Metódy v bioinformatike, 2-AIN-501

Vyučujú:
Broňa Brejová, M-163, brejova@fmph.uniba.sk
Tomáš Vinař, M-163, vinar@fmph.uniba.sk

Web: http://compbio.fmph.uniba.sk/vyuka/mbi/

Diskusné fórum a oznamy: Piazza

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.

Ciele predmetu

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

Známkovanie

3 domáce úlohy 30% (10% každá)
Journal club 10%
Skúška alebo projekt 60%

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

- Dve verzie otázok: biologická a informatická
- Journal club: čítanie 1 článku v skupine a správa (alebo prezentácia)
- Projekt povinný pre doktorandov, nepovinný pre ostatných
- Na skúške povolený ťahák 2 listy A4
- Neopisovať!

Časy a miestnosti

Prednáška štvrtok 15:40-17:10 F1-328
Cvičenia informatici štvrtok 14:00-15:30 F1-328
Cvičenia biológovia štvrtok 17:20-18:50 F1-328 a F1-248
(počítačová učebňa)

Malá ukážka na úvod: hľadanie homológií

Biologický problém

- Človek aj myš majú cca 20 tisíc génov
- Gén je reťazec s niekoľko tisíc znakmi (mRNA sekvencia)
- Veľa z týchto génov aj v spoločnom predkovi človeka a myši pred 75mil. rokmi
- Cca 15-30% báz (znakov) zmutovalo
- **Homológy:** dvojica génov z toho istého predka
- **Cieľ:** Pre daný ľudský gén chceme nájsť homológ v myši
- **Dáta:** Jeden reťazec (ľudský gén), 20 tisíc reťazcov (myšie gény)

ale informatik chce matematicky dobre definovaný problém...

>gi|169791021|ref|NM_000937.3| Homo sapiens polymerase (RNA) II (DNA directed) polypeptide A, 220kDa (POLR2A), mRNA

TTTTTTTTTCCTTTTTGGGAGCTGAAAAATTTCCGGTAAGGGAAAGAAGGGCTCCTTTCGCTCCTTATTT
CCCCGCCTCCTTCCCTCCCCCACCTTCCCCTCCTCCGGCTTTTTCCTCCCAACTCGGGGAGGTCCTTCCC
GGTGGCCGCCCTGACGAGGTCTGAGCACCTAGGCGGAGGCGGCGCAGGCTTTTTGTAGTGAGGTTTGCGC
CTGCGCAGCGCGCCTGCCTCCGCCATGCACGGGGGTGGCCCCCCCTCGGGGGACAGCGCATGCCCGCTGC
GCACCATCAAGAGAGTCCAGTTCGGAGTCCTGAGTCCGGATGAACTGAAGCGAATGTCTGTGACGGAGGG
TGGCATCAAATACCCAGAGACGACTGAGGGAGGCCGCCCCAAGCTTGGGGGGCTGATGGACCCGAGGCAG
GGGGTGATTGAGCGGACTGGCCGCTGCCAAACATGTGCAGGAAACATGACAGAGTGTCCTGGCCACTTTG
GCCACATTGAACTGGCCAAGCCTGTGTTTCACGTGGGCTTCCTGGTGAAGACAATGAAAGTTTTGCGCTG
TGTCTGCTTCTTCTGCTCCAAACTGCTTGTGGACTCTAACAACCCAAAGATCAAGGATATCCTGGCTAAG
TCCAAGGGACAGCCCAAGAAGCGGCTCACACATGTCTACGACCTTTGCAAGGGCAAAAACATATGCGAGG
GTGGGGAGGAGATGGACAACAAGTTCGGTGTGGAACAACCTGAGGGTGACGAGGATCTGACCAAAGAAAA
GGGCCATGGTGGCTGTGGGCGGTACCAGCCCAGGATCCGGCGTTCTGGCCTAGAGTTGTATGCGGAATGG
AAGCACGTTAATGAGGACTCTCAGGAGAAGAAGATCCTGCTGAGTCCAGAGCGAGTGCATGAGATCTTCA
AACGCATCTCAGATGAGGAGTGTTTTGTGCTGGGCATGGAGCCCCGCTATGCACGGCCAGAGTGGATGAT
TGTCACAGTGCTGCCTGTGCCCCCGCTCTCCGTGCGGCCTGCTGTTGTGATGCAGGGCTCTGCCCGTAAC
CAGGATGACCTGACTCACAAACTGGCTGACATCGTGAAGATCAACAATCAGCTGCGGCGCAATGAGCAGA
ACGGCGCAGCGGCCCATGTCATTGCAGAGGATGTGAAGCTCCTCCAGTTCCATGTGGCCACCATGGTGGA
CAATGAGCTGCCTGGCTTGCCCCGTGCCATGCAGAAGTCTGGGCGTCCCCTCAAGTCCCTGAAGCAGCGG
TTGAAGGGCAAGGAAGGCCGGGTGCGAGGGAACCTGATGGGCAAAAGAGTGGACTTCTCGGCCCGTACTG
TCATCACCCCCGACCCCAACCTCTCCATTGACCAGGTTGGCGTGCCCCGCTCCATTGCTGCCAACATGAC
CTTTGCGGAGATTGTCACCCCCTTCAACATTGACAGACTTCAAGAACTAGTGCGCAGGGGGAACAGCCAG
TACCCAGGCGCCAAGTACATCATCCGAGACAATGGTGATCGCATTGACTTGCGTTTCCACCCCAAGCCCA
GTGACCTTCACCTGCAGACCGGCTATAAGGTGGAACGGCACATGTGTGATGGGGACATTGTTATCTTCAA
CCGGCAGCCAACTCTGCACAAAATGTCCATGATGGGGCATCGGGTCCGCATTCTCCCATGGTCTACCTTT
CGCTTGAATCTTAGTGTGACAACTCCGTACAATGCAGACTTTGACGGGGATGAGATGAACTTGCACCTGC
CACAGTCTCTGGAGACGCGAGCAGAGATCCAGGAGCTGGCCATGGTTCCTCGCATGATTGTCACCCCCCA
GAGCAATCGGCCTGTCATGGGTATTGTGCAGGACACACTCACAGCAGTGCGCAAATTCACCAAGAGAGAC
GTCTTCCTGGAGCGGGGTGAAGTGATGAACCTCCTGATGTTCCTGTCGACGTGGGATGGGAAGGTCCCAC
AGCCGGCCATCCTAAAGCCCCGGCCCCTGTGGACAGGCAAGCAAATCTTCTCCCTCATCATACCTGGTCA
CATCAATTGTATCCGTACCCACAGCACCCATCCCGATGATGAAGACAGTGGCCCTTACAAGCACATCTCT
CCTGGGGACACCAAGGTGGTGGTGGAGAATGGGGAGCTGATCATGGGCATCCTGTGTAAGAAGTCTCTGG
GCACGTCAGCTGGCTCCCTGGTCCACATCTCCTACCTAGAGATGGGTCATGACATCACTCGCCTCTTCTA...

Ako zadefinovať potenciálne homológy?

- Mali by mať veľa spoločných (zachovaných) báz
  niektoré bázy môžu byť zmenené, pridané, ubraté
  ATGCACGTTAAT
  AGCACGCTACCAT

- Zobrazenie vo forme zarovnania
  AT**GCACG**T**TA**--**AT**
  A-**GCACG**C**TA**CC**AT**

Informatický problém: najdlhšia spoločná podpostupnosť
longest common subsequence (lcs)

- Vstup: dva reťazce
- Problém: Ako z nich ubrať čo najmenej znakov tak, aby sa potom rovnali?
- Výstup: Spoločná podpostupnosť po ubraní znakov, resp. jej dĺžka

Rozšírenie problému lcs na hľadanie homológov

- Vstup: reťazec X a reťazce $Y_1, Y_2, \ldots, Y_n$
- Problém: Nájsť všetky reťazce Y_i také, že najdlhšej spoločná podpostupnosť X a Y_i pokrýva aspoň 70% dĺžky X aj Y_i.
- Matematicky: $|\mathsf{lcs}(X, Y_i)| \geq 0.7|X| \wedge |\mathsf{lcs}(X, Y_i)| \geq 0.7|Y_i|$
- Výstup: Všetky nájdené reťazce Y_i, dĺžky lcs, zarovnania

Poznámky

- Dôležité: rozdelenie na formuláciu (**čo presne znamená, že chceme nájsť homológy**) a riešenie (**akým spôsobom homológy ideme hľadať**)
- Formulácia problému zvyčajne interakciou medzi biológmi a informatikmi/matematikmi
- Riešenie problému daného formuláciou obvykle informatici (neskôr si ukážeme, ako sa problém lcs rieši pomocou dynamického programovania)
- Program správne riešiaci tento problém nemusí vždy nájsť správne homológy!
- V praxi sa hľadanie homológií definuje trochu zložitejšie:
  rôzne pokuty za mutácie, inzercie, delécie
  bonusy za zachované bázy, ...

Čo nás čaká ďalej

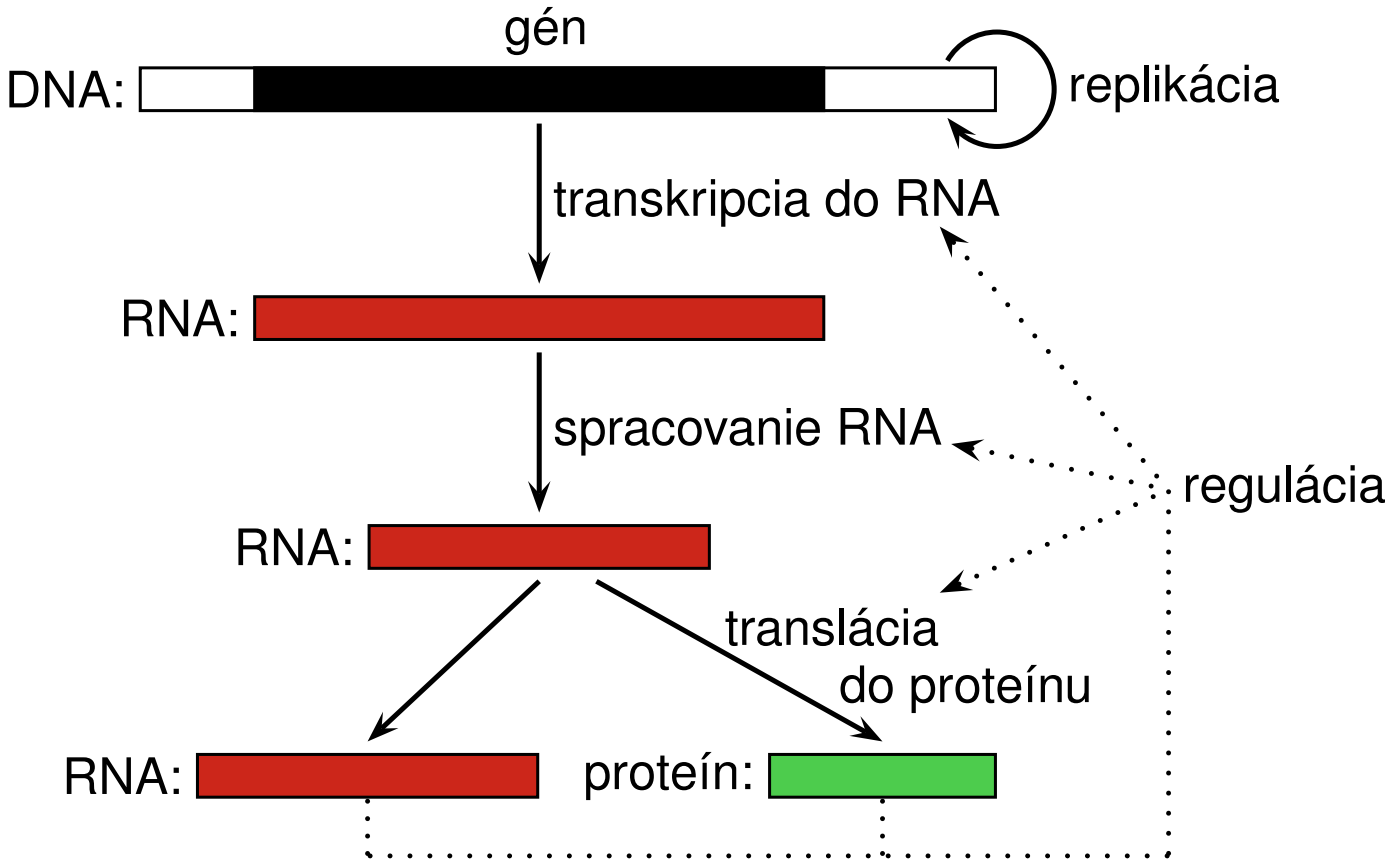
Typická prednáška

- 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

Prehľad predmetu



Osnova predmetu

- **Sekvenovanie, zostavovanie sekvencií**
  Ako sa získavajú DNA sekvencie, akú rolu v tom hraje informatika?
  Grafové algoritmy, skladačka z obrovského množstva malých kúskov
- **Zarovnávanie (sequence alignment)**
  Ktoré sekvencie sú podobné na moju obľúbenú sekvenciu?
  Čo presne robí BLAST a ako mu nastavím parametre?
  Dynamické programovanie, heuristiky a ako povedať niečo o tom, ako dobre fungujú
- **Hľadanie génov**
  Koľko génov má človek?
  Skryté Markovove modely (HMM)

- **Evolúcia, rekonštrukcia fylogenetických stromov**
  Ku komu máme bližšie: ku psom alebo k myšiam?
  A má hroch bližšie k veľrybám alebo k prasatám?
  Pravdepodobnostné modely evolúcie, princíp úspornosti (parsimony), metódy riešenia ťažkých úloh

- **Komparatívna genomika**
  Ako sa líšime od šimpanzov?
  Ktoré časti genómu sa vyvíjajú pomalšie alebo rýchlejšie ako by sme čakali a prečo?
  Spojenie HMM a evolučných modelov

- **Expresia a regulácia génov**
  Ktoré gény slúžia ako iniciátori bunkovej samodeštrukcie?
  Dá sa jednoduchým vyšetrením rozlíšiť, či má konkrétny pacient zhubnú rakovinu a či konkrétny liek bude fungovať?
  Zhlukovanie (clustering), biologické siete a ich vlastnosti

- **Transkripčné faktory**

  Ako funguje mechanizmus riadenia expresie génov?

  Vieme niektoré gény umelo zapínať a vypínať?

  Hľadanie opakujúcich sa motívov v sekvenciách

  Rozpoznávanie známych motívov pomocou strojového učenia

- **Štruktúra a funkcia proteínov**

  Akú 3D štruktúru má môj obľúbený proteín?

  Akú funkciu má v živej bunke?

  Čo spôsobuje Alzheimerovu chorobu a prečo?

  HMM, energetické modely, molekulárne dynamické simulácie

- **RNA**

  Ako predikovať sekundárnu štruktúru RNA a hľadať RNA gény?
  Koľko tRNA je v mojom obľúbenom genóme?
  Dynamické programovanie, stochastické bezkontextové gramatiky

- **Populačná genetika**

  Prečo Tibeťania nemajú problémy so životom vo veľkých výškach?
  Ako to, že plemená psov vyzerajú tak rôzne, a napriek tomu sú
  jeden druh? A odkedy je vlastne pes najlepším priateľom človeka?
  Pravdepodobnostné modely, stochastické algoritmy, štatistika

Súvisiace predmety

- Seminár z bioinformatiky (1)-(4)
  Utorok 17:20 (I-9), oba semestre
- Genomika
  letný semester, PriFUK
- Strojové učenie
  Streda 11:30 (F1-108), Štvrtok 8:10 (F1-328), zimný semester
- Vybrané partie z dátových štruktúr
  letný semester
- Grafové modely v strojovom učení
  letný semester, budúci školský rok

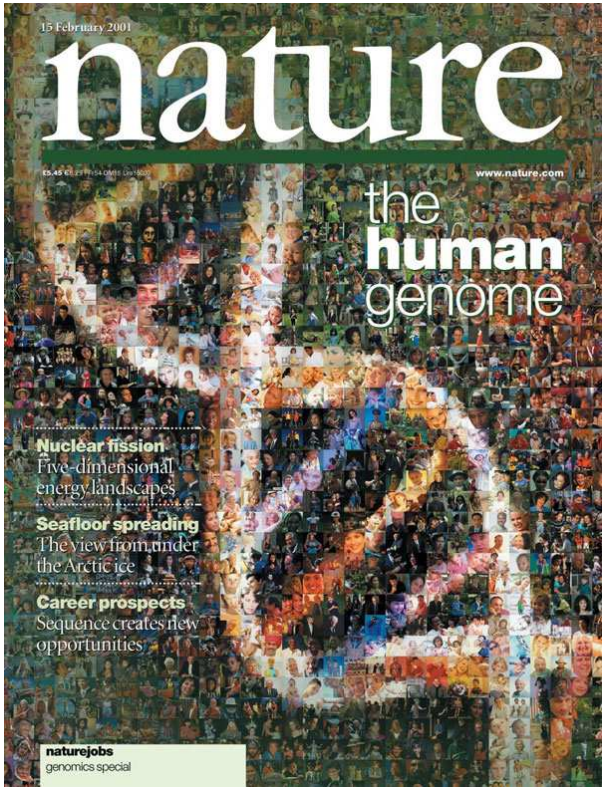
Organizačné poznámky

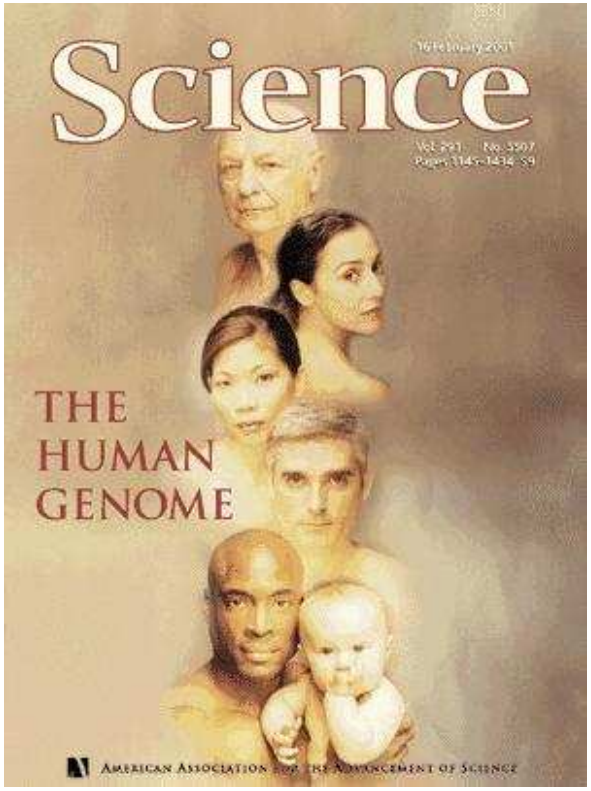
- Nezabudnite sa zapísať!
- Ďalšie info http://compbio.fmph.uniba.sk/vyuka/mbi/
- (Na stránke budú aj všetky prezentácie.)
- Oznamy v systéme Piazza
- Budúci týždeň nebudú cvičenia, iba prednáška (o 15:40)

Sekvenovanie a zostavovanie genómov
(genome sequencing and assembly)

Tomáš Vinař

2.10.2014



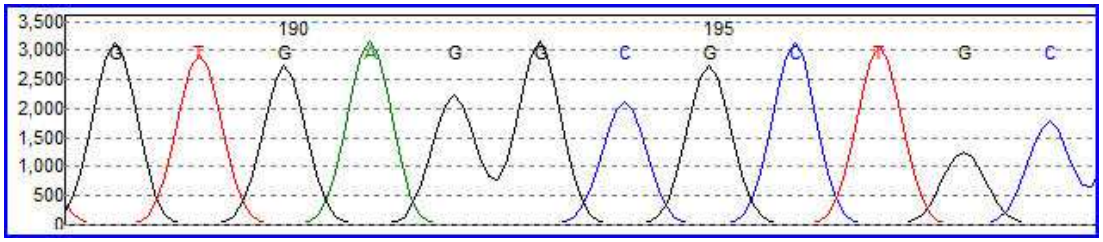


Sekvenovanie genómov

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

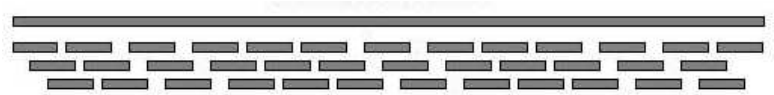
Sangerovo sekvenovanie

- Výsledok: sekvenovací profil (trace)



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

Bioinformatický problém:
zostavenie genómu (sequence assembly)



- **Vstup:** krátke segmenty sekvenovanej DNA
- **Cieľ:** zostaviť pôvodnú DNA
  — riadime sa zhodou v prekrývajúcich častiach segmentov
- Dôležité faktory:
  - **dĺžka genómu**
  - **pokrytie** (coverage) – koľko krát segmenty pokrývajú genóm?
- **BAC-by-BAC:** začíname s mapou genómu, rozdelíme genóm na kratšie BACy (cca 200 KB)
- **Shotgun:** bez mapy — veríme bioinformatike!

Formulácia problému

Najkratšie spoločné nadslovo (shortest common superstring)

Úloha: Daných je niekoľko reťazcov (osekvenovaných segmentov), nájdite **najkratší** reťazec, ktorý obsahuje **všetky** segmenty ako (súvislé) **podreťazce**.

Motivácia: čo najviac využiť prekryvy medzi segmentami

Príklad: AAA, AAC, ACA, ACC, CAA, CAC, CCA, CCC

Riešenie: AAACCCACAA (najkratšie možné)

Najkratšie spoločné nadslovo

- **Problém je NP ťažký**
  takže nepoznáme rýchly algoritmus, ktorý vždy nájde najlepšie riešenie
- **Jednoduchá heuristika:** opakovane nájdi dva segmenty, ktoré sa prekrývajú najviac a zlúč ich do jedného segmentu
- 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: problémy s formuláciou

- **Výpočtovo ťažký problém**
- **Nerealistická formulácia:** v praxi mnoho ďalších faktorov
- **Prečo najkratší reťazec?** Čo ak sa nejaký úsek opakuje?
- **Motivácia:** poznatky zo skúmania zjednodušeného problému môžeme zovšeobecniť
- **Ale:** v zostavovaní sekvencií v praxi iné metódy

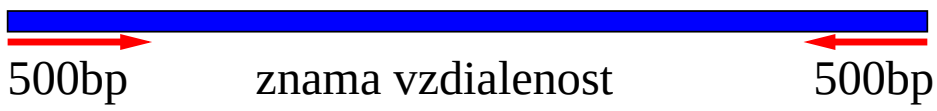
Nerealistická formulácia

Sťažujúce faktory:

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

Nerealistická formulácia

Zľahčujúci faktor: spárované segmenty (Pair-end reads)



plazmid 2−10 kB
kozmid 40 kB

Zjednodušenie: nemusíme spojiť všetko do jedného reťazca, spájame len časti spojené viacerými segmentami

Overlap-Layout-Consensus

Napr. ARACHNE

- **Overlap:** (Prekryv)
  - Nájdeme prekrývajúce sa segmenty
  - Zostavíme ich do väčších **kontigov** (contigs)
- **Layout:** (Rozmiestnenie)
  - Určíme relatívnu polohu jednotlivých kontigov a ich vzdialenosti (pomocou spárovaných segmentov)
  - Dostaneme **superkontigy** (supercontigs, scaffolds) s možnými dierami
  - V ideálnom prípade, superkontigy zodpovedajú chromozómom
- **Consensus:**
  - Pre každú bázu prezrieme všetky prekrývajúce segmenty
  - Ak sa nezhodujú, zoberieme konsenzus (napr. väčšinové pravidlo, s ohľadom na kvalitu bázy)

Sekvenovacie technológie novej generácie

- Komerčne prístupné cca od roku 2004, stále prudký rozvoj
- Obrovské množstvo segmentov naraz
- Rýchlejšie a oveľa lacnejšie

Nevýhody:

- Kratšie segmenty
- Treba robiť veľa sekvenovania naraz z jednej vzorky

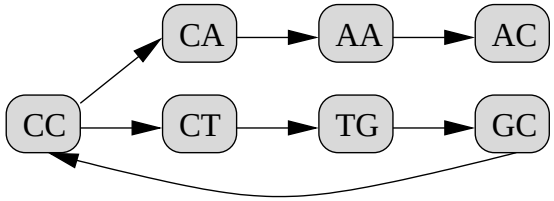
Sekvenovacie technológie novej generácie

Pareek et al 2011

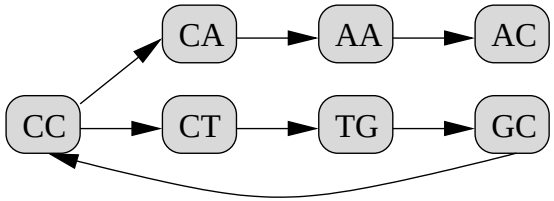
	454 (Roche)	Illumina (Solexa)	SOLiD (AB)
Čas/beh	10 h	2-5 dní	6 dní
Data/beh	400 Mb	3000Mb	4000Mb
Dĺžka fragmentu	400bp	35-100bp	35-50bp
Cena/beh	$8 500	$9 000	$17 500
Cost per Mb	$85	$6	$6

de Bruijnove grafy (napr. VELVET)

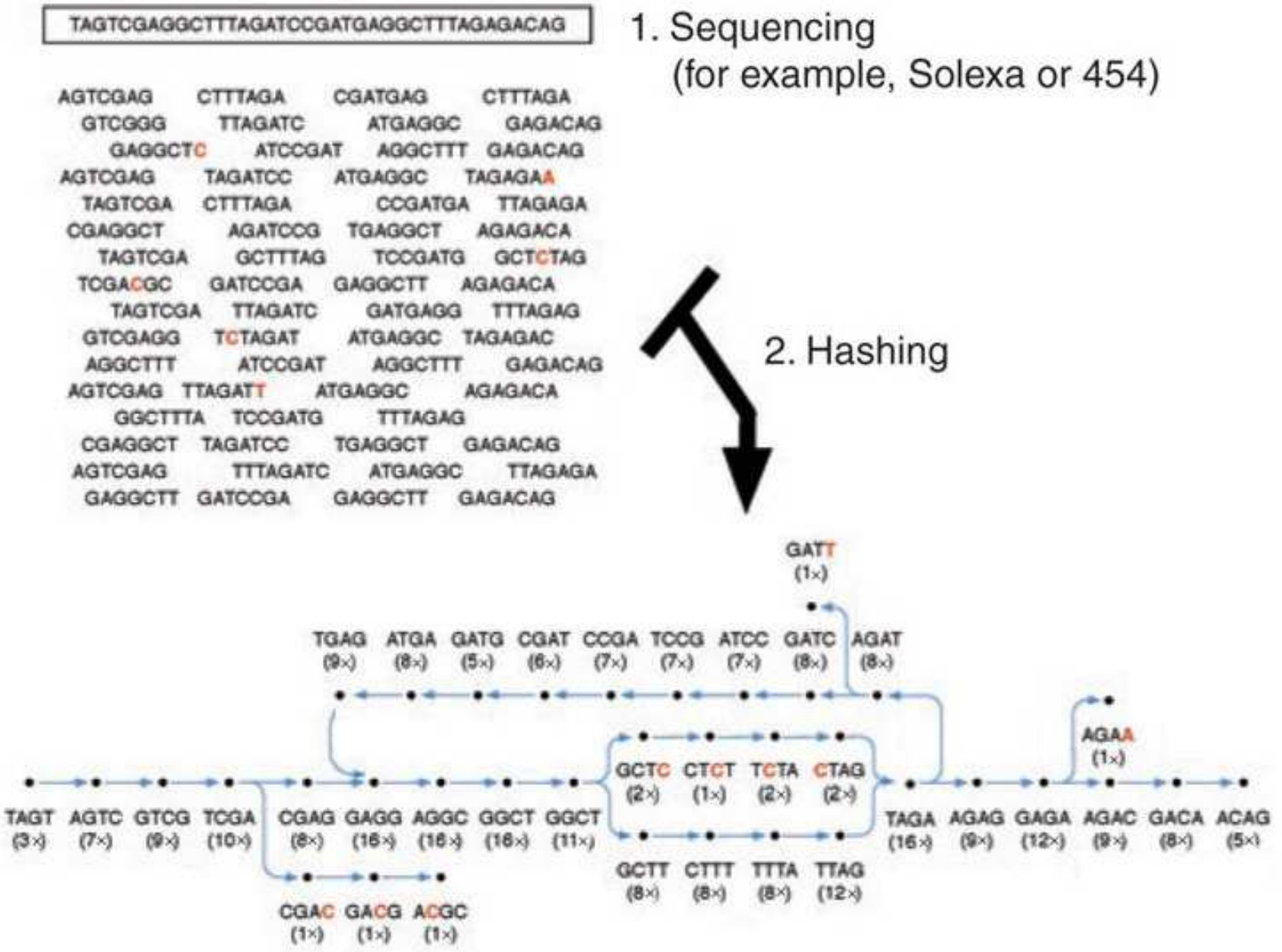
- Predpokladajme jednu orientáciu, žiadne chyby, jeden chromozóm úplne pokrytý segmentami
- Nasekajme segmenty na (prekrývajúce sa) kúsky dĺžky k
- Zostavme z nich **de Bruijnov graf**
  - **vrcholy:** podreťazce dĺžky k všetkých segmentov
  - **hrany:** nadväzujúce k-tice v rámci každého segmentu (s prekryvom $k-1$)
  - Graf je orientovaný (hrany majú smer)
- **Príklad:** $k = 2$, segmenty: CCTGCC, GCCAAC

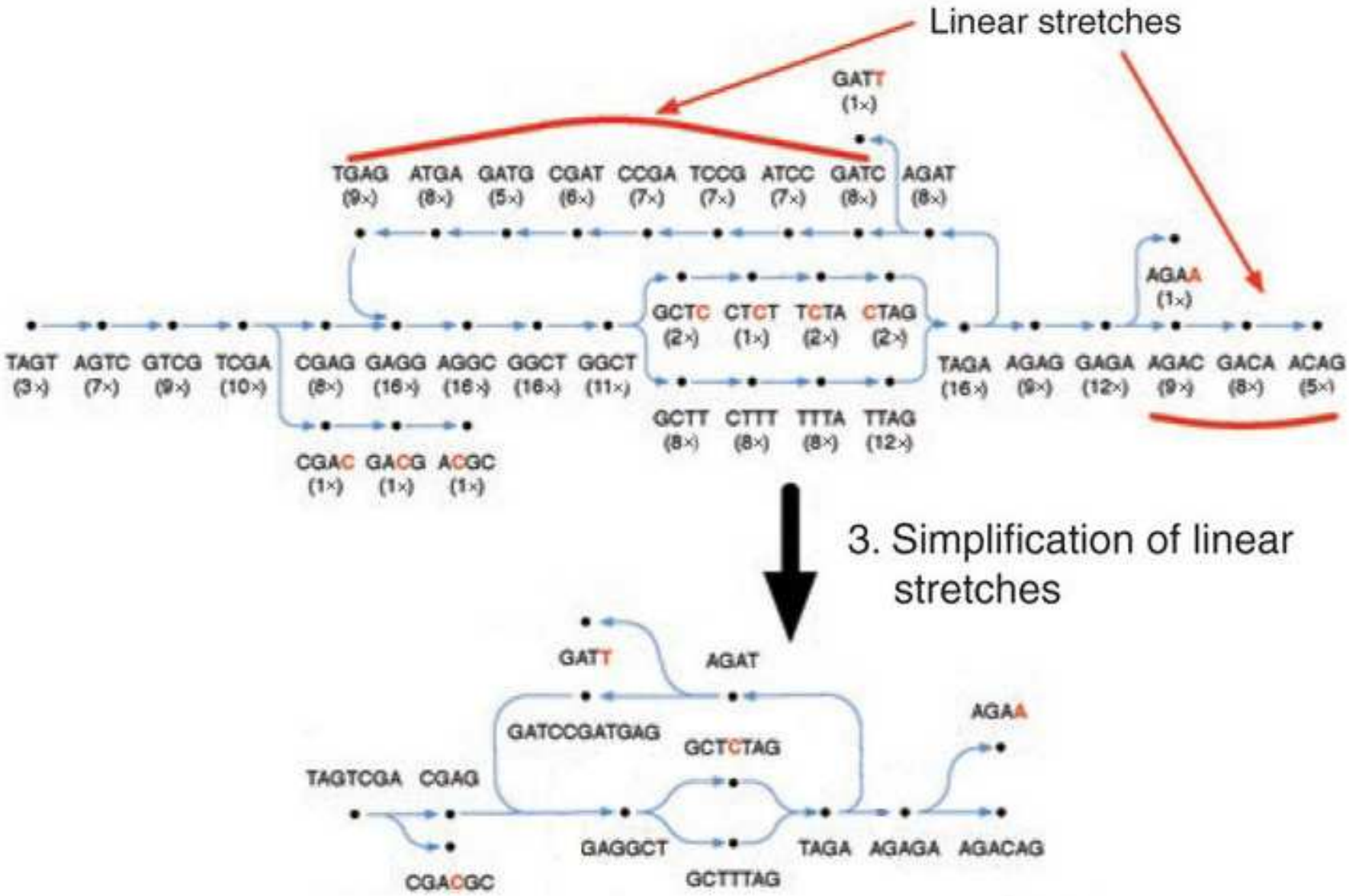


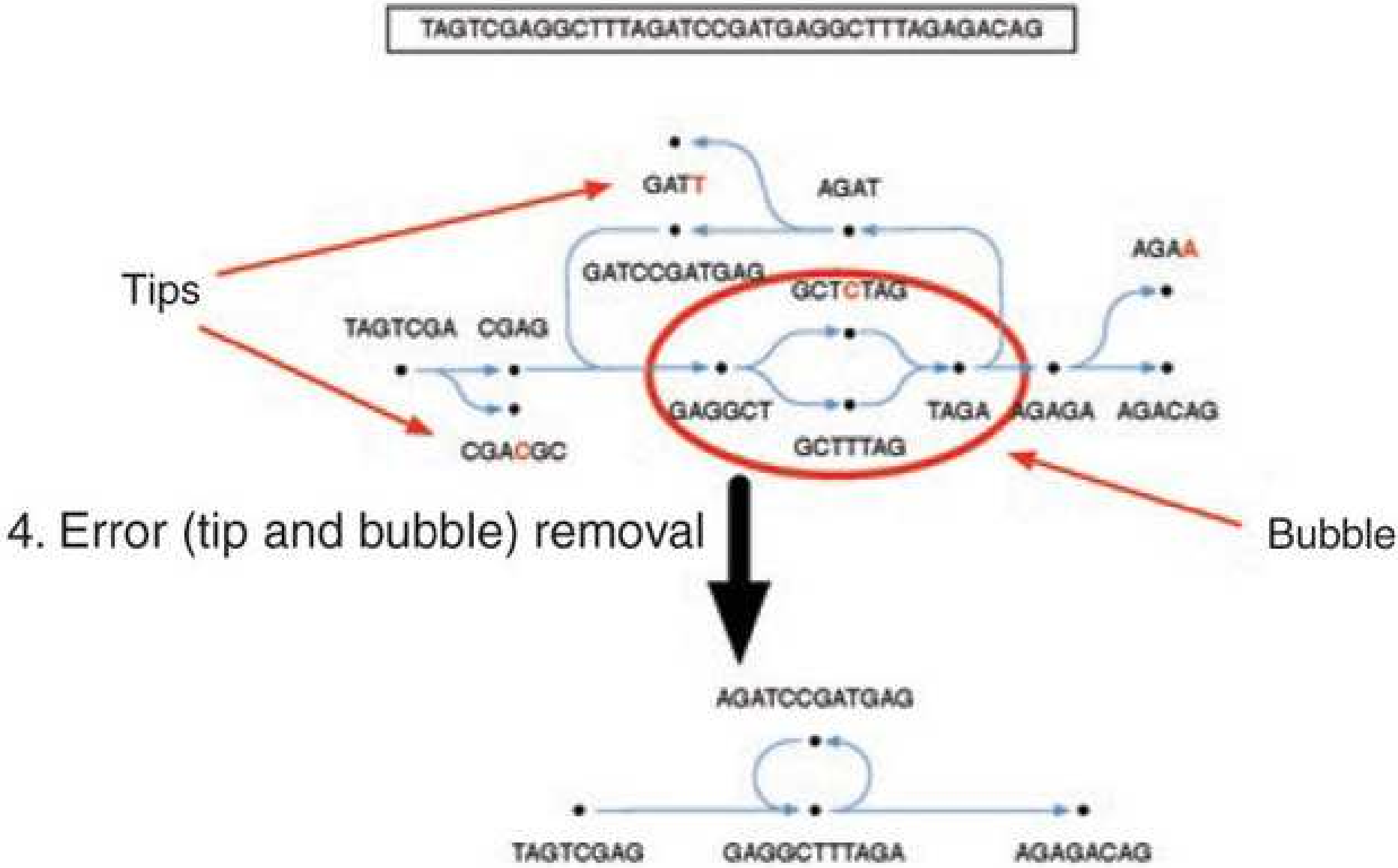
Ako použiť de Bruijnov graf na zostavovanie?



- V ideálnom prípade výsledok je **Eulerovský ťah:** "cesta" po grafe, ktorá prechádza každou hranou práve raz
- **Jednoducho riešiteľný problém v čase** $O(n+m)$
  Vieme overiť túto podmienku, aj nájsť ťah.
- Ak graf **nemá jednoznačný Eulerovský ťah**, rozseknúť na viacero jednoznačných ťahov.


TAGTCGAGGCTTTAGATCCGATGAGGCTTTAGAGACAG
1. Sequencing
(for example, Solexa or 454)
AGTCGAG CTTTAGA CGATGAG CTTTAGA
GTCGGG TTAGATC ATGAGGC GAGACAG
GAGGCTC ATCCGAT AGGCTTT GAGACAG
AGTCGAG TAGATCC ATGAGGC TAGAGAA
TAGTCGA CTTTAGA CCGATGA TTAGAGA
CGAGGCT AGATCCG TGAGGCT AGAGACA
TAGTCGA GCTTTAG TCCGATG GCTCTAG
TCGACGC GATCCGA GAGGCTT AGAGACA
TAGTCGA TTAGATC GATGAGG TTTAGAG
GTCGAGG TCTAGAT ATGAGGC TAGAGAC
AGGCTTT ATCCGAT AGGCTTT GAGACAG
AGTCGAG TTAGATT ATGAGGC AGAGACA
GGCTTTA TCCGATG TTTAGAG
CGAGGCT TAGATCC TGAGGCT GAGACAG
AGTCGAG TTTAGATC ATGAGGC TTAGAGA
GAGGCTT GATCCGA GAGGCTT GAGACAG
2. Hashing
GATT (1×)
TGAG (9×) ATGA (8×) GATG (5×) CGAT (6×) CCGA (7×) TCCG (7×) ATCC (7×) GATC (8×) AGAT (8×)
AGAA (1×)
GCTC (2×) CTCT (1×) TCTA (2×) CTAG (2×)
TAGT (3×) AGTC (7×) GTCG (9×) TCGA (10×) CGAG (8×) GAGG (16×) AGGC (16×) GGCT (16×) GGCT (11×)
TAGA (16×) AGAG (9×) GAGA (12×) AGAC (9×) GACA (8×) ACAG (5×)
GCTT (8×) CTTT (8×) TTTA (8×) TTAG (12×)
CGAC (1×) GACG (1×) ACGC (1×)


Linear stretches
GATT (1×)
TGAG (9×) ATGA (8×) GATG (5×) CGAT (6×) CCGA (7×) TCCG (7×) ATCC (7×) GATC (8×) AGAT (8×)
AGAA (1×)
GCTC (2×) CTCT (1×) TCTA (2×) CTAG (2×)
GCTT (8×) CTTT (8×) TTTA (8×) TTAG (12×)
TAGT (3×) AGTC (7×) GTCG (9×) TCGA (10×) CGAG (8×) GAGG (16×) AGGC (16×) GGCT (16×) GGCT (11×)
TAGA (16×) AGAG (9×) GAGA (12×) AGAC (9×) GACA (8×) ACAG (5×)
CGAC (1×) GACG (1×) ACGC (1×)
3. Simplification of linear stretches
GATT AGAT
AGAA
GATCCGATGAG
GCTCTAG
TAGTCGA CGAG
GAGGCT
TAGA AGAGA AGACAG
CGACGC
GCTTTAG


TAGTCGAGGCTTTAGATCCGATGAGGCTTTAGAGACAG
GATT
AGAT
AGAA
GATCCGATGAG
GCTCTAG
Tips
TAGTCGA
CGAG
GAGGCT
TAGA
AGAGA
AGACAG
CGACGC
GCTTTAG
4. Error (tip and bubble) removal
Bubble
AGATCCGATGAG
TAGTCGAG
GAGGCTTTAGA
AGAGACAG

Použitie NGS: Sekvenovanie nových genómov

- Cieľ: Osekvenovanie veľkej časti žijúcich druhov
- Mnohé genómy s nízkym pokrytím, veľa krátkych kontigov
- Rekonštrukcia genómov predkov
- Nájdenie všetkých funkčných elementov genómov
- Štúdium evolúcie rôznych funkcií
- Prispôsobenie sa rôznym prostrediam

Použitie NGS: Populačná genetika

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

Problémy:

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

Použitie NGS: Environmentálne sekvenovanie – Metagenomika

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

Problémy:

- Oddelenie segmentov patriacich do rôznych genómov

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

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

Problémy:

- Identifikácia miest zostrihu
- Identifikácia väzobných miest podľa hĺbky pokrytia

Zhrnutie

- Sekvenovanie genómu je zložitý proces, v ktorom hrá bioinformatika dôležitú úlohu
- V súčasnosti niekoľko nových technológií, nízka cena, krátke segmenty
- Problém zostavovania genómu, najkratšie spoločné nadslovo
- Overlap-Layout-Consensus
- Eulerovské ťahy v de Bruijnovom grafe
- V zostavenej sekvencii môžu byť chyby, medzery, viaceré kontigy

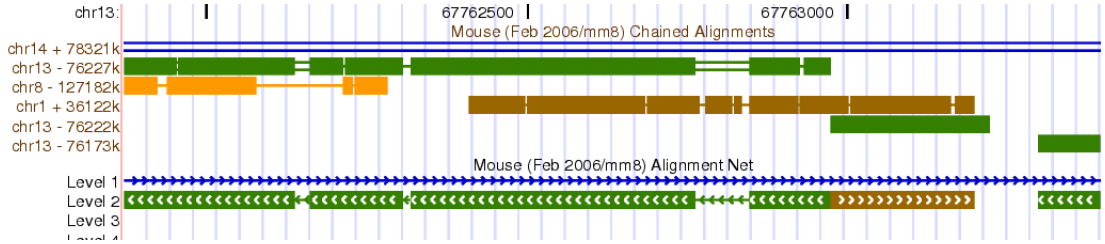
Oznamy

- Budúci týždeň budú iba cvičenia, nebude prednáška.
  Cvičenia pre biológov budú v čase prednášky, t.j. od 15:40
- Prvá domáca úloha: zadanie na budúci týždeň, čas na vypracovanie cca 2 týždne (detaily nabudúce)
- Výber článku na journal club formulárom na stránke do 22.10.
- Ďalšie info http://compbio.fmph.uniba.sk/vyuka/mbi/

Zarovnávanie sekvencií (sequence alignment) 1/2

Broňa Brejová

8.10.2014



[Durbin et al., 1998, kapitola 2]

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
ggcaacctcgaggtgctccttttcaccctgcagtcc
aagatgcgggccaacaaccagaagccctacacaccc
aaagagggcaagatgatttcggacatcaacaaggcc
tgggagcgtctggagaaggccgagcacgaacgcgaa
ttggccctgcgcgaggagctcatccg...
```

Vstup: dve sekvencie

Problém: Lokálne zarovnávanie (local alignment)

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

```
tgatgccgaggatgtgttcgtcgagcatccggacga
gaagtccatcacctacgtggtcacctactatcacta
ctttagcaaactcaagcaggagacggtgcagggcat
aagcgtatcggtaaggtggtcggcattgccatggag
aacgacaaaatggtccacgactacgagaacttcaca
agcgatctgctcaagtggatcgaaacgaccatccag
tcgctgggcgagcgggagttcgaaaactcgctggcc
ggcgtccaagggcagttggcccagttctccaactac
cgcaccatcgagaagccgcccaagtttgtggaaaag
ggcaacctcgaggtgctccttttcaccctgcagtcc
aagatgcgggccaacaaccagaagccctacacaccc
aaagagggcaagatgatttcggacatcaacaaggcc
tgggagcgtctggagaaggccgagcacgaacgcgaa
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 má vyše 100 GB sekvencií.
  Napr. odkiaľ z genómu je daný EST?
- **Určovanie funkcie (napr. proteínu).**
  Podobné sekvencie často majú rovnakú/podobnú funkciu.
- **Štúdium evolúcie.**
  Hľadáme homológy, sekvencie, ktoré sa vyvinuli z toho istého spoločného predka.
  V ideálnom prípade medzery zodpovedajú inzerciám a deléciám, zarovnané bázy zachovaným bázam a substitúciám.
- **Hľadanie génov a iných funkčných prvkov.**
  Menia sa pomalšie ako ostatné sekvencie.

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. Chceme nastaviť tak, aby homológy mali vysoké skóre, náhodné zarovnania nízke.

Problém 1: globálne zarovnanie (global alignment)

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

Výstup: zarovnanie X a Y s najvyšším skóre.

Problém 2: lokálne zarovnanie (local alignment)

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

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 $x_1x_2\ldots x_i$ a $y_1y_2\ldots y_j$.

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 $x_i = y_j$ a sú zarovnané $A[i,j] = A[i-1,j-1] + 1$,
ak $x_i \neq y_j$ a sú zarovnané $A[i,j] = A[i-1,j-1] - 1$,
ak x_i je zarovnané s medzerou $A[i,j] = A[i-1,j] - 1$,
ak y_j je zarovnané s medzerou $A[i,j] = A[i,j-1] - 1$.

Dynamické programovanie pre globálne zarovnanie

Podproblém: $A[i, j]$: najvyššie skóre globálneho 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 a y_j sú zarovnané $A[i, j] = A[i-1, j-1] + s(x_i, y_j)$

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

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

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

Rekurencia:

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

Príklad globálneho zarovnania

CATGTCATA vs CAGTCCTAGA

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

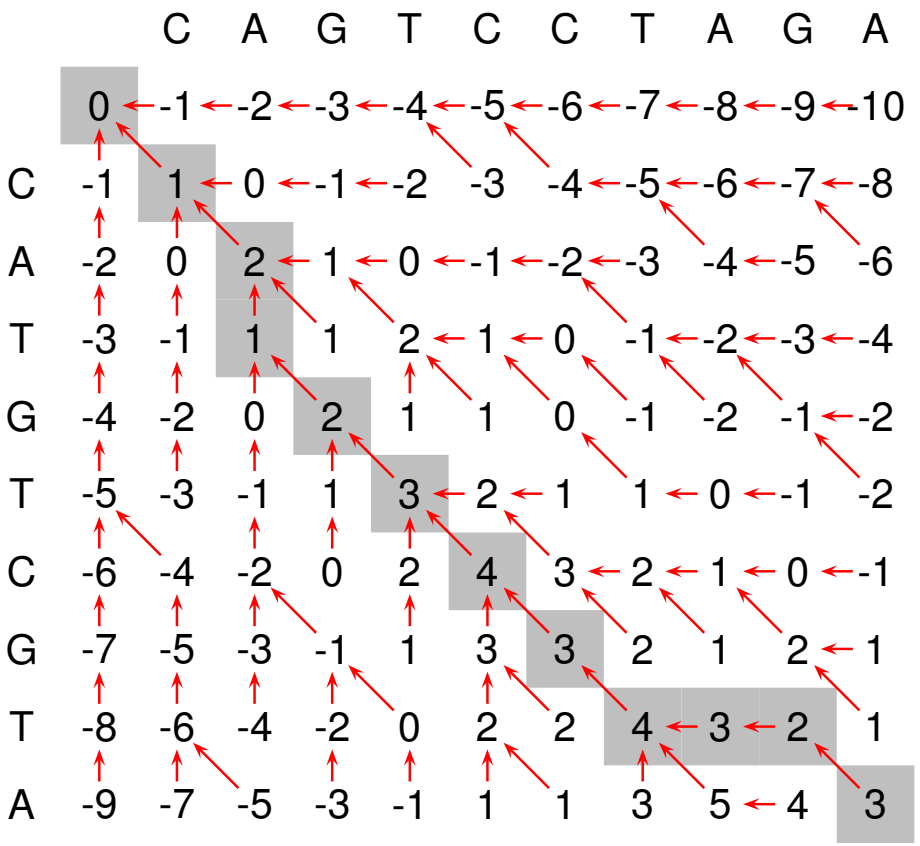
$$A[i,j] = \max \begin{cases} A[i-1,j-1] + s(x_i, y_j), \\ A[i-1,j] - 1, \\ A[i,j-1] - 1 \end{cases}$$

Príklad globálneho zarovnania

CATGTCATA vs CAGTCCTAGA

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

Ako získať zarovnanie?




```
CA-GTCCTAGA
CATGTCAT--A
```

Dynamické programovanie pre lokálne zarovnanie (Smith, Waterman 1981)

Podproblém: $A[i,j]$: najvyššie skóre lokálneho 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 $A[i,j] = A[i-1,j] - 1$

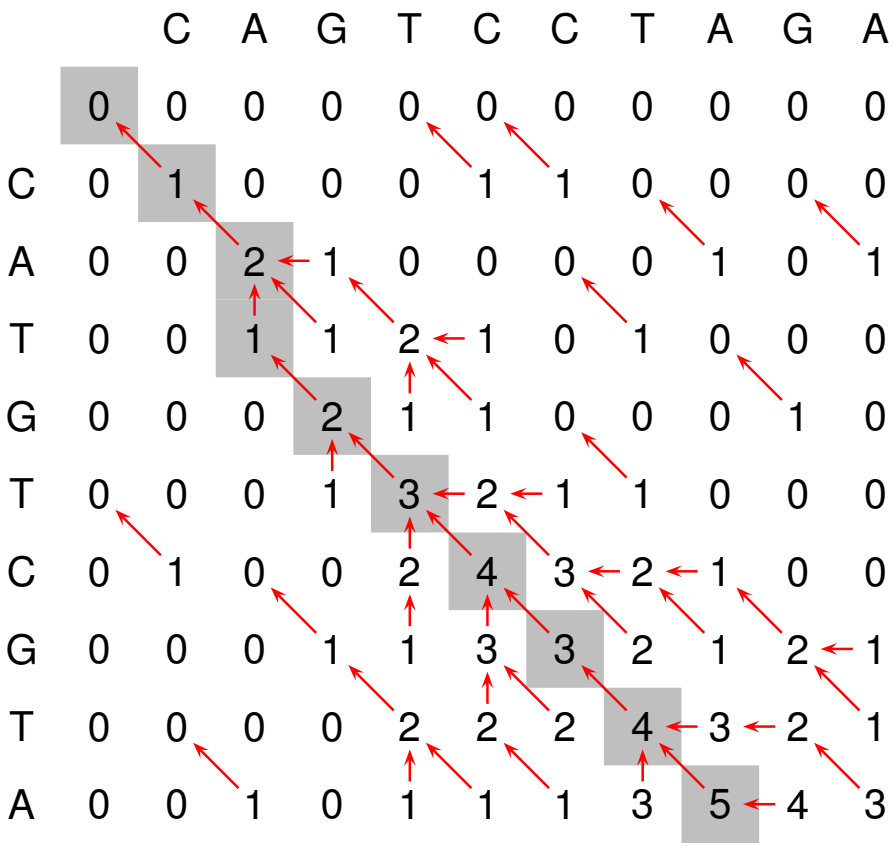
ak y_j je zarovnané s medzerou $A[i,j] = A[i,j-1] - 1$

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




```
CA-GTCCTA
CATGTCATA
```

Zložitejšie skórovanie

Problémy $+1, -1$ skórovania:

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

Úloha skórovacej schémy:

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

Zložitejšie skórovanie: prvý pokus

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

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

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

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

$a + b + c = 1$

Pravdepodobnosť zarovnania A:

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

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

Ktoré je pravdepodobnejšie?

```
CACA
|  |
CCAA
```

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

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

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

Zložitejšie skórovanie: prvý pokus

Zlogaritmujeme: násobenie sa zmení na sčítavanie

môžeme použiť S.-W. alebo N.-W. dyn. prog. algoritmy

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

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

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

Nevýhody takejto schémy:

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

Zložitejšie skórovanie: dva pravdepodobnostné modely

(Pre jednoduchosť teraz neuvažujme medzery)

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

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

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

Model R: Sekvencie X a Y nijako spolu nesúvisia

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

$p(x_i)$: pravdepodobnosť výskytu bázy x_i

Porovnanie modelov H a R: "log likelihood"

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

Zložitejšie skórovanie: dva pravdepodobnostné modely

Porovnanie modelov H a R: "log likelihood"

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

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

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

BLOSUM62 skórovacia matica pre proteíny

BLOcks of aminoacid **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
- Needleman-Wunschov a Smith-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?

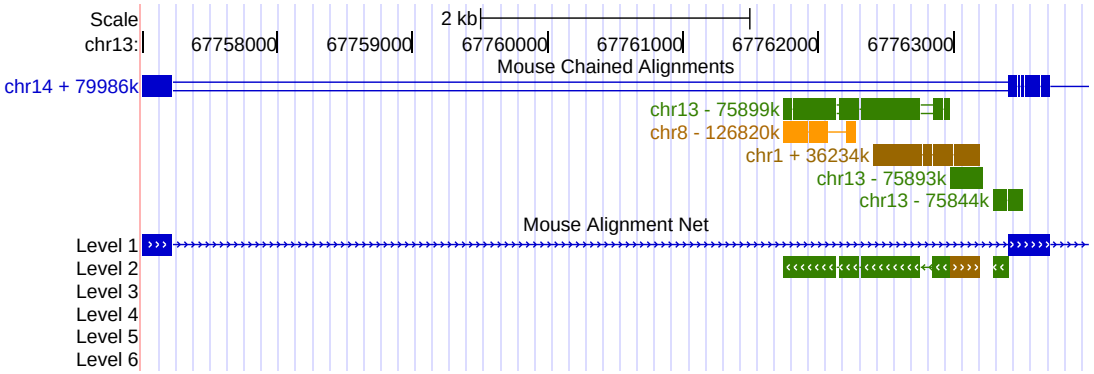
Oznamy

- Na konci prednášky rozdelenie do skupín na journal club
- Domácu úlohu 1 odovzdávajte do 5.11. 9:00
- Budúci týždeň predmet podľa normálneho rozvrhu

Zarovnávanie sekvencií 2/2
(sequence alignment)

Tomáš Vinař

23.10.2014



Zhrnutie z minulej prednášky

- **Problém globálneho a lokálneho zarovnania**

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

  Výstup: zarovnanie X a Y s najvyšším skóre

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

- **Správny algoritmus na riešenie**

  dynamické programovanie

- **Realistické skórovacie schémy**

Máme správny algoritmus na zarovnávanie, čo viac nám chýba?

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

Koľko je to času v skutočnosti?

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

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

Potrebujeme efektívnejší algoritmus,

najmä ak chceme pracovať s celými genómami

Heuristické lokálne zarovnávanie

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

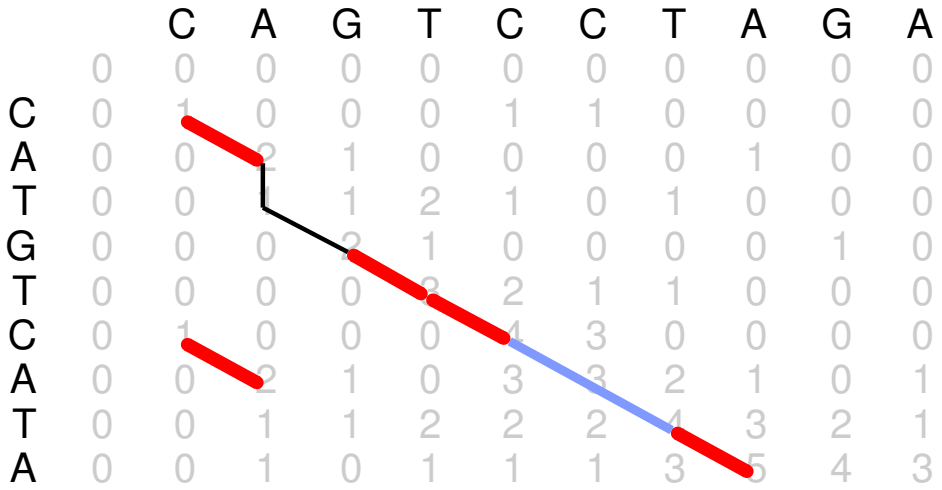
Napríklad: BLASTN [Altschul et al., 1990],
FASTA [Pearson and Lipman, 1988]

- 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).

Heuristické lokálne zarovnávanie

Príklad: začíname z jadier dĺžky $w = 2$

(V praxi sa používa $w = 10$ a viac.)



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

Ako nájdeme zhodujúce sa úseky?

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

Príklad: CAGTCCTAGA vs CATGTCATA

Slovník:

```
AG 2, 8
CA 1
CC 5
CT 6
GA 9
GT 3
TA 7
TC 4
```

Hľadaj:

```
CA → 1
AT → -
TG → -
GT → 3
TC → 4
CA → 1
AT → -
TA → 7
```

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 zhôd?

Dva nukleotidy sa zhodujú s pravdepodobnosťou $1/4$.
w zhôd za sebou s pravdepodobnosťou 4^{-w}.
Stredná hodnota počtu zhôd $nm4^{-w}$.
Zvýšenie w o 1 zníži počet zhôd cca 4 krát.

Senzitivita heuristického algoritmu

Algoritmus:

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

Nenájdené zarovnania: vysoké skóre, ale nemajú jadro dĺžky w

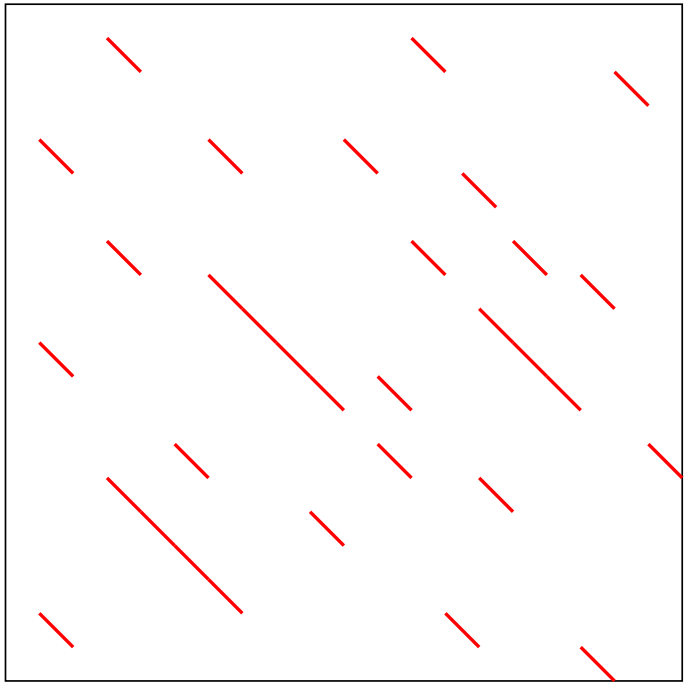
Príklad: `CA-GTCCTA` nenájdeme pre $w \geq 4$
`CATGTCATA`

Senzitivita: aká časť skutočných zarovnaní obsahuje zhodu dĺžky w

Rýchlosť vs. senzitivita

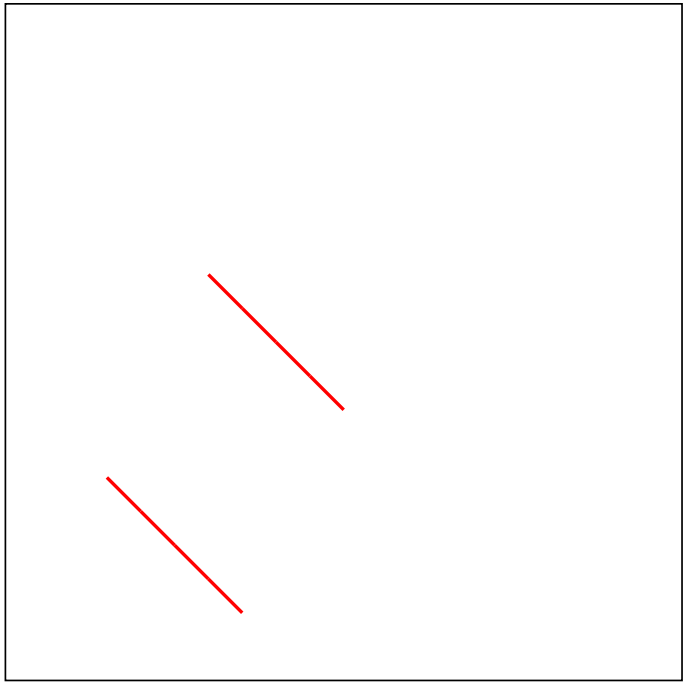
Malé w

veľa náhodných zhôd, pomalé



Veľké w

nenájdeme veľa zarovnaní



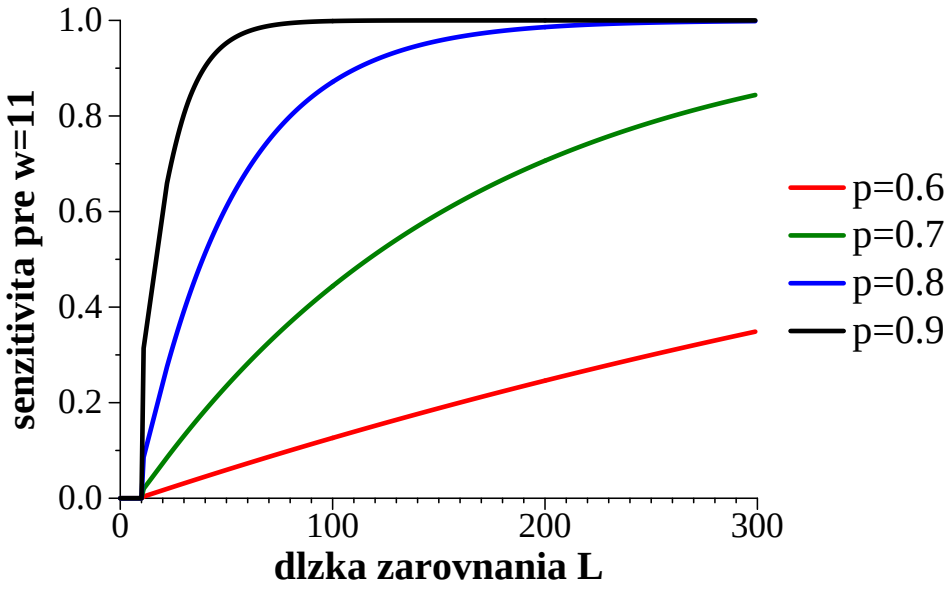
Senzitivita heuristického algoritmu

Odhad senzitivity:

Predpokladáme zarovnanie bez medzier, dĺžky L

Každá pozícia je zhoda s pravdepodobnosťou p

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



(človek-myš: $p \approx 0.7$)

BLAST algoritmus pre proteíny

BLOSUM62 skórovacia matica pre proteíny

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

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

Áno:

```
N I R
N L R
```

6+2+5=13

Nie:

```
A I L
A I L
```

4+4+4=12

Príklady programov

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

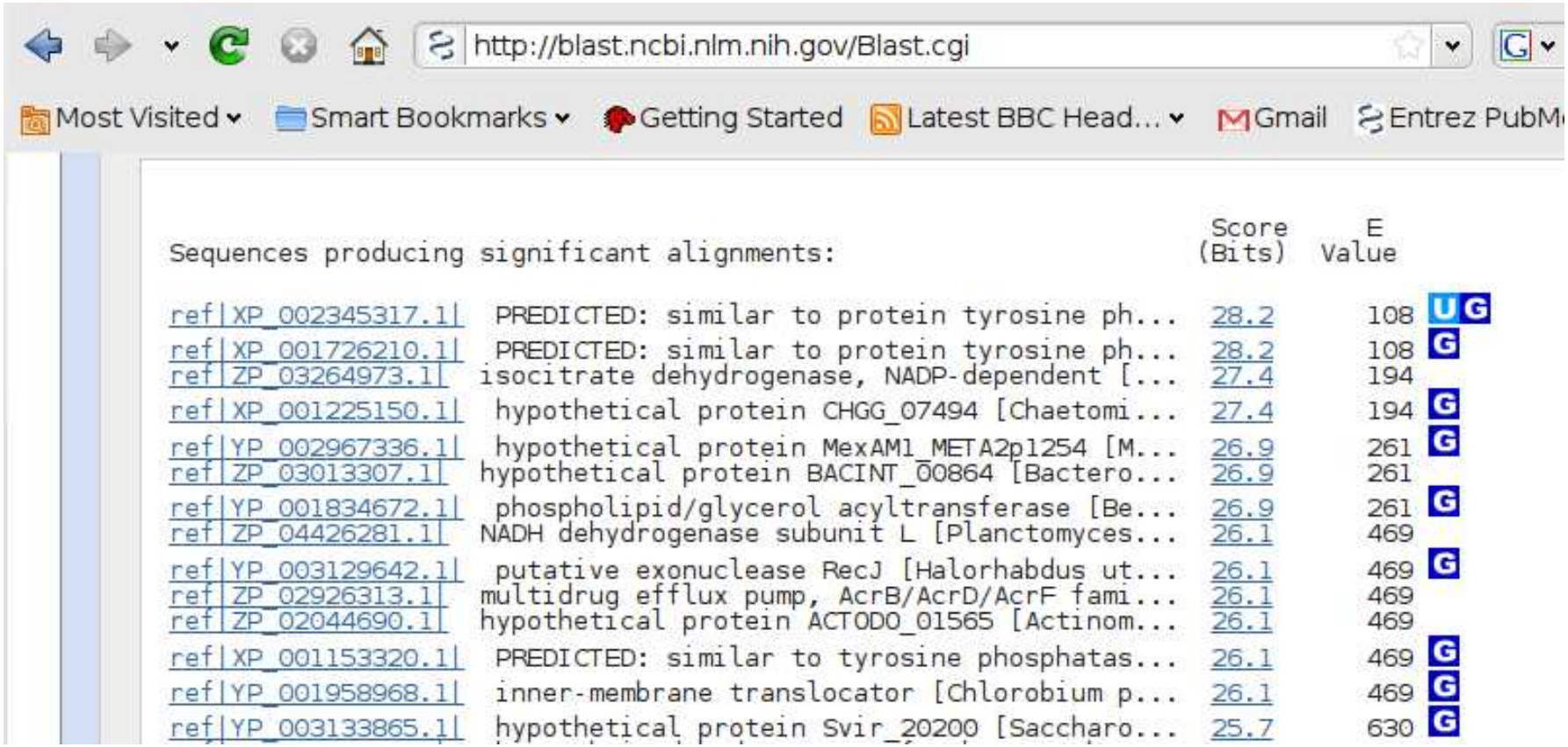
UCSC Blat: veľmi rýchle vyhľadávanie veľmi podobných sekvencií,
napr. EST ku genómu [Kent, 2002].

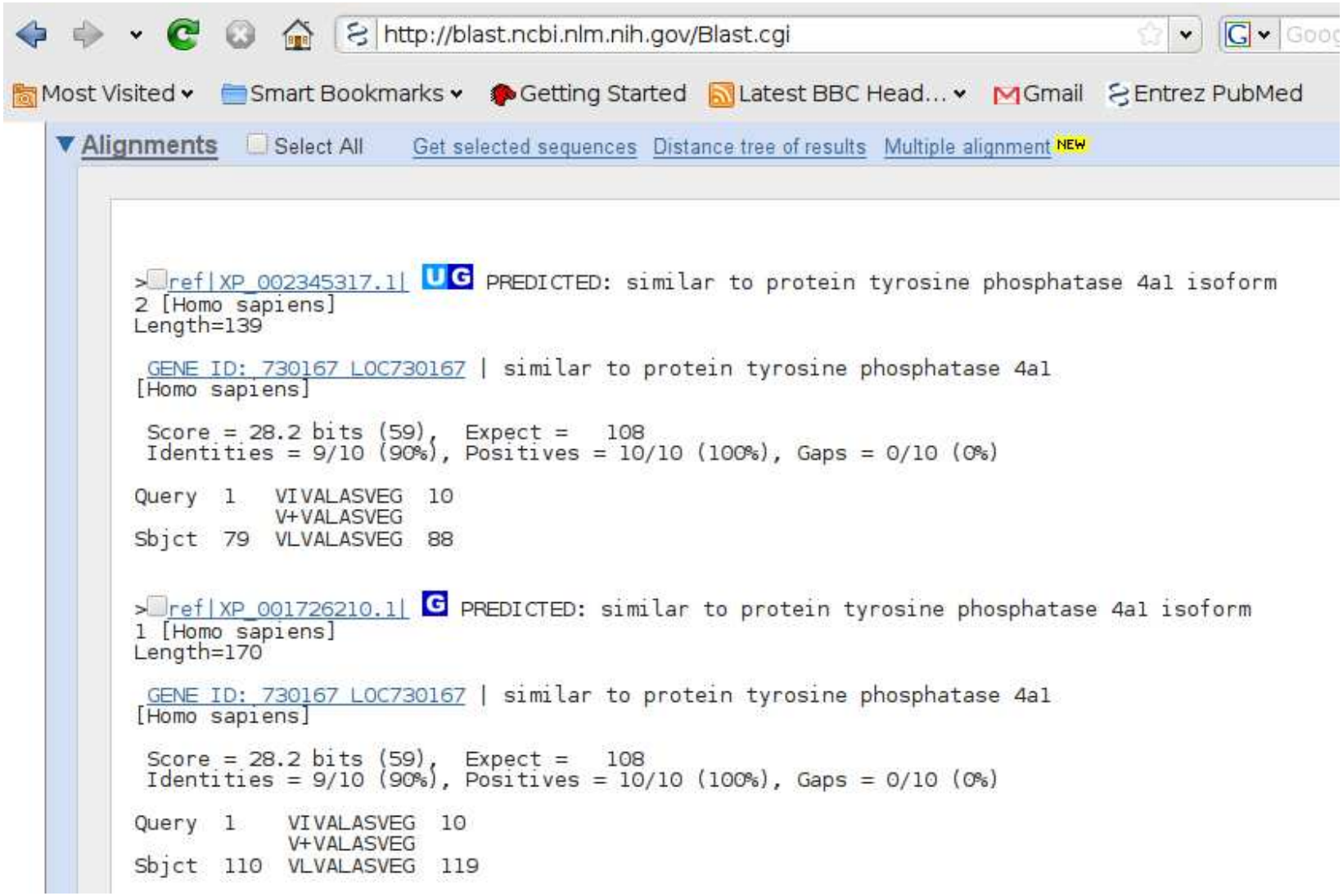
- používa veľké w
- vie rozdeliť EST na exóny

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

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

$\Rightarrow$ nájde vzdialenejšie homológy.


http://blast.ncbi.nlm.nih.gov/Blast.cgi
Most Visited
Smart Bookmarks
Getting Started
Latest BBC Head...
Gmail
Entrez PubM
Sequences producing significant alignments:
Score (Bits)
E Value
ref|XP_002345317.1| PREDICTED: similar to protein tyrosine ph... 28.2 108 U G
ref|XP_001726210.1| PREDICTED: similar to protein tyrosine ph... 28.2 108 G
ref|ZP_03264973.1| isocitrate dehydrogenase, NADP-dependent [... 27.4 194
ref|XP_001225150.1| hypothetical protein CHGG_07494 [Chaetomi... 27.4 194 G
ref|YP_002967336.1| hypothetical protein MexAM1_META2p1254 [M... 26.9 261 G
ref|ZP_03013307.1| hypothetical protein BACINT_00864 [Bactero... 26.9 261
ref|YP_001834672.1| phospholipid/glycerol acyltransferase [Be... 26.9 261 G
ref|ZP_04426281.1| NADH dehydrogenase subunit L [Planctomyces... 26.1 469
ref|YP_003129642.1| putative exonuclease RecJ [Halorhabdus ut... 26.1 469 G
ref|ZP_02926313.1| multidrug efflux pump, AcrB/AcrD/AcrF fami... 26.1 469
ref|ZP_02044690.1| hypothetical protein ACTODO_01565 [Actinom... 26.1 469
ref|XP_001153320.1| PREDICTED: similar to tyrosine phosphatas... 26.1 469 G
ref|YP_001958968.1| inner-membrane translocator [Chlorobium p... 26.1 469 G
ref|YP_003133865.1| hypothetical protein Svir_20200 [Saccharo... 25.7 630 G


http://blast.ncbi.nlm.nih.gov/Blast.cgi
Most Visited
Smart Bookmarks
Getting Started
Latest BBC Head...
Gmail
Entrez PubMed
Alignments
Select All
Get selected sequences
Distance tree of results
Multiple alignment
NEW
>ref|XP_002345317.1| PREDICTED: similar to protein tyrosine phosphatase 4a1 isoform 2 [Homo sapiens]
Length=139
GENE ID: 730167 LOC730167 | similar to protein tyrosine phosphatase 4a1 [Homo sapiens]
Score = 28.2 bits (59), Expect = 108
Identities = 9/10 (90%), Positives = 10/10 (100%), Gaps = 0/10 (0%)
Query 1 VIVALASVEG 10
V+VALASVEG
Sbjct 79 VLVALASVEG 88
>ref|XP_001726210.1| PREDICTED: similar to protein tyrosine phosphatase 4a1 isoform 1 [Homo sapiens]
Length=170
GENE ID: 730167 LOC730167 | similar to protein tyrosine phosphatase 4a1 [Homo sapiens]
Score = 28.2 bits (59), Expect = 108
Identities = 9/10 (90%), Positives = 10/10 (100%), Gaps = 0/10 (0%)
Query 1 VIVALASVEG 10
V+VALASVEG
Sbjct 110 VLVALASVEG 119

Ako rozlíšiť, či ide o významné zarovnanie?

Zarovnanie so skóre S.
Dĺžka dotazu m. Veľkosť databázy n.

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

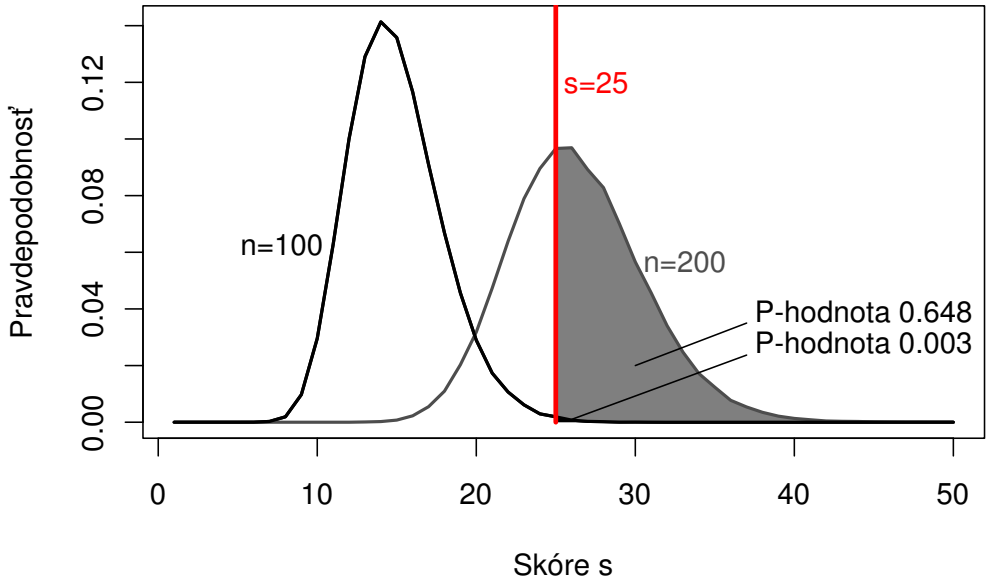
E-value: Očakávaný počet zarovnaní so skóre aspoň S nájdených pre náhodný dotaz dĺžky m v náhodnej databáze dĺžky n.

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

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

Výpočet P-hodnoty simuláciou

Medzi dvomi sekvenciami dĺžky n sme našli lokálne zarovnanie so skóre 25 (schéma +1/-1). Aká je jeho P-value?



Vygenerujeme 2 sekvencie dĺžky n, nájdeme najlepšie lok. zarovnanie
Ako často sme našli skóre aspoň 25?

V praxi je simulácia pomalá, existujú odhady rozdelenia

Genomické zarovnania (whole-genome alignments)

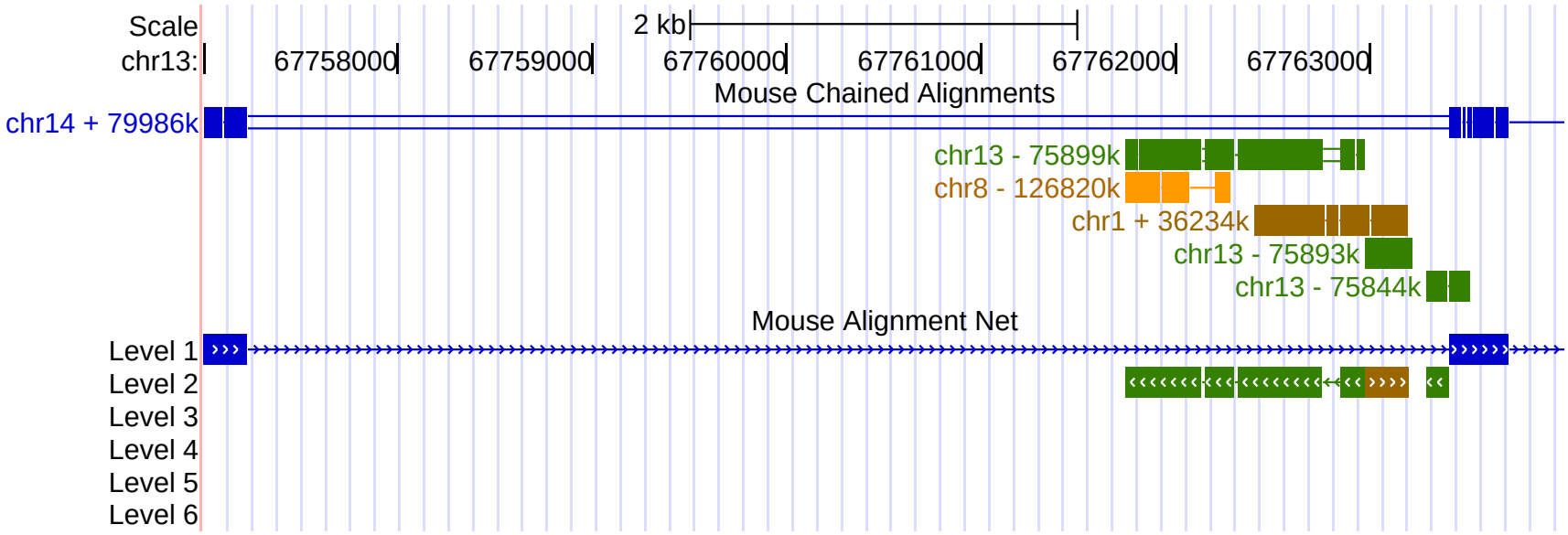
Ku každému úseku ľudského genómu nájsť zodpovedajúcu časť z myši, psa, sliepky, atď. (predpočítané v UCSC browseri) [Kent et al., 2003]

- Lokálne 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)

Genomické zarovnania (whole-genome alignments)

- Začni s lokálnymi zarovnaniami.
- Spoj ich do **reťazí (chains)**, kde povolujeme veľké medzery aj nezarovnané bloky. Vyžadujeme rovnaké poradie a orientáciu blokov v oboch genómoch.
- **Sieť (net)**: vyber reťaze s čo najväčším skóre tak, aby každá ľudská báza bola pokrytá najviac jedným blokom. Dovoľuje sa useknúť časti z reťaze.

  Hierarchická štruktúra: reťaze vo vnútri medzier iných reťazí.


Scale
chr13:
2 kb
67758000
67759000
67760000
67761000
67762000
67763000
Mouse Chained Alignments
chr14 + 79986k
chr13 - 75899k
chr8 - 126820k
chr1 + 36234k
chr13 - 75893k
chr13 - 75844k
Mouse Alignment Net
Level 1
Level 2
Level 3
Level 4
Level 5
Level 6

Viacnásobné zarovnanie, multiple sequence alignment

Zarovnaj viacero sekvencií.

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

Pre všeobecné k NP-ťažké.

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

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

Zhrnutie

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

Organizačné poznámky

- DÚ1 donesieme časom opravenú na prednášku, zadanie DÚ2 bude zverejnené budúci týždeň
- Pracujte na journal clube
- Myslite na možné témy projektu

Hľadanie génov

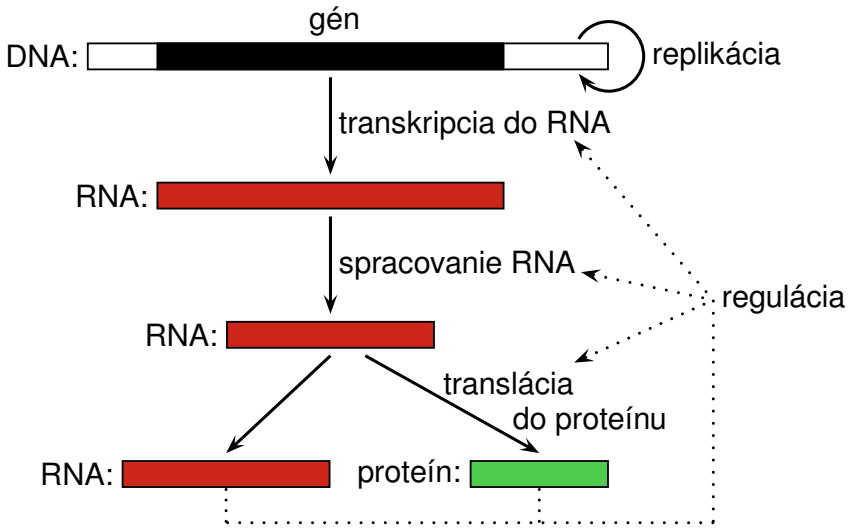
Broňa Brejová

6.11.2014

Čo s osekvenovanými genómami?

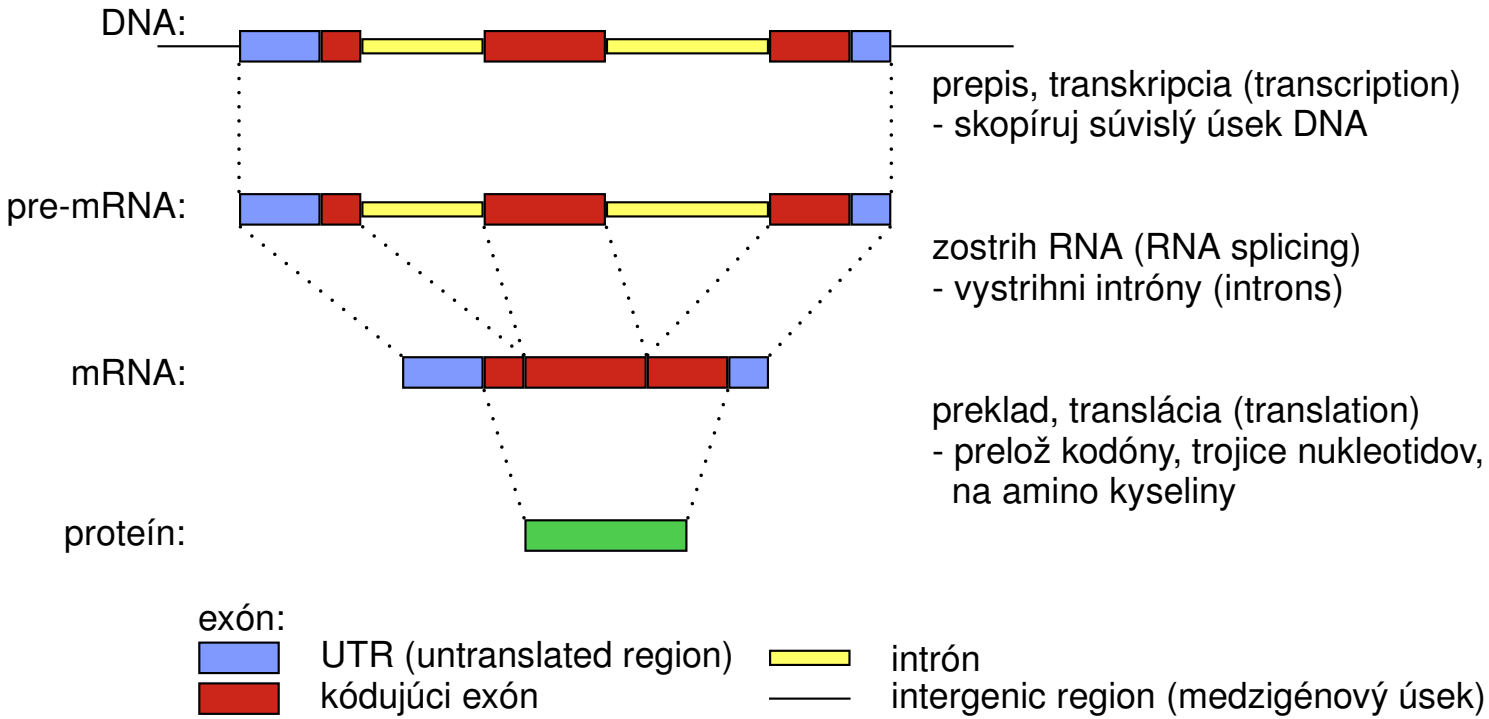
Chceme vedieť, čo genóm kóduje, hľadáme zaujímavé prvky, ako:

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

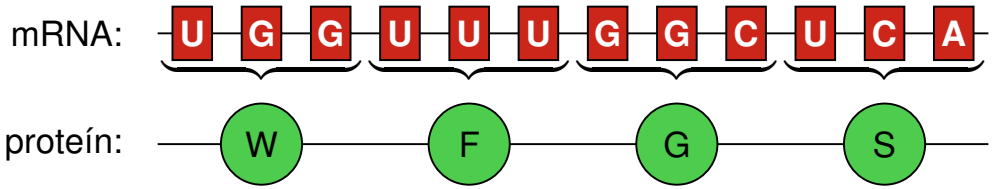


Štruktúra eukaryotických génov

Proces tvorby proteínov:



Translácia: tri bázy mRNA (kodón) → aminokyselina proteínu



Ľudský genóm

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

- repetitívne sekvencie
  - pokrývajú 49% genómu

Bioinformatický problém: hľadanie génov

Cieľ: nájsť všetky gény kódujúce proteíny v genóme.
Tým získame katalóg všetkých proteínov.

Zjednodušenia:

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

Bioinformatický problém: hľadanie génov

Vstup: DNA sekvencia

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

Bioinformatický problém: Hľadanie génov

Cieľ: označ každú bázu ako intrón/exón/medzigénový úsek

cggtgaaactgcacgattgttgctggcttaaagatagaccaatcagagtgtgtaacgtca

tatttagcgtcttctatcatccaatcactgcactttacacactataaatagagcagctca

tgggcgtatttgcgctagtgttgggtgttccgctgtgctgttttccgtcatggctcgca

ctaagcaaactgctcggaagtctactggtggcaaggcgccacgcaaacagttggccacta

aggcagcccgcaaaagcgctccggccaccggcggcgtgaaaaagccccaccgctaccggc

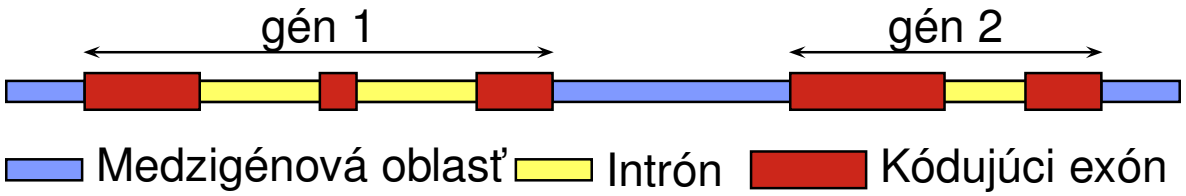
cgggcaccgtggctctgcgcgagatccgccgttatcagaagtccactgaactgcttattc

gtaaactacctttccagcgcctggtgcgcgagattgcgcaggactttaaaacagacctgc

gtttccagagctccgctgtgatggctctgcaggaggcgtgcgaggcctacttggtagggc

tatttgaggacactaacctgtgcgccatccacgccaagcgcgtcactatcatgcccaagg

acatccagctcgcccgccgcatccgcggagagagggcgtgattactgtggtctctctgac



Bioinformatický problém: hľadanie génov

Vstup: DNA sekvencia

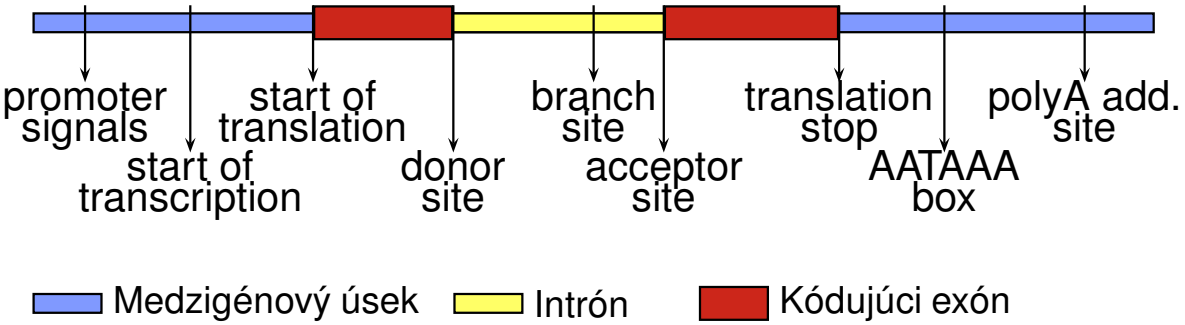
Cieľ: označ každú bázu ako intrón/exón/medzigénový úsek (anotácia)

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

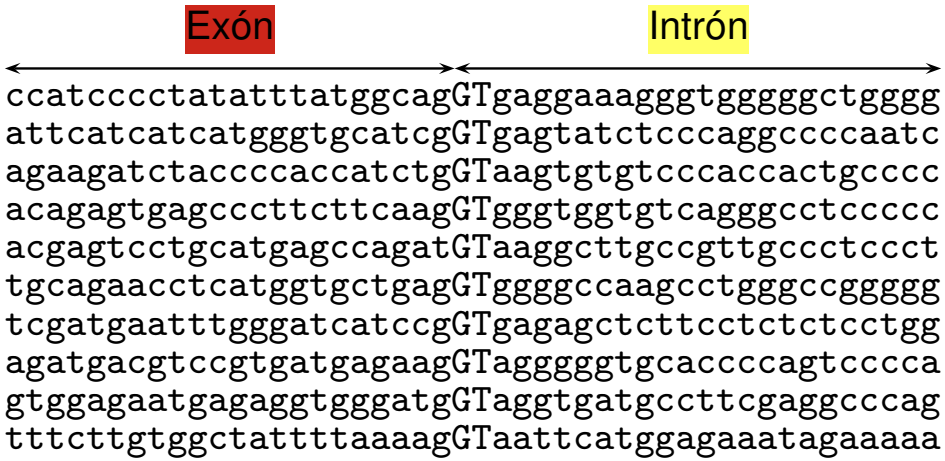
Ako spoznáme gény?

Signály na hraniciach exónov:

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



Príklad signálu: donor splice site



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. úsek		0.27	0.23	0.23	0.27

Bioinformatický problém: hľadanie génov

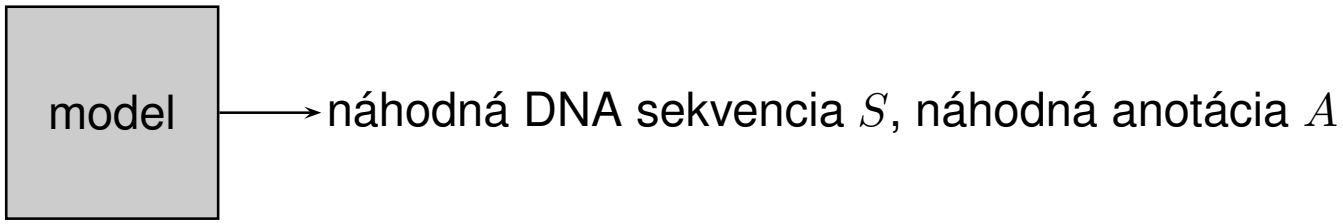
Vstup: DNA sekvencia

Cieľ: označ každú bázu ako intrón/exón/medzigénový úsek (anotácia)

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

Pravdepodobnostný model génov

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

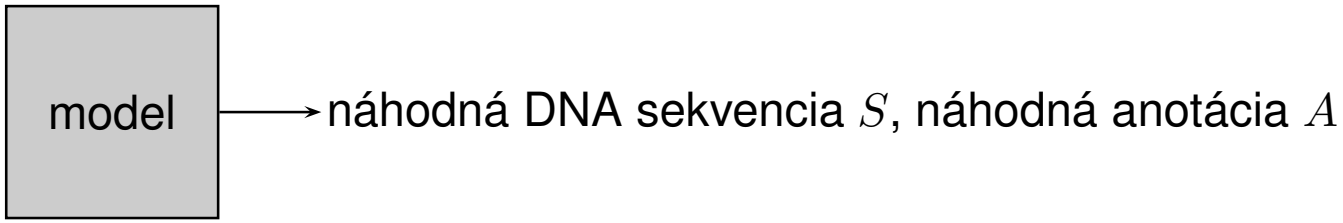


$\Pr(S, A)$ – pravdepodobnosť, že model vygeneruje pár (S, A).

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

Použitie: pre novú sekvenciu S 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

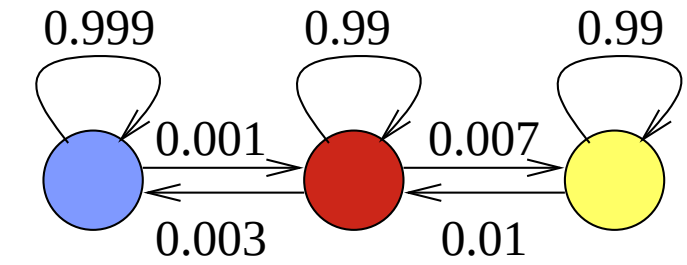
Tabuľka pravdepodobností pre 16 sekvencií, 9 anotácií (súčet 1)

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)

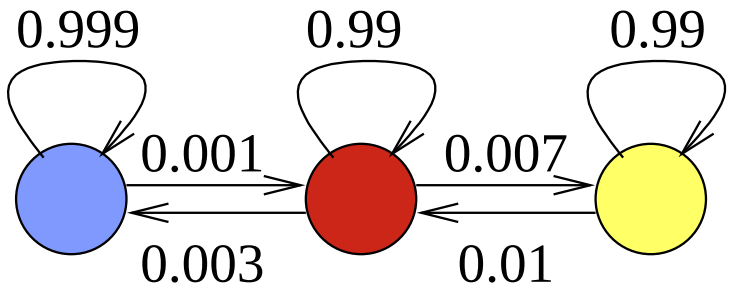
Spôsob, ako zadefinovať model pre dlhšie sekvencie.



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, medzigénový úsek
- 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

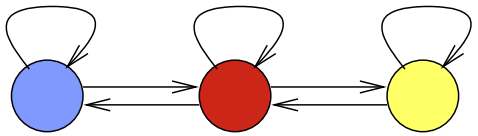
Predpokladajme, že model vždy začína v modrom stave.

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

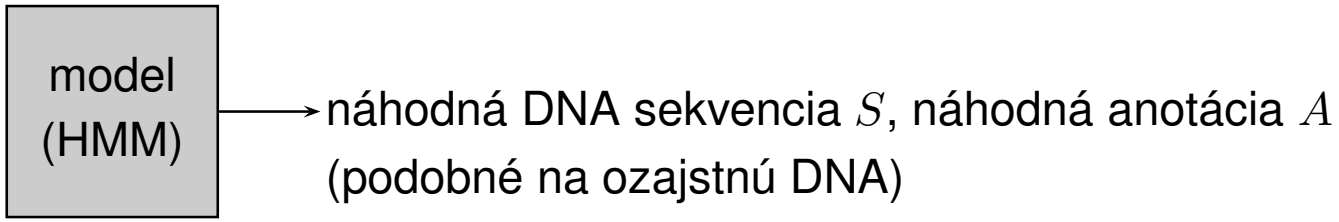
Počiatočná pravdepodobnosť $\pi(u) = \Pr(A_1 = u)$.

a			
	0.99	0.007	0.003
	0.01	0.99	0
	0.001	0	0.999

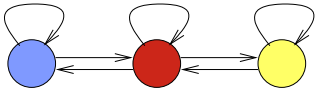
e	a	c	g	t
	0.24	0.27	0.28	0.21
	0.26	0.22	0.22	0.30
	0.27	0.23	0.23	0.27

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

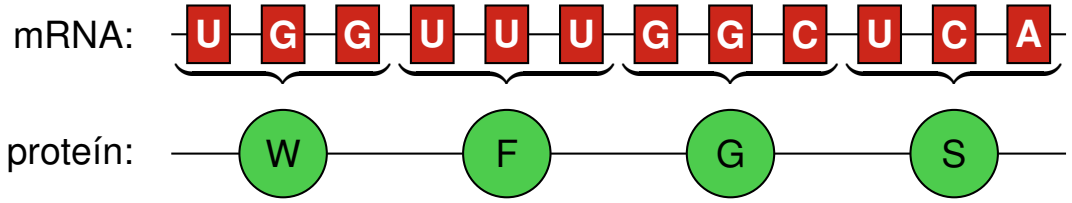


$\Pr(S, A)$ – pravdepodobnosť, že model vygeneruje pár (S, A).

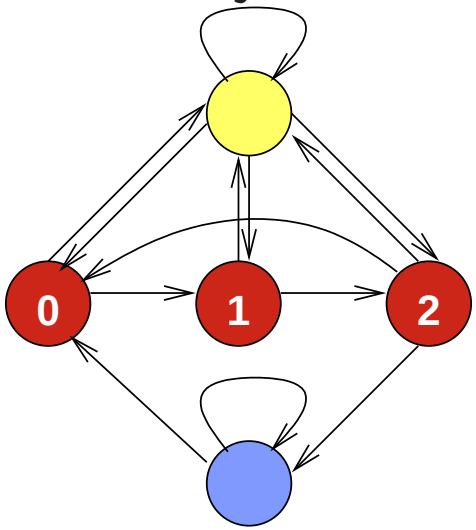
- **Určenie stavov a prechodov v modeli:** ručne, na základe poznatkov o štruktúre génu. 
- **Trénovanie parametrov:** emisné a prechodové pravdepodobnosti určíme na základe sekvencií so známymi génmi (**trénovacia množina**).
- **Použitie:** pre novú sekvenciu S nájdi najpravdepodobnejšiu anotáciu $A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

HMM na hľadanie génov: 3-periodické exóny

Kodón (trojica báz) $\rightarrow$ jedna aminokyselina



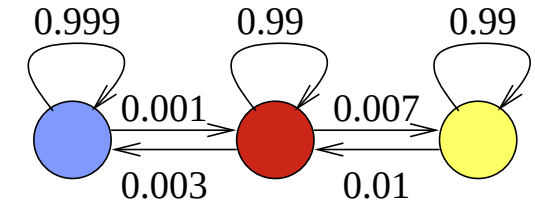
Namiesto jedného stavu pre exón použijeme tri stavy v cykle.

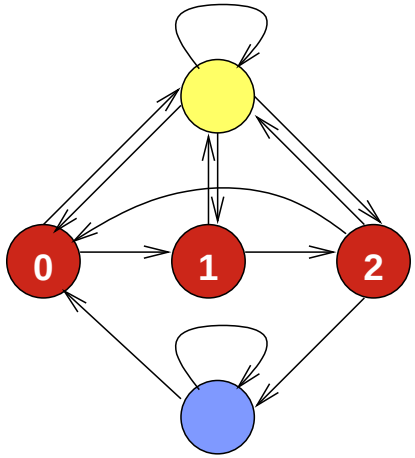


a	0	1	2		
0	0		0		0
1	0	0			0
2		0	0		
					0
		0	0	0	

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

Nové stavy majú odlišné emisné pravdepodobnosti



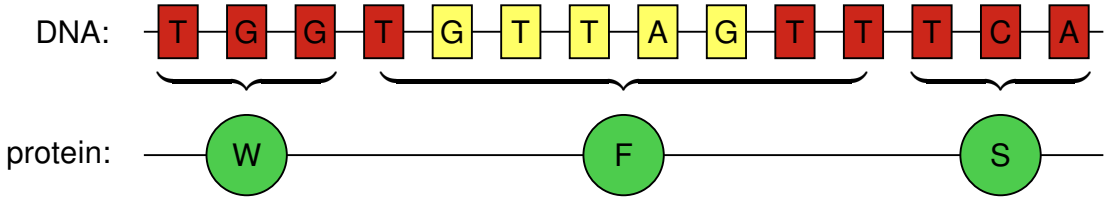


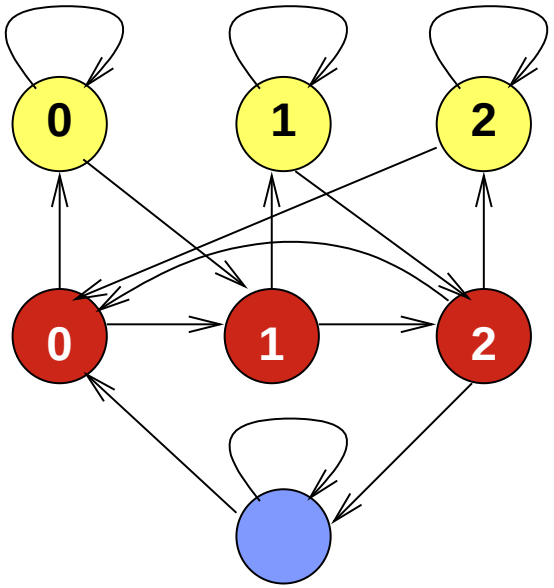
e	a	c	g	t
■	0.24	0.27	0.28	0.21
■	0.26	0.22	0.22	0.30
■	0.27	0.23	0.23	0.27

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

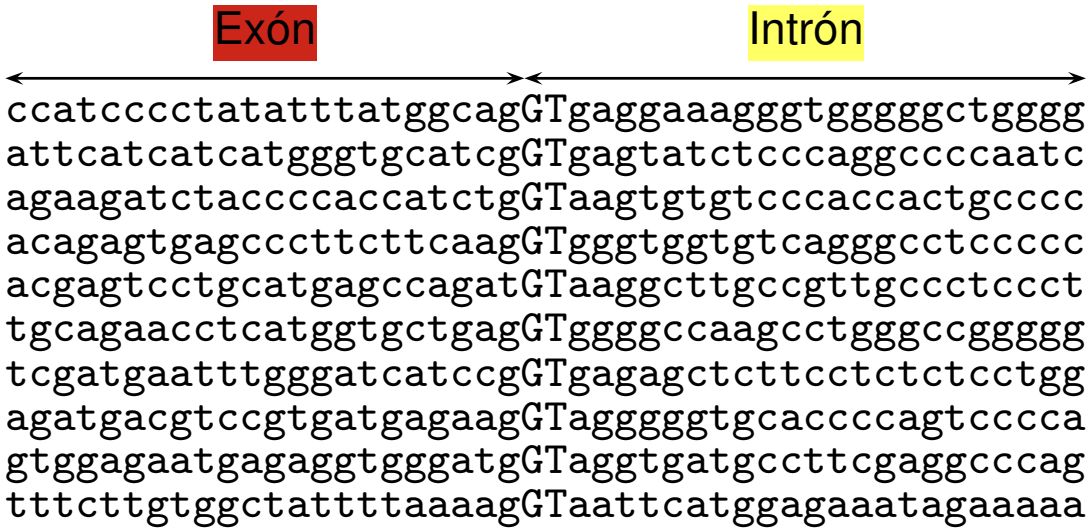
HMM na hľadanie génov: konzistentné kodóny

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

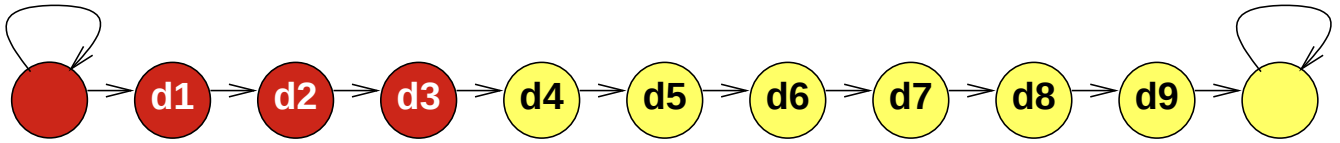




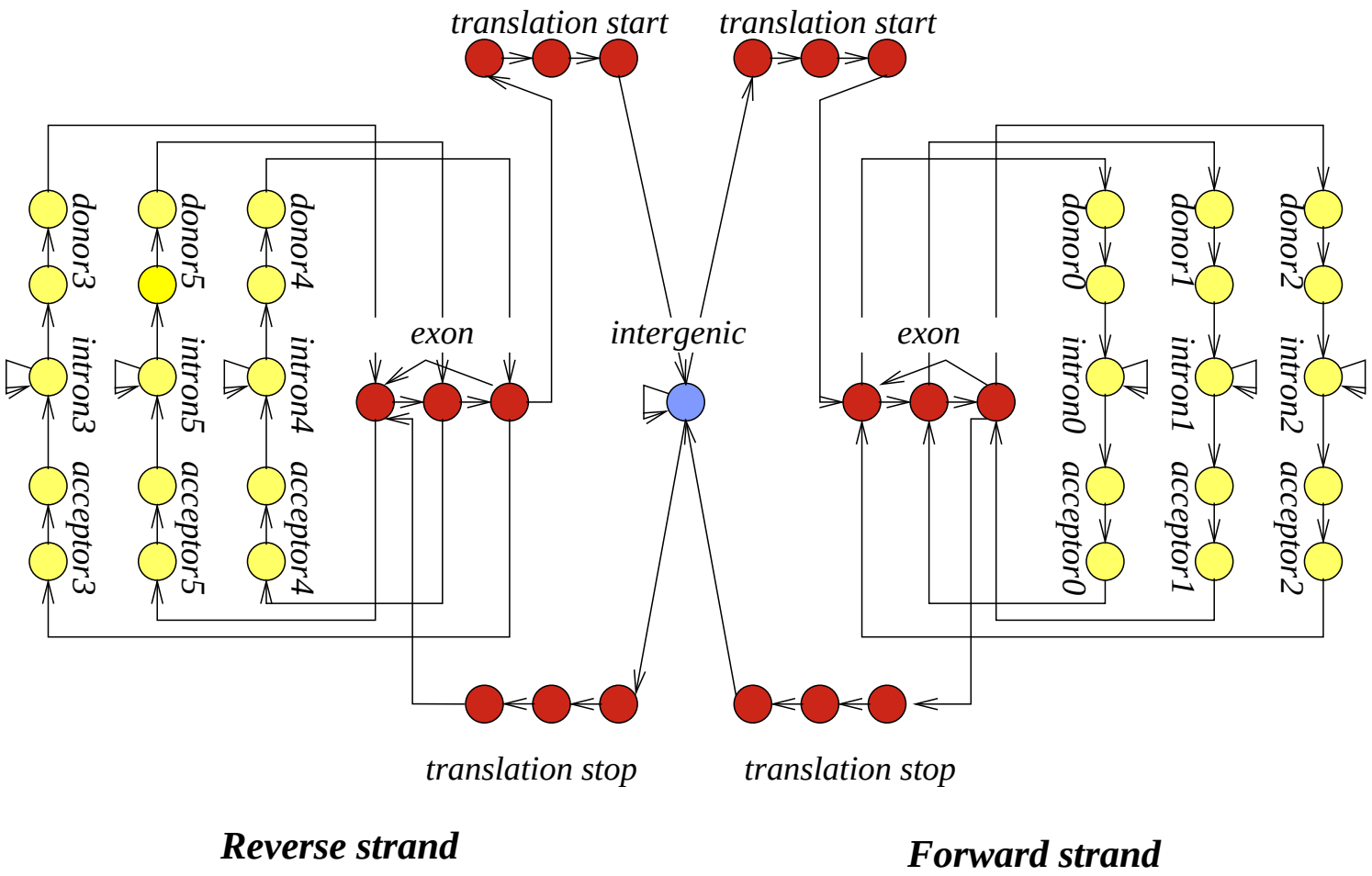
HMM na hľadanie génov: signály



Pridaj sériu stavov medzi exón a intrón:



HMM na hľadanie génov: celkový model



Stavy vyššších 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

...

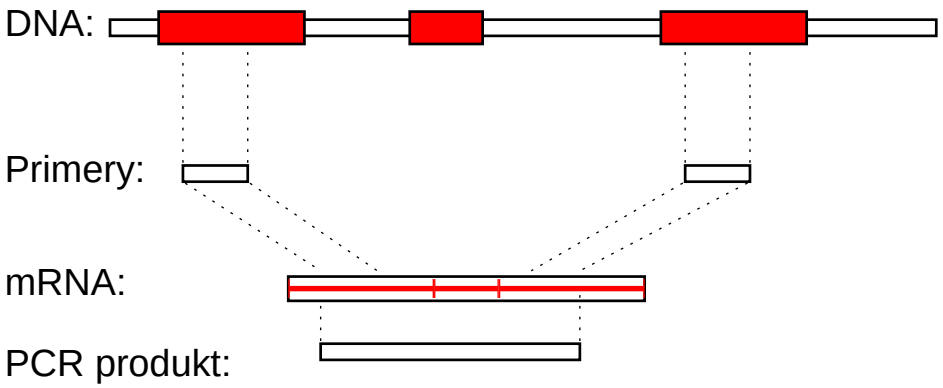
Na charakterizovanie exónov, intrónov atď používame rád 4-5.

Experimentálne overovanie génov

Overenie transkripcie a zostrihu

- Sekvenovanie EST, 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 detekovať 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 DNA sekvencie:

HMMGene [Krogh, 1997] (autor je priekopníkom HMM v bioinf.),
Genscan [Burge and Karlin, 1997] (po mnohé roky štandard),
GeneZilla [Majoros et al., 2004], ExonHunter [Brejová et al., 2005],
Augustus [Stanke and Waack, 2003] (novšie programy založené na zovšeobecnených HMM).
CONTRAST [Gross et al., 2007], CONRAD [DeCaprio et al., 2007] (najnovšia generácia založená na conditional random fields)

Prokaryotické genómy:

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

Vybrané programy na hľadanie génov

Porovnávaním viacerých sekvencií:

Twinscan [Korf et al., 2001]
(prvý úspešný gene finder s dvoma genómami),
Exoniphy [Siepel and Haussler, 2004]
(viacero genómov, nehľadá celé gény),
N-SCAN [Gross and Brent, 2006]
(rozšírenie Twinscanu na viacero genómov).

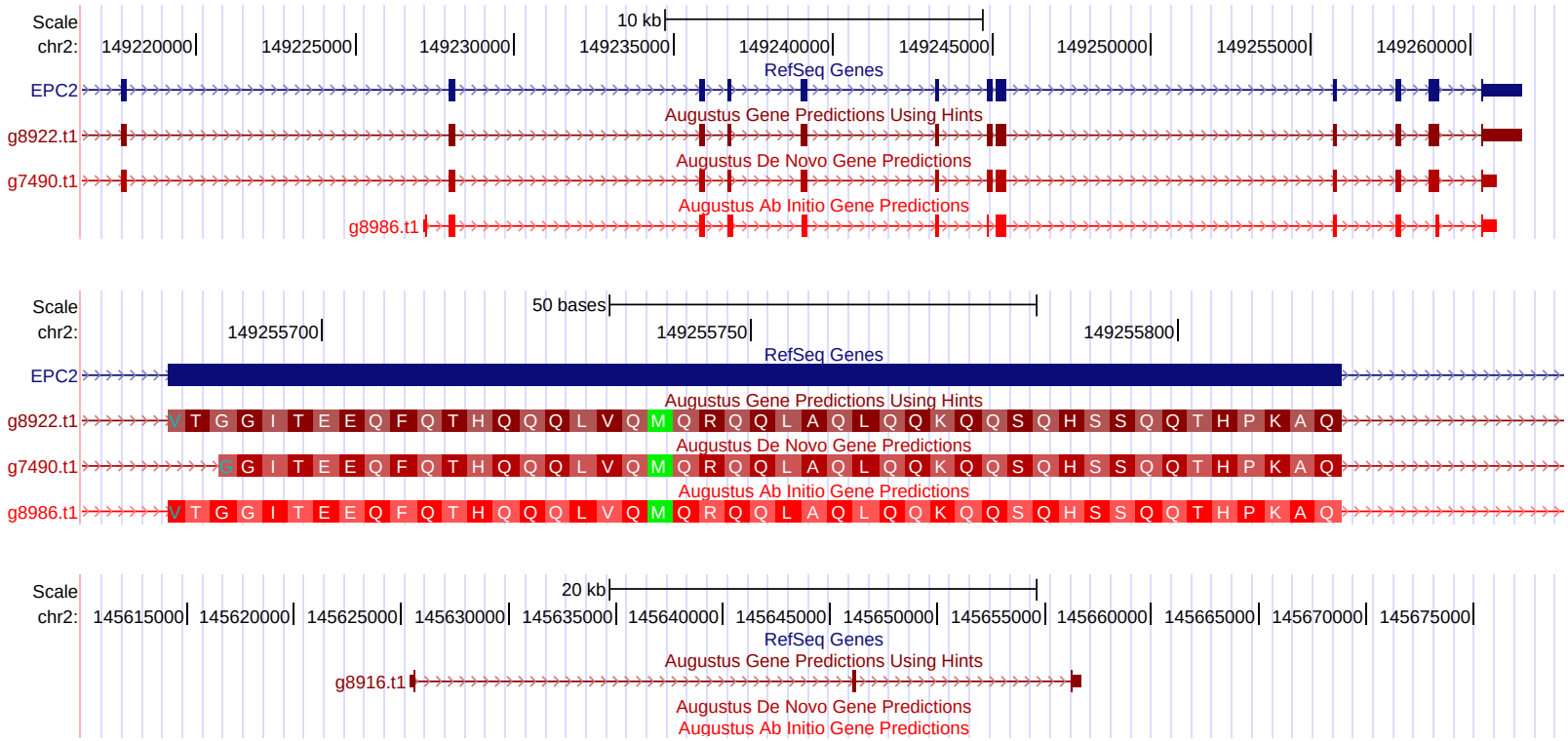
Iná informácia: (napr. EST-y, príbuzné proteíny a pod.)
ExonHunter [Brejová et al., 2005], Augustus [Stanke et al., 2006],
Jigsaw [Allen and Salzberg, 2005],
Fgenesh++ [Solovyev et al., 2006].

Obmedzenia hľadačov génov

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

- 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

Organizačné poznámky

- Domáca úloha 1: odovzdať do budúcej stredy 5.11. 9:00 pod dvere kancelárie M163
- Pracujte na journal clube

http://compbio.fmph.uniba.sk/vyuka/mbi/

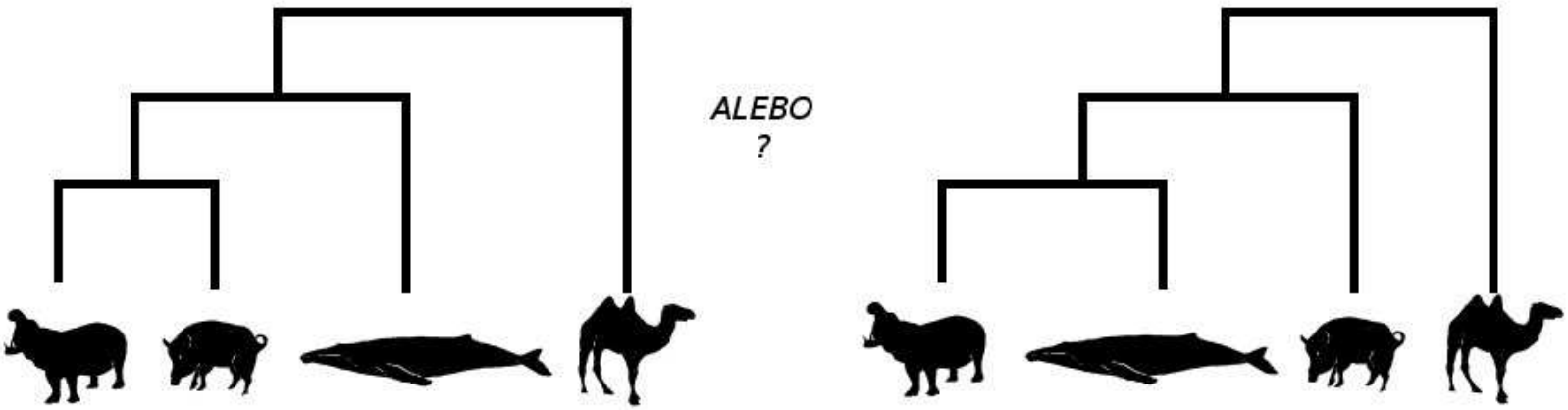
Journal club

- Skupiny na stránke, ak nemáte kontakty, dajte nám vedieť.
- Každý si prečíta pridelený článok.
- Stretnutie skupiny, diskusia, rozdelenie úloh (môžeme prísť)
- Do 19.12. 22:00 odovzdať správu za celú skupinu
  – vlastnými slovami hlavné metódy a výsledky článku
  – pochopiteľná pre študentov tohto predmetu (inf aj bio)
  – netreba pokryť všetko a naopak, môžete využiť aj iné zdroje
  – skúste vložiť vlastný pohľad na tému
  – rozsah cca 1-2 strany na osobu, jeden ucelený text
- V správe vymenujte členov skupiny, ktorí sa podieľali na jej spísaní, dostanú rovnako bodov
- 18.12. nepovinné prezentácie za bonusové body

Evolučné modely a stromy

Tomáš Vinař

30.10.2014



Rekonštrukcia fylogenetických stromov

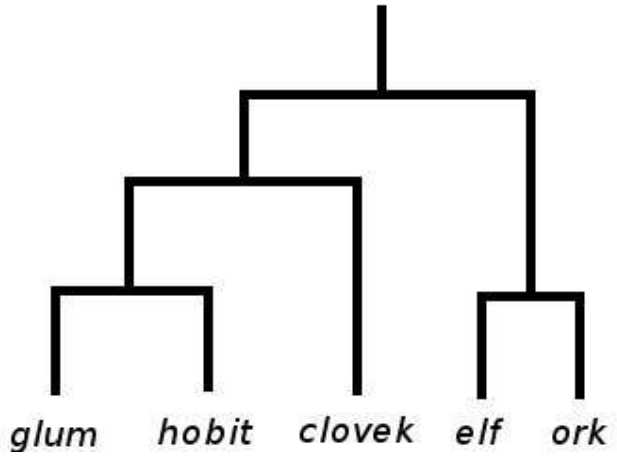
Vstup:

m **zarovnaných** sekvencií,
každá dĺžky n

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

Výstup:

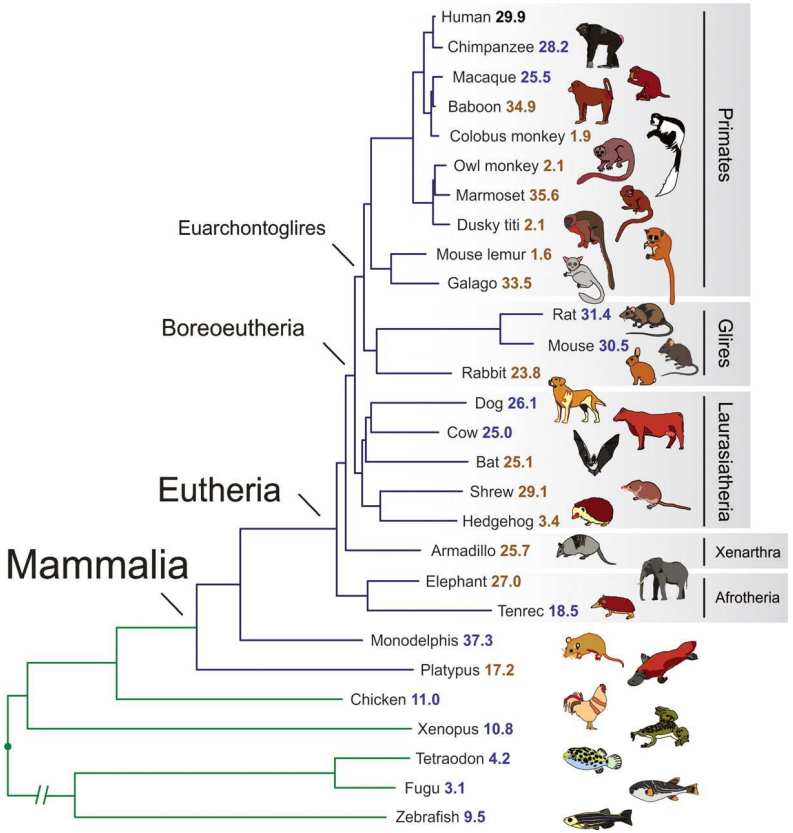
strom predstavujúci
ich evolučnú históriu



Newick format:
(((glum,hobit),človek),(elf,ork))

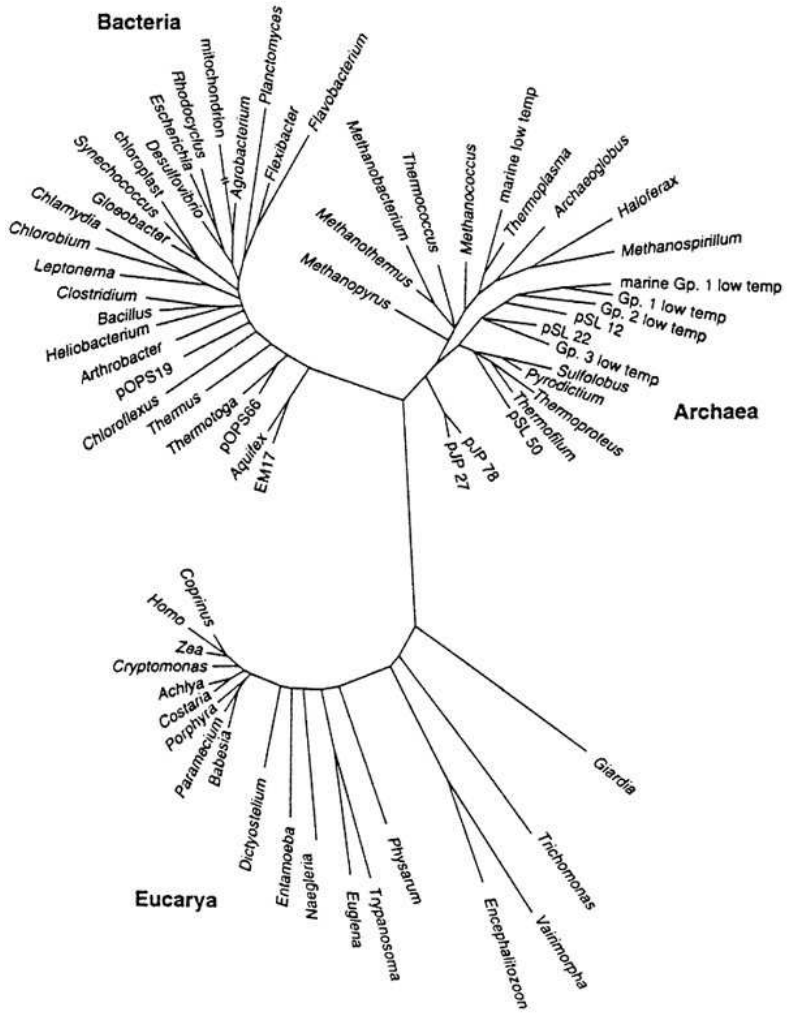
Zakorenené a nezakorenené stromy

[Margulies et al., 2007]



zakorenený pomocou “outgroup”

[Pace, 1997]

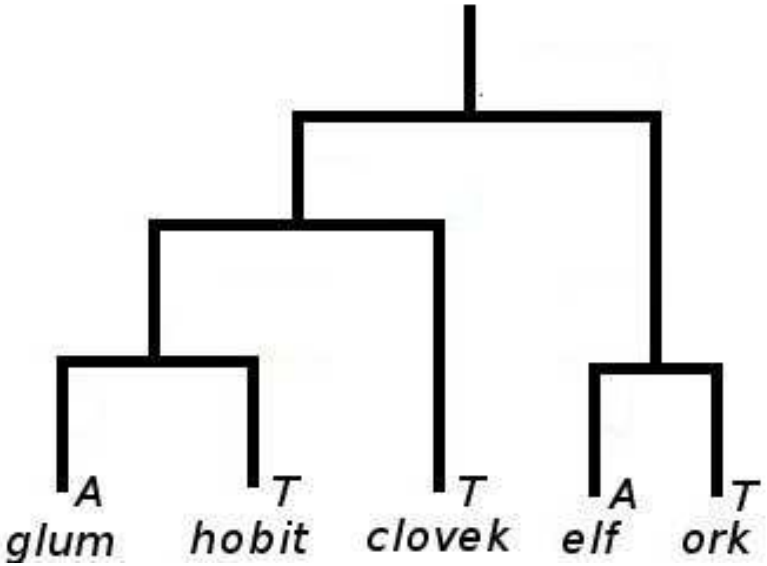


Maximum parsimony (úsporné stromy)

Úloha: Dané sú zarovnané sekvencie súčasných organizmov.
Chceme nájsť fylogenetický strom, ktorý vyžaduje **minimálny počet evolučných zmien**.

Evolučná zmena = mutácia jednej bázy na inú bázu

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



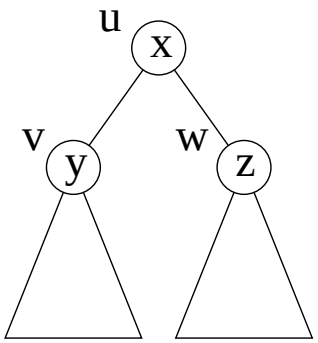
Výpočet ceny konkrétneho stromu

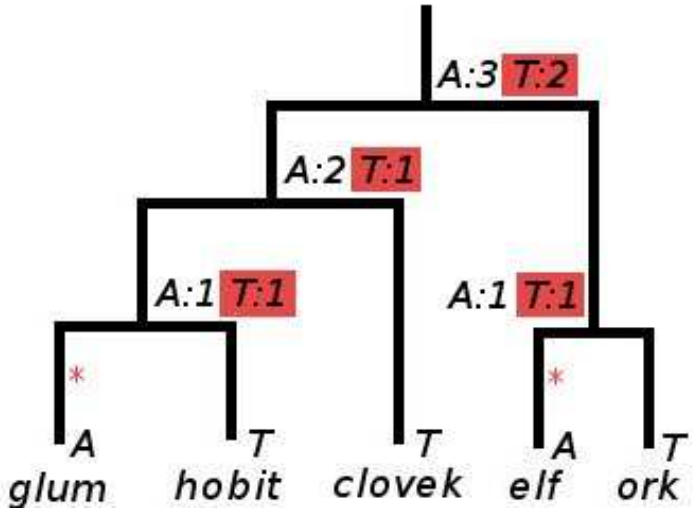
Môžeme rátať **dynamickým programovaním** pre každý stĺpec zarovnania zvlášť.

Pre každý vnútorný vrchol u a symbol x:

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

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





Časová zložitosť: $O(m)$, lineárna

Hľadanie najúspornejšieho stromu

NP-ťažký problém

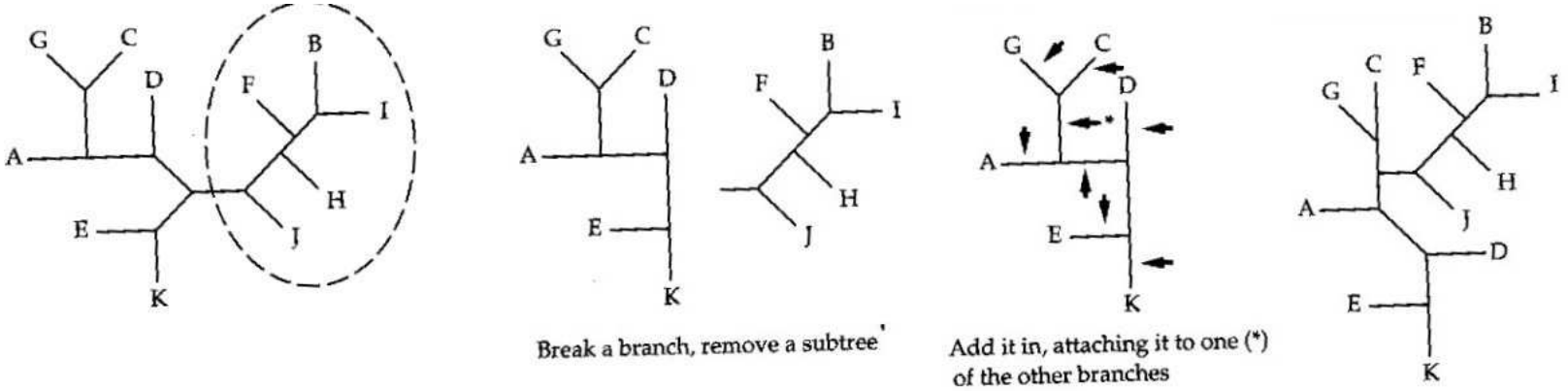
Triviálny algoritmus: vyskúšaj všetky možné stromy.

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

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

Heuristické prehľadávanie:

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



Neighbor Joining (Metóda spájania susedov)

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

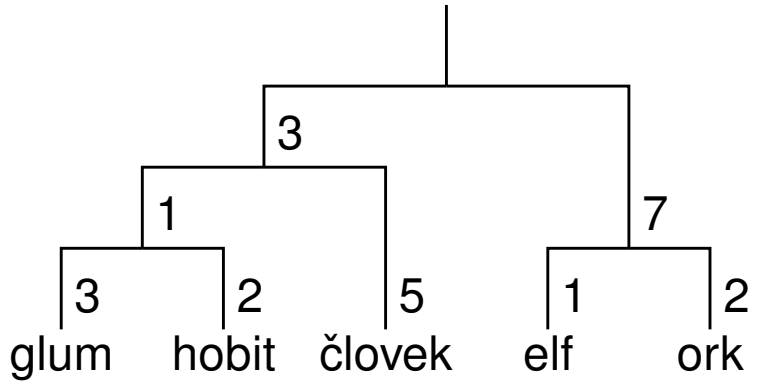
Jednoduchý príklad:

človek	C	A	G	T	T	A
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

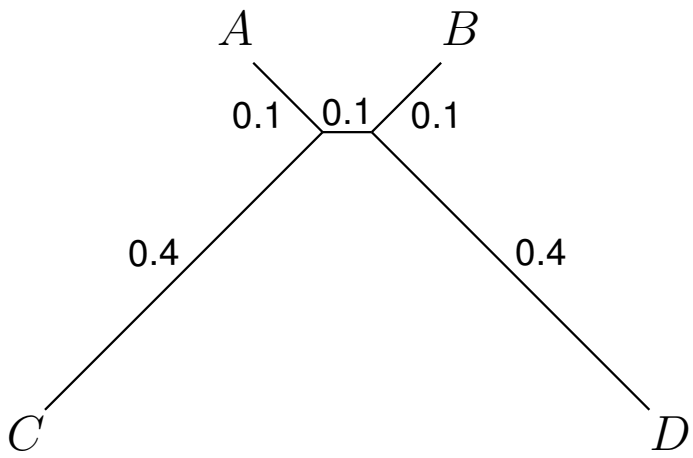
- Predpokladáme, že vzdialenosti $D_{i,j}$ skutočne zodpovedajú vzdialenostiam v strome (**aditivita**)
- Nájdeme dva listy i a j, o ktorých vieme **s určitosťou povedať**, že majú vo výslednom strome spoločného otca
- i a j spojíme a nahradíme ich ich otcom k s novými vzdialenosťami:

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

Časová zložitosť: $O(m^3)$

Ako určiť dva listy na spájanie?

(Prečo nie dva najbližšie?)



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

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

Problém so vzdialenosťami

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

Jukes-Cantorov model evolúcie

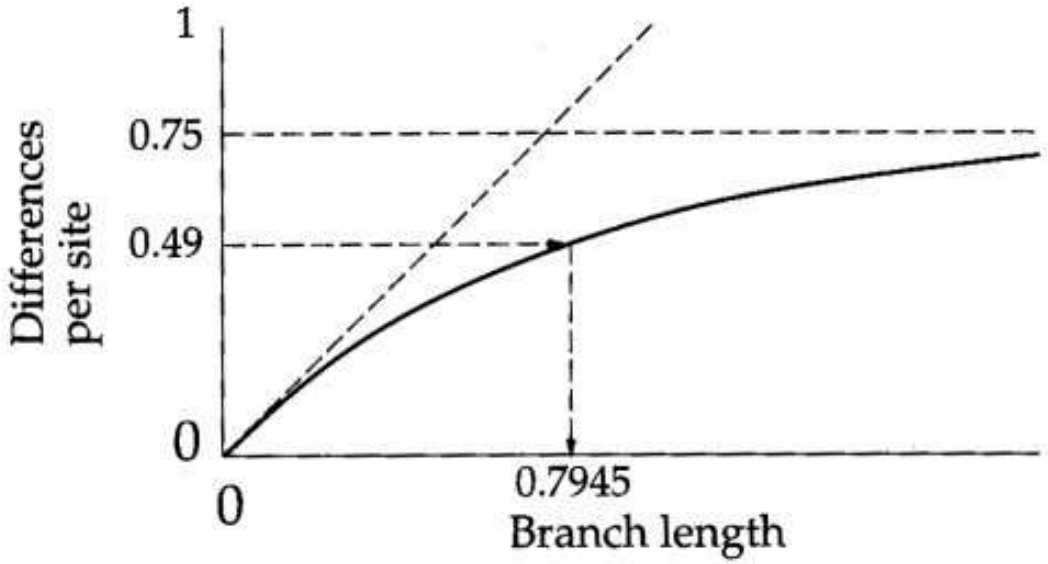
Pravdepodobnosť zmeny bázy na inú:

$\Pr(C_{t+\Delta t} \mid A_t) = \frac{1}{4}(1 - e^{-\frac{4}{3}\alpha\Delta t})$

α: rýchlosť evolúcie (počet substitúcií na jednotku času, ak sa pozeráme na veľmi malé časové jednotky)

Očakávaný počet pozorovaných zmien na bázu:

$D_S(\Delta t) = \frac{3}{4}(1 - e^{-\frac{4}{3}\alpha\Delta t})$



Späť ku spájaniu susedov (Neighbor Joining)

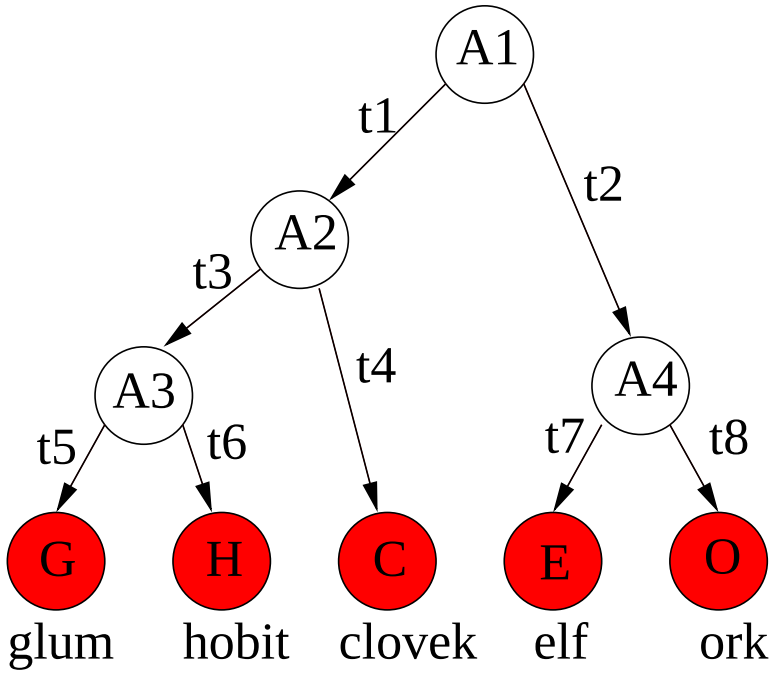
- Podľa takéhoto modelu môžeme korigovať pozorované vzdialenosti

$$D = \frac{3}{4}(1 - e^{-\frac{4}{3}\alpha\Delta t}) \qquad \Rightarrow \qquad \alpha\Delta t = -\frac{3}{4}\log(1 - \frac{4}{3}D)$$

- Často lepšie zložitejšie modely, ktoré zahŕňajú rôzne frekvencie báz, pomer tranzícií a transverzií, variabilnú rýchlosť evolúcie na rôznych pozíciách (pozri [Felsenstein, 2004, kap.13])

Najvierohodnejšie stromy (Maximum likelihood)

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



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

- $\Pr(G, H, C, E, O, A1, \ldots, A4) = \Pr(A1) \cdot \Pr(A2 \mid A1, t_1) \cdot \Pr(A4 \mid A1, t_2) \cdot \Pr(A3 \mid A2, t_3) \cdot \Pr(G \mid A3, t_5) \cdot \Pr(H \mid A3, t_6) \cdot \Pr(C \mid A2, t_4) \cdot \Pr(E \mid A4, t_7) \cdot \Pr(O \mid A4, t_8)$
- Pre **daný strom** a **dané dĺžky hrán** možno jednotlivé pravdepodobnosti spočítať použitím evolučného modelu (napr. Jukes-Cantor)
- **Vierohodnosť (likelihood) stromu:**

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

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

Ako nájsť najvieryhodnejší strom?

- Problém je NP-ťažký ;
  navyše komplikovaný tým, že na výpočet vierohodnosti
  potrebujeme aj dĺžky hrán

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

"Správnosť" fylogenetických algoritmov: Konzistentnosť

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

Porovnanie algoritmov

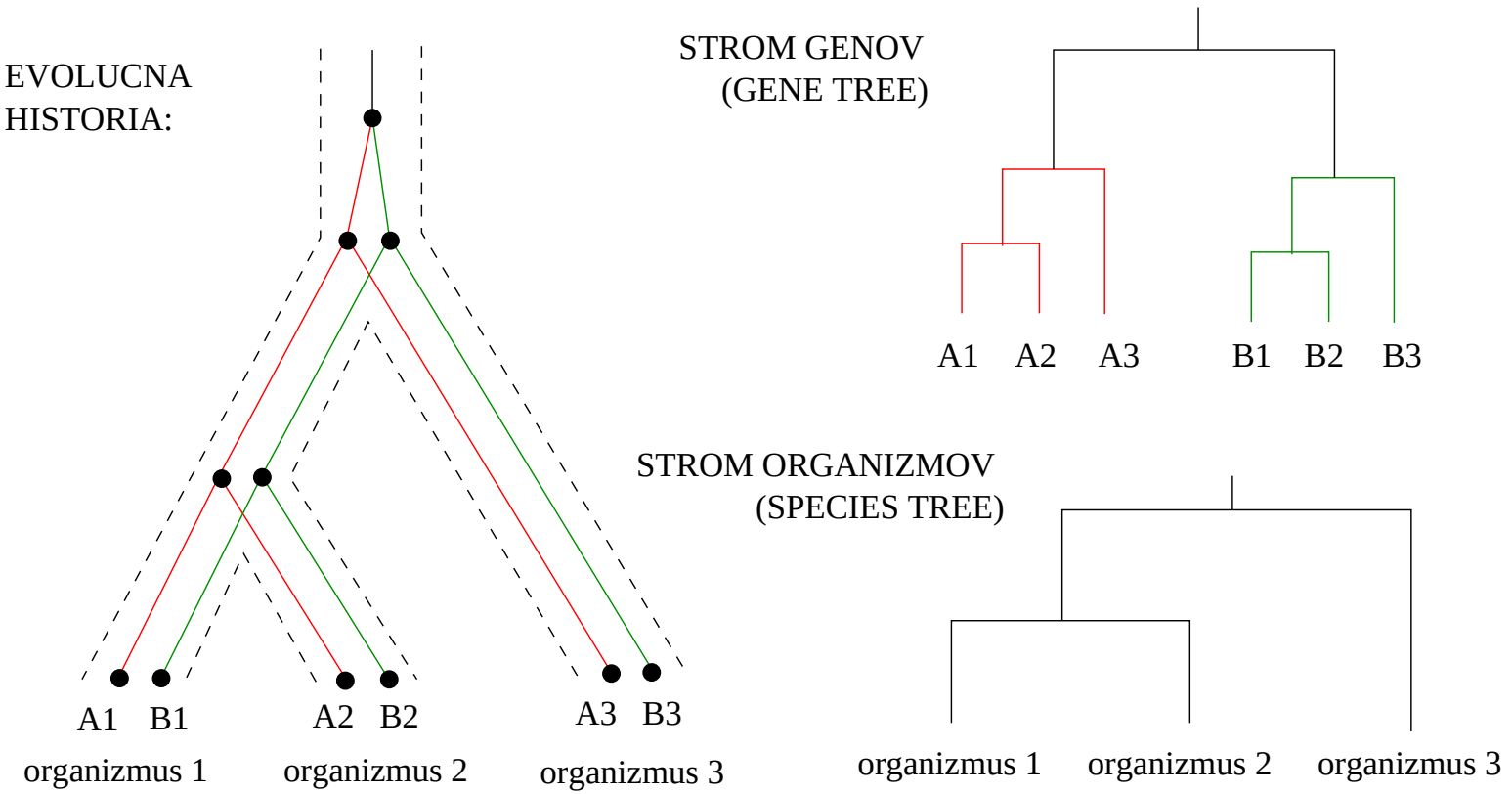
	Zložitosť	**Konzistentný**	**Využitie dát**
Parsimony (úspornosť)	NP-ťažký	NIE	celé sekvencie
Neighbor Joining	$O(m^3)$	ÁNO	iba vzdialenosti
Likelihood (vierohodnosť)	NP-ťažký	ÁNO	celé sekvencie

Odkiaľ zohnať dáta pre fylogenetiku?

- **Mitochondriálna DNA (mtDNA):**
  - Krátky cirkulárny genóm uložený v mitochondriách (človek: cca 16KB)
  - Dedí sa po materskej línii (žiadna rekombinácia)
  - Rýchlejšie mutácie – vhodný nielen pre druhy, ale aj jedince
  - Ľahko sa sekvenuje; osekvenovaný pre mnoho organizmov
- **Ribozomálna RNA (rRNA):**
  - Nepostrádateľná pri syntéze proteínov v ribozómoch
  - $\Rightarrow$ veľmi dobre zachovaná aj medzi druhmi
  - RDPII databáza

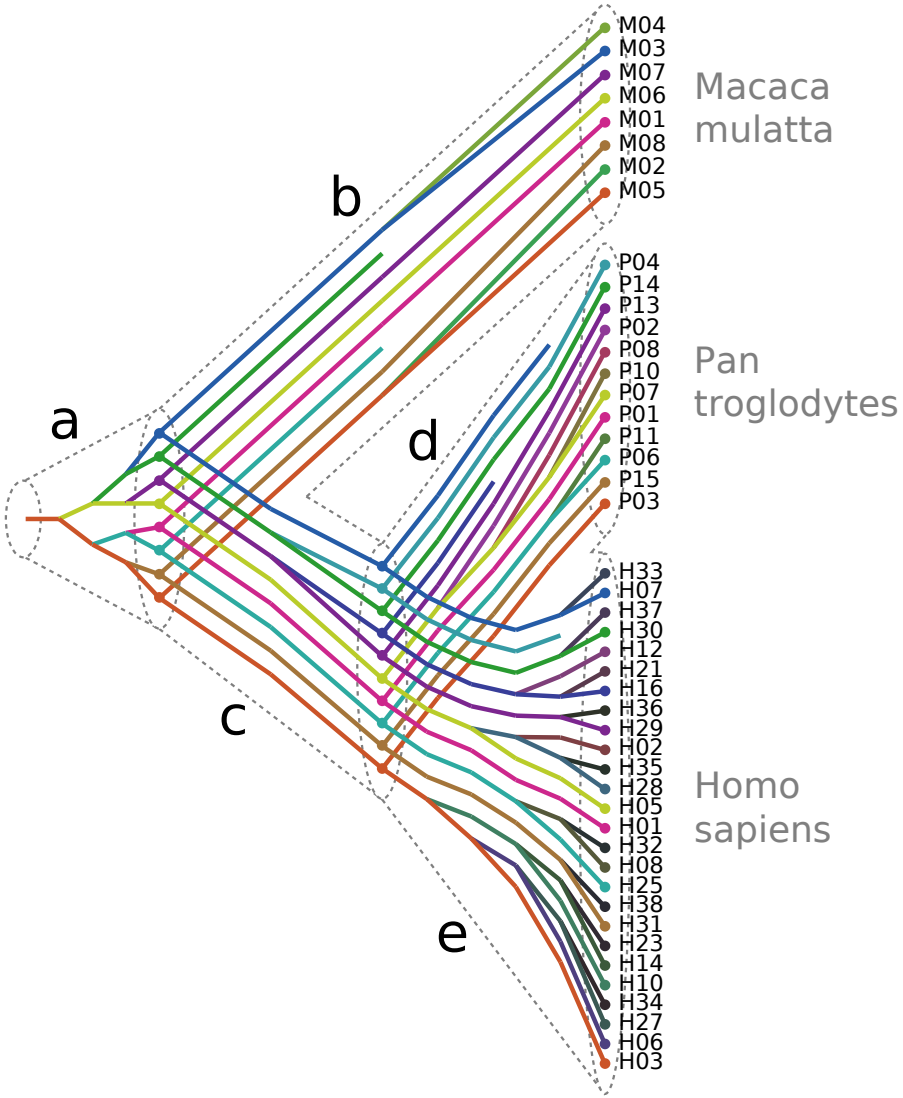
- **DNA sekvencie:** Čo tak:
  - Vybrať si sympatický gén
  - Nájsť jeho homológy v iných organizmoch
  - Použiť tieto na konštrukciu fylogenetického stromu

  Problém!!! Duplikácia génov (a vo všeobecnosti DNA duplikácia)



- **Homológ:** vyvinuli sa zo spoločného predka, podobná sekvencia
- **Ortológ:** najbližší spoločný predok je speciácia (napr. A1/A3)
- **Paralóg:** najbližší spoločný predok je duplikácia (napr. A1/B1,A1/B2)

Zložitejší príklad:



Zhrnutie:

- Modely evolúcie nukleotidov nám dávajú možnosť:
  - Odhadovať skutočnú evolučnú vzdialenosť (počet substitúcií) z počtu pozorovaných zmien medzi sekvenciami
  - Počítať pravdepodobnosť, že uvidíme zmenu nukleotidu za určitý čas t
- Tri metódy na vytváranie evolučných stromov:
  - Úsporné stromy (parsimony)
  - Spájanie susedov (neighbour joining)
  - Vierohodnosť stromov (maximum likelihood)
- Génové a organizmové stromy; komplikácie pri vytváraní stromov

Organizačné poznámky

- **Domáca úloha 2**
  bude na stránke do konca týždňa
- Otázky ohľadom journal clubu?

Komparatívna genomika

Tomáš Vinař

13.11.2014

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. tropicalis AATGGCTTCCATTTTGTGCCGCTGCTGAGGTCTTGTTCTGGGGAAGAT
```

- **Cieľ:** Štúdiom evolučných zmien sa snažíme nájsť funkčne
  významné časti sekvencie (napríklad gény kódujúce proteíny)

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

Prirodzený výber: dôležitá súčasť evolúcie

- Evolúcia DNA sekvencií pomocou mutácií
- Typy mutácií:
  - Neutrálne
  - Škodlivé (deleterious)
    $\Rightarrow$ **Purifikačný výber (purifying selection)**
  - Prospešné (advantageous)
    $\Rightarrow$ **Pozitívny výber (positive selection)**

Aplikácie princípov komparatívnej genomiky

- Hľadanie funkčných oblastí sekvencií
- Hľadanie génov pomocou komparatívnej genomiky
- Hľadanie génov pod vplyvom pozitívneho výberu

Úloha 1: Hľadanie funkčných oblastí sekvencií

Dôsledky purifikačného výberu:

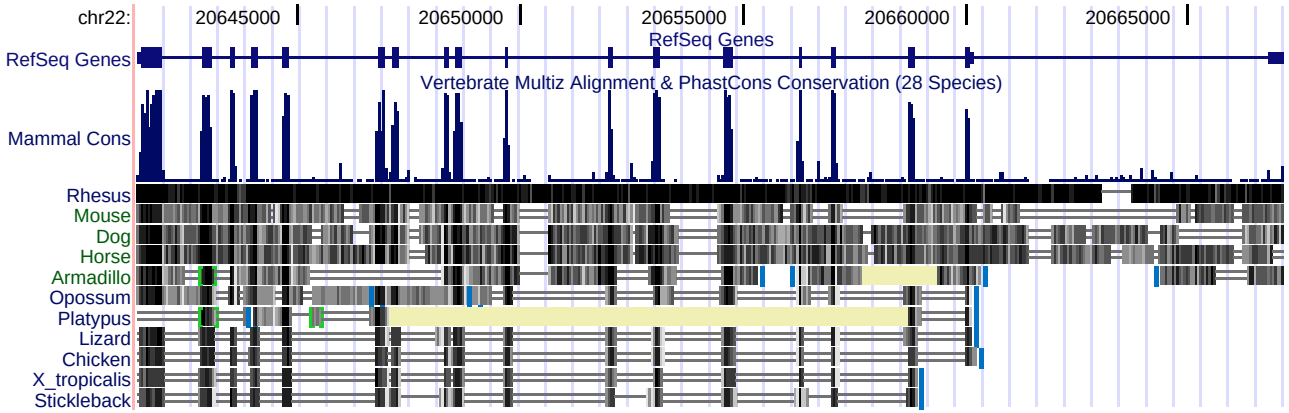
- Funkčné časti sekvencie zostávajú zachované
- Nefunkčné sekvencie sa vyvíjajú rýchlejším tempom ako funkčné

	Kódujúce	Intrón
Ľudské bázy pokryté zarovnaním	98%	48%
Zhoda v zarovnaniach	85%	69%

(myš vs. človek) [Mouse Genome Sequencing Consortium, 2002]

Štúdium purifikačného výberu

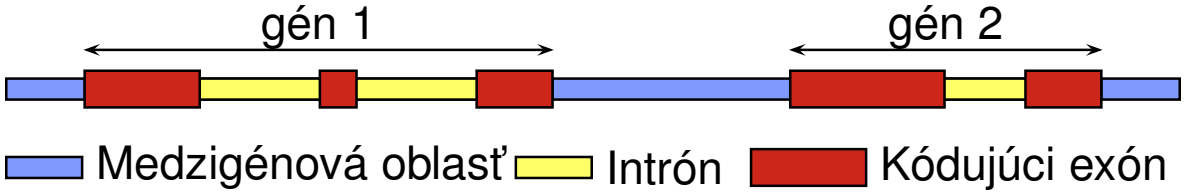
- **Úloha:** Hľadáme **zachované sekvencie** medzi jednotlivými organizmami



- Veľká časť zodpovedá známym funkčným elementom (kódujúce gény, regulačné regióny, a pod.)
- Zachované sekvencie ktoré sa neprekrývajú s funkčnými elementami — nové objekty pre výskum

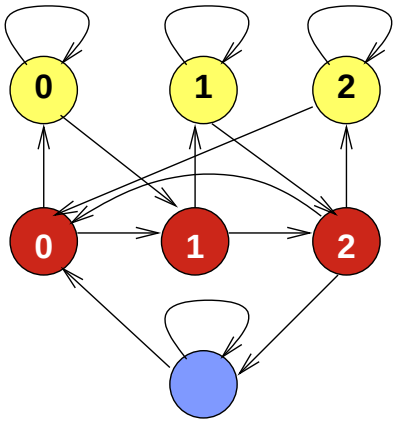
Opakovanie: hľadanie génov

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



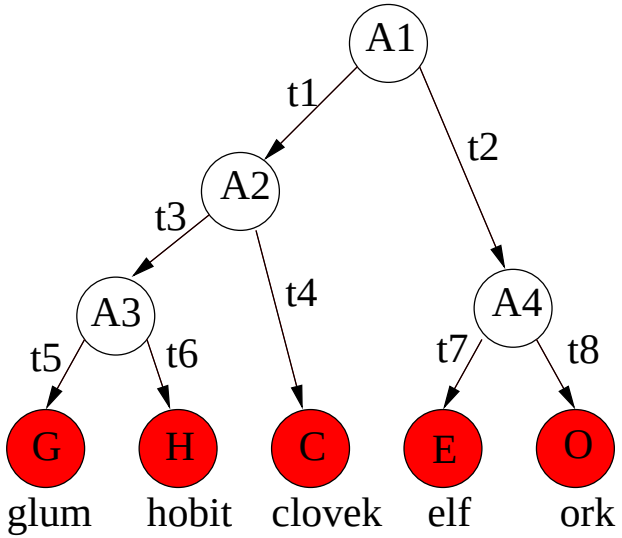
Vytvoríme skrytý Markovovský model (HMM), ktorý vie generovať sekvencie a ich anotácie podobné skutočným.

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



Opakovanie: pravdepodobnostné modely evolúcie

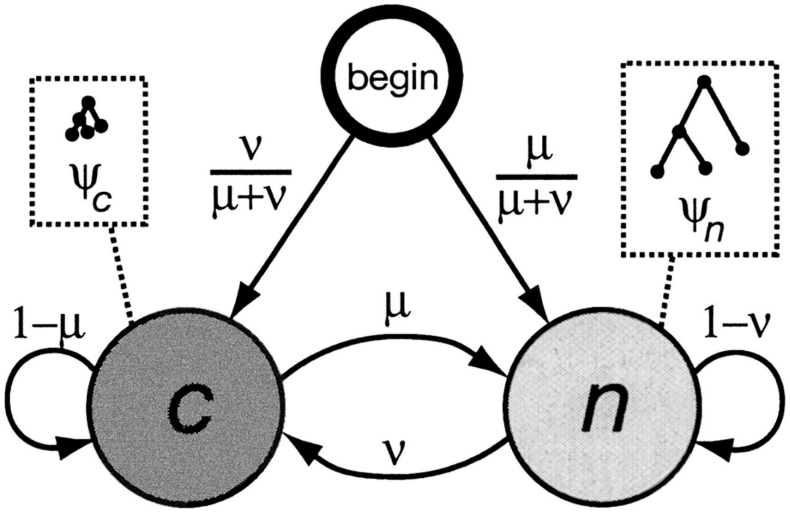
- Strom môžeme chápať ako **jednoduchý generatívny model**

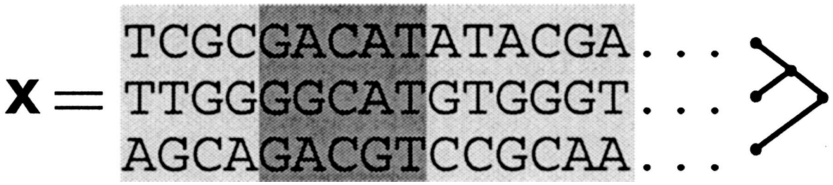


- Pre hranu z Y do X dĺžky t možno pravdepodobnosť mutácie spočítať použitím evolučného modelu, napr. Jukes-Cantor: $\Pr(X = C \mid Y = A, t) = \frac{1}{4}(1 - e^{-\frac{4}{3}\alpha t})$
- Pre celý strom $\Pr(G, H, C, E, O, A1, \ldots, A4) = \Pr(A1) \cdot \Pr(A2 \mid A1, t_1) \cdot \Pr(A4 \mid A1, t_2) \cdot \Pr(A3 \mid A2, t_3) \cdot \Pr(G \mid A3, t_5) \cdot \Pr(H \mid A3, t_6) \cdot \Pr(C \mid A2, t_4) \cdot \Pr(E \mid A4, t_7) \cdot \Pr(O \mid A4, t_8)$

PhastCons: detekcia dobre zachovaných sekvencií

- Použijeme **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

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

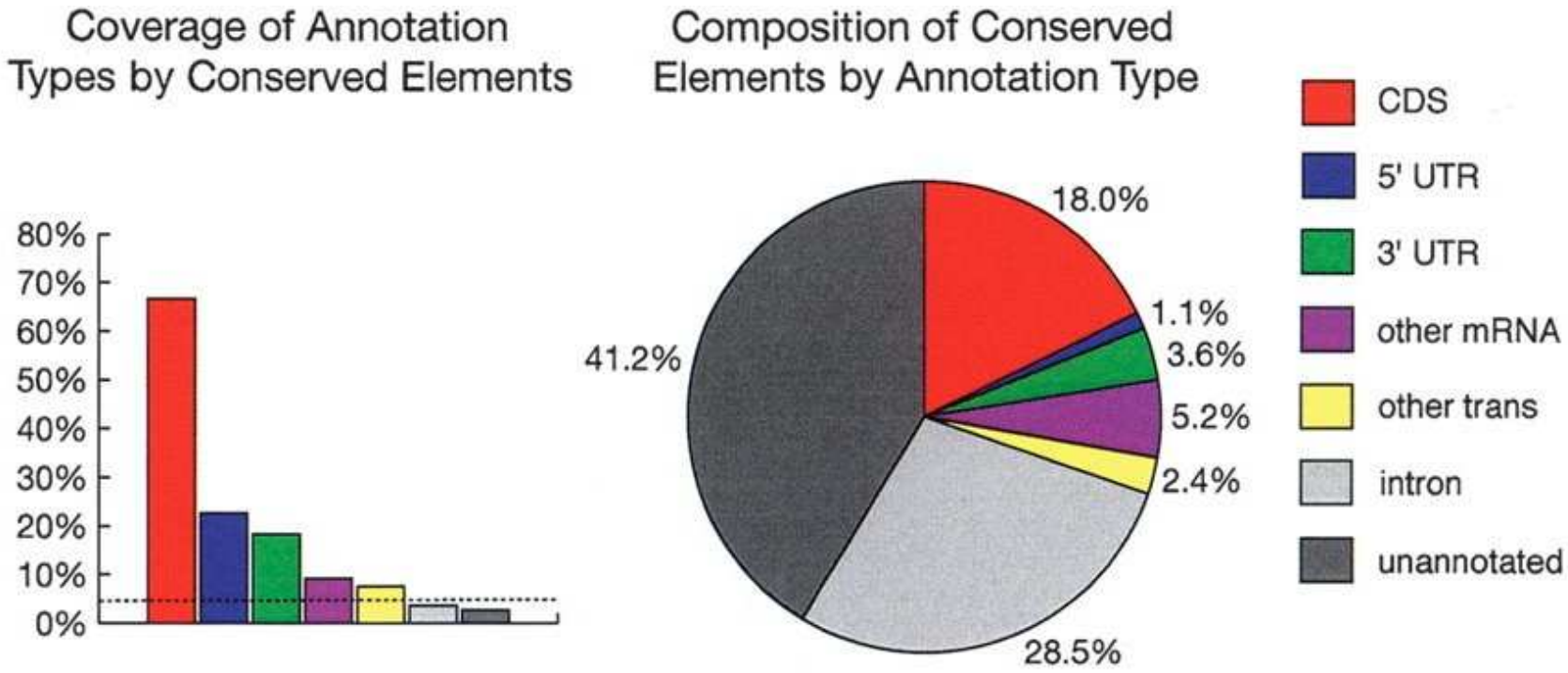
Problém: ako určovať parametre?

- Pri hľadačoch génov sme rátali štatistiky na známych génoch
- Tu ich môžeme nastaviť tak, aby sme maximalizovali pravdepodobnosť dát (zarovnania)
  $\max_{\text{param}} \Pr(\text{dáta}|\text{param})$

Výsledky celogenómovej aplikácie PhastCons-u

[Siepel et al., 2005]

Syntenické zarovnania genómov človeka, myši, sliepky, fugu



Úloha 2: Komparatívne hľadanie génov

- **Synonymné mutácie** nemenia zakódovanú aminokyselinu často neutrálne, rýchlejšie sa hromadia
- **Nesynonymné mutácie** menia zakódovanú aminokyselinu

Trojperiodické mutácie

Mutácia na tretej pozícii kodónu často synonymná.

	Báza v kodóne:		
	prvá	druhá	tretia
Zhoda	82%	87%	61%

(zarovnania myš vs. človek)

3-periodickosť mutácií pomáha nájsť gény.

Genetický kód

Alanine (A)
GC*

Cysteine (C)
TGC
TGT

Aspartic acid (D)
GAC
GAT

Glutamic acid (E)
GAA
GAG

Phenylalanine (F)
TTC
TTT

Glycine (G)
GG*

Histidine (H)
CAC
CAT

Isoleucine (I)
ATA
ATC
ATT

Lysine (K)
AAA
AAG

Leucine (L)
CT*
TTA
TTG

Methionine (M)
ATG

Asparagine (N)
AAC
AAT

Proline (P)
CC*

Glutamine (Q)
CAA
CAG

Arginine (R)
CG*
AGA
AGG

Serine (S)
TC*
AGT
AGC

Threonine (T)
AC*

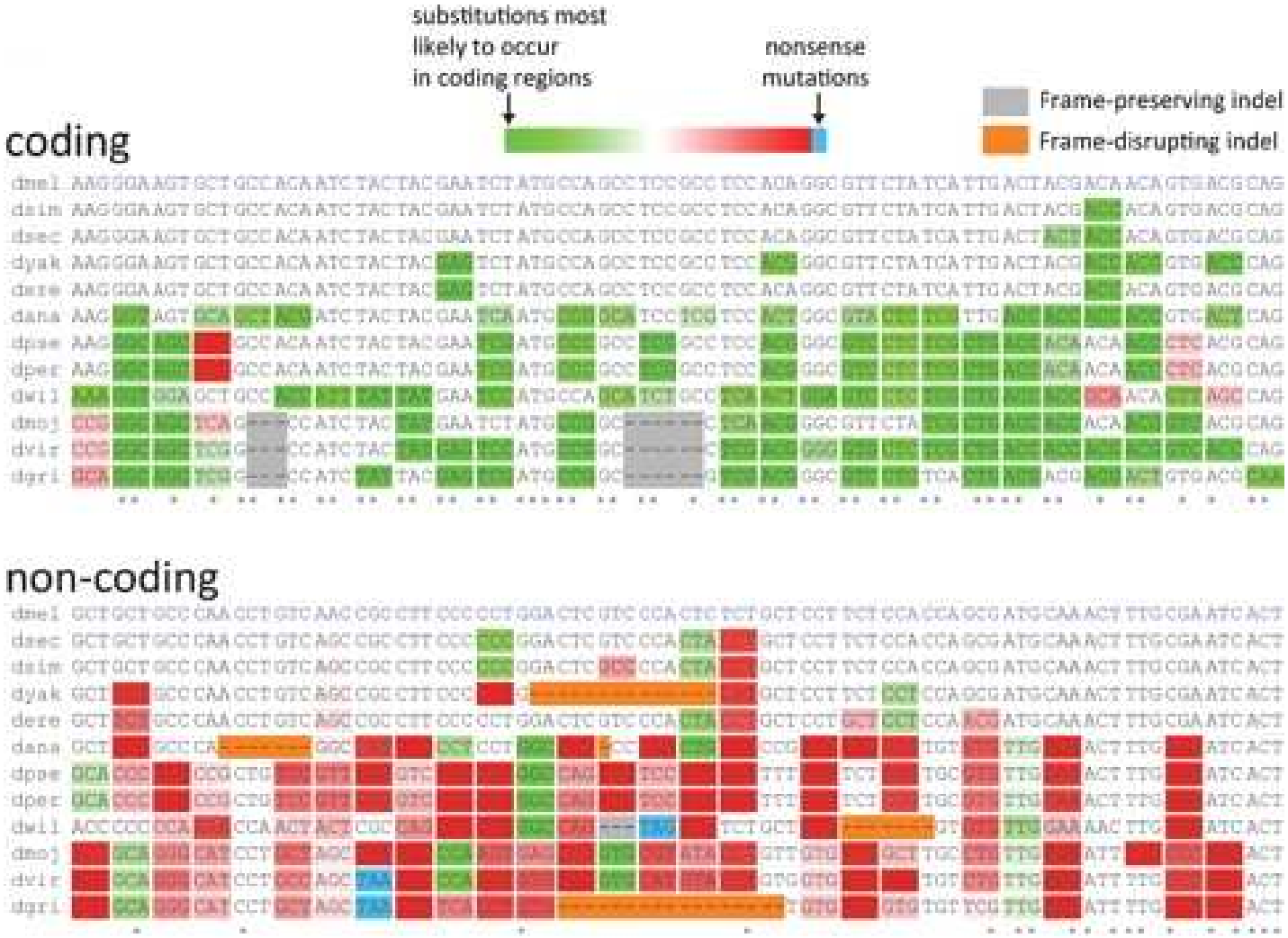
Valine (V)
GT*

Tryptophan (W)
TGG

Tyrosine (Y)
TAC
TAT

Stop codon (*)
TAA
TAG
TGA

Ako mutácie pomáhajú rozlišovať kódujúce oblasti



[Lin et al., 2007]

Fylogenetické HMM pre hľadanie génov

Napríklad Exoniphy [Siepel and Haussler, 2004], N-SCAN [Gross and Brent, 2006]

- Použijeme stavy z hľadača génov
- Pre každý stav máme evolučný model (maticu rýchlostí, dĺžky hrán)

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%

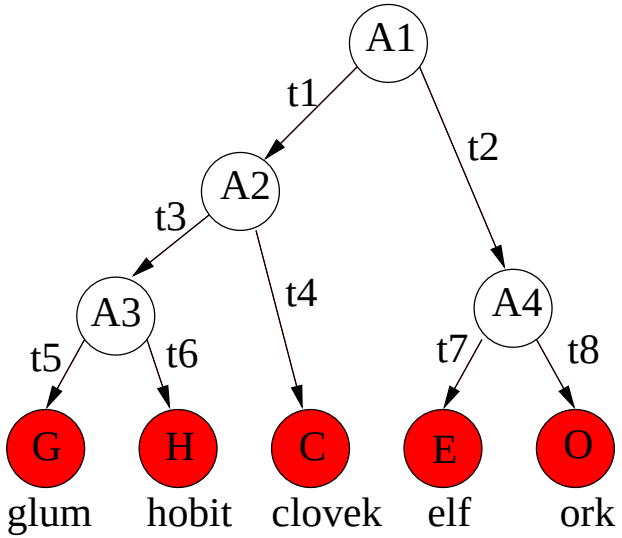
Guigo et al 2006, 1% ľudského genómu

Úloha 3: 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 (napr. **nesynonymné mutácie**)
- Vytvoríme pravdepodobnostný model evolúcie, ktorý bude rozlišovať synonymné a nesynonymné mutácie $\Rightarrow$ identifikácia sekvencií s neobvykle vysokým podielom nesynonymných mutácií

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: $\Pr(X = C \mid Y = A, t) = \frac{1}{4}(1 - e^{-\frac{4}{3}t})$
- Pre celý strom $\Pr(G, H, C, E, O, A1, \ldots, A4) = \Pr(A1) \cdot \Pr(A2 \mid A1, t_1) \cdot \Pr(A4 \mid A1, t_2) \cdot \Pr(A3 \mid A2, t_3) \cdot \Pr(G \mid A3, t_5) \cdot \Pr(H \mid A3, t_6) \cdot \Pr(C \mid A2, t_4) \cdot \Pr(E \mid A4, t_7) \cdot \Pr(O \mid A4, t_8)$

Všeobecnejšie modely mutácií

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

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

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

- **Rovnovážny stav:** frekvencie $\pi_A, \pi_C, \pi_G, \pi_T$ nemení sa v čase

- Pre daný čas t, môžeme vypočítať pravdepodobnosť každej substitúcie (**transition probabilities**):

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

Znižovanie počtu parametrov — HKY matica

Hasegawa, Kishino a Yano [Hasegawa et al., 1985]

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

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

Substitučný model pre kodóny

Namiesto jednotlivých báz uvažuje trojice [Goldman and Yang, 1994]

Rýchosť zmeny z kodónu i na kodón j:

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

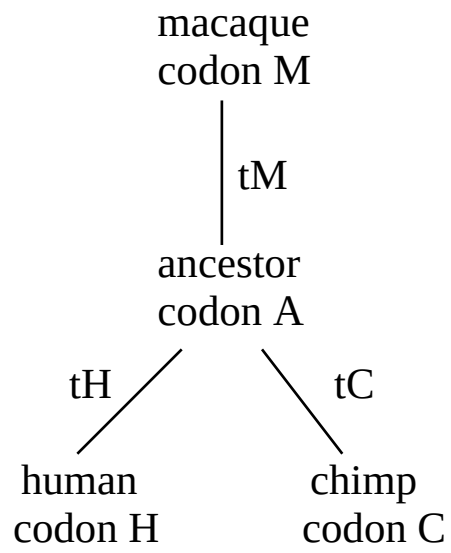
Príklad: $\mu_{AAC,GGC} = 0$, $\mu_{CTA,CTT} = \beta\pi_{CTT}$, $\mu_{CTA,CCA} = \omega\alpha\pi_{CCA}$

Parametre: Frekvencie kodónov π_j, ω, $\kappa = \alpha/\beta$

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

Generatívny model evolúcie kodónov

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



Inferencia (program PAML)

Nájdeme parametre (π, κ, ω, dĺžky hrán) maximalizujúce **vierohodnosť dát (likelihood):**

$$\Pr(H, C, M \mid \pi, \kappa, \omega, \text{dĺžky hrán}) = \sum_A \Pr(A, H, C, M \mid \pi, \kappa, \omega, \text{dĺžky hrán})$$

Testovanie pozitívneho výberu — likelihood ratio tests

[Zhang et al., 2005, Yang and Nielsen, 2002]

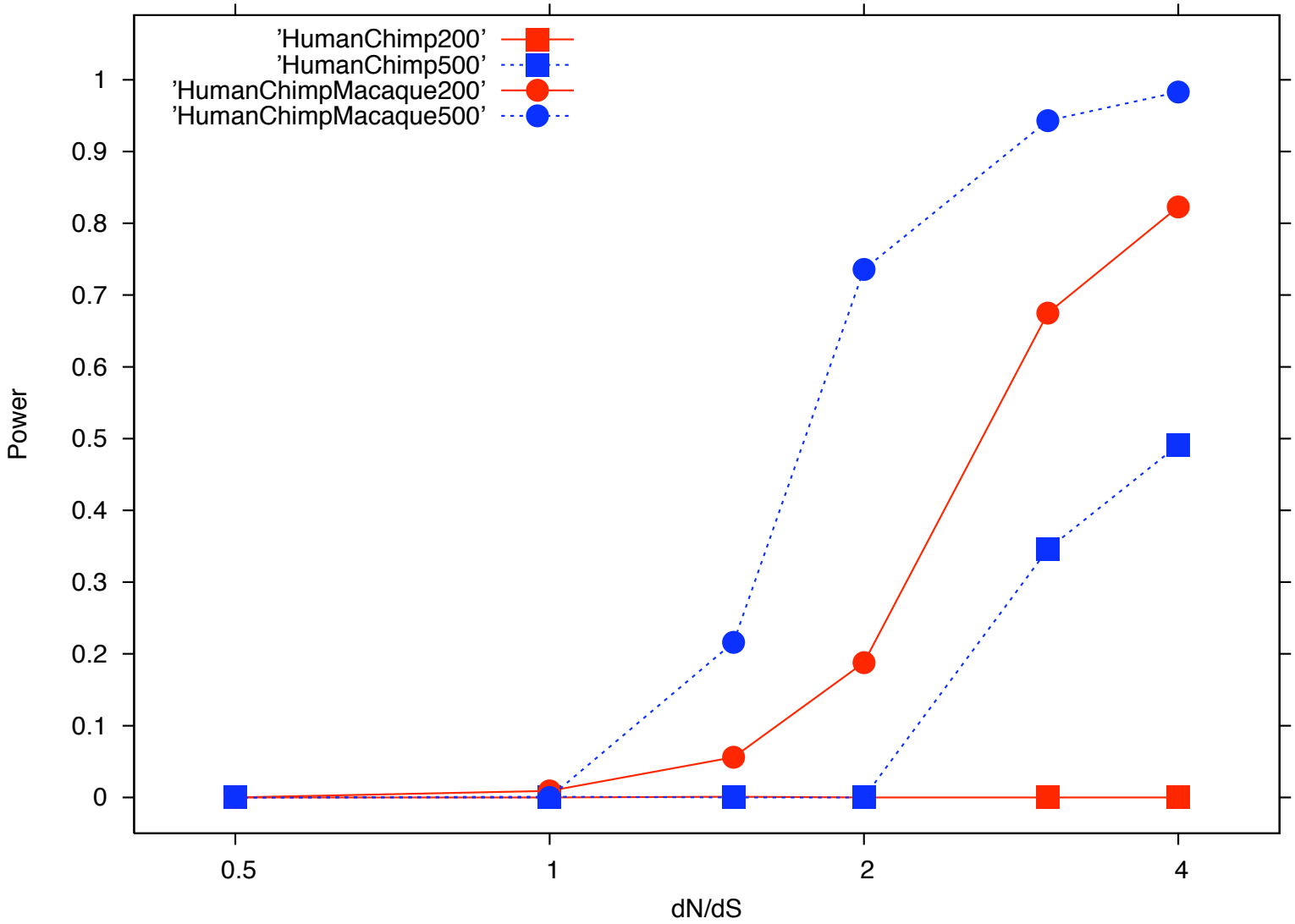
- Spočítame vierohodnosť dát L_A keď $\omega < 1$
- Spočítame vierohodnosť dát L_B bez obmedzenia ω
- Vždy platí $L_B \geq L_A$
- Ak skutočné $\omega < 1$, $L_A \approx L_B$

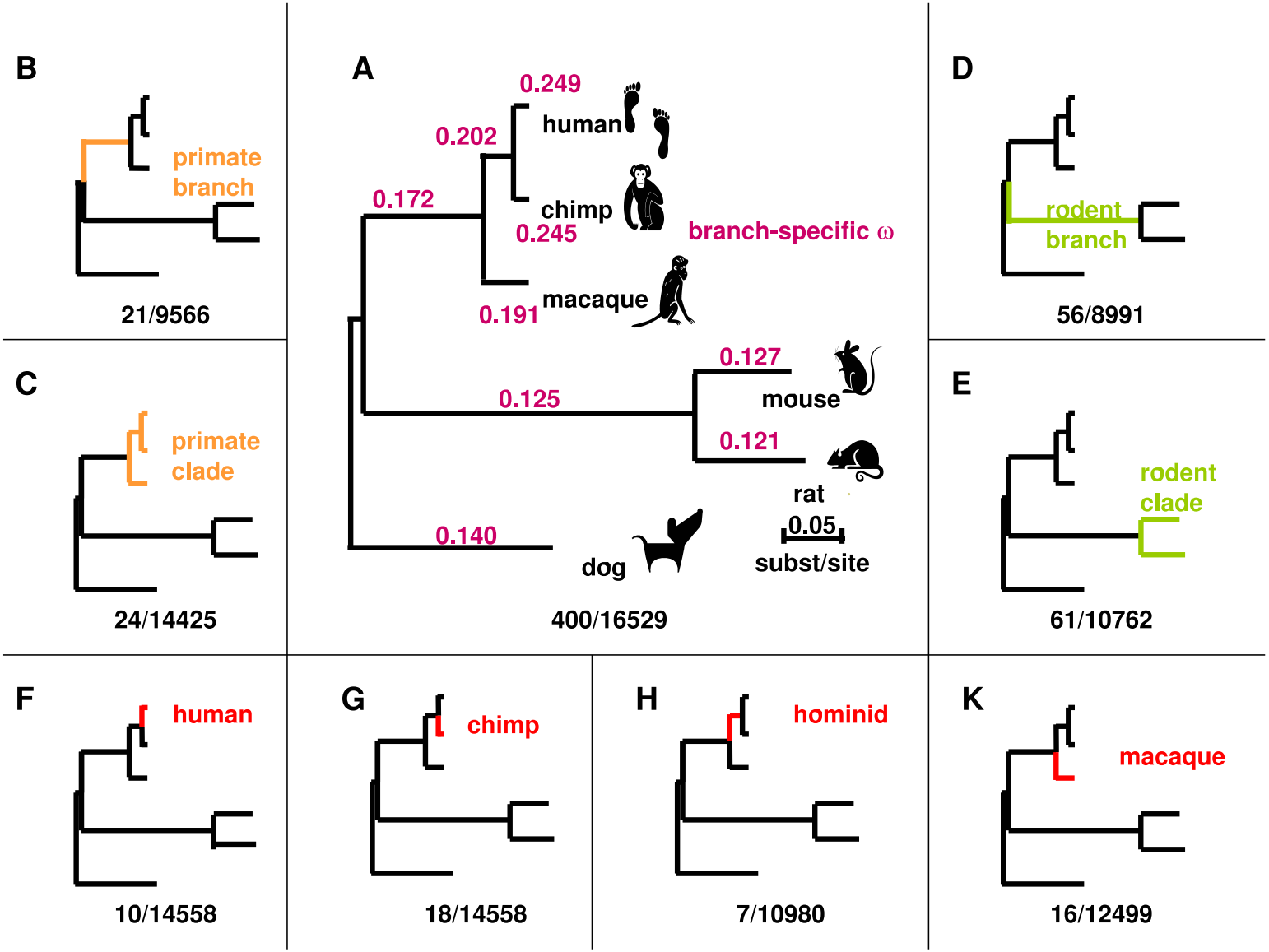
Test pozitívneho výberu

Ak L_B je štatisticky významne väčšie ako L_A,
gén je **pod vplyvom pozitívneho výberu**.

Za predpokladu, že $\omega < 1$, platí $2\log(L_B/L_A) \approx \chi^2_1$

Viacej genómov pomáha vylepšiť účinnosť testov




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
0.140
dog
0.05
subst/site
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

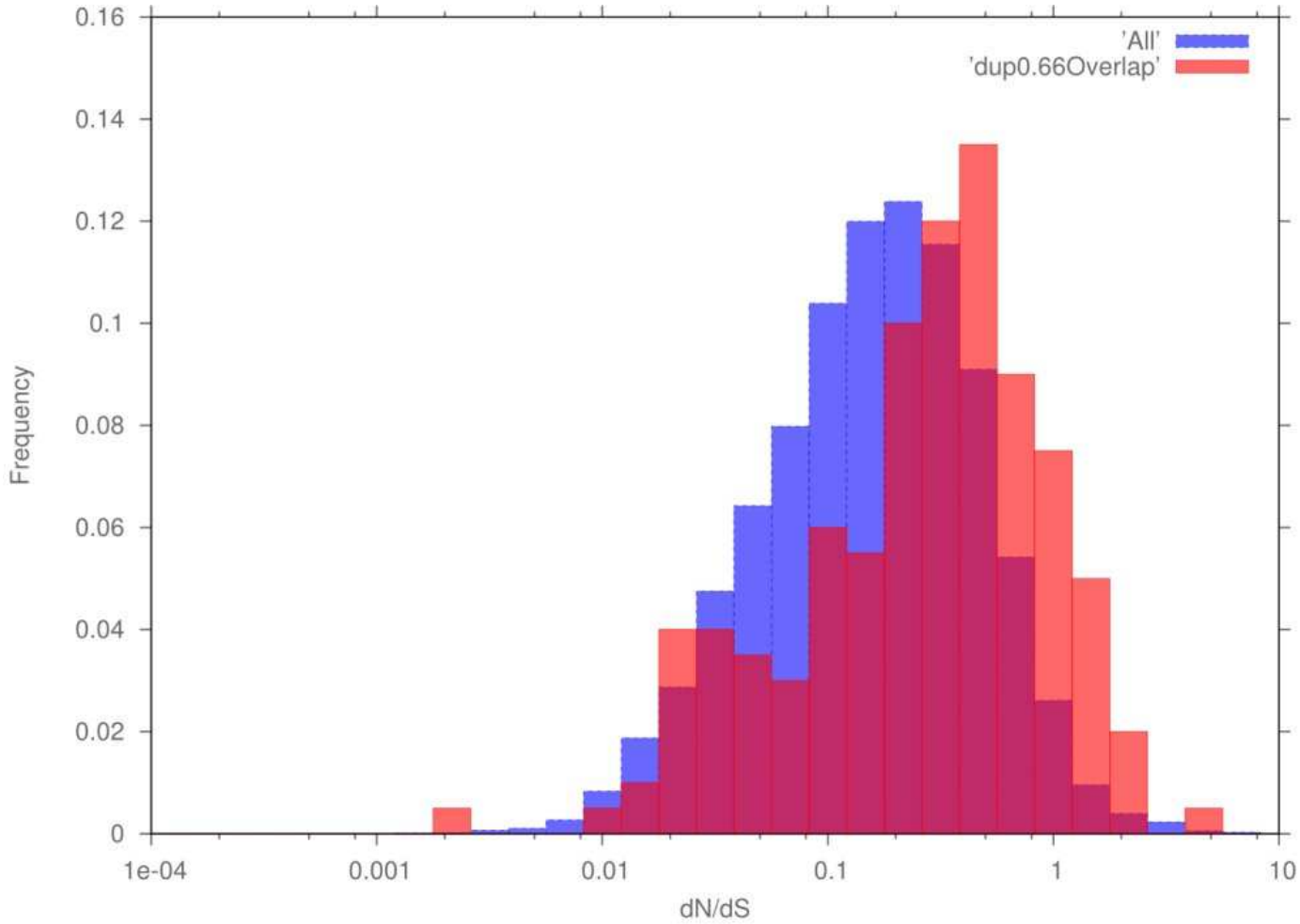
Funkčné kategórie obohatené o gény s pozitívnym výberom

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

Pozitívny výber v duplikovaných génoch



Zhrnutie

- Prirodzený výber má významnú úlohu v evolúcii organizmov
- **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

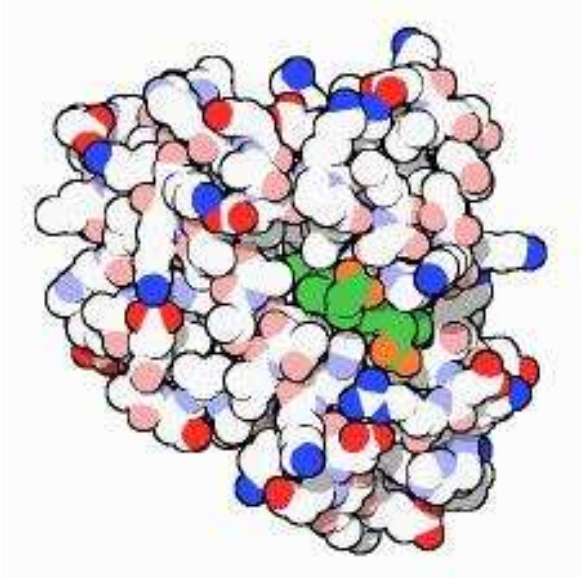
Organizačné poznámky

- DÚ2 do budúcej stredy rána
- DÚ3 zverejnená budúci týždeň, odovzdať do 17.12.
- Štvrtok 18.12. nepovinné prezentácie journal clubu
- Piatok 19.12. návrhy projektov (ak chcete robiť projekt)
- Piatok 19.12. správy zo journal clubu
- 23.1. odovzdanie projektov

Štruktúra a funkcia proteínov

Broňa Brejová

27.11.2014



Proteíny

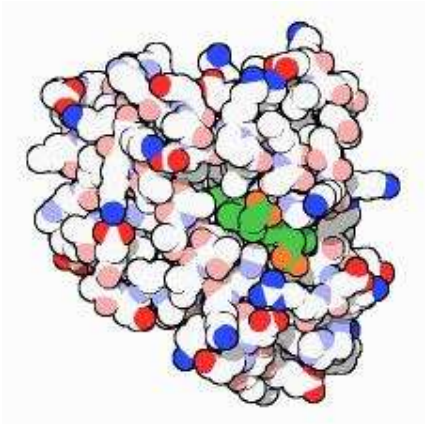
Reťazce 20 rôznych aminokyselín s rôznymi chemickými vlastnosťami:

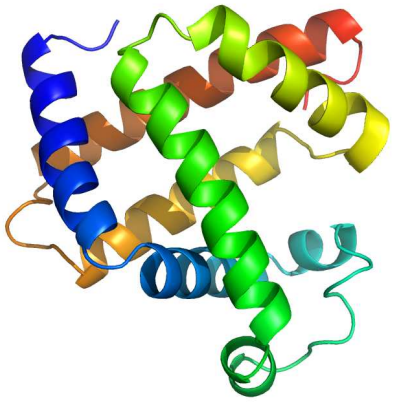
Amino Acid	Side chain	Hydrophobic	Polar	Charged	
Alanine (A)	-CH3	X	-	-	-
Arginine (R)	-(CH2)3NH-C(NH)NH2	-	X	basic	-
Asparagine (N)	-CH2CONH2	-	X	-	-
Aspartic acid (D)	-CH2COOH	-	X	acidic	-
Cysteine (C)	-CH2SH	X	-	acidic	-
Glutamic acid (E)	-CH2CH2COOH	-	X	acidic	-
Glutamine (Q)	-CH2CH2CONH2	-	X	-	-
Glycine (G)	-H	-	-	-	-
Histidine (H)	-CH2-C3H3N2	-	X	weak basic	Aromatic
Isoleucine (I)	-CH(CH3)CH2CH3	X	-	-	Aliphatic
Leucine (L)	-CH2CH(CH3)2	X	-	-	Aliphatic
Lysine (K)	-(CH2)4NH2	-	X	basic	-
Methionine (M)	-CH2CH2SCH3	X	-	-	-
Phenylalanine (F)	-CH2C6H5	X	-	-	Aromatic
Proline (P)	-CH2CH2CH2-	X	-	-	-
Serine (S)	-CH2OH	-	X	-	-
Threonine (T)	-CH(OH)CH3	-	X	weak acidic	-
Tryptophan (W)	-CH2C8H6N	X	-	-	Aromatic
Tyrosine (Y)	-CH2-C6H4OH	X	X	-	Aromatic
Valine (V)	-CH(CH3)2	X	-	-	Aliphatic

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
- Náročný a drahý proces
- Databáza štruktúr PDB
  97 000 proteínových štruktúr
  (UniProt má takmer má vyše 87 miliónov sekvencií)
- Štrukturálna genomika: snaha určovať štruktúry vo veľkom

Bioinformatický problém: určovanie štruktúry proteínov

(protein structure prediction, protein folding)

Vstup: sekvencia proteínu

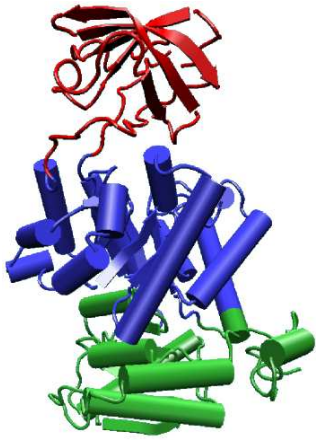
Výstup: 3D pozície atómov alebo aminokyselín

Ab initio metódy

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

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

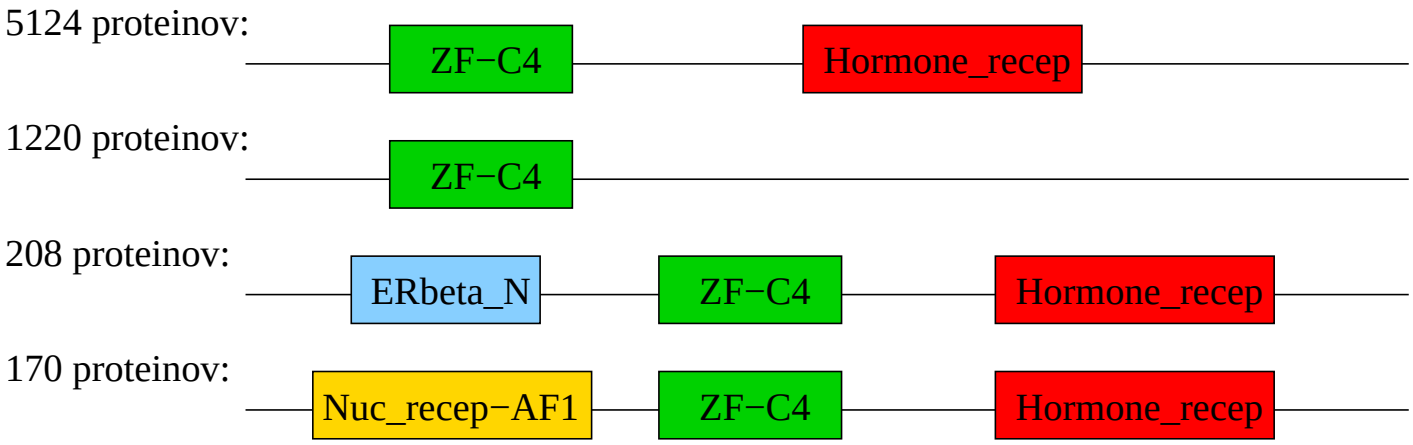
Domény v proteínoch klasikované do rodín

79% proteínov aspoň jedna známa doména

57% proteínových sekvencií pokrývajú známe domény

Príklad:

4 z 91 architektúr obsahujúcich doménu Zinc finger, C4 type (databáza Pfam)

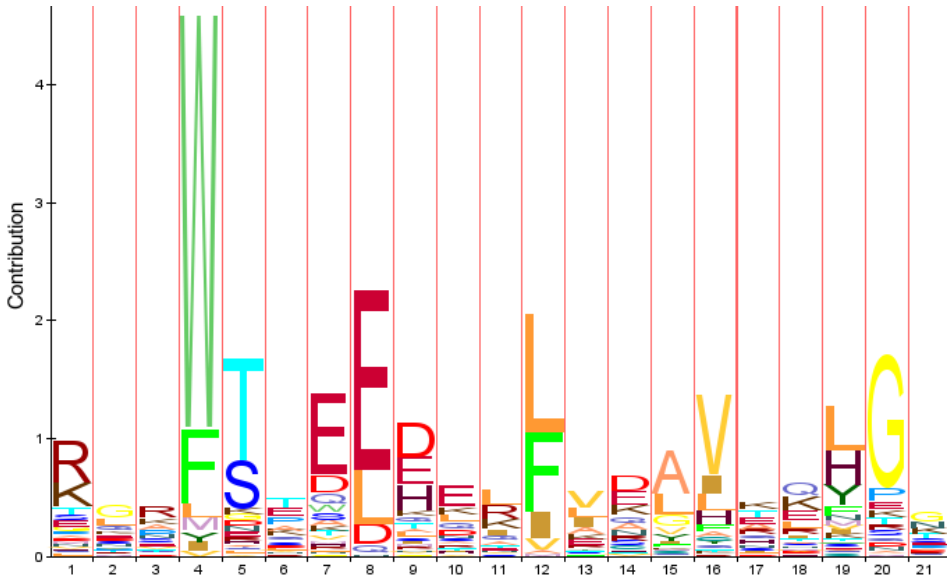


Hľadanie rodín

Cieľ: Zisti, do ktorej rodiny patrí daný proteín

- Zarovnania medzi známymi prvkami rodiny a novým proteínom nemusia nájsť vzdialených členov
- Viacnásobné zarovnanie rodiny ukáže dôležité zachovane pozície
- Rodinu reprezentujeme pravdepodobnostným profilom

```
MEEWSASEANLFEEALEKYGKDF
PDEWTVEDKVLFEQAFSFHGKT.
GTKWTAEENKKFENALAFYDKDT
SKNWSEDDLQLLIKAVNLFPAGT
EKPWSNQETLLLLEAIETYGDD.
AREWTDQETLLLLEGLEMHKDD.
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: logaritmus pomeru pravdepodobností v dvoch modeloch

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

Hračkársky príklad PSSM

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

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

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

Hračkársky príklad PSSM

- Majme zarovananie 10 sekvencií s nasledujúcimi počtami

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

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

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

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

Pseudocounts

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

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

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

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

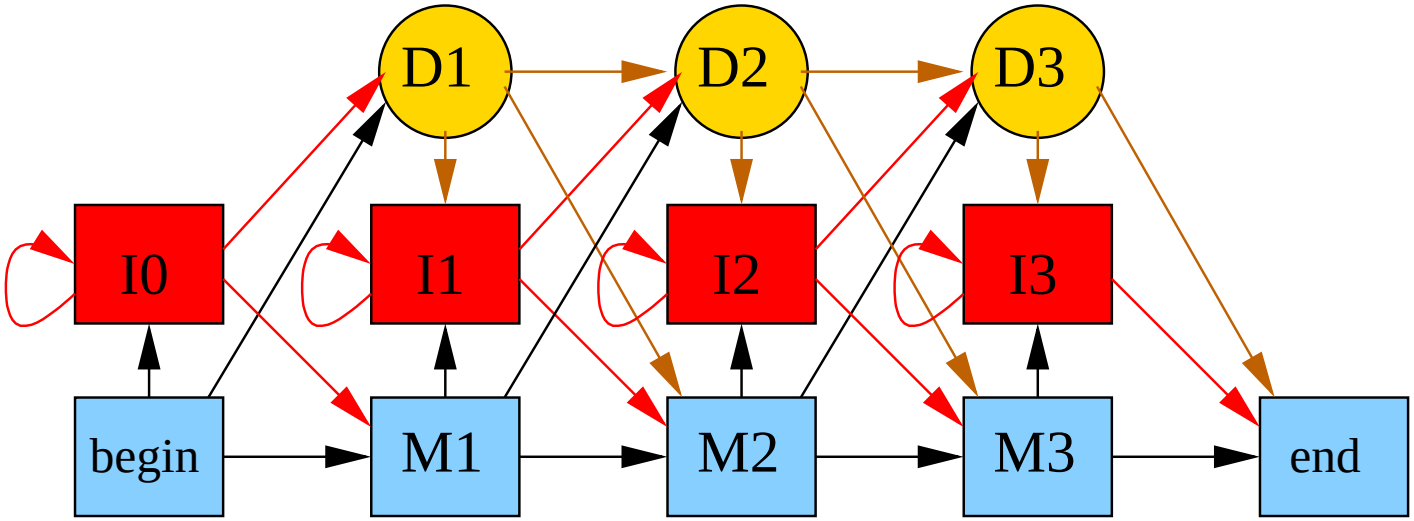
Potom postupujeme ako predtým

Profilové HMM

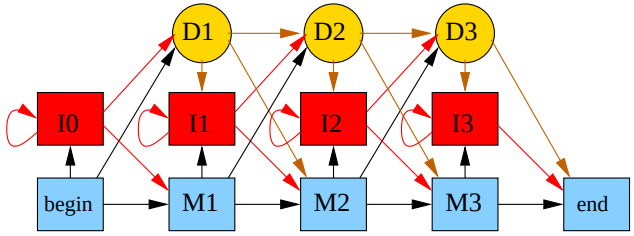
Rozšír profil o inzercie a delécie

PSSM profil ako HMM:

Profilové HMM: match state, insert state, delete state



Konštrukcia profilového HMM



- Začneme s 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

- Pre profilové HMM používame Viterbiho algoritmus
  (alebo aposteriórnu pravdepodobnosť)
- PSSM profily môžeme zarovnať dynamickým programovaním
  s jednotným skóre pre medzery
- Rodiny domén reprezentované ako profilové HMM
  napr. databáza Pfam
- PSI-Blast vytvára PSSM za pochodu z podobných proteínov
- PSSM sa používajú aj na reprezentáciu motívov v DNA (minulá
  prednáška)

Protein threading

Čo ak k proteínu nenájdeme žiadnu doménu?

- Aj proteíny s pomerne odlišnou sekvenciou môžu mať podobnú štruktúru
- Môžem skúsiť "napasovať" proteín na každú známu štruktúru
- Určitý typ zarovnania, ale pri skórovaní beriem do úvahy aj interakcie medzi amino kyselinami blízko v štruktúre
- Výpočtovo ťažký problém

Zhrnutie: akú štruktúru má proteín?

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

Minimalizácia energie je výpočtovo veľmi náročná
Súťaž CASP raz za dva roky

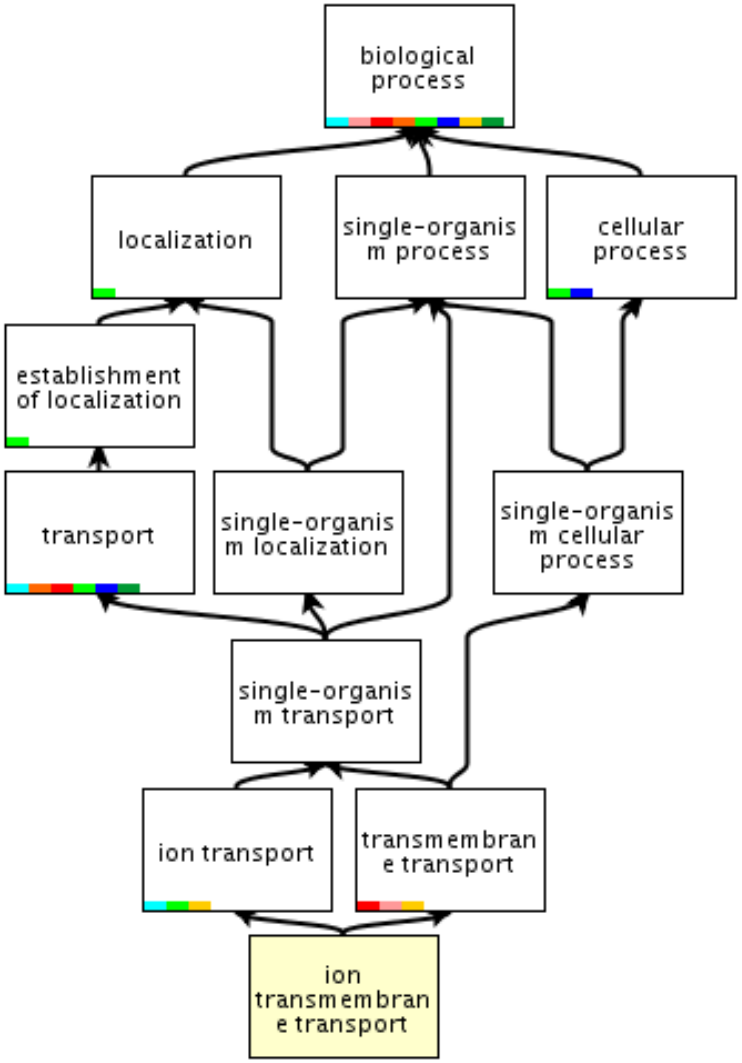
Zaujímavosti: Folding@home, Foldit

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:

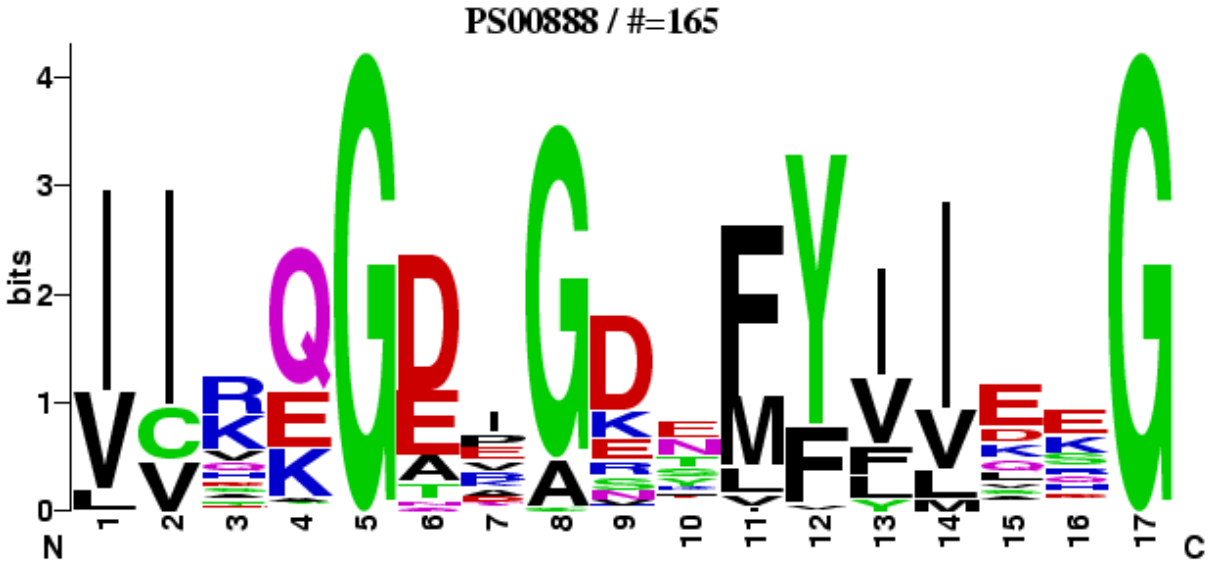


Ďalšie použitia HMM a profilov na proteíny

- Určovanie sekundárnej štruktúry
- Určovanie transmembránových proteínov a signálnych peptidov
- Určovanie funkčných motívov a posttranslačných modifikácií (databáza PROSITE)

Cyclic nucleotide-binding domain signature 1:

`[LIVM]-[VIC]-x-{H}-G-[DENQTA]-x-[GAC]-{L}-x-[LIVMFY](4)-x(2)-G`



Organizačné poznámky

- DÚ 2 na stránke, odovzdať do 3.12.2014

Regulácia génovej expresie

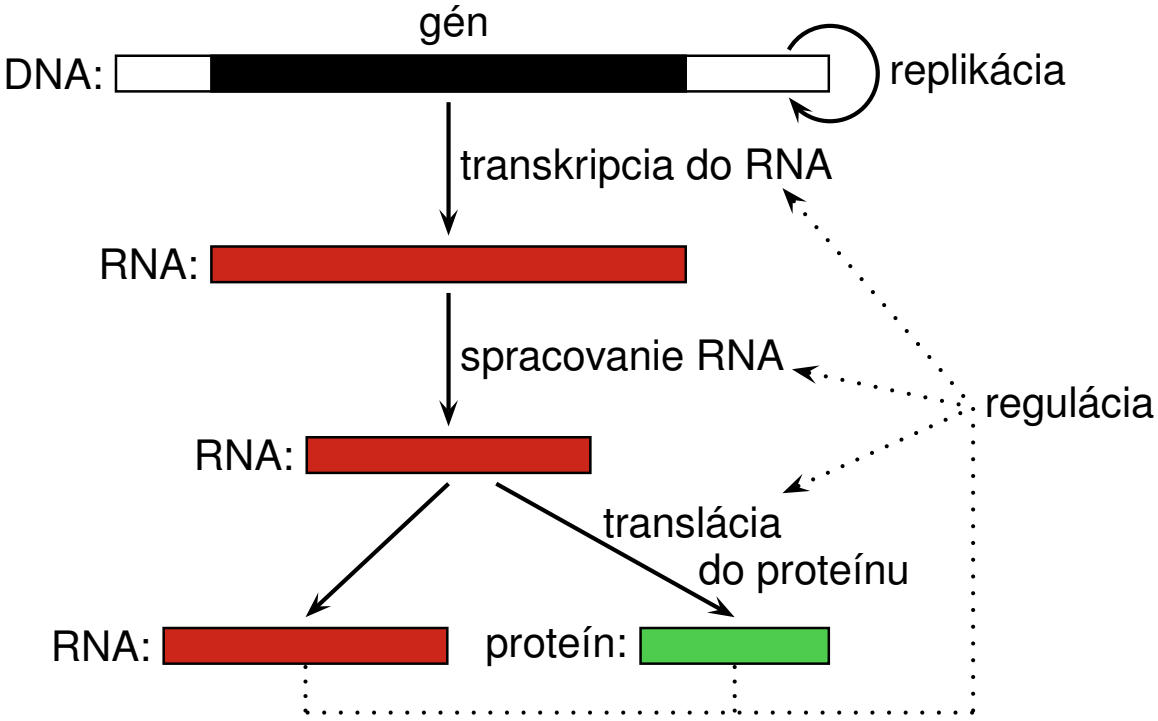
Broňa Brejová

20.11.2014

Aká informácia je uložená v DNA?

Gény: Predpisy na tvorbu proteínov a funkčných RNA molekúl.

Riadenie ich expresie: kedy a koľko sa má tvoriť.

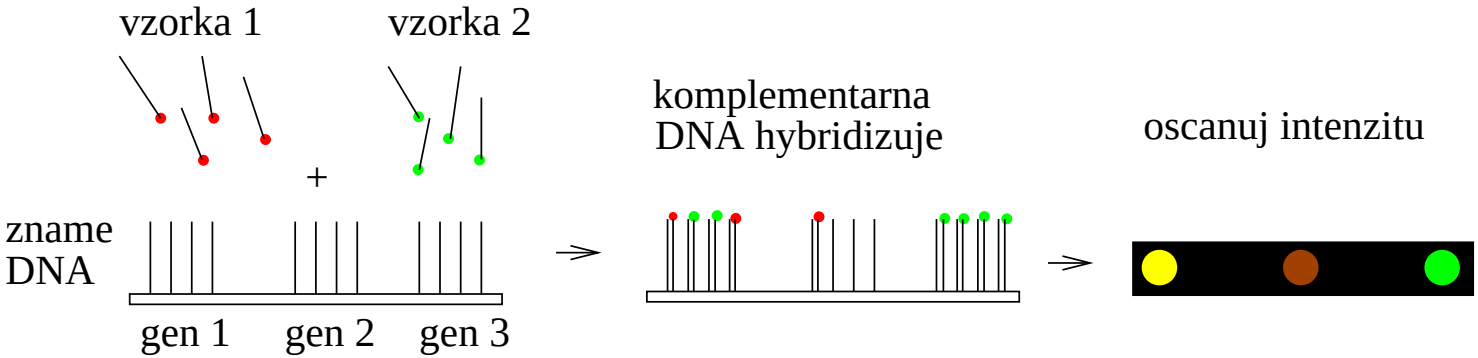


Regulácia na úrovni transkripcie, spracovania, translácie, posttranslačných modifikácií, . . .

Ciele

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

Technológia: expression array, microarray



Meranie množstva mRNA prítomnej v bunke pre **veľa génov** naraz.

Zopakujeme za rôznych podmienok.

Alternatíva: RNA-seq

sekvenujeme RNA extrahovanú z bunky,

mapujeme na genóm, hĺbka pokrytia zodpovedá úrovni expresie

Príklad expression array dát

Pomer expresie génu v meranej a kontrolnej vzorke fg/bg

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

...

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

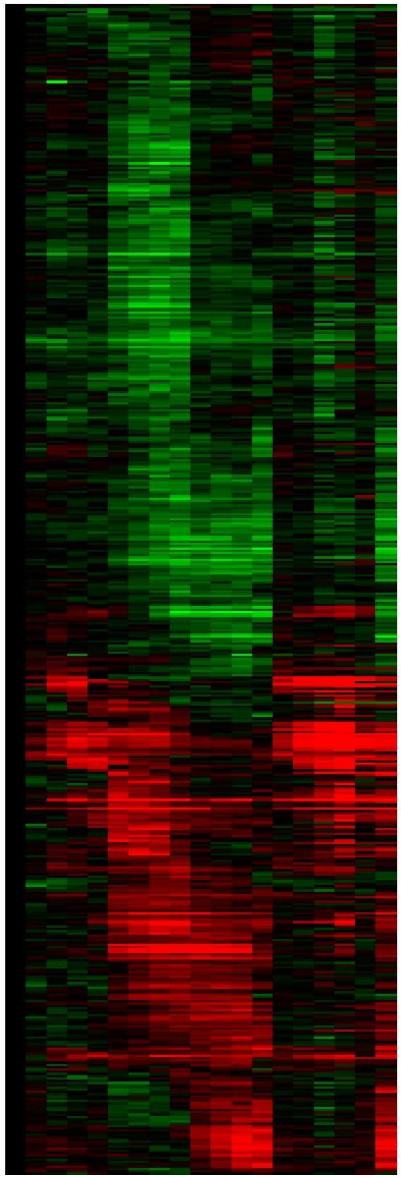
Vizualizácia

Červená: fg>bg

Zelená: fg<bg

517 génov (z 8600)

19 experimentov



Dnes: iný typ dát

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

Všetky ostatné prednášky: pracujeme so sekvenciami

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

Prvá sada problémov: predspracovanie dát

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

Merania z microarray 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 $d(x,y) = 1 - C(x,y)$

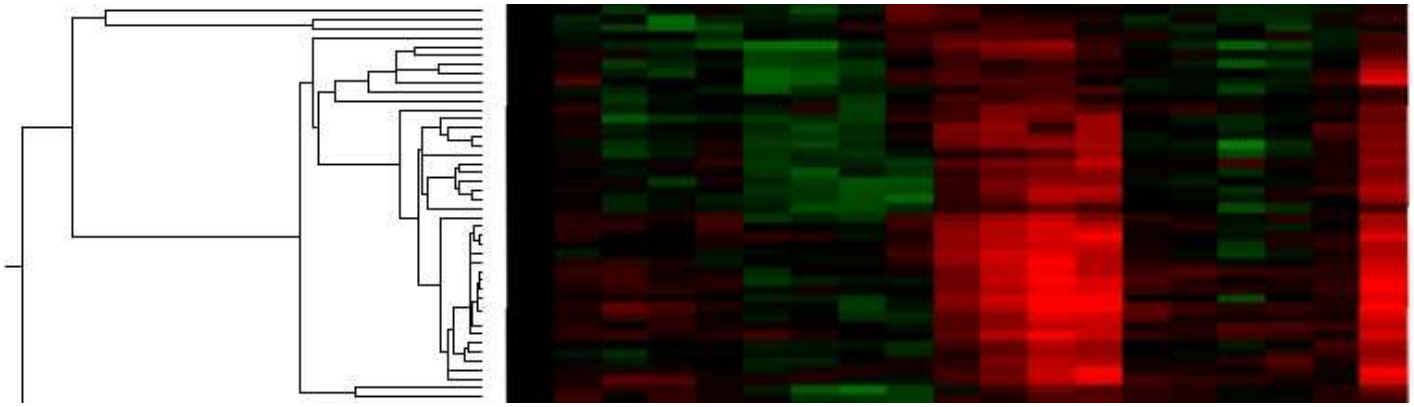
Aj iné možnosti, napr. Euklidovská vzdialenosť

Hierarchické zhlukovanie

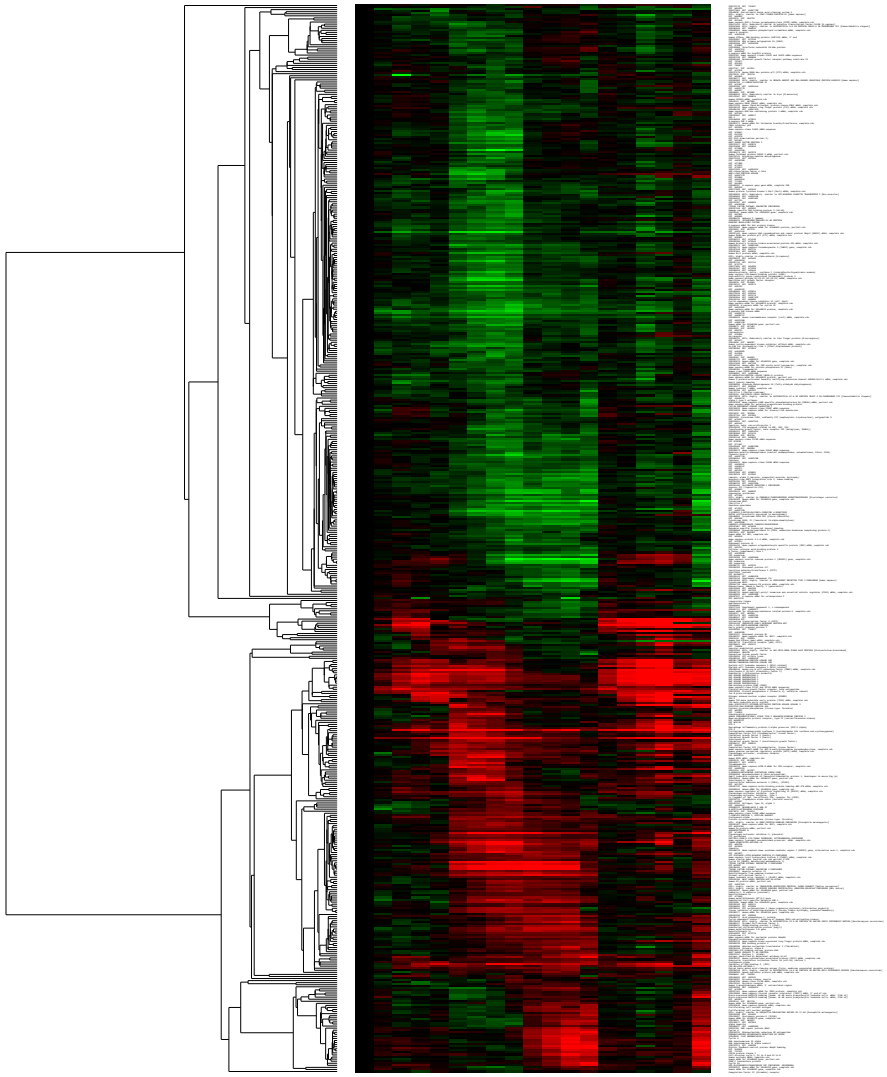
- Podobné na metódu spájania susedov vo fylogenetických stromoch
- Začneme s každým génom v samostatnej skupinke
- Nájdeme dve najbližšie skupinky a spojíme ich do jednej
- Opakujeme, kým nie sú všetky gény spolu
- Vzdialenosť skupiniek: napr. vzdialenosť najbližších génov z jednej a druhej, alebo priemer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

Príklad



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

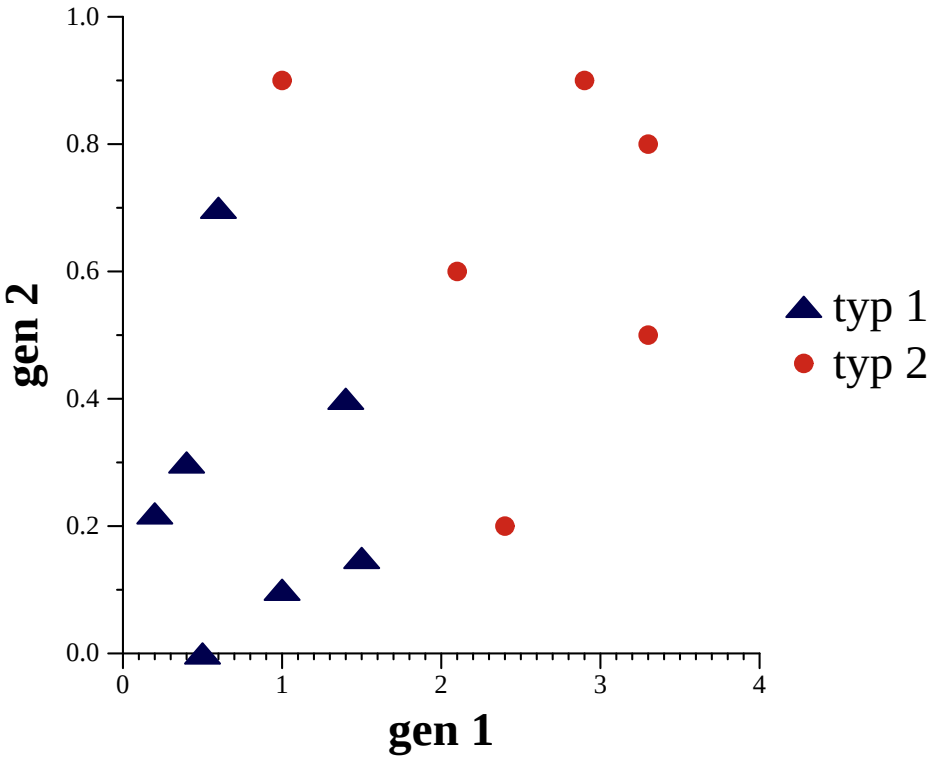


Klasifikácia

- 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:



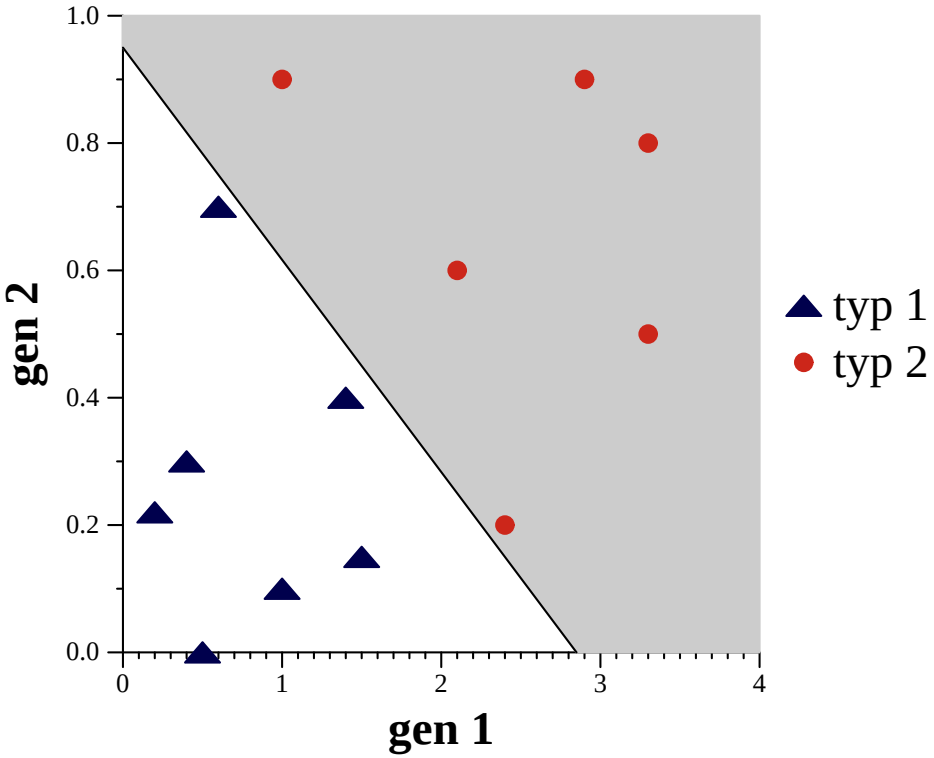
Typ vzorca: lineárne funkcie (lineárny diskriminant)

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:



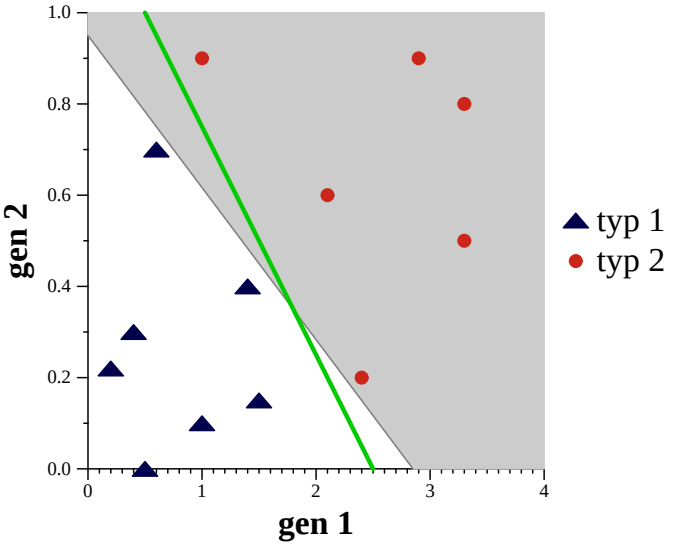
$a = 1, b = 3, c = -2.85$

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

Neural networks, 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

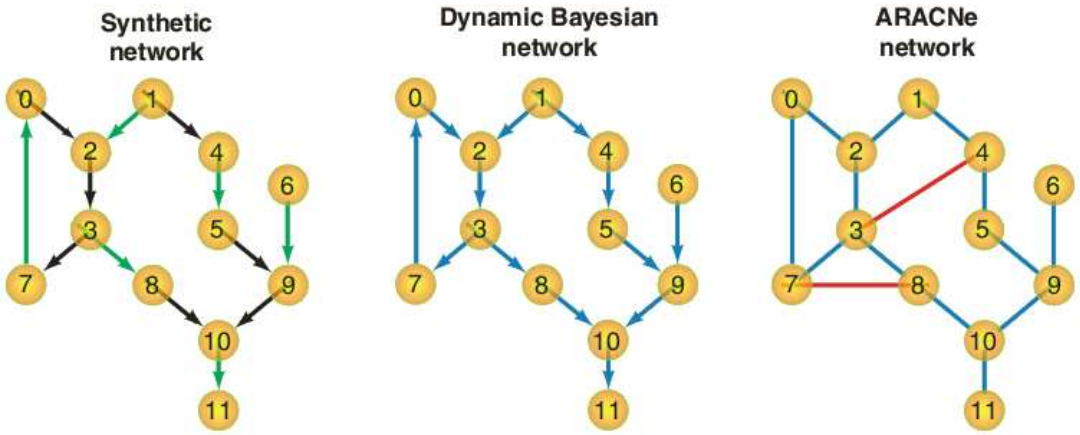
podobne ako stav v HMM

Regulačné siete z microarray dát

Vstup: Séria microarray experimentov, možno so známymi podmienkami (časové rady, delečný mutant)

Výstup: regulačná sieť, vrcholy sú gény, orientovaná hrana z A do B ak A reguluje B

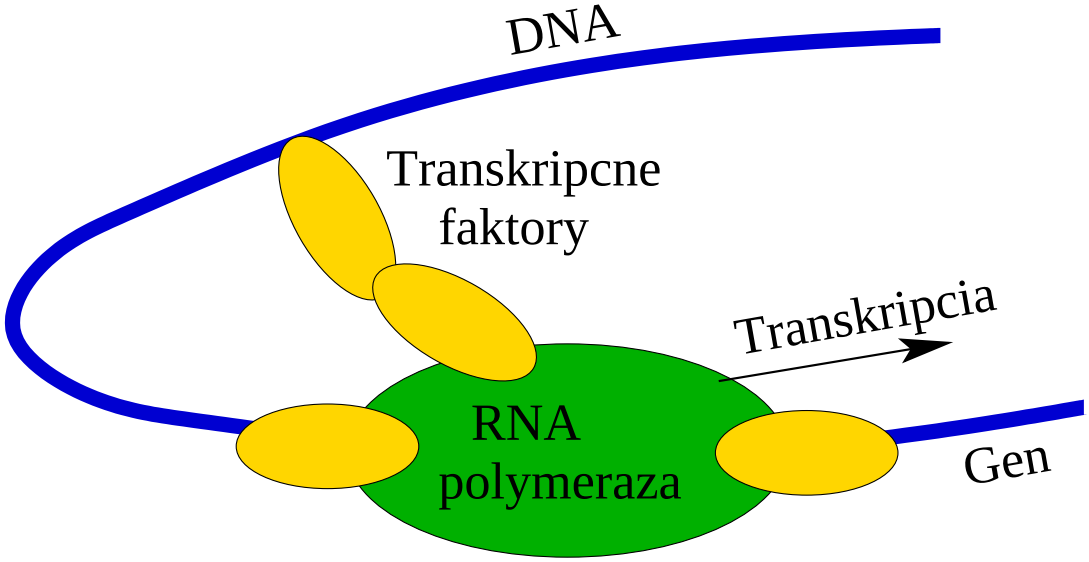
Podobnosť profilov expresie nám môže dať neorientované hrany
Chceme vylúčiť hrany, ktoré vznikli tranzitivitou
Chceme správne orientovať hrany (ťažký problém)



Hartemink - Med. Phys, 2003

Transkripčné faktory (TF)

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



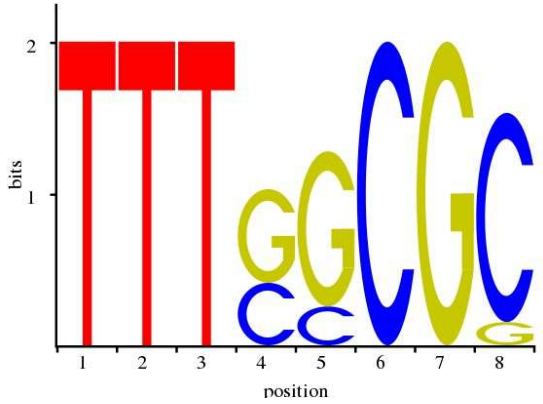
Človek má vyše 2000 TF-ov

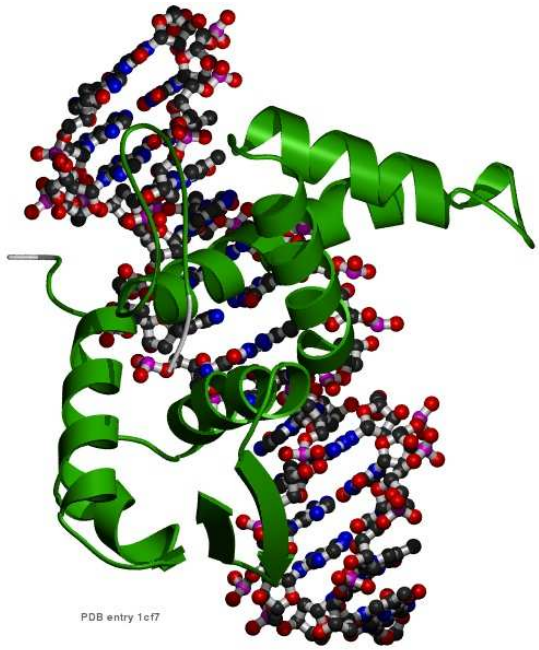
Môžu zvyšovať alebo znižovať mieru expresie,

fungovať v skupinách

Príklad: transkripčný faktor E2F1

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





- 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 skórovacej matice

Zrátame **počty**, pridáme **pseudocount** (napr. 0.5)

A	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
C	0.5	0.5	0.5	4.5	2.5	10.5	0.5	9.5
G	0.5	0.5	0.5	6.5	8.5	0.5	10.5	1.5
T	10.5	10.5	10.5	0.5	0.5	0.5	0.5	0.5

Zrátame **frekvencie**, napr. $e_1(A) = 0.5/12 = 0.0417$

Nulová hypotéza napr. $q(A) = q(T) = 0.3$, $q(C) = q(G) = 0.2$

Skóre bázy x na pozícii i je $s_i(x) = \log(e_i(x)/q(x))$

V príklade použitý prirodzený logaritmus (ln)

Napr. $s_1(A) = \ln(0.0417/0.3) = -2.0$

Hľadanie výskytov v genóme

- Hľadanie motívu v genóme: skús každú pozíciu, či je výskytom
- Väčšinou veľa falošných výskytov
- Vieme spočítať E-value: 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 nové väzobné miesta výpočtovými metódami?

- Zoberieme skupinu génov, o ktorých máme dôvod predpokladať, že sú regulované tým istým transkripčným faktorom (napr. skupinu génov s podobným profilom expresie)
- Vezmeme oblasti DNA sekvencie, ktoré by mohli obsahovať väzobné miesto tohto transkripčného faktoru (napr. 500 báz pred začiatkom každého z týchto génov)
- Snažíme sa nájsť **čo najšpecifickejší** motív, ktorý sa vyskytuje vo všetkých týchto sekvenciách resp. sa vyskytuje **častejšie, ako by sme očakávali.**

Príklad: Consensus Pattern Problem (CPP)

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

- **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?

- **Kompromis:** Skúšame všetky konsenzus sekvencie ℓ podreťazcov.

  PTAS (polynomial-time approximation scheme)

Praktickejší prístup k hľadaniu motívov

Pravdepodobnostný model generujúci sekvenciu S pomocou matice frekvencií báz v motíve M 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 M

Tento model definuje rozdelenie $\Pr(S \mid M)$.

Praktickejší prístup k hľadaniu motívov

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

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

- 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 M
  (napr. zostavenú podľa jedného okna dĺžky L)
- **Iterácia:**
  1. Priraď každej pozícii j v sekvencii S_i váhu $s_{i,j}$, ktorá zodpovedá pravdepodobnosti, že na pozícii $S_i[j]$ začína výskyt motívu M.
  2. Spočítaj M zo všetkých možných výskytov v $S_1, \ldots, S_k$ váhovaných podľa $s_{i,j}$

Iterácie zvyšujú vierohodnosť dát, kým nedojde ku konvergencii.
Skúšame veľakrát z rôznych počiatočných M

Ako nájsť väzobné miesta experimentálne?

Chromatin immunoprecipitation (ChIP)

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

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

Problém: zistíme len približnú polohu väzobného miesta

Zhrnutie

- Microarray nám môže dať informácie o úrovni expresie veľa génov naraz
- 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)
- Microarray dáta pomáhajú zostaviť regulačné siete
- Väzobné motívy môžeme reprezentovať rôznym spôsobom
  (reťazec, regulárny výraz, skórovacia matica)
- Tieto motívy nie sú dosť špecifické, preto sa ťažko rozpoznávajú ich výskyty v genóme
- EM algoritmus na hľadanie nových motívov v koregulovaných sekvenciách

Organizačné poznámky

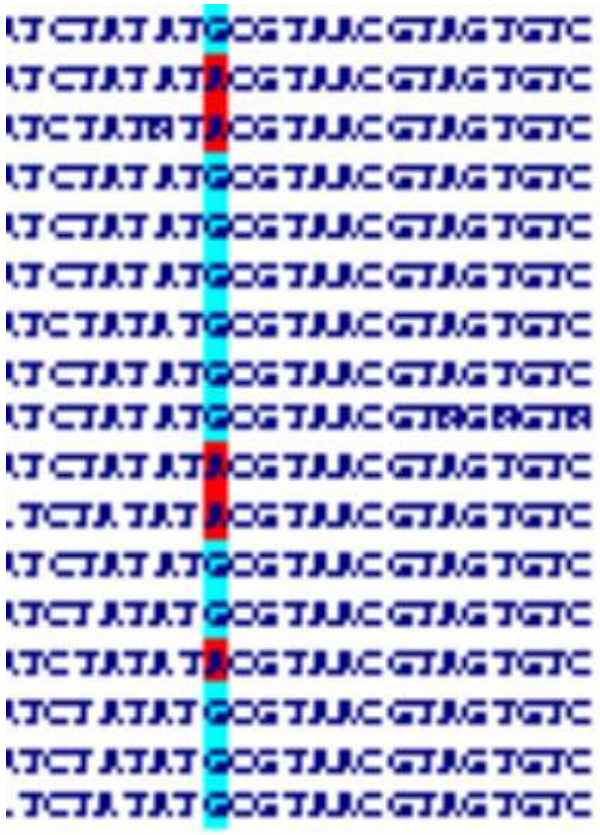
- DÚ 3 do stredy 17.12.
- Dnes posledná prednáška s novým učivom a posledné cvičenia
- Štvrtok 18.12. nepovinné prezentácie journal clubu
  – ktoré skupiny chcú?
- Piatok 19.12. návrhy projektov e-mailom (ak chcete robiť projekt)
- Piatok 19.12. správy zo journal clubu
- Písomná skúška: iba jeden riadny termín
  – kedy?
- 23.1. odovzdanie projektov
- Body z domácich úloh a journal clubu oznámime elektronicky (sledujte oznamy na Piazze)

Polymorfizmus a populačná genetika

Tomáš Vinař

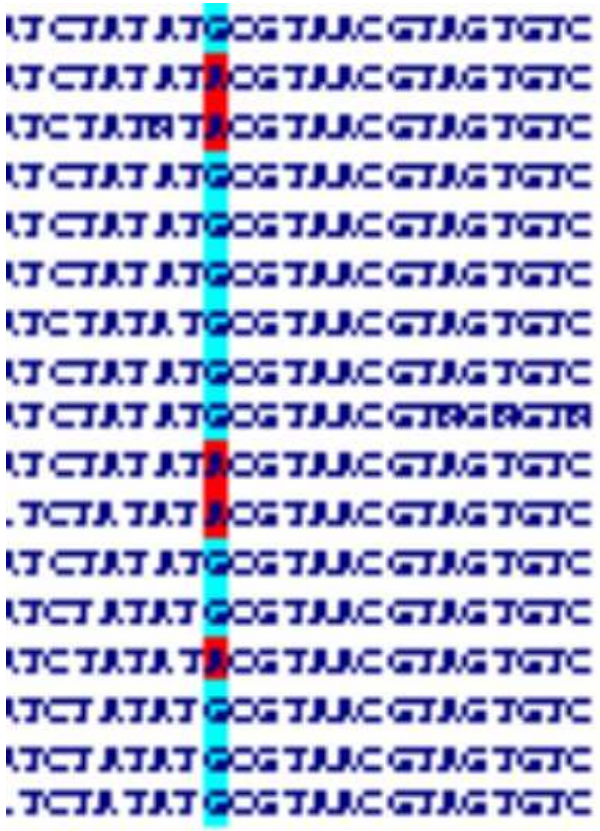
11.12.2014

Populačná genetika



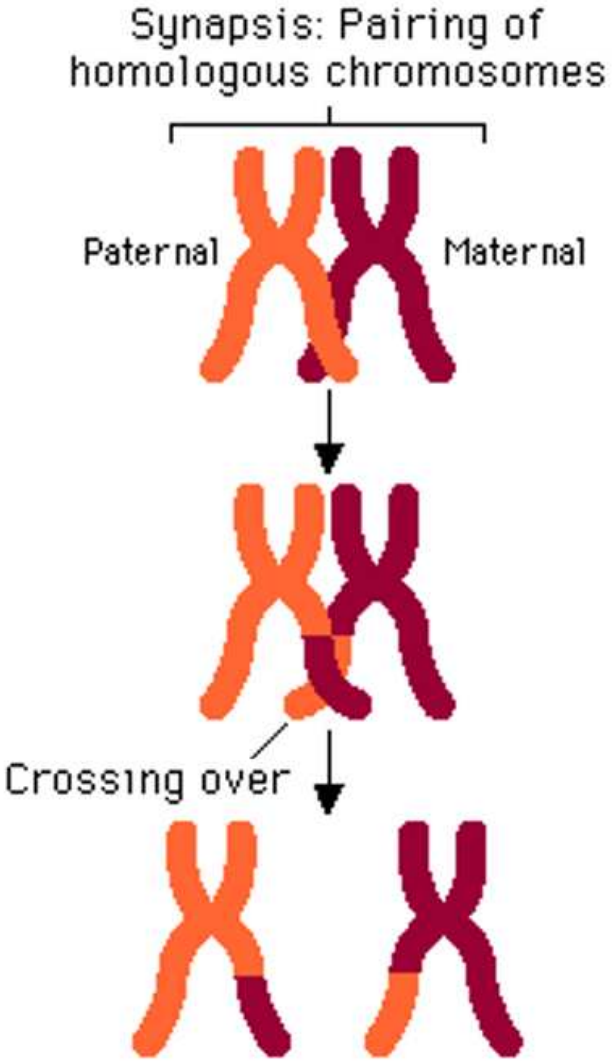
- Rôzne jedince toho istého druhu nemajú identický genóm
- Tieto rozdiely vplývajú na fenotyp (výzor, správanie, choroby, . . .)
- Genómy viacerých jedincov môžeme sekvenovať a porovnávať s referenčnou verziou
- História a charakter populácie (podpopulácie, migrácia, historická veľkosť populácie)
- Úloha jednotlivých genetických rozdielov
  (škodlivé mutácie – deleterious,
  priaznivé mutácie – advantageous)

SNPy (Single Nucleotide Polymorphisms)



- Malá zmena na správnom mieste v DNA spôsobí veľké fenotypické zmeny
- SNP: jednobázová variabilita medzi jedincami ($> 1\%$ jedincov)
- Obvykle iba 2 formy: **väčšinová** a **menšinová** alela

Stručný úvod do ľudskej genetiky



- Človek je **diploidný**: má v bunke po dva chromozómy 1...22
  plus pohlavné chromozómy X,X alebo X,Y
- Jeden chromozóm z páru od matky, jeden od otca
- Pre daný SNP s alelami a, A môže byť **homozygot** (aa alebo AA), alebo **heterozygot** (aA)
- Cca 1-3 **rekombinácie** v 1 ľudskom chromozóme počas meiózy (tvorba pohlavných buniek)

Ako vznikajú SNPy - jednoduchý model (Wrightov-Fisherov)

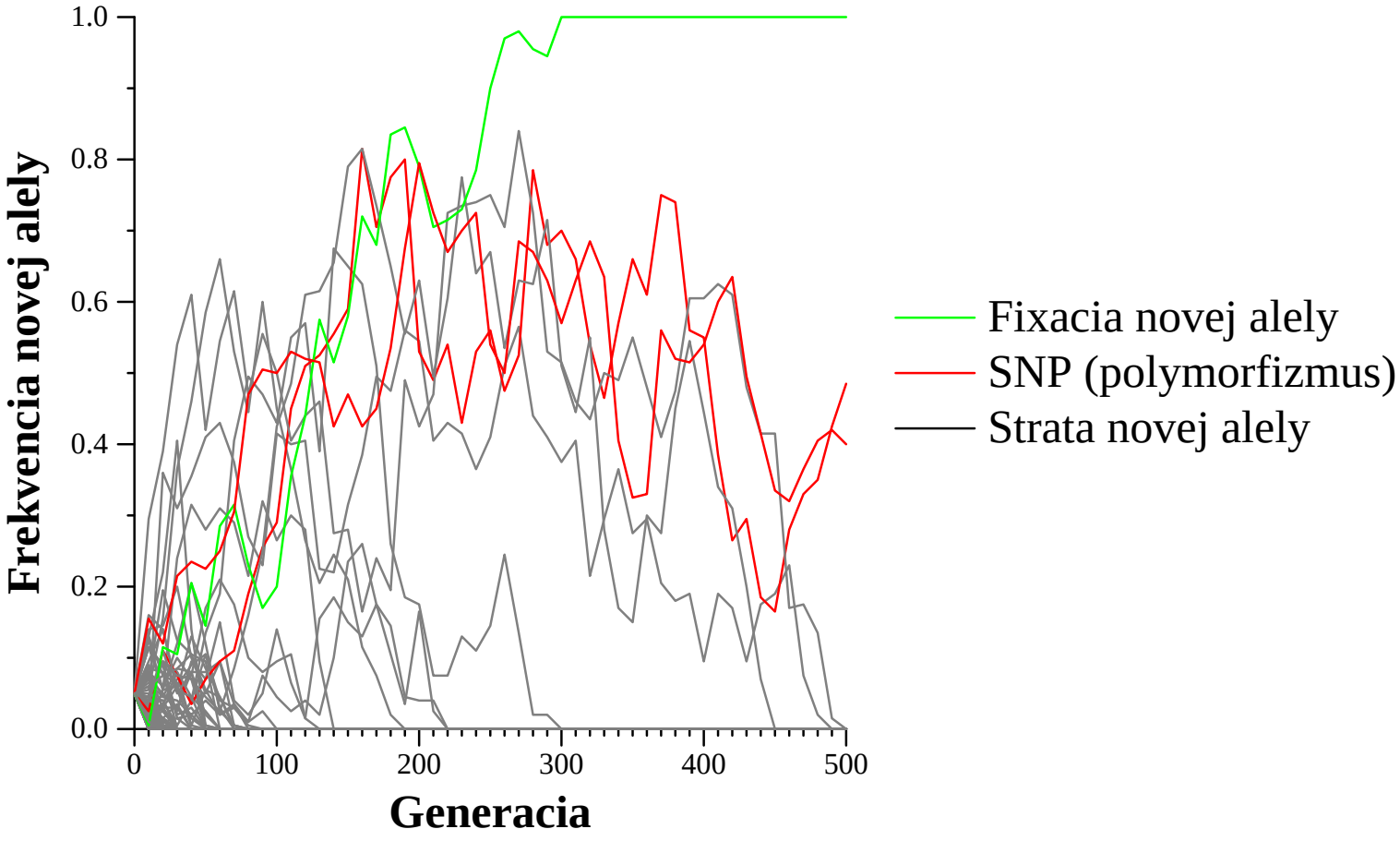
- Populácia N haploidných organizmov
- Jedna alela (forma A alebo a)
- Nová generácia vzniká "skopírovaním" náhodného rodiča (random mating), bez vplyvu prirodzeného výberu (no selection)
- Nech X_t je počet výskytov novej alely a v generácii t
- **Markovovský reťazec** so stavmi $X_t \in \{0, 1, \ldots, N\}$

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

- Stavy 0 a N sú **pohlcujúce** (ak $X_t = n$, tak aj $X_{t+1} = n$) žiadny iný pevný bod

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** (selection).
- Zložitejšie pravdepodobnostné modely
- Odhady parametrov na základe údajov o súčasnej diverzite (frekvencii jednotlivých menšinových alel a pod.)

Stabilný polymorfizmus - výhody diverzity

- Farbosleposť opíc nového sveta (New World monkeys)
  - gén na chromozóme X, dve alely citlivé na iné časti spektra
  - samičky s aa, samčeky s a nevidia červenú časť spektra
  - samičky s AA, samčeky s A nevidia zelenú časť spektra
  - samičky s aA vidia trichromaticky

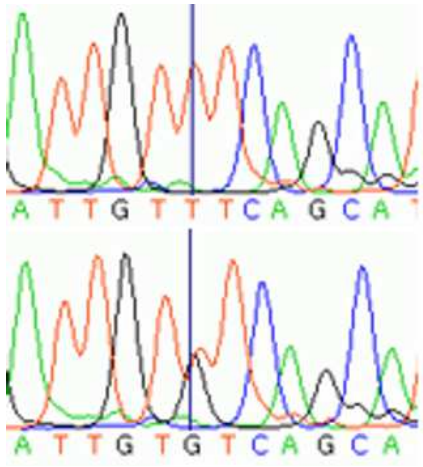
- Kosáčiková anémia (sickle cell anemia)
  - Aa: normálna bunka + **imunita voči malárii**
  - aa: poškodené krvinky

Haplotypovanie v diploidnom genóme

- Dve kópie chromozómu $\Rightarrow$ dve kópie každého SNPu:

```
haplotyp 1:    001100010110
haplotyp 2:    001000011110
               ------------
genotyp:       001200012110
```

- **Problém:** Bežnými metódami získame väčšinou iba genotyp



homozygot

heterozygot

Od genotypov k haplotypom

Jeden genotyp: príliš málo informácie

```
haplotyp 1:   001100010110      haplotyp 1:   001100011110
haplotyp 2:   001000011110      haplotyp 2:   001000010110
              ------------                    ------------
genotyp:      001200012110      genotyp:      001200012110
```

Skupina genotypov: ich haplotypy vznikli z malého množstva haplotypov (founding population) malým počtom mutácií
$\Rightarrow$ **počet rôznych haplotypov v populácii by mal byť malý**
(krátky región, s malou pravdepodobnosťou rekombinácie)

Od genotypov k haplotypom

Vstup: veľa genotypov

Predpoklad: malý počet haplotypov v populácii

Haplotype inference by pure parsimony (HIPP): Nájdite najmenší počet haplotypov, z ktorých je možné poskladať všetky vstupné genotypy.

Príklad:

```
Genotypy:      Riešenie 1:       Riešenie 2:
02120          (01110,00100)     (01110,00100)
22110          (00110,11110)     (01110,10110)
20120          (10110,00100)     (10110,00100)
               ------------      ------------
               5 haplotypov      3 haplotypy
```

Nanešťastie: **HIPP je NP-ťažký problém**

Prístupy k haplotypovaniu

- Heuristické pravidlo: Clarkov algoritmus [Clark, 1990]
  (heuristika pre HIPP)
- Celočíselné programovanie [Gusfield, 2003]
- Perfect phylogeny haplotyping [Gusfield, 2002]
  (pridanie ďalšieho predpokladu umožňuje efektívny algoritmus)
- Pravdepodobnostné modely: PHASE [Stephens et al., 2001],
  Haplotyper [Niu et al., 2002]

Overovanie haplotypovania

- Experimentálne haplotypovanie je drahé
- Genotypovanie trojíc rodičia, dieťa

```
Matka        0  2  2  2
Otec         2  1  2  2
Dieťa        2  2  2  1

Dieťa (M)    0  0  x  1
Dieťa (O)    1  1  y  1

Matka (D)    0  0  x  1
Matka  2     0  1  y  0

Otec (D)     1  1  y  1
Otec  2      0  1  x  0
```

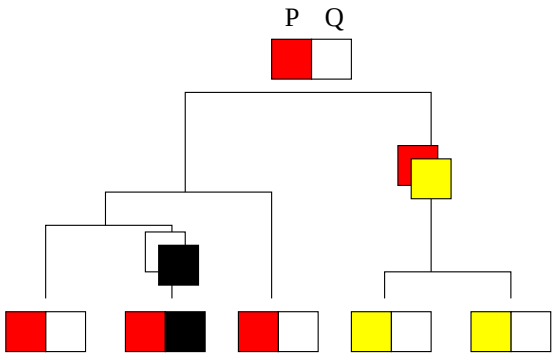
Závislosti medzi SNPmi

Uvažujme SNP s alelami p/P a ďalší SNP s alelami q/Q.
Nameriame frekvencie haplotypov $\Pr(pq)$, $\Pr(PQ)$, $\Pr(pQ)$, $\Pr(Pq)$

Na rozdielnych chromozómoch:

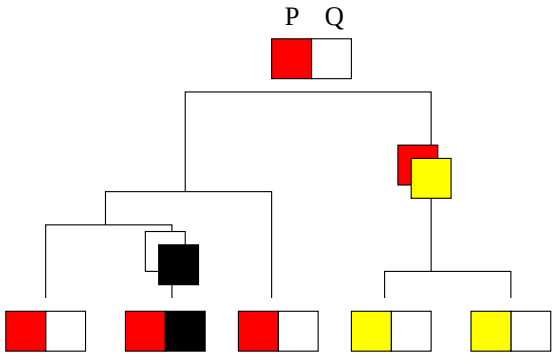
- Pravdepodobnosti výskytu jednotlivých alel sú nezávislé
- $\Pr(pq) = \Pr(p)\Pr(q)$, $\Pr(PQ) = \Pr(P)\Pr(Q)$, atď
- **linkage equilibrium (LE)**

Blízko seba na tom istom chromozóme:



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

Väzbová nerovnováha (linkage disequilibrium, LD)



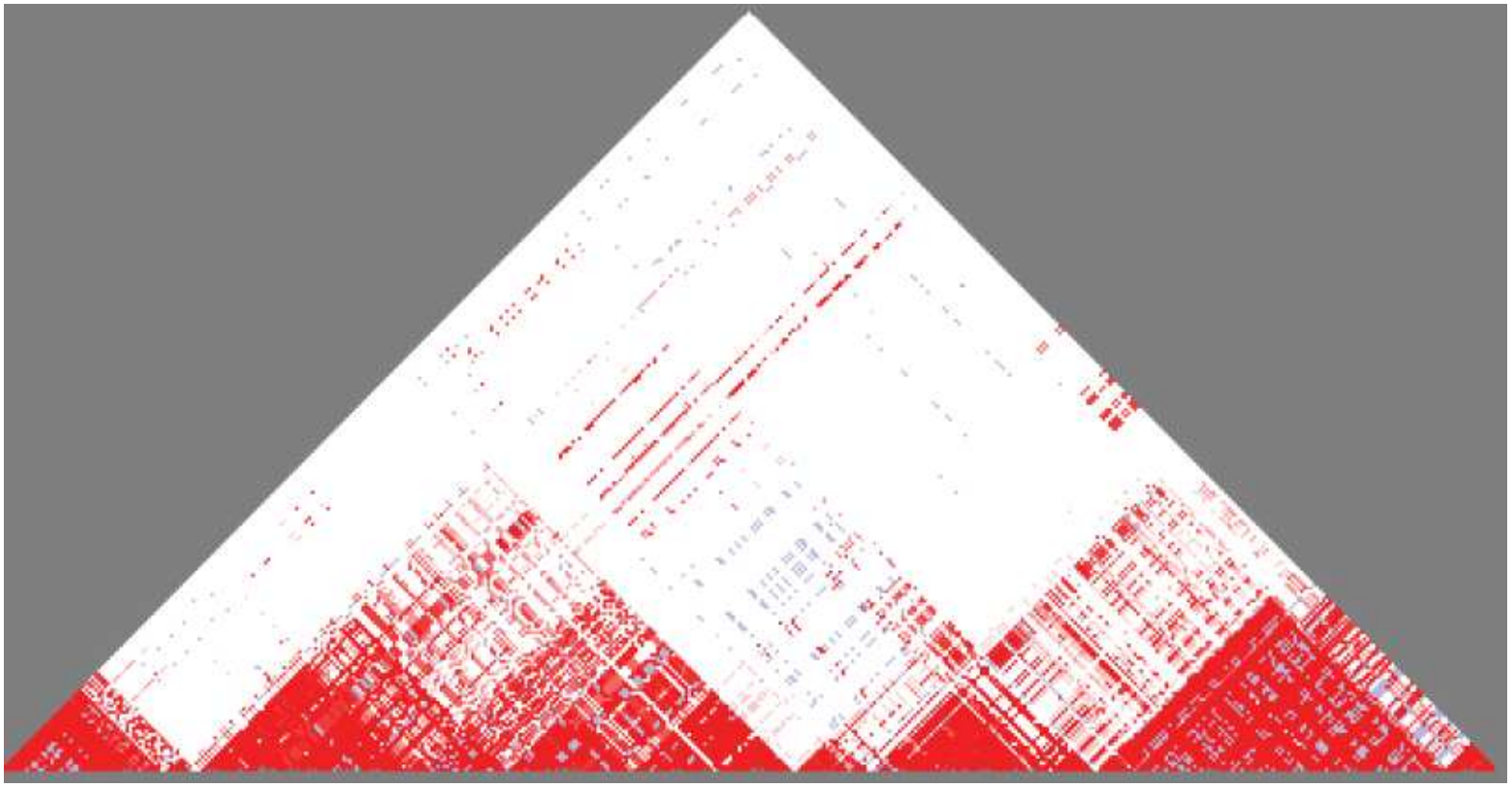
Rekombinácia znižuje LD

Ak predpokladáme rovnomernú rekombináciu:

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

Linkage disequilibrium v ľudskom genóme

[The International HapMap Consortium, 2005]



Encode región ENm014 (500kB, chr 7), 90 ľudí Utah

Miera linkage disequilibria (LD)

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

Nameriame frekvencie haplotypov $\Pr(pq)$, $\Pr(PQ)$, $\Pr(pQ)$, $\Pr(Pq)$

$$\boxed{D = \Pr(pq)\Pr(PQ) - \Pr(pQ)\Pr(Pq)}$$

Ak dva SNPy sú v stave LE, $D = 0$

lebo $\Pr(pq) = \Pr(p)\Pr(q)$, $\Pr(PQ) = \Pr(P)\Pr(Q)$, atď

Ako detegovať LD?

- $D = \Pr(pq)\Pr(PQ) - \Pr(pQ)\Pr(Pq)$
- Veľkosť D závisí od frekvencie jednotlivých alel (p a q)
- Aby sme mohli **porovnávať** medzi rôznymi SNPmi, potrebujeme normalizáciu
- Uvažujme nasledujúce veličiny (n počet jedincov vo vzorke):

$$\rho = \frac{D}{\sqrt{\Pr(p)\Pr(P)\Pr(q)\Pr(Q)}}$$

$$\chi^2 = \rho^2 n = \frac{nD^2}{\Pr(p)\Pr(P)\Pr(q)\Pr(Q)}$$

- Ak sú SNP-y v stave LE, χ^2 sa správa podľa $\chi^2(1)$ distribúcie $\Rightarrow$ ak $\chi^2 > 3.841$, P a Q sú v stave disekvilibria ($P < 0.05$)

Príklad:

- 1000 jedincov s nasledujúcimi haplotypmi:

	Q	q	
P	474	611	0.543
p	142	773	0.458
	0.308	0.692	

- $\chi^2 = 2000 \frac{(\Pr(PQ)\Pr(pq) - \Pr(Pq)\Pr(pQ))^2}{\Pr(p)\Pr(P)\Pr(q)\Pr(Q)} = 2000 \frac{0.0699^2}{0.053} = 184.78$
  Môžeme vylúčiť hypotézu, že P a Q sú v stave LE

- Ak by sa štúdie bolo zúčastnilo iba 15 účastníkov (30 haplotypov) s podobnými výsledkami, nebolo by možné LE vylúčiť (hodnota χ^2 by nebola štatisticky významná)

Mapovanie asociácií (Trait/Disease Association Mapping)

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

Testovanie asociácie jedného SNPu

Počet haplotypov (chr15:44,228,468):

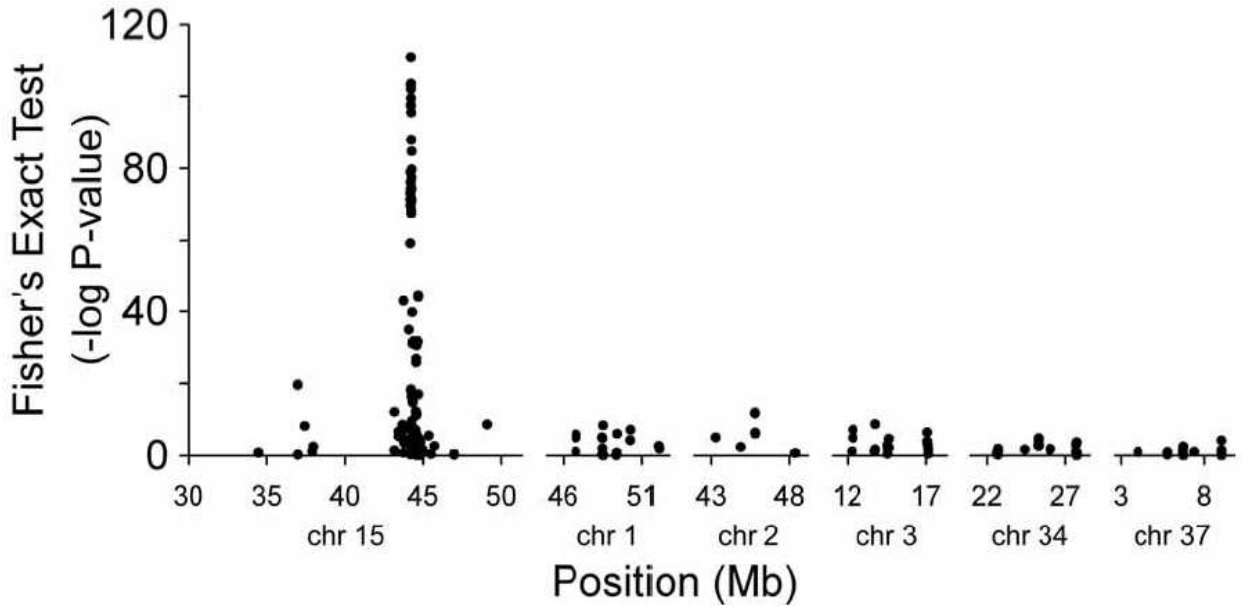
	pôvodná alela	odvodená alela	spolu
malý pes (< 9 kg)	14	535	549
veľký pes (> 31 kg)	339	38	377
spolu	353	573	

Fisherov test: (Fisher's exact test) Testuje nezávislosť premenných reprezentovaných riadkami a stĺpcami kontingenčnej tabuľky.

V tomto prípade: $P = 2.2 \times 10^{-16}$

Používajú sa aj zložitejšie štatistické metódy

Hľadanie asociácií v celom genóme (Whole-Genome Association Scan)

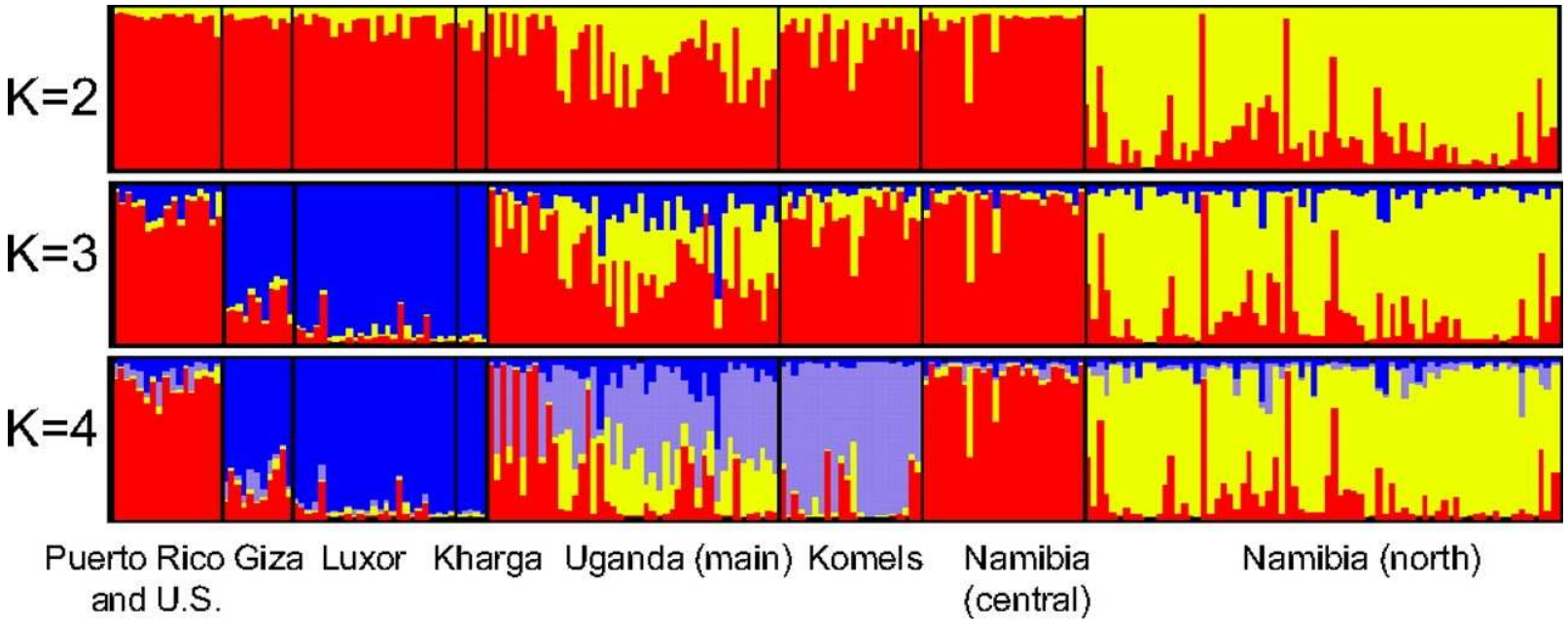


- V prípade štúdie veľkosti psov: WGAS identifikoval 84 kB región
- Pozíciu ďalej treba spresniť ďalšími experimentami
- **Malé LD bloky** $\Rightarrow$ potreba veľkého rozlíšenia SNPov
- **Veľké LD bloky** $\Rightarrow$ príliš veľké výsledné regióny

Štruktúra populácie

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

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

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

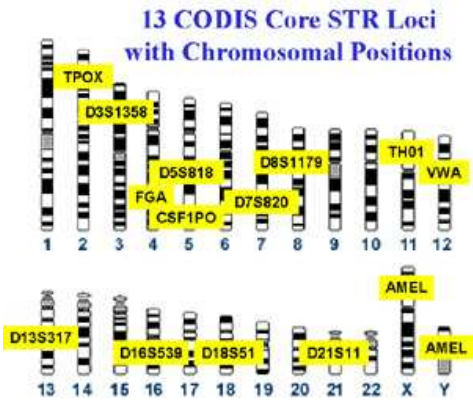
konverguje k hodnote $E[Q \mid X]$

Zhrnutie

- **SNPy (single nucleotide polymorphizms)** priebežne vznikajú a zanikajú v populáciách
- Ich frekvencia ovplyvnená navyše prirodzeným výberom
- Resekvenovaním diploidného organizmu získame **genotyp**, ktorý je potrebné výpočtovými metódami rozložiť na **haplotypy**
- Bez rekombinácie korelácia medzi SNPmi na tom istom chromozóme (**linkage disequilibrium**)
- Rekombinácie vytvárajú v genóme LD bloky
- Prítomnosť LD blokov možno využiť pri mapovaní asociácií znakov (**whole-genome association mapping**)
- Pri LD 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)

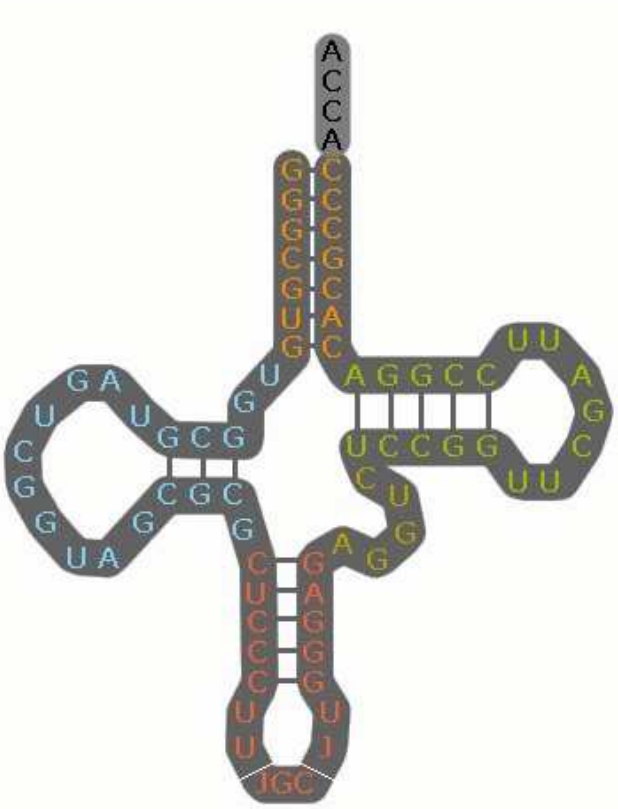
Organizačné poznámky

- DÚ 3 do stredy 17.12.
- Budúci štvrtok posledná prednáška s novým učivom a posledné cvičenia
- Štvrtok 18.12. nepovinné prezentácie journal clubu
- Piatok 19.12. návrhy projektov e-mailom (ak chcete robiť projekt)
- Piatok 19.12. správy zo journal clubu
- Písomná skúška: iba jeden riadny termín
- 23.1. odovzdanie projektov
- Budúci štvrtok dohodneme:
  – či chcete prezentovať projekty (dohodnite sa v skupinách)
  – kedy bude skúška (doneste si termíny iných skúšok)

RNA

Broňa Brejová

4.12.2014



Vlastnosti RNA

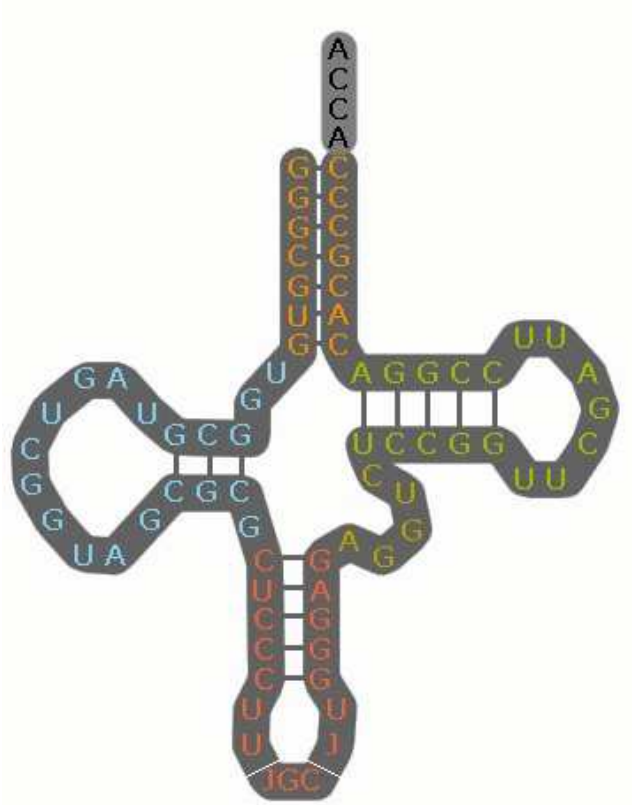
Ako sa líši od DNA?

- 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

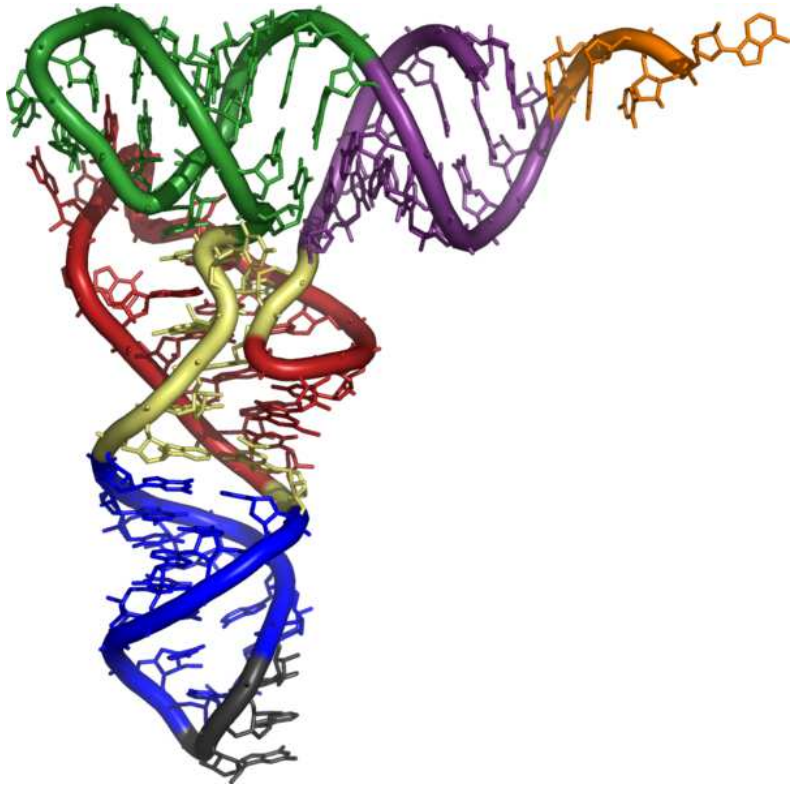
Štruktúra RNA

Príklad: transferová RNA (transfer RNA)

Sekundárna štruktúra
(secondary structure):
páry nukleotidov

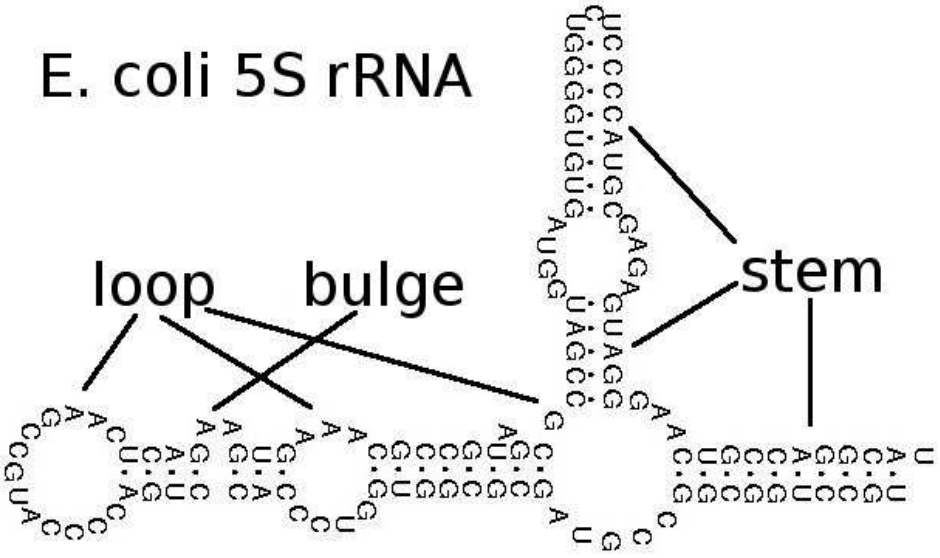


Terciárna štruktúra
(tertiary structure):
3D súradnice



Sekundárna štruktúra RNA

Prvky sekundárnej štruktúry



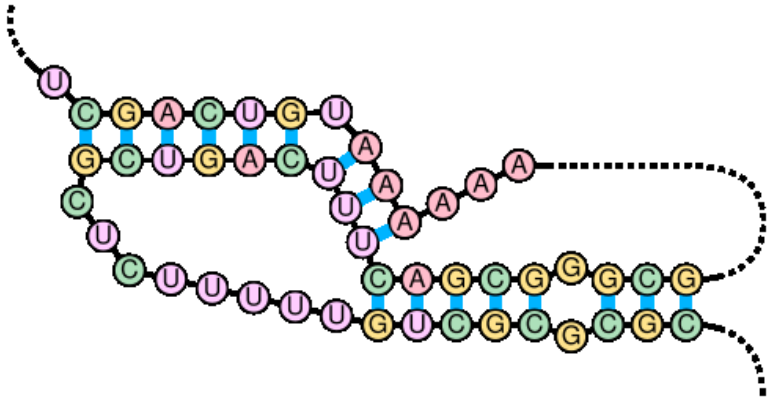
V tomto prípade spárované bázy tvoria **dobre uzátvorkovaný výraz:**

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

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

Sekundárna štruktúra RNA

Pseudouzol: výnimka z dobrého uzátvorkovania



Mnohé algoritmy na prácu so sekundárnou štruktúrou ignorujú pseudouzly.

Zhruba 1.4% RNA nukleotidových párov v pseudouzloch.

Problém: RNA secondary structure prediction

Vstup: RNA sekvencia

Cieľ: nájsť spárované bázy

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

Príklad: `((.((((()))(((.))))))))`

`GAACACAUGUAAAAUUUGUC`

Možno riešiť dynamickým programovaním: [Nussinov et al., 1978]

Majme RNA $X_1, \ldots, X_n$.

Spočítajme riešenie pre každý podreťazec $X_i, X_{i+1}, \ldots, X_j$ $(1 \leq i \leq j \leq n)$.

Nech $A[i, j]$ je maximálny počet párov v tomto podreťazci.

Príklad: $A[1, 3] = 0$ (žiaden pár v GAA),

$A[1, 4] = 1$ (v GAAC pár G-C)

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:

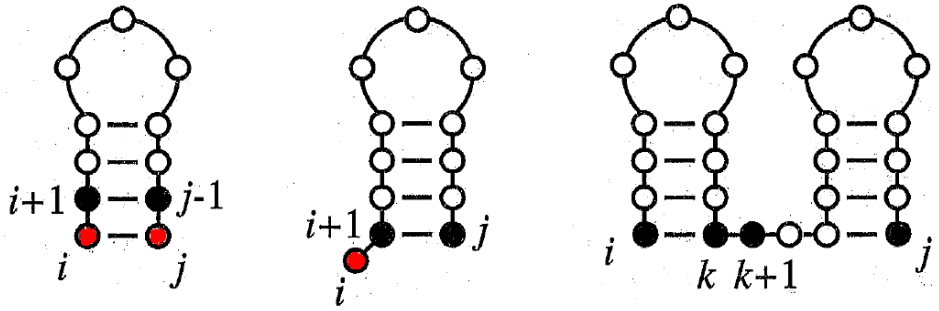
Podreťazce dĺžky 1: žiadne páry $A[i, i] = 0$

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

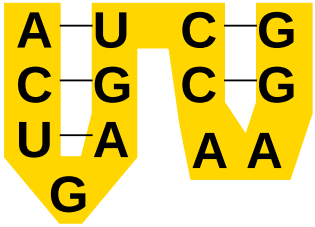


Rekurencia: $A[i,j] = \max \begin{cases} A[i+1, j-1] + c(X_i, X_j), \\ A[i+1, j], \\ \max_{k=i+1 \ldots j-1} \{A[i,k] + A[k+1, j]\} \end{cases}$

	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

Minimum free energy (MFE) folding

Realistickejšia formulácia problému určovania sek. štruktúry RNA.

Predpoklad: molekula v rovnovážnom stave s minimálnou Gibbsovou voľnou energiou (Gibbs free energy).

Energie pre niektoré sekvencie experimentálne zmerané.

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

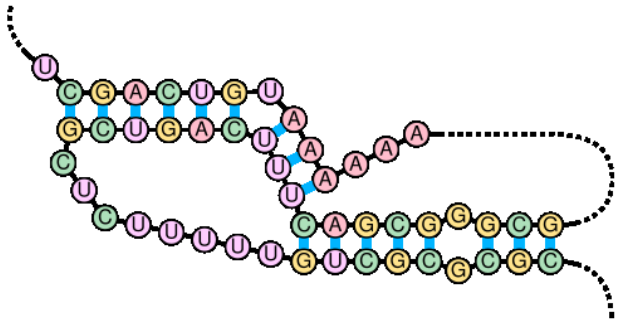
Odvedené z nameraných dát.

Príklad:

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

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

Algoritmy dovoľujúce pseudouzly



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

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

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

Pravdepodobnostné modely na predikciu štruktúry

HMM nevhodné: závislosti medzi vzdialenými spárovanými bázami.

Stochastická bezkontextová gramatika,
stochastic context free grammar (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)
$S \to aSu|uSa|cSg|gSc|aS|cS|gS|uS|Sa|Sc|Sg|Su|SS|\epsilon$

V každom kroku zvoľ jeden (napr. najľavejší) neterminál,
prepíš ho náhodne zvoleným pravidlom:

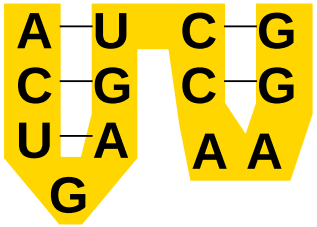
$S \to SS \to aSuS \to acSguS \to acuSaguS \to acugSaguS \to$
$acugaguS \to acugagucSg \to acugagucc Sgg \to acugagucc Sagg \to$
$acugagucc aSagg \to acugaguccaagg$

Stochastic context free grammars

Príklad:

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

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



Bázy vygenerované v jednom kroku sú spárované.
CYK dyn. prog. algoritmus nájde najpravdepodobnejšie odvodenie pre danú RNA v čase $O(n^3)$
Parametre možno trénovať zo známych RNA štruktúr, podobne ako pri hľadaní génov.

Gramatiky vs. minimalizácia energie

Výhody gramatík:

Parametre gramatík možno automaticky trénovať, netreba náročné experimenty.

Gramatiky sa dajú elegantne rozšíriť na modely viacerých sekvencií.

Nevýhody gramatík:

Nie je jednoduché zostaviť vhodnú gramatiku so zložitou sadou parametrov.

Nedosahujú takú presnosť ako minimalizácia energie.

Conditional log-linear models:

zovšeobecnené SCFG, trénovanie maximalizuje podmienenú pravdepodobnosť správnej odpovede (discriminative training).

Dosahujú lepšiu presnosť ako minimalizácia energie.

Evolúcia RNA sekvencií

Často vidíme koreláciu medzi mutáciami v spárovaných bázach. Napr. pár C-G sa zmení na G-C alebo A-U, aby sa zachovala štruktúra.

Príklad: niekoľko sekvencií z D ramena tRNA

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

Korelácie medzi spárovanými bázami zvyšujú našu dôveru v správnosť štruktúry.

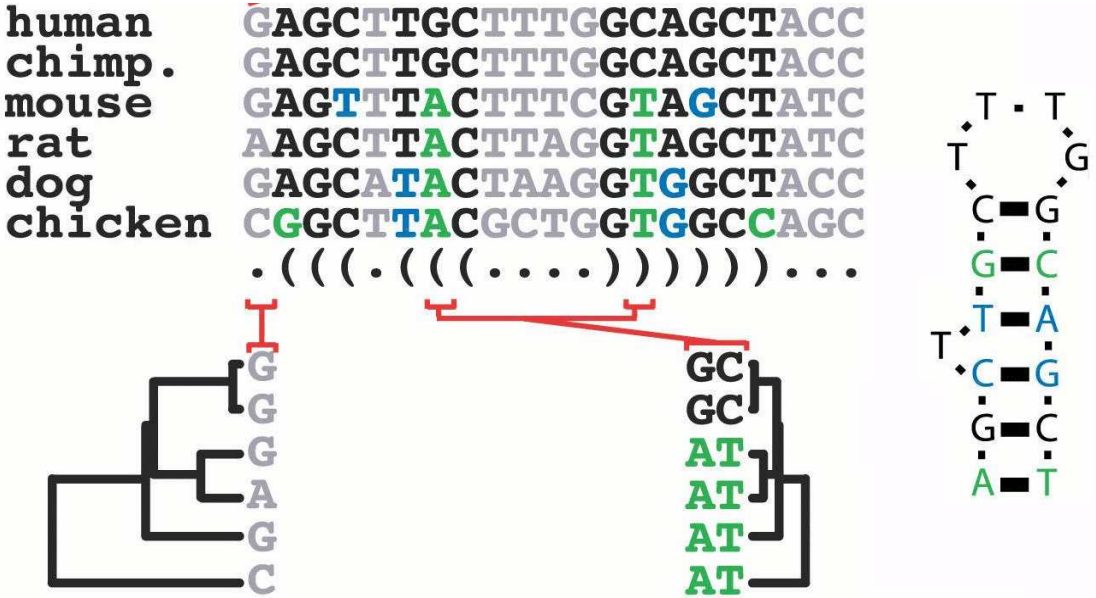
Hľadanie spoločnej štruktúry pre viacero sekvencií

- Ak sú sekvencie dostatočne podobné, môžeme ich zarovnať a potom hľadať štruktúru s veľa korelovanými pármi.

  Phylo-SCFG: namiesto jednotlivých báz emituje stĺpce zarovnania podľa fylogenetického stromu.

  Nespárované bázy emituje bežnou substitučnou maticou,

  spárované bázy substitučnou maticou dvojíc (16×16).



Hľadanie spoločnej štruktúry pre viacero sekvencií

- Ak sú sekvencie dostatočne podobné, môžeme ich zarovnať a potom hľadať štruktúru s veľa korelovanými pármi.
- Ak sú sekvencie málo podobné, nevieme spoľahlivo zarovnať, štruktúra však môže byť zachovaná.
  Môžeme hľadať zarovnanie a štruktúru súčasne.
  Presný algoritmus pomalý: $O(n^{3m})$ pre m sekvencií.
  [Sankoff, 1985]
  Zrýchlenie rôznymi heuristikami: predfiltrovanie, obmedzenie triedy sekundárnych štruktúr atď.

Hľadanie nových RNA génov v genóme

- Hľadaj úseky DNA so stabilnou sekundárnou štruktúrou (silnejší signál, ak máme zarovanie viacerých sekvencií).
- Výsledky treba normalizovať vzhľadom na dĺžku génu a GC%.
- Experimentálne overenie transkripcie

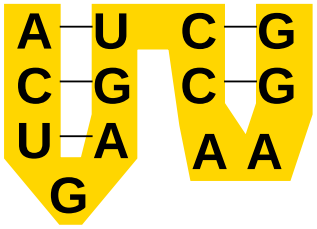
Problém: hľadanie známych typov RNA génov v genóme

- Databáza Rfam: 1372 rodín podobných RNA génov
- Pre každú rodinu zarovnanie a pravdepodobnostný model
- Obdoba Pfamu pre rodiny proteínov
- Proteínové rodiny reprezentujeme profilmi, profilovými HMM
- Nevhodné pre RNA: závislosti medzi vzdialenými pozíciami
- Používame kovariančné modely (covariance model, CM), čo je špeciálny typ SCFG

Rfam: RNA families database

Covariance model (CM): SCFG reprezentácia RNA rodiny.

Zostavíme podľa zarovnania + známej štruktúry.



$$
\begin{array}{llll}
S \to B_1 & P_1 \to aP_2u & P_4 \to cP_5g \\
B_1 \to P_1P_4 & P_2 \to cP_3g & P_5 \to gL_2c \\
 & P_3 \to uL_1a & L_2 \to aL_3 \\
 & L_1 \to gE_1 & L_4 \to aE_2 \\
 & E_1 \to \epsilon & E_2 \to \epsilon
\end{array}
$$

S =start, E_i =end

P_i =pár, L_i =nespárovaná báza vľavo, R_i =nespárovaná báza vpravo.

Ďalšie neterminály modelujú indely.

P_i, L_i, R_i emitujú bázy/páry s pravdepod. podľa stĺpca zarovnania.

Napr. $P_1 \to aP_2u|uP_2a|cP_2g|cP_2u$

Covariance model (CM)

Použitie:

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

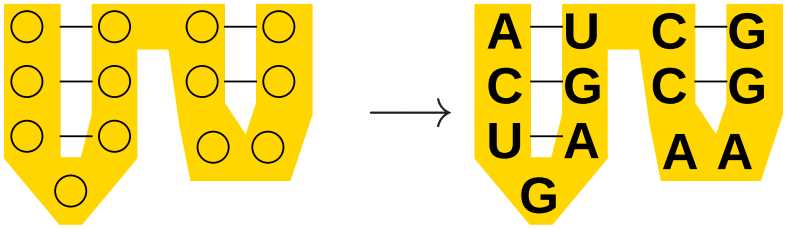
Dynamické programovanie: čas $O(MND + M_bND^2)$,
$M =$ počet neterminálov v gramatike, úmerný dĺžke zarovnania,
$M_b =$ počet bifurkácií v gramatike (zvyčajne oveľa menší ako M),
$N =$ dĺžka DNA sekvencie,
$D =$ max. dĺžka RNA génu v DNA (úmerná M).

Zrýchlenie: nájdi sľubné úseky podobné na sekvencie v RNA rodine (iba na základe podobnosti sekvencií), aplikuj CM iba na ne.

Problém: RNA secondary structure design

Daná RNA sekundárna štruktúra (párovanie).
Nájdi sekvenciu, pre ktorú je táto štruktúra optimálna.

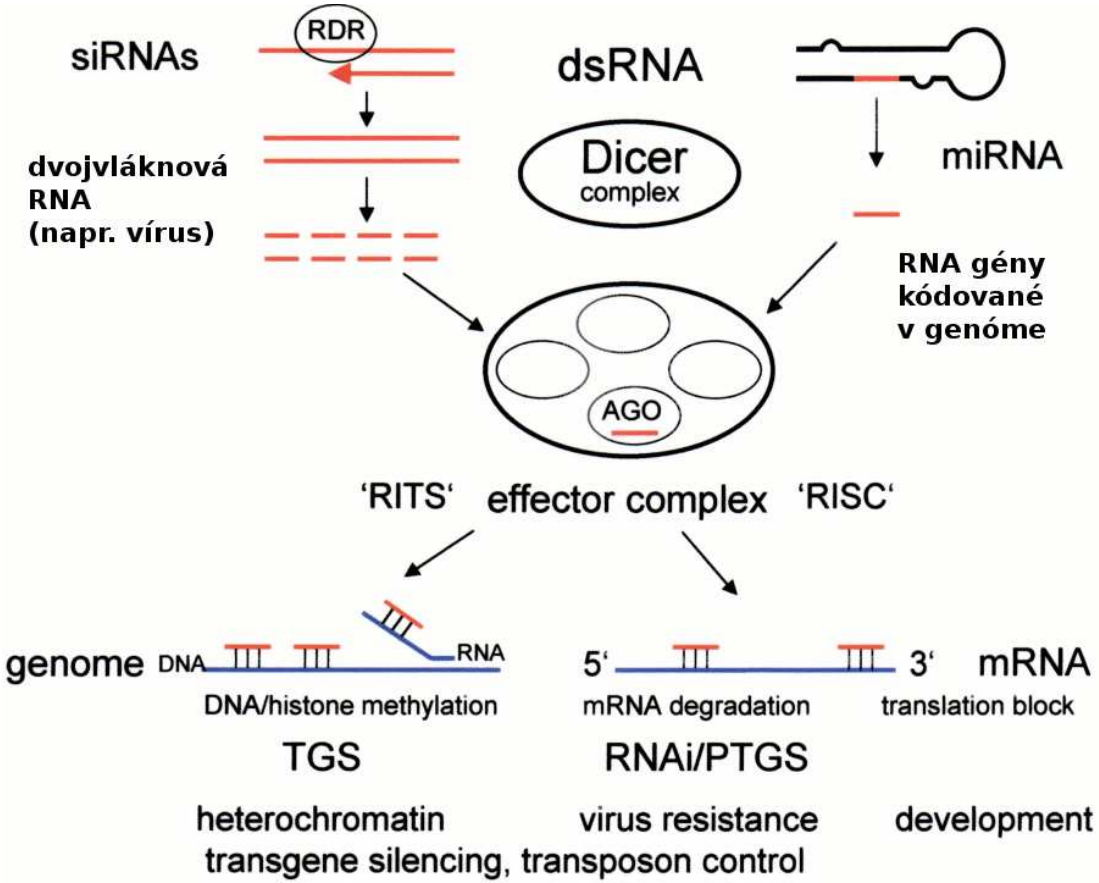
Nie je známy efektívny algoritmus, heuristiky často nájdu sekvenciu pomerne rýchlo.



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

RNA interference, RNAi

Krátke RNA (cca 22nt), viažu sa na 3' UTR a znižujú expresiu génu.



[Matzke and Matzke, 2004]

Problém: miRNA gene finding

Proces tvorby funkčnej miRNA:

primárny transkript → 70nt helix-slučka → 21-23nt jednovláknová RNA

lin-4

Cieľ: nájdi všetky miRNA gény v genóme

- Existencia transkriptu: EST databázy, prípadne over experimentom
- RNA štruktúra s dostatočne dlhým helixom
- Zachovaná sekvencia v genomických zarovnaniach; slučka menej zachovaná ako helix
- Motívy nájdené v okolí helixu v časti génov

Over experimentom: akumulácia transkriptov v neprítomnosti Diceru.

Problém: miRNA target prediction

Nájdi miesta na 3' koncoch génov, kde sa viažu známe miRNA.

```
HMGA2   5' CCGACAUUCAAUUUCUACCUCA 3'      NF2     5' UACAAGAGAUUCUCCUGCCUCA 3'
                   : | :| ||||||||                       || | | | :|| ||:|||||
let-7a  3' UUGAUAUGUUGGAUGAUGGAGU 5'      let-7a  3' UUGAUAUGUUGGAUGAUGGAGU 5'
```

- (Čiastočná) zhoda s miRNA génom (hľadanie podobností v sekvenciách)
- silná väzba s miRNA génom (výpočet energie väzby)
- zachovaná sekvencia vo viacerých genómoch (evolučné modely, zarovnania)

Problém: siRNA design

RNAi sa využíva laboratórne na umelé zníženie expresie génu.

Vstup: 3'UTR vybraného génu + databáza génov v organizme
Cieľ: Nájdi siRNA, ktorá má vhodné štruktúrne vlastnosti
(nie každá sekvencia správne spolupracuje s RISC komplexom),
vyskytuje sa vo vybranom géne, nevyskytuje sa v iných génoch.

- Výpočty štruktúry a energií (napr. 5' koniec by mal mať slabšiu energiu väzby ako 3' koniec)
- Klasifikátory kombinujúce rôzne ukazovatele do jedného skóre trénované na experimentálnych dátach
- Sensitívne a rýchle vyhľadávanie krátkych reťazcov v databáze génov, dovoľuje 1-2 rozdielne bázy

Zhrnutie

- Určovanie sekundárnej štruktúry RNA:
  minimalizácia energie podľa nameraných parametrov,
  alebo pravdepodobnostné modely (stochastické bezkontextové gramatiky).
- Spoľahlivejšie výsledky, keď použijeme zarovnanie viacerých sekvencií, ale niekedy je ťažké správne zarovnať.
- Známe rodiny možno reprezentovať pomocou kovariančných modelov a hľadať ďalšie výskyty.
- Väčšina problémov sa dá riešiť dynamickým programovaním, ktoré je pomerne pomalé a ignoruje pseudouzly.
- Ďalšie problémy: design RNA štruktúr, miRNA gény