Príprava pilotného predmetu "Metódy v bioinformatike"

Príloha k ročnej správe k projektu KEGA 058-016UK-4/2010 za rok 2010

Obsah

1 Stav a plán prípravy pilotného predmetu

Pilotný predmet "Metódy v bioinformatika" bude vyučovať Mgr. Bronislava Brejová, PhD a Mgr. Tomáš Vinař, PhD v zimnom semestri školského roku 2010/2011. Predmet bude prezentovaný študentom informatických a biologických zameraní, prevažne magisterského štúdia.

Predmet bude vyučovaný formou dvojhodinových prednášok, ktoré budú spoločné pre biológov aj informatikov a dvojhodinových cvičení, kde budú biológovia a informatici rozdelení na dve rôzne skupiny, pričom biologická skupina bude mať cvičenia v počítačovej učebni.

Na prednáškach budú prezentované základné problémy bioinformatiky, ich formulácia a prístupy k ich riešeniu. Cvičenia pre informatikov budú obsahovať doplňujúci materiál z biológie, ako aj materiál rozvíjajúci metódy a algoritmy. Cvičenia pre biológov budú obsahovať doplňujúci materiál z informatiky, ako aj praktické cvičenia pri počítači zamerané na praktickú aplikáciu získaných vedomostí.

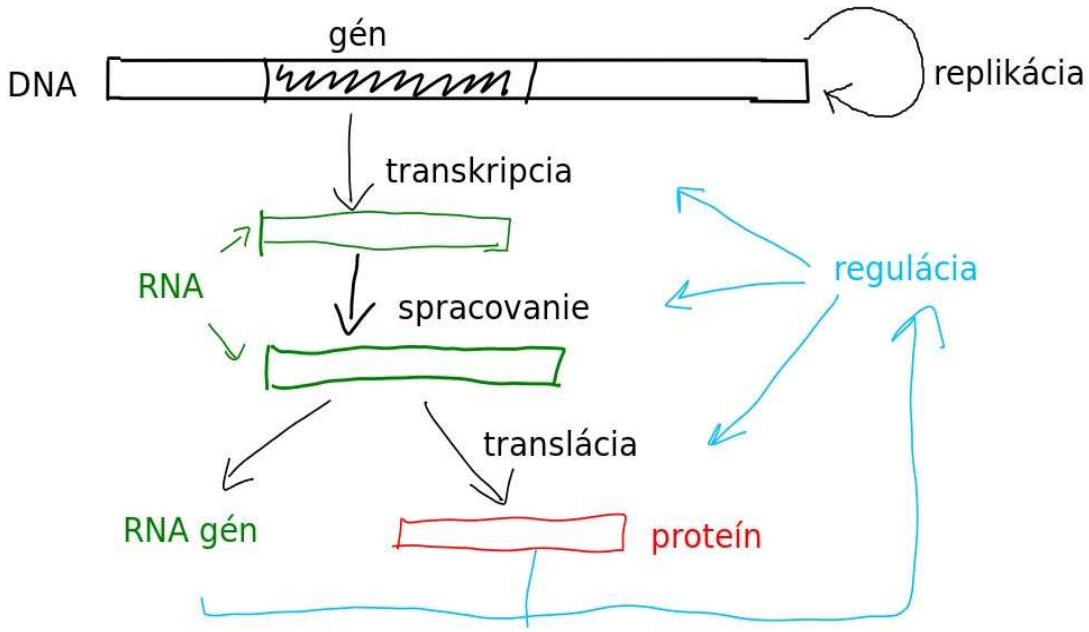
V čase písania tejto správy prebieha intenzívna príprava materiálov pre tento predmet. Predbežná verzia príprav prednášok je zahrnutá v tomto materiály, tento materiál v ďalšom období rozvinieme do podoby skrípt.

2 Úvodné informácie o predmete

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

Osnova predmetu



- *Sekvenovanie, zostavovanie sekvencií*
  Ako sa získavajú DNA sekvencie, akú rolu v tom hraje informatika
  Formulácia ako grafový problém
- *Zarovnávanie (sequence alignment)*
  Ako nájsť podobné sekvencie, typy zarovnaní, význam parametrov
  Dynamické programovanie, efektívna práca s reťazcami
- *Hľadanie génov*
  Ako výpočtovo hľadať gény v genóme
  Pravdepodobnostné modely, skryté Markovove modely (HMM),
  dynamické programovanie
- *Evolúcia, rekonštrukcia fylogenetických stromov*
  Prístupy založené na vzdialenostiach a úspornosti (parsimony)
  Pravdepodobnostné modely evolúcie

- *Komparatívna genomika*
  Ako využiť informáciu z viacerých príbuzných genómov
  Spojenie HMM a evolučných modelov
- *Štruktúra a funkcia proteínov*
  Ako informaticky zistiť čo najviac o novom proteíne
  HMM, energeticke modely proteínovej štruktúry
- *Expresia a regulácia génov*
  Ako analyzovať výsledky microarrays
  Zhlukovanie (clustering), biologické siete a ich vlastnosti
- *Transkripčné faktory*
  Predikcia väzobných miest
  Hľadanie opakujúcich sa motívov v sekvenciách
  Rozpoznávanie známych motívov pomocou strojového učenia
- *RNA*
  Ako predikovať sekundárnu štruktúru RNA, hľadať RNA gény
  Dynamické programovanie, stochastické bezkontextové gramatiky
- *Populačná genetika*
  Štúdium rozdielov medzi jedincami v rámci druhu
  Pravdepodobnostné modely, stochastické algoritmy

3 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
TTTTTTTTTCCTTTTGGGAGCTGAAAAATTTCCGGTAAGGGAAAGAAGGGCTCCTTTCGCTCCTTATTT
CCCCGCCTCCTTCCCTCCCCCACCTTCCCCTCCTCCGGCTTTTTCCTCCCAACTCGGGGAGGTCCTTCCC
GGTGGCCGCCCTGACGAGGTCTGAGCACCTAGGCGGAGGCGGCAGGCTTTTTGTAGTGAGGTTTGCGC
CTGCGCAGCGCGCCTGCCTCCGCCATGCACGGGGGTGGCCCCCCCTCGGGGGACAGCGCATGCCCGCTGC
GCACCATCAAGAGAGTCCAGTTCGGAGTCCTGAGTCCGGATGAACTGAAGCGAATGTCTGTGACGGAGGG
TGGCATCAAATACCCAGAGACGACTGAGGGAGGCCGCCCCAAGCTTGGGGGGCTGATGGACCCGAGGCAG
GGGGTGATTGAGCGGACTGGCCGCTGCCAAACATGTGCAGGAAACATGACAGAGTGTCCTGGCCACTTTG
GCCACATTGAACTGGCCAAGCCTGTGTTTCACGTGGGCTTCCTGGTGAAGACAATGAAAGTTTTGCGCTG
TGTCTGCTTCTTCTGCTCCAAACTGCTTGTGGACTCTAACAACCCAAAGATCAAGGATATCCTGGCTAAG
TCCAAGGGACAGCCCAAGAAGCGGCTCACACATGTCTACGACCTTTGCAAGGGCAAAAACATATGCGAGG
GTGGGGAGGAGATGGACAACAAGTTCGGTGTGGAACAACCTGAGGGTGACGAGGATCTGACCAAAGAAAA
GGGCCATGGTGGCTGTGGGCGGTACCAGCCCAGGATCCGGCGTTCTGGCCTAGAGTTGTATGCGGAATGG
AAGCACGTTAATGAGGACTCTCAGGAGAAGAAGATCCTGCTGAGTCCAGAGCGAGTGCATGAGATCTTCA
AACGCATCTCAGATGAGGAGTGTTTTGTGCTGGGCATGGAGCCCCGCTATGCACGGCCAGAGTGGATGAT
TGTCACAGTGCTGCCTGTGCCCCCGCTCTCCGTGCGGCCTGCTGTTGTGATGCAGGGCTCTGCCCGTAAC
CAGGATGACCTGACTCACAAACTGGCTGACATCGTGAAGATCAACAATCAGCTGCGGCGCAATGAGCAGA
```

Ako zadefinovať potenciálne homológy?

- Mali by mať veľa spoločných (zachovaných) báz
  niektoré bázy môžu byť zmenené, pridané, ubraté
  AT*GCACG*T*TAAT***
  AGCACG*C*TA*CC*AT
- Zobrazenie vo forme zarovnania
  *AT**GCACG**TTA*--*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: $|\mathrm{lcs}(X, Y_i)| \geq 0.7|X| \wedge |\mathrm{lcs}(X, Y_i) \geq 0.7|Y_i|$
- Výstup: Všetky nájdené reťazce Y_i, dĺžky lcs, zarovnania

Poznámky

- Informatikovi nepovieme, *ako hľadať homológy*
  povieme iba, *čo presne chceme nájsť*
- Formulácia problému zvyčajne interakciou medzi biológmi a informatikmi/matematikmi
- O dva týždňe 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

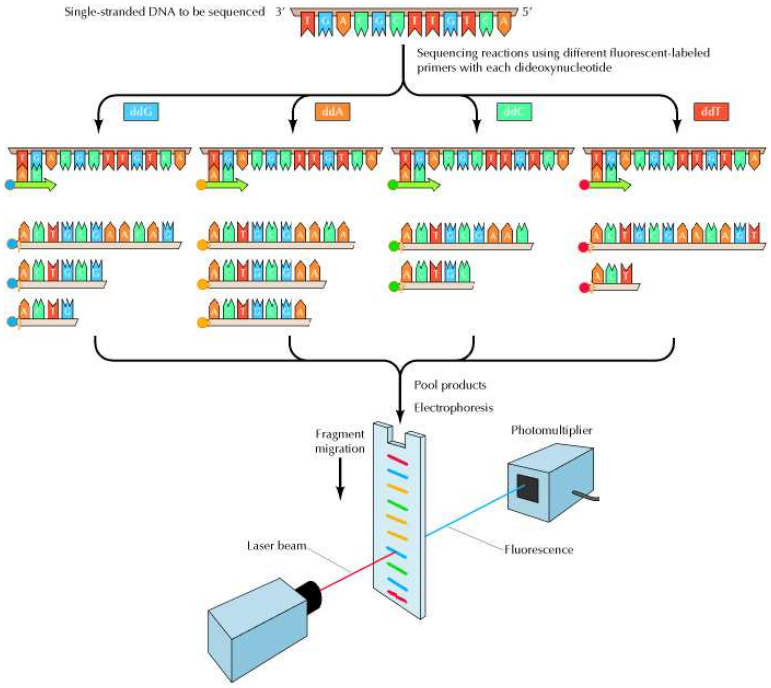
- Biologické pozadie problému
- Formulácia ako informatický problém
- Idea algoritmu (riešenia problému)

Typické cvičenia

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

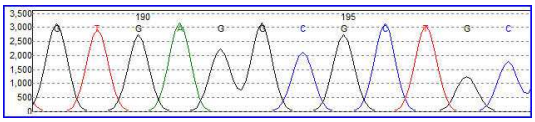
4 Sekvenovanie a zostavovanie genómov (seqencing, genome assembly)

Sangerovo sekvenovanie (Sanger sequencing), 1977



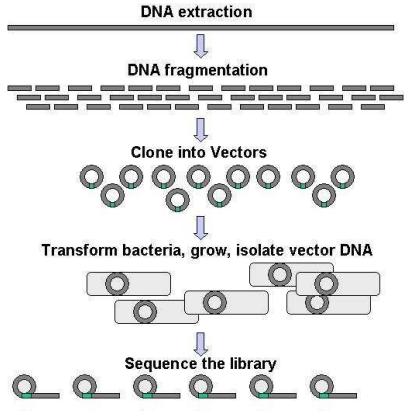
[Cooper and Hausman, 2004, Fig.3.21–22]

- 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 odsekvenovať dlhú DNA sekvenciu?

Sekvenovanie dlhých DNA sekvencií



Dostávame veľa segmentov (sequencing reads)
s neznámym umiestnením v genóme
Ako to dať celé dokopy?

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!

Sekvenovanie genómov

[Roberts2001]

1976	MS2 (RNA vírus) 40 kB
1988	projekt sekvenovania ľudského genómu (15 rokov)
1995	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)
čoskoro	1000 ľudských genómov, 22 cicavcov

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 najviac $3,5\times$ horšie ako optimálne [Kaplan2005]
  T.j. je to 3,5-aproximačný algoritmus
  (možno aj 2-aproximačný, otvorený problém)
- Existuje aj 2,5-aproximačný algoritmus [Sweedyk2000]

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 (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)

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

Overlap-Layout-Consensus

Napr. ARACHNE [Batzoglou2002]

- *Overlap:* (Prekryv)
  - Nájdeme prekrývajúce sa segmenty
  - Zostavíme ich do väčších *kontigov* (contigs)
- *Layout:* (Rozmiestnenie)
  - Určíme relatívnu polohu jednotlivých kontigov a ich vzdialenosti (pomocou pair-end reads)
  - Dostaneme *superkontigy* (supercontigs) 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)

ARACHNE: Hľadanie prekryvov

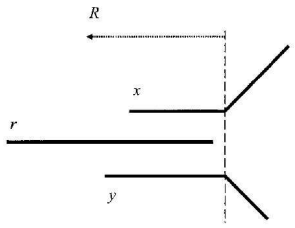
- *Triviálny algoritmus:* porovnaj každé dva segmenty
  Časová zložitosť: $O(n^2)$, kde $n \approx 2,000,000$
- *Efektívnejší algoritmus:* dostatočne prekrývajúce sa segmenty veľmi pravdepodobne zdieľajú 100% zachovaný úsek dĺžky k ($k \approx 24$)
  - Vybudujeme "zoznam" všetkých k-tic zo všetkých segmentov, spolu s číslami príslušných segmentov
  - Zotriedime zoznam podľa k-tic $O(n \log n)$
  - Segmenty s veľkým prekryvom budú vo výsledku pohromade
- *Príklad:* 1:TAATAT 2:GTCTGA 3:TATAAA
  Zotriedené trojice: (AAA,3) (AAT,1) $\boxed{\text{(ATA,1) (ATA,3)}}$ (CTG,2) (GTC,2) (TAA,1) (TAA,3) $\boxed{\text{(TAT,1) (TAT,3)}}$ (TCT,2) (TGA,2)

ARACHNE: Hranice opakujúcich sa sekvencií

- *Príklad:* 10xTTAATA, 10xATATTA, 3xTTAGCT

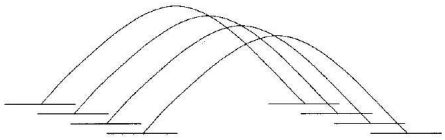
  TTAATATTAGCT?
  TTAATATTAATATTAATATTAATATTAGCT?
  TTAATATTA + ATATTAGCT?

- Rozširovanie skončí akonáhle nájdeme potenciálnu hranicu opakujúcej sa sekvencie:

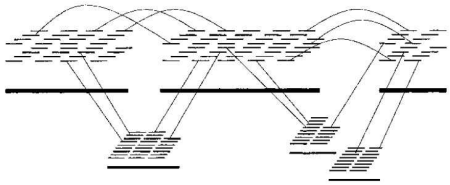


ARACHNE: Využitie spárovaných segmentov

- *Rozširovanie kontigov:*



- *Vytváranie superkontigov:*



ARACHNE: Výsledky a vplyv pokrytia

Zostavenie genómu D. melanogaster zo simulovaných segmentov:

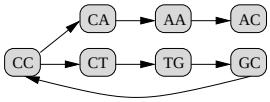
	Počet	Priem. dĺžka	Chýb
10x pokrytie:			
Kontigy	678	174 kB	
Superkontigy	65	1.8 MB	115
5x pokrytie:			
Kontigy	8774	13 kB	
Superkontigy	450	254 kB	175

EULER: Iná formulácia problému [Pevzner2001]

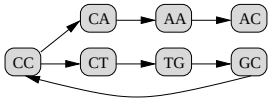
- Predpokladajme jednu orientáciu, žiadne chyby,
  jeden chromozóm úplne pokrytý segmentami
- Nasekajme segmenty na (prekrývajúce sa) kúsky dĺžky k
- Zostavme z nich *de Bruijnov graf*

- *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
- Orientovaný graf má Eulerovský ťah z u do v *práve vtedy,*
  keď po pridaní hrany (v, u) je graf silne súvislý
  a počet vchádzajúcich a vychádzajúcich hrán je rovnaký v každom vrchole.
- *Jednoducho riešiteľný problém v čase* $O(n + m)$
  Vieme overiť túto podmienku, aj nájsť ťah

Čo ak de Bruijnov graf nemá Eulerovský ťah?

- Znásobíme niektoré hrany
- Tieto zodpovedajú opakovaniam

Čo ak de Bruijnov graf má viacero Eulerovských ťahov?

- Zoberieme taký ťah, ktorý obsahuje pôvodné segmenty ako podcesty
- Zase ťažký problém, ale v praxi pomáhajú jednoduché pravidlá
- Opatrné riešenie: Ak z vrcholu 2 cesty, rozdeľ na kontigy

A čo spárované segmenty?

- Nájdi vrcholy v grafe, ktoré im zodpovedajú
- Ak je v grafe jediná cesta medzi týmito vrcholmi vhodnej dĺžky,
  premeň spárované segmenty na jeden veľký segment

Ďalšie problémy, ktoré treba vyriešiť:

- Sekvenovacie chyby
- Dvojité vlákna

Sekvenovacie technológie novej generácie

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

Nevýhody:

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

Sekvenovacie technológie novej generácie

Mardis, 2008, (*) Raes 2009

	454 (Roche)	Illumina (Solexa)	SOLiD (AB)
Time/run	7h	4 dni	5 dní
Data/run	100 Mb	1300Mb	3000Mb
Read length	250bp	32-40bp	35bp
Cost per run	$8 000	$9 000	$17 000
Cost per Mb	$80	$6	$6
(*) Data/run	500 Mb	18000Mb	30000Mb
(*) Read length	400bp	75bp	100bp

Informatické problémy

- Ťažšie zostavovanie z kratších kúskov
- Mapovanie kúskov na hotový genóm
- Veľmi veľa dát, treba veľmi rýchle programy
- Množstvo dát rastie rýchlejšie ako Moorov zákon

populačná genetika

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

Environmentálne sekvenovanie – metagenomics

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

Sekvenovanie nových organizmov

- Cieľ: Odsekvenovanie veľkej časti žijúcich organizmov
- Mnohé organizmy s nízkym pokrytím, veľa krátkych kontigov
- Rekonštrukcia genómov predkov
- Štúdium evolúcie rôznych funkcií
- Nájdenie všetkých funkčných elementov genómov
- Evolúcia komplexných funkcií s účasťou viacerých génov

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

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 superkontigy

5 Sequence alignment (zarovnávanie sekvencií) 1/2

Podrobnejšie informácie [Durbin et al., 1998, kap.2,6]

Problém: Lokálne zarovnávanie (local alignment)

Vstup: dve sekvencie, dotaz (query) a databáza (database, target).Databáza a niekedy aj dotaz môžu byť veľmi dlhé.

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

Výstup: podobné úseky (zarovnania, alignments).

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

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

Na čo sú dobré zarovnania?

- *Orientácia v obrovských databázach.*Genbank 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. Pozri kap. **??**.

Formulácia problému

Skórovanie zarovnania: pre ilustráciu jednoduchý príklad: zhoda +1, nezhoda -1, medzera -1.

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

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

Vo všeobecnosti sa snažíme nastaviť skóre tak, aby homológy mali vysoké skóre, náhodné zarovnania nízke skóre.

Problém 1: globálne zarovnanie (global alignment) Vstup: sekvencie $P = p_1p_2 \dots p_n$ a $Q = q_1q_2 \dots q_m$.
Výstup: zarovnanie P a Q s najvyšším skóre.

Problém 2: lokálne zarovnanie (local alignment) Vstup: sekvencie $P = p_1p_2 \dots p_n$ a $Q = q_1q_2 \dots q_m$.
Výstup: zarovnania podreťazcov $p_i \dots p_j$ a $q_k \dots q_\ell$ s najvyšším skóre.

Dynamické programovanie pre globálne zarovnanie (Needleman, Wunsch 1970)

Tento algoritmus je známy ako Needleman-Wunschov algoritumus [Needleman and Wunsch, 1970], ale ich pôvodná verzia bola menej efektívna. Podobný algoritmus ako pre najdlhší spoločný podreťazec, kap. **??**.

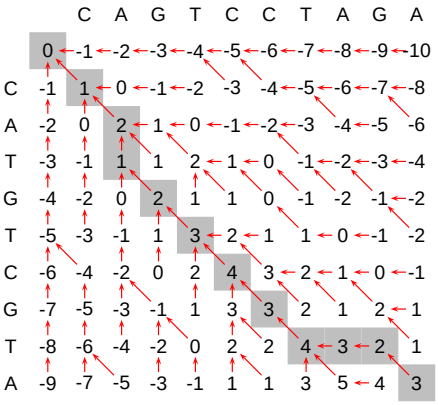
Podproblém: $A[i,j]$: najvyššie skóre globálneho zarovnania reťazcov $p_1p_2\ldots p_i$ a $q_1q_2\ldots q_j$.

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

Rekurencia: $A[i,j] = \max\{A[i-1,j-1] + c(p_i,q_j), A[i-1,j] - 1, A[i,j-1] - 1\}$kde $c(x,y) = 1$ ak $x = y$ a $c(x,y) = -1$ ak $x \neq y$

Príklad globálneho zarovnania




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

Dynamické programovanie pre lokálne zarovnanie (Smith, Waterman 1981)

Tento algoritmus je známy ako Smith-Watermanov algoritmus [Smith and Waterman, 1981].

Podproblém: $A[i,j]$: najvyššie skóre lokálneho zarovnania reťazcov $p_1p_2\ldots p_i$ a $q_1q_2\ldots q_j$, ktoré obsahuje bázy p_i a q_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 $p_i = q_j$ a sú zarovnané $A[i,j] = A[i-1,j-1] + 1$
ak $p_i \neq q_j$ a sú zarovnané $A[i,j] = A[i-1,j-1] - 1$
ak p_i je zarovnané s medzerou $A[i,j] = A[i-1,j] - 1$
ak q_j je zarovnané s medzerou $A[i,j] = A[i,j-1] - 1$
ak p_i a q_j nie sú časťou zarovnania s kladným skóre $A[i,j] = 0$

Rekurencia: $A[i,j] = \max\{0, A[i-1,j-1] + c(p_i, q_j), A[i-1,j] - 1, A[i,j-1] - 1\}$kde $c(x,y) = 1$ ak $x = y$ a $c(x,y) = -1$ ak $x \neq y$

Príklad lokálneho zarovnania

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

```
   CA-GTCCTA
CATGTCATA
```

Zložitejšie skórovanie

Problémy $+1, -1$ skórovania:

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

Úloha skórovacej schémy:

- Chceme vedieť rozlíšiť *lepšie 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 P a Q sú *správne zarovnané homológy*
$p =$ pravdepodobnosť, že sa dve bázy *zhodujú*
$q =$ pravdepodobnosť, že sa *nezhodujú*
$r =$ pravdepodobnosť, že báza je *zarovnaná s medzerou*
$p + q + r = 1$
Pravdepodobnosť zarovnania:

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

$\Pr(P, Q \mid H) = p^{22} q^6 r^3$
$\log \Pr(P, Q \mid H) = 22 \log p + 6 \log q + 3 \log r$

Zložitejšie skórovanie: prvý pokus

Zhoda: $\log p$ Nezhoda: $\log q$ Medzera: $\log r$

Nevýhody takejto schémy:

- Vždy záporné skóre ⇒ čo s lokálnymi zarovnaniami?
- Neužitočné pre porovnávanie rôznych párov sekvencií

Zložitejšie skórovanie: dva pravdepodobnostné modely

(Pre jednoduchosť teraz neuvažujme medzery)

Model H: Sekvencie P a Q sú *správne zarovnané homológy*
$\Pr(P, Q \mid H) = \prod_{i=1}^{n} s(p_i, q_i)$
$s(p_i, q_i)$: pravdepodobnosť, že vidíme zarovnané práve bázy p_i a q_i
Model R: Sekvencie P a Q nijako spolu nesúvisia
$\Pr(P, Q \mid R) = \prod_{i=1}^{n} s(p_i)s(q_i)$
$s(p_i)$: pravdepodobnosť výskytu bázy p_i
Porovnanie modelov H a R: "log likelihood"

$$\log \frac{\Pr(P, Q \mid H)}{\Pr(P, Q \mid R)} = \sum_{i=1}^{n} \log \frac{s(p_i, q_i)}{s(p_i)s(q_i)}$$

Porovnanie modelov H a R: "log likelihood"

$$\log \frac{\Pr(P, Q \mid H)}{\Pr(P, Q \mid R)} = \sum_{i=1}^{n} \log \frac{s(p_i, q_i)}{s(p_i)s(q_i)}$$

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

BLOSUM62 skórovacia matica pre proteíny

*BLO*cks 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
...
```

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

- Vyber *biologicky relevantné zarovnania* proteínov (BLOCKS)
- Páry s nanajvýš 62% identitou
- $s(p, q)$: ako často vidíme aminokyseliny p a q zarovnané
- $s(p)$: ako často sa vyskytuje aminokyselina p

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

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

Skóre za začatie medzery (gap opening cost) o,skóre za rozšírenie medzery o jedna (gap extension cost) e. Medzera dĺžky g má skóre $o + e(g-1)$.Zvolíme $e < o$.

Základné nastavenia blastn: zhoda +2, nezhoda -3, $o = 5$, $e = 2$.Príklad vyššie: 22 zhôd, 6 nezhôd, 1 medzera dĺžky 3→ skóre $2 \cdot 22 - 3 \cdot 6 - 5 - 2 \cdot 2 = 16$.

Zhrnutie

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

Problémy na zamyslenie

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

6 Zarovnávanie sekvencií 2/2 (sequence alignment)

Podrobnejšie informácie [Durbin et al., 1998, kap.2,6]

Zhrnutie z minulej prednášky

- *Problém globálneho a lokálneho zarovnania*
  Vstup: sekvencie $P = p_1 p_2 \dots p_n$ a $Q = q_1 q_2 \dots q_m$.
  Výstup: zarovnanie P a Q s najvyšším skóre
  resp. zarovnania podreťazcov $p_i \dots p_j$ a $q_k \dots q_\ell$ 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 moderný počítač)

n	time
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
Pozri kap. 2.6 v [Durbin et al., 1998].

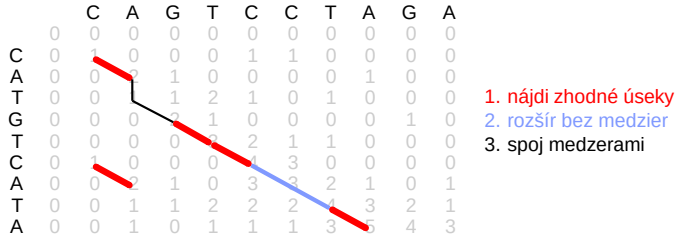
Heuristické lokálne zarovnávanie

- Nie je zaručené, že nájdeme najlepšie zarovnanie, ale program pobeží rýchlejšie.
- Prehľadaj 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).

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



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:

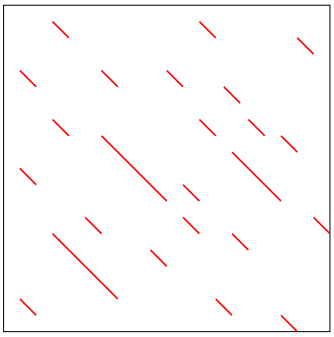
```
CA-GTCCTA
CATGTCATA
```

nenájdeme pre $w \geq 4$

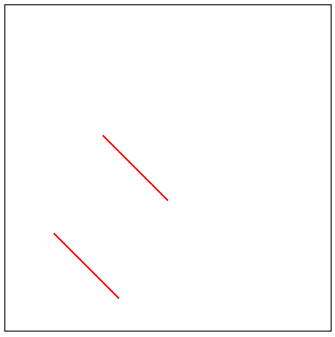
Senzitivita: aká časť *skutočných zarovnaní* obsahuje zhodu dĺžky w

Rýchlosť vs. senzitivita

Malé w
veľa náhodných zhôd, pomalé



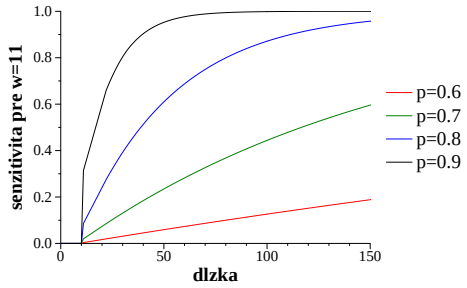
Veľké w
nenájdeme veľa zarovnaní



Senzitivita heuristického algoritmu

Odhad senzitivity: Predpokladáme zarovnanie bez medzier, dĺžky L
Každá pozícia je zhoda s pravdepodobnosťou p

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



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]

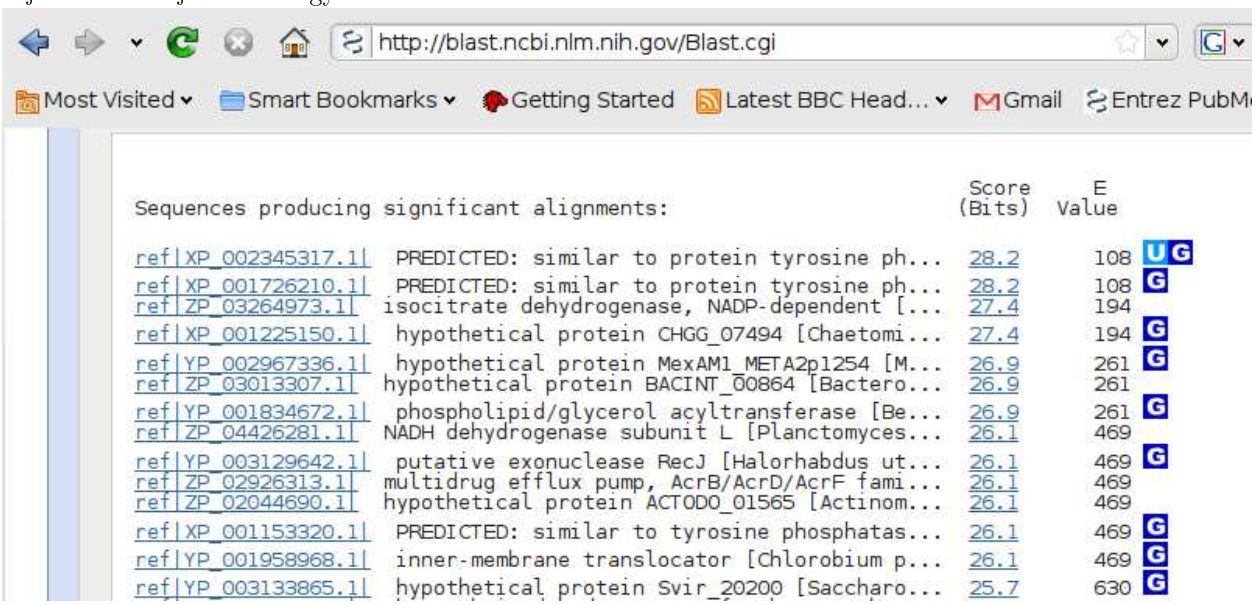
UCSC Blat: veľmi rýchle vyhľadávanie veľmi podobných sekvencií, napr. EST ku genómu [Kent, 2002].

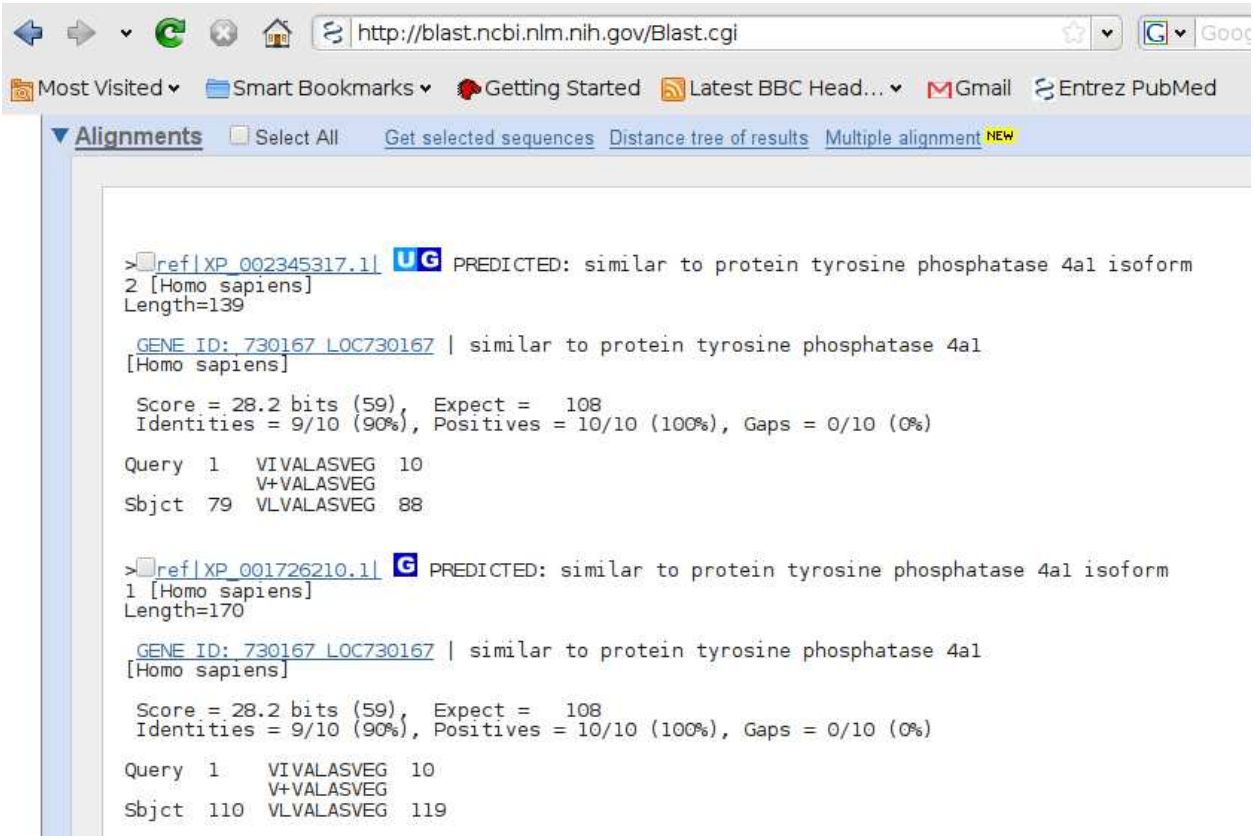
- používa veľké w
- vie rozdeliť EST na exóny

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.

⇒ nájde vzdialenejšie homológy.





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.

Pozn: $P = 1 - e^{-E}$
$\Rightarrow$ pri veľmi malých hodnotách sú E-value a P-value takmer identické.

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

Genomické zarovnania (whole-genome alignments)

ku každému úseku ľudského genómu nájsť zodpovedajúce časti z myši, potkana, kury, atď. (predpočítané v UCSC browseri) [Kent et al., 2003]

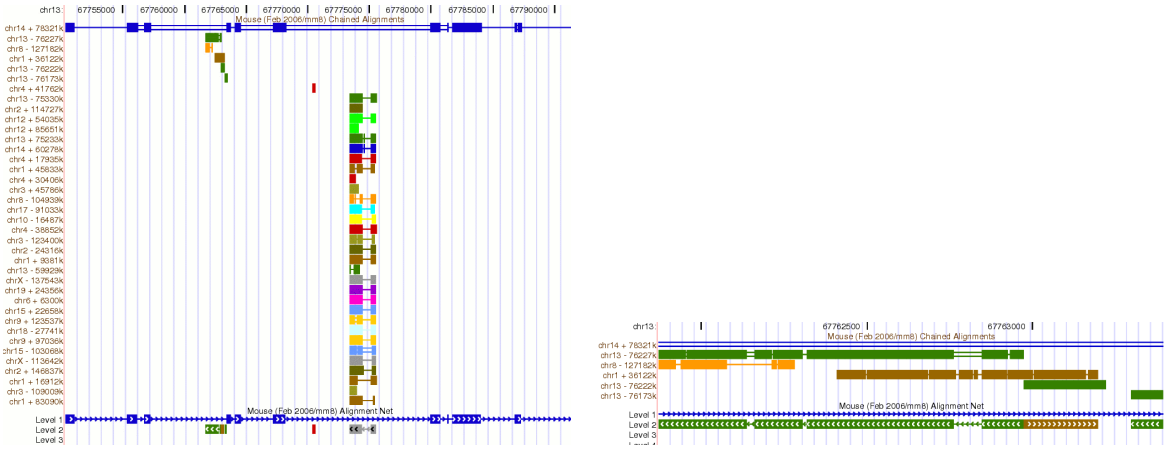
- Lokálne zarovnania nájdu exóny a iné zachované časti, sú však úseky, ktoré sa príliš zmenili.
- Pri duplikovaných úsekoch nevieme rozhodnúť, ktoré dvojice úsekov patria k sebe.
- *Synteny:* lokálne zarovnania, ktoré sa nachádzajú v dvoch genómoch v tom istom poradí a orientácii.Pomáha nám určiť, ktoré dvojice úsekov majú spoločného predka. *(orthology)*

Genomické zarovnania (whole-genome alignments)

Postup:

- Začni s lokálnymi zarovnaniami bez medzier.
- 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.



Viacnásobné zarovnanie, multiple sequence alignment

Zarovnaj viacero sekvencií.*Zložitosť:* $O(2^k n^k)$ pre k sekvencií dĺžky nPre 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

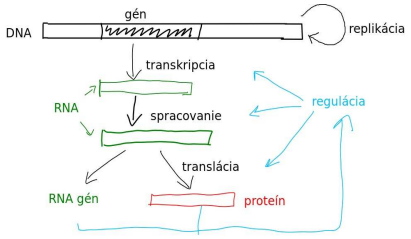
- Zarovnávanie, alignment, je základný nástroj bioinformatiky
- Formulácia problému: voľba skórovacej schémy
- Riešenie problému: presné ale pomalé algoritmy a rýchlejšie heuristiky, ktoré nie vždy nájdu všetko
- Špecializované programy na rôzne úlohy súvisiace so zarovnávaním

7 Hľadanie génov

Čo s osekvenovanými genómami?

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

- gény kódujúce proteíny (dnešná prednáška)
- RNA gény, (kap. 13)
- signály pre reguláciu transkripcie, zostrihu, atď
- pseudogény (nefunkčné kópie génov)
- repetitívne sekvencie, opakovania (sequence repeats)



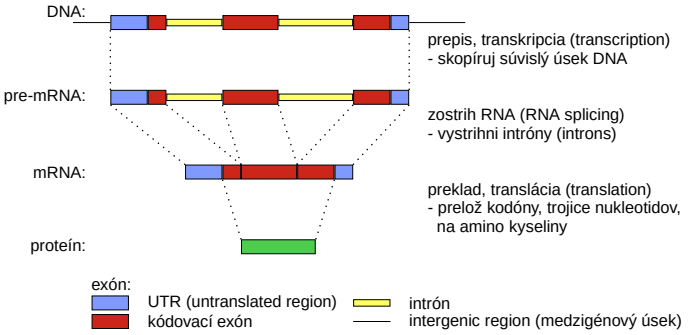
Ľudský genóm

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

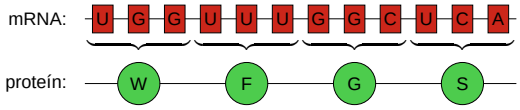
- repetitívne sekvencie
  – pokrývajú 49% genómu

Štruktúra eukaryotických génov

Proces tvorby proteínov:



Translácia: tri bázy mRNA (kodón) → aminokyselina proteínu



Bioinformatický problém: hľadanie génov

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

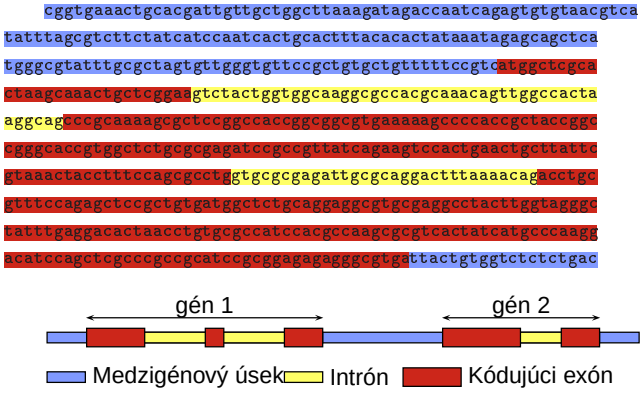
Zjednodušenia:

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

Bioinformatický problém: hľadanie génov

Vstup: DNA sekvencia

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



Bioinformatický problém: hľadanie génov

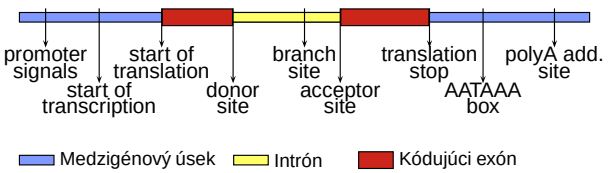
Vstup: DNA sekvencia

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

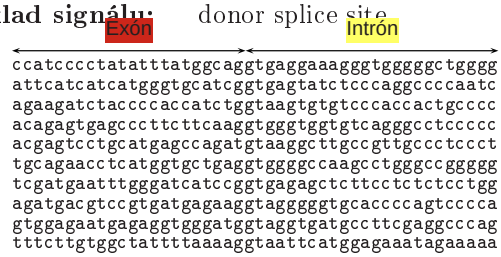
- Toto nie je dobre definovaný problém!
  Ako spoznáme, čo je gén?

Ako spoznáme gény?

Signály na hraniciach exónov:krátke reťazce, kde sa viažu komplexy zúčastnujúce sa na expresii génu



Príklad signálu: donor splice site



Zloženie sekvencie:

- iná frekvencia k-tic báz v kódujúcich a nekódujúcich úsekoch,
- kódujúce úseky sú 3-periodické,
- stop kodóny (TAA, TGA, TAG) len na konci posledného kódovacieho 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

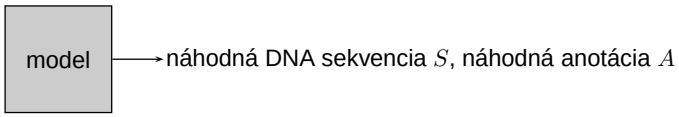
Bioinformatický problém: hľadanie génov

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.*

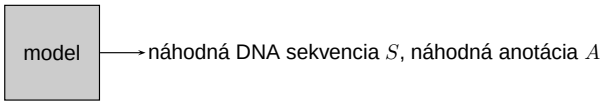
Žiadna informácia nám neumožňuje jednoznačne určiť, čo je gén.Skombinujeme dostupnú informáciu pravdepodobnostným modelom.



$\Pr(S, A)$ – 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



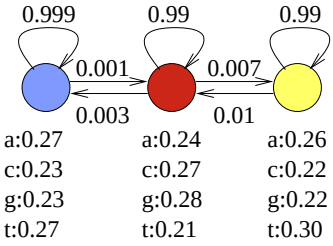
Použitie: pre sekvenciu S nájdi najpravdepodobnejšiu anotáciu A

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	0.010					
aa	0.009					
aa	0					
aa	0.007					
aa	0.001					
aa	0.010					

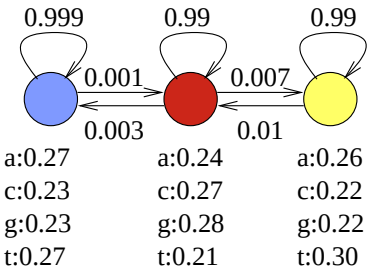
Skrytý Markovov model, hidden Markov model (HMM)

Spôsob, ako zadefinovať model pre dlhšie sekvencie.



- 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)

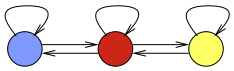


Predpokladajme, že model vždy začína v modrom stave.
Príklad:

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

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

Matematické označenie



Sekvencia $S_1, \ldots, S_n$
Anotácia $A_1, \ldots, A_n$

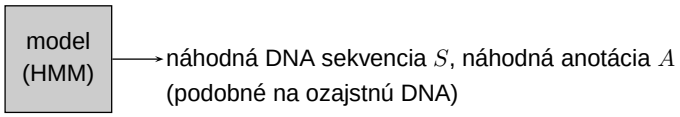
Parametre modelu:
Prechodová pravdepodobnosť $a(u, v) = \Pr(A_{i+1} = v | A_i = u)$,
Emisná pravdepodobnosť $e(u, x) = \Pr(S_i = x | A_i = u)$,
Počiatočná pravdepodobnosť $\pi(u) = \Pr(A_1 = u)$.

a			
	0.99	0.007	0.003
	0.01	0.99	0
	0.001	0	0.999

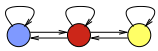
e	a	c	g	t
	0.24	0.27	0.28	0.21
	0.26	0.22	0.22	0.30
	0.27	0.23	0.23	0.27

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

Hľadanie génov s HMM

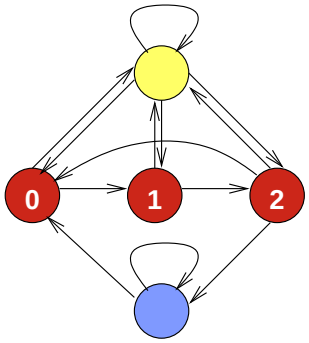


$\Pr(S, A)$ – pravdepodobnosť, že model vygeneruje pár (S, A).

- *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)

Rozšírenie na 3-periodické exóny

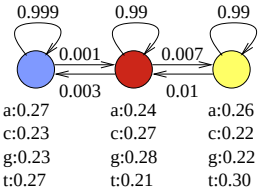
Kodón (trojica báz) $\rightarrow$ jedna aminokyselina
Namiesto jedného stavu pre exón použijeme tri stavy v cykle.

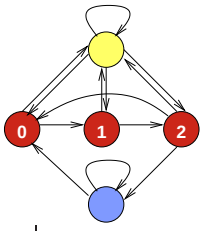


a	0	1	2		
0	0		0		0
1	0	0			0
2		0	0		
					0
		0	0	0	

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

Nové stavy majú odlišné emisné pravdepodobnosti



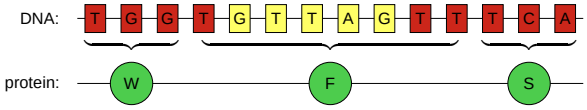


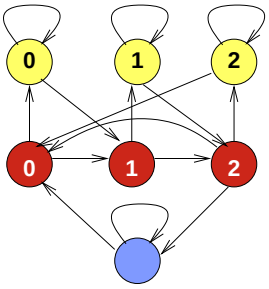
e	a	c	g	t
	0.24	0.27	0.28	0.21
	0.26	0.22	0.22	0.30
	0.27	0.23	0.23	0.27

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

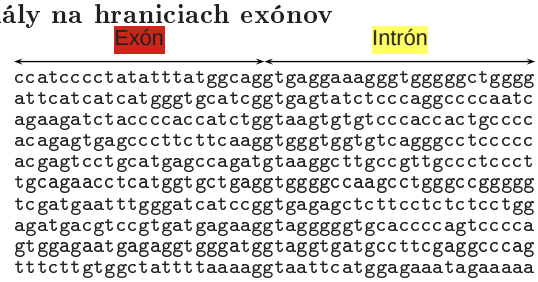
Konzistentné kodóny na intrónových hraniciach

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

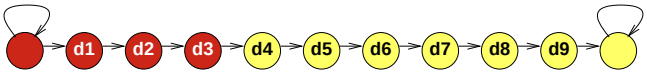




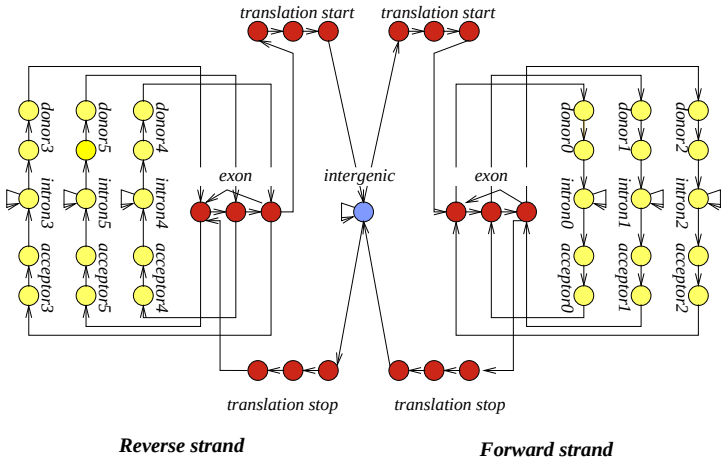
Signály na hraniciach exónov



Pridaj sériu stavov medzi exón a intrón:



Príklad HMM na hľadanie génov



Stavy vyšších rádov

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

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

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

...

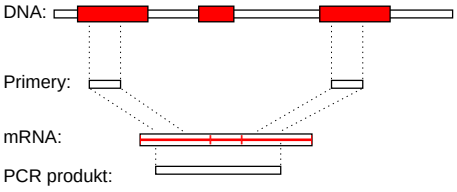
Na charakterizovanie exónov, intrónov atď používame rád 4-5.

Experimentálne overovanie génov

Overenie transkripcie a zostrihu

- EST sequencing: 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.



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.

Viac detailov o PCR a hmotnostnej spektrometrii v kap. **??**.

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)

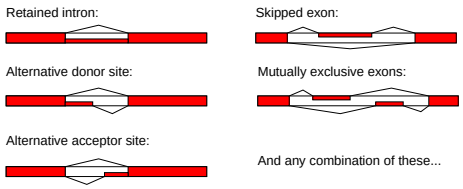
Prokaryotické genómy: GeneMark [Lukashin and Borodovsky, 1998], Glimmer [Delcher et al., 1999] a ďalšie.

Porovnávaním viacerých sekvencií: Viacej detailov v kap. **??**. 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.)Viacej detailov v kap. **??**. ExonHunter [Brejová et al., 2005], Augustus [Stanke et al., 2006], Jigsaw [Allen and Salzberg, 2005],
Fgenesh++ [Solovyev et al., 2006].

Obmedzenia gene finderov

- Alternatívny zostrih (alternative splicing): veľa génov môže vyprodukovať viacero mRNA molekúl. Gene findery väčšinou hľadajú iba jednu.

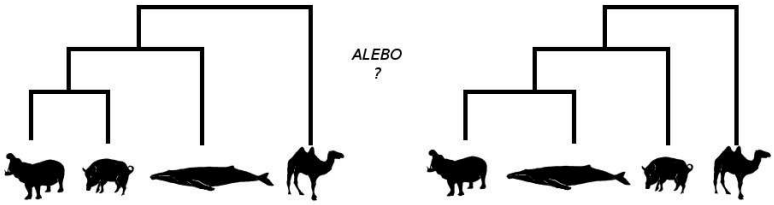


- Pretínajúce sa gény, resp. gény v intrónoch.
- Netypické gény (neobvyklé signály, veľmi krátke alebo dlhé exóny alebo intróny atď.)
- Hľadanie UTR a začiatku/konca transkripcie.

Zhrnutie

- Novo osekvenované genómy treba anotovať:
  určovať funkcie jednotlivým úsekom 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

8 Evolučné modely a stromy



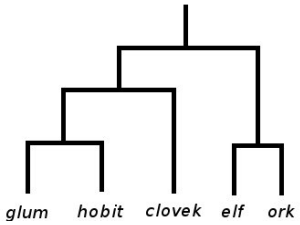
[Lusseau2003]

Rekonštrukcia fylogenetických stromov

Vstup: m zarovnaných (aligned) 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

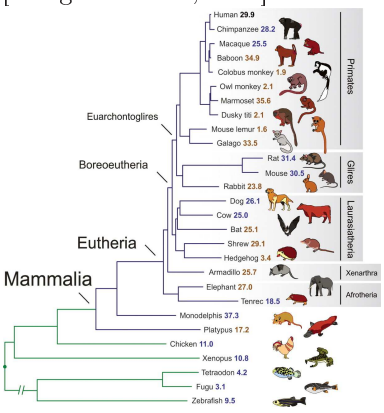
Výstup:



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

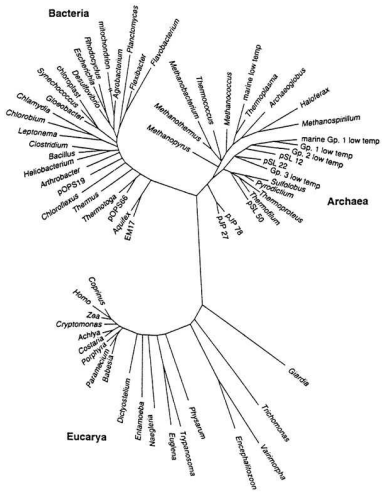
Zakorenené a nezakorenené stromy

[Margulies et al., 2007]



(zakorenený pomocou“outgroup”)

[Pace, 1997]



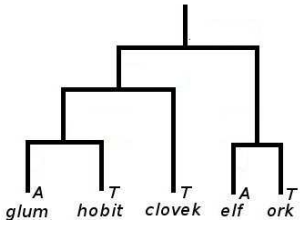
Parsimony (úsporné stromy)

Úloha: Dané sú sekvencie súčasných organizmov

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

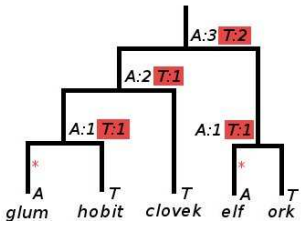
evolučná zmena = mutácia jednej bázy na inú bázu

Podotázka 1: Pre daný fylogenetický strom, doplniť *ancestrálne sekvencie* tak, aby bol potrebný najmenší počet zmien.

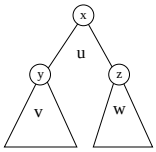


Podotázka 2: Nájsť *najúspornejší* evolučný strom, t.j. taký ktorý nám dá najmenší počet zmien

Výpočet ceny konkrétneho stromu



- dynamické programovanie
- pre každý vnútorný vrchol u a symbol x:$N_{u,x}$: koľko zmien je nutných v podstrome u, ak v koreni u bude symbol x?
- $N_{u,x} = \min_y\{[x \neq y] + N_{v,y}\} + \min_z\{[x \neq z] + N_{w,z}\}$

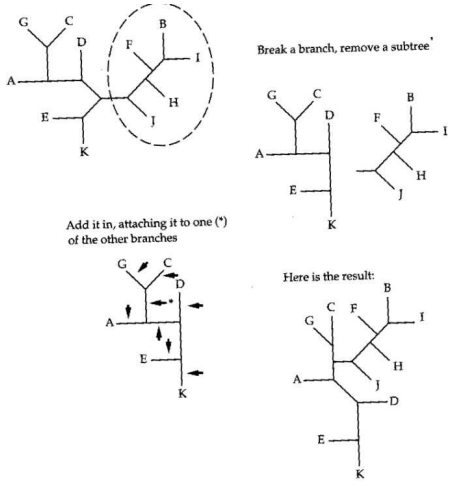


- *Časová zložitosť:* $O(n)$, lineárna

Hľadanie stromu s najväčšou parsimony

Triviálny algoritmus: vyskúšaj všetkých $\frac{(2n-3)!}{2^{n-2}(n-2)!}$ stromov
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	3	2
elf	4	0	3	6	2
Glum	3	3	0	3	5
hobit	3	6	3	0	4
ork	2	2	5	4	0

Idea spájania susedov

[Studier1988]

- 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:

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

Časová zložitosť: $O(n^3)$

Ako určiť dva listy na spájanie?

(Prečo nie dva najbližšie?)

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

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

Problém so vzdialenosťami

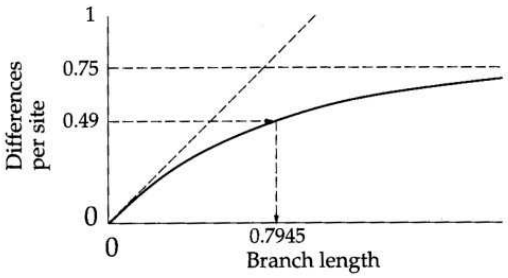
- Počas evolúcie sa môže stať, že tá istá báza zmutuje *viackrát* (trebárs aj späť na originálnu bázu)
- Pri počítaní vzdialeností ale vidíme iba nanajvýš jednu zmenu na každej pozícii $\Rightarrow$ odhad vzdialenosti menší ako v skutočnosti

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)

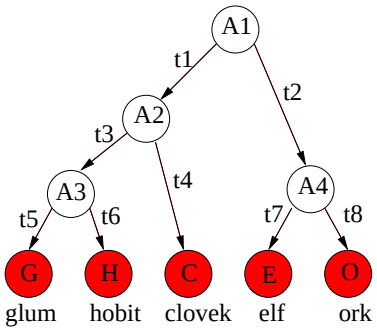
- Podľa takéhoto modelu môžeme korigovať pozorované vzdialenosti

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

- V praxi môžeme použiť zložitejšie modely, ktoré zahŕňajú rôzne frekvencie báz, transition/transversion ratio, variabilnú rýchlosť evolúcie na rôznych pozíciách (pozri [Felsenstein, 2004, kap.13])

Najvierohodnejšie stromy (Maximum likelihood)

- Strom môžeme chápať ako *jednoduchý generatívny model*



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

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

- Možno urobiť efektívne jednoduchým dynamickým programovaním (podobne ako v prípade parsimony); *Felsensteinov algoritmus*
- $\Rightarrow$ Pre daný strom a dĺžky hrán vieme spočítať vierohodnosť v čase $O(n)$

Ako nájsť najvieryhodnejší strom?

- Problém je NP-ťažký [Chor2005]; navyše komplikovaný tým, že na výpočet vierohodnosti *potrebujem 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.

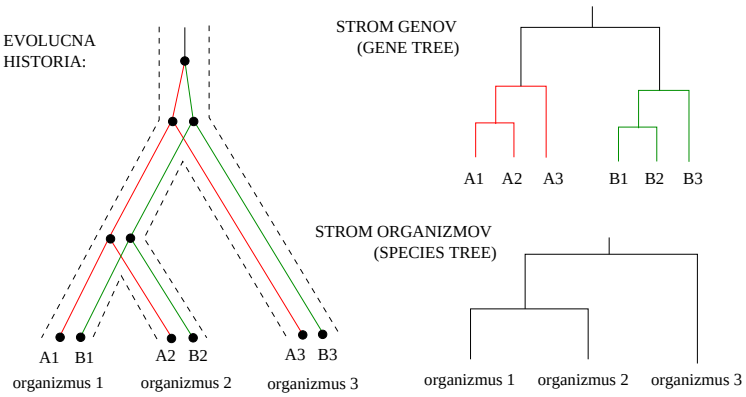
Porovnanie algoritmov

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

Odkiaľ zohnať dáta pre fylogenetiku?

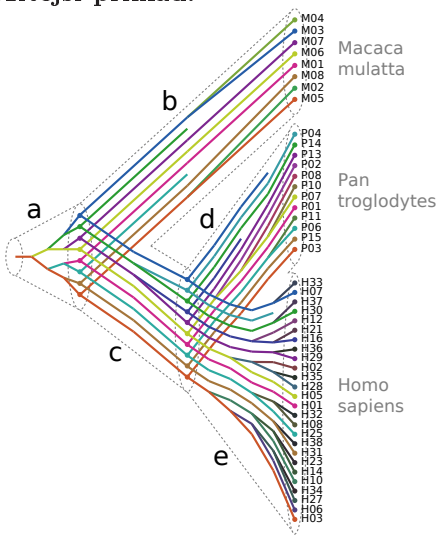
- *Mitochondriálna DNA (mtDNA):*
  - Krátky cirkulárny genóm bakteriálneho pôvodu uložený v mitochondriách (človek: cca 16KB)
  - Dedí sa zásadne po materskej línii (žiadne problémy s rekombináciou)
  - Rýchlejšie mutácie – vhodný nielen pre organizmy, ale aj jedince
  - Ľahko odsekvenovateľný a odsekvenovaný pre mnoho organizmov
- *Ribozomálna RNA (rRNA):*
  - Nachádza sa v ribozómoch
  - Nepostrádateľná funkcia pri syntéze proteínov
  - $\Rightarrow$ veľmi dobre zachovaná aj medzi organizmami
  - 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:* podobný gén na úrovni sekvencie
- *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 pravdepodobnosti, 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

9 Komparatívna genomika

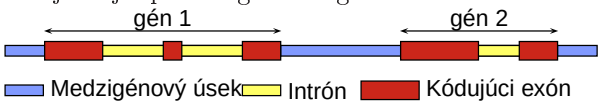
Komparatívna genomika

- Zostavíme viacnásobné zarovnania genómov
  zarovnané miesta by mali byť ortológy
- Štúdiom evolučných zmien sa snažíme nájsť gény a iné funkčné prvky
- Kombinujeme techniky na anotáciu (HMM) a modely evolúcie

```
        Human AGTGGCTGCCAGGCTG---GGATGCTGAGGCCTTGTTTGCAGGGAGGT
      Rhesus AGTGGCTGCCAGGCTG---GGTTGCTGAGGCCTTGTTTGCCGGGAGGT
        Mouse GGTGGCTGCCGGGCTG---GGTGGCTGAGGCCTTGTTGGTGGGGTGGT
          Dog AGTGGCTGCCCGGCTG---GGTGGCTGAGGCCTTATTTGCAGGGAGGT
        Horse GATGGCTGCCGGGCTG---GGCTGCCGAGGCCTTGTTCGTGGGGAGGT
    Armadillo AGTGGCTGCCGGGCTG---GGAGGCCAAGGCCTTGTTCGCGGGCAGGT
      Chicken AGTGGCTGCCAGTCTGCGCCGTGGCCGACGTCTTGCTCGGGGGAAGGT
X. tropicalis AATGGCTTCCATTTTGTGCCGCTGCTGAGGTCTTGTTCTGGGGAAGAT
```

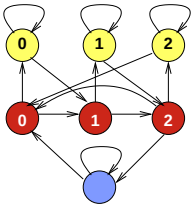
Opakovanie: hľadanie génov

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



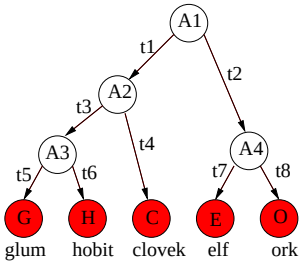
Vytvoríme skrytý Markovovský model (HMM), ktorý vie generovať sekvencie a ich anotácie podobné skutočným.

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: $\Pr(X = C \mid Y = A, t) = \frac{1}{4}(1 - e^{-\frac{4}{3}t})$
- 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 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: $\pi_A, \pi_C, \pi_G, \kappa = \alpha/\beta$

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)*

Štúdium purifikačného výberu

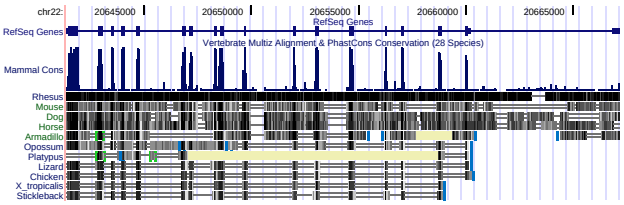
- Dôležité funkčné časti sekvencie musia zostať zachované
- Nefunkčné sekvencie sa vyvíjajú rýchlejším tempom ako funkčné
- Hľadáme *zachované sekvencie* medzi jednotlivými organizmami
- Veľká časť zodpovedá známym funkčným elementom (kódujúce gény, regulačné regióny, a pod.)
- Zachované sekvencie ktoré sa neprekrývajú s funkčnými elementami — nové objekty pre výskum

Štúdium purifikačného výberu

Hľadáme *zachované sekvencie* medzi jednotlivými organizmami, ktoré sú často funkčné.

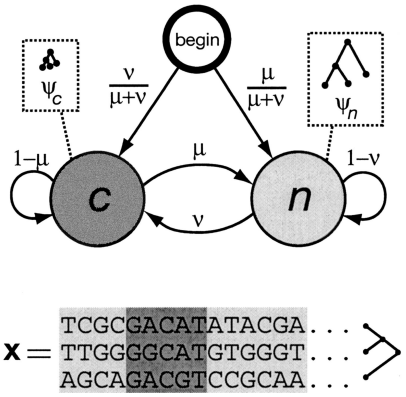
	Kódujúce	Intrón
Ľudské bázy pokryté zarovnaním	98%	48%
Zhoda v zarovnaniach	85%	69%

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



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

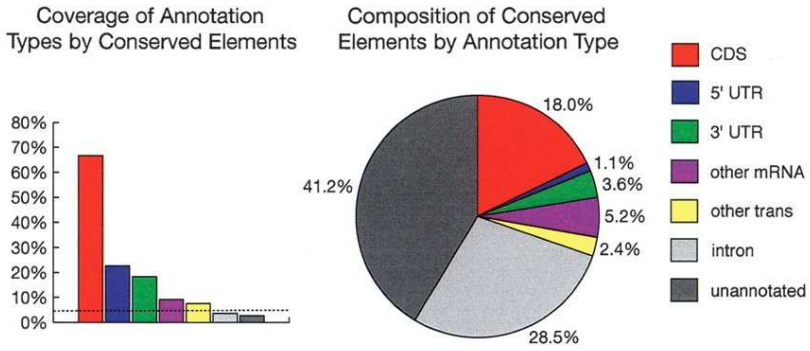
- Model určuje rozdelenie pravdepodobnosti cez zarovnania a anotácie (tu: anotácia = označenie zachovaných sekvencií)
- Pre dané zarovnanie hľadáme najpravdepodobnejšiu anotáciu
- Kombinácia Viterbiho a Felsensteinovho algoritmu

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, kuraťa, fugu



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*

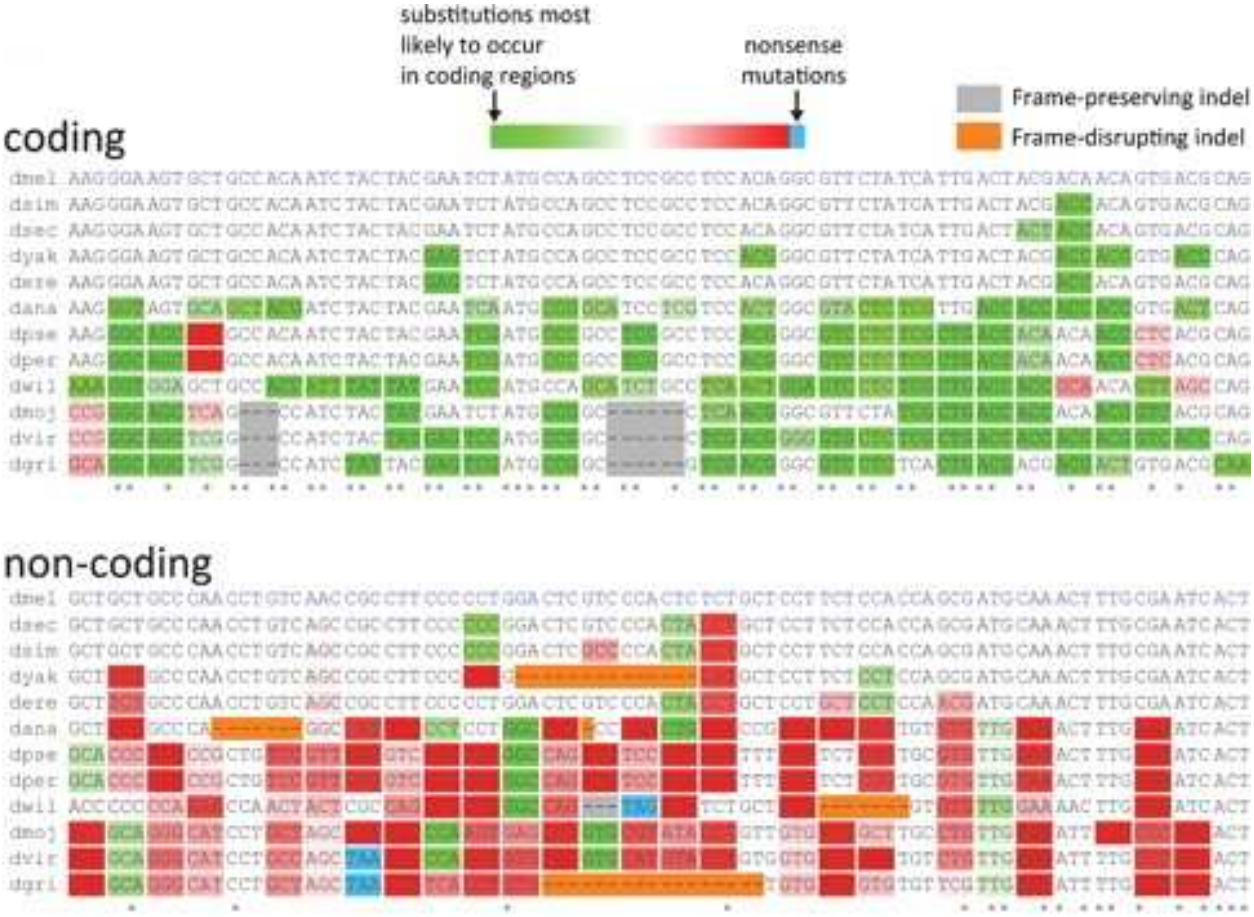
Valine (V)
GT*

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

Substitučný model pre kodóny

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

Rýchosť zmeny z kodónu i na kodón j:

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

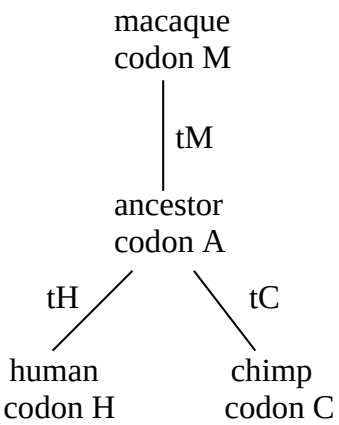
Príklad: $\mu_{AAC,GGC} = 0$, $\mu_{CTA,CTT} = \beta\pi_{CTT}$, $\mu_{CTA,CCA} = \omega\alpha\pi_{CCA}$

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]

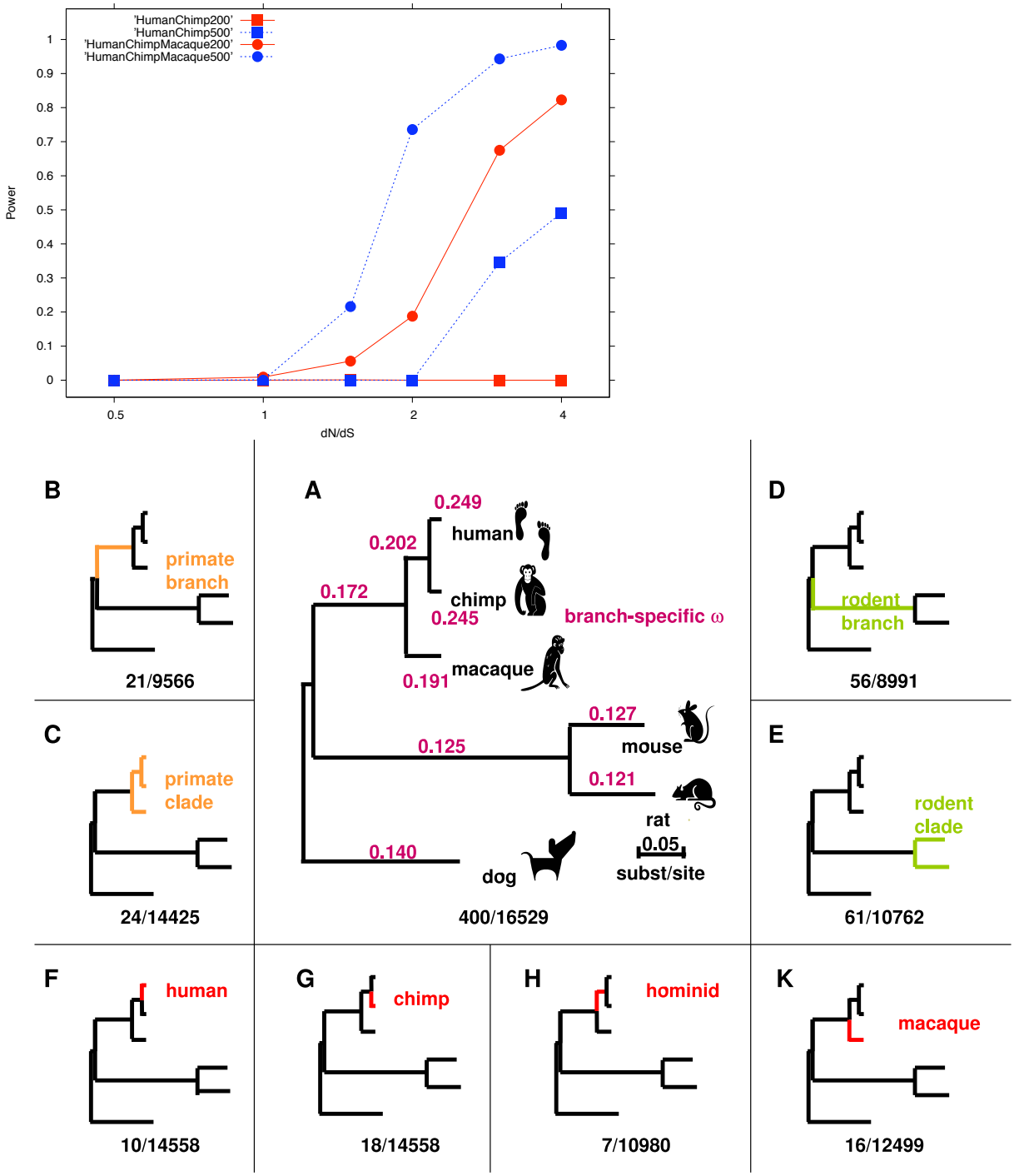
- Spočítame vierohodnosť dát L_A keď $\omega < 1$
- Spočítame vierohodnosť dát L_B bez obmedzenia ω
- Vždy platí $L_B \geq L_A$
- Ak skutočné $\omega < 1$, $L_A \approx L_B$

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



Funkčné kategórie obohatené o gény s pozitívnym výberom

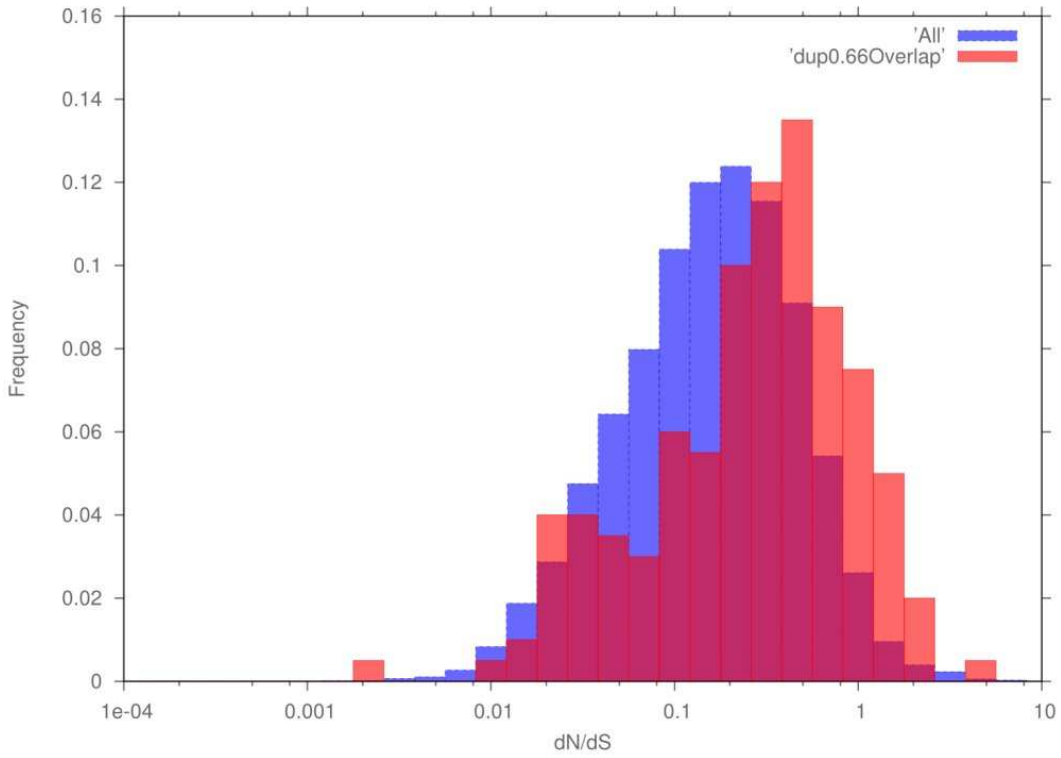
Defense: cellular defense response, antigen processing and presentation, response to virus, response to bacterium

Immunity: adaptive immune response, adaptive immune response somatic recomb, lymphocyte mediated immunity, immunog-

lobulin 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

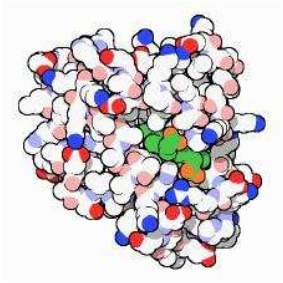
Pozitívny výber v duplikovaných génoch



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 nesynonymických 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

10 Štruktúra a funkcia proteínov



Proteíny

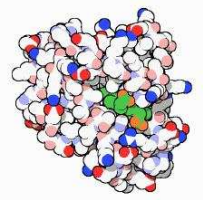
Reťazce 20 rôznych aminokyselín s rôznymi chemickými vlastnosťami:

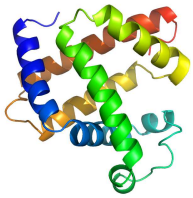
Amino Acid	Side chain	Hydrophobic	Polar	Charged	
Alanine (A)	-CH3	X	-	-	-
Arginine (R)	-(CH2)3NH-C(NH)NH2	-	X	basic	-
Asparagine (N)	-CH2CONH2	-	X	-	-
Aspartic acid (D)	-CH2COOH	-	X	acidic	-
Cysteine (C)	-CH2SH	X	-	acidic	-
Glutamic acid (E)	-CH2CH2COOH	-	X	acidic	-
Glutamine (Q)	-CH2CH2CONH2	-	X	-	-
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
  56 800 proteínových štruktúr
  (UniProt má takmer má takmer 10 miliónov sekvencií)
- Štrukturálna genomika: snaha určovať štruktúry vo veľkom

Bioinformatický problém: určovanie štruktúry proteínov

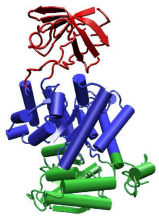
(protein structure prediction, protein folding)
Vstup: sekvencia proteínu
Výstup: 3D pozície atómov alebo aminokyselín

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)



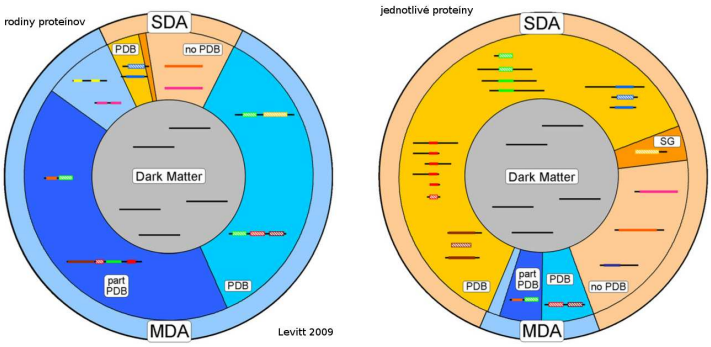
- Časť proteínu s nezávislou štruktúrou
- Veľa proteínov sa skladá z viacerých domén
- Domény sa tiež v proteínoch preskupujú
  počas evolúcie

Rodina (family)

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

Sekvencie, rodiny, štruktúry

SDA=single domain architecture, MDA=multidomain architecture
78% sekvencií aspoň jedna známa doména
Iba 12% rodín MDA proteínov nemá doménu so známou štruktúrou

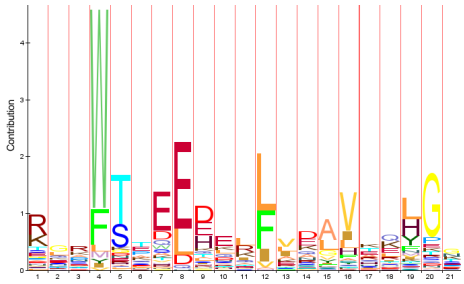


Hľadanie rodín

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 zachované 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 $q(L) = 70\%$ a alanín $q(A) = 30\%$
- Majme zarovananie 10 sekvencií s nasledujúcimi počtami

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

- 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
```

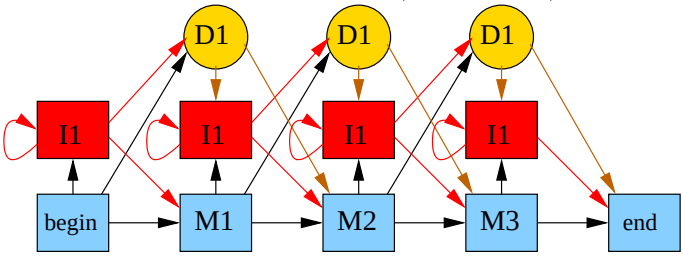
- Skóre LAAL je $0.19 + 1 + 1.58 + 0.36 = 3.13$
  Skóre ALAL je $-0.58 - 0.81 + 1.58 + 0.36 = 0.55$

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 $e_i(a) = \frac{E_i(a)+c}{\sum_b (E_i(b)+c)}$
- Podobne 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 profilový HMM (databáza Pfam)
- PSI-Blast vytvára PSSM za pochodu z podobných proteínov

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 threading
- Pre krátke proteíny môžem skúsiť minimalizovať energiu,
  inak získané štruktúry doplniť/vylepšiť minimalizáciou energie

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

Funkcia proteínu

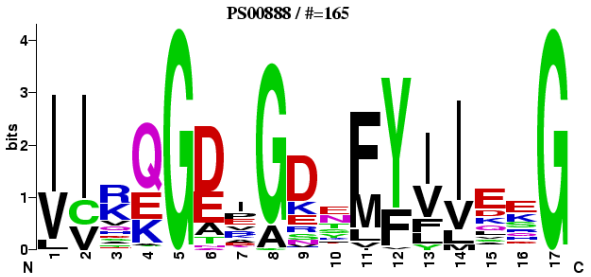
- Pre niektoré proteíny určená laboratórne
- Na ďalšie proteíny prenášame bioinformaticky pomocou podobnosti sekvencie, prítomnosti domén, polohy v genóme a ďalších dát
- Swissprot/Uniprot zhromažďuje údaje o funkcii proteínov
- Klasifikácie proteínu pomocou Gene ontology (GO)
  Hierarchická štruktúra pojmov:
  GO:0008150 biological process
  GO:0051179 localization
  GO:0051234 establishment of localization
  GO:0006810 transport
  GO:0006811 ion transport
  GO:0034220 ion transmembrane transport

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:

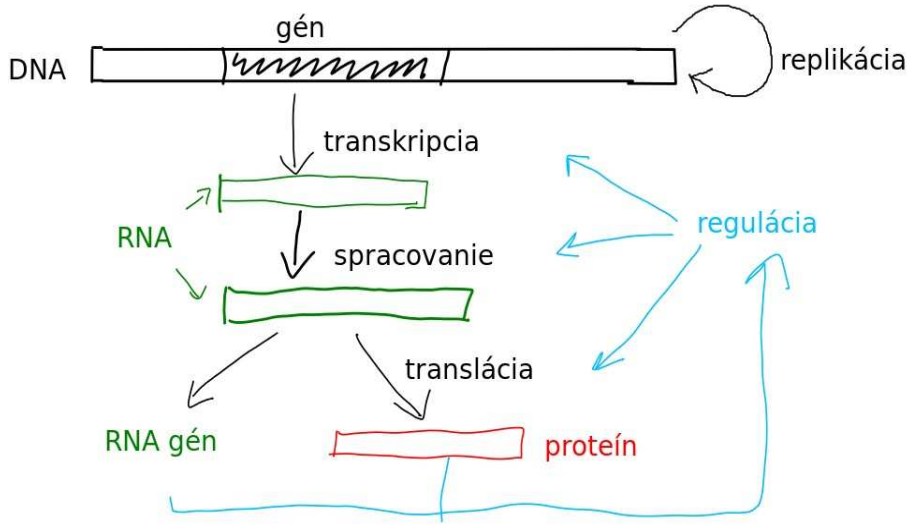
```
[LIVM]-[VIC]-x-{H}-G-[DENQTA]-x-[GAC]-{L}-x-[LIVMFY](4)-x(2)-G
```



11 Regulácia génovej expresie

Aká informácia je uložená v DNA?

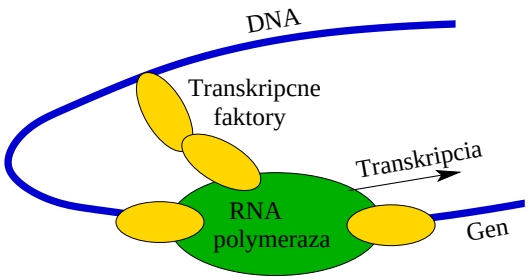
Gény: Predpisy na tvorbu proteínov a funkčných RNA molekúl.*Riadenie ich expresie:* kedy a koľko sa má tvoriť.



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

Transkripčné faktory

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

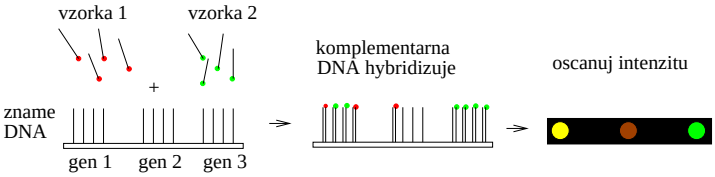


Človek má vyše 2000 TF-ov
Môžu zvyšovať alebo znižovať mieru expresie,
fungovať v skupinách

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

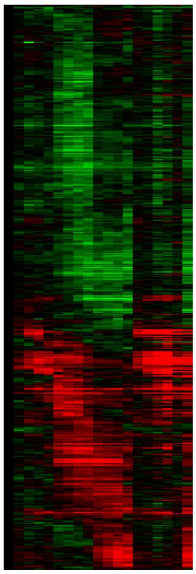
Pomer expresie génu v meranej a kontrolnej vzorke fg/bg

	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 a komparatívna genomika
- štruktúra a funkcia proteínov
- aj ďalšie tri prednášky

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 $\overline{y}$

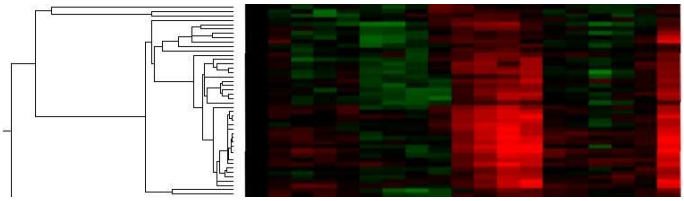
$$C(x,y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

Číslo od -1 do 1, 1 pre lineárne korelované dáta
Vzdialenosť $d(x,y) = 1 - C(x,y)$
Aj iné možnosti, napr. Euklidovská vzdialenosť

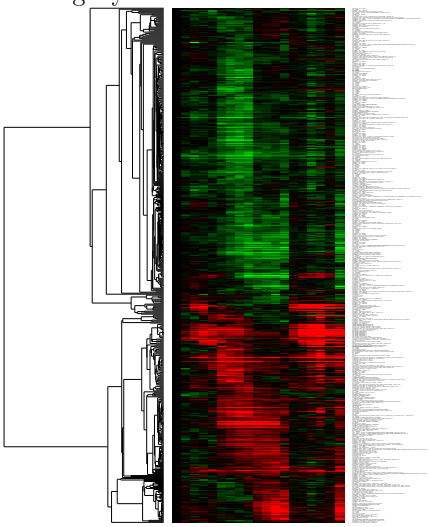
Hierarchické zhlukovanie

- 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

Príklad



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

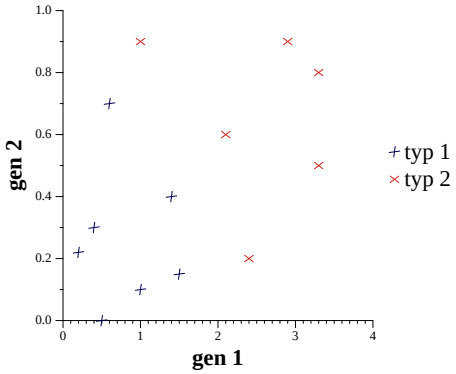


Klasifikácia

- 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

Hračkársky príklad: expresia 2 génov

Trénovacie dáta so známym typom:

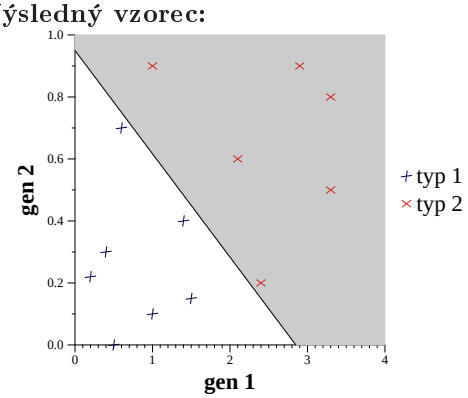


Typ vzorca: lineárne funkcie (lineárny diskriminant)
tumor typu 1 ak $ax + by + c < 0$

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

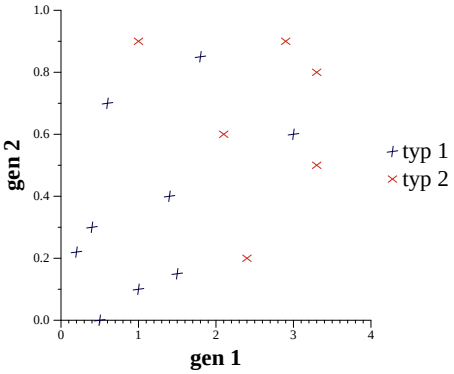
Hračkársky príklad: expresia 2 génov

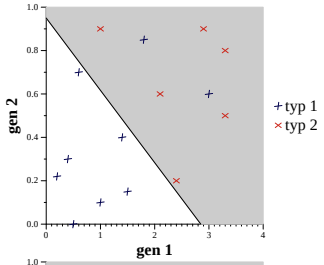
Výsledný vzorec:



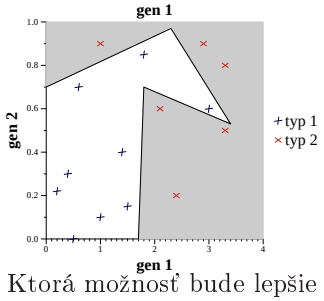
$a = 1, b = 3, c = -2.85$
tumor typu 1 ak $x + 3y - 2.85 < 0$

Nie vždy vieme nájsť dokonalé oddelenie priamkou





Možnosť 1:
zmierime sa s niekoľkými chybami
na trénovacích dátach.

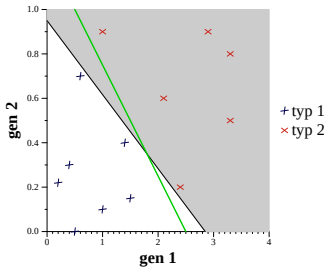


Možnosť 2:
rozšírime triedu hypotéz,
napr. na lomené čiary
s najviac 4 segmentami.

Ktorá možnosť bude lepšie fungovať na nových dátach?

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

Biologické siete

Reprezentácia veľkého množstva objektov a vzťahov pomocou *grafu*.

Regulačné siete: vrcholy sú gény, orientovaná hrana z A do B ak A reguluje B

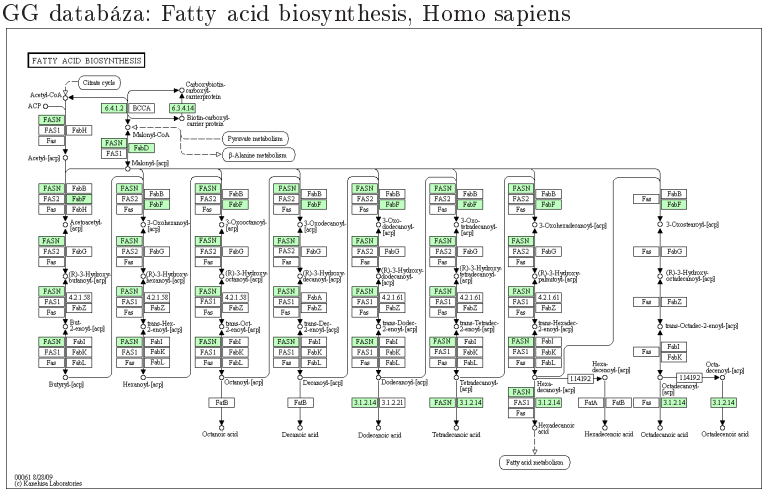
Siete interakcií: vrcholy sú proteíny, neorientovaná hrana (A, B) ak A a B fyzicky interagujú

Metabolické siete: vrcholy sú produkty metabolizmu, hrany predstavujú reakcie medzi nimi (a súvisiace enzýmy)

Iné: neorientované grafy, reprezentujúce korelovanú expresiu génov, koregulované gény a pod.

Metabolická sieť

KEGG databáza: Fatty acid biosynthesis, Homo sapiens



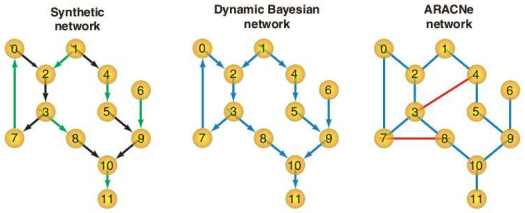
Použitie sietí

- Skúmanie globálnych vlastností:
  stupne vrcholov, vzdialenosti vrcholov v sieti
  porovnávanie s pravdepodobnostými modelmi grafov
- Porovnávanie sietí medzi organizmami, štúdium evolúcie sietí
- Hľadanie motívov v sieťach, napr. husté podgrafy v sieti interakcií často zodpovedajú proteínovým komplexom
- Štúdium lokálneho okolia zaujímavého génu
- Simulácia molekulárnej dynamiky

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

Zhrnutie

- Microarray nám môže dať informácie o úrovni expresie veľa génov naraz
- Problémy s normalizáciou a štatistickým spracovaním
- 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)
- Rôzne typy vzťahov môžeme reprezentovať ako sieť (graf)
- Microarray dáta pomáhajú zostaviť siete

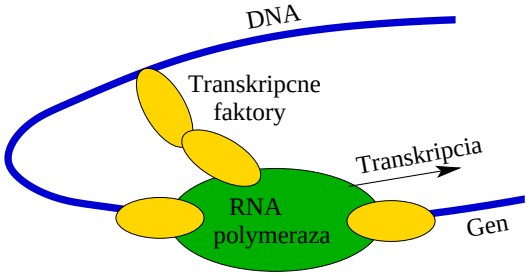
12 Väzobné motívy v DNA sekvenciách

Minulý týždeň

- Miroarray: technológia na meranie expresie (množstva mRNA) pre veľa génov naraz
- Vieme nájsť gény s podobným profilom expresie
- Možno majú ten istý regulátor (alebo jeden reguluje druhý)
- Dnes budeme skúmať, kam sa regulátory viažu v DNA

Transkripčné faktory (TF)

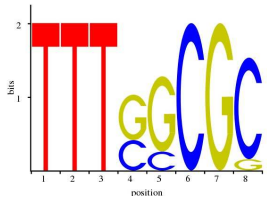
Regulácia začatia transkripcie pomocou transkripčných faktorov:
proteíny viažúce DNA, pomáhajú pritiahnuť RNA polymerázu

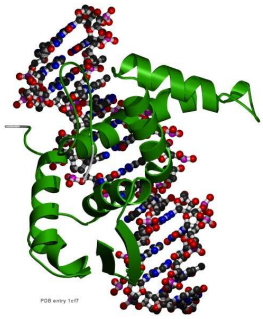


Človek má vyše 2000 TF-ov
Môžu zvyšovať alebo znižovať mieru expresie,
fungovať v skupinách

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, ako sme videli pri proteínových doménach
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

Videli sme pri proteínových doménach.
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
- Rozdiel medzi väzobnými miestami in vivo (v bunke) a in vitro (krátka DNA v skúmavke)
- Na zlepšenie špecifickosti hľadáme zhluky väzobných miest, miesta podporené experimentálne, evolučne zachované

Poznámky

- Databázy motívov, napr. TRANSFAC, JASPAR
- Podobne reprezentujeme aj motívy v proteínových sekvenciách a prvky génovej štruktúry (napr. miesta zostrihu)
- Motívy môžeme reprezentovať aj metódami strojového učenia:
  klasifikácia väzobných miest vs. zvyšok genómu
  pomocou neurónvých sietí, SVM, Bayesovských sietí a pod.

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 alebo detekuje pomocou expression array

Problém: zistíme len približnú polohu väzobného miesta

Bioinformatický problém: hľadanie nových motívov

- Máme niekoľko kusov DNA, z ktorých každý (resp. väčšina) by mal obsahovať väzobné miesto hľadaného TF.
- Získame ich pomocu ChIP,
  prípadne z microarray dát vezmeme sekvenciu okolo začiatku niekoľkých ko-regulovaných génov
- Chceme nájsť čo najšpecifickejší motív, ktorý sa vyskytuje v čo najviac vstupných sekvenciách.
- Rôzne spôsoby ako sformulovať ako presný bioinformatický problém,
  väčšina NP-ťažká
  v praxi exponenciálne algoritmy, heuristiky

Príklad formulácie problému hľadania motívov

Consensus Pattern Problem
Vstup: dĺžka motívu L, reťazce (sekvencie) $S_1, S_2, \ldots, S_k$
Výstup: motív (reťazec) S dĺžky L
a výskyt motívu v každom S_i (reťazec s_i dĺžky L)
také, že celkový počet nezhôd medzi S 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

Zhrnutie

- 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
- Hľadanie nových motívoch v experimentálne nájdených sekvenciách

13 RNA

Úvod do základných techník možno nájsť v kapitolách 9 a 10 knihy [Durbin et al., 1998].

Vlastnosti RNA

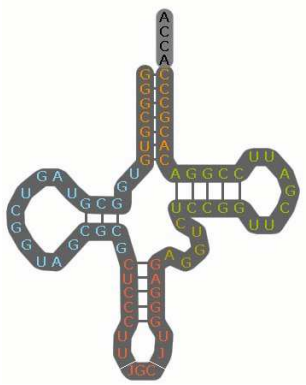
Ako sa líši od DNA?

- 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)

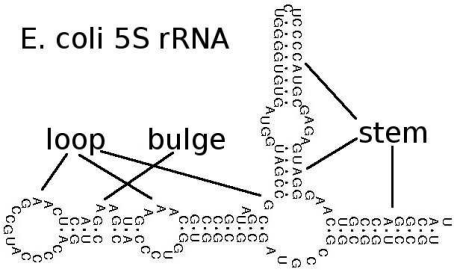
Sekundárna štruktúra(secondary structure):páry nukleotidov



Terciárna štruktúra(tertiary structure):3D súradnice

Sekundárna štruktúra RNA

Prvky sekundárnej štruktúry

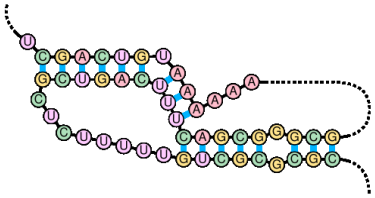


V tomto prípade spárované bázy tvoria *dobre uzátvorkovaný výraz:*

```
((((((((((.....(((   ).)).((   ))...)))))))))).
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 $i < j < i' < j'$.

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

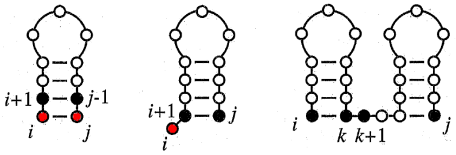
Vstup: RNA sekvencia
Cieľ: nájsť spárované bázy
Veľmi zjednodušená formulácia: nájdi dobre uzátvorkované spárovanie s najväčším počtom komplementárnych párov A-U, C-G. [Nussinov1978]

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)

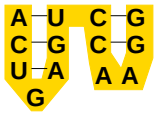
Rekurencia:
Podreťazce dĺžky 1: žiadne páry $A[i, i] = 0$
Dlhšie podreťazce: 3 prípady
– X_i a X_j sú pár: $A[i, j] = A[i + 1, j - 1] + 1$
– X_i je nespárované: $A[i, j] = A[i + 1, j]$
– X_i je pár s X_k pre $i < k < j$: $A[i, j] = A[i, k] + A[k + 1, j]$



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

	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

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



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 stemoch, dĺžky slučiek (loops) atď.Odvedené z nameraných dát. (pomerne nové parametre pozri [Mathews et al., 2004])

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] Existuje niekoľko implementácií napr. RNAstructure [Mathews et al., 2004], Vienna RNA package/webserver [Hofacker, 2003], mfold package/webserver [Zuker, 2003].

Algoritmy dovoľujúce pseudouzly

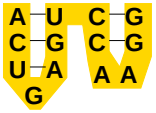
Vo všeobecnosti NP-ťažký problém [Lyngso and Pedersen, 2000]. Pomalé dyn. programovanie $O(n^4) - O(n^6)$ nájde niektoré typy pseudouzlov [Rivas and Eddy, 1999]. Tiež môžeme použiť heuristiky [Ren et al., 2005] (opakované vytváranie silných stemov).

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:
$S \to SS \to aSuS \to acSguS \to acuSaguS \to acugSaguS \to acugaguS \to acugagucSg \to acugaguccSgg \to acugaguccSagg \to acugaguccaSagg \to acuguguccaagg$



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. [Dowell2004]

Conditional log-linear models: [Do2006] 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

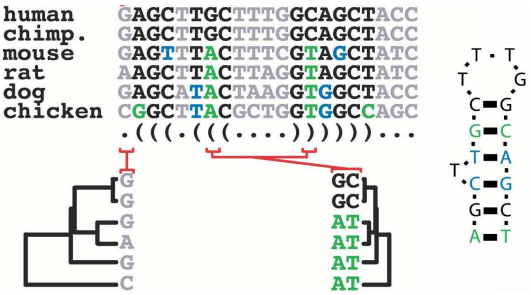
```
((((............))))
GCUCAGCC.CGGG...AGAGC
GCCUAGCC.UGGUCA.AGGGC
GUCUAGC...GGA...AGGAU
GAGCAGUU.CGGU...AGCUC
GUUCAAUC..GGU...AGAAC
```

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

stromu.(Pfold [Knudsen and Hein, 2003], Evofold [Pedersen et al., 2006]) Nespárované bázy emituje bežnou substitučnou maticou,spárované bázy substitučnou maticou dvojíc (16×16).



- 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

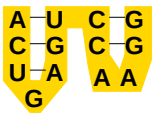
- Hľadaj úseky DNA so stabilnou sekundárnou štruktúrou
  (silnejší signál, ak máme zarovanie viacerých sekvencií).
- Výsledky treba normalizovať vzhľadom na dĺžku génu a GC%. Napr. RNAz [Washietl et al., 2005], Evofold [Pedersen et al., 2006].

Problém: hľadanie známych typov RNA génov v genóme

- Databáza Rfam: 1372 rodín podobných RNA génov [Griffiths-Jones2005]
- 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.



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

S =start, E_i =end
P_i =pár, L_i =nespárovaná báza vľavo, R_i =nespárovaná báza vpravo.Toto je iba úryvok z gramatiky. V skutočnosti, každý neterminál generuje rôzne bázy alebo páry s pravdepodobnosťou podľa stĺca zarovnania (napr. $P_1 \to aP_2u|uP_2a|cP_2g|cP_2u$). Tiež gramatika obsahuje ďalšie neterminály, ktoré modelujú indely.

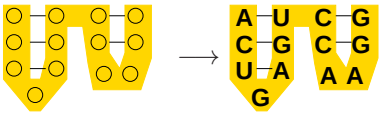
Použitie: hľadaj výskyty génu v DNA (local alignment),nájdi štruktúru nového génu z tej istej rodiny (global alignment).

Dynamické programovanie: čas $O(MND + M_b ND^2)$, kde$M$ = 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

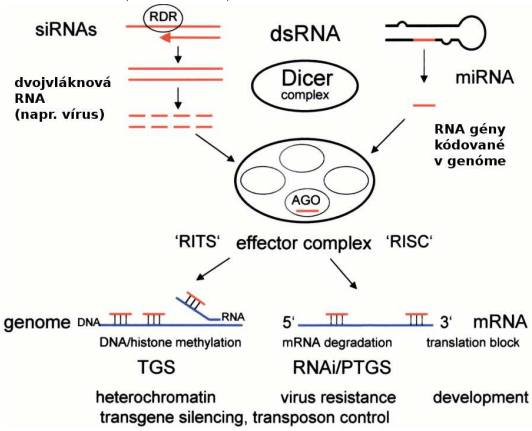
Daná RNA sekundárna štruktúra (párovanie).Nájdi sekvenciu, pre ktorú je táto štruktúra optimálna.Nie je známy efektívny algoritmus, heuristiky často nájdu sekvenciu pomerne rýchlo.[Andronescu2004]



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 stem-loop → 21-23nt jednovláknová RNA

lin-4

Cieľ: nájdi všetky miRNA gény v genóme

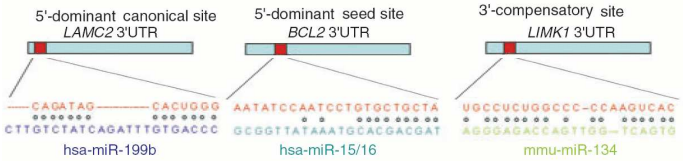
- Existencia transkriptu: EST databázy, prípadne over experimentom
- RNA štruktúra s dostatočne dlhým stemom

- Zachovaná sekvencia v genomických zarovnaniach; loop menej zachovaný ako stem
- Motívy nájdené v okolí stemu v časti génov

[Lai2003,Ohler2004]
Over experimentom: akumulácia transkriptov v neprítomnosti Diceru. [Ambros2003]

Problém: miRNA target prediction

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



[Sethupathy2006]

- (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)

Našli sa však aj miesta, ktoré nie sú evolučne zachované,vynechanie tretej podmienky však veľmi znižuje špecifickosť.

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

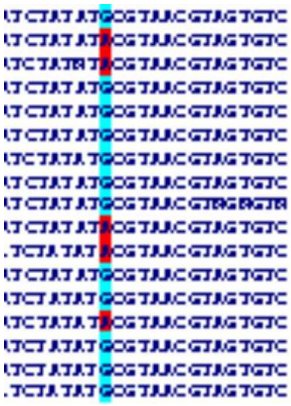
Existujú rôzne programy a webservery, napr. [Chalk et al., 2004].

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

14 Polymorfizmus genómov a diverzita

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

Ako vznikajú SNPy - jednoduchý model

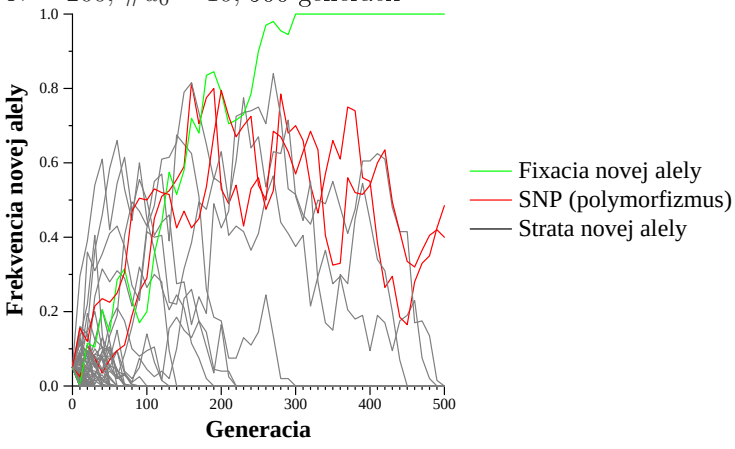
- Populácia N haploidných organizmov
- Jedna alela (forma A alebo a)
- Nová generácia vzniká "skopírovaním" náhodného rodiča (random mating), bez vplyvu prirodzeného výberu (no selection)
- *Markovovský proces* so stavmi $\#a \in \{0, 1, \ldots, N\}$

$$\Pr(\#a_t = j \,|\, \#a_{t-1} = i) = \left(\frac{i}{N}\right)^j \left(\frac{N-i}{N}\right)^{N-j} \binom{N}{j}$$

- Stavy 0 a N sú *pohlcujúce*; žiadny iný pevný bod

Náhodný genetický drift

$N = 200$, $\#a_0 = 10$, 500 generácií



- Mutácie spolu s náhodným genetickým driftom zavádzajú do populácie nové alely, ktoré po čase buď zaniknú, alebo ovládnu populáciu (fixation).

- Rýchlosť procesu je ovplyvnená efektami ako *štruktúra populácie* alebo *prirodzený výber* (selection).
- *Populačná genetika:*
  - Pravdepodobnostné modely pod vplyvom ďalších efektov
  - Odhady parametrov na základe údajov o súčasnej diverzite (frekvencií jednotlivých menšinových alel a pod.)
  - Odpovede na otázky o histórii populácie (aká stará populácia, aká veľká populácia), úlohe jednotlivých SNPov (škodlivé mutácie – deleterious, priaznivé mutácie – advantageous)

Dobrý úvod do problematiky: [Hartl, 2000]

Stabilný polymorfizmus - výhody diverzity
Pozri tiež [Dawkins, 2004, p.151-155]

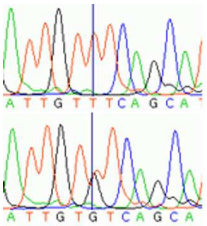
- *Farbosleposť opíc nového sveta* (New World monkeys) [Surridge2003]
  - niektoré jedince nevidia červenú časť spektra
  - niektoré jedince nevidia zelenú časť spektra
- *Kosáčiková anémia (sickle cell anemia)* [Passvol1978]
  - Aa: normálna bunka + *imunita voči malárii*
  - aa: poškodené krvinky

Polymorfizmus v diploidnom genóme

- Dve kópie každého chromozómu - z "otcovskej" a z "materskej" strany
- ⇒ Dve kópie každého SNPu:
  haplotyp: všetko okrem SNPov vynecháme, 0: menšinová alela, 1: väčšinová alela

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

- *Problém:* Resekvenovaním získame iba genotyp



Od genotypov k haplotypom

- Jeden genotyp: príliš málo informácie
- Skupina genotypov: ich haplotypy vznikli z malého množstva haplotypov (founding population) malým počtom mutácií ⇒ *počet rôznych haplotypov v populácii by mal byť malý*
- *Haplotype inference by pure parsimony (HIPP):* Pre danú *populáciu genotypov* nájdite *najmenší* počet haplotypov, z ktorých je možné poskladať všetky genotypy v populácii

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
```

Prístupy k haplotypovaniu

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

Rekombinácia a Linkage Disequilibrium (LD)

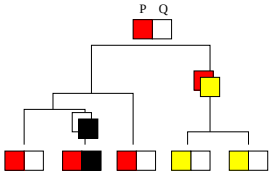
Závislosti medzi SNPmi

Uvažujme dva SNPy P, Q. Označme $p = \Pr(P = 0), q = \Pr(Q = 0)$

Na rozdielnych chromozómoch:

- Pravdepodobnosti výskytu jednotlivých alel sú nezávislé
- $\Pr(PQ = 00) = pq$, $\Pr(PQ = 11) = (1 - p)(1 - q)$, atď
- *linkage equilibrium (LE)*

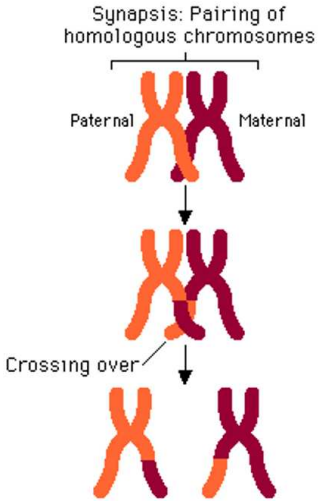
Na tom istom chromozóme:



- Málokedy mutácia na to istom mieste 2x
- Kombinácie nie sú úplne náhodné (jedna z kombinácia nemôže nikdy nastať ak horeuvedený predpoklad platí)
- Korelácie medzi SNPmi ⇒ *linkage disequilibrium (LD)*

Miera LD: ($D = 0$ ak dva SNPy sú v stave LE) $\boxed{D = \Pr(PQ = 00)\Pr(PQ = 11) - \Pr(PQ = 01)\Pr(PQ = 10)}$

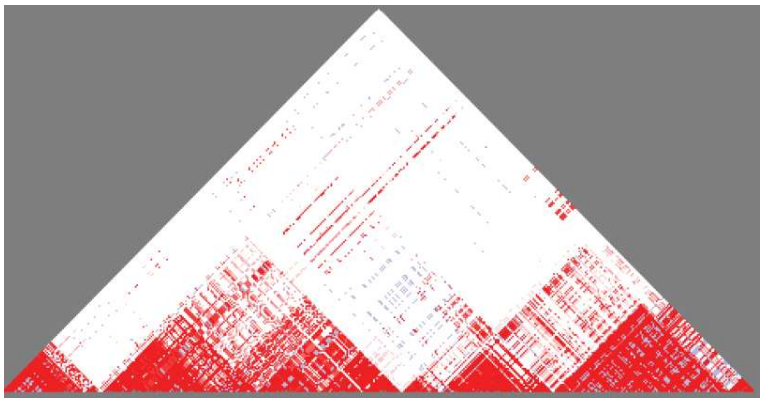
Rekombinácia znižuje LD



Ak predpokladáme uniformnú 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

Ako detekovať LD?

[Brown1975]

- Veľkosť D závisí od frekvencie jednotlivých alel (p a q)
- Aby sme mohli *porovnávať* medzi rôznymi SNPmi, potrebujeme normalizáciu
- Uvažujme nasledujúcu veličinu:

$$\rho = \frac{D}{\sqrt{p(1-p)q(1-q)}}$$

- Ak počet haplotypov vo vzorke je n, tak$\chi^2 = \rho^2 n$ 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	

- $\chi^2 = 2000\frac{(\Pr(PQ)\Pr(pq)-\Pr(Pq)\Pr(pQ))^2}{p(1-p)q(1-q)} = 2000\frac{0.0699^2}{0.053} = 184.78$
  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 $\rho^2 n$ by bola štatisticky bezvýznamná)

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

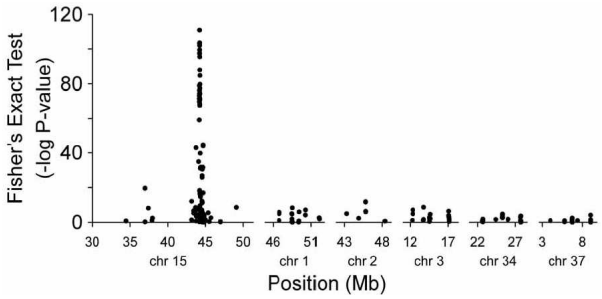
Počet haplotypov (chr15:44,228,468): [Sutter2007]

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

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

V tomto prípade: $P = 2.2 \times 10^{-16}$ (spočítané pomocou R)

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



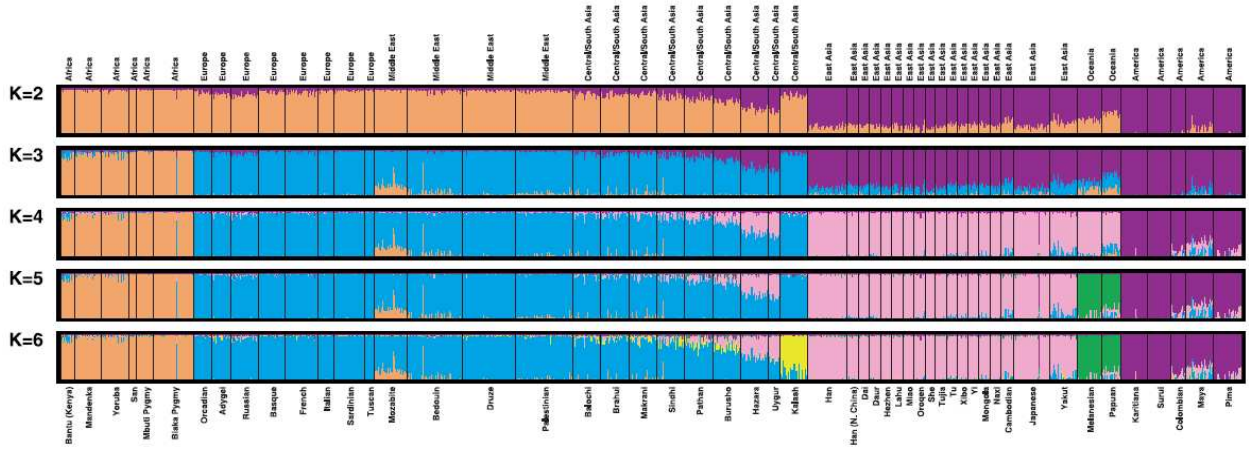
- V prípade štúdie veľkosti psov: WGAS identifikoval 84 kB región
- Pozíciu ďalej treba spresniť ďalšími experimentami
- *Malé LD bloky* $\Rightarrow$ potreba veľkého rozlíšenia SNPov
- *Veľké LD bloky* $\Rightarrow$ príliš veľké výsledné regióny

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 chromozómmi), ak pracujeme s viacerými subpopuláciami naraz
- $\Rightarrow$ chybné výsledky pri LD a WGAS

Štruktúra ľudskej populácie

[Rosenberg2002]



- Program STRUCTURE [Pritchard2000] 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
  - $Z_{i,j}$ - priradenie subpopulácie SNPu j v haplotype i
  - Q_i - aká časť SNPov v haplotype i patrí ku ktorej subpopulácii
- Model definuje $\Pr[X \mid P, Q, Z]$ a prior distribúcie 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

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

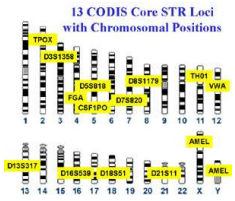
  konverguje k hodnote $E[Q \mid X]$

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 na mapovanie asociácií znakov (*whole-genome association mapping*)
- Pri LD analýzach treba brať do úvahy *štruktúru populácie*, ktorú možno odhadnúť pomocou výpočtových metód

Ďalšie typy polymorfizmov

- *Jednobázové indely* [Mills2006]
- *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)
- *Veľké úseky s variabilnou multiplicitou* (Large scale copy number variations)
  Napr. pozri [Shao et al., 2007]

Literatúra

[Allen and Salzberg, 2005] Allen, J. E. and Salzberg, S. L. (2005). JIGSAW: integration of multiple sources of evidence for gene prediction. *Bioinformatics*, 21(18):3596–3603.

[Altschul et al., 1990] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410.

[Altschul et al., 1997] Altschul, S. F., Madden, T. L., Schaffer, A. A., an d Z. Zhang, J. Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database se arch programs. *Nucleic Acids Research*, 25(17):3389–3392.

[Blanchette et al., 2004] Blanchette, M., Kent, W. J., Riemer, C., Elnitski, L., Smit, A. F. A., Roskin, K. M., Baertsch, R., Rosenbloom, K., Clawson, H., Green, E. D., Haussler, D., and Miller, W. (2004). Aligning multiple genomic sequences with the threaded blockset aligner. *Genome research*, 14(4):708–715.

[Brejová et al., 2005] Brejová, B., Brown, D. G., Li, M., and Vinař, T. (2005). ExonHunter: A comprehensive approach to gene finding. *Bioinformatics*, 21(Suppl 1):i57–i65. Proceedings of the 15th International Conference on Inteligent Systems for Molecular Biology (ISMB 2005).

[Burge and Karlin, 1997] Burge, C. B. and Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268(1):78–94.

[Chalk et al., 2004] Chalk, A. M., Wahlestedt, C., and Sonnhammer, E. L. L. (2004). Improved and automated prediction of effective siRNA. *Biochemical and biophysical research communications*, 319(1):264–264.

[Clark, 1990] Clark, A. G. (1990). Inference of haplotypes from PCR-amplified samples of diploid populations. *Molecular biology and evolution*, 7(2):111–112.

[Cooper and Hausman, 2004] Cooper, G. M. and Hausman, R. E. (2004). *The Cell: A Molecular Approach, 3rd ed.* Sinauer.

[Dawkins, 2004] Dawkins, R. (2004). The ancestor's tale.

[DeCaprio et al., 2007] DeCaprio, D., Vinson, J. P., Pearson, M. D., Montgomery, P., Doherty, M., and Galagan, J. E. (2007). Conrad: gene prediction using conditional random fields. *Genome research*, 17(9):1389–1398.

[Delcher et al., 1999] Delcher, A. L., Harmon, D., Kasif, S., White, O., and Salzberg, S. L. (1999). Improved microbial gene identification with GLIMMER. *Nucleic acids research*, 27(23):4636–4641.

[Dembo et al., 1994] Dembo, A., Karlin, S., and Zeitouni, O. (1994). Limit distributions of maximal non-aligned two-sequence segmental score. *The Annals of Probability*, 22:2022–2039.

[Durbin et al., 1998] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press.

[Edgar, 2004] Edgar, R. C. (2004). MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC bioinformatics*, 5:113.

[Felsenstein, 2004] Felsenstein, J. (2004). *Inferring Phylogenies*. Sinauer.

[Goldman and Yang, 1994] Goldman, N. and Yang, Z. (1994). A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Molecular biology and evolution*, 11(5):725–726.

[Gross and Brent, 2006] Gross, S. S. and Brent, M. R. (2006). Using multiple alignments to improve gene prediction. *Journal of computational biology : a journal of computational molecular cell biology*, 13(2):379–383.

[Gross et al., 2007] Gross, S. S., Do, C. B., Sirota, M., and Batzoglou, S. (2007). CONTRAST: a discriminative, phylogeny-free approach to multiple informant de novo gene prediction. *Genome biology*, 8(12):R269.

[Gusfield, 2002] Gusfield, D. (2002). Haplotyping as perfect phylogeny: conceptual framework and efficient solutions. In *RECOMB '02: Proceedings of the sixth annual international conference on Computational biology*, pages 166–175, New York, NY, USA. ACM Press.

[Gusfield, 2003] Gusfield, D. (2003). Haplotype Inference by Pure Parsimony. In *Combinatorial Pattern Matching (CPM 2003)*, Lecture Notes on Computer Science, pages 144–155. Springer.

[Hartl, 2000] Hartl, D. L. (2000). A primer of population genetics (3rd ed).

[Hasegawa et al., 1985] Hasegawa, M., Kishino, H., and Yano, T. (1985). Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of molecular evolution*, 22(2):160–164.

[Higgins et al., 1996] Higgins, D. G., Thompson, J. D., and Gibson, T. J. (1996). Using CLUSTAL for multiple sequence alignments. *Methods in enzymology*, 266:383–402.

[Hofacker, 2003] Hofacker, I. L. (2003). Vienna RNA secondary structure server. *Nucleic acids research*, 31(13):3429–3431.

[Karlin and Altschul, 1990] Karlin, S. and Altschul, S. F. (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc Natl Acad Sci U S A*, 87(6):2264–2268.

[Kent, 2002] Kent, W. J. (2002). BLAT–the BLAST-like alignment tool. *Genome Research*, 12(4):656–664.

[Kent et al., 2003] Kent, W. J., Baertsch, R., Hinrichs, A., Miller, W., and Haussler, D. (2003). Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. *Proceedings of the National Academy of Sciences of the United States of America*, 100(20):11484–9.

[Knudsen and Hein, 2003] Knudsen, B. and Hein, J. (2003). Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic acids research*, 31(13):3423–3428.

[Korf et al., 2001] Korf, I., Flicek, P., Duan, D., and Brent, M. R. (2001). Integrating genomic homology into gene structure prediction. *Bioinformatics*, 17(S1):S140–148. ISMB 2001.

[Krogh, 1997] Krogh, A. (1997). Two methods for improving performance of an HMM and their application for gene finding. In *Proceedings of the fifth International Conference on Intelligent Systems for Molecular Biology (ISMB 1997)*, pages 179–186.

[Lin et al., 2007] Lin, M. F., Carlson, J. W., Crosby, M. A., Matthews, B. B., Yu, C., Park, S., Wan, K. H., Schroeder, A. J., Gramates, L. S., St Pierre, S. E., Roark, M., Wiley Jr., K. L., Kulathinal, R. J., Zhang, P., Myrick, K. V., Antone, J. V., Celniker, S. E., Gelbart, W. M., and Kellis, M. (2007). Revisiting the protein-coding gene catalog of Drosophila melanogaster using 12 fly genomes. *Genome Res*, 17(12):1823–1826.

[Lukashin and Borodovsky, 1998] Lukashin, A. V. and Borodovsky, M. (1998). GeneMark.hmm: new solutions for gene finding. *Nucleic acids research*, 26(4):1107–1115.

[Lyngso and Pedersen, 2000] Lyngso, R. B. and Pedersen, C. N. (2000). RNA pseudoknot prediction in energy-based models. *Journal of computational biology*, 7(3-4):409–417.

[Majoros et al., 2004] Majoros, W. H., Pertea, M., and Salzberg, S. L. (2004). TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders. *Bioinformatics*, 20(16):2878–2879.

[Margulies et al., 2007] Margulies, E. H. et al. (2007). Analyses of deep mammalian sequence alignments and constraint predictions for 1% of the human genome. *Genome research*, 17(6):760–764.

[Mathews et al., 2004] Mathews, D. H., Disney, M. D., Childs, J. L., Schroeder, S. J., Zuker, M., and Turner, D. H. (2004). Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences of the United States of America*, 101(19):7287–7292.

[Matzke and Matzke, 2004] Matzke, M. A. and Matzke, A. J. M. (2004). Planting the seeds of a new paradigm. *PLoS biology*, 2(5):E133.

[Needleman and Wunsch, 1970] Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–443.

[Niu et al., 2002] Niu, T., Qin, Z. S., Xu, X., and Liu, J. S. (2002). Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *American journal of human genetics*, 70(1):157–159.

[Nussinov et al., 1978] Nussinov, R., Piecznik, G., Grigg, J., and Kleitman, D. (1978). Algorithms for loop matchings. *SIAM Journal on Applied Mathematics*, 35:68–82.

[Pace, 1997] Pace, N. R. (1997). A molecular view of microbial diversity and the biosphere. *Science*, 276(5313):734–740.

[Pearson and Lipman, 1988] Pearson, W. R. and Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the United States of America*, 85(8):2444–2448.

[Pedersen et al., 2006] Pedersen, J. S., Bejerano, G., Siepel, A., Rosenbloom, K., Lindblad-Toh, K., Lander, E. S., Kent, J., Miller, W., and Haussler, D. (2006). Identification and classification of conserved RNA secondary structures in the human genome. *PLoS Comput Biol*, 2(4):e33.

[Ren et al., 2005] Ren, J., Rastegari, B., Condon, A., and Hoos, H. H. (2005). HotKnots: heuristic prediction of RNA secondary structures including pseudoknots. *RNA*, 11(10):1494–1494.

[Rivas and Eddy, 1999] Rivas, E. and Eddy, S. R. (1999). A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of molecular biology*, 285(5):2053–2058.

[Sankoff, 1985] Sankoff, D. (1985). Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM Journal on Applied Mathematics*, 45(5):810–825.

[Shao et al., 2007] Shao, W., Tang, J., Song, W., Wang, C., Li, Y., Wilson, C. M., and Kaslow, R. A. (2007). CCL3L1 and CCL4L1: variable gene copy number in adolescents with and without human immunodeficiency virus type 1 (HIV-1) infection. *Genes and immunity*, 8(3):224–231.

[Siepel et al., 2005] Siepel, A., Bejerano, G., Pedersen, J. S., Hinrichs, A. S., Hou, M., Rosenbloom, K., Clawson, H., Spieth, J., Hillier, L. W., Richards, S., Weinstock, G. M., Wilson, R. K., Gibbs, R. A., Kent, W. J., Miller, W., and Haussler, D. (2005). Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome research*, 15(8):1034–1040.

[Siepel and Haussler, 2004] Siepel, A. and Haussler, D. (2004). Computational identification of evolutionarily conserved exons. In *RECOMB 2004: Proceedings of the 8th Annual International Conference on Research in Computational Molecular Biology*, pages 177–186. ACM Press.

[Smith and Waterman, 1981] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197.

[Solovyev et al., 2006] Solovyev, V., Kosarev, P., Seledsov, I., and Vorobyev, D. (2006). Automatic annotation of eukaryotic genes, pseudogenes and promoters. *Genome biology*, 7 Suppl 1:.1–12.

[Stanke et al., 2006] Stanke, M., Schoffmann, O., Morgenstern, B., and Waack, S. (2006). Gene prediction in eukaryotes with a generalized hidden Markov model that uses hints from external sources. *BMC bioinformatics*, 7:62.

[Stanke and Waack, 2003] Stanke, M. and Waack, S. (2003). Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics*, 19(S2):II215–II225. ECCB 2003.

[Stephens et al., 2001] Stephens, M., Smith, N. J., and Donnelly, P. (2001). A new statistical method for haplotype reconstruction from population data. *American journal of human genetics*, 68(4):978–979.

[The International HapMap Consortium, 2005] The International HapMap Consortium (2005). A haplotype map of the human genome. *Nature*, 437(7063):1299–1300.

[Washietl et al., 2005] Washietl, S., Hofacker, I. L., and Stadler, P. F. (2005). Fast and reliable prediction of noncoding RNAs. *Proceedings of the National Academy of Sciences of the United States of America*, 102(7):2454–2459.

[Yang and Nielsen, 2002] Yang, Z. and Nielsen, R. (2002). Codon-substitution models for detecting molecular adaptation at individual sites along specific lineages. *Molecular biology and evolution*, 19(6):908–917.

[Zhang et al., 2005] Zhang, J., Nielsen, R., and Yang, Z. (2005). Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level. *Molecular biology and evolution*, 22(12):2472–2479.

[Zuker, 2003] Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic acids research*, 31(13):3406–3415.

[Zuker and Stiegler, 1981] Zuker, M. and Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic acids research*, 9(1):133–138.