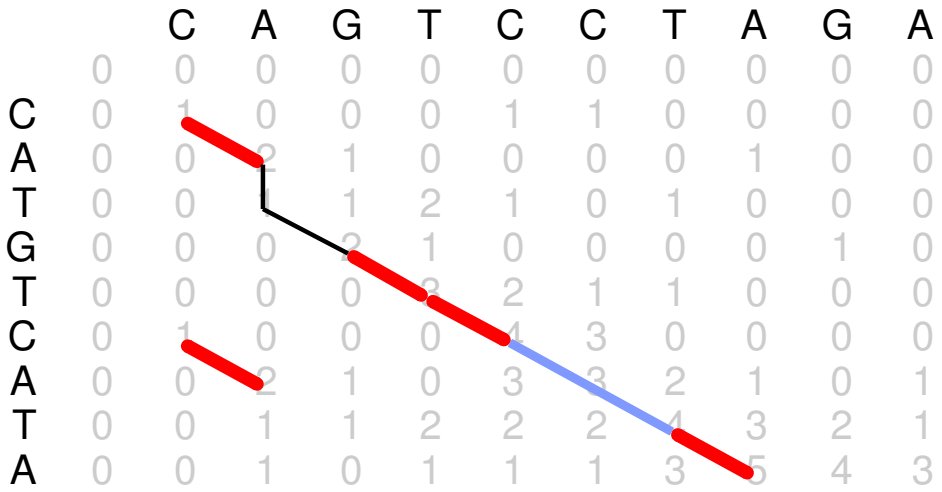
Hľadaj:

```
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 $w = 11$ a viac.)



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 časťou zarovnania s vysokým skóre.

Vyfiltrujeme ich pri rozširovaní, ale spomaľujú program.

Koľko náhodných jadier?

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

Jadro, t.j. w zhôd za sebou s pravdepodobnosťou 4^{-w}.

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

Zvýšenie w o 1 zníži jadier 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 zarovnania: vysoké skóre, ale nemajú jadro dĺžky w

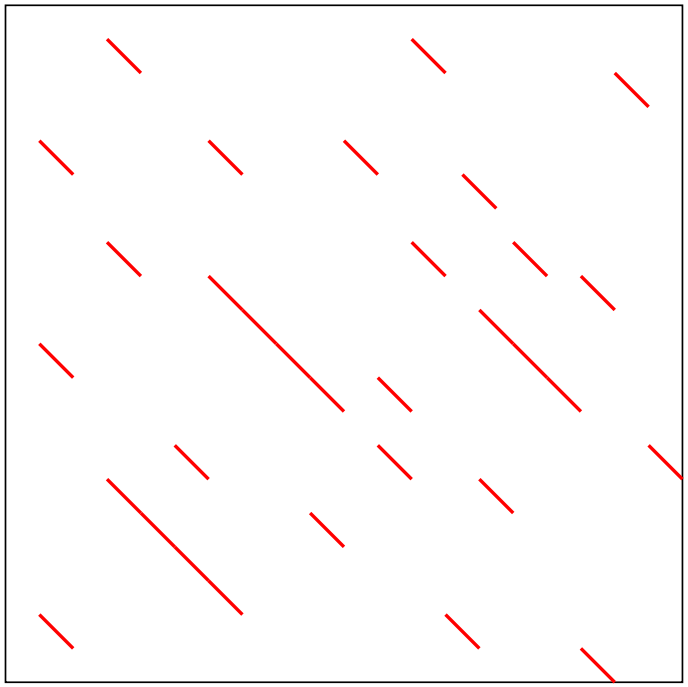
Príklad:
```
CA-GTCCTA
||  |||  ||
CATGTCATA
```
nenájdeme pre $w \geq 4$

Senzitivita: aká časť skutočných zarovnaní obsahuje jadro, t.j. zhodu dĺžky w

Rýchlosť vs. senzitivita

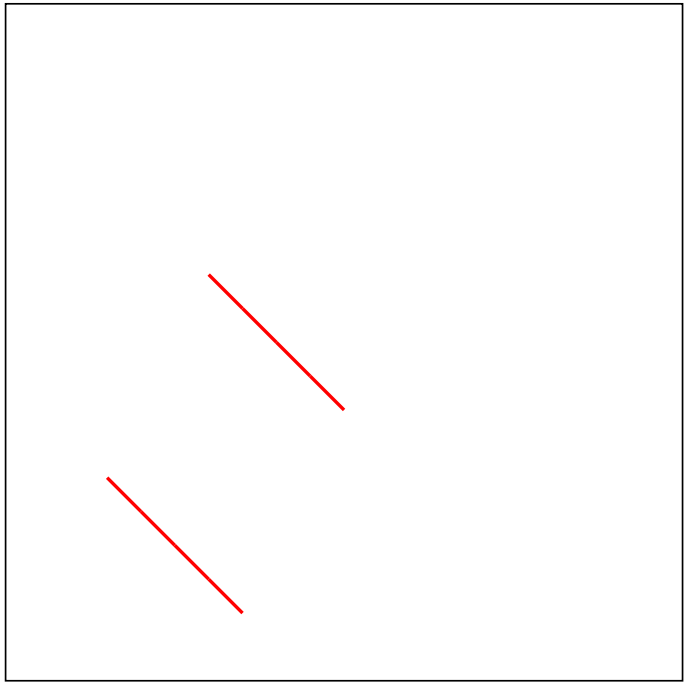
Malé w

veľa náhodných zhôd, pomalé



Veľké w

nenájdeme veľa zarovnaní



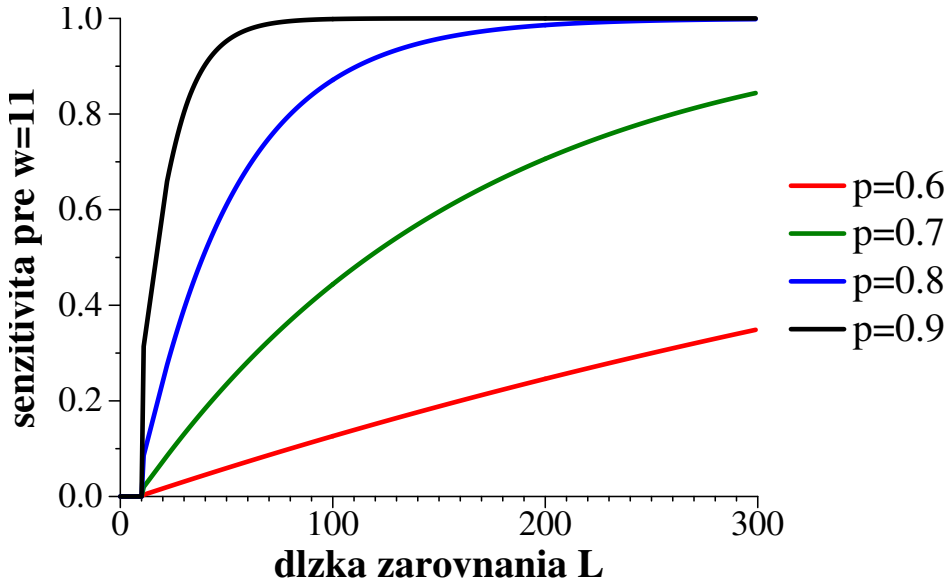
Senzitivita heuristického algoritmu

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 na rôzne účely

NCBI BLAST: blastn pre DNA/RNA, blastp pre proteíny,
tblastx preloží DNA do proteínu a použije blastp

UCSC Blat: pomerne rýchle vyhľadávanie veľmi podobných sekvencií, napr. kde je daná sekvencia v genóme

- používa veľké w
- vie nájsť zarovnania s veľkými medzerami (napr. intróny pri mRNA)

Minimap2: mapuje dlhé čítania na genóm alebo porovnáva dva príbuzné genómy

- používa techniku minimizerov na ušetrenie pamäti (neukladá všetky úseky dĺžky w)
- veľmi rýchly

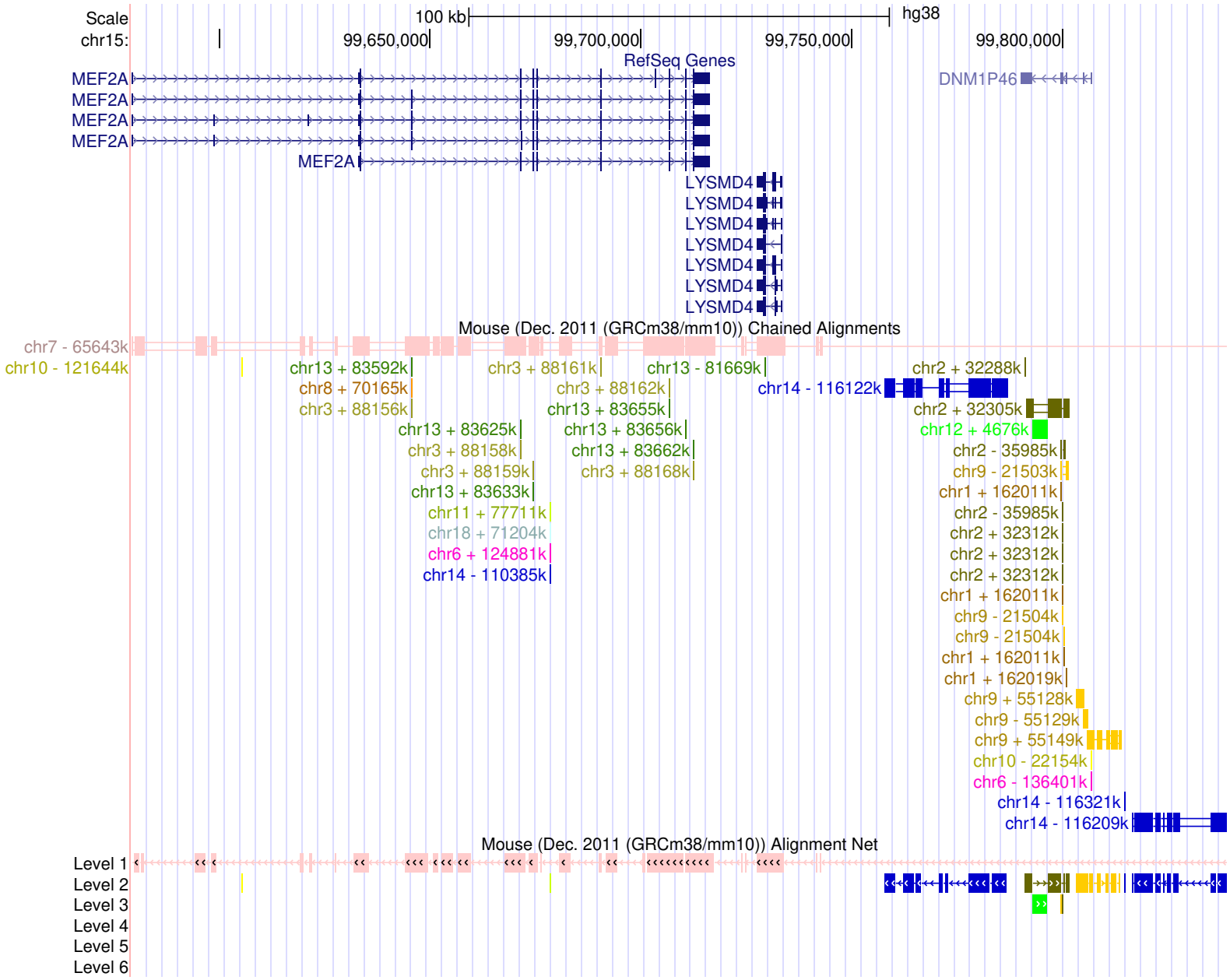
BWA-MEM/BWA-MEM2: mapuje krátke čítania na genóm

- namiesto jadier fixnej dĺžky používa maximálne presné zhody, zložitejšie dátové štruktúry

Genomické zarovnania (whole-genome alignments)

Ku každému úseku ľudského genómu nájsť zodpovedajúcu časť z myši, psa, sliepky, atď. (predpočítané v UCSC browseri)

- Lokálne zarovnania 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 zarovnania, 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 (ortológy)


Scale
chr15:
100 kb
hg38
99,650,000
99,700,000
99,750,000
99,800,000
RefSeq Genes
MEF2A
MEF2A
MEF2A
MEF2A
MEF2A
DNM1P46
LYSMD4
LYSMD4
LYSMD4
LYSMD4
LYSMD4
LYSMD4
LYSMD4
Mouse (Dec. 2011 (GRCm38/mm10)) Chained Alignments
chr7 - 65643k
chr10 - 121644k
chr13 + 83592k
chr8 + 70165k
chr3 + 88156k
chr3 + 88161k
chr13 - 81669k
chr3 + 88162k
chr13 + 83655k
chr13 + 83625k
chr3 + 88158k
chr3 + 88159k
chr13 + 83633k
chr11 + 77711k
chr18 + 71204k
chr6 + 124881k
chr14 - 110385k
chr13 + 83656k
chr13 + 83662k
chr3 + 88168k
chr14 - 116122k
chr2 + 32288k
chr2 + 32305k
chr12 + 4676k
chr2 - 35985k
chr9 - 21503k
chr1 + 162011k
chr2 - 35985k
chr2 + 32312k
chr2 + 32312k
chr2 + 32312k
chr1 + 162011k
chr9 - 21504k
chr9 - 21504k
chr1 + 162011k
chr1 + 162019k
chr9 + 55128k
chr9 - 55129k
chr9 + 55149k
chr10 - 22154k
chr6 - 136401k
chr14 - 116321k
chr14 - 116209k
Mouse (Dec. 2011 (GRCm38/mm10)) Alignment Net
Level 1
Level 2
Level 3
Level 4
Level 5
Level 6

Viacnásobné zarovnanie, multiple sequence alignment

Cieľ: Zarovnaj viacero sekvencií.

```
      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-----------------
```

Ako by ste riešili skórovanie, aký by ste použili algoritmus?

Viacnásobné zarovnanie, multiple sequence alignment

```
      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-----------------
```

Skórovanie: napr. súčet párových skór všetkých dvojíc sekvencií.

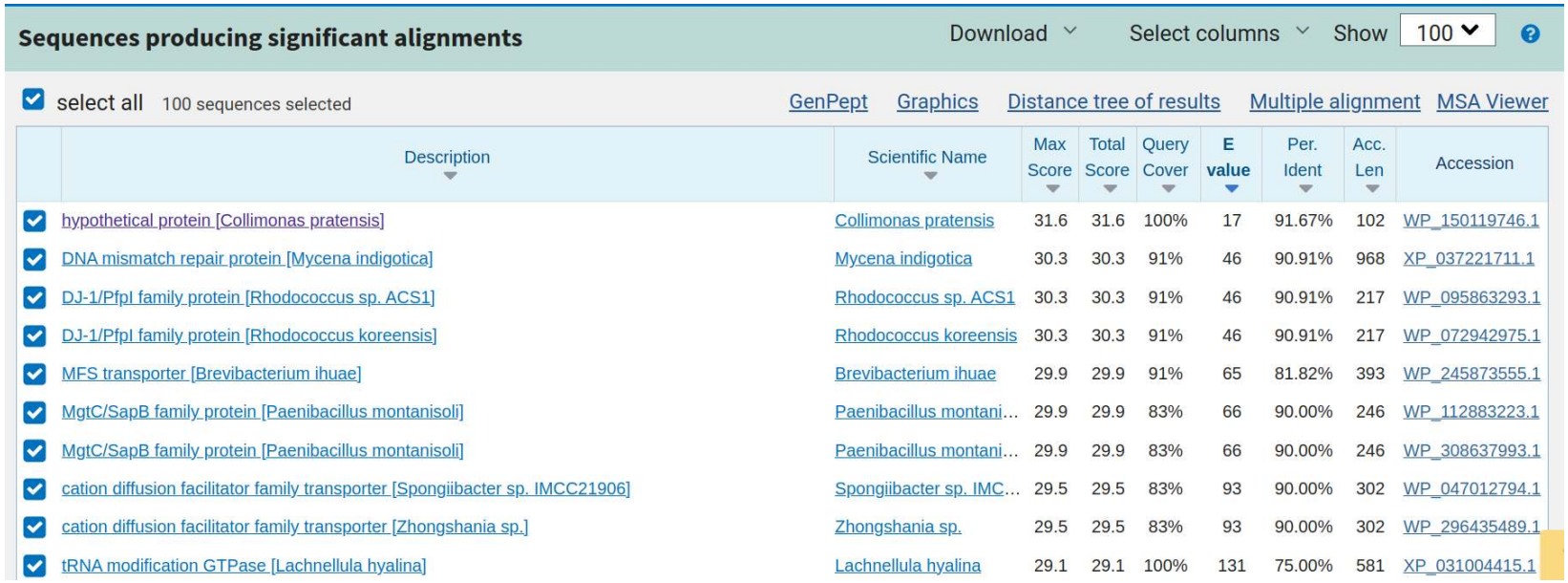
V každej dvojici vyhodím stĺpce s dvomi pomlčkami.

Zložitosť dynamického programovania: $O(2^k n^k)$ pre k sekvencií dĺžky n.

Pre všeobecné k NP-ťažké.

Heuristické algoritmy, napr. CLUSTAL-W, MUSCLE, TBA, MAFFT.

Často zarovnávajú hierarchicky vždy dve skupiny do jednej väčšej.



Sequences producing significant alignments Download Select columns Show 100

select all 100 sequences selected GenPept Graphics Distance tree of results Multiple alignment MSA Viewer

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
hypothetical protein [Collimonas pratensis]	Collimonas pratensis	31.6	31.6	100%	17	91.67%	102	WP_150119746.1
DNA mismatch repair protein [Mycena indigotica]	Mycena indigotica	30.3	30.3	91%	46	90.91%	968	XP_037221711.1
DJ-1/PfpI family protein [Rhodococcus sp. ACS1]	Rhodococcus sp. ACS1	30.3	30.3	91%	46	90.91%	217	WP_095863293.1
DJ-1/PfpI family protein [Rhodococcus koreensis]	Rhodococcus koreensis	30.3	30.3	91%	46	90.91%	217	WP_072942975.1
MFS transporter [Brevibacterium ihuae]	Brevibacterium ihuae	29.9	29.9	91%	65	81.82%	393	WP_245873555.1
MgtC/SapB family protein [Paenibacillus montanisoli]	Paenibacillus montani...	29.9	29.9	83%	66	90.00%	246	WP_112883223.1
MgtC/SapB family protein [Paenibacillus montanisoli]	Paenibacillus montani...	29.9	29.9	83%	66	90.00%	246	WP_308637993.1
cation diffusion facilitator family transporter [Spongiibacter sp. IMCC21906]	Spongiibacter sp. IMC...	29.5	29.5	83%	93	90.00%	302	WP_047012794.1
cation diffusion facilitator family transporter [Zhongshania sp.]	Zhongshania sp.	29.5	29.5	83%	93	90.00%	302	WP_296435489.1
tRNA modification GTPase [Lachnellula hyalina]	Lachnellula hyalina	29.1	29.1	100%	131	75.00%	581	XP_031004415.1

Výsledok programu BLAST voči RefSeq proteínovej databáze na serveroch NCBI https://blast.ncbi.nlm.nih.gov/

Download GenPept Graphics

hypothetical protein [Collimonas pratensis]

Sequence ID: WP_150119746.1 Length: 102 Number of Matches: 1

Range 1: 20 to 31 GenPept Graphics ▼ Next Match ▲ Pr

Score	Expect	Identities	Positives	Gaps
31.6 bits(67)	17	11/12(92%)	11/12(91%)	0/12(0%)

```
Query  1   VIVALASVEGAS  12
           VIVALASV GAS
Sbjct  20  VIVALASVIGAS  31
```

Download GenPept Graphics

DNA mismatch repair protein [Mycena indigotica]

Sequence ID: XP_037221711.1 Length: 968 Number of Matches: 1

Range 1: 482 to 492 GenPept Graphics ▼ Next Match ▲ Pr

Score	Expect	Identities	Positives	Gaps
30.3 bits(64)	46	10/11(91%)	10/11(90%)	0/11(0%)

```
Query  2    IVALASVEGAS  12
            IVALASVE AS
Sbjct  482  IVALASVEDAS  492
```

Ako rozlíšiť, či ide o významné zarovnanie?

Dĺžka dotazu m. Veľkosť databázy n.

Zarovnanie so skóre S.

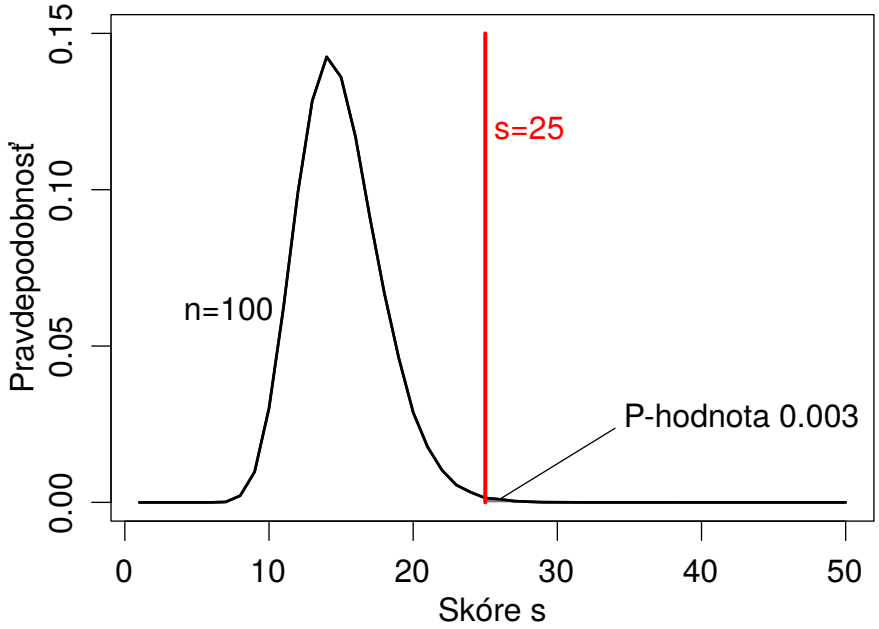
P-hodnota: Pravdepodobnosť, že pre náhodný dotaz dĺžky m v náhodnej databáze dĺžky n nájdeme zarovnanie so skóre aspoň S.

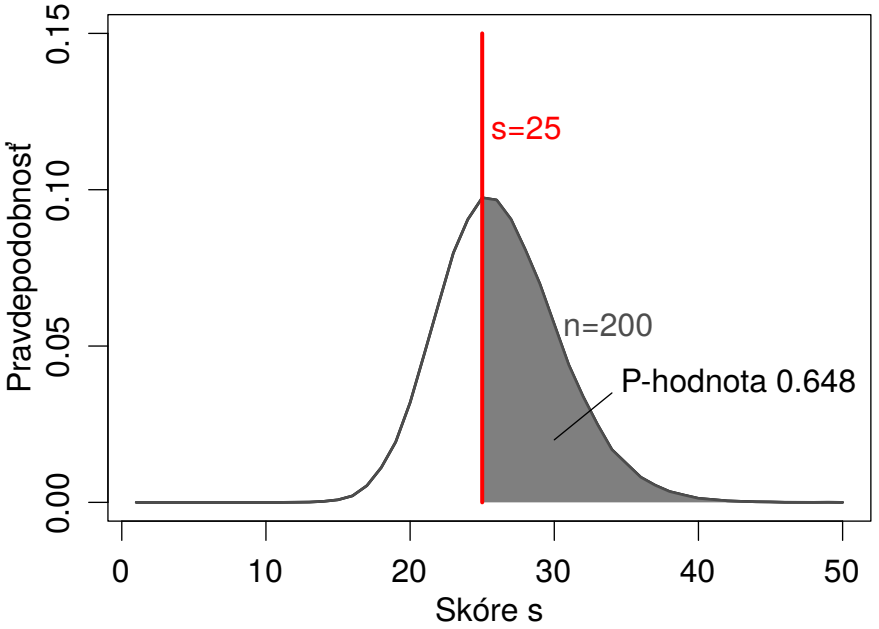
E-hodnota: Očakávaný počet zarovnaní so skóre aspoň S nájdených pre náhodný dotaz dĺžky m v náhodnej databáze dĺžky n.

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

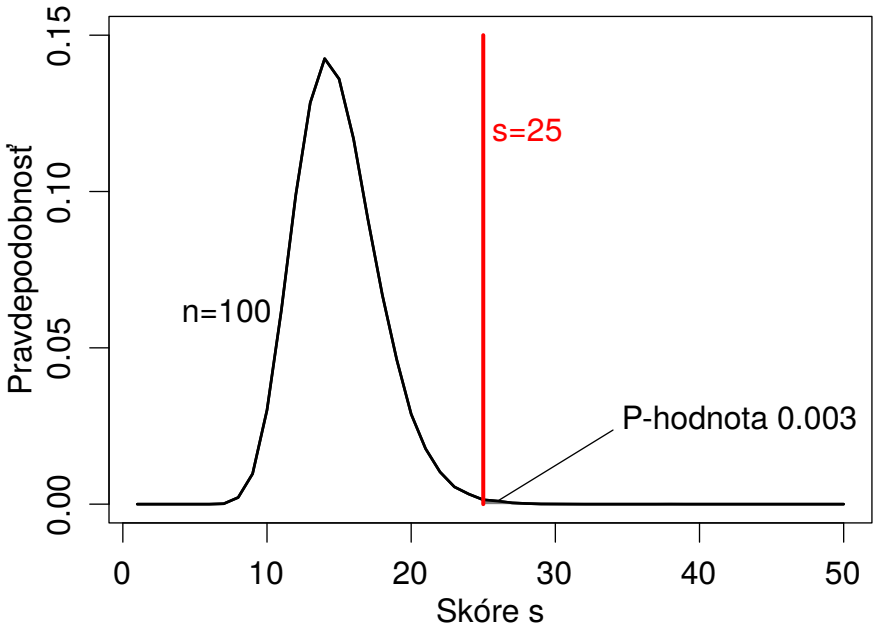
Výpočet P-hodnoty simuláciou

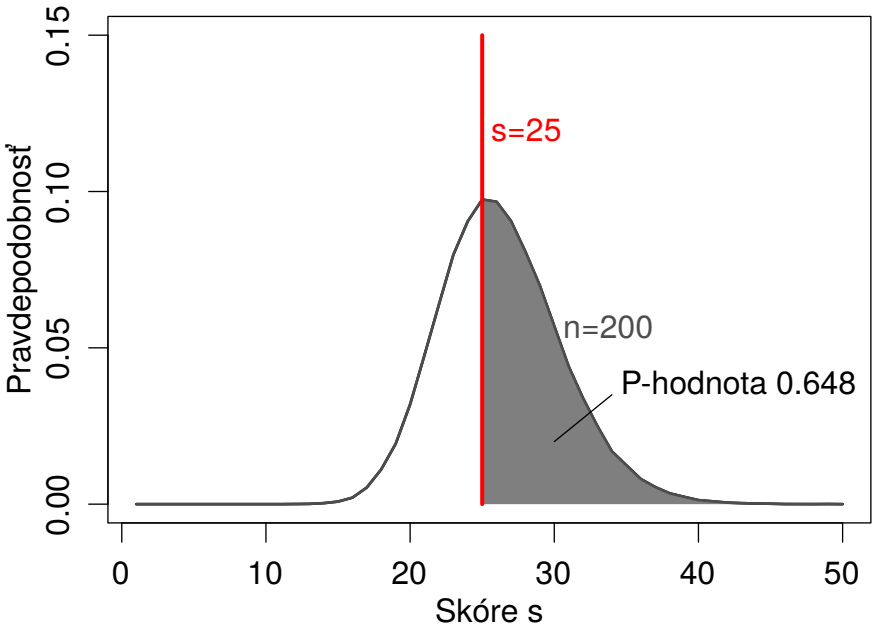
- Vygenerujeme náhodne dve sekvencie dĺžky n
- Spočítame ich najlepšie lokálne zarovnanie (schéma +1/-1)
- Zaznamenáme si výsledné skóre
- Opakujeme veľa krát





Výpočet P-hodnoty simuláciou (pokr.)





P-hodnota pre skóre 25:

Aká časť zarovnaní má skóre 25 alebo vyššie?

V praxi je simulácia pomalá, existujú matematické odhady rozdelenia.

Karlin and Althschul 1990, Dembo et al. 1994

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
- Odhad štatistickej významnosti (E-hodnota, P-hodnota) je dôležitý nástroj na rozpoznávanie reálnych zarovnaní od tých, čo sa vyskytli náhodou
- Špecializované programy na rôzne úlohy súvisiace so zarovnávaním
  – Informatici na ďalších cvičeniach ďalšie finty na zlepšenie jadier
  – Biológovia ukážky použitia programov

Organizačné poznámky

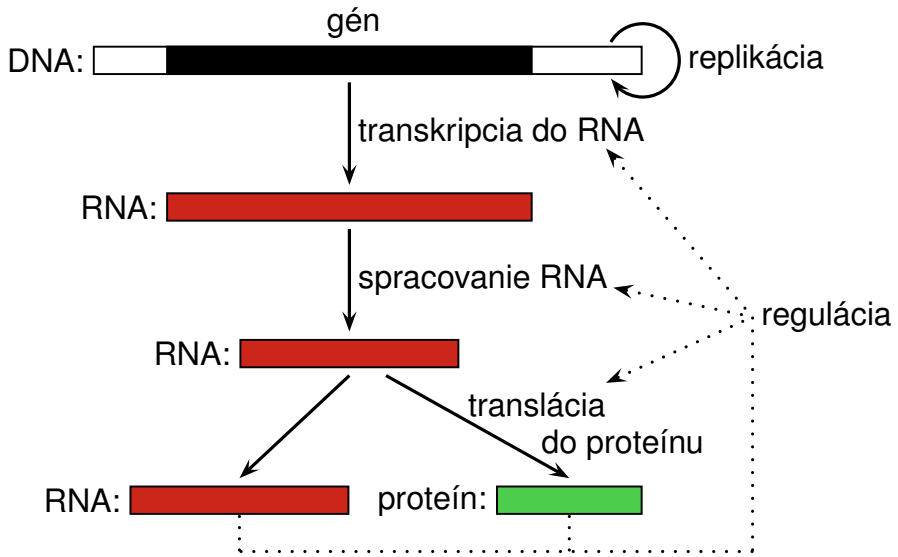
- DÚ1 bude zverejnená dnes, odovzdávanie do stredy 13.11. 22:00
- Journal club: Rozdelenie do skupín / inštrukcie na konci prednášky
- Najbližší journal club termín 22.11.

Hľadanie génov

Tomáš Vinař

24.10.2024

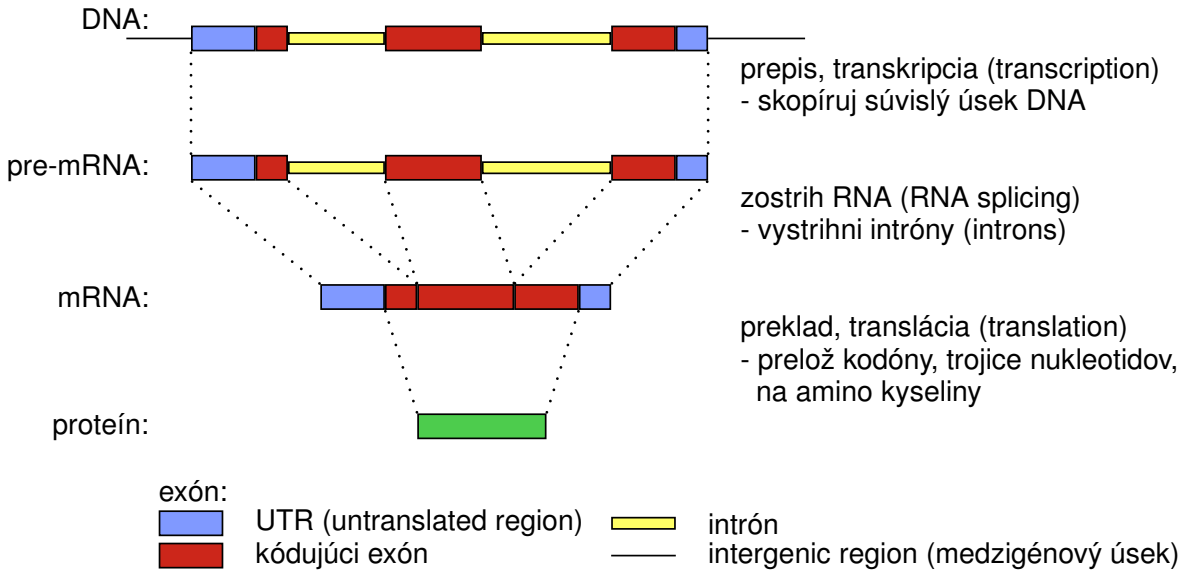
Čo s osekvenovanými genómami?



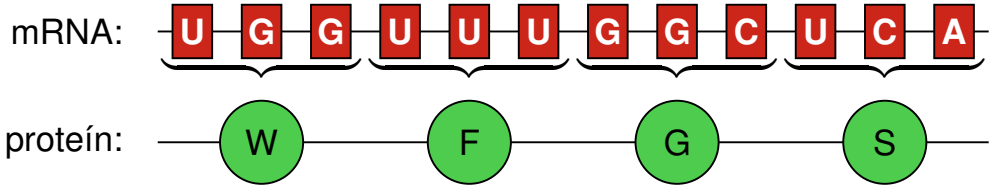
- gény kódujúce proteíny (dnešná prednáška)
- RNA gény
- signály pre reguláciu transkripcie, zostrihu, atď
- pseudogény (nefunkčné kópie génov)
- repetitívne sekvencie, opakovania (sequence repeats)

Štruktúra eukaryotických génov

Proces syntézy proteínov:



Translácia: tri bázy mRNA (kodón) $\rightarrow$ aminokyselina proteínu



Ľudský genóm

Gény kódujúce proteíny

– cca 20 000, pokrývajú 40% genómu

– cca 10 exónov v géne

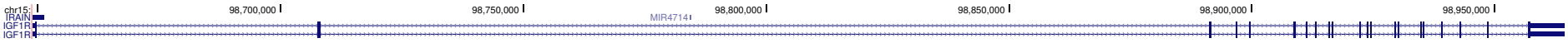
– exóny pokrývajú 2% genómu

– kódujúce exóny 1.2% genómu

Repetitívne sekvencie

– pokrývajú 49% genómu

Príklad: gén IGF1R zaberá 315 569nt, z toho kóduje 4101nt v 21 exónoch



Bioinformatický problém: hľadanie génov

Cieľ: nájsť všetky gény kódujúce proteíny v genóme.

Tým získame katalóg všetkých proteínov.

Zjednodušenia:

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

Bioinformatický problém: hľadanie génov

Vstup: sekvencia DNA

```
cggtgaaactgcacgattgttgctggcttaaagatagaccaatcagagtgtgtaacgtca
tatttagcgtcttctatcatccaatcactgcactttacacactataaatagagcagctca
tgggcgtatttgcgctagtgttgggtgttccgctgtgctgttttccgtcatggctcgca
ctaagcaaactgctcggaagtctactggtggcaaggcgccacgcaaacagttggccacta
aggcagcccgcaaaagcgctccggccaccggcggcgtgaaaaagccccaccgctaccggc
cgggcaccgtggctctgcgcgagatccgccgttatcagaagtccactgaactgcttattc
gtaaactacctttccagcgcctggtgcgcgagattgcgcaggactttaaaacagacctgc
gtttccagagctccgctgtgatggctctgcaggaggcgtgcgaggcctacttggtagggc
tatttgaggacactaacctgtgcgccatccacgccaagcgcgtcactatcatgcccaagg
acatccagctcgcccgccgcatccgcggagagagggcgtgattactgtggtctctctgac
```

Bioinformatický problém: Hľadanie génov

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

cggtgaaactgcacgattgttgctggcttaaagatagaccaatcagagtgtgtaacgtca

tatttagcgtcttctatcatccaatcactgcactttacacactataaatagagcagctca

tgggcgtatttgcgctagtgttgggtgttccgctgtgctgtttttccgtcatggctcgca

ctaagcaaactgctcggaagtctactggtggcaaggcgccacgcaaacagttggccacta

aggcagcccgcaaaagcgctccggccaccggcggcgtgaaaaagccccaccgctaccggc

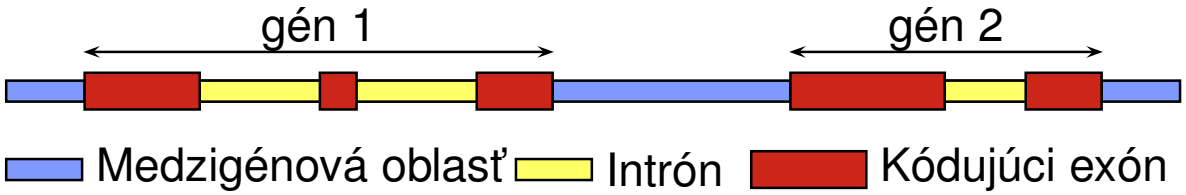
cgggcaccgtggctctgcgcgagatccgccgttatcagaagtccactgaactgcttattc

gtaaactacctttccagcgcctggtgcgcgagattgcgcaggactttaaaacagacctgc

gtttccagagctccgctgtgatggctctgcaggaggcgtgcgaggcctacttggtagggc

tatttgaggacactaacctgtgcgccatccacgccaagcgcgtcactatcatgcccaagg

acatccagctcgcccgccgcatccgcggagagagggcgtgattactgtggtctctctgac



Bioinformatický problém: hľadanie génov

Vstup: sekvencia DNA

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

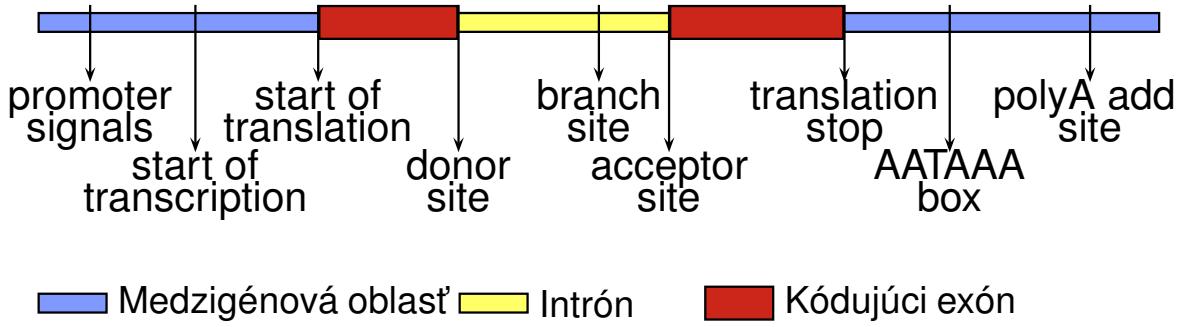
- Toto nie je dobre definovaný problém!

  Ako spoznáme, čo je gén?

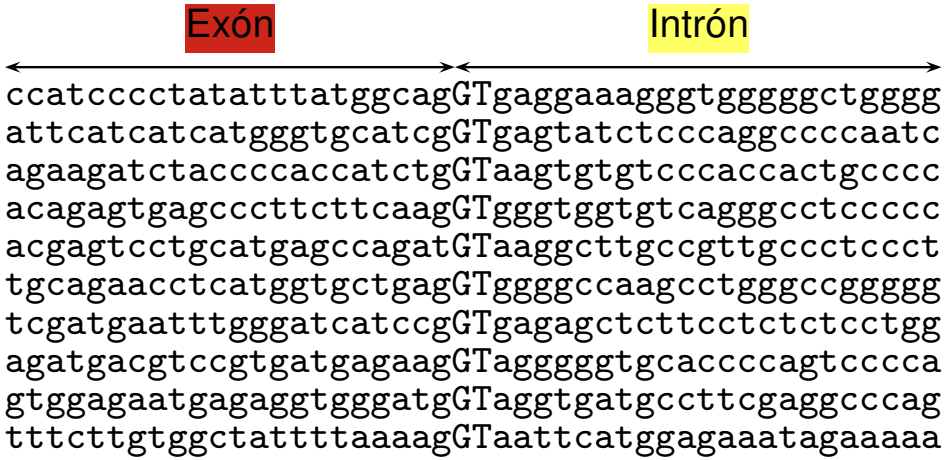
Ako spoznáme gény?

Signály na hraniciach exónov:

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



Príklad signálu: miesto zostrihu



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 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
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

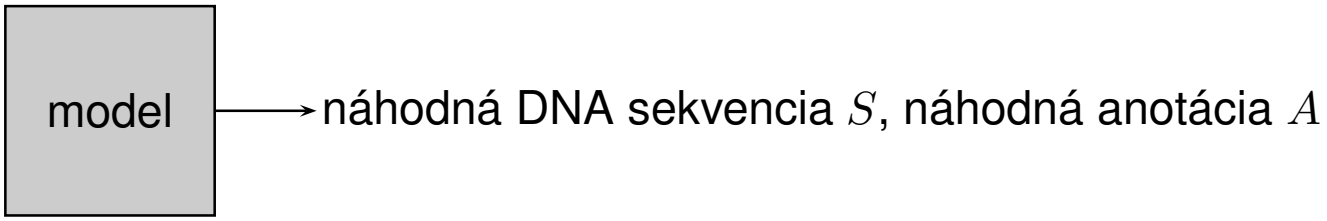
Cieľ: označ každú bázu ako intrón/exón/medzigénovú oblasť (anotácia)

- 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 (alebo: segmentáciu pôvodnej sekvencie na neprekrývajúce sa regióny, ktoré reprezentujú intróny, exóny a medzigénové úseky) **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.



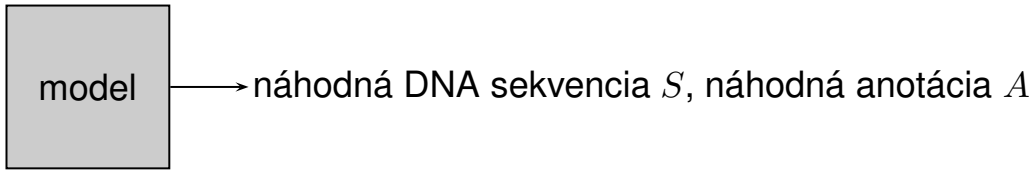
$\Pr(S, A)$ – pravdepodobnosť, že model vygeneruje pár (S, A).

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

Použitie: pre novú sekvenciu S nájdi najpravdepodobnejšiu anotáciu

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

Pravdepodobnostný model génov



Použitie: pre sekvenciu S nájdi najpravdepodobnejšiu anotáciu A

Hračkársky príklad modelu: sekvencie dĺžky 2

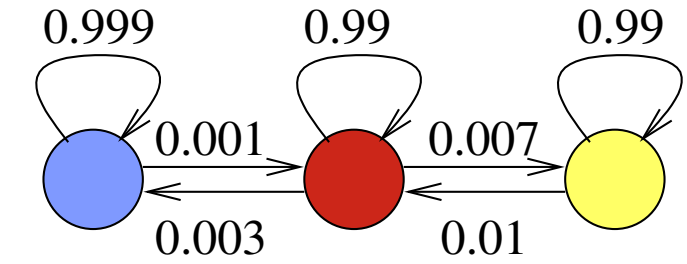
Tabuľka pravdepodobností pre 16 sekvencií, 9 anotácií (súčet 1)

Najpravdepodobnejšia anotácia pre $S =$aa je aa.

aa	0.008	ac	0.009	ag	0.0085	...
aa	0	ac	0	...		
aa	0.011	...				
aa	0					
aa	0.009					
aa	0					
aa	0.007					
aa	0					
aa	0.010					

Skrytý Markovov model, hidden Markov model (HMM)

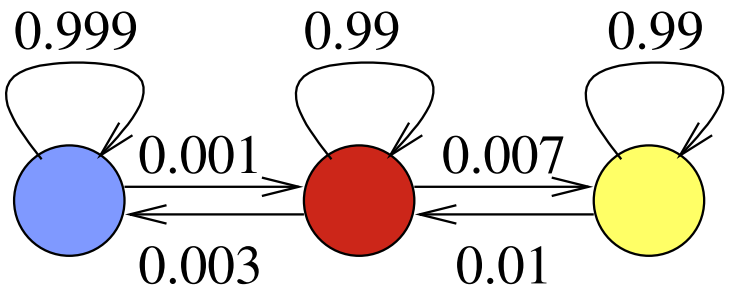
Spôsob, ako zadefinovať model pre dlhšie sekvencie.



a:0.27	a:0.24	a:0.26
c:0.23	c:0.27	c:0.22
g:0.23	g:0.28	g:0.22
t:0.27	t:0.21	t:0.30

- 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)



a:0.27	a:0.24	a:0.26
c:0.23	c:0.27	c:0.22
g:0.23	g:0.28	g:0.22
t:0.27	t:0.21	t:0.30

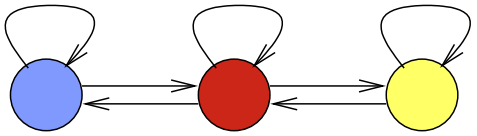
Predpokladajme, že model vždy začína v modrom stave.

Príklad:

$$\Pr(\text{aca}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 = 0.000017$$

$$\Pr(\text{aca}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 = 0.017$$

Matematické označenie



Sekvencia $S_1, \ldots, S_n$

Anotácia $A_1, \ldots, A_n$

Parametre modelu:

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

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

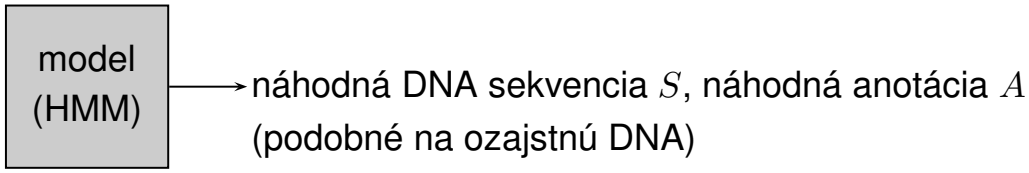
Počiatočná pravdepodobnosť $\pi(u) = \Pr(A_1 = u)$.

a			
	0.99	0.007	0.003
	0.01	0.99	0
	0.001	0	0.999

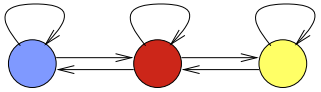
e	a	c	g	t
	0.24	0.27	0.28	0.21
	0.26	0.22	0.22	0.30
	0.27	0.23	0.23	0.27

Výsledná pravdepodobnosť: $\Pr(A_1, \ldots, A_n, S_1, \ldots, S_n) =$
$\pi(A_1)e(A_1, S_1) \prod_{i=2}^{n} a(A_{i-1}, A_i)e(A_i, S_i)$

Hľadanie génov s HMM

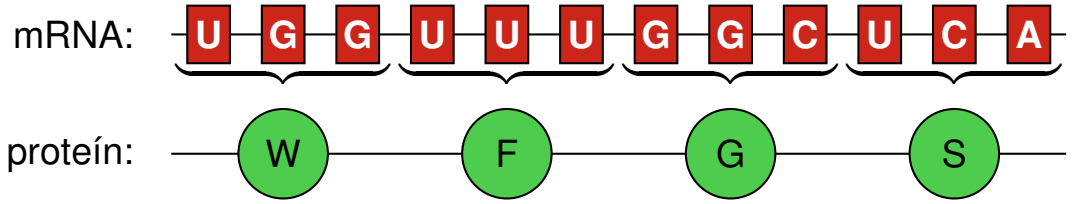


$\Pr(S, A)$ – pravdepodobnosť, že model vygeneruje pár (S, A).

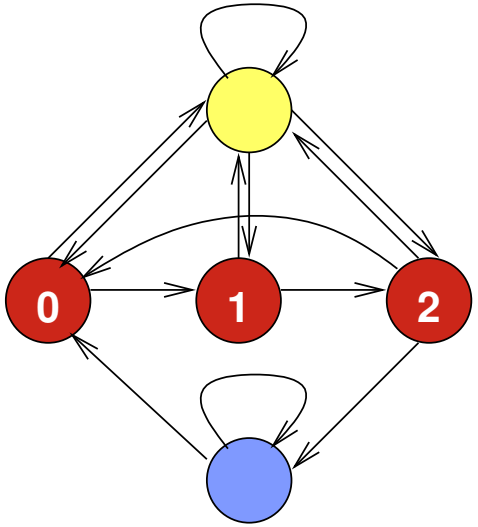
- **Určenie stavov a prechodov v modeli:** ručne, na základe poznatkov o štruktúre génu. 
- **Trénovanie parametrov:** emisné a prechodové pravdepodobnosti určíme na základe sekvencií so známymi génmi (**trénovacia množina**).
- **Použitie:** pre novú sekvenciu S nájdi najpravdepodobnejšiu anotáciu $A = \arg\max_A \Pr(A|S)$
  Viterbiho algoritmus v čase $O(nm^2)$ (dynamické programovanie)

HMM na hľadanie génov: 3-periodické exóny

Kodón (trojica báz) $\rightarrow$ jedna aminokyselina



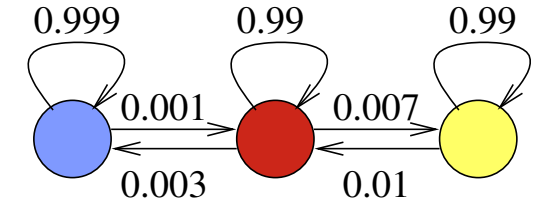
Namiesto jedného stavu pre exón použijeme tri stavy v cykle.

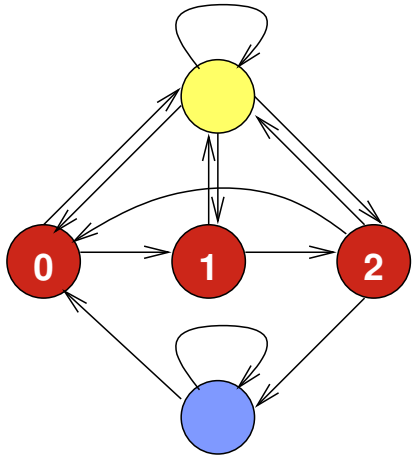


a	0	1	2		
0	0		0		0
1	0	0			0
2		0	0		
					0
		0	0	0	

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

Nové stavy majú odlišné emisné pravdepodobnosti



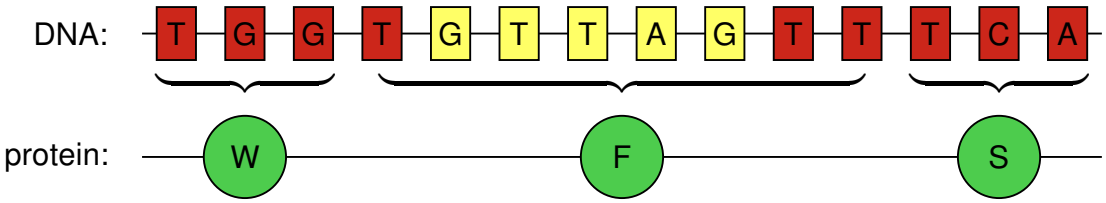


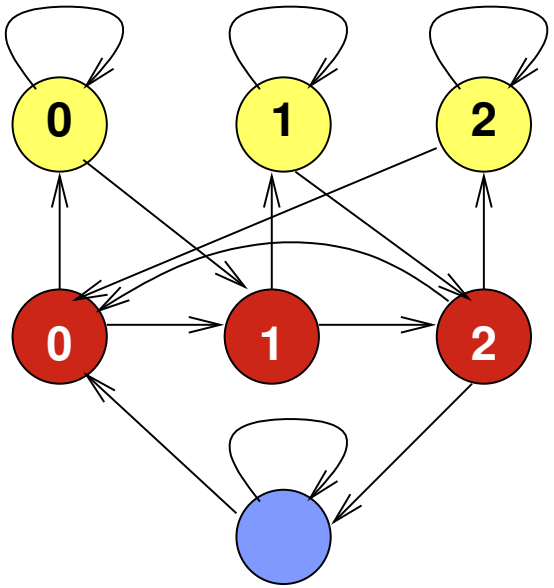
e	a	c	g	t
■	0.24	0.27	0.28	0.21
■	0.26	0.22	0.22	0.30
■	0.27	0.23	0.23	0.27

e	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

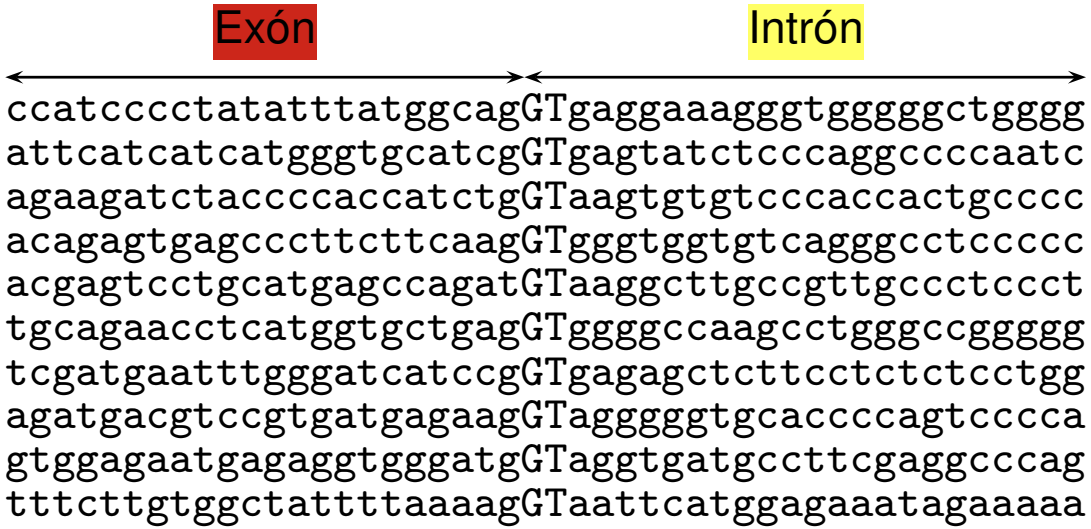
HMM na hľadanie génov: konzistentné kodóny

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

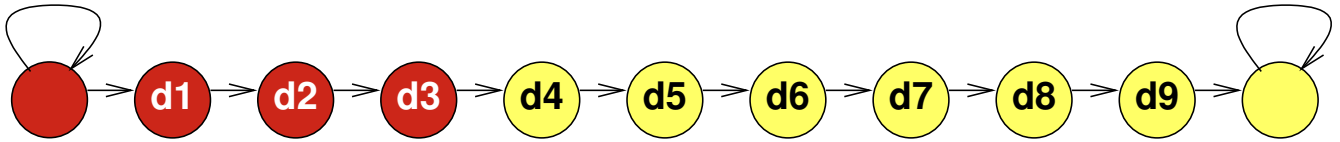




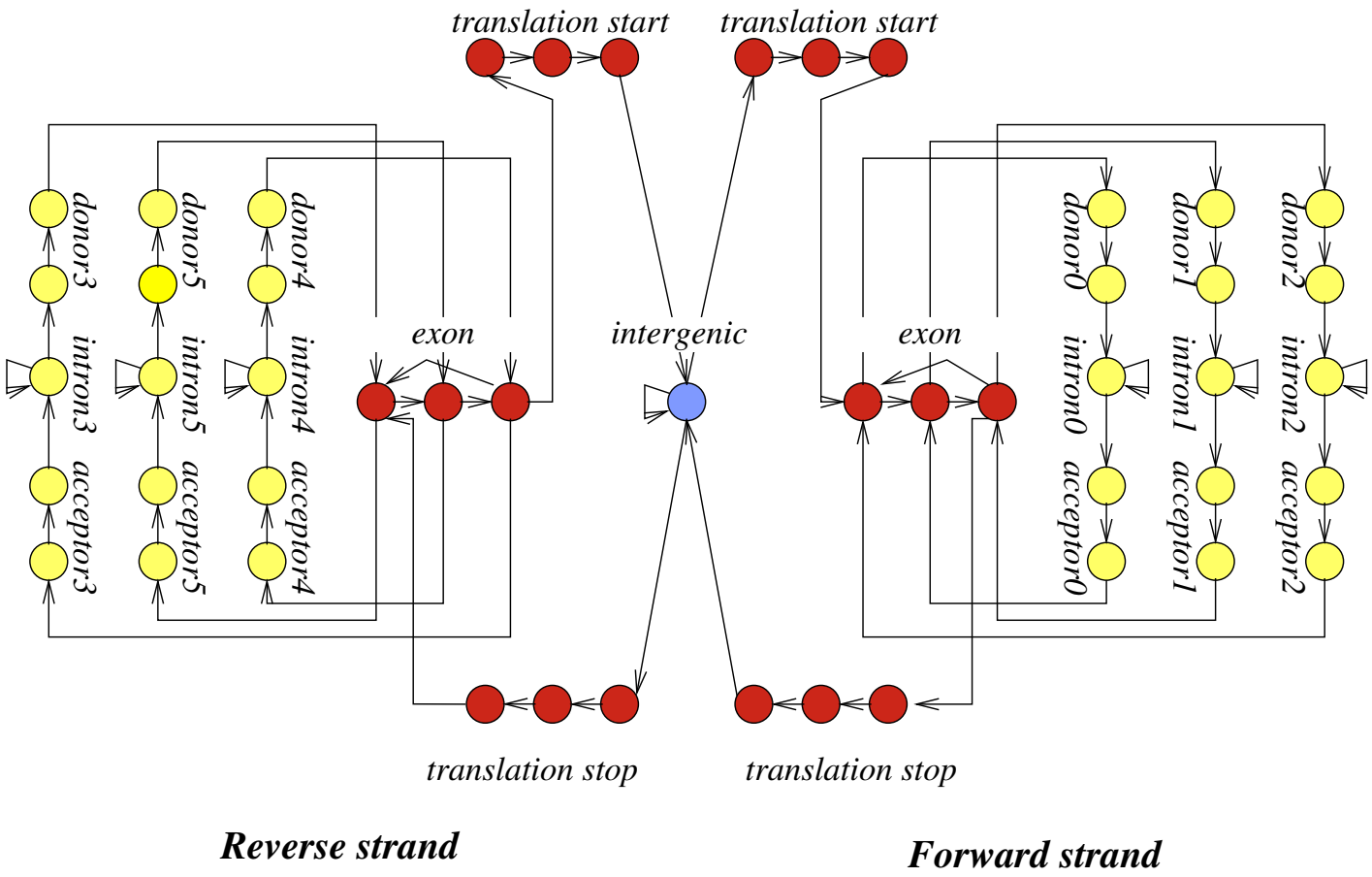
HMM na hľadanie génov: signály



Pridaj sériu stavov medzi exón a intrón:



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

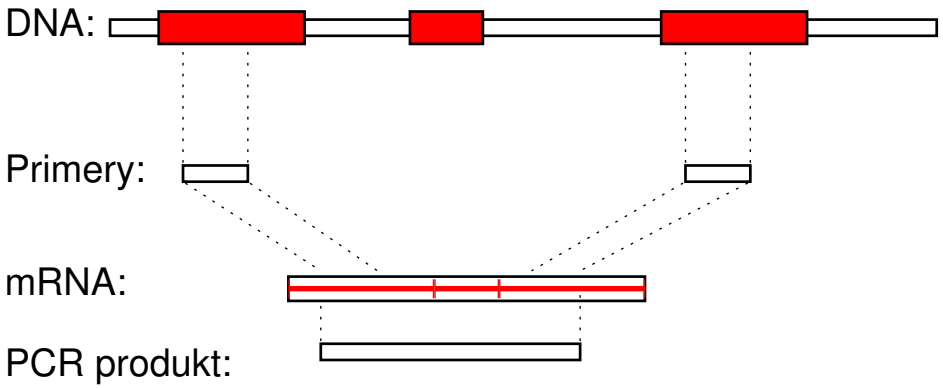
...

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

Experimentálne overovanie génov

Overenie transkripcie a zostrihu

- **RNA-Seq:** sekvenovanie častí mRNA extrahovaných z bunky. Nie je cielené 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 a Karlin 1997] (po mnohé roky štandard),

GeneZilla [Majoros a kol. 2004], ExonHunter [Brejová a kol. 2005], Augustus [Stanke a Waack 2003] (novšie programy založené na zovšeobecnených HMM).

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

Prokaryotické genómy:

GeneMark [Lukashin a Borodovsky 1998], Glimmer [Delcher a kol. 1999] a ďalšie.

Vybrané programy na hľadanie génov

Porovnávaním viacerých sekvencií:

Twinscan [Korf a kol. 2001]

(prvý úspešný gene finder s dvoma genómami),

Exoniphy [Siepel a Haussler 2004]

(viacero genómov, nehľadá celé gény),

N-SCAN [Gross a Brent 2006]

(rozšírenie Twinscanu na viacero genómov).

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

ExonHunter [Brejová a kol. 2005], Augustus [Stanke a kol. 2006],

Jigsaw [Allen a Salzberg 2005], Fgenesh++ [Solovyev 2006].

Augustus patrí dodnes medzi často používané programy.

Obmedzenia hľadačov génov

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

- 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.

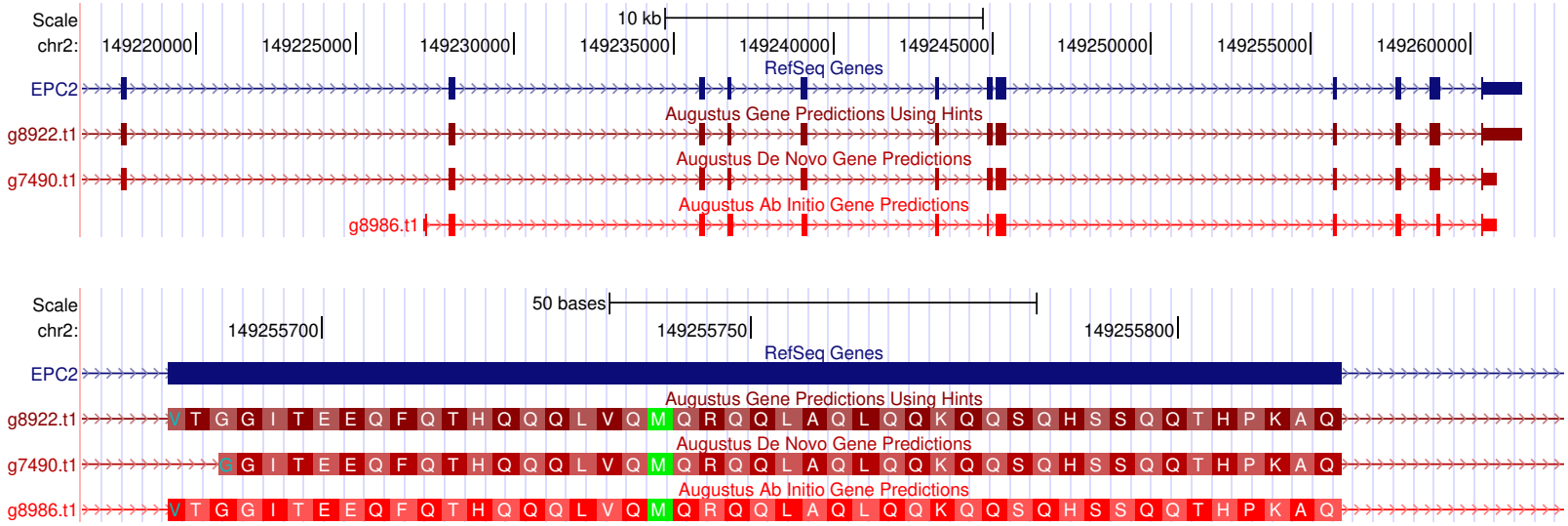
Hľadače génov robia často chyby

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 génov má človek?

Do 2001: Rôzne odhady: **50 000–140 000** génov

2001: predbežná verzia ľudského genómu: **30 000–40 000** génov

2004: sekvencia ľudského genómu: **20 000–25 000** génov

2007: v katalógoch Ensembl, RefSeq a VEGA spolu **24 500** génov

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

Ale sú gény, o ktorých ešte nevieme?

2010: RefSeq má **22 333** génov

Stále neistota ± 1000 [Pertea, Salzberg 2010]

Rôzni ľudia sa môžu líšiť v desiatkach génov

2012: Projekt ENCODE odhaduje **20 687** génov kódujúcich proteíny,

v priemere 6 alternatívnych transkriptov na gén,

plus 8 800 krátkych a 9 600 dlhých RNA génov

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: skupiny

- Zoznam skupín na stránke predmetu, každej skupine tiež príde spoločný email, aby ste sa vedeli navzájom nakontaktovať
- Niektoré skupiny sú anglicky hovoriace

Journal club: prvé stretnutie - pred 22.11.

- Každý si najprv prečíta článok,
  potom sa koná stretnutie, kde o článku diskutujete, vysvetlíte si navzájom nejasnosti, plánujete písanie správy
- Dátum, čas a miesto oznámte aspoň 24 hodín vopred v diskusii na Moodli
- Po stretnutí napíšte krátku správu zo stretnutia do príslušnej položky v Moodli (kto sa zúčastnil, čo sa dohodlo, či sú nejaké problémy, stačí pár viet)
- 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)
- Vysvetlite pojmy, ktoré sú nad rámec tohto predmetu
- Netreba pokryť všetko, môžete využiť aj iné zdroje
- Podrobne vysvetliť aspoň jednu bioinformatickú metódu a aspoň jeden biologický výsledok (alebo overovanie správnosti metódy na dátach)
- Ako článok súvisí s učivom preberaným na predmete
- Nájdite zopár citujúcich prác, ktoré výsledky využili alebo vylepšili
- 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 do budúcej stredy 13.11. 22:00

  Otázky k zadaniu emailom

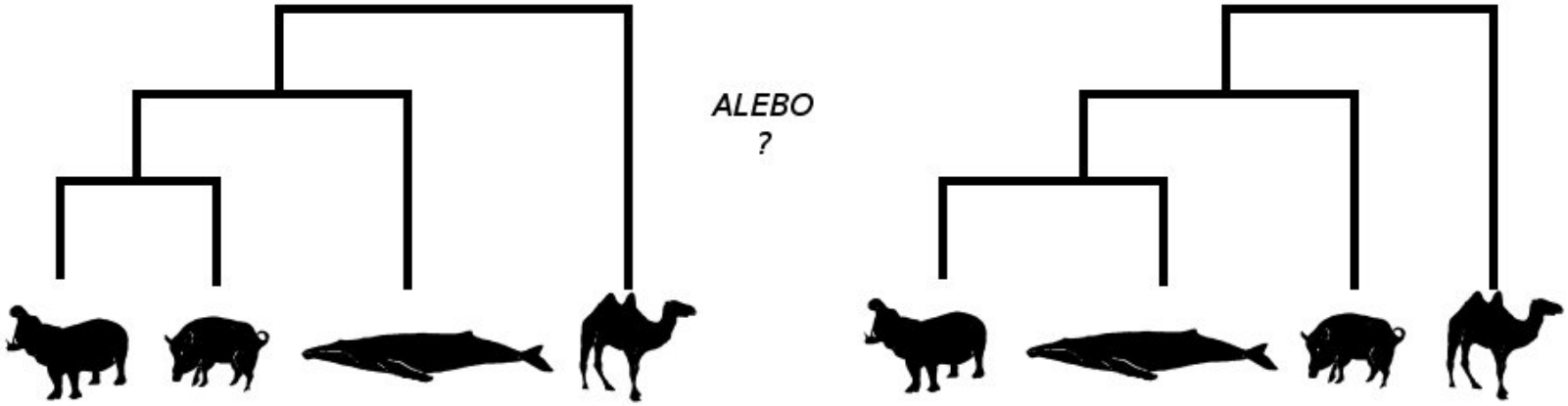
- Pracujte na journal clube

  (prečítajte si článok, naplánujte si stretnutie pred 22.11.)

Evolučné modely a stromy

Broňa Brejová

07.11.2024



Rekonštrukcia fylogenetických stromov

Vstup:

m **zarovnaných** sekvencií,

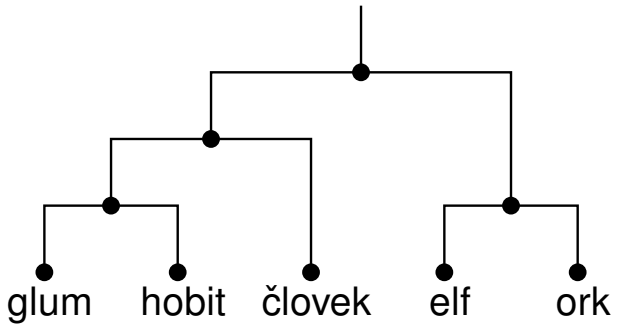
každá dĺžky n

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

Výstup:

strom predstavujúci

ich evolučnú históriu

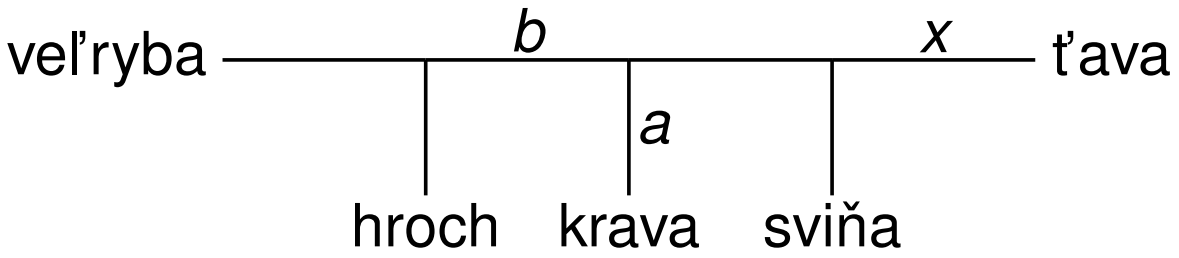


Newick format:

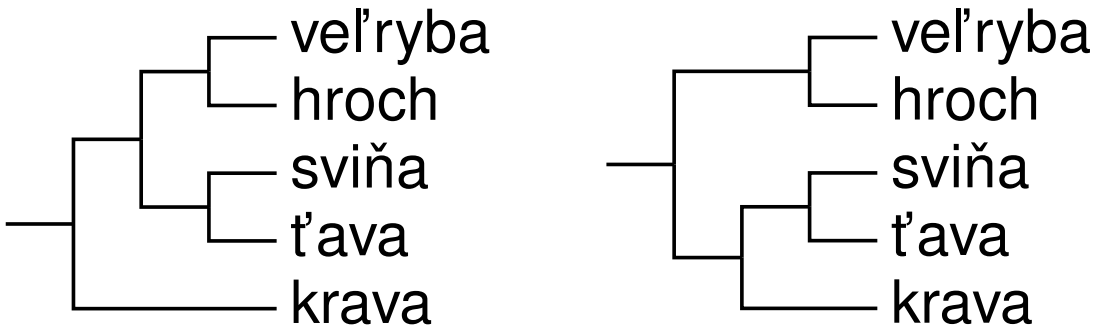
(((glum,hobit),človek),(elf,ork))

Zakorenené a nezakorenené stromy

Nezakorenený strom (unrooted tree)



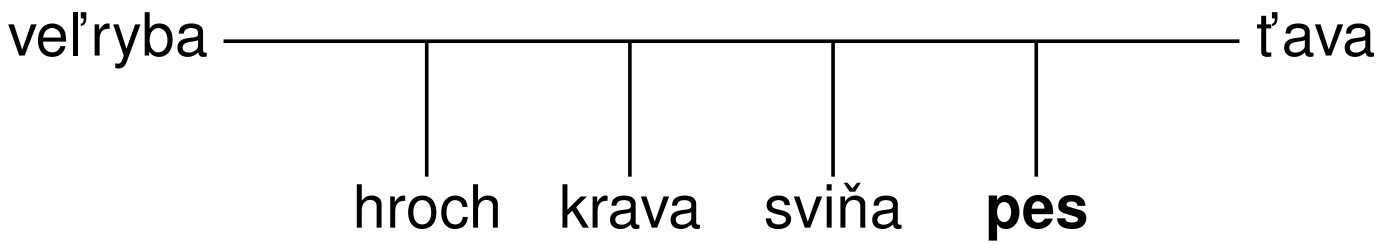
Dve zo siedmich zakorenených verzií stromu

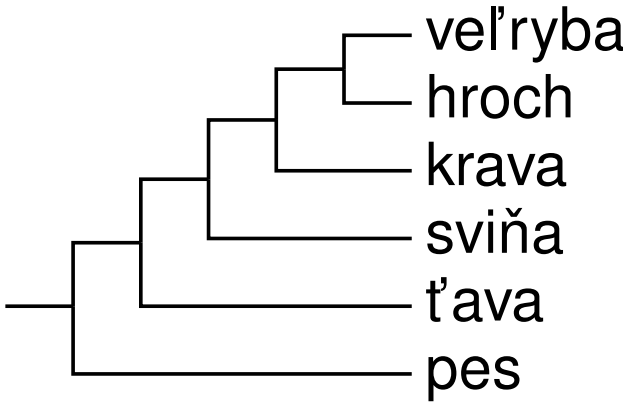


Väčšina metód rekonštruuje nezakorenené stromy

Zakorenenie stormu pomocou vonkajšej skupiny

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





Maximum parsimony (úsporné stromy)

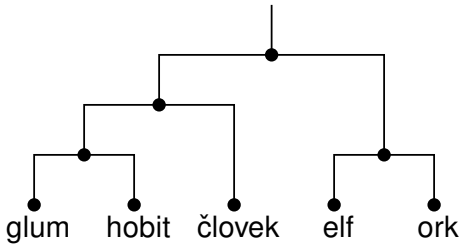
Úloha: Dané sú zarovnané sekvencie súčasných organizmov.

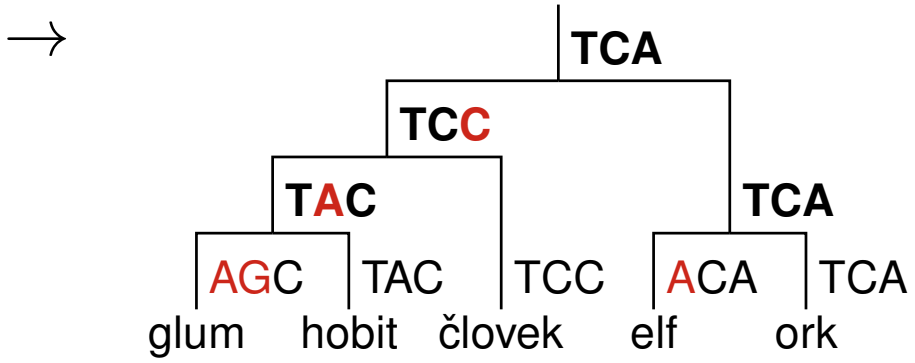
Chceme nájsť fylogenetický strom, ktorý vyžaduje **minimálny počet evolučných zmien**.

Evolučná zmena = mutácia jednej bázy na inú bázu

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

glum	AGC
hobit	TAC
človek	TCC
elf	ACA
ork	TCA

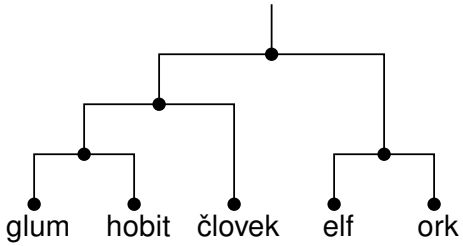


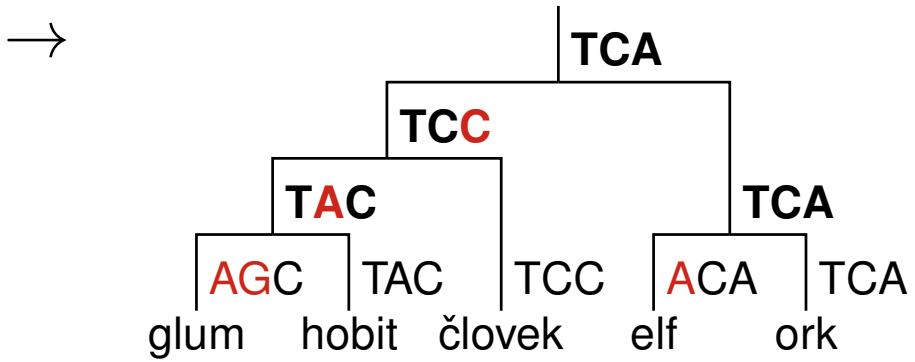


5 zmien

Podotázka: Výpočet ceny konkrétneho stromu

glum	AGC
hobit	TAC
človek	TCC
elf	ACA
ork	TCA





5 zmien

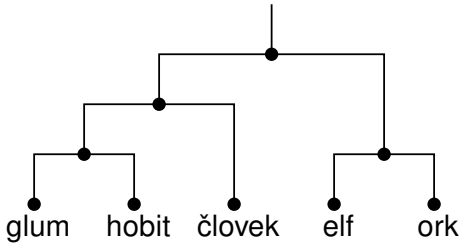
Môžeme rátať **dynamickým programovaním** pre každý stĺpec zarovnania zvlášť (cvičenia informatici).

Časová zložitosť: $O(m)$, lineárna

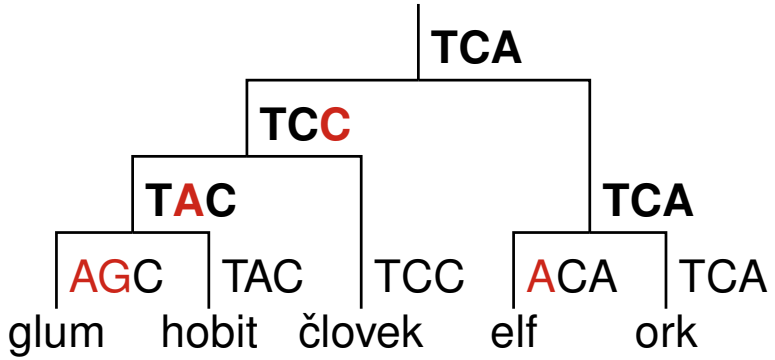
Zopakujeme pre každý stĺpec zarovnania: $O(mn)$

Vieme: Výpočet ceny konkrétneho stromu

glum	AGC
hobit	TAC
človek	TCC
elf	ACA
ork	TCA



→

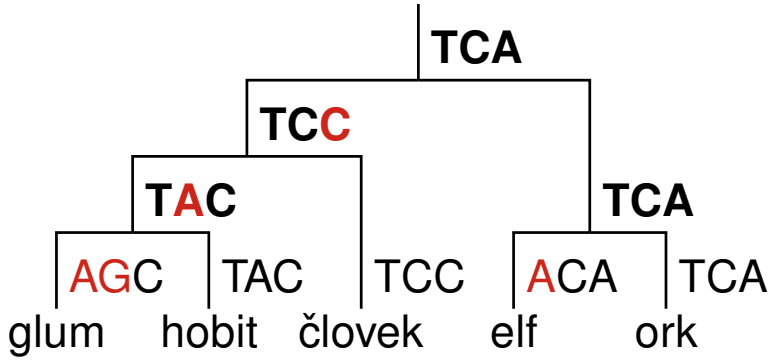


5 zmien

Chceme: Nájsť strom s najmenšou cenou

glum	AGC
hobit	TAC
človek	TCC
elf	ACA
ork	TCA

→



Hľadanie najúspornejšieho stromu

NP-ťažký problém

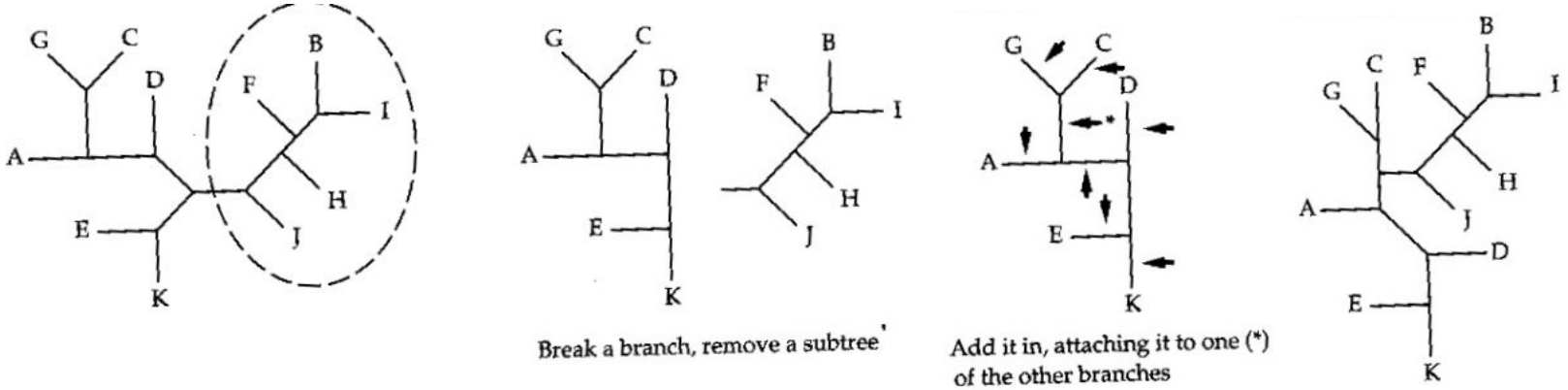
Triviálny algoritmus: vyskúšaj všetky možné stromy.

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

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

Heuristické prehľadávanie:

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



Neighbor Joining (Metóda spájania susedov)

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

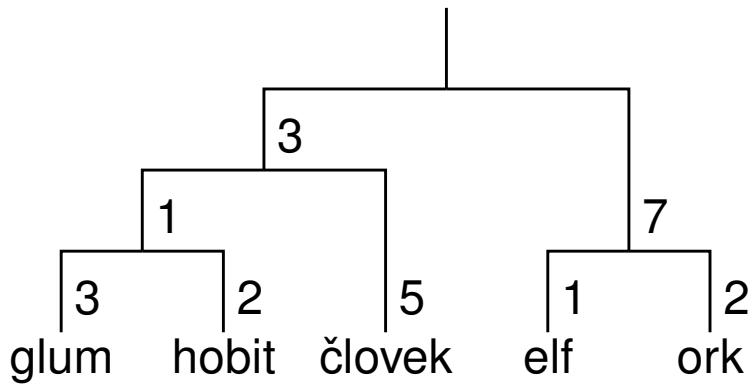
Jednoduchý príklad:

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

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

Idea spájania susedov

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



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

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

Idea spájania susedov

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

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

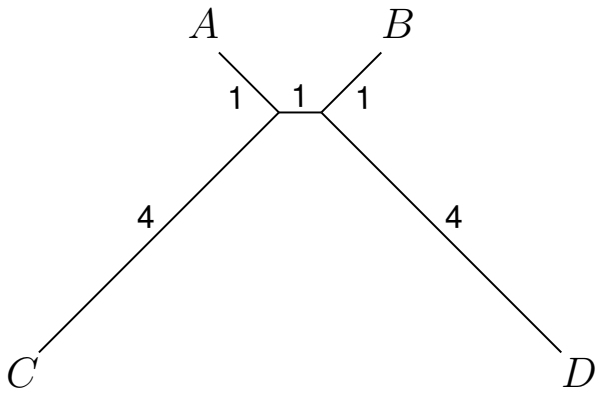
	g	h	č	e	o
g	0	5	9	15	16
h	5	0	8	14	15
č	9	8	0	16	17
e	15	14	16	0	3
o	16	15	17	3	0

Spojíme e a o
$\longrightarrow$

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

Ako určiť dva listy na spájanie?

(Prečo nie dva najbližšie?)



	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:

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

m: počet listov

r_i: súčet riadku resp. stĺpca i

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

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

D

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

L

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

nové D

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

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

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

Spájanie susedov: zhrnutie

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

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

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

Problém so vzdialenosťami

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

Jukesov-Cantorov model evolúcie

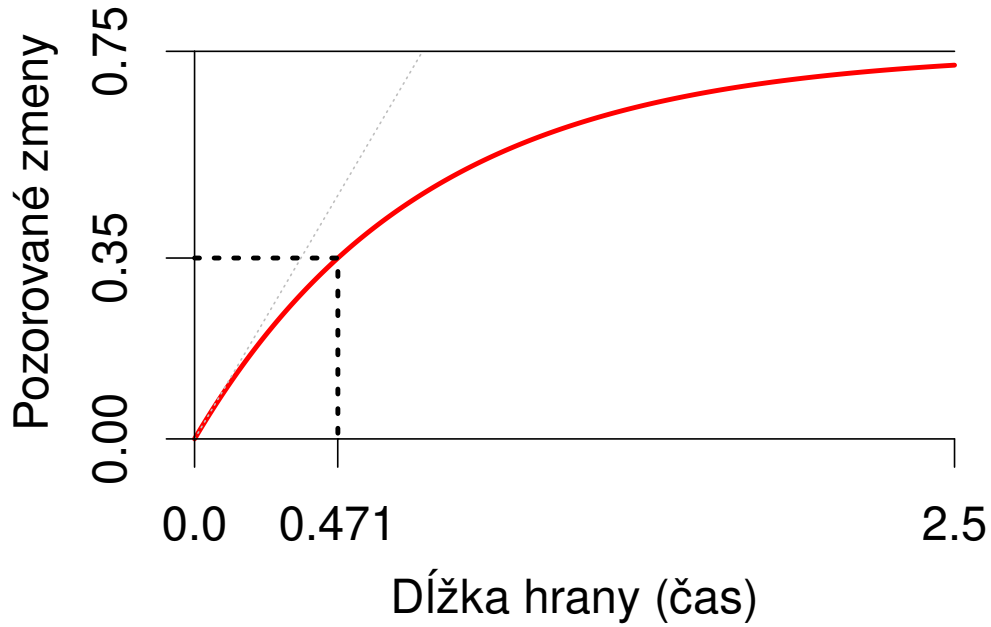
Pravdepodobnosť zmeny bázy na inú:

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

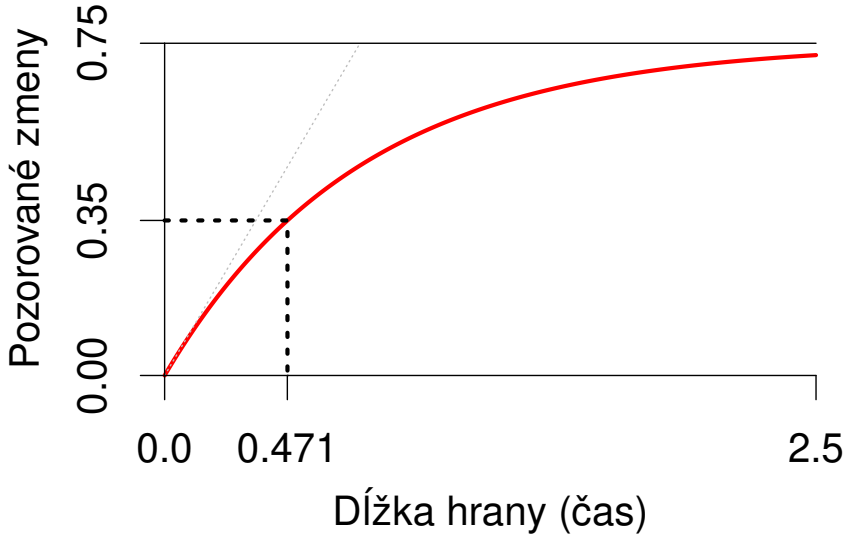
α: rýchlosť evolúcie (počet substitúcií na jednotku času)

Očakávaný počet pozorovaných zmien na bázu za čas t:

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



Späť ku spájaniu susedov (Neighbor Joining)



- Podľa takéhoto modelu môžeme korigovať pozorované vzdialenosti

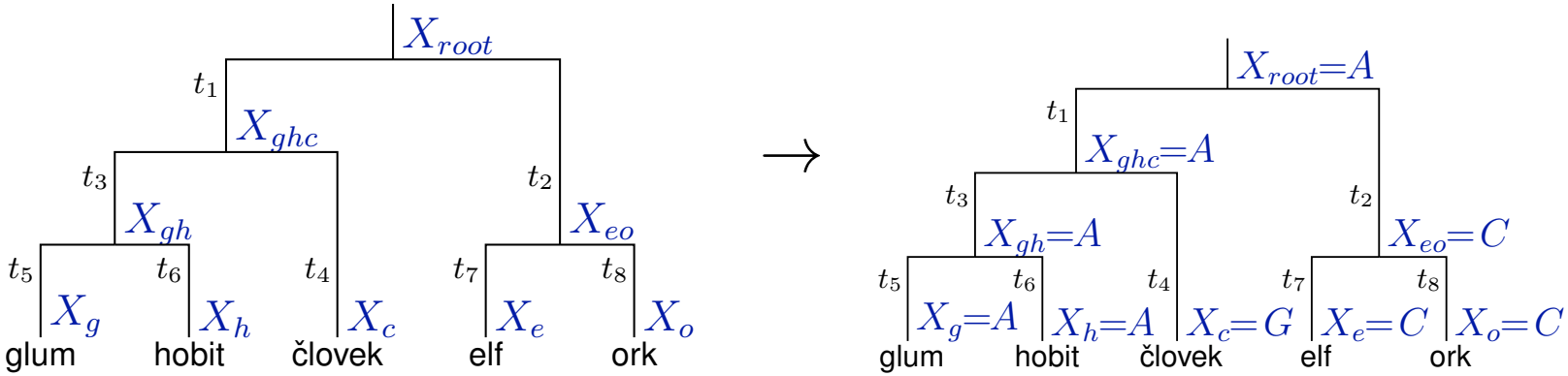
$$D = \frac{3}{4}\left(1 - e^{-\frac{4}{3}\alpha t}\right) \qquad \Rightarrow \qquad \alpha t = -\frac{3}{4}\ln\left(1 - \frac{4}{3}D\right)$$

- Nabudúce / na cvičeniach uvidíme aj zložitejšie modely evolúcie

Najvierohodnejšie stromy (Maximum likelihood)

Strom s danými dĺžkami hrán môžeme chápať

ako **jednoduchý generatívny model**

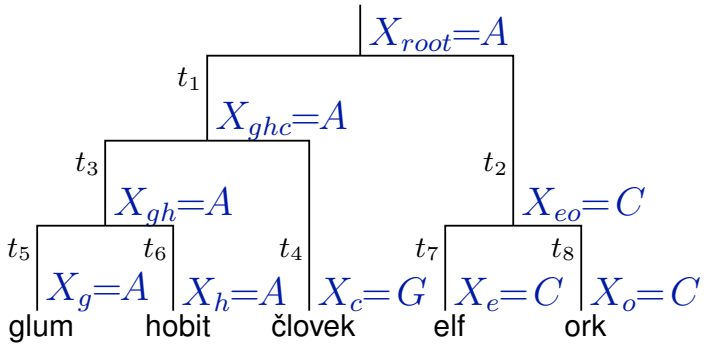


Pravdepodobnosť, že vygeneruje konkrétne bázy vo vrcholoch:

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

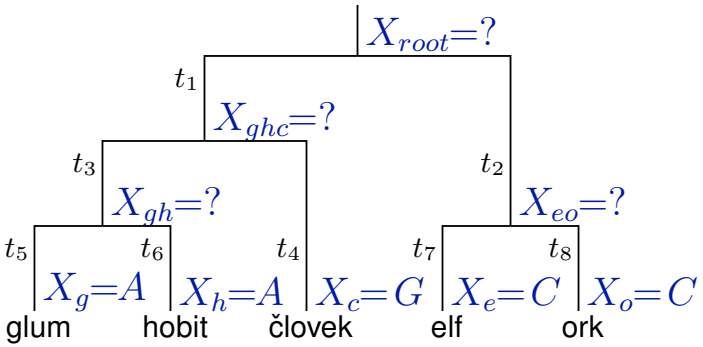
$\Pr(A \xrightarrow{t_2} C)$ je skratka za $\Pr(X_{eo} = C \mid X_{root} = A)$

Vieme počítať (súčin):



Chceme počítať

vierohodnosť stromu:



Vierohodnosť (likelihood) stromu:

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

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

Rátame pomocou **Felsensteinovho algoritmu**

(jednoduché dynamické programovanie, podobne ako pre úspornosť)

Pre dané zarovnanie, strom a dĺžky hrán

spočíta vierohodnosť v čase $O(nm)$

Ako nájsť najvierohodnejší strom?

- Problém je NP-ťažký ;

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

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

Konzistentnosť fylogenetických algoritmov

- "Rozumne" správajúce sa algoritmy: ak dĺžka sekvencií (n) rastie, ich odpoveď by sa mala približovať ku správnej odpovedi.
- 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	$O(m^2)$	ÁNO	iba vzdialenosti
Likelihood (vierohodnosť)	NP-ťažký	ÁNO	celé sekvencie

Odkiaľ zohnať dáta pre fylogenetiku?

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

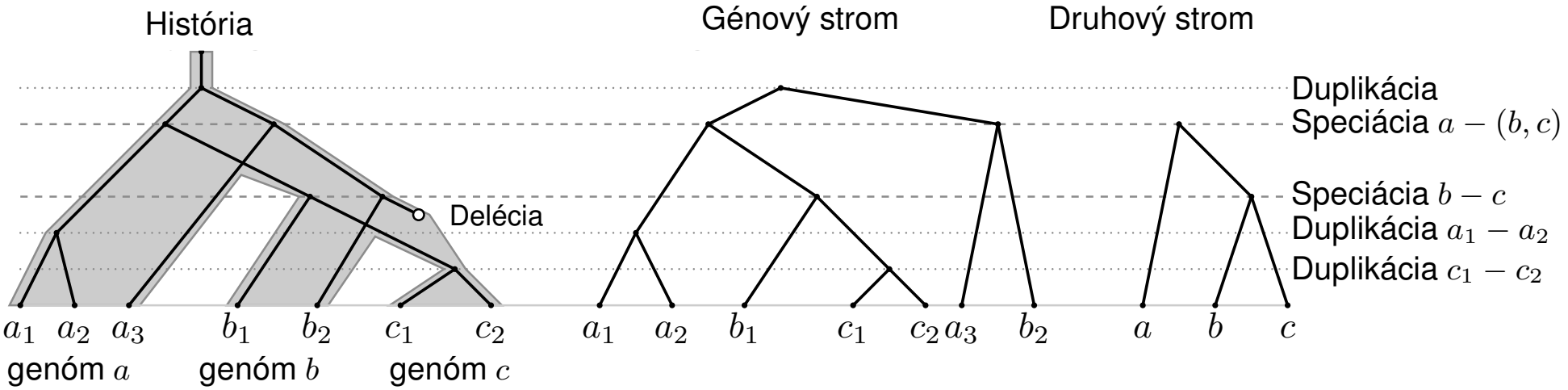
Chceme využiť aj dalšie časti genómu. Čo tak:

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

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

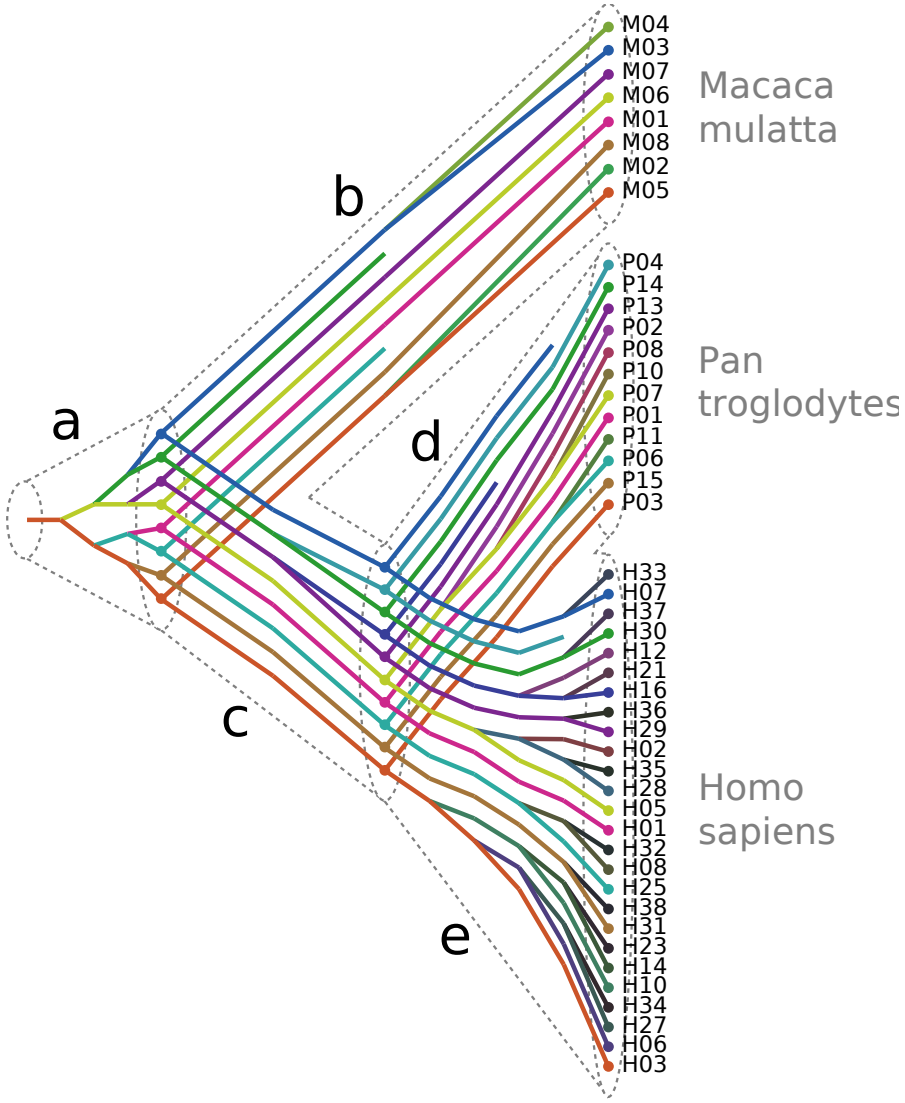
História duplikovaného génu

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



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

Zložitejší príklad duplikácie génu:



Zhrnutie

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

Organizačné poznámky

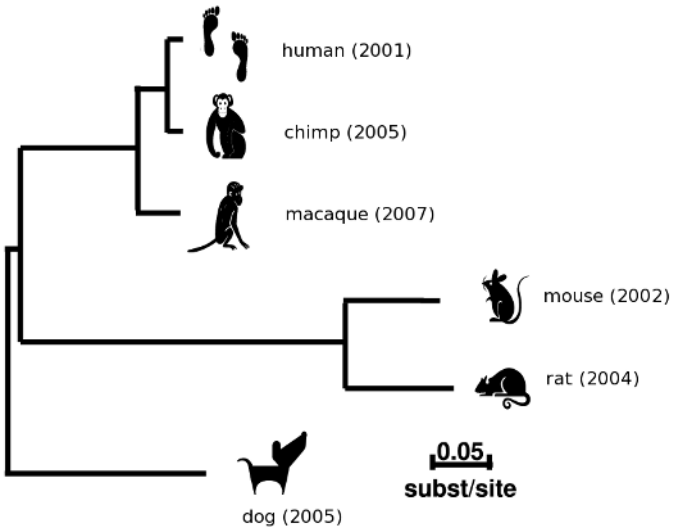
- Domáca úloha 2 bude zverejnená zajtra, dátum odovzdania 4.12. 22:00
  Informatici budú implementovať algoritmus, ktorý bude preberaný budúci týždeň na cvičeniach

- Nezabudnite na **prvé stretnutie** ohľadom journal clubu
  (najneskôr 22.11., osobne alebo online).
  Pred stretnutím oznámte čas a miesto do diskusie na Moodli
  Po stretnutí napíšte krátku správu v Moodli
  (kto sa zúčastnil, čo sa dohodlo, či sú nejaké problémy, stačí pár viet)

Komparatívna genomika

Tomáš Vinař

14.11.2024



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)

    $\Rightarrow$ **Purifikačný výber (purifying selection)**
  - Prospešné (advantageous)

    $\Rightarrow$ **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 zarovnania 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---GGCTGCCGAGGCCTTGTTCGTGGGGAGGT
Armadillo AGTGGCTGCCGGGCTG---GGAGGCCAAGGCCTTGTTCGCGGGCAGGT
   Chicken AGTGGCTGCCAGTCTGCGCCGTGGCCGACGTCTTGCTCGGGGGAAGGT
   X. trop AATGGCTTCCATTTTGTGCCGCTGCTGAGGTCTTGTTCTGGGGAAGAT
```

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

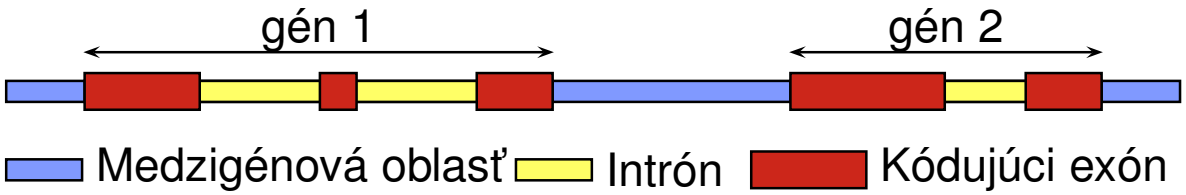
Príklad 1: Hľadanie funkčných oblastí sekvencií

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

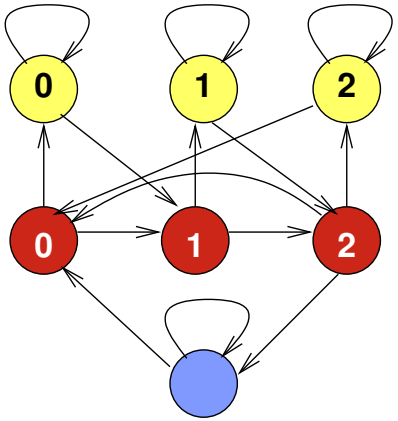
Opakovanie: hľadanie génov

Úlohou je nájsť polohu génov v genóme a ich exónovú štruktúru.



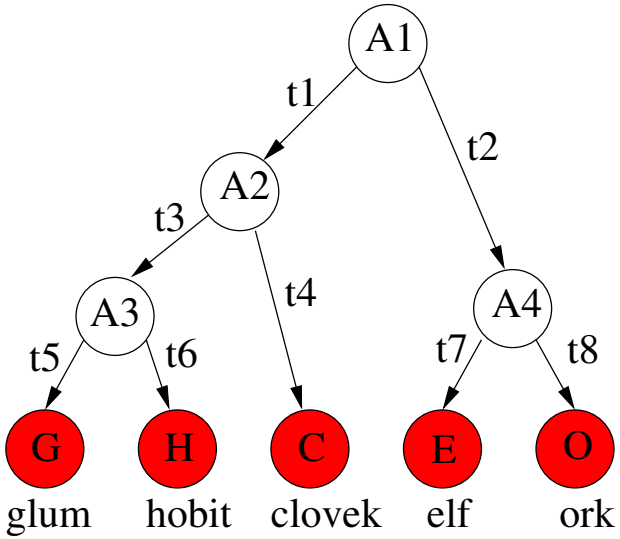
Vytvoríme skrytý Markovovský model (HMM), ktorý vie generovať sekvencie a ich anotácie podobné skutočným.

Pýtame sa, ktorá anotácia je najpravdepodobnejší pár k danej sekvencii.



Opakovanie: pravdepodobnostné modely evolúcie

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

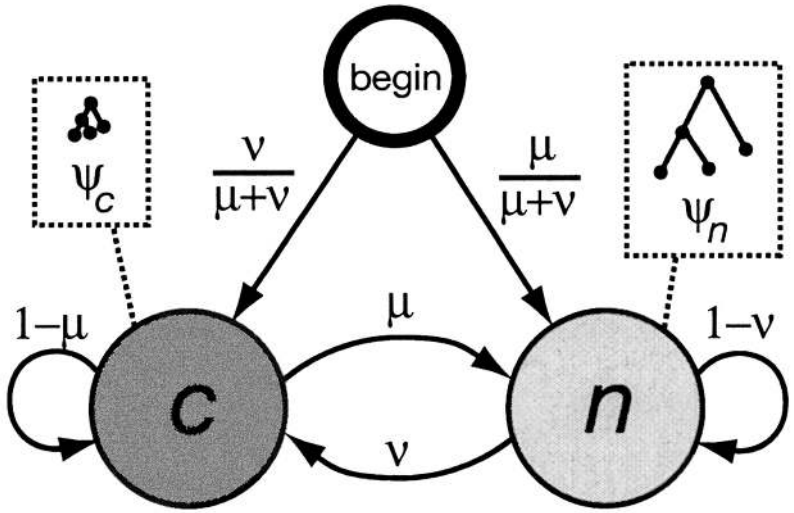


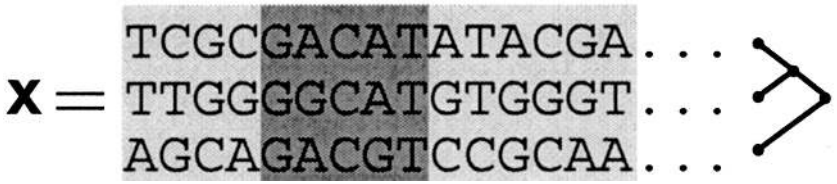
- Pre hranu z Y do X dĺžky t možno pravdepodobnosť mutácie spočítať použitím evolučného modelu, napr. Jukes-Cantor:
  $\Pr(A \xrightarrow{t} C) = \frac{1}{4}(1 - e^{-\frac{4}{3}\alpha t})$

- Pre celý strom $\Pr(G, H, C, E, O, A1, \ldots, A4) = \Pr(A1) \cdot \Pr(A1 \xrightarrow{t_1} A2) \cdot \Pr(A1 \xrightarrow{t_2} A4) \cdot \Pr(A2 \xrightarrow{t_3} A3) \cdot \Pr(A2 \xrightarrow{t_4} C) \cdot \Pr(A3 \xrightarrow{t_5} G) \cdot \Pr(A3 \xrightarrow{t_6} H) \cdot \Pr(A4 \xrightarrow{t_7} E) \cdot \Pr(A4 \xrightarrow{t_8} O)$

PhastCons: detekcia dobre zachovaných sekvencií

Fylogenetické HMM: kombinácia HMM a fylogenetického stromu.

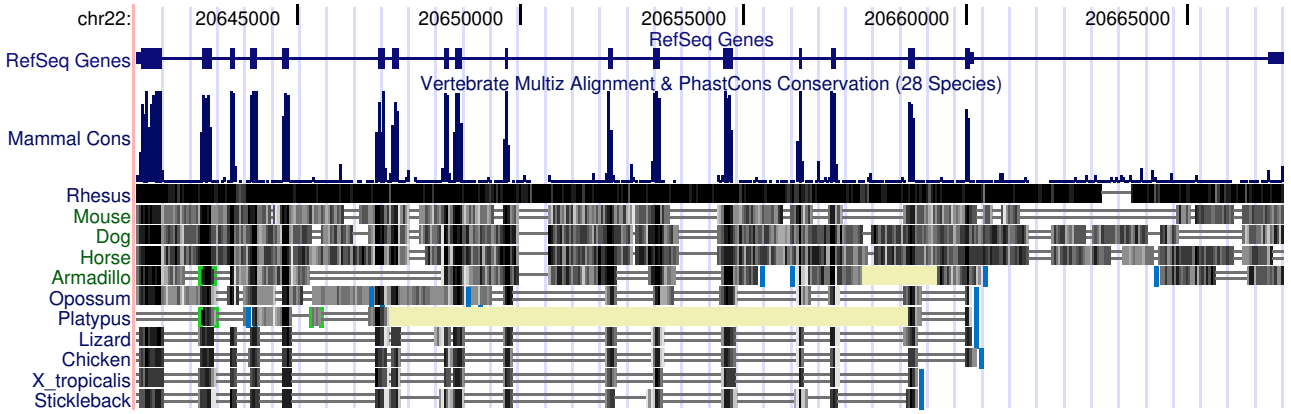




- Dva stavy: zachovaná sekv., neutrálna sekv.
- V každom stave generujeme celý stĺpec zarovnania
- Zachovaná sekvencia má kratšie hrany stromu, teda menšia divergencia sekvencií

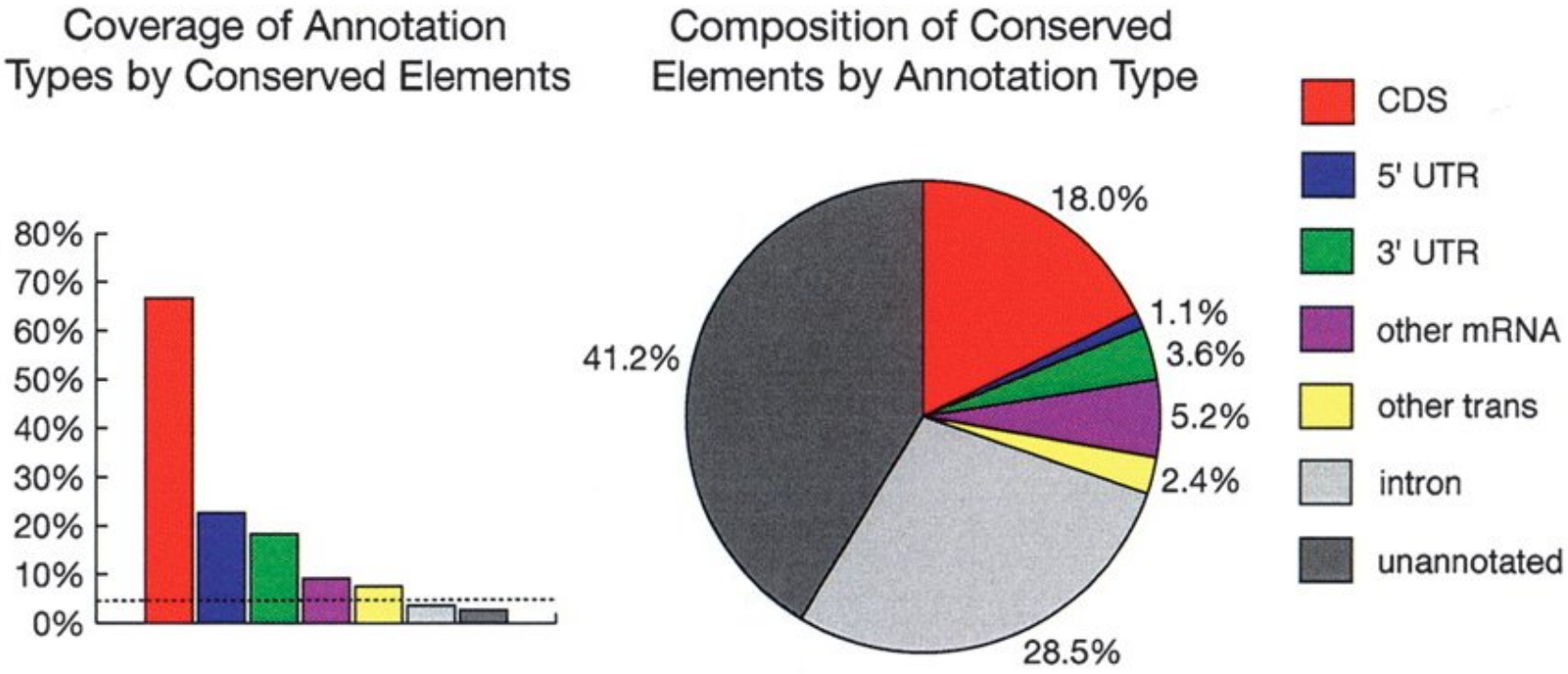
Použitie fylogenetického HMM

- Model určuje rozdelenie pravdepodobnosti cez zarovnania 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



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 zarovnania zlepšiť presnosť

	Exóny		Gény	
Program	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
- Všeobecnenší model:

  zavedieme μ_{xy} **rýchlosť substitúcie** z bázy x na bázu y
- Matica rýchlostí (substitution rate matrix)

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

  Pre daný čas t, môžeme vypočítať pravdepodobnosť každej substitúcie z bázy x na bázu y (**transition probabilities**): $\Pr(x \xrightarrow{t} y)$

Znižovanie počtu parametrov — HKY matica

Hasegawa, Kishino a Yano

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

- **ekvilibrium:** frekvencie $\pi_A, \pi_C, \pi_G, \pi_T$
- rozlišujeme **tranzície** $C \Leftrightarrow T, A \Leftrightarrow G$ a **transverzie** $\{C, T\} \Leftrightarrow \{A, G\}$
  tranzície sú κ krát častejšie (typicky $\kappa \approx 2$)
- Máme iba štyri parametre: $\pi_A, \pi_C, \pi_G, \kappa$
  (π_T sa dopočíta do 1)

Substitučný model pre kodóny

Namiesto jednotlivých báz uvažujeme trojice

Rýchosť zmeny z kodónu i na kodón j:

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

Príklad: $\mu_{AAC,GGC} = 0$, $\mu_{CTA,CTT} = \pi_{CTT}$, $\mu_{CTA,CCA} = \omega\kappa\pi_{CCA}$

Parametre: Frekvencie kodónov π_j, ω, κ

neutrálna evolúcia $\omega = 1$, pozitívny výber $\omega > 1$,

purifikačný výber $\omega < 1$

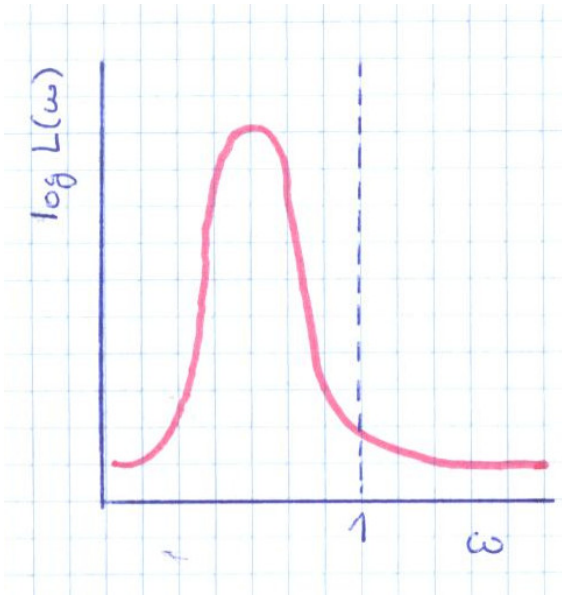
Aplikácia kodónového substitučného modelu

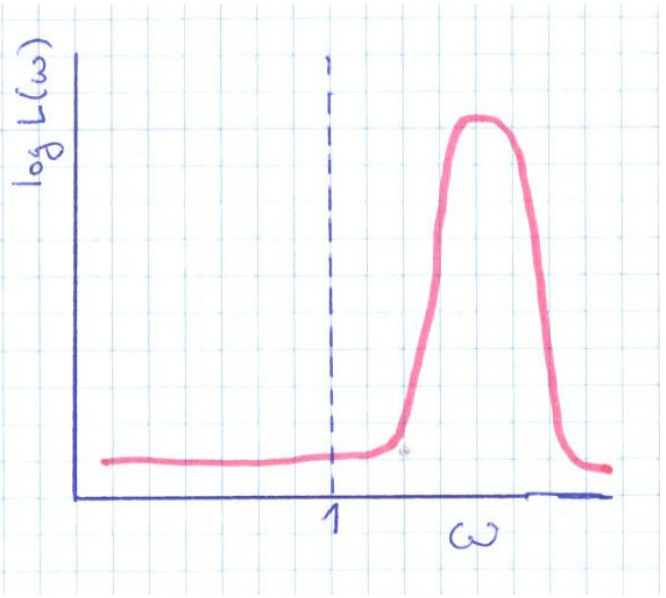
```
             F   V   I   H   D   S   E   G   D   G   E   C   M   Q   E
človek (C) TTT GTG ATC CAC GAC TCC GAG GGG GAC GGC GAG TGC ATG CAG GAG
kosmáč (K) TTT GTG ATC CAC GAG AAC AAC AAG GAC GGC GAG TGC ATG CAG GAT
             F   V   I   H   E   N   N   K   D   G   E   C   M   Q   D
```

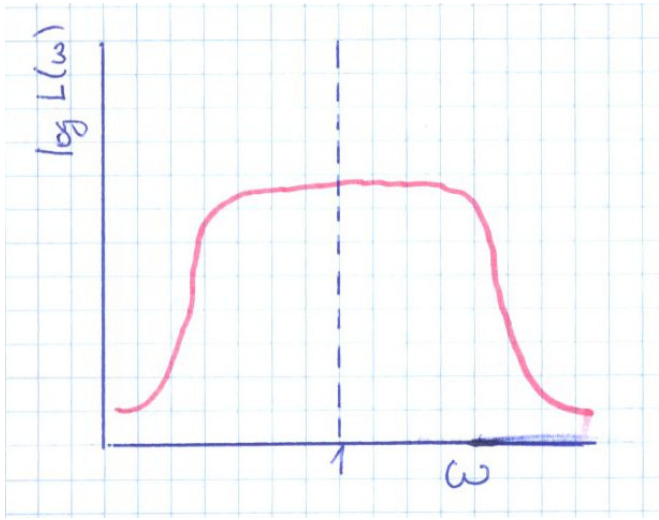
- Na základe celých genómov môžeme odhadnúť základné parametre modelu π_*, κ
- Pre dané ω a t vieme spočítať vierohodnosť

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

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


log L(ω)
1
ω


log L(ω)
1
ω


log L(ω)
1
ω

Test pomerov vierohodností (Likelihood-ratio test)

- $L(\omega)$ môže byť najväčšie pre $\omega > 1$,
  ale môže to byť spôsobené len štatistickou varianciou 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 ω)
- Vždy platí $L_B \geq L_A$
- Ak skutočné $\omega < 1$, $L_A \approx L_B$ (nulová hypotéza)
  nás zaujímajú prípady $L_B >> L_A$
  $\Rightarrow$ gén pod vplyvom pozitívneho výberu (alt. hypotéza)

Za predpokladu, že $\omega < 1$, platí $2\log(L_B/L_A) \approx \chi^2_1$
$\Rightarrow$ možno priradiť P-hodnotu nulovej hypotéze $\omega < 1$

Hľadanie génov pod vplyvom pozitívneho výberu: Zhrnutie

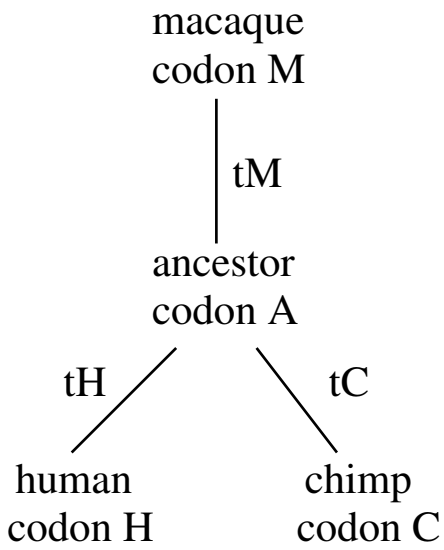
- Nájdeme zarovnanie toho istého génu z dvoch organizmov (na úrovni kodónov)
- Odhadneme základné parametre kodónového modelu na základe porovnania celých genómov
- Parameter ω modeluje selekciu
- Spočítame vierohodnosť $L_A = \max_{\omega<1} L(\omega)$
  a vierohodnosť $L_B = \max_{\omega} L(\omega)$
- 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

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

Zbavíme sa ancestrálnych sekvencií:

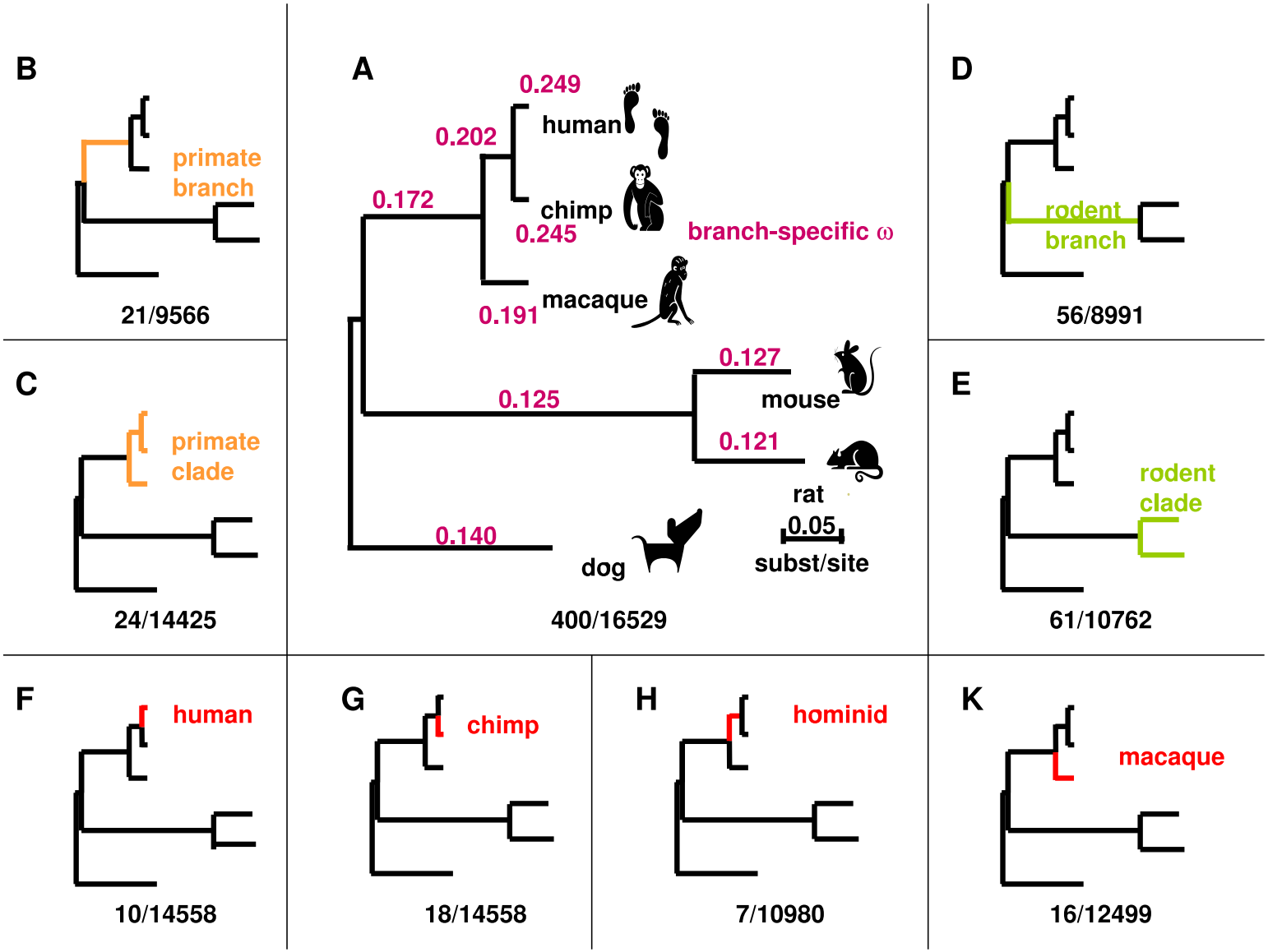
$$\Pr(H, C, M \,|\, \omega, t_H, t_C, t_M) =$$
$$\sum_A \Pr(A, H, C, M \,|\, \omega, t_H, t_C, t_M)$$



Vierohodnosť ω:

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

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


B
primate branch
21/9566
A
0.249
human
0.202
chimp
0.172
0.245
branch-specific ω
macaque
0.191
0.127
mouse
0.125
0.121
rat
0.05
subst/site
0.140
dog
400/16529
D
rodent branch
56/8991
C
primate clade
24/14425
E
rodent clade
61/10762
F
human
10/14558
G
chimp
18/14558
H
hominid
7/10980
K
macaque
16/12499

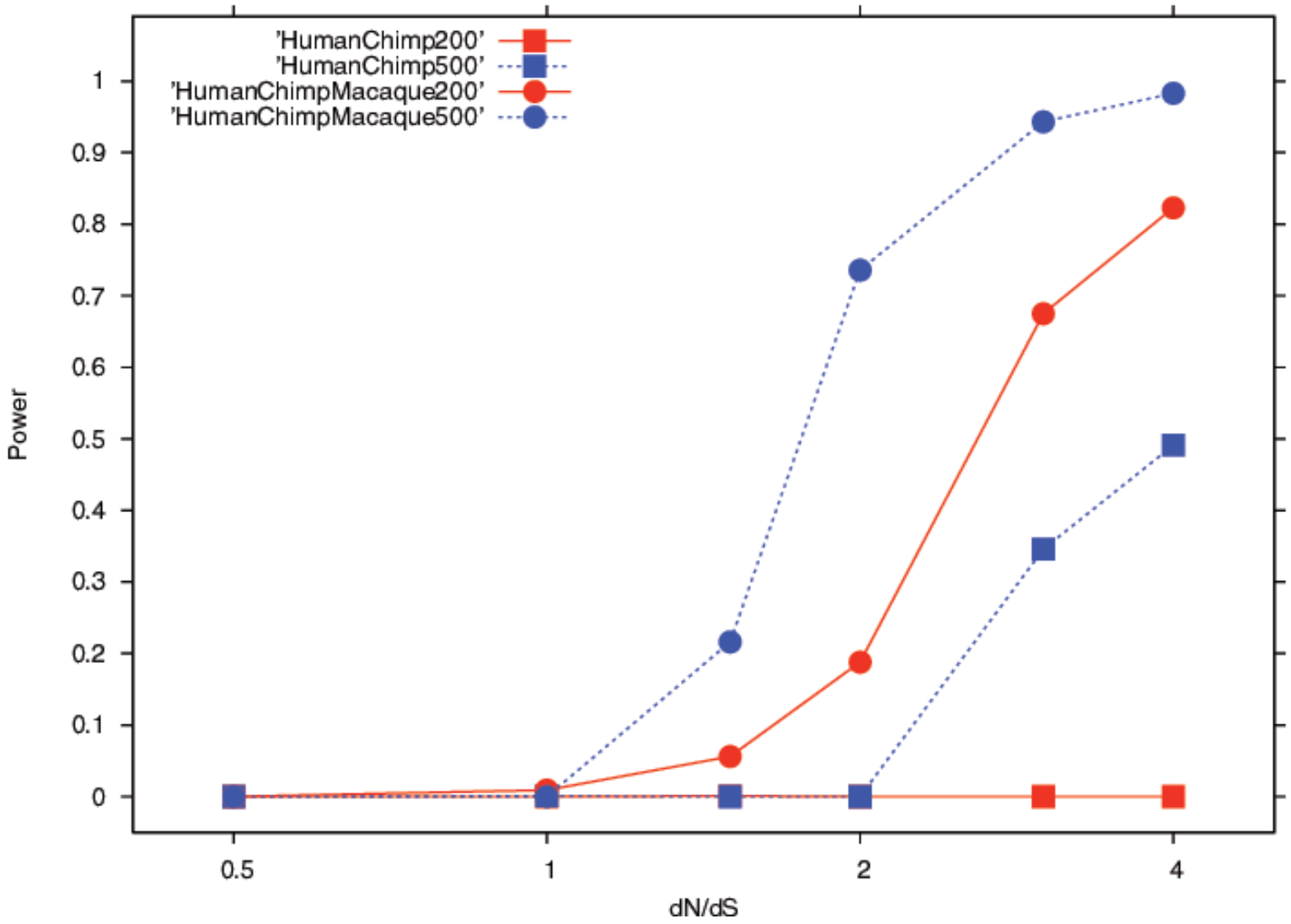
Funkčné kategórie obohatené o gény s pozitívnym výberom

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

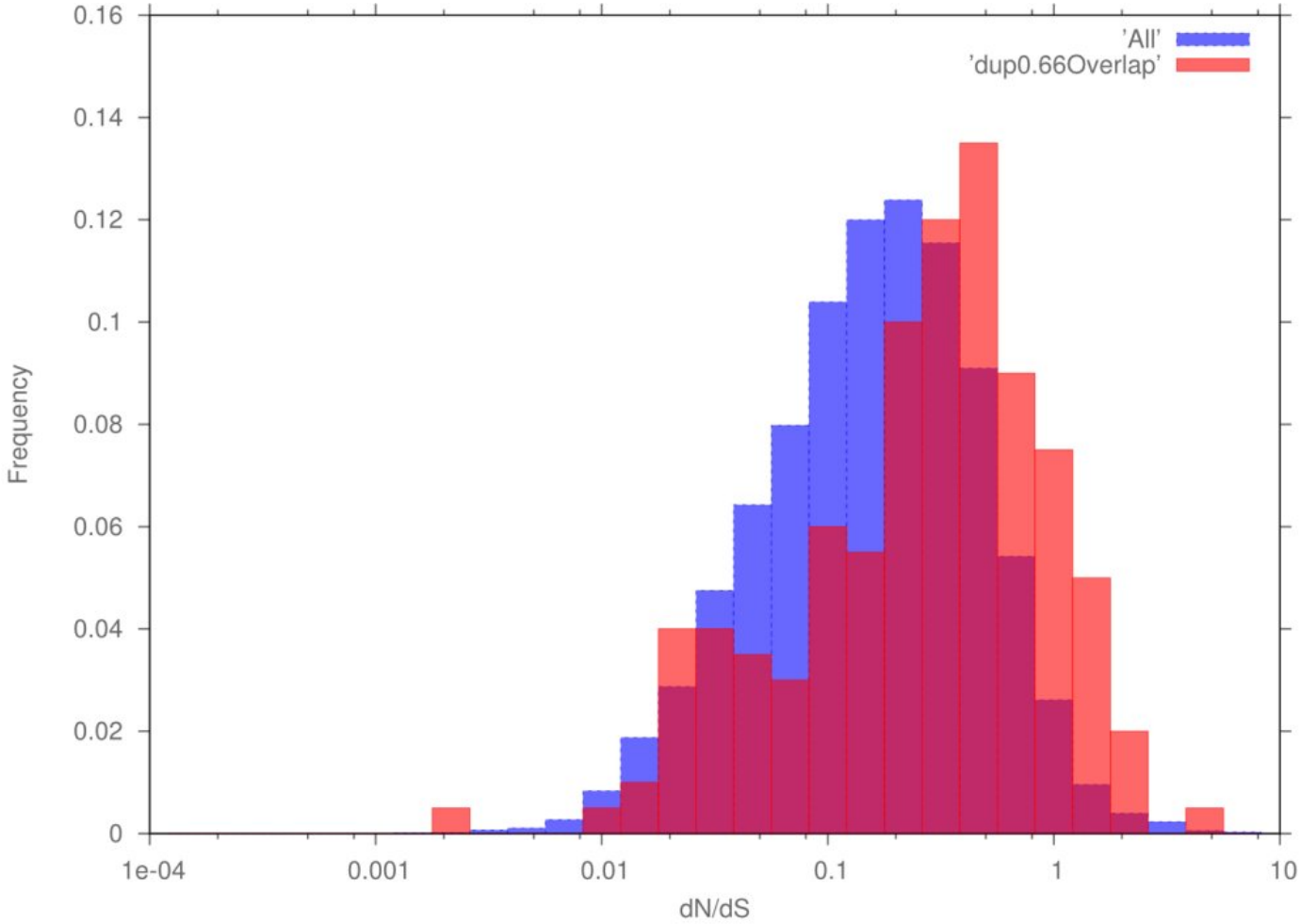
Immunity: adaptive immune response, adaptive immune response somatic recomb, lymphocyte mediated immunity, immunoglobulin mediated immune response, B cell mediated immunity, innate immune response, complement activation alternative pathway, regulation of immune system process, positive regulation of 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

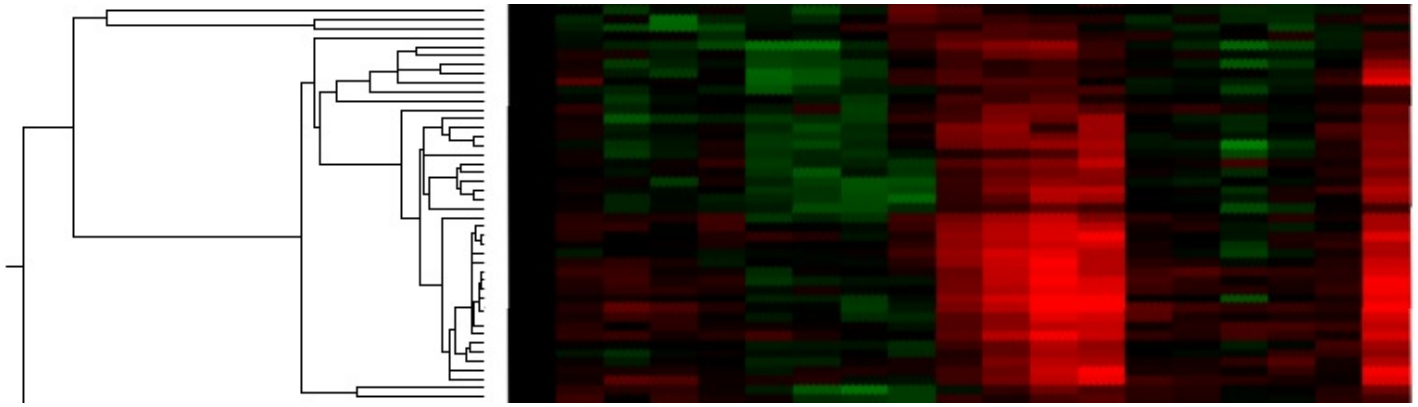
Oznamy

- Body z DÚ 1 budú časom v Moodle
- DÚ 2 je na stránke, odovzdať do 4.12.
- Stretnutia journal clubu sa väčšinou uskutočnili, posledná skupina nezabudnite po stretnutí napísať krátky sumár do Moodle
- Ak máte nejaké otázky k článku, kontaktujte vyučujúcich
- Biológovia nájdu komentáre k návrhu projektu v Moodli, v prípade otázok kontaktujte B. Brejovú

Regulácia génovej expresie

Tomáš Vinař

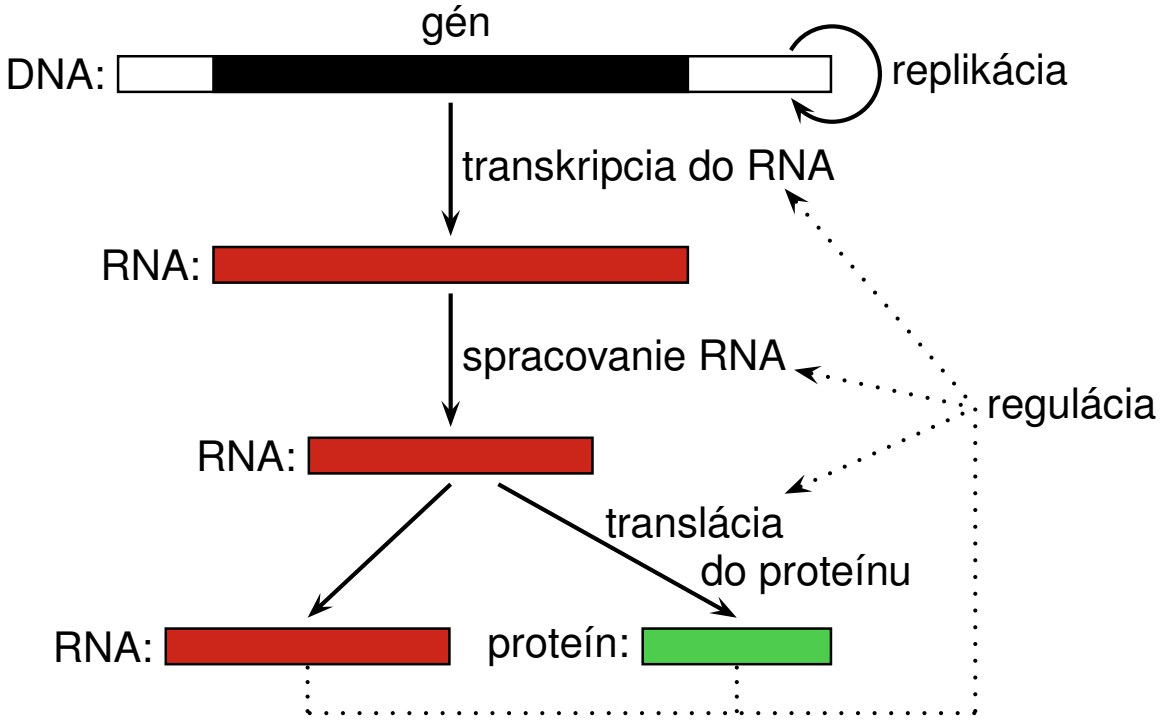
21.11.2024



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

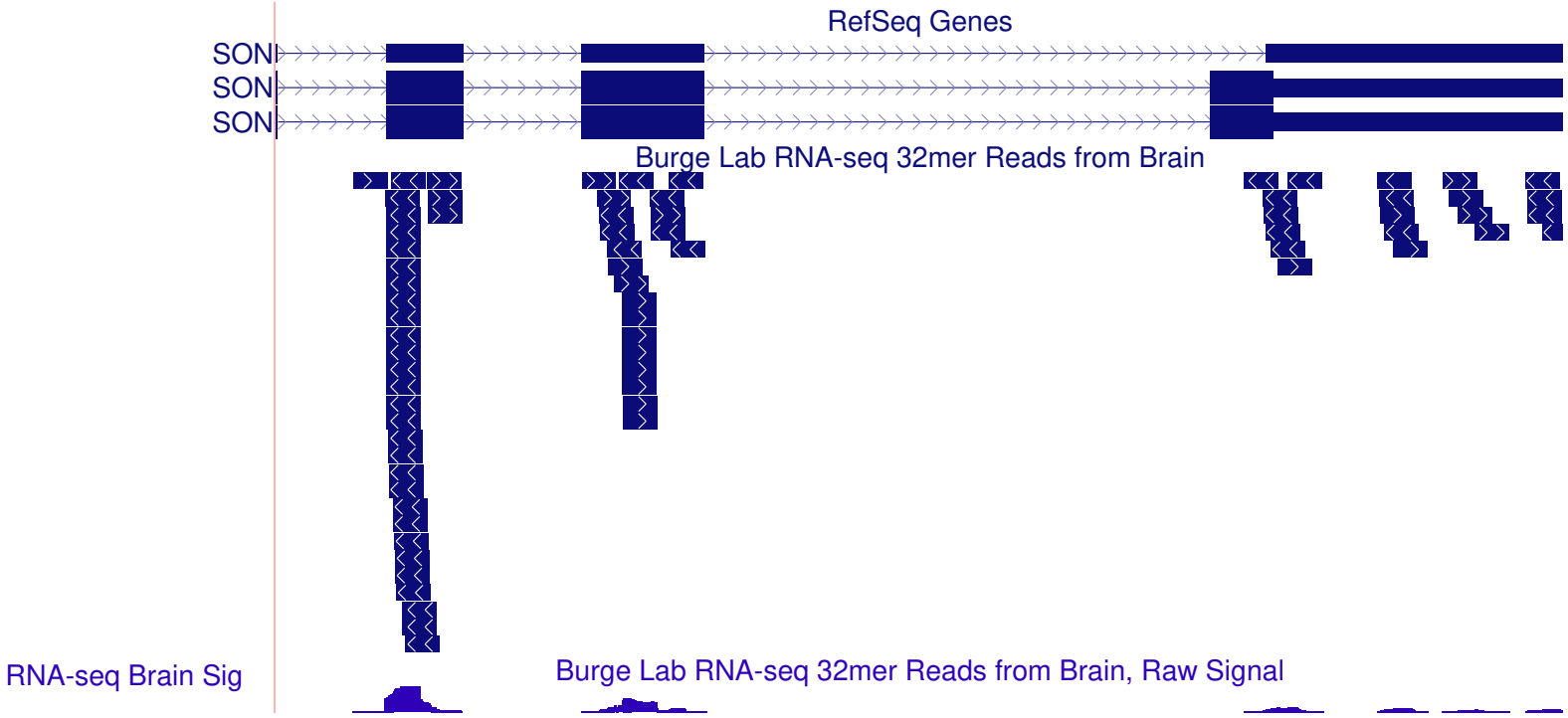
- Zistiť, za akých podmienok je daný gén exprimovaný
  (súvisí s funkciou génu)
- Ktoré gény ho regulujú
- Detaily regulačného mechanizmu
  (väzobné miesta, zmeny v množstve expresie, . . .)

Technológia: RNA-seq

Sekvenujeme RNA extrahovanú z bunky,

mapujeme na genóm, hĺbka pokrytia zodpovedá úrovni expresie,

opakujeme za rôznych podmienok



Používa sa aj staršia technológia microarray (expression array)

Príklad dát o expresii

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”

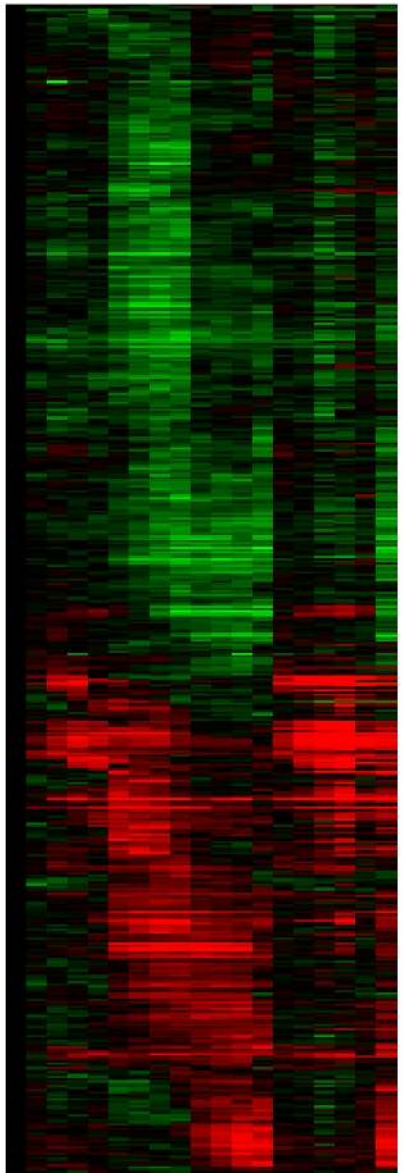
Vizualizácia

Červená: fg$>$bg

Zelená: fg$<$bg

517 génov (z 8600, ktoré boli merané)

19 experimentov



Dnes: iný typ dát

Všetky ostatné prednášky: pracujeme so sekvenciami

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

Dnes: tabuľka čísel

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

Prvá sada problémov: predspracovanie dát

- Zarovnávanie čítaní na genóm
- Počet čítaní alebo fragmentov DNA zarovnaných ku každému génu
- Normalizácia, aby sme mali porovnateľné výsledky z rôznych experimentov
- Normalizácia vzhľadom na dĺžku a iné vlastnosti génu

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)

Cieľ: 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: $x_1, x_2, \ldots, x_n$, priemer $\overline{x}$

Profil génu 2: $y_1, y_2, \ldots, y_n$, priemer $\overline{y}$

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

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

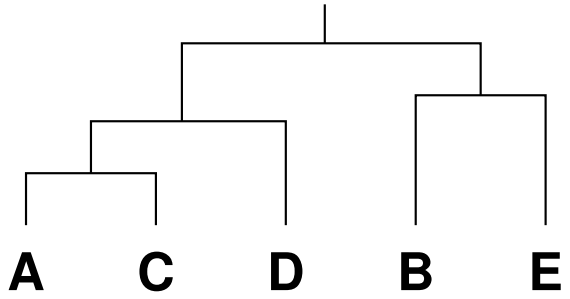
Vzdialenosť $d(x,y) = 1 - C(x,y)$

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

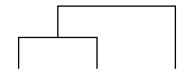


Hierarchické zhlukovanie - príklad

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

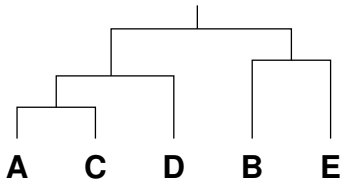
	A	B	C	D	E
gén A	0	0.6	0.1	0.3	0.7
gén B	0.6	0	0.5	0.5	0.4
gén C	0.1	0.5	0	0.6	0.6
gén D	0.3	0.5	0.6	0	0.8
gén E	0.7	0.4	0.6	0.8	0

	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

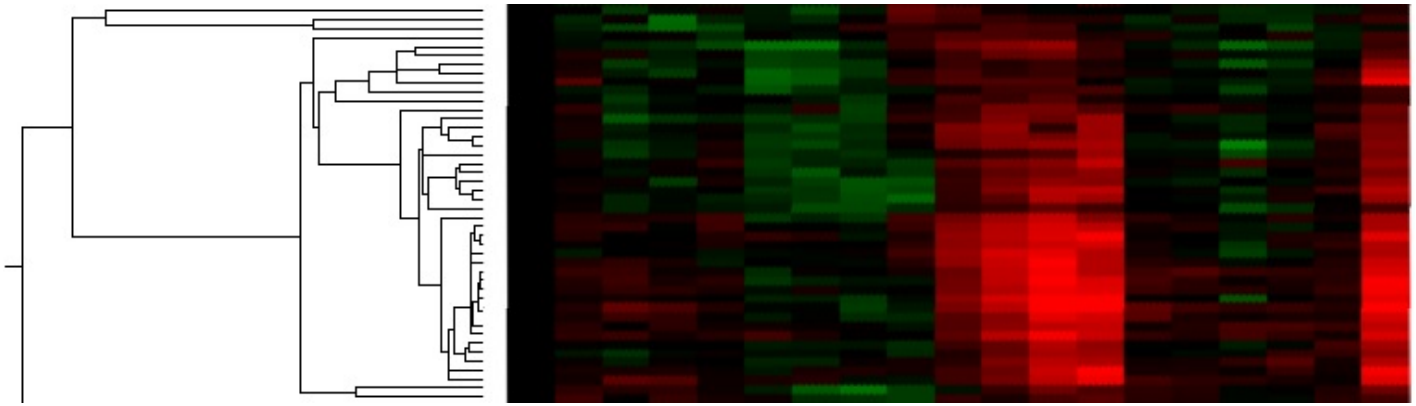


	A+C+D	B	E
A+C+D	0	0.5	0.6
B	0.5	0	0.4
E	0.6	0.4	0

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

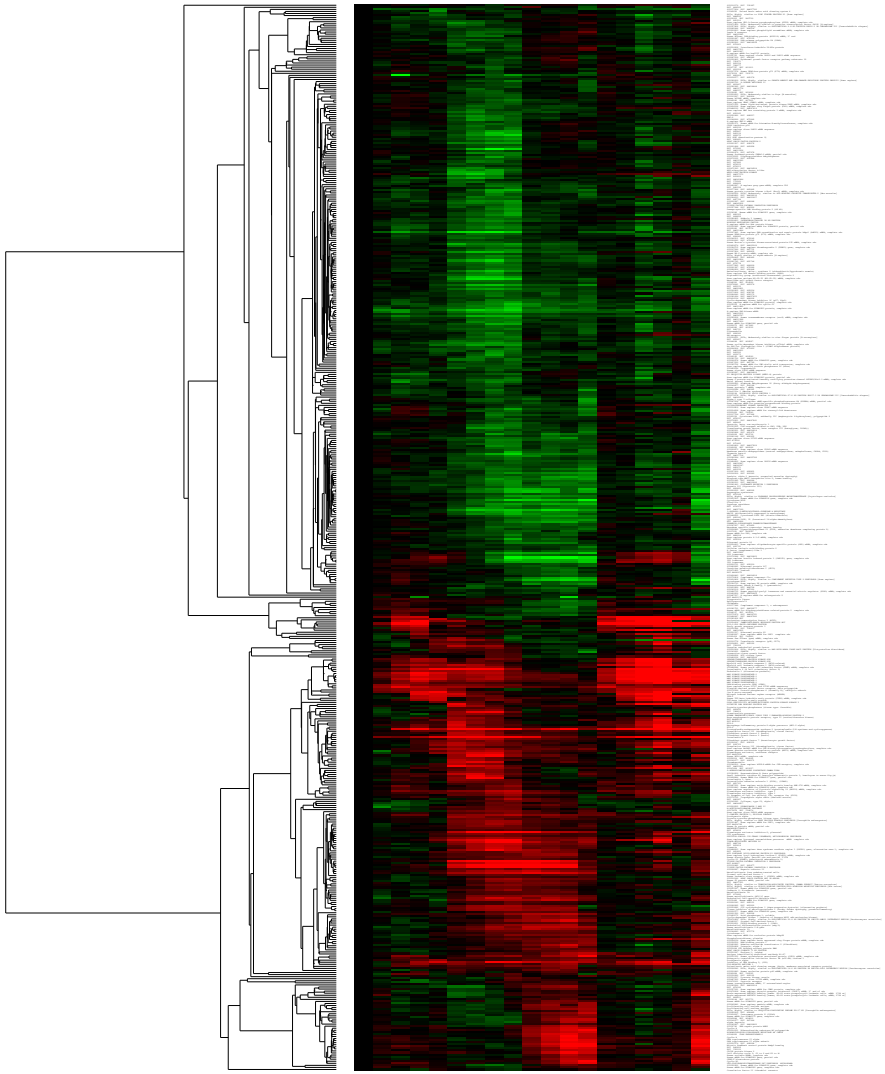


Príklad: časť dát o expresii



Zhlukovanie tiež pomáha vizualizácii dát,

podobné gény sa dostanú ku sebe

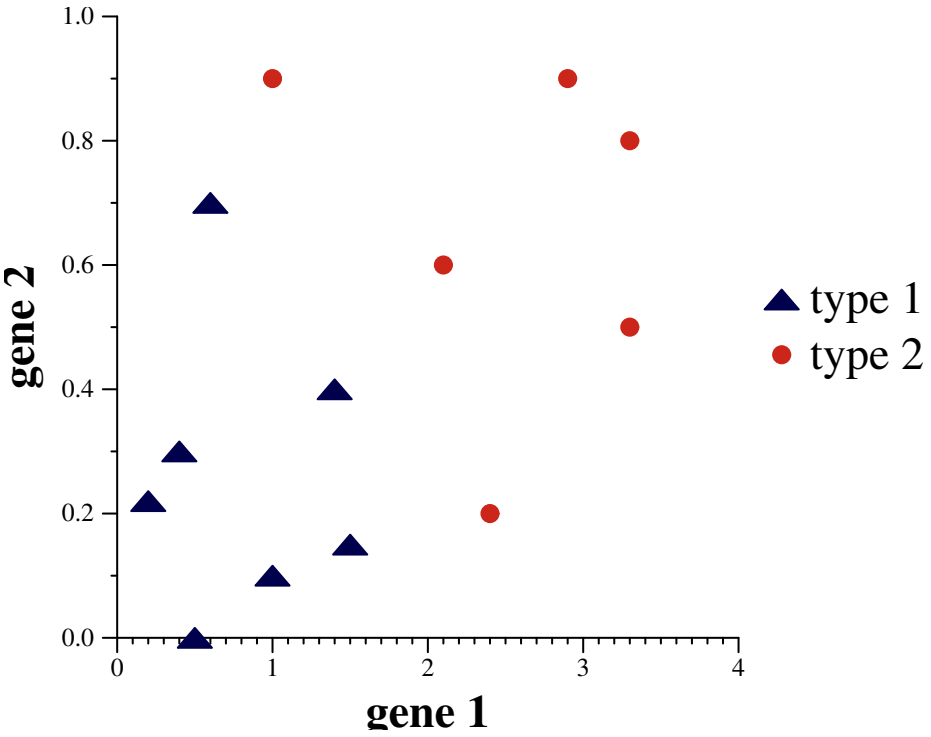


Klasifikácia

- 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 trénovacích dátach hľadáme hodnoty parametrov, pre ktoré vzorec najlepšie funguje
- Fungovanie vzorca testujeme na testovacích dátach (nepoužité na trénovanie)
- Hotový vzorec použijeme na dáta s neznámym typom

Jednoduchý príklad: expresia 2 génov

Trénovacie dáta so známym typom:



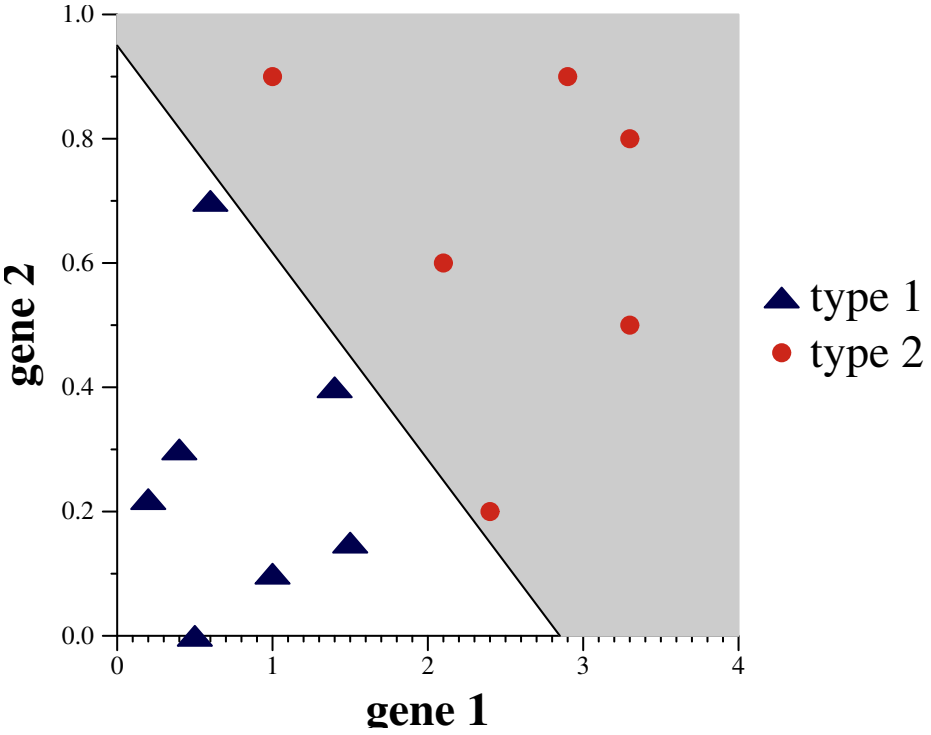
Typ vzorca: lineárne funkcie (lineárny diskriminant)

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

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

Jednoduchý príklad: expresia 2 génov

Výsledný vzorec:



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

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

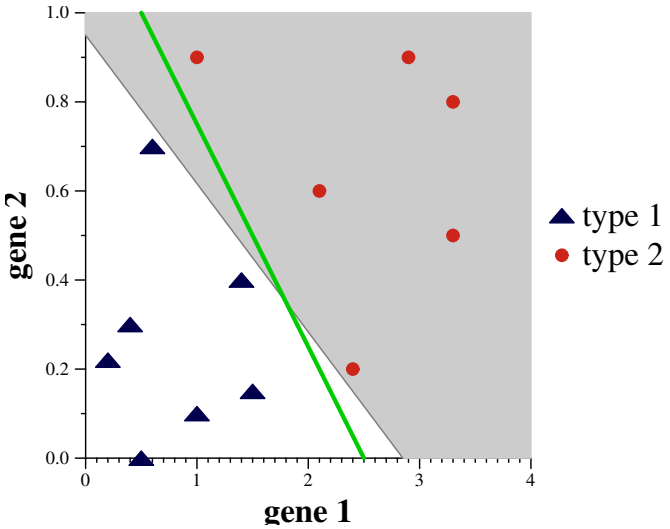
Populárne techniky na klasifikáciu

Logistic regression, logistická regresia:

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

Support vector machines

(SVM): hľadanie lineárneho diskriminátora s nulovou trénovacou chybou, ktorý je najďalej od všetkých trénovacích 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

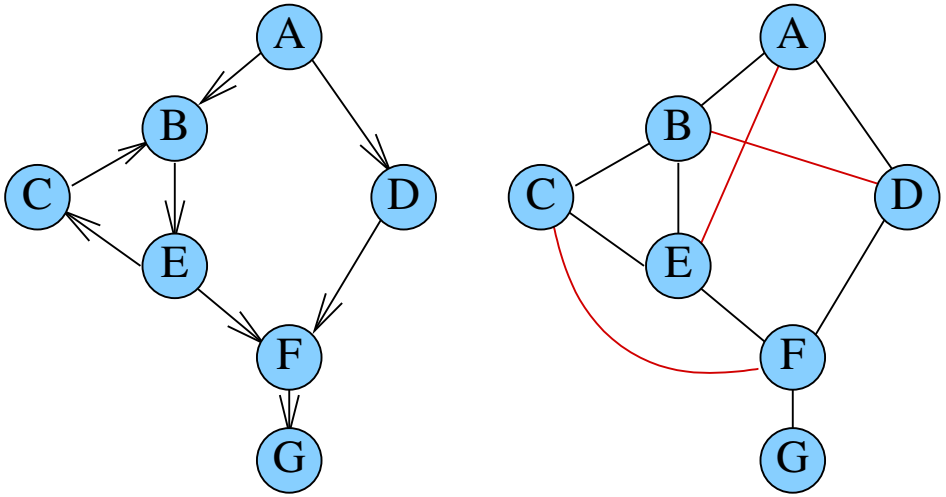
podobne ako stav v HMM

Regulačné siete z profilov expresie

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

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

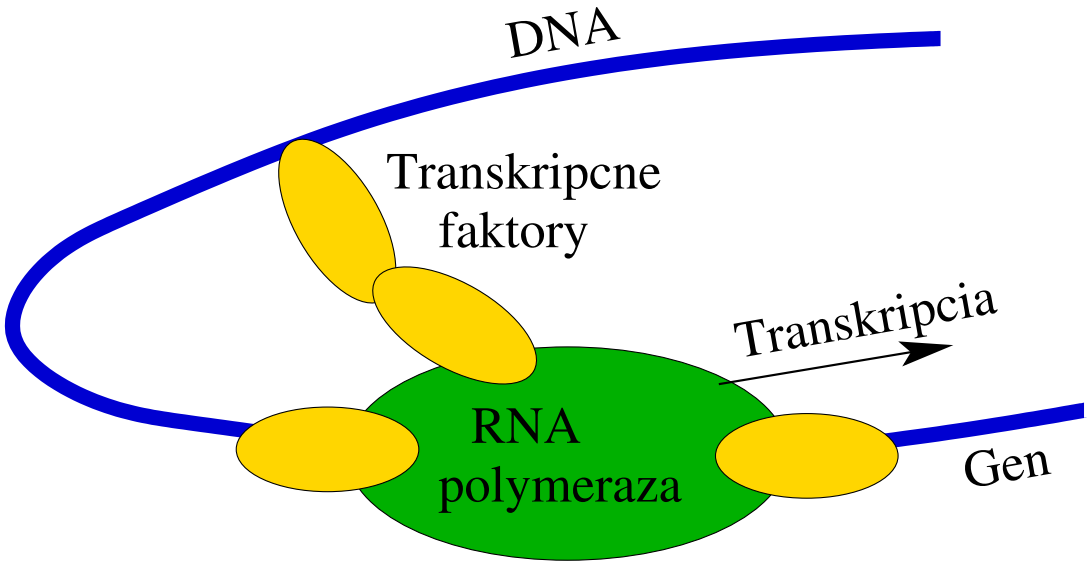
Podobnosť profilov expresie nám môže dať neorientované hrany. Chceme vylúčiť hrany, ktoré vznikli tranzitivitou a správne orientovať hrany (ťažký problém)



Transkripčné faktory (TF)

Regulácia začatia transkripcie pomocou transkripčných faktorov:

proteíny viažúce DNA, pomáhajú pritiahnuť RNA polymerázu

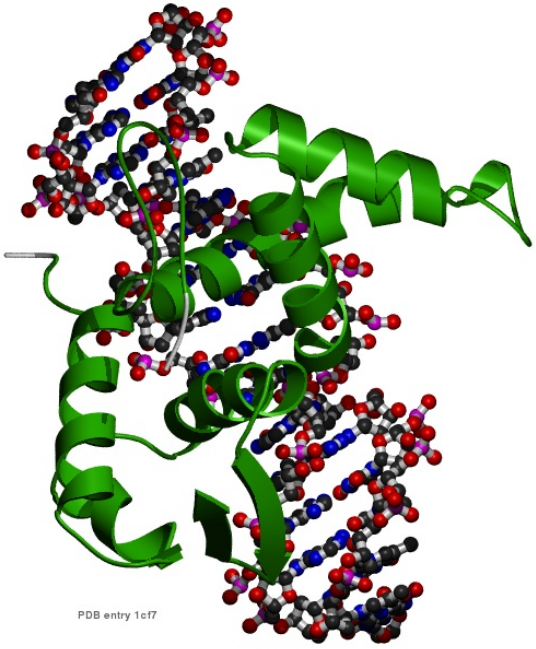


Človek má vyše 2000 TF-ov

Môžu zvyšovať alebo znižovať mieru expresie,

fungovať v skupinách

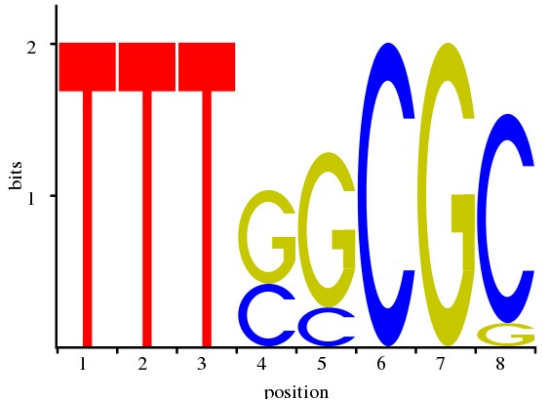
Príklad: transkripčný faktor E2F1



- Reguluje bunkový cyklus
- Viaže TTTCCCGC alebo TTTCGCGC,
  prípadne ďalšie varianty

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

- 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

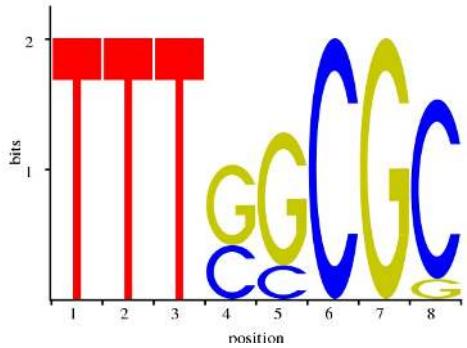
Príklad: motív TTTGGCGC + 1 nezhoda

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

TTT**CC**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

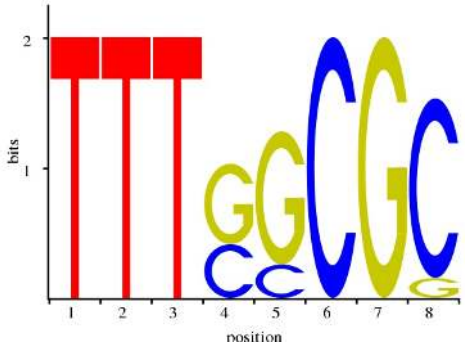
Príklad: motív TTT[CG][CG]CGC

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

TT**A**GGCGC nie je výskyt

Zostavenie motívu: povoľ 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

TTTGGCG**G** 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 motívu v genóme

- Zoberieme motív v niektorej reprezentácii:
  - Konsenzus, napr. TTTGGCGC + 1 nezhoda
  - Regulárny výraz, napr. TTT[CG][CG]CGC
  - Skórovacia matica, napr. prah $T = 8$ a matica:

```
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
```

- Pre každú pozíciu v genóme testujeme, či je výskytom motívu
- Výskyty sú potenciálne väzobné miesta

Hľadanie výskytov motívu v genóme – problém

- Hľadanie motívu v genóme: skús každú pozíciu, či je výskytom
- Okrem **väzobných miest** často aj veľa **náhodných 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 špecifickosti hľadáme
  - zhluky väzobných miest,
  - miesta podporené experimentálne,
  - evolučne zachované
- Databázy motívov, napr. TRANSFAC, JASPAR

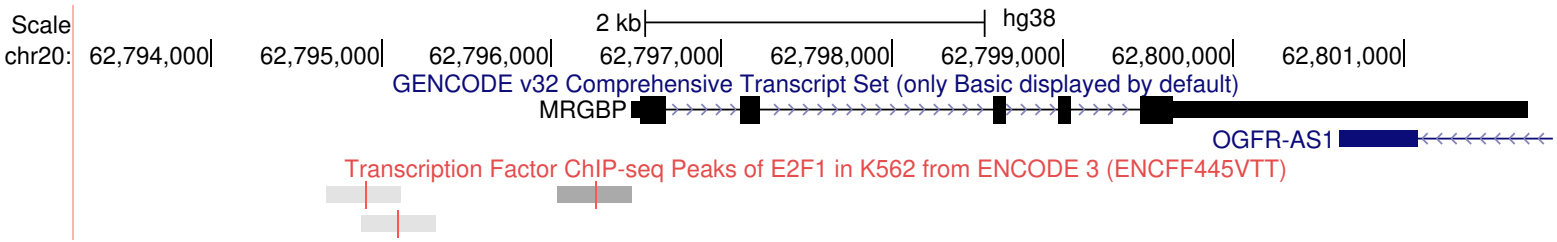
Ako nájsť väzobné miesta experimentálne?

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 (**ChIP-seq**)

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.
- **V súčasnosti:** zoberieme oblasti detegované pomocou ChIP-seq okolo väzobných miest, nájdený motív použijeme na presnejšie určenie polohy väzby TF
- **Pôvodne:** zoberieme skupinu génov s podobným profilom expresie a teda možno regulovaných tým istým TF, hľadáme motív v oblastiach pred týmito génmi

Príklad: Consensus Pattern Problem (CPP)

Jednoduchá formulácia problému hľadania motívov

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

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 $S_1, \ldots, S_k$

  Problém: Nemusí fungovať — prečo?
  Ale dá sa dokázať, že cena riešenia
  bude najviac dvojnásobok optima
  (2-aproximačný algoritmus)

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

Príklad:

Vstup: $L = 4$

CAAACAT,

AGTAGC,

TAACCA,

TCTCCTC

Výstup:

motív TAAC

výskyty a nezhody

AAAC 1,

TAGC 1,

TAAC 0,

TCTC 2

spolu 4 nezhody

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 $\Pr(S \mid W)$.

Hľadanie motívov cez pravdepodobnostné modely

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

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

- Ťažký problém, používajú sa heuristické algoritmy
- Napríklad EM (expectation maximalization)
- 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. Priraď 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 $S_1, \ldots, S_k$ váhovaných podľa $p_{i,j}$

Iterácie zvyšujú vierohodnosť dát, kým nedôjde ku konvergencii.

Skúšame veľakrát z rôznych počiatočných W

Príklad algoritmu EM

A	0.10	0.10	0.10	0.10	0.10
C	0.10	0.10	0.10	0.70	0.70
G	0.10	0.10	0.10	0.10	0.10
T	0.70	0.70	0.70	0.10	0.10

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

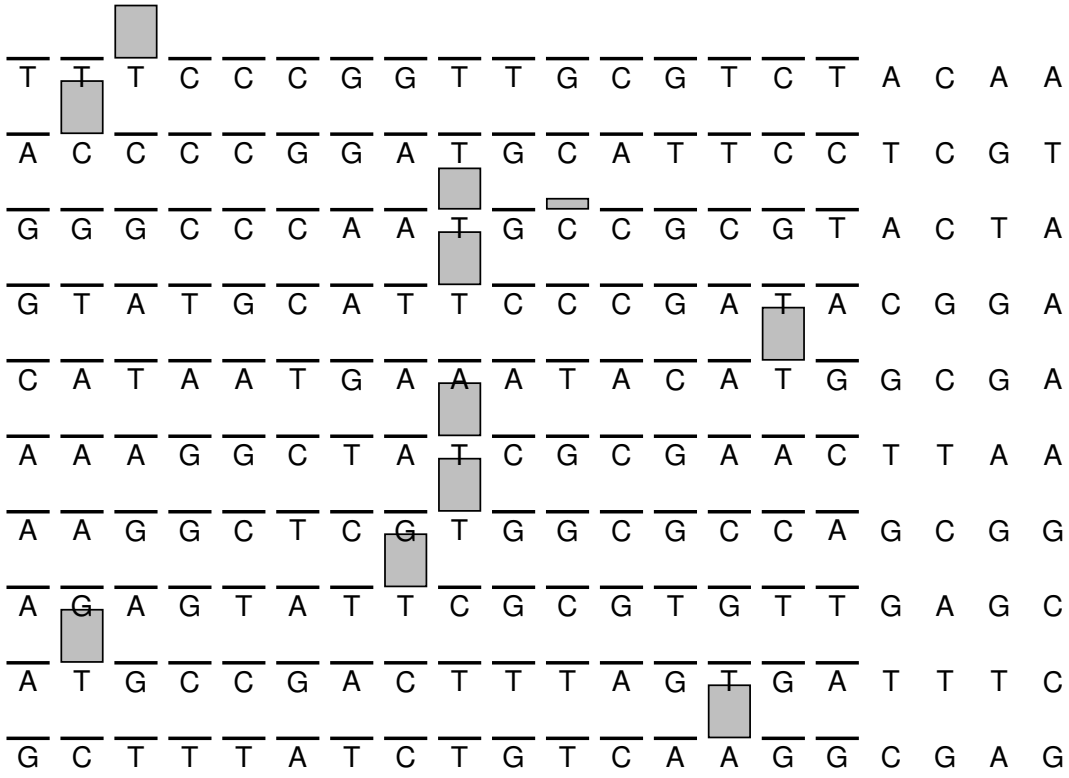
Príklad algoritmu EM: ďalšia iterácia

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

A	0.47	0.09	0.01	0.02	0.03
C	0.02	0.11	0.20	0.80	0.58
G	0.08	0.22	0.48	0.15	0.35
T	0.42	0.58	0.30	0.03	0.03

Príklad algoritmu EM: po 20 iteráciách

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



Zhrnutie

- RNA-seq meră úroveň expresie pre všetky gény 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,
  okrem väzobných miest môžu mať aj ďalšie náhodné výskyty
- EM algoritmus na hľadanie nových motívov v sekvenciách

Oznamy

- DÚ 2 je na stránke, odovzdať do 4.12.
- Termíny na konci semestra
  – DÚ3 streda 18.12., správy zo journal clubu piatok 20.12.

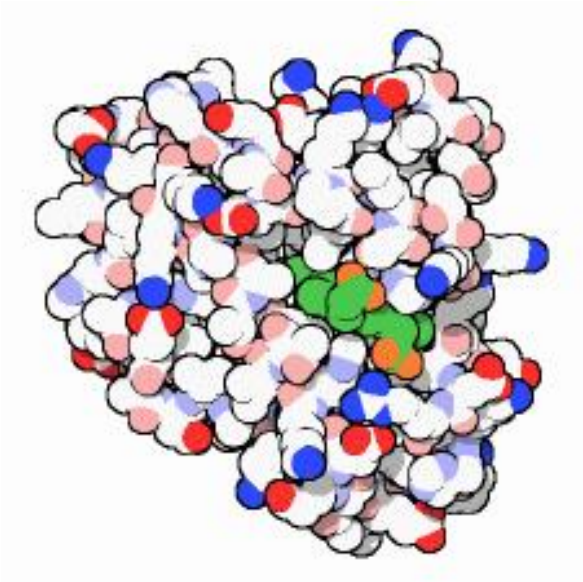
Správa zo journal clubu

- Pochopiteľná pre študentov tohto predmetu (inf aj bio)
- Vysvetlite pojmy, ktoré sú nad rámec tohto predmetu
- Netreba pokryť všetko, môžete využiť aj iné zdroje
- Podrobne vysvetliť aspoň jednu bioinformatickú metódu a aspoň jeden biologický výsledok (alebo overovanie správnosti metódy na dátach)
- Ako článok súvisí s učivom preberaným na predmete
- Nájdite zopár citujúcich prác, ktoré výsledky využili alebo vylepšili
- Rozsah cca 1-2 strany na osobu, jeden ucelený text
- Píšte vlastnými slovami, citujte zdroje
- V správe vymenujte členov skupiny, ktorí sa podieľali na jej spísaní, dostanú rovnako bodov
- Pdf odovzdať cez Moodle (stačí 1 za skupinu)

Štruktúra a funkcia proteínov

Broňa Brejová

27.11.2024



Proteíny

Reťazce 20 rôznych aminokyselín s rôznymi chemickými vlastnosťami:

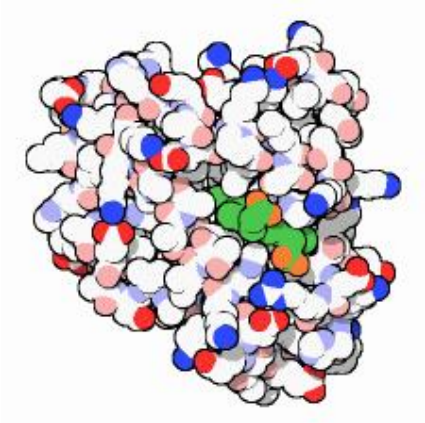
Aminokyselina	Postranný reťazec	Jeho vlastnosti
Alanín (A)	-CH3	hydrofóbny
Arginín (R)	-(CH2)3NH-C(NH)NH2	bázický
Asparagín (N)	-CH2CONH2	hydrofilný
Kyselina asparágová (D)	-CH2COOH	kyslý
Cysteín (C)	-CH2SH	hydrofóbny
Kyselina glutámová (E)	-CH2CH2COOH	kyslý
Glutamín (Q)	-CH2CH2CONH2	hydrofilný
Glycín (G)	-H	hydrofilný
Histidín (H)	-CH2-C3H3N2	bázický
Izoleucín (I)	-CH(CH3)CH2CH3	hydrofóbny
Leucín (L)	-CH2CH(CH3)2	hydrofóbny
Lyzín (K)	-(CH2)4NH2	bázický
Metionín (M)	-CH2CH2SCH3	hydrofóbny
Fenylalanín (F)	-CH2C6H5	hydrofóbny
Prolín (P)	-CH2CH2CH2-	hydrofóbny
Serín (S)	-CH2OH	hydrofilný
Treonín (T)	-CH(OH)CH3	hydrofilný
Tryptofán (W)	-CH2C8H6N	hydrofóbny
Tyrozín (Y)	-CH2-C6H4OH	hydrofóbny
Valín (V)	-CH(CH3)2	hydrofóbny

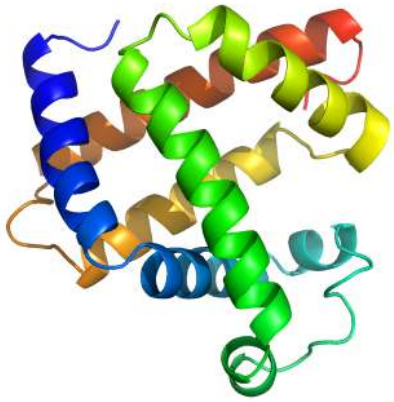
Štruktúra proteínov

- **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 crystalography)
  vyžaduje proteín v kryštalickej forme
- NMR (nuclear magnetic resonance spectroscopy)
  hlavne používaná na kratšie proteíny
- Cryo-EM (cryogenic electron microscopy)
  vhodná na veľké proteínové komplexy, rastúca popularita
- Náročný a drahý proces
- Databáza štruktúr PDB
  228 000 proteínových štruktúr (83% X-ray)
  (UniProt má 250 miliónov sekvencií)

Určovanie štruktúry proteínov ako bioinformatický problém

(protein structure prediction, protein folding)

Vstup: sekvencia proteínu X

Výstup: 3D pozície atómov alebo aminokyselín

(1) 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
- Veľmi ťažký výpočtový problém
  — simulácia molekulárnej dynamiky
  — optimalizačné metódy, napr. gradientová metóda, simulované žíhanie
- Používané na malé proteíny a zlepšenie približných štruktúr

Určovanie štruktúry proteínov ako bioinformatický problém

(protein structure prediction, protein folding)

Vstup: sekvencia proteínu X

Výstup: 3D pozície atómov alebo aminokyselín

(1) Ab initio metódy

(2) Metódy založené na homológii

Hľadáme homológy proteínu X, t.j. podobné proteíny

Štruktúra sa väčšinou evolučne mení pomalšie ako sekvencia

ak niektorý homológ má známu štruktúru, aj X má asi podobnú

Určovanie štruktúry proteínov bolo dlho považované za otvorený problém,

ktorý nevieme bioinformaticky riešiť, ak sa nedá použiť metóda (2)

Určovanie štruktúry proteínov ako bioinformatický problém

(protein structure prediction, protein folding)

Vstup: sekvencia proteínu X

Výstup: 3D pozície atómov alebo aminokyselín

(1) Ab initio metódy

(2) Metódy založené na homológii

(3) Metódy založené na hlbokých neurónových sieťach

Od roku 2018 veľký pokrok

Hlavne program AlphaFold od firmy DeepMind/Google

Nobelova cena za chémiu 2024: dvaja z autorov, Demis Hassabis a John Jumper

(a David Baker za návrh nových proteínov)

Najnovšie prístupy: hlboké neurónové siete

- Súťaž CASP raz za dva roky
- V roku 2018 a 2020 vyhral AlphaFold od firmy DeepMind/Google.
  V roku 2020 AlphaFold2 vyhral s veľkým náskokom.
  2/3 predpovedaných štruktúr mali vysokú presnosť.
  Využíva nové prvky, aj existujúce prístupy.
  V roku 2022 väčšina metód inšpirovaná AlphaFold2.
- Kľúčová myšlienka využitá aj pred AlphaFold-om: **detekcia ko-evolúcie**
  - k skladanému proteínu zarovnaj veľké množstvo homológov
    (aj bez známych štruktúr)
  - hľadaj dvojice pozícií, ktoré sa menia súčasne
  - takéto dvojice sú potenciálne v kontakte

Najnovšie prístupy: hlboké neurónové siete

- **AlphaFold 1 (2018):**

  (1) Predikcia vzdialeností amino kyselín pomocou neurónovej siete

  (2) Hľadanie štruktúry, ktorá dobre sedí so vzdialenosťami
  a fyzikálnym modelom využitím štandardnej numerickej optimalizácie
  (gradientové metódy) [animácia]

- **AlphaFold 2 (2020):**

  kombinuje oba kroky do jednej neurónovej siete,
  ktorá sa opakovane spúšťa na svojich výsledkoch

- **AlphaFold 3 (2024):**

  Iná neurónová sieť,
  umožnuje skladať aj komplexy kombinujúce viac proteínov,
  alebo proteín a inú molekulu (DNA, RNA, ióny, a pod.)

Limitácie programu AlphaFold

Vyplývajú z dostupných dát pre trénovanie

- Nedá sa využiť na proteíny bez homológov (napr. umelo vytvorené alebo tie, ktoré rýchlo mutujú, napr. protilátky)
- Nie je úplne presný v predpovedaní vplyvu mutácie na štruktúru
- Predpovedá jednu štruktúru, ale veľa proteínov má viacero možných polôh
- Flexibilnejšie časti proteínov (disordered) sú často predpovedané s nízkou spoľahlivosťou (vyznačenou vo výsledkoch ako low confidence)
- AlphaFold3 nevie spracovať všetky typy molekúl viažúcich proteíny

Praktické prístupy k určovaniu štruktúry proteínu

Pre daný proteín X:

- Pozrieme do PDB, či má X známu štruktúru
- V databázach môžeme nájsť aj štruktúru pre X od AlphaFold
- Môžeme spustiť AlphaFold na X
- Môžeme hľadať homológy X so známou štruktúrou

Hľadanie homológov proteínu

Dôležité pre rôzne účely:

– určenie približnej štruktúry a funkcie proteínu

– štúdium evolúcie proteínu

– vstup pre AlphaFold

Videli sme:

– dynamické programovanie

– heuristické zrýchlenia (BLAST a spol.)

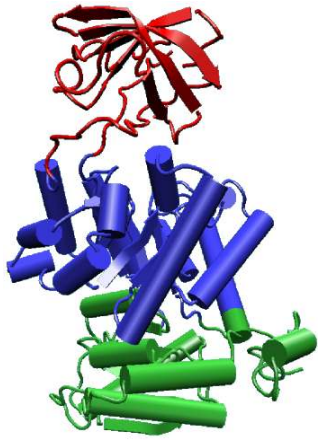
– skórovacie matice (BLOSUM)

Nevedia nájsť vzdialenejšie homológy

Dnes si ukážeme prístupy založené na **pravdepodobnostných profiloch**

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

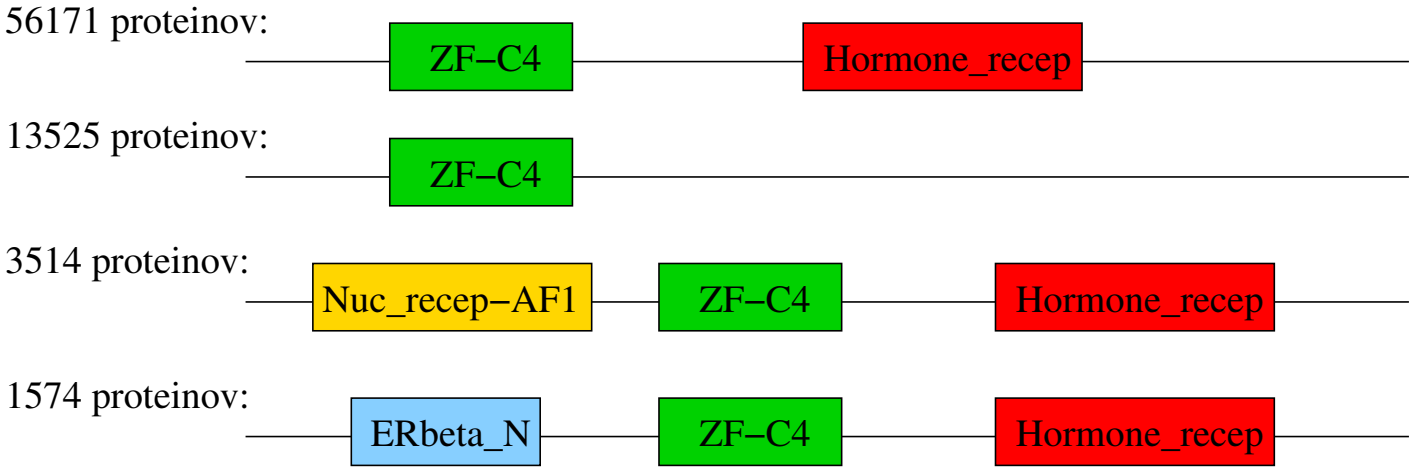
Domény v proteínoch rozdelené do viac ako 20 tisíc rodín

76% proteínov aspoň jedna známa doména

49% proteínových sekvencií pokrývajú známe domény

Príklad:

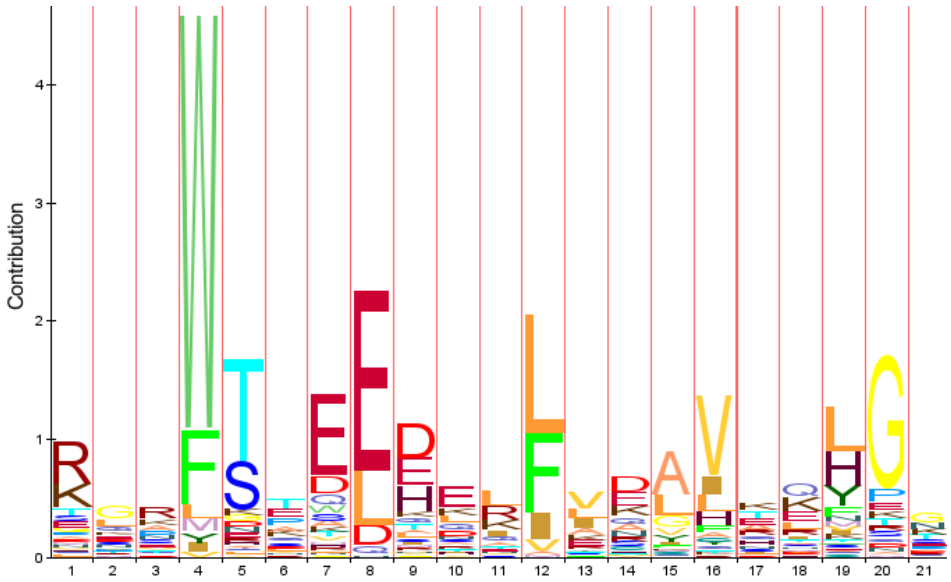
4 z 654 architektúr obsahujúcich domény Zinc finger, C4 type (Pfam)



Charakterizácia rodín proteínov

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

```
MEEWSASEANLFEEALEKYGKDF
PDEWTVEDKVLFEQAFSFHGKT.
GTKWTAEENKKFENALAFYDKDT
SKNWSEDDLQLLIKAVNLFPAGT
EKPWSNQETLLLLEAIETYGDD.
ARFWTDQETLLLLEGLEMHKDD.
KPEWSDKEILLLLEAVMHYGDD.
DDTWTAQELVLLSEGVEMYS...
KKNWSDQEMLLLLEGIEMYE...
DENWSKEDLQKLLKGIQEFGAD.
EDDWSQAEQKAFETALQKYPKGT
EEAWTQSQQKLLELALQQYPKGA
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 $x_1, x_2, \ldots, x_n$
  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 sekvencie $x_1, \ldots, x_n$:
  logaritmus pomeru pravdepodobností v dvoch modeloch

$$\log \frac{e_1(x_1) \ldots e_n(x_n)}{q(x_1) \ldots q(x_n)}$$

(neskôr rozpíšeme na súčet dielčích skóre pre aminokyseliny)

Hračkársky príklad PSSM

- Uvažujme len leucín L a alanín A
- Majme zarovananie 10 sekvencií s počtami / frekvenciami $e_i(x)$ v tabuľke

	počty				frekvencie			
	1	2	3	4	1	2	3	4
A	2	6	9	1	0,2	0,6	0,9	0,1
L	8	4	1	9	0,8	0,4	0,1	0,9

- Nulová hypotéza $q(A) = 0{,}3$, $q(L) = 0{,}7$
- Pravdepodobnosť sekvencie LAAL
  – v profile $0{,}8 \cdot 0{,}6 \cdot 0{,}9 \cdot 0{,}9 = 0{,}3888$,
  – v nulovom modeli $0{,}7 \cdot 0{,}3 \cdot 0{,}3 \cdot 0{,}7 = 0{,}0441$
- Skóre LAAL: $\log_2(0{,}3888/0{,}0441) = 3{,}14$
  Skóre LALA: $\log_2(0{,}0048/0{,}0441) = -3{,}20$

Pravdepodobnostný profil rodiny

- $e_i(x)$: frekvencia výskytu písmena x v stĺpci i zarovnania rodiny
- $q(x)$: frekvencia výskytu písmena x v nulovom modeli
- $s_i(x) = \log \frac{e_i(x_i)}{q(x_i)}$ skóre písmena x v stĺpci i zarovnania rodiny
- Skóre sekvencie $x_1, \ldots, x_n$:

  logaritmus pomeru pravdepodobností v dvoch modeloch

$$\log \frac{e_1(x_1) \cdot \ldots \cdot e_n(x_n)}{q(x_1) \cdot \ldots \cdot q(x_n)}$$
$$= \log \left(\frac{e_1(x_1)}{q(x_1)} \cdot \ldots \cdot \frac{e_n(x_n)}{q(x_n)} \right)$$
$$= \log \frac{e_1(x_1)}{q(x_1)} + \ldots + \log \frac{e_n(x_n)}{q(x_n)}$$
$$= s_1(x_1) + \ldots + s_n(x_n)$$

Hračkársky príklad PSSM

- Majme zarovananie 10 sekvencií s počtami / frekvenciami $e_i(x)$ v tabuľke

	počty				frekvencie			
	1	2	3	4	1	2	3	4
A	2	6	9	1	0,2	0,6	0,9	0,1
L	8	4	1	9	0,8	0,4	0,1	0,9

- Nulová hypotéza $q(A) = 0{,}3$, $q(L) = 0{,}7$
- 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 LALA je $0{,}19 + 1 - 2{,}81 - 1{,}58 = -3{,}2$

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

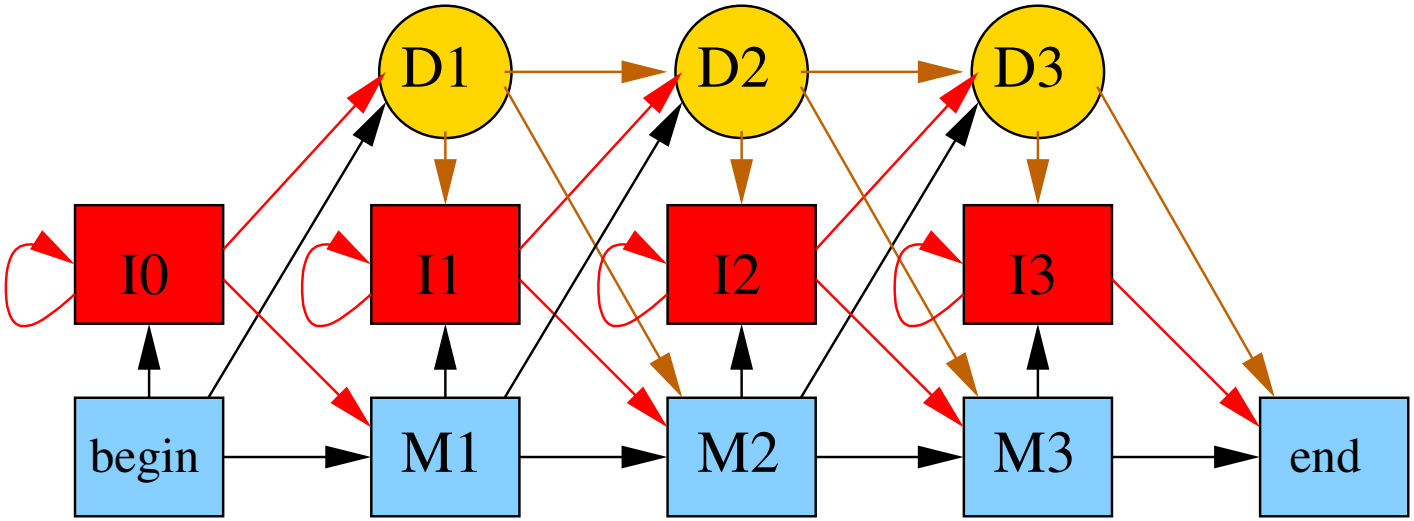
Potom postupujeme ako predtým

Profilové HMM

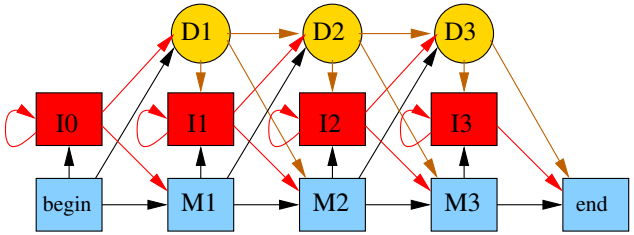
Rozšírime profil o inzercie a delécie

PSSM profil ako HMM:

Profilové HMM: match state, insert state, delete state



Konštrukcia profilového HMM



- Začneme z viacnásobného zarovnania
- 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?

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

Nájdi výskyty profilu v proteínovej sekvencii

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

Výsledné skóre alebo pravdepodobnosť sa použije na rozhodnutie, či proteín patrí do rodiny

Štruktúra proteínov, zhrnutie

(protein structure prediction, protein folding)

Vstup: sekvencia proteínu X

Výstup: 3D pozície atómov alebo aminokyselín

(1) Ab initio metódy

(2) Metódy založené na homológii

(3) Metódy založené na hlbokých neurónových sieťach

Praktické prístupy k určovaniu štruktúry proteínu X

- Pozrieme do PDB, či má X známu štruktúru
- V databázach môžeme nájsť aj štruktúru pre X od AlphaFold
- Môžeme spustiť AlphaFold na X
- Môžeme hľadať homológy X so známou štruktúrou
  resp. domény v X pomocou profilov

Využitie proteínových štruktúr

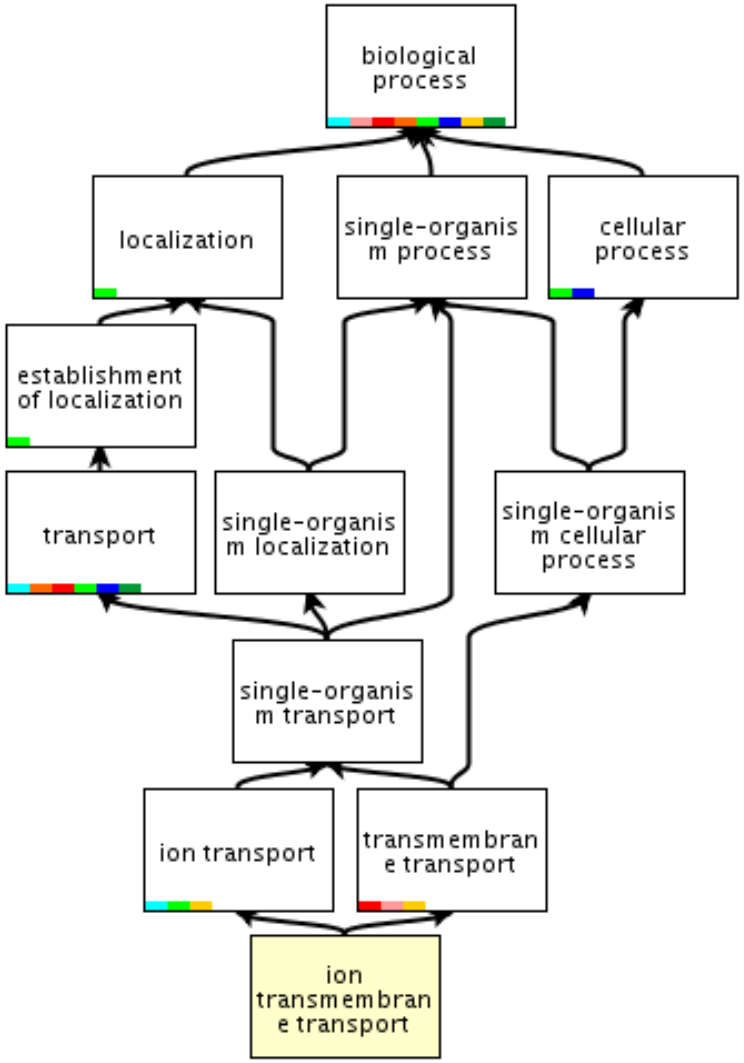
- Presnejšie definovanie domén v databázach ako Pfam
- Skúmanie efektu mutácií na štruktúru / funkciu
- Modelovanie interakcií medzi proteínmi, proteínových komplexov
- Objavovanie nových liečiv, ktoré sa budú viazať na určitý proteín
- Dizajn umelých proteínov s vhodnými vlastnosťami

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:



Oznamy

- DÚ3 je zverejnená, odovzdvajte do 18.12. 22:00
- Budúci štvrtok 12.12.:
  – prednáška aj cvičenia bežia normálne
- Štvrtok 19.12.:
  – nepovinné prezentácie journal clubu v čase prednášky
  – cvičenia pre informatikov budú
  – biológovia predbežné prezentácie projektov, čas dohodneme
- Termíny na konci semestra
  – DÚ3 streda 18.12., správy zo journal clubu piatok 20.12.
- Budúci štvrtok dohodneme:
  – či chcete prezentovať journal club (dohodnite sa v skupinách)
- Termín skúšky pre informatikov?

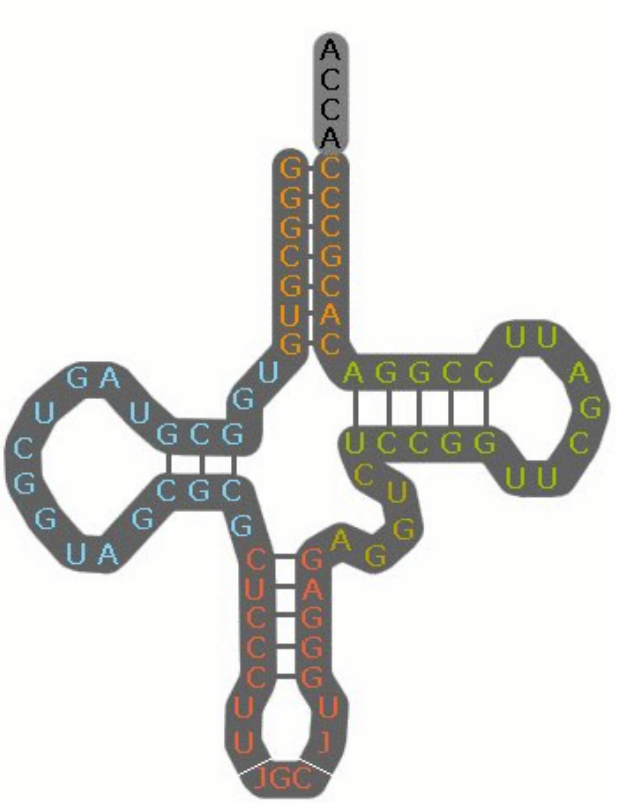
Správa zo journal clubu

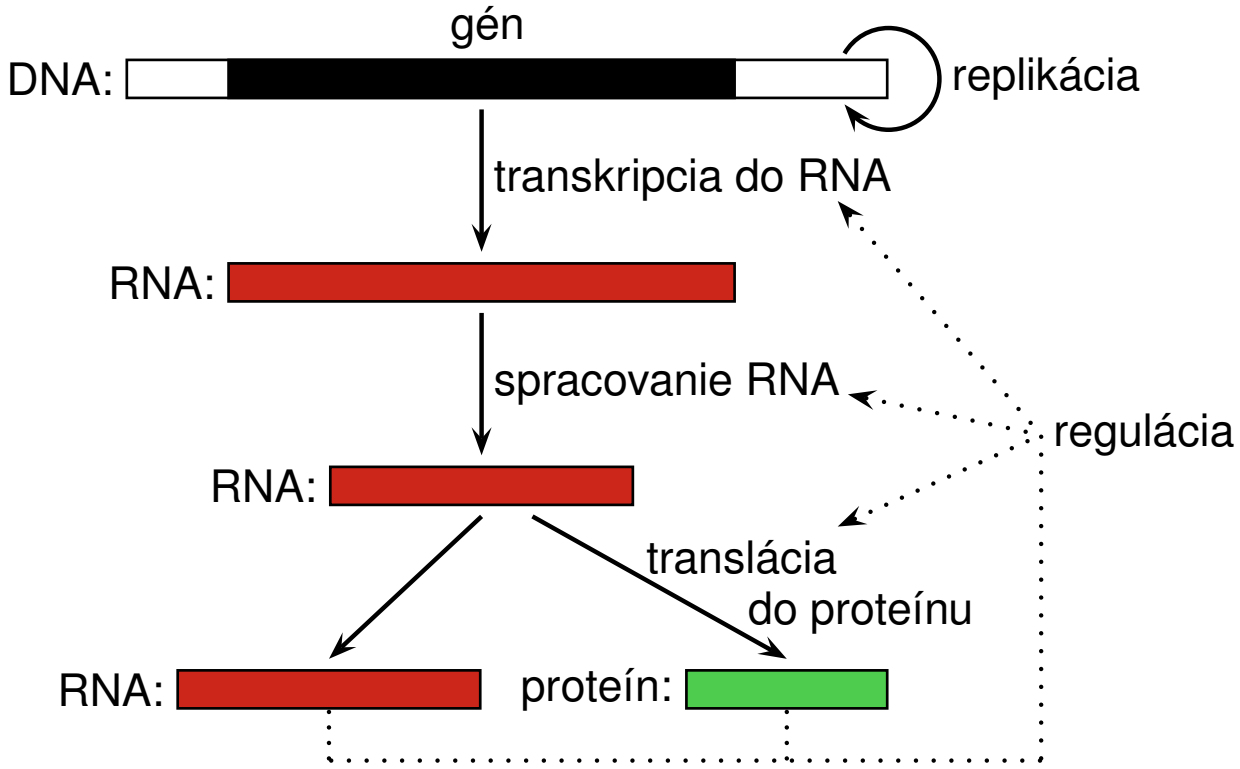
- Pochopiteľná pre študentov tohto predmetu (inf aj bio)
- Vysvetlite pojmy, ktoré sú nad rámec tohto predmetu
- Netreba pokryť všetko, môžete využiť aj iné zdroje
- Podrobne vysvetliť aspoň jednu bioinformatickú metódu a aspoň jeden biologický výsledok (alebo overovanie správnosti metódy na dátach)
- Ako článok súvisí s učivom preberaným na predmete
- Nájdite zopár citujúcich prác, ktoré výsledky využili alebo vylepšili
- Rozsah cca 1-2 strany na osobu, jeden ucelený text
- Píšte vlastnými slovami, citujte zdroje
- V správe vymenujte členov skupiny, ktorí sa podieľali na jej spísaní, dostanú rovnako bodov
- Pdf odovzdať cez Moodle (stačí 1 za skupinu)

RNA

Tomáš Vinař

5.12.2024




gén
DNA:
replikácia
transkripcia do RNA
RNA:
spracovanie RNA
regulácia
RNA:
translácia
do proteínu
RNA:
proteín:

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)
- rôzne funkcie v bunke:
  centrálna úloha pri expresii génov (mediátorová, transferová, ribozómová RNA),
  regulácia expresie,
  katalytické funkcie,
  prenos genetickej informácie pre RNA vírusy

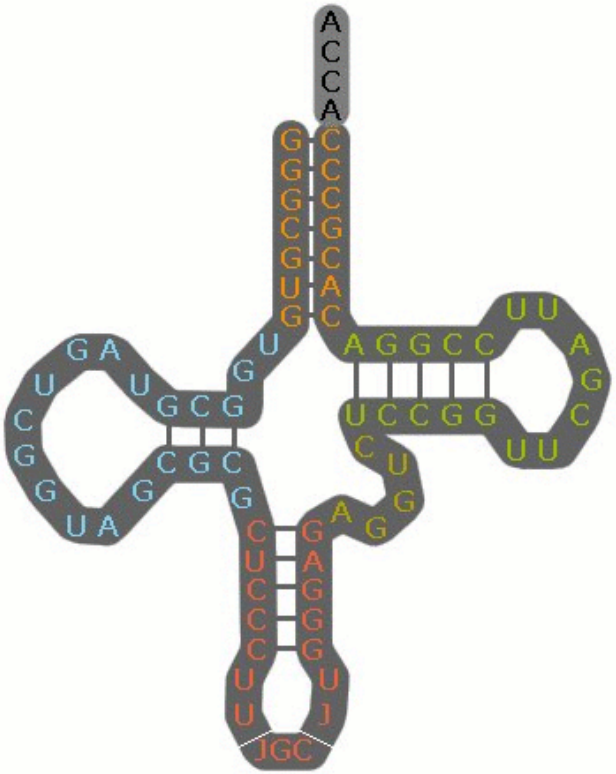
Štruktúra RNA

Príklad: transferová RNA (transfer RNA)

Sekundárna štruktúra

(secondary structure):

páry nukleotidov

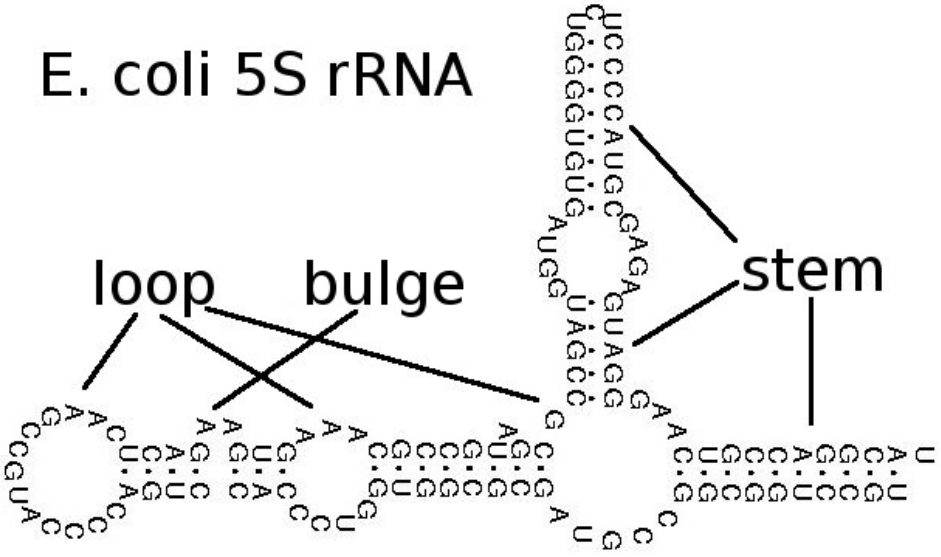


Terciárna štruktúra

(tertiary structure):

3D súradnice

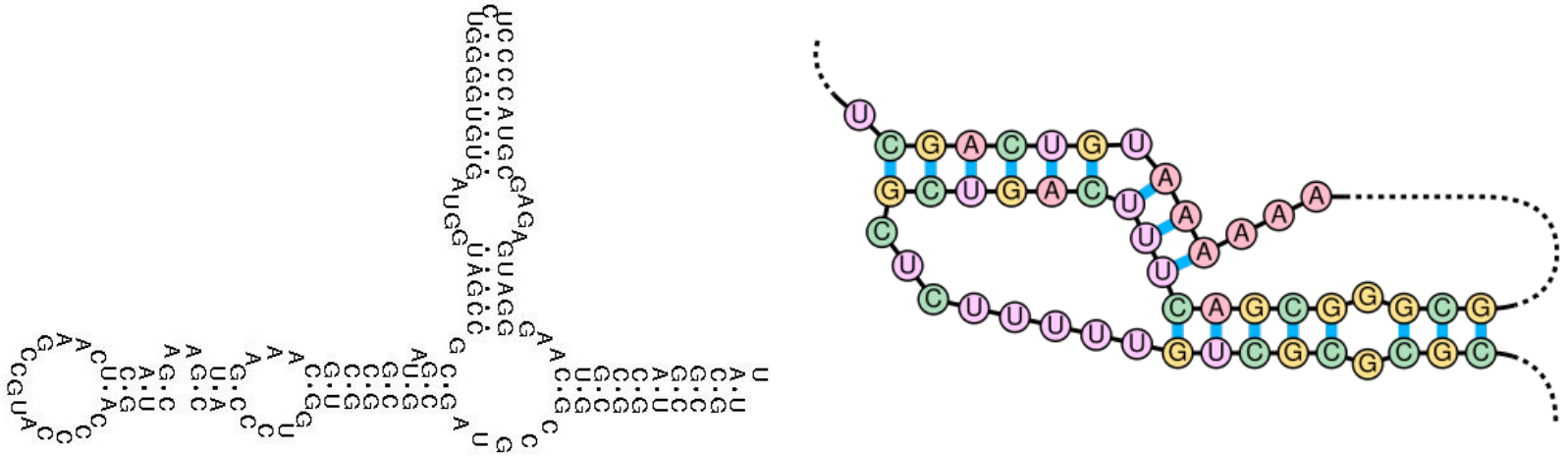
Sekundárna štruktúra RNA



Zápis sekundárnej štruktúry pomocou zátvoriek:

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

Dobre uzátvorkované výrazy vs pseudouzly



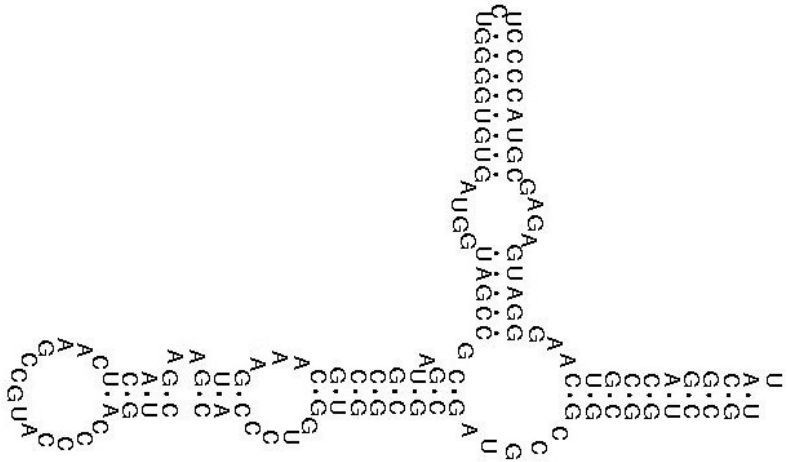
Príklad **vľavo:** spárované bázy tvoria **dobre uzátvorkovaný výraz:**

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

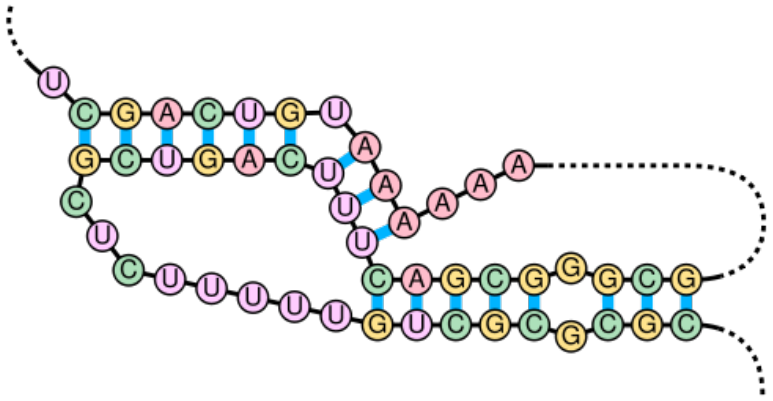
Príklad **vpravo: pseudouzol** (výnimka z dobrého uzátvorkovania)

```
.((((((.(((...[[[.[[[[[[)))))))))).    .]]]]]].]]]
UCGACUGUAAAAAAGCGGGCGACUUUCAGUCGC...UGUCGCGCGC
```

Dobre uzátvorkované výrazy vs pseudouzly



bez pseudouzlu

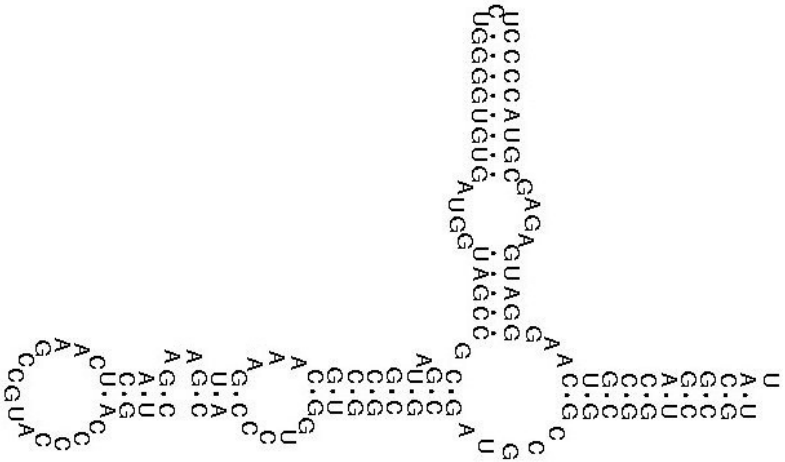


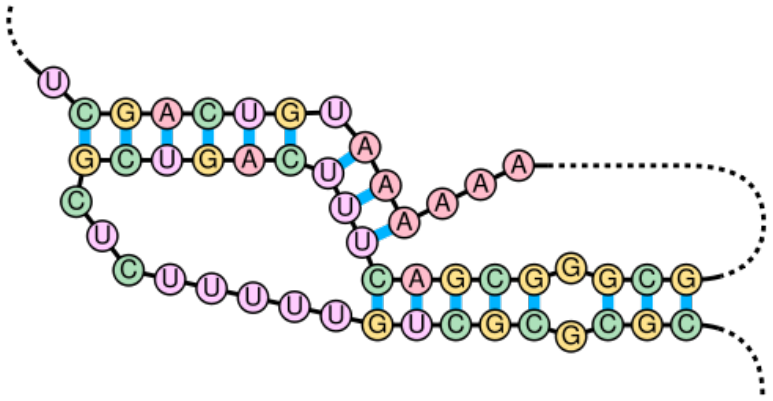
pseudouzol

Mnohé algoritmy na prácu so sekundárnou štruktúrou ignorujú pseudouzly.

Zhruba 1.4% RNA nukleotidových párov v pseudouzloch.

Dobre uzátvorkované výrazy vs pseudouzly



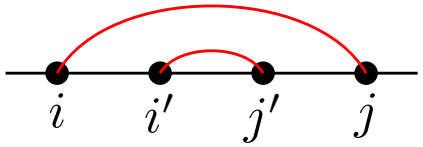


Matematické vyjadrenie štruktúry bez pseudozulov:

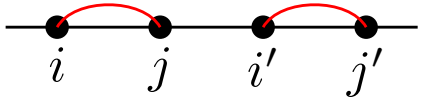
Ak máme páry medzi pozíciami i a j a i' a j' pričom $i < i'$,

tak buď $i < i' < j' < j$ alebo $i < j < i' < j'$.

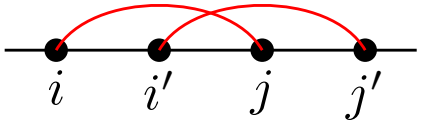
dobre:



dobre:



zle:



Problém: určovanie štruktúry RNA

Vstup: RNA sekvencia

Cieľ: nájsť spárované bázy

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

Príklad:

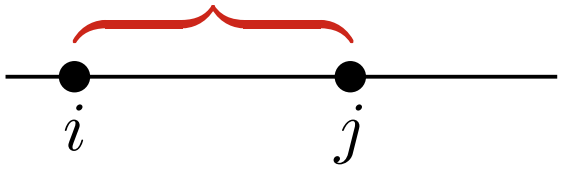
```
Vstup:   ((.(((()))(((.)))))))
Výstup: GAACACAUGUAAAAUUUGUC
```

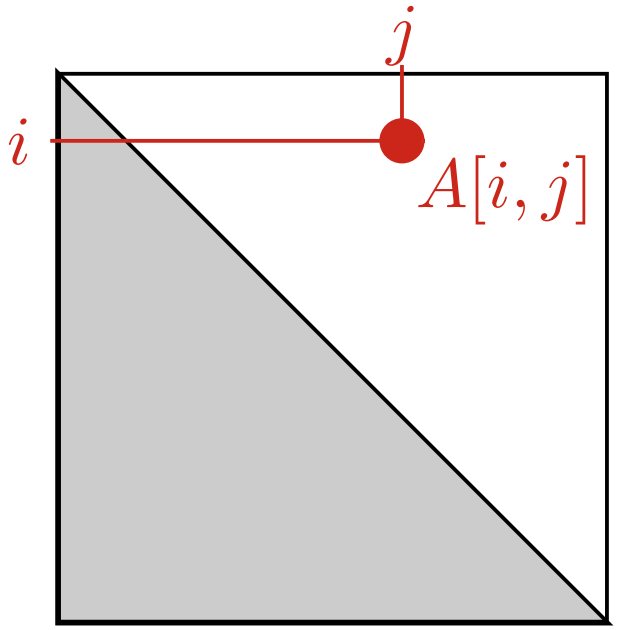
Nussinovovej algoritmus

Dynamické programovanie:

Majme RNA $x_1, \ldots, x_n$.

Nech $A[i, j]$ je maximálny počet párov v podreťazci $x_i, x_{i+1}, \ldots, x_j$.





Nussinovovej algoritmus

Dynamické programovanie:

Majme RNA $x_1, \ldots, x_n$.

Nech $A[i, j]$ je maximálny počet párov v podreťazci $x_i, x_{i+1}, \ldots, x_j$.

Rekurencia:

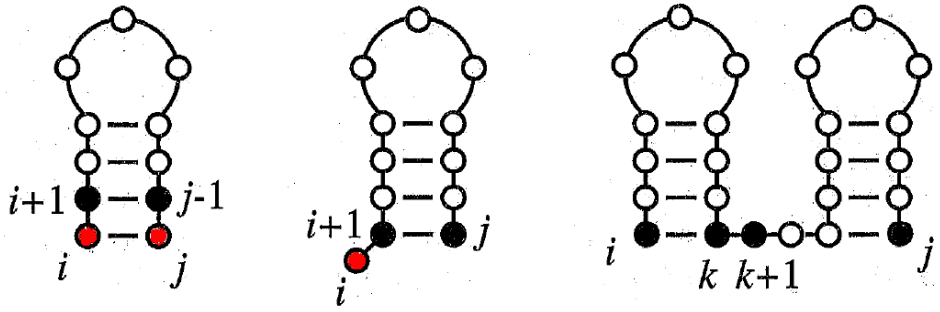
Podreťazce dĺžky 1: žiadne páry $A[i, i] = 0$

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 $i < k < j$: $A[i, j] = A[i, k] + A[k + 1, j]$

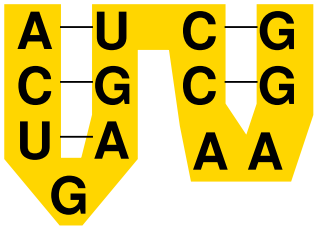


Rekurencia: $A[i,j] = \max \begin{cases} A[i+1, j-1] + c(x_i, x_j), \\ A[i+1, j], \\ \max_{k=i+1 \ldots j-1}\{A[i,k] + A[k+1,j]\} \end{cases}$

	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

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

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



Zložitosť:

$O(n^3)$ čas

$O(n^2)$ pamäť

Štruktúra s minimálnou voľnou energiou (MFE folding)

Realistickejšia formulácia problému určovania sekundárnej 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.

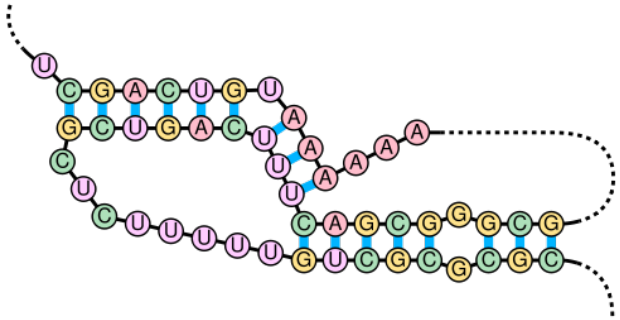
Príklad:

```
5′  CX  3′
3′  GY  5′
```

	Y: A	C	G	U
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 (opakované vytváranie silných helixov) alebo celočíselné lineárne programovanie (posledné cvičenia pre informatikov)

Pravdepodobnostné modely na predikciu štruktúry

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

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 (SCFG)

neterminály (veľké písmená) podobné na stavy v HMM,

terminály (malé písmená) reprezentujú nukleotidy.

Pravidlá prepisujú neterminál na reťazec terminálov a neterminálov.

Každé pravidlo má pravdepodobnosť.

Príklad: jeden neterminál, 14 pravidiel (ϵ =prázdny reťazec)

$$S \to \overbrace{aSu}^{0.1} \mid \overbrace{uSa}^{0.1} \mid \overbrace{cSg}^{0.1} \mid \overbrace{gSc}^{0.1} \mid$$

$$\overbrace{aS}^{0.05} \mid \overbrace{cS}^{0.05} \mid \overbrace{gS}^{0.05} \mid \overbrace{uS}^{0.05} \mid \overbrace{Sa}^{0.05} \mid \overbrace{Sc}^{0.05} \mid \overbrace{Sg}^{0.05} \mid \overbrace{Su}^{0.05} \mid \overbrace{SS}^{0.1} \mid \overbrace{\epsilon}^{0.1}$$

V každom kroku zvoľ jeden (napr. najľavejší) neterminál,

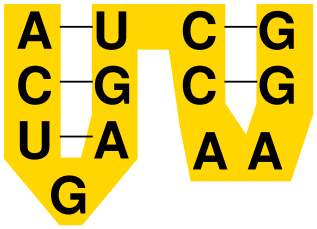
prepíš ho náhodne zvoleným pravidlom:

$$S \to SS \to aSuS \to acSguS \to acuSaguS \to acugSaguS \to$$
$$acugaguS \to acugagucSg \to acugagucc Sgg \to acugagucc Sagg \to$$
$$acugagucc aSagg \to acugaguccaagg$$

Stochastické bezkontextové gramatiky

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

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



Bázy vygenerované v jednom kroku sú spárované.

Úloha: Nájdi najpravdepodobnejšie odvodenie danej RNA

$\Rightarrow$ 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 všetky aspekty problému,
- nižšia presnosť ako minimalizácia energie.

Evolúcia RNA sekvencií

Často vidíme koreláciu medzi mutáciami v spárovaných bázach.

Napr. C sa zmení na A, spárované G sa súčasne zmení na U

Príklad: niekoľko sekvencií z D ramena tRNA

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

Úloha: daných je niekoľko (zarovnaných) sekvencií RNA

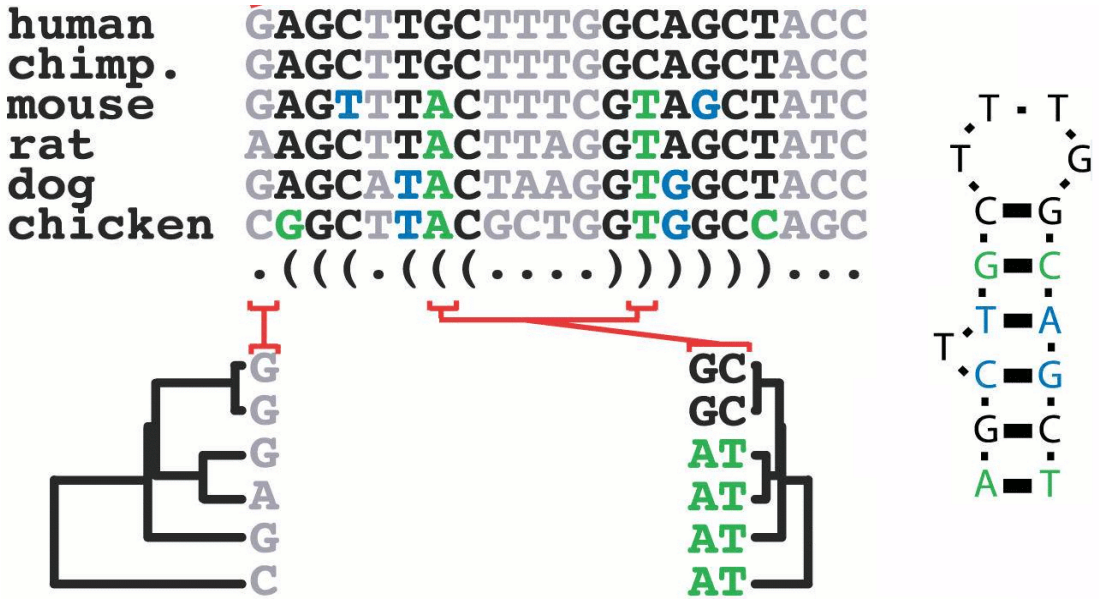
nájdite ich spoločnú RNA štruktúru

(korelácie medzi spárovanými bázami potvrdzujú správnosť štruktúry)

Hľadanie spoločnej štruktúry pre viacero sekvencií

Phylo-SCFG:

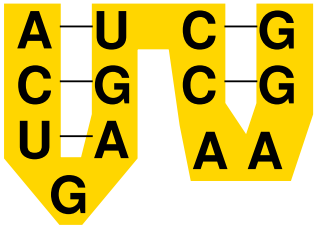
- namiesto jednotlivých báz emituje stĺpce zarovnania podľa fylogenetického stromu.
- nespárované bázy emituje bežnou substitučnou maticou,
- spárované bázy substitučnou maticou dvojíc (16×16).



Problém: hľadanie známych typov RNA génov v genóme

- Databáza Rfam: štruktúry pre >4000 rodín RNA génov
- Pre každú rodinu zarovnanie a pravdepodobnostný model
- Pre RNA kovariančné modely (covariance model, CM): špeciálny typ SCFG
- Podobná myšlienka ako profilové HMM pre proteínové rodiny

Kovariančný model



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

- 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**

  Napr. $P_1 \to \overbrace{aP_2u}^{0.2} \mid \overbrace{uP_2a}^{0.2} \mid \overbrace{cP_2g}^{0.4} \mid \overbrace{cP_2u}^{0.1}$

- veľkosť gramatiky úmerná dĺžke modelovanej RNA rodiny

Kovariančný model

Použitie:

hľadaj výskyty génu v DNA (lokálne zarovnanie),

nájdi 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:

nájdi sľubné úseky podobné na sekvencie v RNA rodine

(iba na základe podobnosti sekvencií)

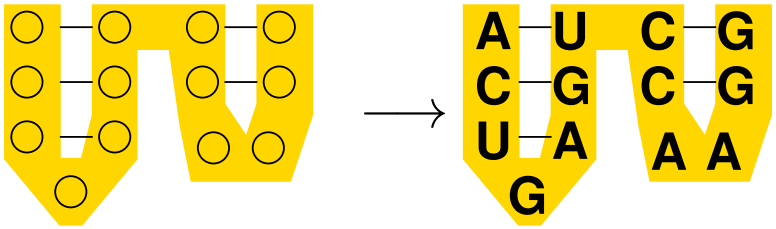
aplikuj CM iba na sľubné úseky

Problém: dizajn RNA

Daná RNA sekundárna štruktúra (párovanie).

Nájdi 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, po prednáške cvičenia pre biológov
- Budúci štvrtok 19.12.:
  – posledné cvičenia pre informatikov
  – nepovinné prezentácie journal clubu v čase prednášky (chceme?)
- Termíny na konci semestra
  – DÚ3 streda 18.12., správy zo journal clubu piatok 20.12.

Skúška pre informatikov (BIN, INF, DAV, AIN)

- Treba získať aspoň polovicu bodov
- Na stránke sú ukážky jednoduchých príkladov, cca 50% bodov
  – v prípade záujmu pred skúškou konzultačné hodiny (kedy?)
- Zvyšné príklady budú prekvapením, v minulosti sa vyskytli:
  – Krátke príklady na pochopenie základných pojmov
  – Navrhnite/modifikujte algoritmus alebo model
- Povolené pomôcky:
  – písacie potreby, ťahák 2 listy A4, jednoduchá kalkulačka
- **Termín** 14.1.2025, 9:00
  Hláste prípadné závažné konflikty,
  Dátumy opravných termínov dohodneme s tými,
  ktorých sa budú týkať

Polymorfizmus a populačná genetika

Tomáš Vinař a Broňa Brejová

12.12.2024

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

UK Biobank:

500 000 genómov (Nov.2023) plus rozsiahle medicínske dáta o účastníkoch

Plus mnoho ďalších veľkých projektov

Mapovanie asociácií (Trait/Disease Association Mapping)

- Znaky (a choroby) vznikajú kombináciou genetických a environmentálnych vplyvov
- Cieľ: Identifikovať genetické vplyvy
  - Aký je risk choroby s dedičným faktorom u danej osoby?
  - Ako fungujú choroby (na základe génov, ktorých mutácie ich spôsobujú)?
  - 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

Diploidné genómy

- Človek má **diploidný genóm**:
  má v bunkách 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 (formami) 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

Diploidné genómy

- Človek má **diploidný genóm**:
  má v bunkách 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 (formami) a, A
  môže byť **homozygot** (aa alebo AA),
  alebo **heterozygot** (aA)
- **Haplotyp:** kombinácia aliel rôznych SNPov na tom istom chromozóme
  (zdedená od jedného rodiča)
  Diploidný jedinec má teda dva haplotypy

chr1 od matky: ...A...T...G... ...

chr1 od otca: ...T...C...A... ...

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 **vylúčime nulovú hypotézu**, našli sme asociáciu, nemusí však ísť o príčinu

Ak ju nevylúčime, nepreukázali sme súvis SNPu s veľkosťou

(môže ale existovať, možno treba viac dát)

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 (χ^2) test: obľúbený približný test, vhodný ak máme vysoké počty

Na testovanie genetických asociácií sa používajú aj zložitejšie štatistické modely (napr. diploidný genóm, príbuzenské vzťahy, . . .)

Testovanie nezávislosti v kontingenčnej tabuľke χ^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 χ^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 $\chi^2 = \sum_{i\in\{m,v\}} \sum_{j\in\{A,a\}} \frac{(O_{i,j}-E_{i,j})^2}{E_{i,j}}$

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

χ^2 je určitá miera rozdielnosti tabuliek O a E.

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

Testovanie nezávislosti v kontingenčnej tabuľke χ^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, χ^2 je približne z rozdelenia $\chi^2(1)$,

t.j. **chí kvadrát s jedným stupňom voľnosti**.

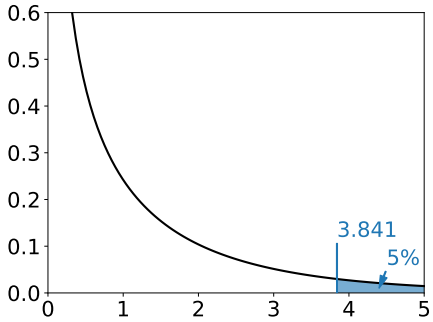
1 stupeň: ak poznáme E a 1 políčko z O, zvyšok O vieme dopočítať.

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
p	142	773

$\chi^2 = 184.78$, P-hodnota $4.4 \cdot 10^{-42}$

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
p	2	12

$\chi^2 = 3.0867$, P-hodnota 0.07893

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

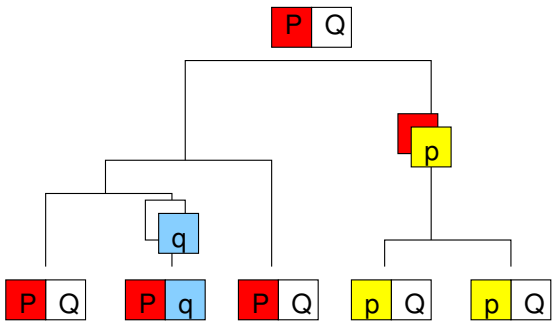
Ale pozor, pre takéto malé hodnoty χ^2 **nepresný**

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)**

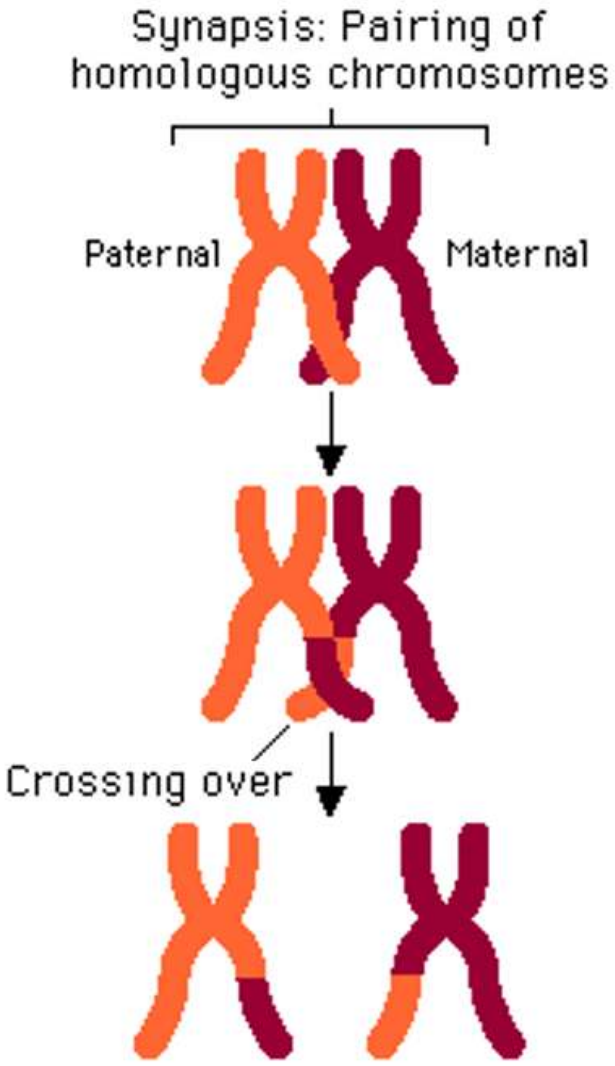
Blízko seba na tom istom chromozóme:



- Málokedy mutácia na to istom mieste 2x, zriedkavá rekombinácia
- Kombinácie nie sú úplne náhodné
- Korelácie medzi SNPmi

  $\Rightarrow$ **väzbová nerovnováha, linkage disequilibrium (LD)**

Rekombinácia



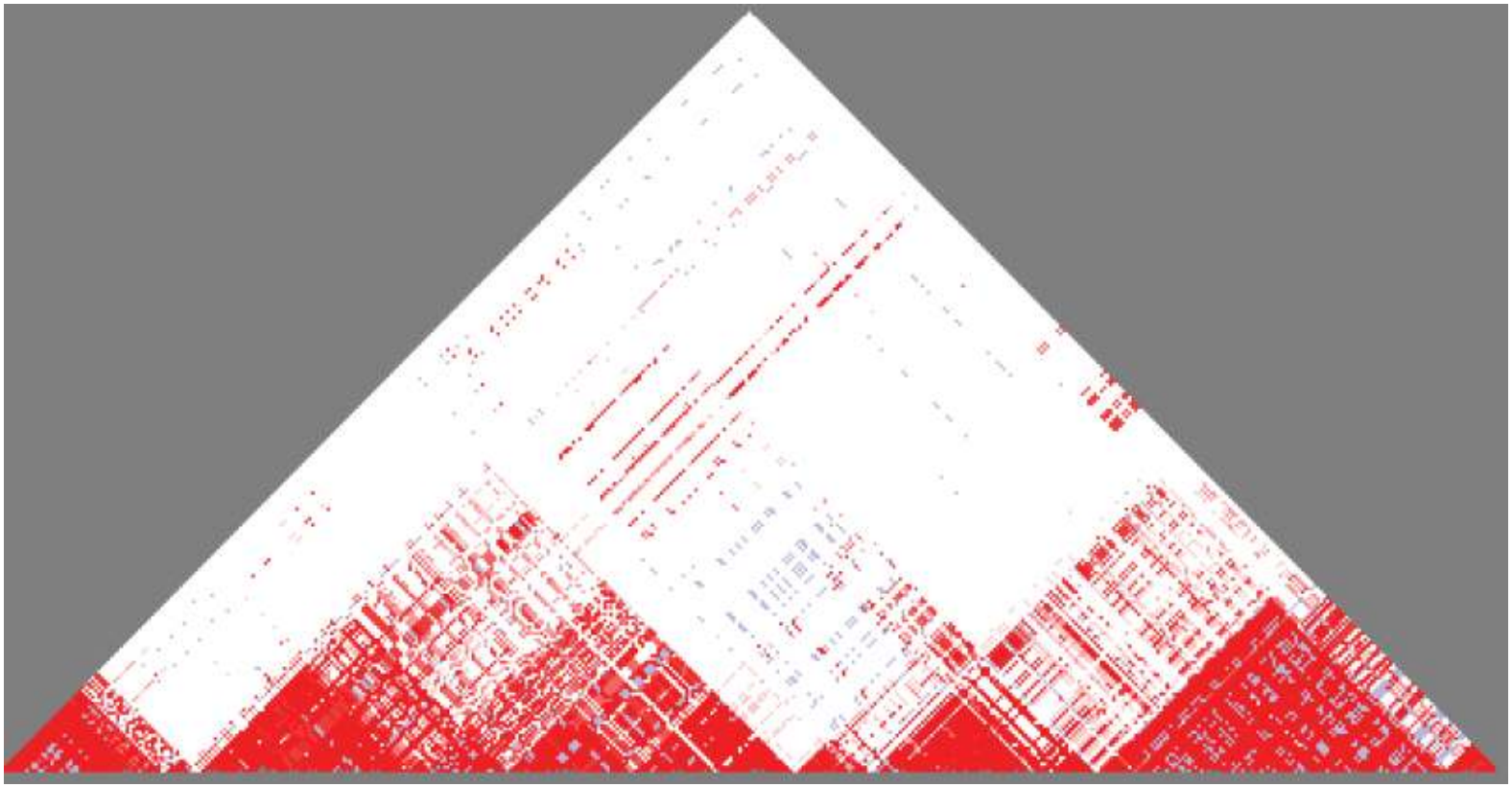
Cca 1-3 **rekombinácie** v 1 ľudskom chromozóme počas meiózy (tvorba pohlavných buniek)

Rekombinácia znižuje LD

Ak predpokladáme rovnomernú rekombináciu:

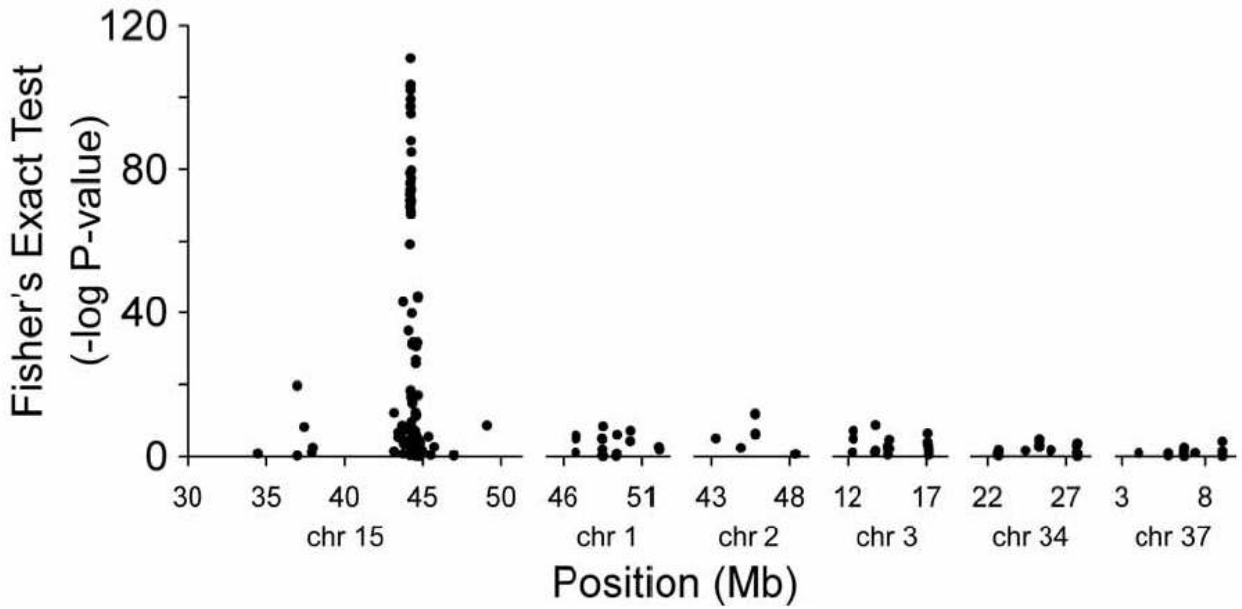
- Čím vzdialenejšie SNPy, tým nižšie LD
- Čím staršie SNPy, tým nižšie LD
- Ďalšie aspekty: štruktúra populácie, prirodzený výber, rekombinačné hotspoty

Väzbová nerovnováha (LD) v ľudskom genóme



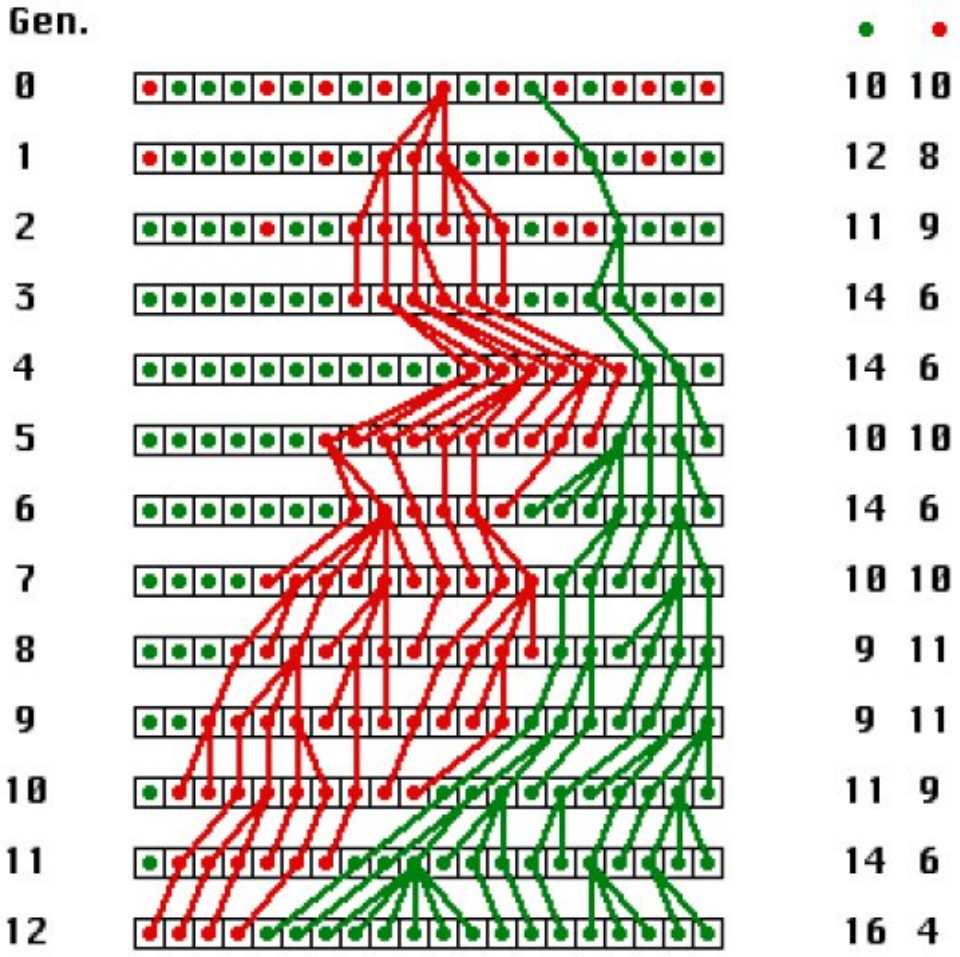
Región ENm014 (500kB, chr 7), 90 ľudí Utah, project HapMap

Späť k psom: Hľadanie asociácií v celom genóme (Genome-Wide Association Study, GWAS)



- V prípade štúdie veľkosti psov: GWAS identifikoval 84 kB región
- Pozíciu ďalej treba spresniť ďalšími experimentami
- **Veľké LD bloky** $\Rightarrow$ veľké výsledné regióny

Základný model populačnej genetiky: Wrightov-Fisherov model



Životný cyklus SNPov vo Wrightovom-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 $X_t \in \{0, 1, \ldots, N\}$

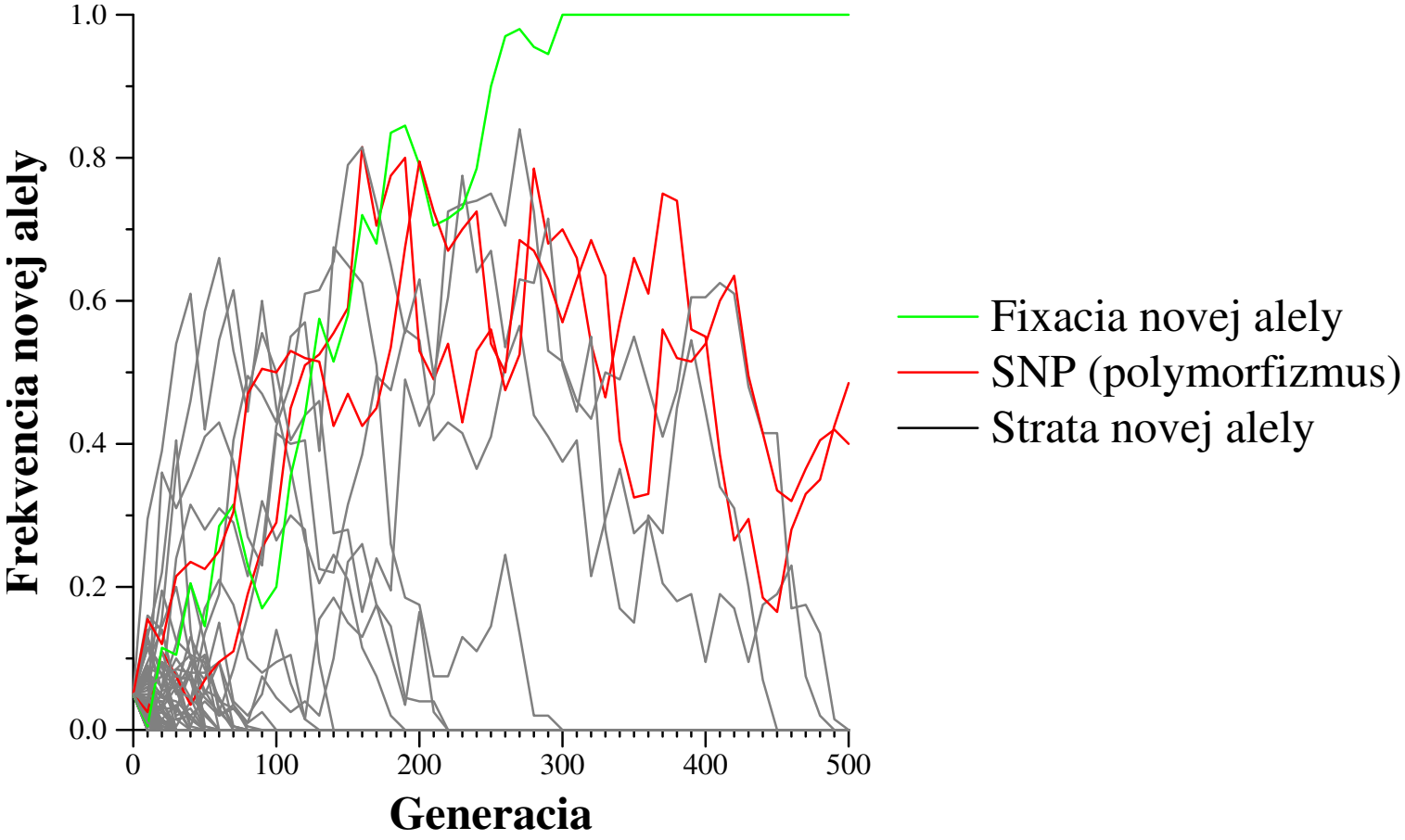
$$\Pr(X_t = j \mid X_{t-1} = i) = \left(\frac{i}{N}\right)^j \left(\frac{N-i}{N}\right)^{N-j} \binom{N}{j}$$

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

- Stavy 0 and N sú **pohlcujúce**

Náhodný genetický drift

$N = 200$, $X_0 = 10$, 500 generácií



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 menšinovej alely SNPov
- Heterozygotnosť u diploidných jedincov
- Počet a veľkosť LD blokov

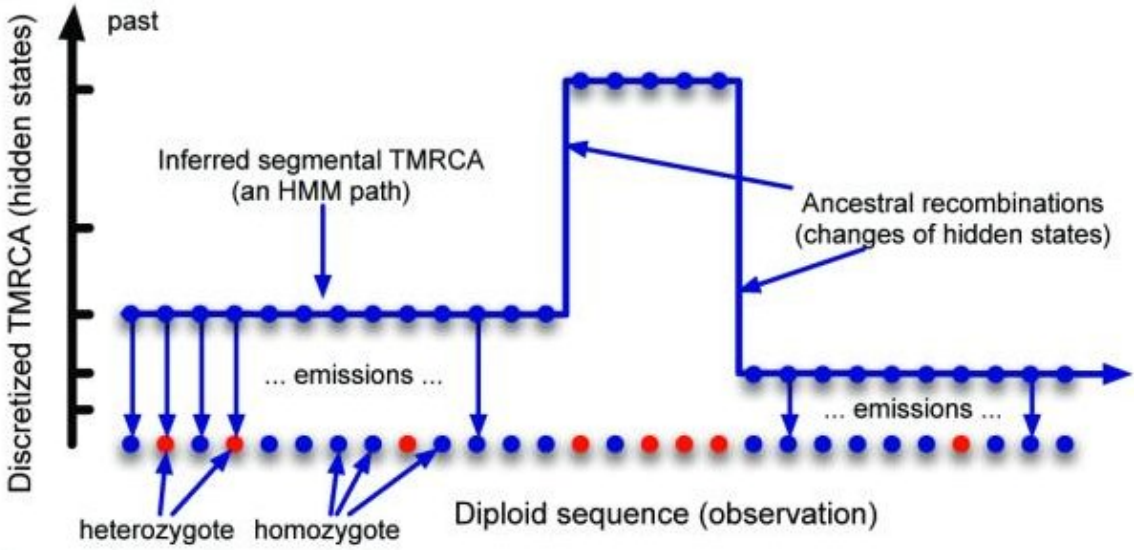
Použitie:

Snažíme sa nájsť parametre modelu, ktoré najlepšie vysvetľujú pozorované dáta u osekvenovaných jedincov.

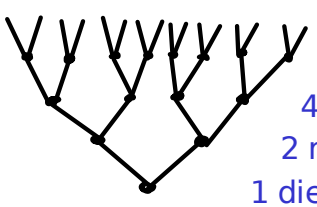
História ľudskej populácie z genómu jedinca (Li, Durbin 2011)

- **Parametre modelu:** história vývoja efektívnej veľkosti ľudskej populácie v čase
- **Použité dáta:** Pozície heterozygotných SNPov v rámci genómu

  Z ich premenlivej hustoty určí rozdelenie časov ku najbližšiemu spoločnému predkovi (TMRCA)

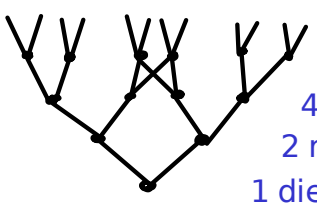


Čas k najbližšiemu spoločnému predkovi a počet mutácií



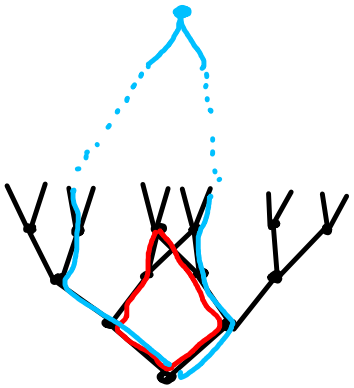
16 praprarodičov
8 prarodičov
4 starí rodičia
2 rodičia
1 dieťa

bez sobášov medzi blízkymi príbuznými



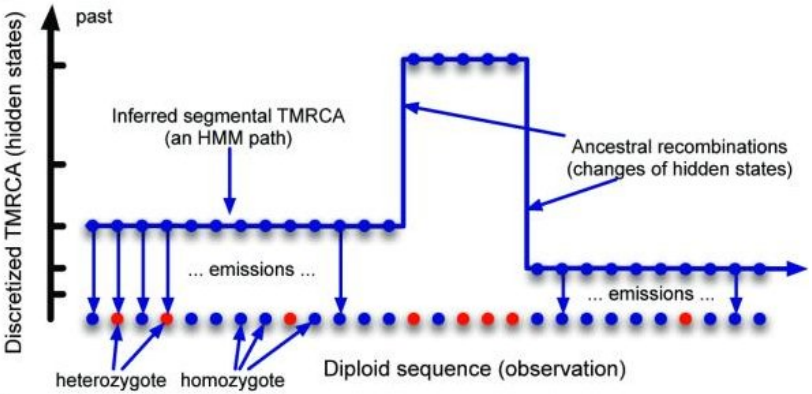
12 praprarodičov
6 prarodičov
4 starí rodičia
2 rodičia
1 dieťa

bratranec a sesternica majú dieťa

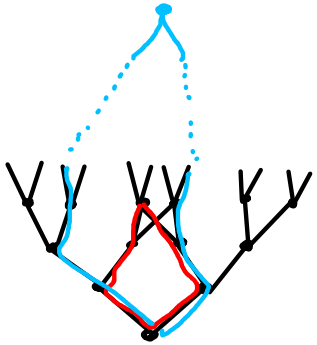


dávny spoločný predok,
veľa mutácií

nedávny spoločný predok,
málo mutácií



Čas k najbližšiemu spoločnému predkovi a veľkosť populácie



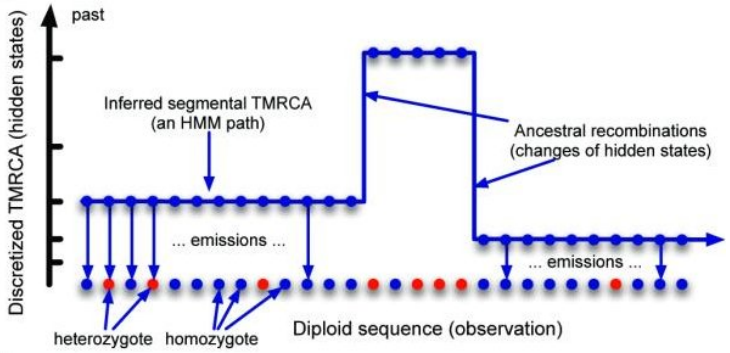
dávny spoločný predok,
veľa mutácií

nedávny spoločný predok,
málo mutácií

Ak v čase t žilo málo ľudí

– predkovia sa budú viac “opakovať”

– čas t sa častejšie vyskytne ako TMRCA

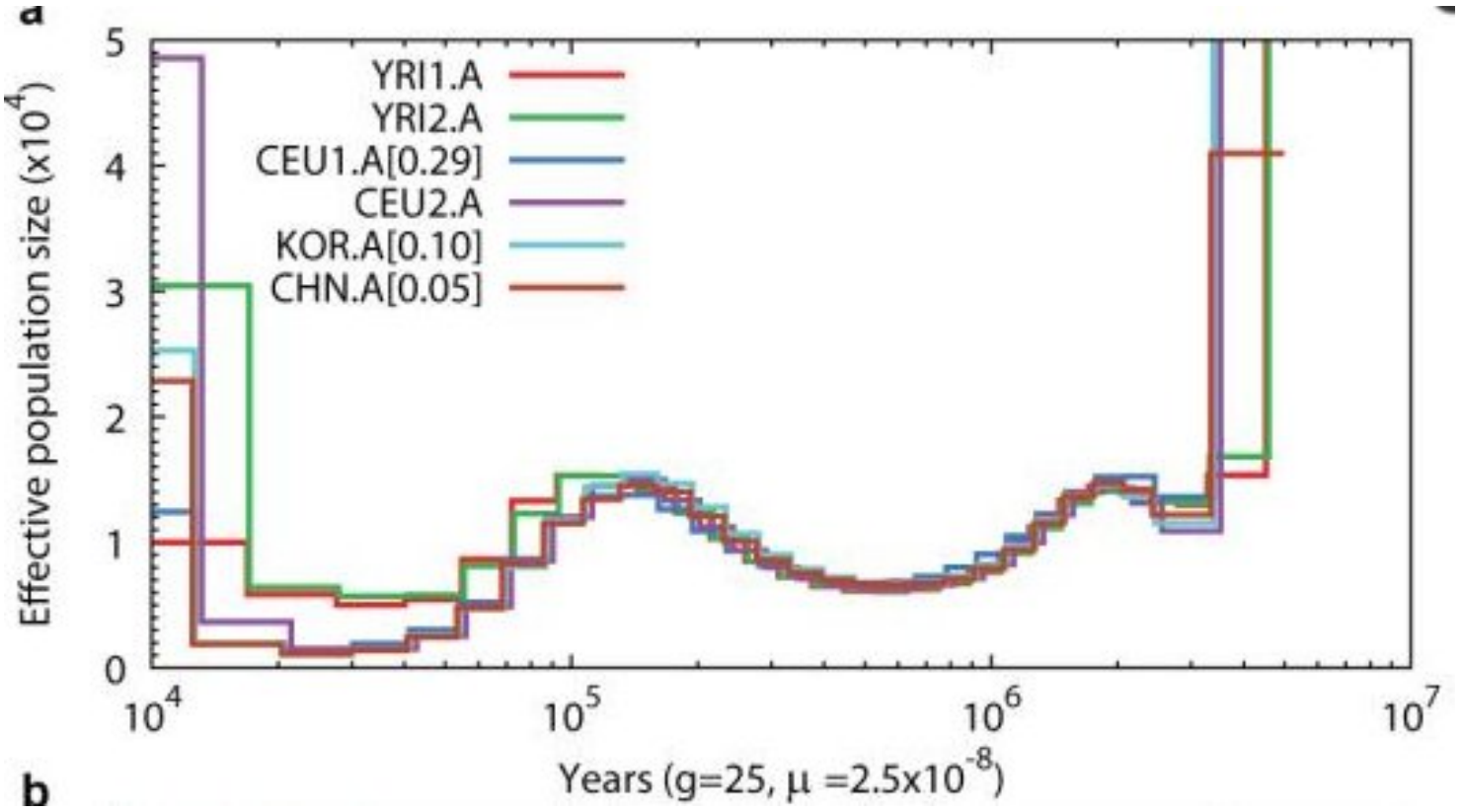


Z lokálnej hustoty mutácií odhademe TMRCA

Z rozdelenia TMRCA odhadneme veľkosti populácie

Príklad: História ľudskej populácie z genómu jedinca

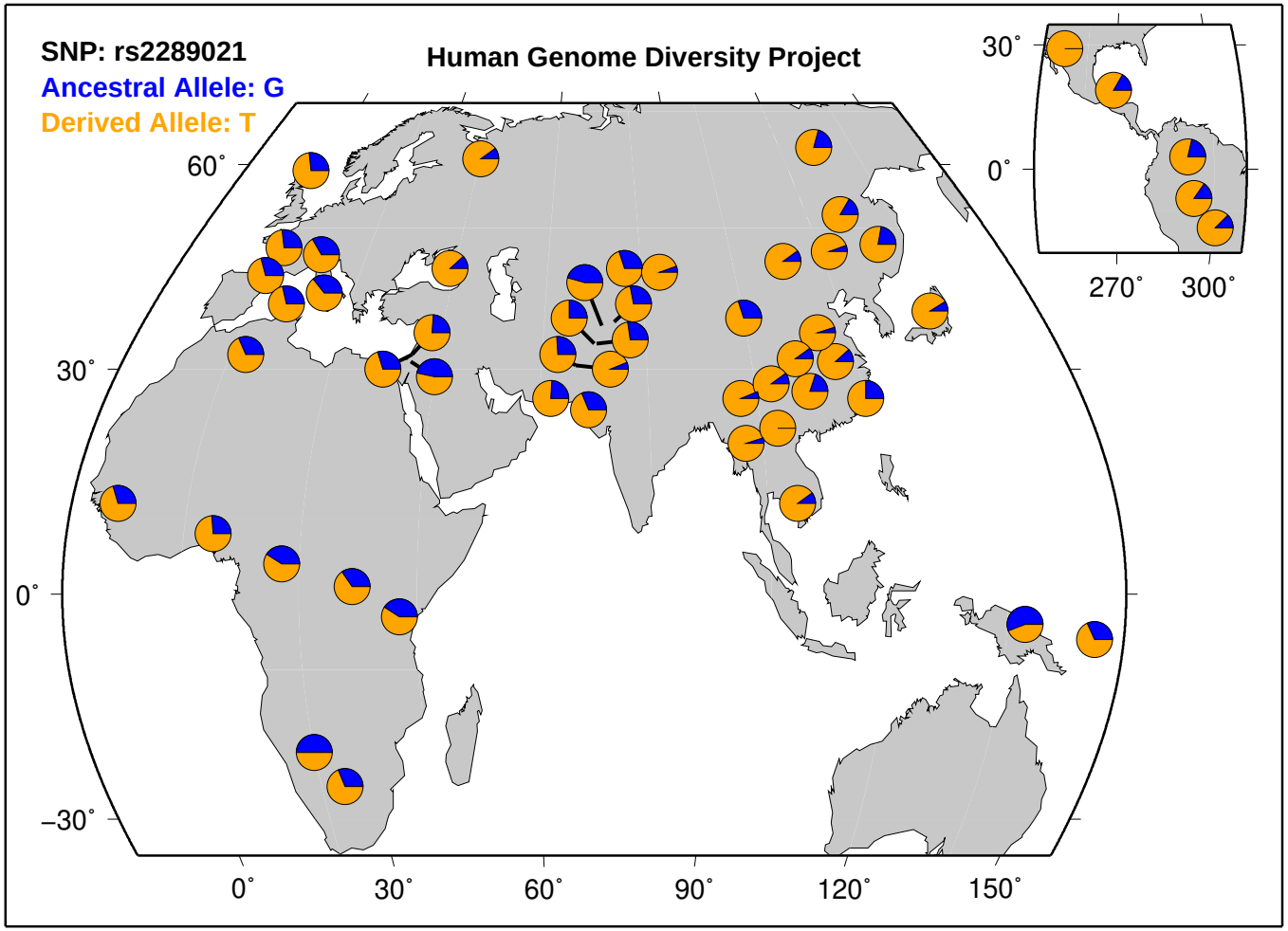
Úloha: Nájdi históriu vývoja efektívnej veľkosti ľudskej populácie, ktorá najlepšie vysvetľuje pozorované dáta



Štruktúra populácie

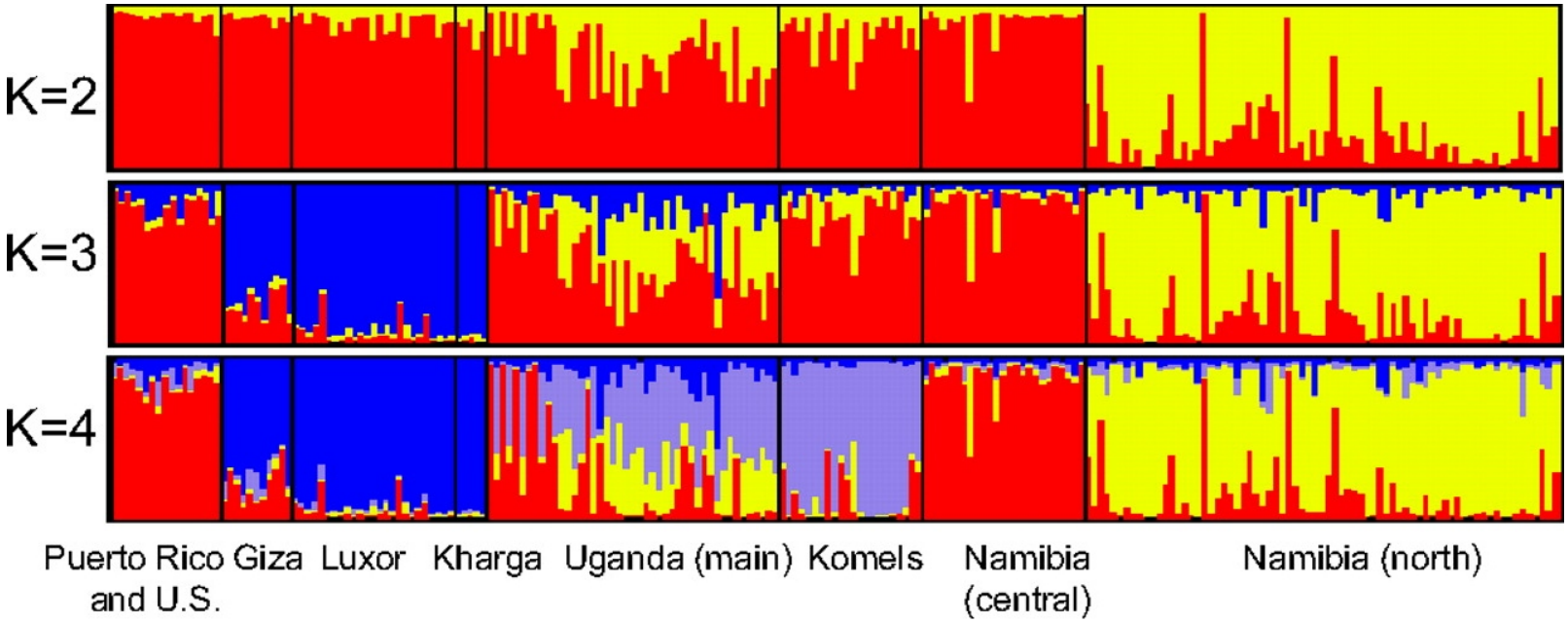
- Doteraz sme predpokladali, že nová generácia vzniká **náhodným párovaním** (random mating)
- Väčšina organizmov sa vyvíja v **subpopuláciách**, s obmedzeným prenosom genetického materiálu medzi subpopuláciami
- Frekvencie toho istého SNPu v dvoch subpopuláciách môžu byť značne odlišné
- $\Rightarrow$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

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 $\Pr[X \mid P, Q, Z]$ a apriórne rozdelenie pre P, Q
- **Výstup:** $E[Q \mid X]$

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 $\Pr(P, Q \mid X, Z^{(i-1)})$
  - Vyber náhodnú vzorku $Z^{(i)}$ z distribúcie $\Pr(Z \mid X, P^{(i)}, Q^{(i)})$
- Pre vhodné m, c, priemer postupnosti

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

konverguje k hodnote $E[Q \mid X]$

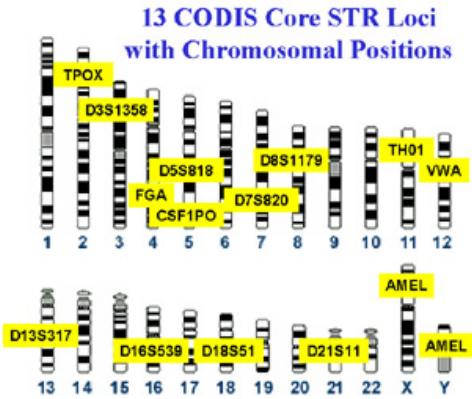
Zhrnutie

- **SNPy (single nucleotide polymorphizms)** 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 (**genome-wide 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 štandarný “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)