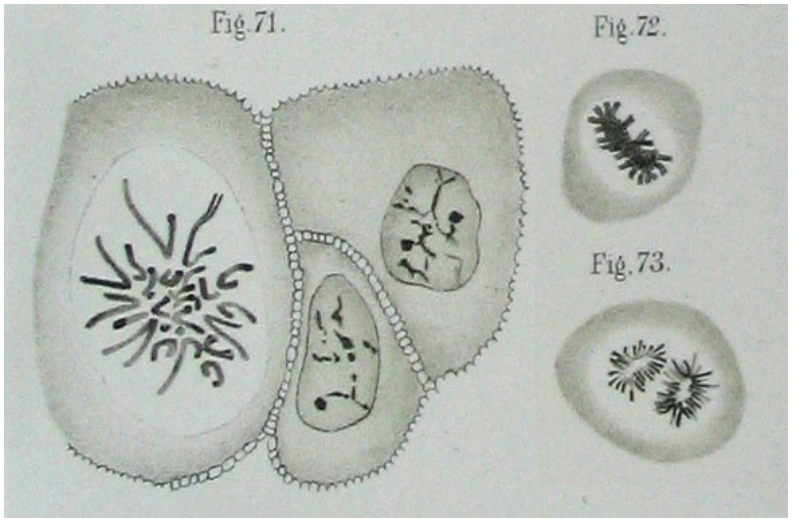
Biológia pre informatikov

Askar Gafurov

22.9.2022



Walther Flemming, 1881

Hlavné postavy

Deoxyribonukleová kyselina (DNA)

Obsahuje genetickú informáciu prenášanú z generácie na generáciu.

Dlhý reťazec nukleotidov z množiny $\{A, C, G, T\}$

(adenín, cytozín, guanín, tymín).

Informácia uložená v symbolickej, digitálnej forme.

Ribonukleová kyselina (RNA)

Blízka príbuzná DNA, tymín T nahradený uracylom U

Proteíny (bielkoviny)

Katalyzujú biochemické reakcie v bunke (enzýmy),

prenášajú signály v rámci bunky/medzi bunkami,

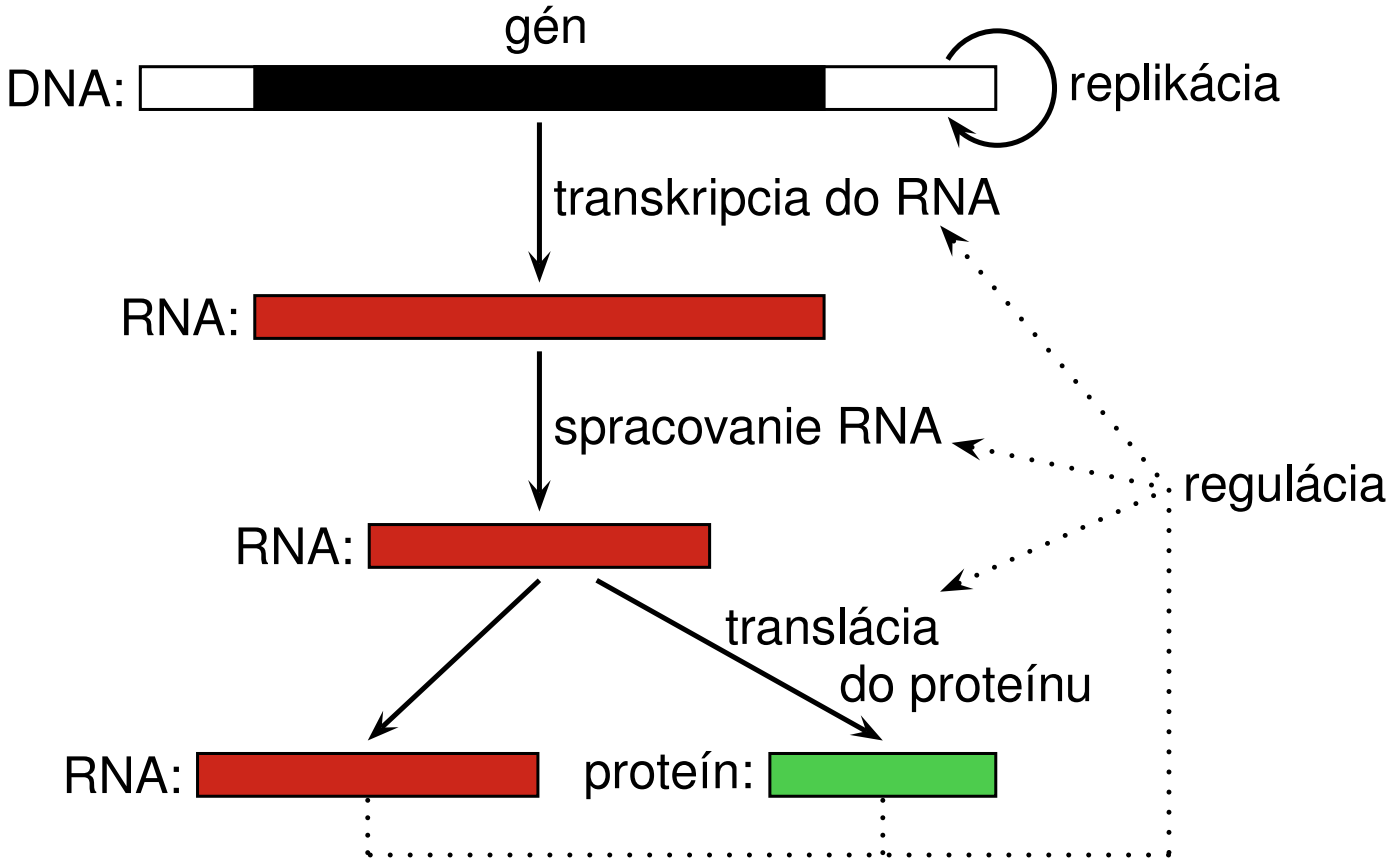
sú dôležité pre stavbu bunky a pohyb.

Reťazec aminokyselín (20 rôznych aminokyselín).

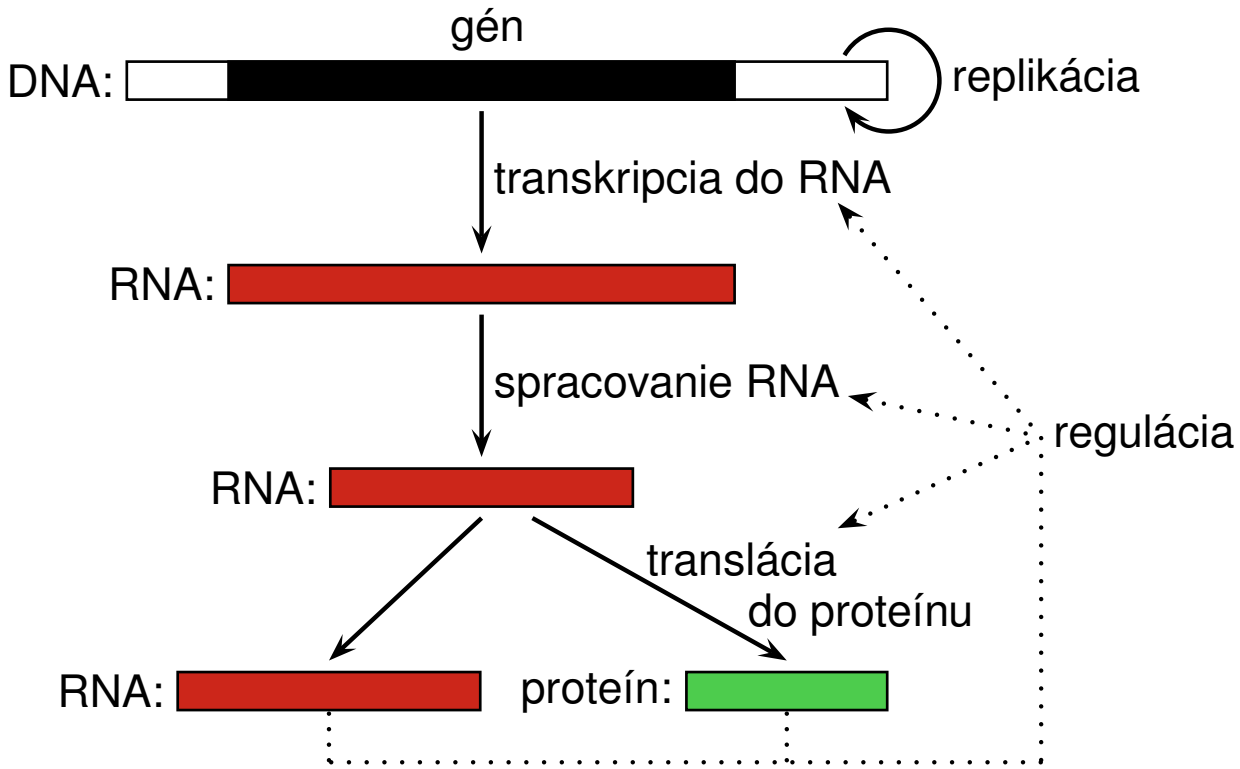
Aká informácia je uložená v DNA?

Gény: Predpisy na tvorbu proteínov a funkčných RNA molekúl.

Riadenie ich expresie: kedy a koľko sa má tvoriť.



Centrálna dogma (Francis Crick 1958,1970)

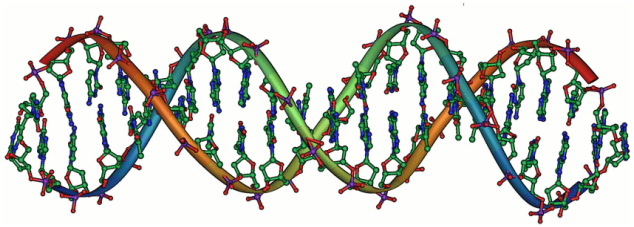


"The central dogma of molecular biology deals with the detailed residue-by-residue transfer of sequential information. It states that such information cannot be transferred back from protein to either protein or nucleic acid."

DNA, chromozómy

DNA: dve komplementárne vlákna, strands (páry A-T, C-G),
v opačnej orientácii (konce sa nazývajú 5’ a 3’).
Napr. ACCATG je komplementárny s CATGGT.

Tvar dvojitej špirály:



Dvojvláknová štruktúra poskytuje redundanciu, možnosť opravy pri poškodení jedného vlákna.
Pri delení bunky sa dvojvláknová DNA rozdelí a ku každému vláknu sa doplní komplement (DNA replikácia).

Chromozóm: Súvislý úsek dvojvláknovej DNA a podporných proteínov.

Ľudský genóm má 22 párov chromozómov plus dva pohlavné,
spolu 3GB.

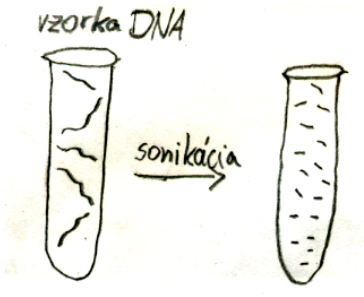
Technológia: sekvenovanie DNA

- Postup na zisťovanie poradia báz v chromozómoch genómu.
- Chromozómy sa nasekajú na krátke kúsky,

  každý sa sekvenuje zvlášť

  napr. Sangerovým sekvenovaním.

  – využíva prírodné enzýmy, napr. DNA polymerázu

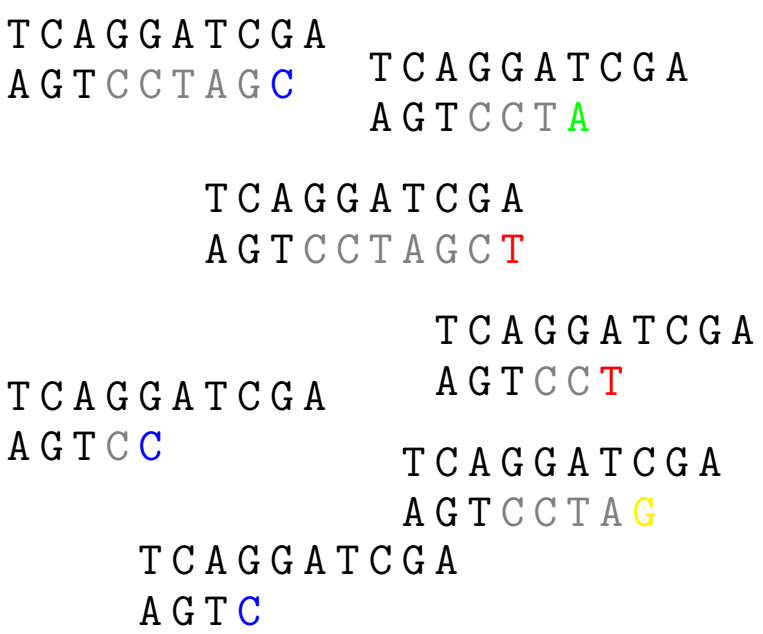


Sangerovo sekvenovanie (Sanger sequencing)

Sekvenujeme AGCTAGGACT (zobrazená sprava doľava)

Primer AGT + enzýmy + nukleotidy + modifikované ofarbené nukleotidy

Výsledky sekvenovacej reakcie:



Na géli zoradíme podľa dĺžky:

AGTCCTAGCT
AGTCCTAGC
AGTCCTAG
AGTCCTA
AGTCCT
AGTCC
AGTC

Odčítaním farieb dostaneme komplementárne vlákno: AGTCCTAGCT

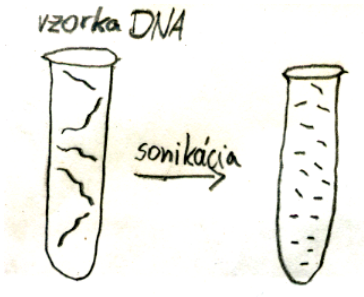
Technológia: sekvenovanie DNA

- Postup na zisťovanie poradia báz v chromozómoch genómu.
- Chromozómy sa nasekajú na krátke kúsky,

  každý sa sekvenuje zvlášť

  napr. Sangerovým sekvenovaním.

  – využíva prírodné enzýmy, napr. DNA polymerázu



- **Bioinformatický problém:** skladanie celej sekvencie z kúskov.
- Dostupnosť genómov umožňuje

  katalogizovať gény a iné funkčné úseky,

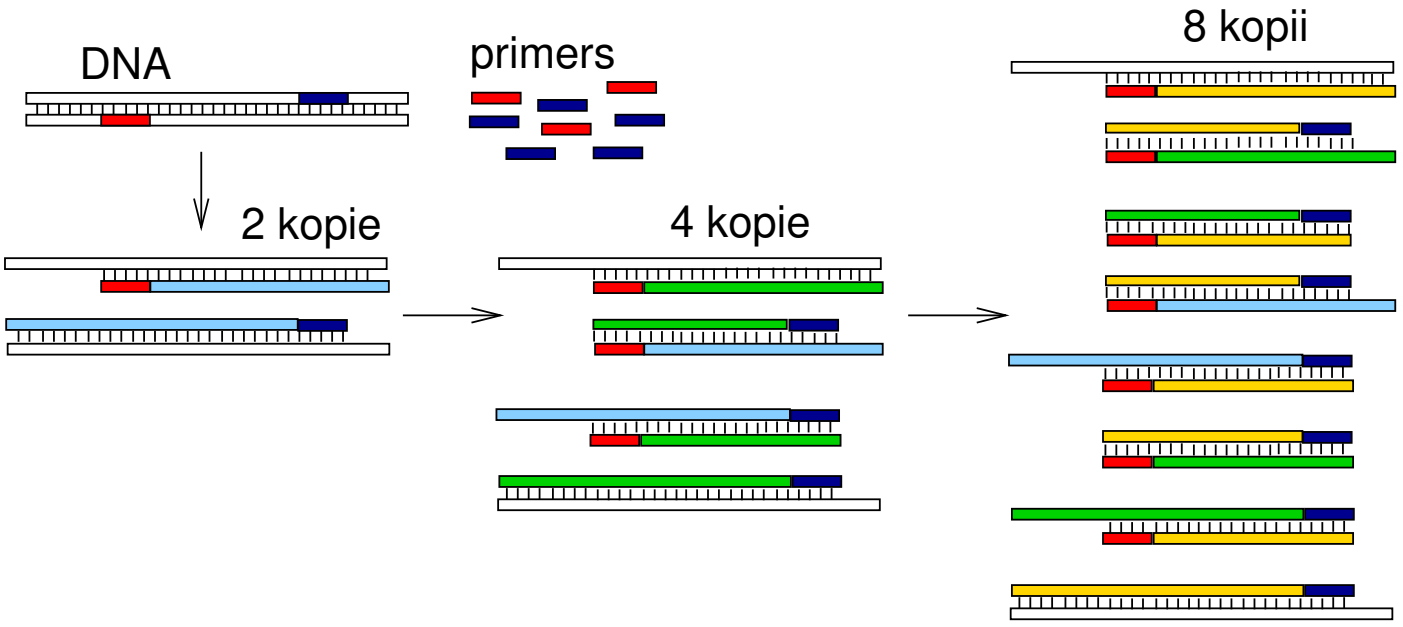
  hľadať podobnosti a rozdiely medzi druhmi a jedincami.

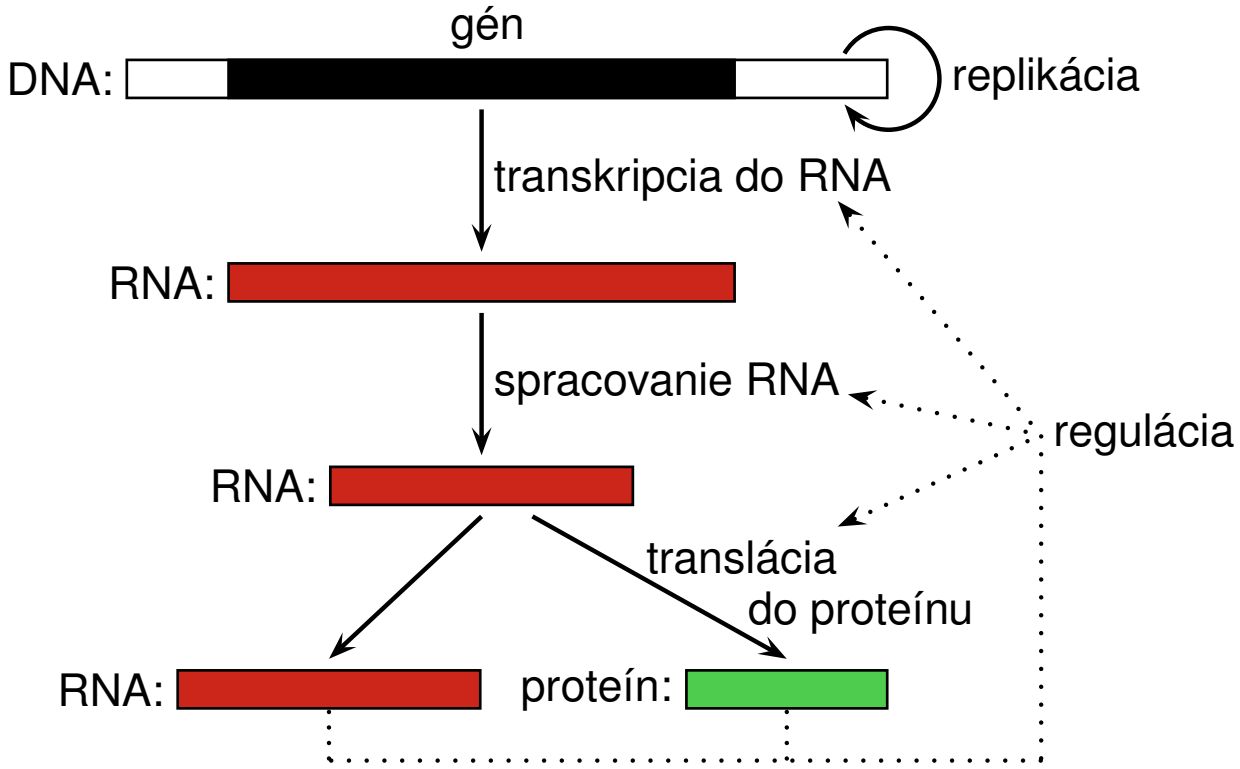
PCR (polymerase chain reaction)

Zvolíme si dva krátke úseky DNA (primers)

PCR testuje či sú v DNA blízko seba (stovky, tisíce báz)

Ak áno, namnoží úsek medzi nimi

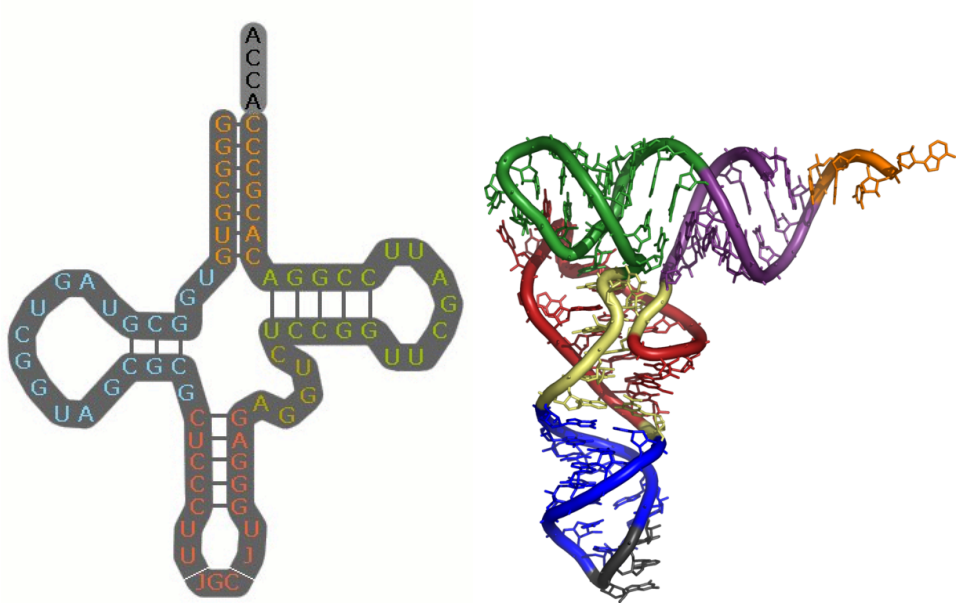



gén
DNA:
replikácia
transkripcia do RNA
RNA:
spracovanie RNA
regulácia
RNA:
translácia
do proteínu
RNA:
proteín:

RNA

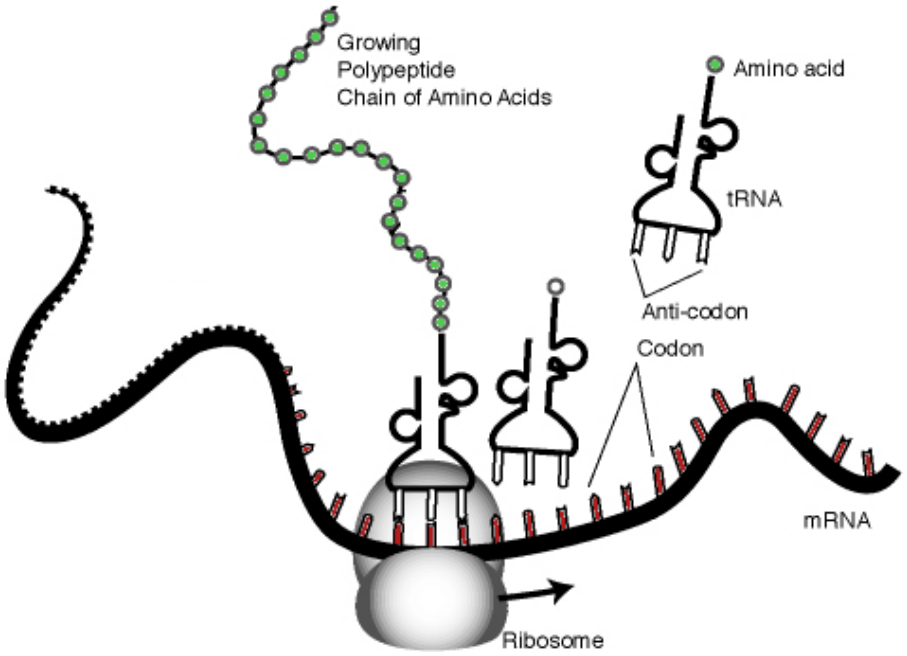
Ako sa líši od DNA?

- obsahuje ribózu namiesto deoxyribózy
- obsahuje uracil namiesto tymínu (bázy A,C,G,U)
- jednovláknové reťazce, zvyčajne kratšie
- zložitá sekundárna štruktúra: spárované komplementárne úseky

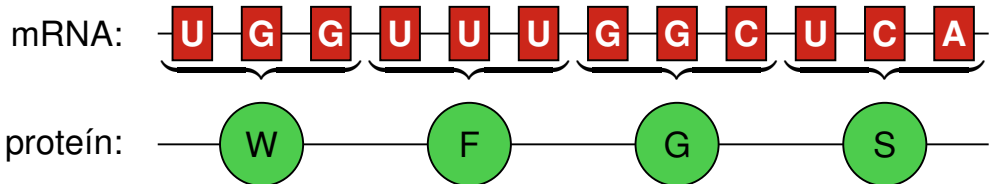


transferová RNA (tRNA)

Translácia



Kodón (trojica nukleotidov) určuje 1 aminokyselinu



Genetický kód

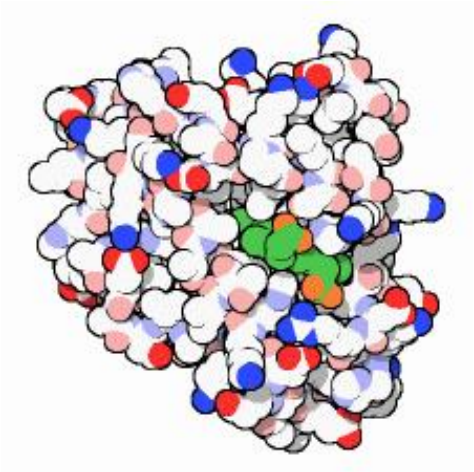
Ala / A	GCT, GCC, GCA, GCG	Leu / L	TTA, TTG, CTT, CTC, CTA, CTG
Arg / R	CGT, CGC, CGA, CGG, AGA, AGG	Lys / K	AAA, AAG
Asn / N	AAT, AAC	Met / M	ATG
Asp / D	GAT, GAC	Phe / F	TTT, TTC
Cys / C	TGT, TGC	Pro / P	CCT, CCC, CCA, CCG
Gln / Q	CAA, CAG	Ser / S	TCT, TCC, TCA, TCG, AGT, AGC
Glu / E	GAA, GAG	Thr /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

Proteíny

Reťazce 20 rôznych aminokyselín s rôznymi chemickými vlastnosťami:

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

Štruktúra proteínov

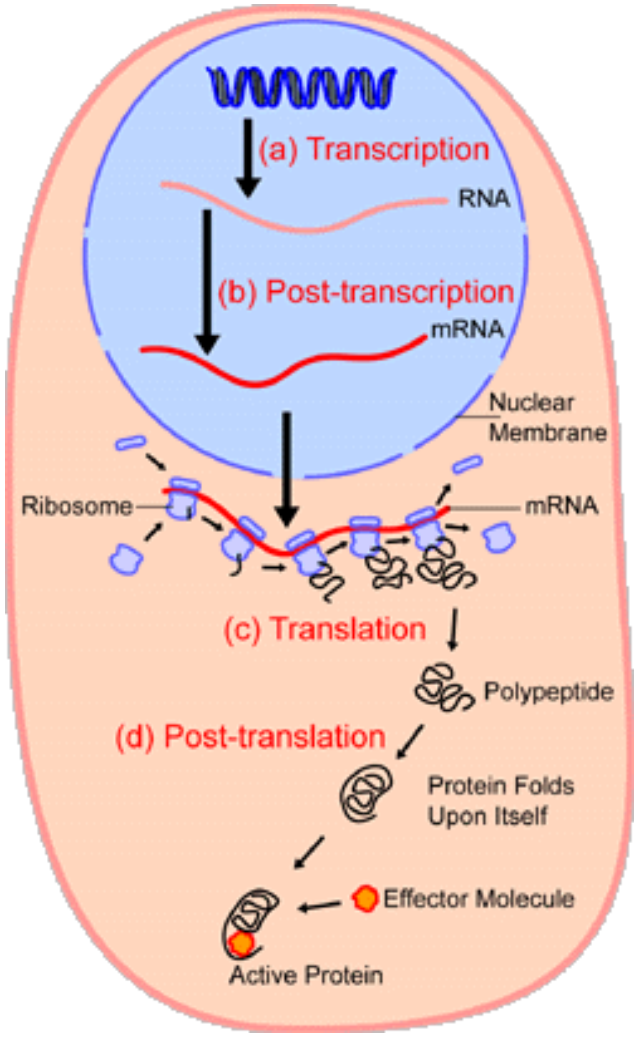


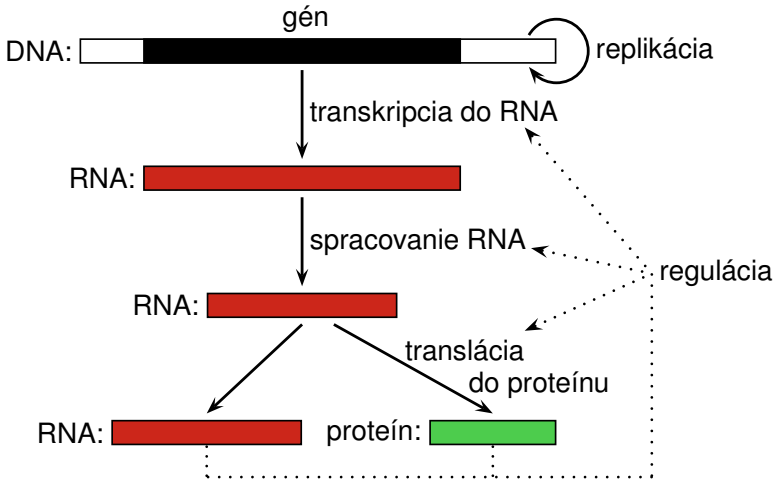
Myoglobín, prvý proteín so známou štruktúrou (Kendrew a kol. 1958).

Proteíny sa vyskytujú poskladané v určitej stabilnej štruktúre,
prípadne prechádzajú medzi niekoľkými stavmi.

Hydrofóbne aminokyseliny neinteragujú s vodou,
zväčša sa vyskytujú vo vnútri štrukúry.

Štruktúra proteínu určuje jeho funkciu.

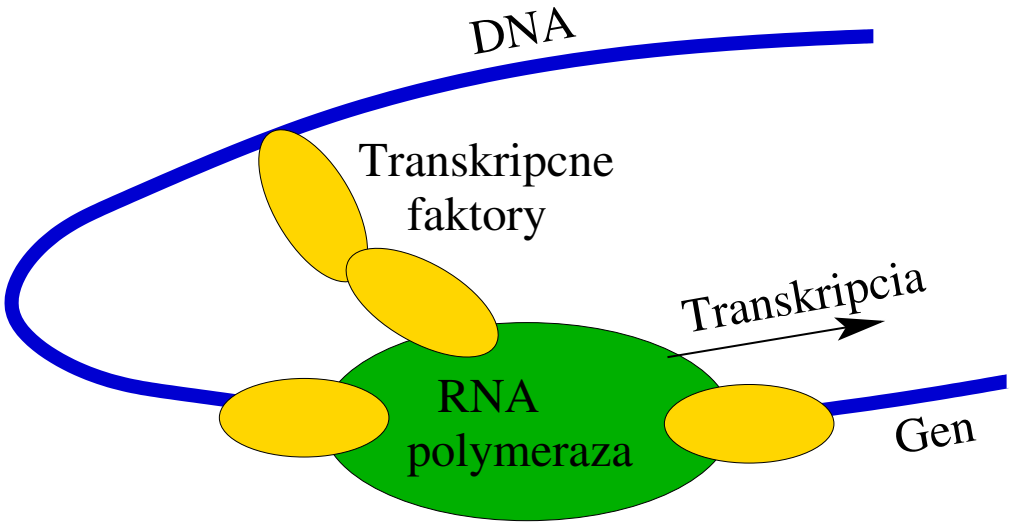

(a) Transcription
RNA
(b) Post-transcription
mRNA
Nuclear
Membrane
Ribosome
mRNA
(c) Translation
Polypeptide
(d) Post-translation
Protein Folds
Upon Itself
Effector Molecule
Active Protein


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

Regulácia expresie

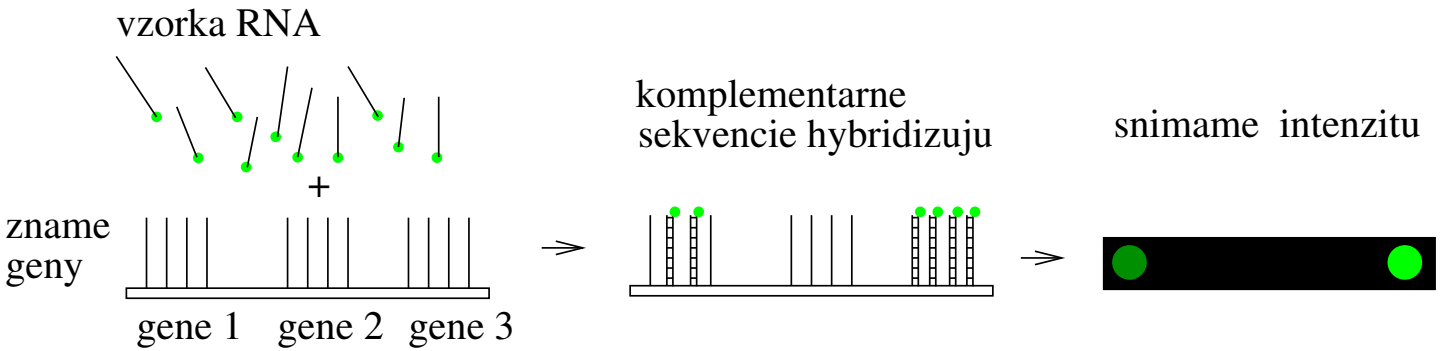
Bunky v rôznych tkanivách toho istého organizmu zdieľajú ten istý genóm, vyzerajú a fungujú však veľmi rôzne.
Niektoré proteíny sa tvoria len za určitých okolností, alebo v premenlivom množstve.

Regulácia začatia transkripcie pomocou transkripčných faktorov:



Bioinformatický problém: zisti, ktoré faktory ovplyvňujú ktorý gén, kde presne sa viažu.

Technológia: microarray



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

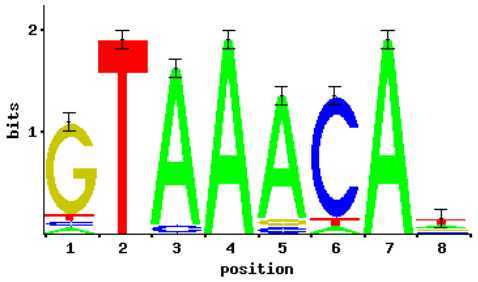
Zopakujeme za rôznych podmienok, študujeme korelácie medzi génmi.

Môžu byť dôsledkom spoločného regulátora (transkripčného faktoru).

Bioinformatický problém:

niekoľko ko-regulovaných génov,

nájdi motív, ku ktorému sa môže viazať spoločný transkripčný faktor **(motif finding)**



Príklad microarray dát

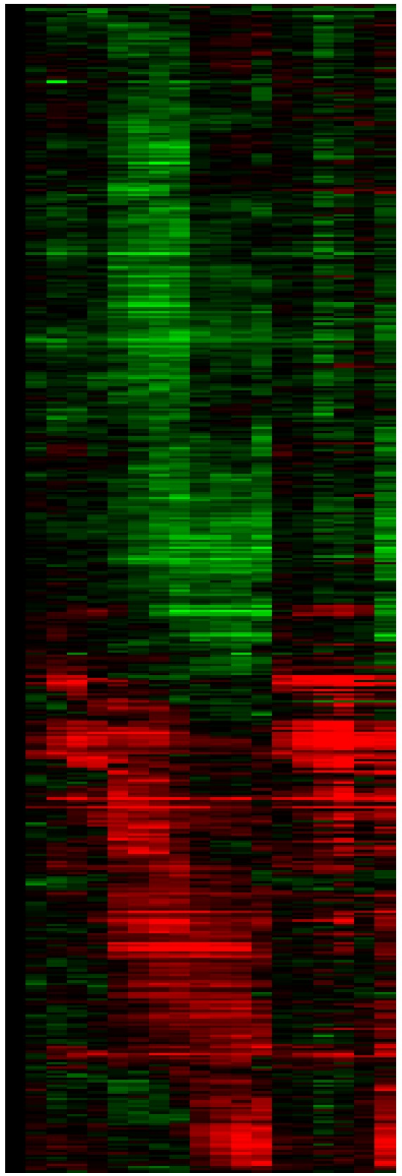
Pomer expresie génu v meranej a kontrolnej vzorke fg/bg

Červená: fg$>$bg

Zelená: fg$<$bg

517 génov

19 experimentov



Mutácie DNA

V DNA občas dochádza k zmenám, mutáciám

(napr. pod vplyvom prostredia, či chybou pri replikácii).

Typy mutácií:

substitúcia, substitution (jedna báza sa zmení na inú),

inzercia, insertion (vloží sa niekoľko nových báz),

delécia, deletion (vynechá sa niekoľko báz),

zmeny väčšieho rozsahu (napr. translokácie).

Bioinformatické problémy:

Ktoré sekvencie vznikli z spoločného predka mutovaním?

(hľadanie homológov, homology search)

Ktoré bázy v dvoch príbuzných sekvenciách si navzájom zodpovedajú?

(sequence alignment, zarovnávanie sekvencií)

Populačná genetika

Mutácie sa šíria v populácii z rodičov na potomkov.

Nebezpečné mutácie rýchlejšie vymiznú, prospešné sa rýchlejšie ujmú (prírodný výber, natural selection).

Polymorfizmus: genetický rozdiel medzi organizmami v rámci druhu.

Vedie k rozdielnosti vo fenotype, napr. výzor, dedičné choroby.

Sekvenovaním viacerých jedincov toho istého druhu získame prehľad o polymorfizme.

Bioinformatický problém:

Nájdi polymorfizmus zodpovedný za určitý znak (napr. chorobu).

Evolúcia

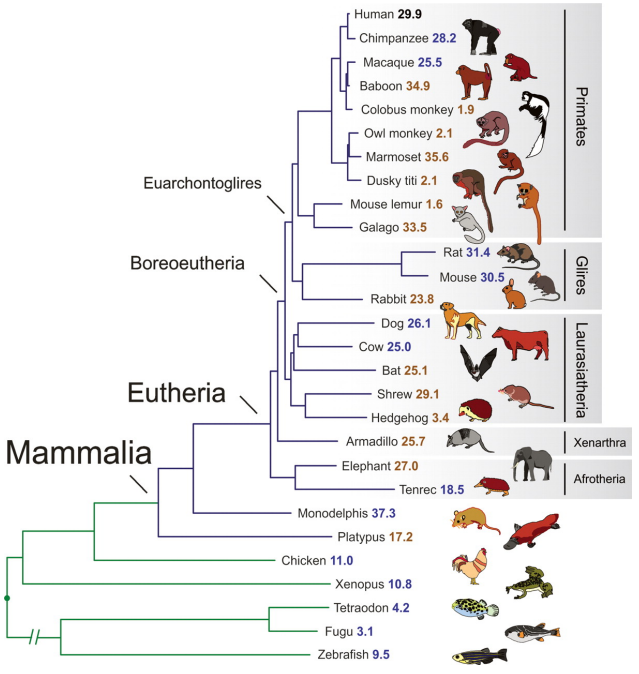
Vznik nových druhov (speciation):

Po rozdelení populácie na viacero oddelených častí nedochádza k výmene genetického materiálu.

Hromadia sa zmeny až kým nie je možné párenie: vznik nových druhov.

Bioinformatický problém:

Na základe dnešných sekvencií urči strom reprezentujúci vývoj druhov (fylogenetický strom, phylogenetic tree)



Prokaryotické vs. eukaryotické organizmy

Prokaryoty: baktérie, jednoduché jednobunkové organizmy.

Nemajú jadro (DNA priamo v cytoplazme),

majú kruhový chromozóm (a prípadné kratšie plasmidy),

jednoduchšia štruktúra génu atď.

Eukaryoty: živočíchy, rastliny, huby, niektoré jednobunkové organizmy.

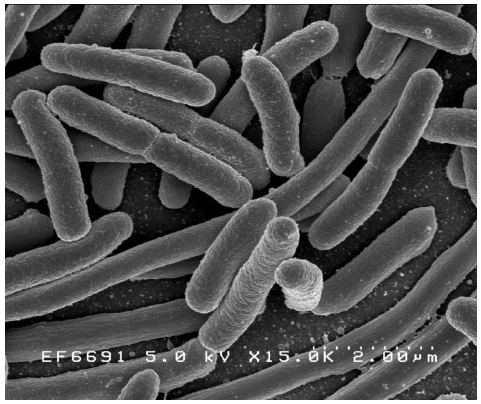
Bunka obsahuje jadro s DNA, viacero organel.

Mitochondrie a chloroplasty sú pohltené prokaryoty, ktoré sa stali časťou eukaryotickej bunky.

Dlhší genóm v niekoľkých lineárnych chromozómoch.

Modelové organizmy

Dôležité pre biologický výskum, vieme o nich viac než o príbuzných druhoch. Poznatky širšie aplikovateľné.



Escherichia coli: baktéria žijúca v črevách. Jednoduchá manipulácia, delenie každých 20 min. Štúdium základných životných procesov: DNA replikácia, expresia génov, atď. Genóm s 4000 génmi, 4.6MB.

Saccharomyces cerevisiae: pekárske droždie. Jednoduchý eukaryotický organizmus. Genóm s 6000 génmi, 13MB. Delenie každé 2 hodiny. Štúdium špecificky eukaryotických javov.

Modelové organizmy

Arabidopsis thaliana: malá kvitnúca rastlina, 6-týždňový životný cyklus. Skúmanie javov špecifických pre rastliny.

Caenorhabditis elegans: malý červ, nematód, žijúci v pôde. Štúdium vývinu (ontogenéza, development), diferenciácie buniek.

Drosophila melanogaster: vínna muška. Štúdium genetiky, gény riadiace vývin jedinca.

Stavovce: žaba Xenopus laevis (veľké, ľahko manipuľovateľné vajíčka), akvarijná ryba Danio rerio (priehľadné embryá), myš Mus musculus (existuje veľa plemien so špeciálnymi vlastnosťami).

Dostupné dáta

- DNA sekvencie: celé genómy, ich časti
- Ich anotácia: súradnice génov a iných funkčných častí
- Sekvencie RNA, ich štruktúra
- Sekvencie proteínov, ich funkcia a štruktúra
- Merania množstva RNA/proteínu v bunke
- ...

Dáta založené na experimentoch alebo výsledky výpočtových metód

Veľa chýb (v oboch prípadoch)

Ďalšie informácie

- Zvelebil, Baum: Understanding Bioinformatics, kap. 1
- Vysokoškolské učebnice molekulárnej biológie
- Anglická wikipédia
- Tutoriály na stránke predmetu

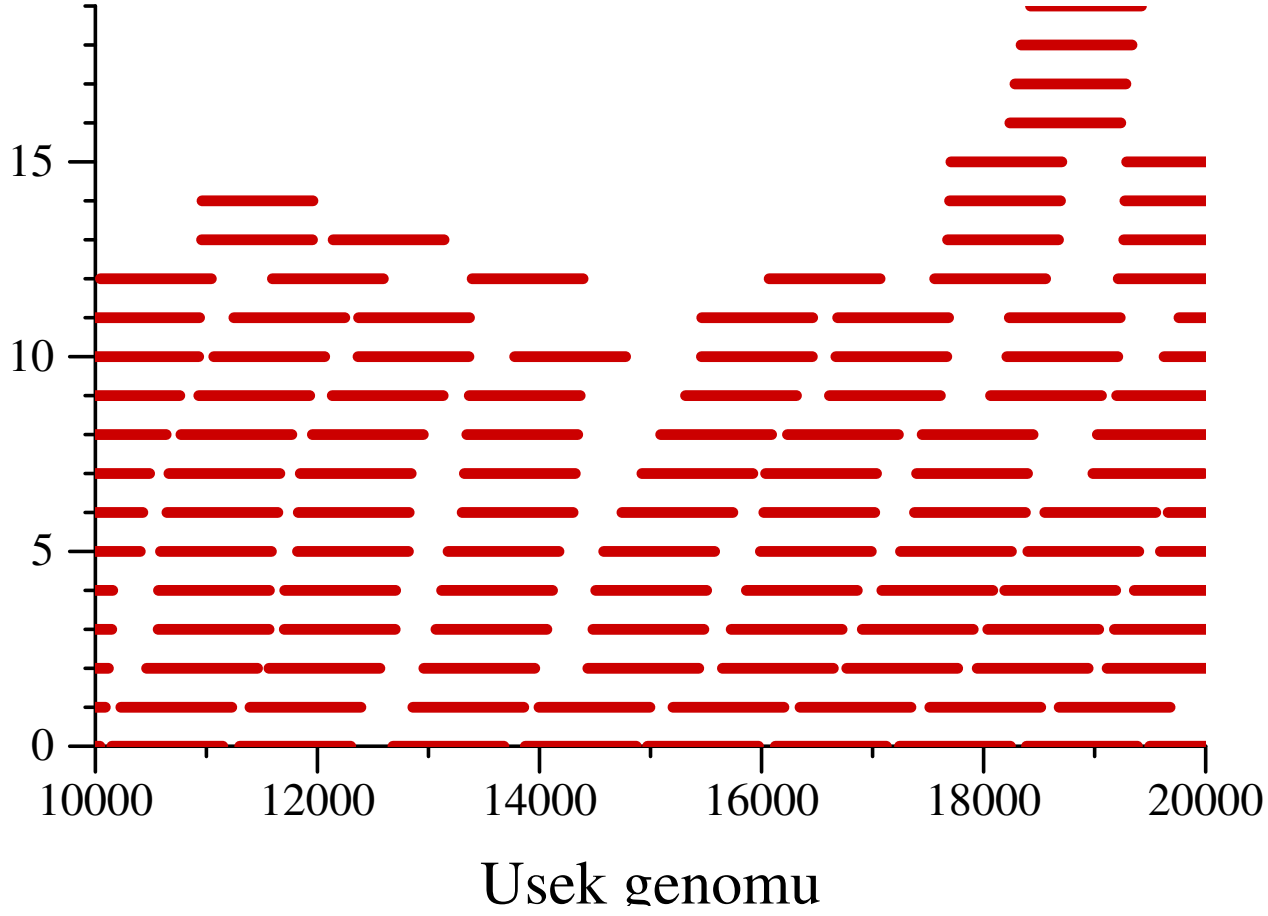
Úvod do pravdepodobnosti, sekvenovanie genómov (cvičenie)

Askar Gafurov

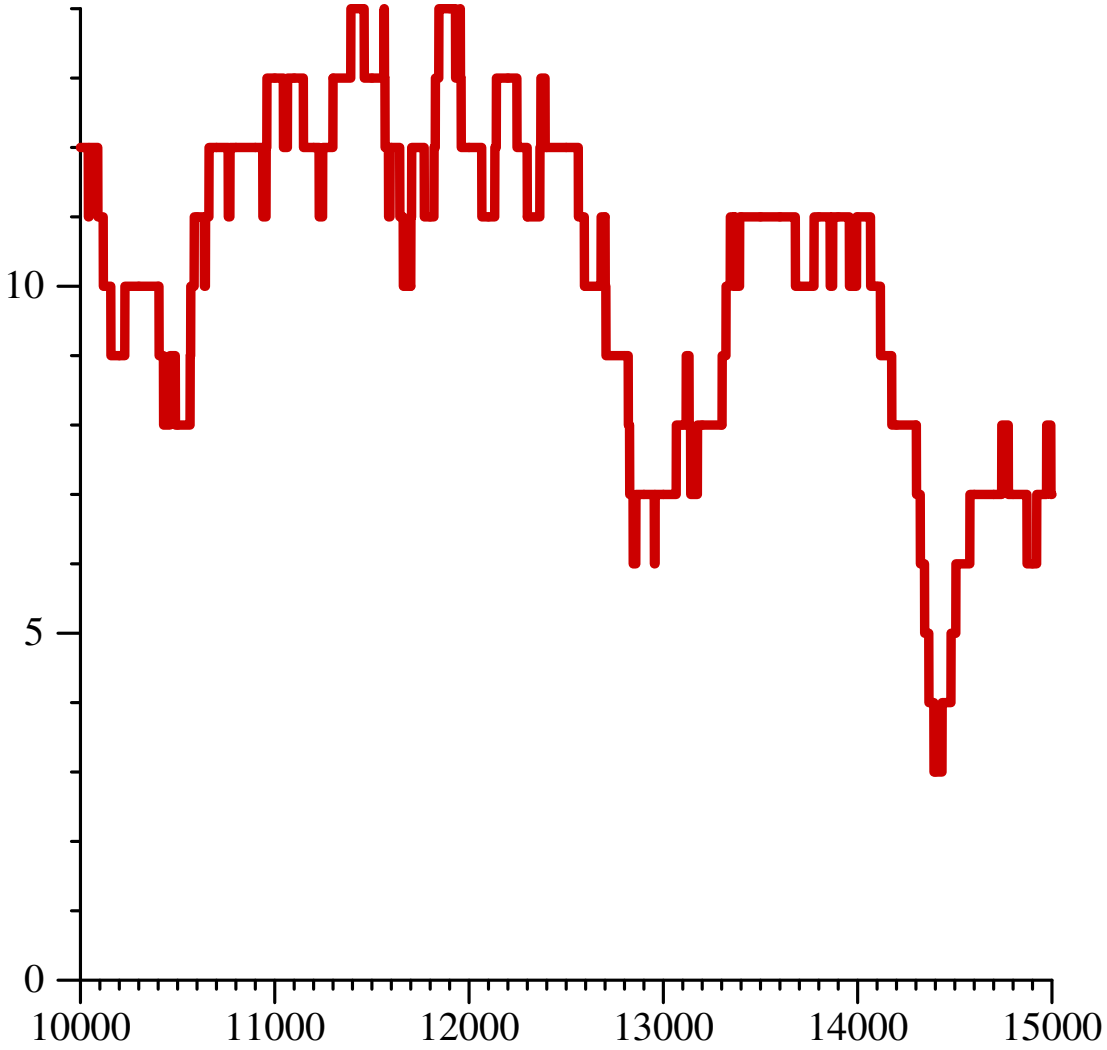
3.10.2019

- $G =$ dĺžka genómu, napr. 1 000 000
- $N =$ počet čítaní (readov), napr. 10 000
- $L =$ dĺžka čítania, napr. 1000
- $T =$ potrebná dĺžka prekryvu, napr. 50

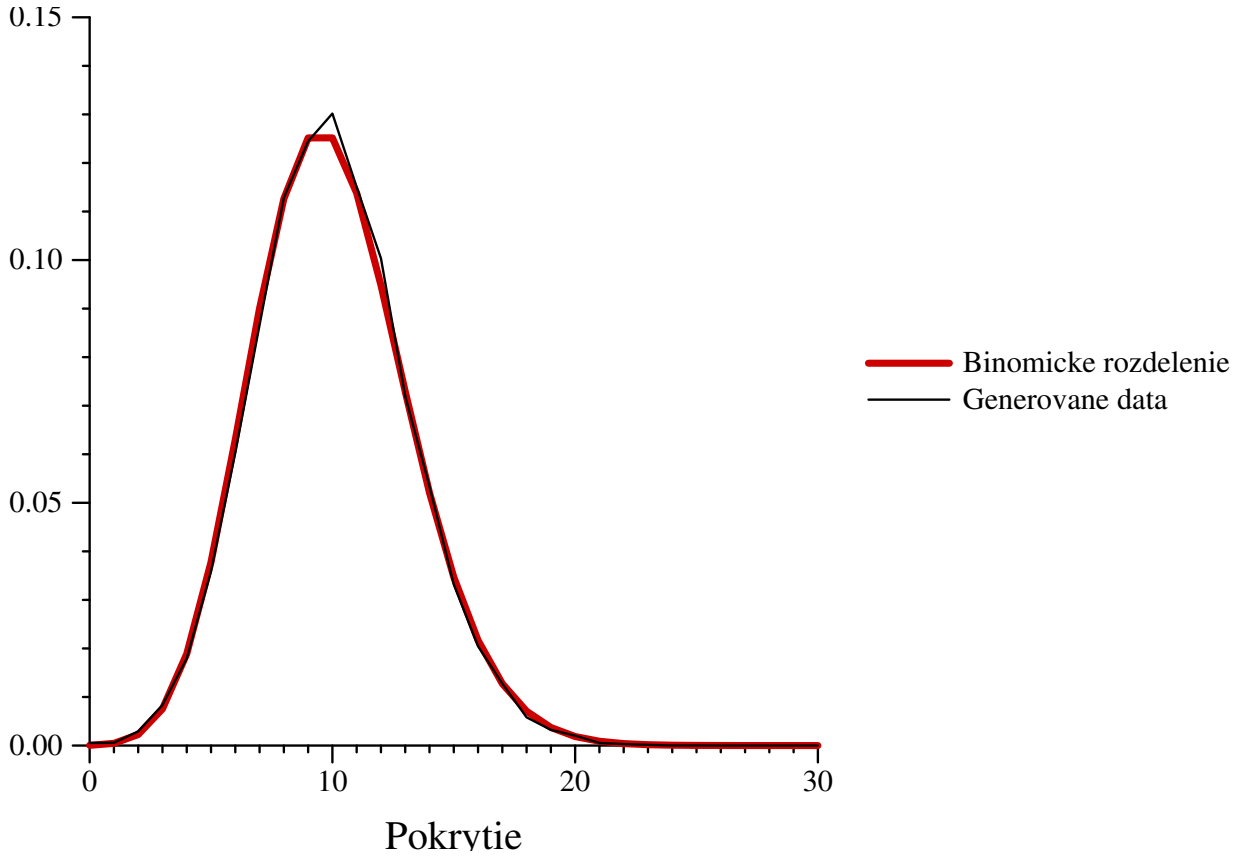
Náhodne generované čítania



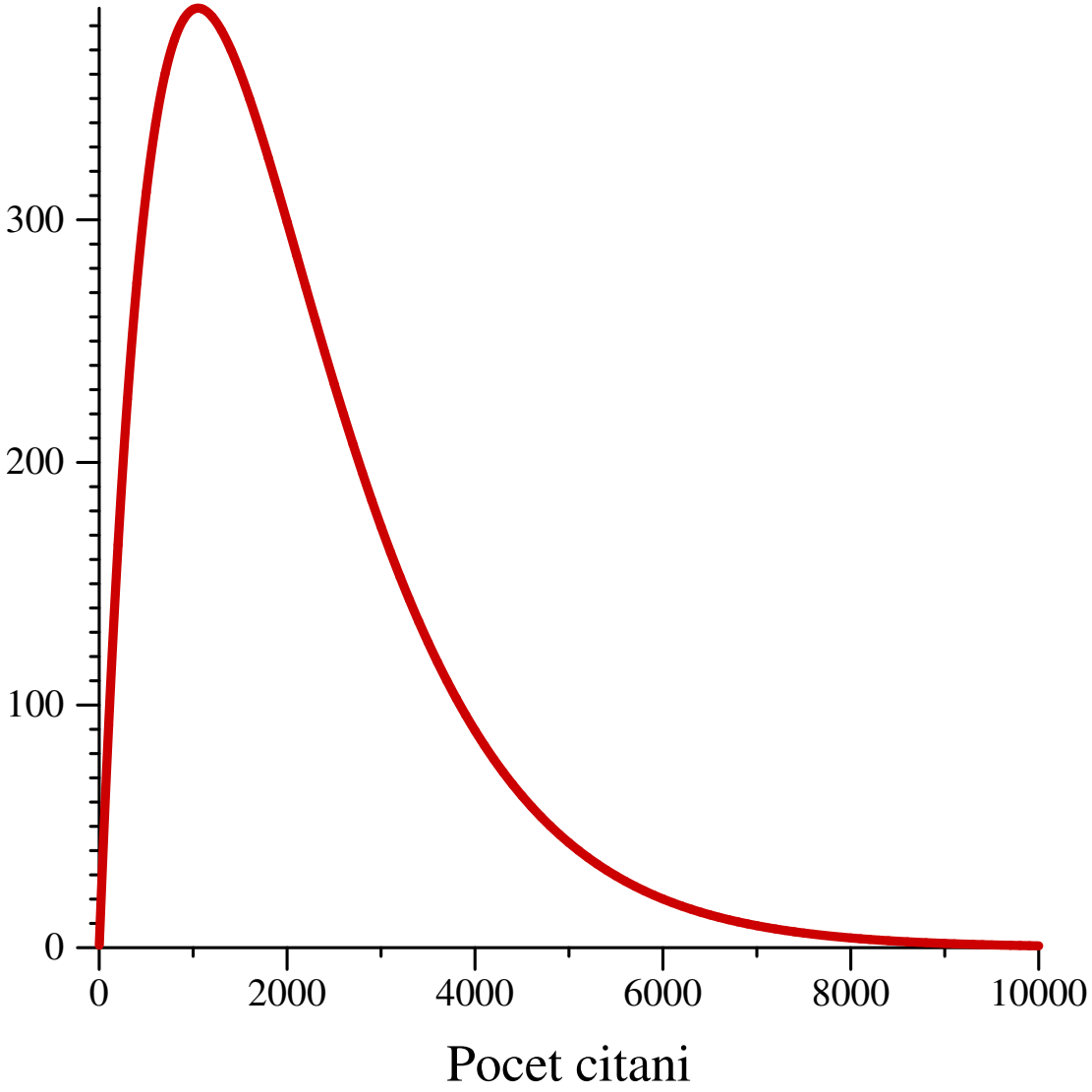
Pokrytie jednotlivých báz



Počet báz s určitým pokrytím



Predpokladaný počet kontigov od počtu čítaní



```
nepokr: 0 koncov: 0     nepokr: 0 koncov: 0     nepokr: 0 koncov
nepokr: 274 koncov: 2   nepokr: 282 koncov: 1   nepokr: 0 koncov
nepokr: 0 koncov: 0     nepokr: 0 koncov: 0     nepokr: 8 koncov
nepokr: 0 koncov: 0     nepokr: 12 koncov: 1    nepokr: 0 koncov
nepokr: 122 koncov: 1   nepokr: 135 koncov: 1   nepokr: 111 konc
nepokr: 13 koncov: 1    nepokr: 1 koncov: 1     nepokr: 56 konco
nepokr: 265 koncov: 1   nepokr: 0 koncov: 0     nepokr: 10 konco
nepokr: 0 koncov: 0     nepokr: 0 koncov: 0     nepokr: 130 konc
nepokr: 217 koncov: 1   nepokr: 3 koncov: 1     nepokr: 0 koncov
nepokr: 0 koncov: 0     nepokr: 0 koncov: 0     nepokr: 86 konco
nepokr: 139 koncov: 2   nepokr: 0 koncov: 0     nepokr: 0 koncov
nepokr: 76 koncov: 1    nepokr: 221 koncov: 1   nepokr: 26 konco
nepokr: 0 koncov: 0     nepokr: 1 koncov: 1     nepokr: 0 koncov
nepokr: 0 koncov: 0     nepokr: 0 koncov: 0     nepokr: 0 koncov
nepokr: 0 koncov: 0     nepokr: 0 koncov: 0     nepokr: 12 konco
nepokr: 103 koncov: 2   nepokr: 0 koncov: 0     nepokr: 71 konco
nepokr: 69 koncov: 1    nepokr: 0 koncov: 0
```

Úvod do dynamického programovania, proteomika

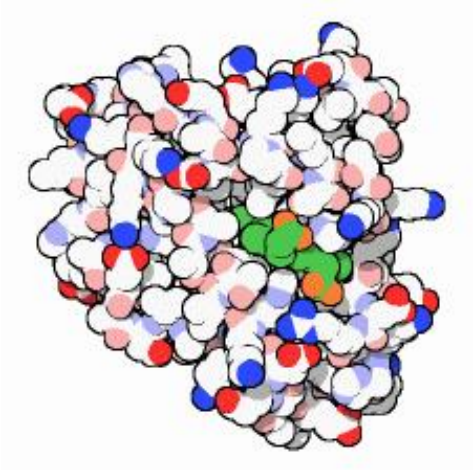
Askar Gafurov

7.10.2021

Proteomika

Proteín: sekvencia pozostáva z 20 rôznych aminokyselín

MGLSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASE
DLKKHGATVLTALGGILKKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKH
PGDFGADAQGAMNKALELFRKDMASNYKELGFQG



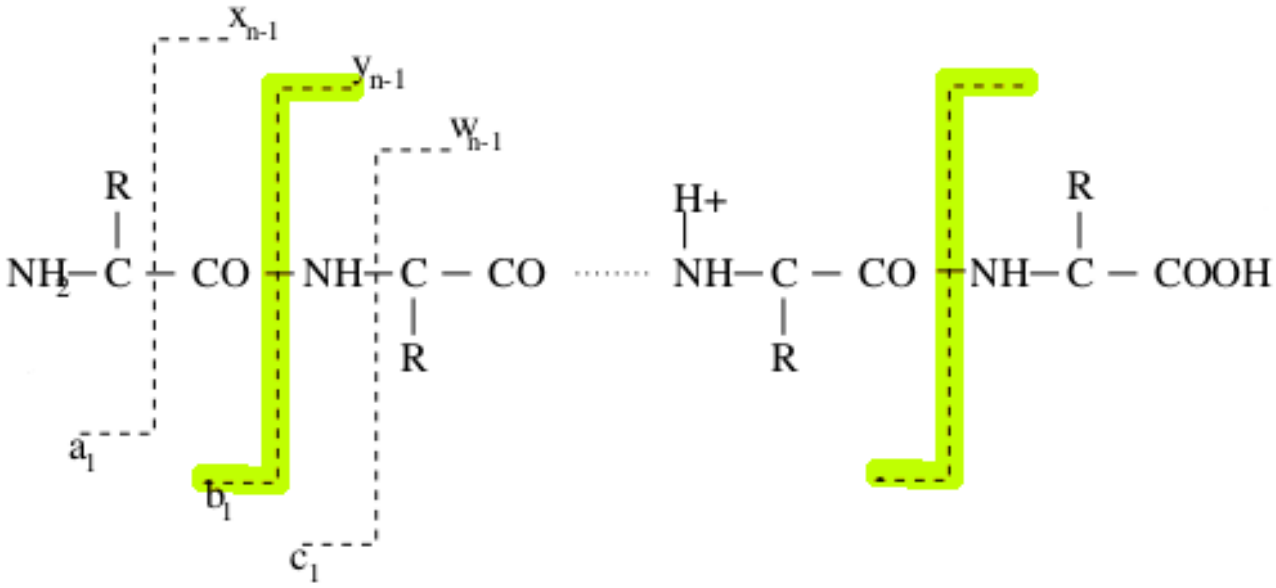
Z bunky sme izolovali určitý proteín, chceme zistiť jeho sekvenciu.

Hmotnostná spektrometria (mass spectometry)

- Meria pomer hmostnosť/náboj molekúl vo vzorke
- Používa sa na identifikáciu proteínov
- Proteín nasekáme enzýmom trypsín (seká na `[KR]{P}`) na peptidy
- Meriame hmostnosť kúskov, porovnáme s databázou proteínov.
- Tandemová hmotnostná spektrometria (MS/MS) ďalej fragmentuje každý kúsok a dosiahne podrobnejšie spektrum, ktoré obsahuje viac informácie
- V niektorých prípadoch tak vieme sekvenciu proteínu určiť priamo z MS/MS, bez databázy proteínov

Tandemová hmotnostná spektrometria MS/MS

Štiepenie peptidu na prefixy a sufixy



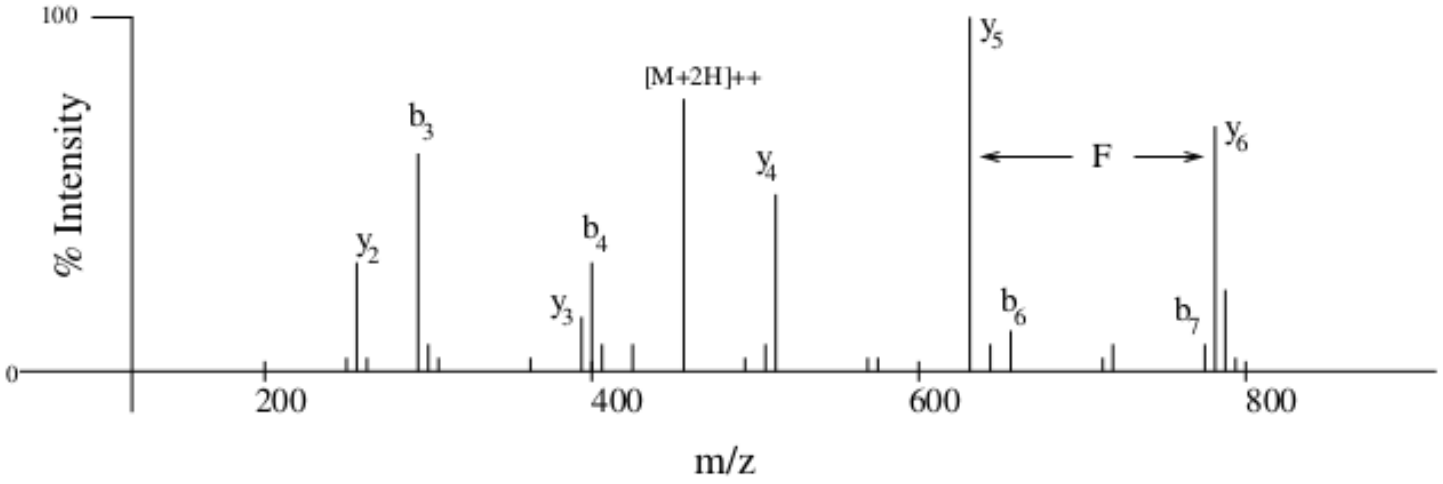
zdroj: Bafna and Reinert

b-ióny: prefixy

y-ióny: sufixy

Tandemová hmotnostná spektrometria MS/MS

88	145	292	405	534	663	778	924	b-ions
S	G	F	L	E	E	D	K	
924	837	780	633	520	391	262	141	y-ions



zdroj: Bafna and Reinert

Sekvenovanie peptidov pomocou MS/MS

Vstup: celková hmotnosť peptidu M,

hmotnosti aminokyselín $a[1], \ldots, a[20]$ (celé čísla),

spektrum ako tabuľka $f[0], \ldots, f[M]$, ktorá hmotnosti určí skóre podľa signálu v okolí príslušného bodu grafu

Označenie:

Nech $x = x_1 \ldots x_k$ je postupnosť aminokyselín

Nech $m(x) = \sum_{j=1}^{k} a[x_j]$ je hmotnosť x

Nech $\mathcal{M}_P(x) = \{m(x_1 \ldots x_j) \mid j = 1, \ldots, k\}$ sú hmotnosti prefixov x

Nech $\mathcal{M}_S(x) = \{m(x_j \ldots x_k) \mid j = 1, \ldots, k\}$ sú hmotnosti sufixov x

Problém 1: uvažujeme iba b-ióny (prefixy)

Výstup: postupnosť aminokyselín x taká, že $m(x) = M$ a $\sum_{m \in \mathcal{M}_P(x)} f[m]$ je maximálna možná

Príklad

Uvažujme len 3 aminokyseliny X,Y,Z

$M = 23, a[X] = 4, a[Y] = 6, a[Z] = 7$

m	4	6	7	11	12	17	18	19
$f[m]$	1	1	1	1	1	1	1	1

Hmotnosti prefixov $\mathcal{M}_P(XZYY) =$
$\{m(), m(X), m(XZ), m(XZYY), m(XZYY)\} = \{0, 4, 11, 17, 23\}$

Hmotnosti sufixov $\mathcal{M}_S(XZYY) =$
$\{m(), m(Y), m(YY), m(ZYY), m(XZYY)\} = \{0, 6, 12, 19, 23\}$

Skóre XZYY: $\sum_{m \in \mathcal{M}_P(ZYXX)} f[m] = 0 + 1 + 1 + 1 + 0 = 3$

Skóre XZXXX: $\sum_{m \in \mathcal{M}_P(ZYZZZ)} f[m] =$
$f[0] + f[4] + f[11] + f[15] + f[19] + f[23] = 0 + 1 + 1 + 0 + 1 + 0 = 3$

Sekvenovanie peptidov pomocou MS/MS

Problém 2: uvažujeme prefixy aj sufixy, sčítame ich skóre

Výstup: postupnosť aminokyselín x taká, že $m(x) = M$ a $\sum_{m \in \mathcal{M}_P(x)} f[m] + \sum_{m \in \mathcal{M}_S(x)} f[m]$ je maximálna možná

Problém 3: uvažujeme prefixy aj sufixy, sčítame ich skóre, ale každú hmotnosť započítame najviac raz

Výstup: postupnosť aminokyselín x taká, že $m(x) = M$ a $\sum_{m \in \mathcal{M}_P(x) \cup \mathcal{M}_S(x)} f[m]$ je maximálna možná

Príklad

$M = 23, a[X] = 4, a[Y] = 6, a[Z] = 7$

m	4	6	7	11	12	17	18	19
$f[m]$	1	1	1	1	1	1	1	1

$\mathcal{M}_P(XZYY) = \{0, 4, 11, 17, 23\} \quad \mathcal{M}_S(XZYY) = \{0, 6, 12, 19, 23\}$

$\mathcal{M}_P(XZXXX) = \{0, 4, 11, 15, 19, 23\}$
$\mathcal{M}_S(XZXXX) = \{0, 4, 8, 12, 19, 23\}$

Problém 2: $\sum_{m \in \mathcal{M}_P(x)} f[m] + \sum_{m \in \mathcal{M}_S(x)} f[m]$

Skóre XZYY: $0 + 1 + 1 + 1 + 0 + 0 + 1 + 1 + 1 + 0 = 6$

Skóre XZXXX: $0 + 1 + 1 + 0 + 1 + 0 + 0 + 1 + 0 + 1 + 1 + 0 = 6$

Problém 3: $\sum_{m \in \mathcal{M}_P(x) \cup \mathcal{M}_S(x)} f[m]$

XZYY: $\{0, 4, 6, 11, 12, 17, 19, 23\}, 1 + 1 + 1 + 1 + 1 + 1 + 0 = 6$

XZXXX: $\{0, 4, 8, 11, 12, 15, 19, 23\}, 1 + 0 + 1 + 1 + 0 + 1 + 0 = 4$

Ekvivalencia problémov

Problém 2: maximalizujeme $\sum_{m \in \mathcal{M}_P(x)} f[m] + \sum_{m \in \mathcal{M}_S(x)} f[m]$

Iná formulácia: maximalizujeme $\sum_{m \in \mathcal{M}_p(x)} g[m]$

kde $g[m] = f[m] + f[M - m]$

Ekvivalencia problémov

Problém 3: maximalizujeme $\sum_{m \in \mathcal{M}_P(x) \cup \mathcal{M}_S(x)} f[m]$

Iná formulácia: maximalizujeme $\sum_{m \in \mathcal{M}_p(x) \cup \mathcal{M}_S(x), m \leq M/2} h[m]$

kde $h[m] = \begin{cases} f[m] + f[M-m] & \text{ak } m < M/2 \\ f[m] & \text{ak } m = M/2 \end{cases}$

Jadrá zarovnaní

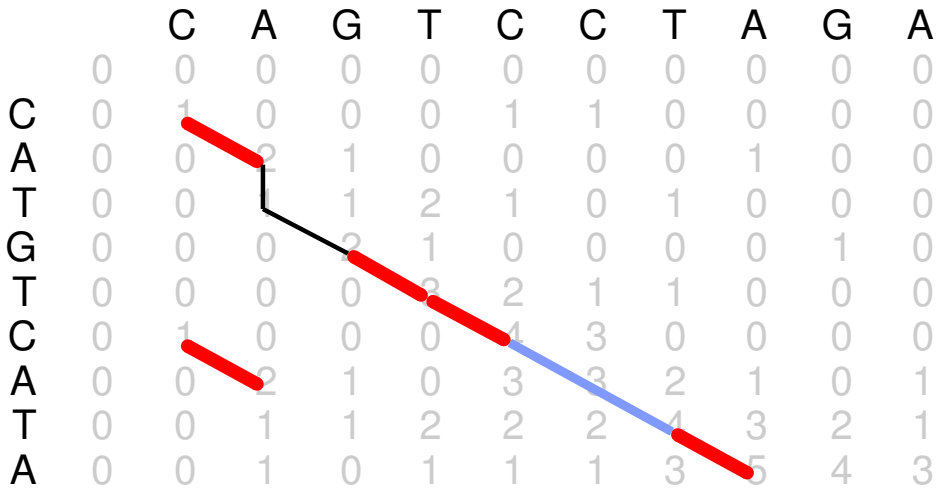
Broňa Brejová

20.10.2022

Opakovanie: Heuristické lokálne zarovnávanie, BLAST

Príklad: $k = 2$ (začíname z jadier dĺžky 2).

(V praxi sa používa $k = 10$ a viac.)



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

Senzitivita heuristického algoritmu

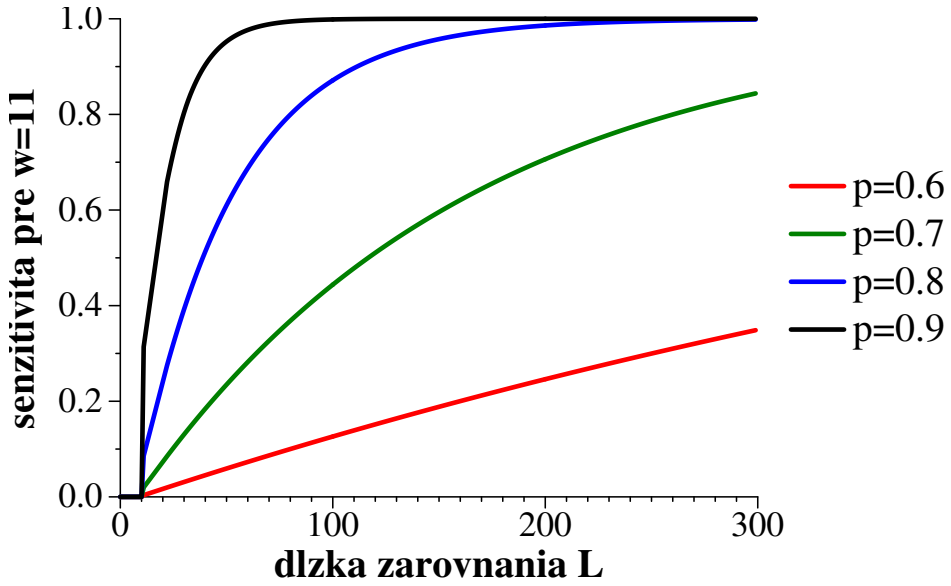
Odhad senzitivity:

Predpokladáme zarovnanie bez medzier, dĺžky L

Každá pozícia je zhoda s pravdepodobnosťou p

Senzitivita:

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



Senzitivita heuristického algoritmu

Predpokladáme zarovnanie bez medzier, dĺžky L

Každá pozícia je zhoda s pravdepodobnosťou p

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

Budeme počítať

$A[n] = 1 - f(n, p) = \Pr(\text{zarovnanie neobsahuje } k \text{ zhôd za sebou})$

Opakovanie: ako funguje hľadanie jadier

DB: ulož k-mery do hašovacej tabuľky Query: hľadaj v tabuľke

```
AGTGGCTGCCAGGCTGG
AGTGG,0
 GTGGC,1
  TGGCT,2
   GGCTG,3
    GCTGC,4
     CTGCC,5
      TGCCA,6
       GCCAG,7
        CCAGG,8
         CAGGC,9
          AGGCT,10
           GGCTG,11
            GCTGG,12
```

```
cGaGGCTGCCtGGtTGG
CGAGG
 GAGGC
  AGGCT <-
   GGCTG <-
    GCTGC <-
     CTGCC <-
      TGCCT
       GCCTG
        CCTGG
         CTGGT
          TGGTT
           GGTTG
            GTTGG
```

Šetrenie pamäťou: BLAT

$k = 5, s = 3$

```
AGTGGCTGCCAGGCTGG
AGTGG,0
    GGCTG,3
        TGCCA,6
            CAGGC,9
                GCTGG,12
```

```
cGaGGCTGCCtGGtTGG
CGAGG
 GAGGC
  AGGCT
   GGCTG <-
    GCTGC
     CTGCC
      TGCCT
       GCCTG
        CCTGG
         CTGGT
          TGGTT
           GGTTG
            GTTGG
```

Šetrenie pamäťou: minimizery

$k = 5, s = 3$

```
AGTGGCTGCCAGGCTGG
AGTGG,0
 GTGGC
  TGGCT
   GGCTG,3
    GCTGC,4
     CTGCC,5
      TGCCA
       GCCAG
        CCAGG,8
         CAGGC,9
          AGGCT,10
           GGCTG
            GCTGG
```

```
cGaGGCTGCCtGGtTGG
CGAGG
 GAGGC
  AGGCT*
   GGCTG
    GCTGC
     CTGCC* <--
      TGCCT
       GCCTG
        CCTGG*
         CTGGT*
          TGGTT
           GGTTG*
            GTTGG
```

BLAST vs BLAT vs minimizers

n: dĺžka DB, m: dĺžka query, krok s

Program	k-merov v slovníku	k-merov hľadáme	jadro zaručené pri
BLAST	n	m	k zhôd pri sebe
BLAT	n/s	m	$k+s-1$ zhôd pri sebe
minimizery	cca $2n/(s+1)$	cca $2m/(s+1)$	$k+s-1$ zhôd pri sebe

V počtoch k-merov sme zanedbali členy typu $-w+1$

Algoritmy pre HMM

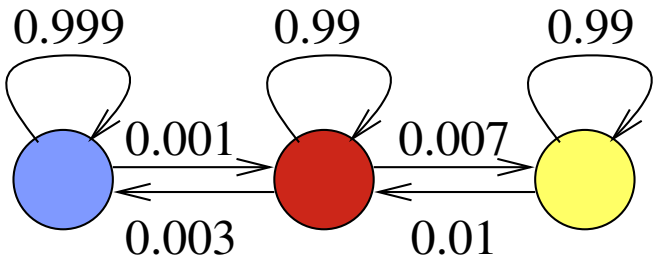
Askar Gafurov

7.11.2019

Opakovanie: HMM (skrytý Markovov model)

→ náhodná DNA sekvencia S, náhoc
(podobné na ozajstnú DNA)

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

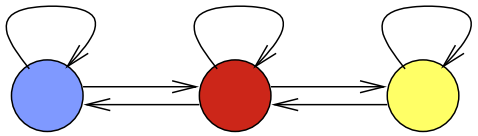


a:0.27	a:0.24	a:0.26
c:0.23	c:0.27	c:0.22
g:0.23	g:0.28	g:0.22
t:0.27	t:0.21	t:0.30

Predpokladajme, že model vždy začína v modrom stave.

$$\Pr(\text{aca}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 = 0.000017$$
$$\Pr(\text{aca}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 = 0.017$$

Parametre HMM (označenie)



Sekvencia $S_1, \ldots, S_n$

Anotácia $A_1, \ldots, A_n$

Parametre modelu:

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

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

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

a			
	0.99	0.007	0.003
	0.01	0.99	0
	0.001	0	0.999

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

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

Viterbiho algoritmus

Pre danú sekvenciu S nájde najpravdepodobnejšiu anotáciu
$A = \arg\max_A \Pr(A|S)$

Dynamické programovanie v čase $O(nm^2)$

Podproblém $V[i, u]$: pravdepodobnosť najpravdepodobnejšej cesty končiacej po i krokoch v stave u, pričom vygeneruje $S_1 S_2 \ldots S_i$

Rekurencia:

$V[1, u] = \pi_u \cdot e_{u,S_1}$
$V[i, u] = \max_w V[i-1, w] \cdot a_{w,u} \cdot e_{u,S_i}$

Algoritmus:

Inicializuj $V[1, *]$
for $i = 2 \ldots n$ (n=dĺžka reťazca)
    for $u = 1 \ldots m$ (m =počet stavov)
        vypočítaj $V[i, u]$
Maximálne $V[n, j]$ je pravdepodobnosť najpravdepodobnejšej cesty

Dopredný algoritmus

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S

$\Pr(S) = \sum_A Pr(A, S)$

Podproblém $F[i, u]$: pravdepodobnosť, že po i krokoch vygenerujeme $S_1, S_2, \ldots S_i$ a dostaneme sa do stavu u.

$F[i, u] = \Pr(A_i = u \wedge S_1, S_2, \ldots, S_i) =$
$\sum_{A_1, A_2, \ldots, A_i = u} \Pr(A_1, A_2, ..., A_i \wedge S_1, S_2, ..., S_i)$

Rekurencia:

$F[1, u] = \pi_u \cdot e_{u,S_1}$
$F[i, u] = \sum_v F[i-1, v] \cdot a_{v,u} \cdot e_{u,S_i}$

Celková pravdepodobnosť $\Pr(S) = \sum_u F[n, u]$

Spätný algoritmus

Obdoba dopredného algoritmu

Dopredný algoritmus: $F[i,u] = \Pr(A_i = u \wedge S_1, \ldots, S_i)$

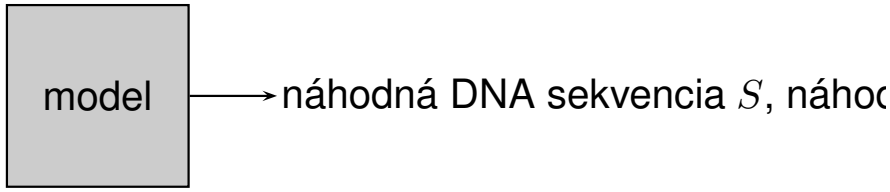
$F[1,u] = \pi_u \cdot e_{u,S_1}$

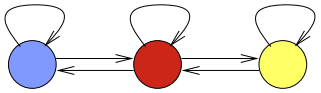
$F[i,u] = \sum_v F[i-1,v] \cdot a_{v,u} \cdot e_{u,S_i}$

$\Pr(S) = \sum_u F[n,u]$

Spätný algoritmus: $B[i,u] = \Pr(S_{i+1} \ldots, S_n | A_i = u)$

Hľadanie génov s HMM



- **Určenie stavov a prechodov v modeli:** ručne, na základe poznatkov o štruktúre génu. 
- **Trénovanie parametrov:** pravdepodobnosti určíme na základe sekvencií so známymi génmi (**trénovacia množina**).
  Model zostavíme tak, aby páry (S, A) s vlastnosťami podobnými skutočným génom mali veľkú pravdepodobnosť $\Pr(S, A)$
- **Použitie:** pre novú sekvenciu S nájdi najpravdepodobnejšiu anotáciu $A = \arg\max_A \Pr(A|S)$ Viterbiho algoritmom

Trénovanie HMM

- Stavový priestor + povolené prechody väčšinou ručne
- Parametre (pravdepodobnosti prechodu, emisie a počiatočné) automaticky z trénovacích sekvencií
- Čím zložitejší model a viac parametrov máme, tým potrebujeme viac trénovacích dát, aby nedošlo k preučeniu, t.j. k situácii, keď model dobre zodpovedá nejakým zvláštnostiam trénovacích dát, nie však ďalším dátam.
- Presnosť modelu testujeme na zvláštnych testovacích dátach, ktoré sme nepoužili na trénovanie.

Trénovanie HMM z anotovaných sekvencií

Vstup: topológia modelu a niekoľko trénovacích párov $S^{(i)}, A^{(i)}$

Cieľ: nastaviť π_u, $e_{u,x}$, $a_{u,v}$ tak, aby $\prod_i \Pr(S^{(i)}, A^{(i)})$ bola čo najväčšia

Dosiahneme jednoduchým počítaním frekvencií

Napr. $a_{u,v}$: nájdeme všetky výskyty stavu u a zistíme, ako často za nimi ide stav v

Trénovanie HMM z neanotovaných sekvencií

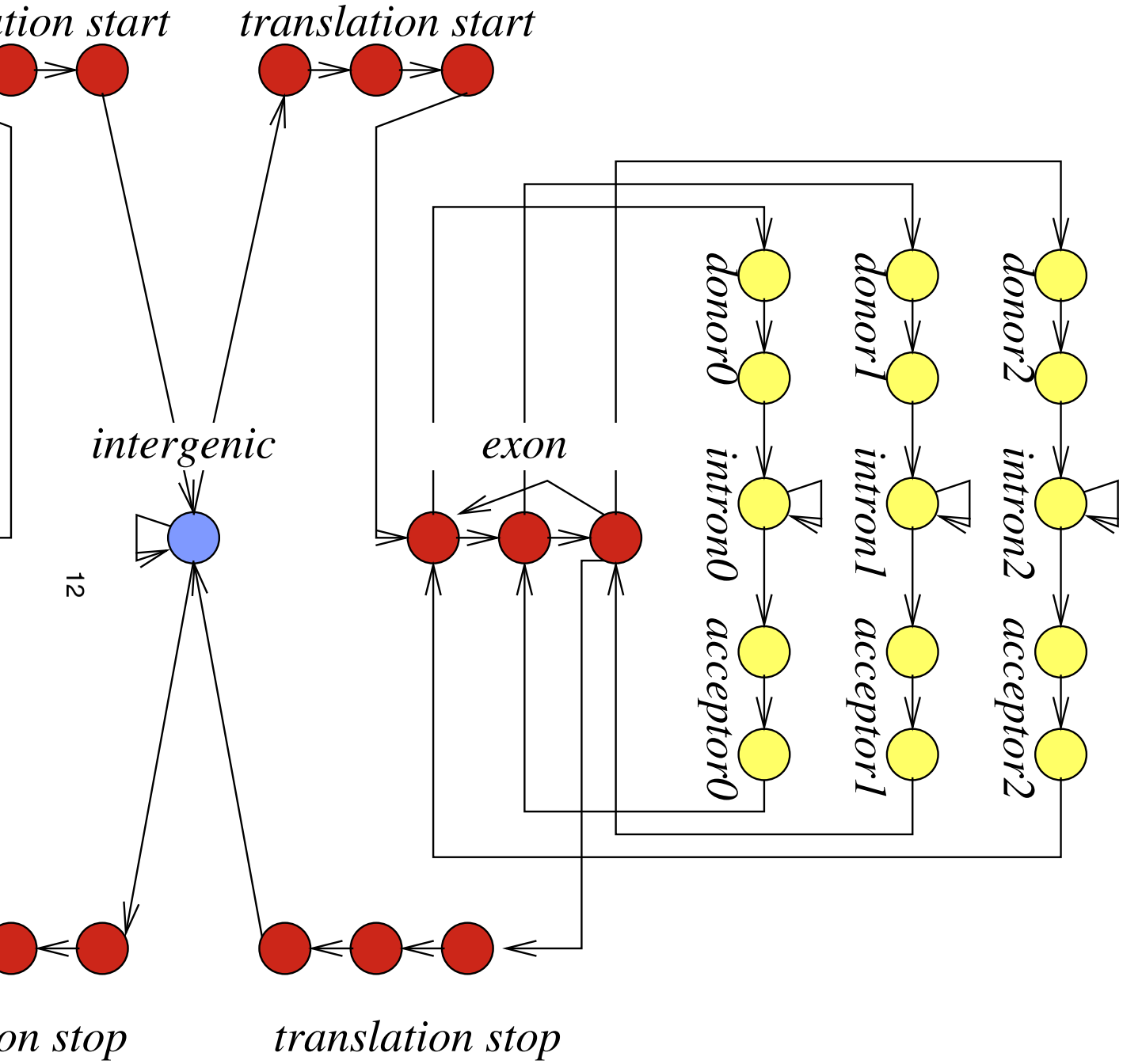
Vstup: topológia modelu a niekoľko trénovacích sekvencií $S^{(i)}$
anotácie $A^{(i)}$ nepoznáme

Cieľ: nastaviť π_u, $e_{u,x}$, $a_{u,v}$ tak, aby $\prod_i \Pr(S^{(i)})$ bola čo najväčšia

Používajú sa heuristické iteratívne algoritmy, napr. Baum-Welchov, ktorý je verziou všeobecnejšieho algoritmu EM (expectation maximization).

Tvorba stavového priestoru modelu

Príklad HMM na hľadanie génov



Forward strand

Substitution Models

Tomáš Vinař

November 4, 2021

Substitution models, notation

$P(b|a,t)$: probability that if we start with symbol a, after time t we will see symbol b

Transition probability matrix:

$$S(t) = \begin{pmatrix} P(A|A,t) & P(C|A,t) & P(G|A,t) & P(T|A,t) \\ P(A|C,t) & P(C|C,t) & P(G|C,t) & P(T|C,t) \\ P(A|G,t) & P(C|G,t) & P(G|G,t) & P(T|G,t) \\ P(A|T,t) & P(C|T,t) & P(G|T,t) & P(T|T,t) \end{pmatrix}$$

Substitution models, basic properties

- $S(0) = I$
- $\lim_{t\to\infty} S(t) = \begin{pmatrix} \pi_A & \pi_C & \pi_G & \pi_T \\ \pi_A & \pi_C & \pi_G & \pi_T \\ \pi_A & \pi_C & \pi_G & \pi_T \\ \pi_A & \pi_C & \pi_G & \pi_T \end{pmatrix}$

  Distribution π is called stationary (equilibrium)
- $S(t_1 + t_2) = S(t_1)S(t_2)$ (multiplicativity)
- Jukes-Cantor model should also satisfy

$$S(t) = \begin{pmatrix} 1-3s(t) & s(t) & s(t) & s(t) \\ s(t) & 1-3s(t) & s(t) & s(t) \\ s(t) & s(t) & 1-3s(t) & s(t) \\ s(t) & s(t) & s(t) & 1-3s(t) \end{pmatrix}$$

$$S(t) = \begin{pmatrix} 1-3s(t) & s(t) & s(t) & s(t) \\ s(t) & 1-3s(t) & s(t) & s(t) \\ s(t) & s(t) & 1-3s(t) & s(t) \\ s(t) & s(t) & s(t) & 1-3s(t) \end{pmatrix}$$

$$S(2t) = S(t)^2 =$$

$$= \begin{pmatrix} 1-6s(t)+12s(t)^2 & 2s(t)-4s(t)^2 & 2s(t)-4s(t)^2 & 2s(t)-4s(t)^2 \\ 2s(t)-4s(t)^2 & 1-6s(t)+12s(t)^2 & 2s(t)-4s(t)^2 & 2s(t)-4s(t)^2 \\ 2s(t)-4s(t)^2 & 2s(t)-4s(t)^2 & 1-6s(t)+12s(t)^2 & 2s(t)-4s(t)^2 \\ 2s(t)-4s(t)^2 & 2s(t)-4s(t)^2 & 2s(t)-4s(t)^2 & 1-6s(t)+12s(t)^2 \end{pmatrix}$$

$$\approx \begin{pmatrix} 1-6s(t) & 2s(t) & 2s(t) & 2s(t) \\ 2s(t) & 1-6s(t) & 2s(t) & 2s(t) \\ 2s(t) & 2s(t) & 1-6s(t) & 2s(t) \\ 2s(t) & 2s(t) & 2s(t) & 1-6s(t) \end{pmatrix}$$

for $t \to 0$

Substitution rate matrix (matica rýchlostí, matica intenzít)

- Substitution rate matrix for Jukes-Cantor model:

$$R = \begin{pmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{pmatrix}$$

- For very small t we have $S(t) \approx I + Rt$
- Rate α is the probablity of a change per unit of time for very small t, or derivative of $s(t)$ with respect to t at $t = 0$
- Solving the differential equation for the Jukes-Cantor model we get $s(t) = (1 - e^{-4\alpha t})/4$

Jukes-Cantor model

$$S(t) = \begin{pmatrix} (1+3e^{-4\alpha t})/4 & (1-e^{-4\alpha t})/4 & (1-e^{-4\alpha t})/4 & (1-e^{-4\alpha t})/4 \\ (1-e^{-4\alpha t})/4 & (1+3e^{-4\alpha t})/4 & (1-e^{-4\alpha t})/4 & (1-e^{-4\alpha t})/4 \\ (1-e^{-4\alpha t})/4 & (1-e^{-4\alpha t})/4 & (1+3e^{-4\alpha t})/4 & (1-e^{-4\alpha t})/4 \\ (1-e^{-4\alpha t})/4 & (1-e^{-4\alpha t})/4 & (1-e^{-4\alpha t})/4 & (1+3e^{-4\alpha t})/4 \end{pmatrix}$$

The rate matrix is typically normalized so that there is on average one substitution per unit of time, here $\alpha = 1/3$

Jukes-Cantor model, summary

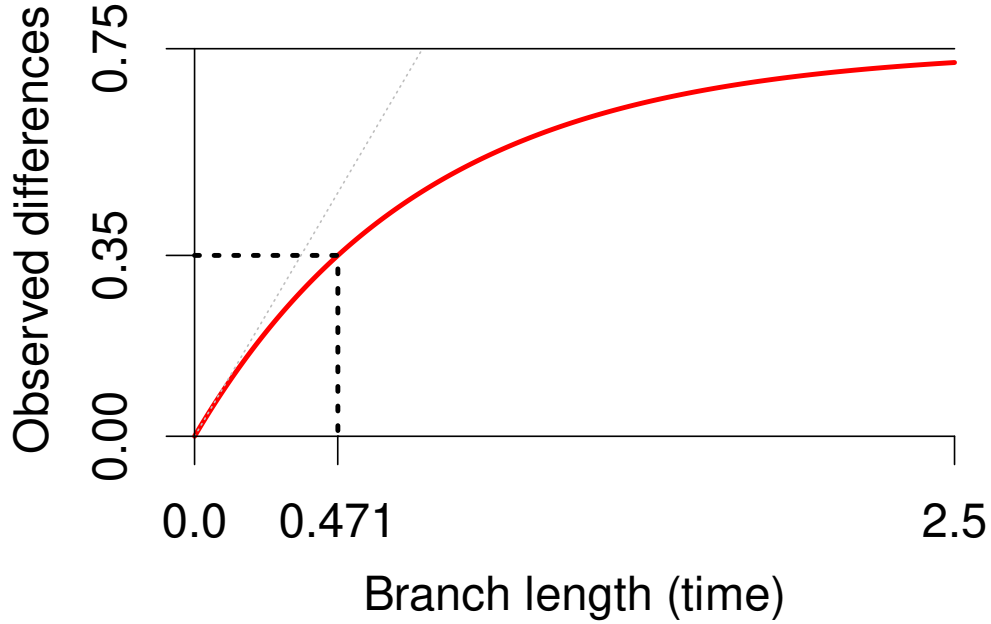
- $S(t)$: matrix 4×4, where $S(t)_{a,b} = P(b|a,t)$ is the probability that if we start with base a, after time t we have base b.
- Jukes-Cantor model assumes that $P(b|a,t)$ is the same for all $a \neq b$
- For a given time t, off-diagonal elements are $s(t)$, diagonal $1 - 3s(t)$
- Rate matrix R: for J-C off-diagonal α, diagonal -3α
- For very small t we have $S(t) \approx I - Rt$
- Rate α is the probablity of a change per unit of time for very small t, or derivative of $s(t)$ with respect to t for $t = 0$
- Solving the differential equation for the Jukes-Cantor model, we get $s(t) = (1 - e^{-4\alpha t})/4$
- The rate matrix is typically normalized so that there is on average one substitution per unit of time, that is, $\alpha = 1/3$

Correction of evolutionary distances

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

The expected number of observed changes per base in time t:

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



Correction of observed distances

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

More complex models

- General rate matrix R

$$R = \begin{pmatrix} \cdot & \mu_{AC} & \mu_{AG} & \mu_{AT} \\ \mu_{CA} & \cdot & \mu_{CG} & \mu_{CT} \\ \mu_{GA} & \mu_{GC} & \cdot & \mu_{GT} \\ \mu_{TA} & \mu_{TC} & \mu_{TG} & \cdot \end{pmatrix}$$

- μ_{xy} is the rate at which base x changes to a different base y
- Namely, $\mu_{xy} = \lim_{t \to 0} \frac{\Pr(y \mid x,t)}{t}$
- The diagonal is added so that the sum of each row is 0
- There are models with a smaller number of parameters (compromise between J-C and an arbitrary matrix)

Kimura model

- A and G are purines, C and T pyrimidines
- Purines more often change to other purines
  and pyrimidines to pyrimidines
- Transition: change within group $A \Leftrightarrow G$, $C \Leftrightarrow T$,
  Transversion: change to a different group $\{A, G\} \Leftrightarrow \{C, T\}$
- Two parameters: rate of transitions α, rate of transversions β
- $$R = \begin{pmatrix} -2\beta - \alpha & \beta & \alpha & \beta \\ \beta & -2\beta - \alpha & \beta & \alpha \\ \alpha & \beta & -2\beta - \alpha & \beta \\ \beta & \alpha & \beta & -2\beta - \alpha \end{pmatrix}$$

HKY model (Hasegawa, Kishino, Yano)

- Extension of Kimura model, which allows different probabilities of A, C, G, T in the equilibrium
- If we set time to infinity, original base is not important, base frequencies stabilize in an equilibrium.
- Jukes-Cantor has probability of each base in the equilibrium 1/4.
- In HKY the equilibrium frequencies $\pi_A, \pi_C, \pi_G, \pi_T$ are parameters (summing to 1)
- Parameter κ: transition / transversion ratio (α/β)
- Rate matrix: $\mu_{x,y} = \begin{cases} \kappa\pi_y & \text{if mutation from } x \text{ to } y \text{ is transition} \\ \pi_y & \text{if mutation from } x \text{ to } y \text{ is transversion} \end{cases}$

From rate matrix R to transition probabilities $S(t)$

- J-C and some other models have explicit formulas for $S(t)$
- For more complex models, such formulas are not available
- In general, $S(t) = e^{Rt}$
- Exponential of a matrix A is defined as $e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$
- If R is diagonalized $R = UDU^{-1}$, where D is a diagonal matrix, then $e^{Rt} = Ue^{Dt}U^{-1}$ and the exponential function is applied to the diagnal elements of D
- Diagonalization always exists for symmetric matrices R
  (the diagonal contains eigenvalues)

Algoritmy pre HMM

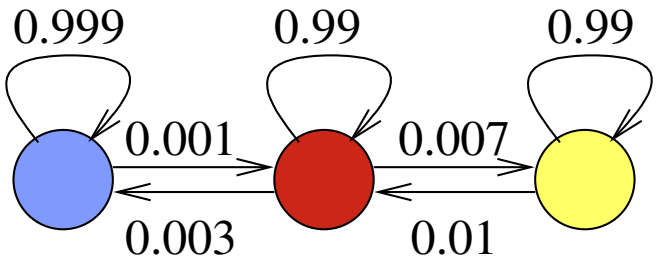
Askar Gafurov

7.11.2019

Opakovanie: HMM (skrytý Markovov model)

→ náhodná DNA sekvencia S, náhoc
(podobné na ozajstnú DNA)

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

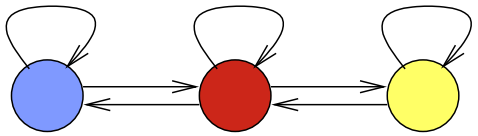


a:0.27	a:0.24	a:0.26
c:0.23	c:0.27	c:0.22
g:0.23	g:0.28	g:0.22
t:0.27	t:0.21	t:0.30

Predpokladajme, že model vždy začína v modrom stave.

$$\Pr(\text{aca}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 = 0.000017$$
$$\Pr(\text{aca}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 = 0.017$$

Parametre HMM (označenie)



Sekvencia $S_1, \ldots, S_n$

Anotácia $A_1, \ldots, A_n$

Parametre modelu:

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

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

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

a			
	0.99	0.007	0.003
	0.01	0.99	0
	0.001	0	0.999

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

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

Viterbiho algoritmus

Pre danú sekvenciu S nájde najpravdepodobnejšiu anotáciu
$A = \arg\max_A \Pr(A|S)$

Dynamické programovanie v čase $O(nm^2)$

Podproblém $V[i, u]$: pravdepodobnosť najpravdepodobnejšej cesty končiacej po i krokoch v stave u, pričom vygeneruje $S_1 S_2 \ldots S_i$

Rekurencia:

$V[1, u] = \pi_u \cdot e_{u,S_1}$
$V[i, u] = \max_w V[i-1, w] \cdot a_{w,u} \cdot e_{u,S_i}$

Algoritmus:

Inicializuj $V[1, *]$
for $i = 2 \ldots n$ (n=dĺžka reťazca)
    for $u = 1 \ldots m$ (m =počet stavov)
        vypočítaj $V[i, u]$
Maximálne $V[n, j]$ je pravdepodobnosť najpravdepodobnejšej cesty

Dopredný algoritmus

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S

$\Pr(S) = \sum_A Pr(A, S)$

Podproblém $F[i, u]$: pravdepodobnosť, že po i krokoch vygenerujeme $S_1, S_2, \ldots S_i$ a dostaneme sa do stavu u.

$F[i, u] = \Pr(A_i = u \wedge S_1, S_2, \ldots, S_i) =$
$\sum_{A_1, A_2, \ldots, A_i = u} \Pr(A_1, A_2, ..., A_i \wedge S_1, S_2, ..., S_i)$

Rekurencia:

$F[1, u] = \pi_u \cdot e_{u,S_1}$
$F[i, u] = \sum_v F[i-1, v] \cdot a_{v,u} \cdot e_{u,S_i}$

Celková pravdepodobnosť $\Pr(S) = \sum_u F[n, u]$

Spätný algoritmus

Obdoba dopredného algoritmu

Dopredný algoritmus: $F[i,u] = \Pr(A_i = u \wedge S_1, \ldots, S_i)$

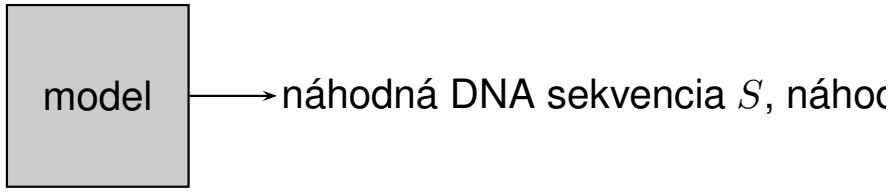
$F[1,u] = \pi_u \cdot e_{u,S_1}$

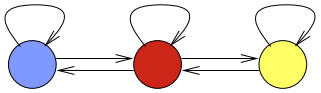
$F[i,u] = \sum_v F[i-1,v] \cdot a_{v,u} \cdot e_{u,S_i}$

$\Pr(S) = \sum_u F[n,u]$

Spätný algoritmus: $B[i,u] = \Pr(S_{i+1} \ldots, S_n | A_i = u)$

Hľadanie génov s HMM



- **Určenie stavov a prechodov v modeli:** ručne, na základe poznatkov o štruktúre génu. 
- **Trénovanie parametrov:** pravdepodobnosti určíme na základe sekvencií so známymi génmi (**trénovacia množina**).
  Model zostavíme tak, aby páry (S, A) s vlastnosťami podobnými skutočným génom mali veľkú pravdepodobnosť $\Pr(S, A)$
- **Použitie:** pre novú sekvenciu S nájdi najpravdepodobnejšiu anotáciu $A = \arg\max_A \Pr(A|S)$ Viterbiho algoritmom

Trénovanie HMM

- Stavový priestor + povolené prechody väčšinou ručne
- Parametre (pravdepodobnosti prechodu, emisie a počiatočné) automaticky z trénovacích sekvencií
- Čím zložitejší model a viac parametrov máme, tým potrebujeme viac trénovacích dát, aby nedošlo k preučeniu, t.j. k situácii, keď model dobre zodpovedá nejakým zvláštnostiam trénovacích dát, nie však ďalším dátam.
- Presnosť modelu testujeme na zvláštnych testovacích dátach, ktoré sme nepoužili na trénovanie.

Trénovanie HMM z anotovaných sekvencií

Vstup: topológia modelu a niekoľko trénovacích párov $S^{(i)}, A^{(i)}$

Cieľ: nastaviť π_u, $e_{u,x}$, $a_{u,v}$ tak, aby $\prod_i \Pr(S^{(i)}, A^{(i)})$ bola čo najväčšia

Dosiahneme jednoduchým počítaním frekvencií

Napr. $a_{u,v}$: nájdeme všetky výskyty stavu u a zistíme, ako často za nimi ide stav v

Trénovanie HMM z neanotovaných sekvencií

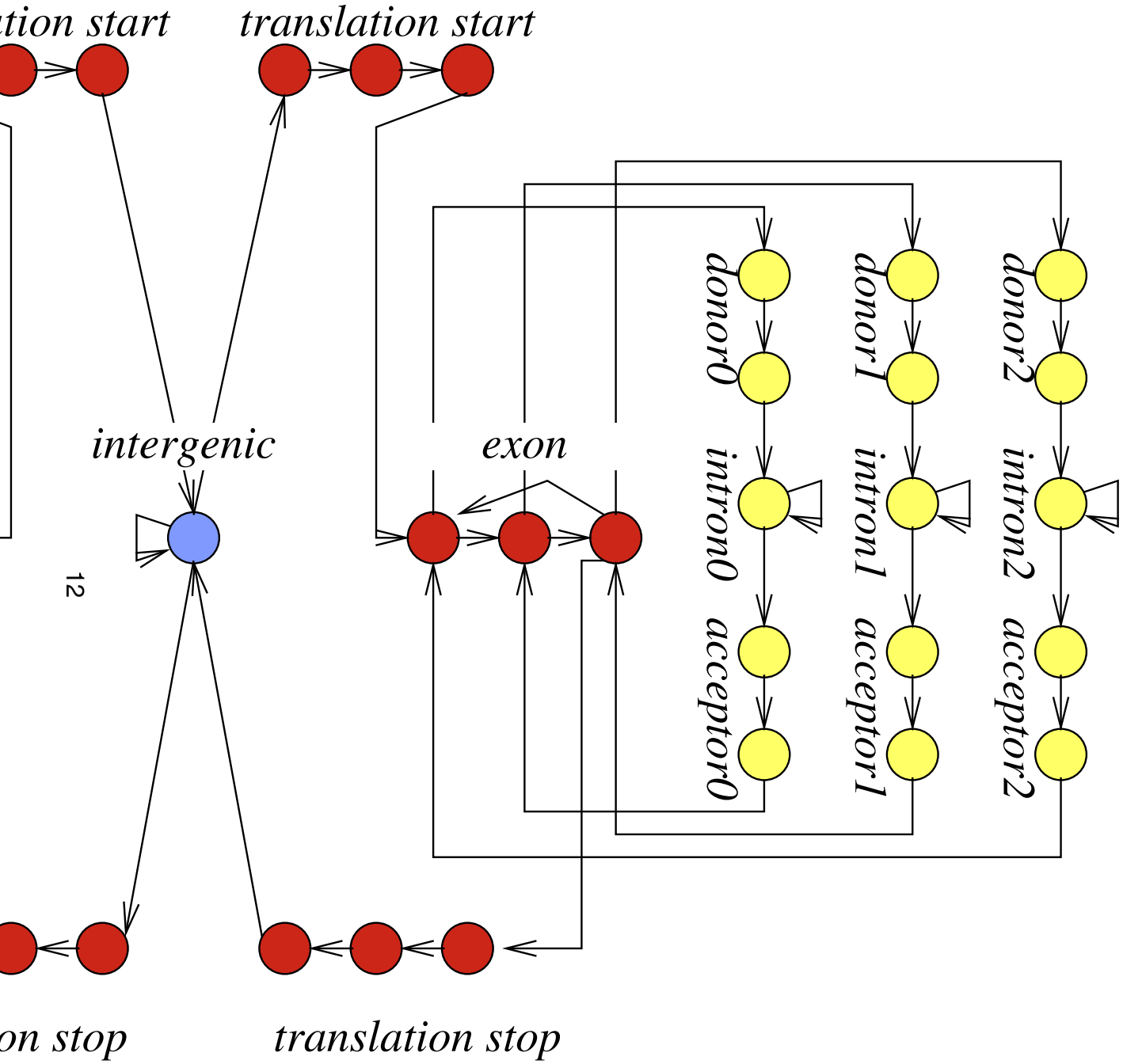
Vstup: topológia modelu a niekoľko trénovacích sekvencií $S^{(i)}$
anotácie $A^{(i)}$ nepoznáme

Cieľ: nastaviť π_u, $e_{u,x}$, $a_{u,v}$ tak, aby $\prod_i \Pr(S^{(i)})$ bola čo najväčšia

Používajú sa heuristické iteratívne algoritmy, napr. Baum-Welchov, ktorý je verziou všeobecnejšieho algoritmu EM (expectation maximization).

Tvorba stavového priestoru modelu

Príklad HMM na hľadanie génov



Forward strand

Course Summary

Broňa Brejová

December 16, 2021

Probabilistic models

- Hidden Markov models (gene finding, phylogenetic HMMs for conserved elements, profile HMMs for protein families)
- Phylogenetic trees and substitution models
- Stochastic context-free grammars
- Gibbs sampling
- Maximum likelihood method
- Expectation maximization (EM)

Statistical methods

- Statistical significance, E-value, P-value
- Positive selection test
- Linkage disequilibrium, association mapping

Practice in dynamic programming

- Sequence alignment
  (global, local, affine gaps, saving memory)
- Hidden Markov models (Viterbi and forward algorithms)
- Computation on trees
  (parsimony, Felsenstein algorithm for likelihood)
- Mass spectrometry (MS/MS)
- Secondary RNA structure

Other

- Integer linear programming
- deBruijn graphs
- Clustering and classification

How to model real-life problems

- Consider what data are avilable, what are relevant questions
- Formulate as a computer-science problem (e.g. score optimization)
- Probabilistic models often lead to a systematic choice of a scoring scheme
- The resulting problem often NP hard
  – Heuristics, approximation algorithms
  – ILP and other techniques for exact solutions
  – Can we change problem formulation?
- Testing: are computation results relevant in a given domain?
  (is our formulation sufficiently realistic?)

Ďalšie predmety

- **Strojové učenie** 2-INF-150, Vinař/Boža (ZS, 4P, 6kr)
- **Vybrané partie z dátových štruktúr** 2-INF-237, Kováč (ZS, 4P, 6kr)
- **Seminár z bioinformatiky (1)-(4)** 2-AIN-50[56],25[12] (oba semestre, 2S, 2kr)
- **Manažment dát** 1-DAV-202, Brejová, Vinař, Boža (LS, 1P/2C, 4kr)
- **Genomika** 2-INF-269, Nosek a kol. (LS, 2P/1C, 4kr)
- **Výzvy súčasnej bioinformatiky** 1-BIN-105, Brejová, Vinař (LS, 2S, 2kr)
- http://compbio.fmph.uniba.sk/vyuka/

Integer Linear Programming

Tomáš Vinař

December 16, 2021

Practical programs for NP-hard problems

They always find the optimal solution, often in reasonable time,
but on some inputs very long runtimes

- ILP: CPLEX, Gurobi (commercial), SCIP (non-commercial)
- SAT: Minisat, Lingeling, glucose, CryptoMiniSat, painless
- TSP: Concorde

Other NP-complete problems can be transformed to one of these problems

ILP: Integer linear programming

Linear programming:

real-valued variables $x_1, \ldots, x_n$

minimize $\sum_i a_i x_i$ for given weights $a_1, \ldots, a_n$

under constraints of the form $\sum_i b_i x_i \leq c$

LP can be solved in polynomial time

Integer linear programming:

Add a constraint that some variables are integers or binary

NP-hard problem

Expressing known NP-hard problems as ILP

Knapsack

Given n items with weights $w_1 \ldots w_n$ and costs $c_1 \ldots c_n$.

Choose a subset so that overall weight is at most T and the overall cost is highest possible?

Expressing known NP-hard problems as ILP

Set cover

We have n subsets $S_1 \ldots, S_n$ of a set $U = \{1 \ldots m\}$.

Choose the smallest number of the input subsets so that their union is the whole set U.

Protein threading

Protein A has a known sequence and structure, protein B only sequence.

Align A and B so that if two amino acids are close in A, their equivalents in B should be “compatible”.

Choose “cores” in A which should remain conserved without insertions, deletions and in the same order

Cores are separated by “loops”, whose length can arbitrarily change and whose alignements will not be scored

Protein threading, problem formulation

Input: sequence $B = b_1 \ldots b_n$,

lengths of m cores $c_1 \ldots .c_m$,

scoring tables

– E_{ij}: how well $b_j \ldots b_{j+c_i-1}$ agrees with sequence of core i,

– $E_{ijk\ell}$: how well would cores i and k interact, if they start at pos. j, ℓ.

Task: choose starts of cores $x_1, x_2, \ldots, x_m$ so that

– they are in the correct order and without overlaps,

– they achieve maximum possible score

Note: we do not specify how to choose cores and scoring tables,

which is a modeling, not an algorithmic problem

Protein threading, ILP

Notation: sequence $B = b_1 \dots b_n$, lengths of m cores $c_1 \dots c_m$,

E_{ij}: how well $b_j \dots b_{j+c_i-1}$ agrees with sequence of core i,

$E_{ijk\ell}$: how well would cores i and k interact, if they start at pos. j, ℓ,

unknown starts of cores $x_1, \dots, x_m$.

ILP formulation:

Protein threading, ILP

Notation: sequence $B = b_1 \dots b_n$, lengths of m cores $c_1 \dots c_m$,

E_{ij}: how well $b_j \dots b_{j+c_i-1}$ agrees with sequence of core i,

$E_{ijk\ell}$: how well would cores i and k interact, if they start at pos. j, ℓ,

unknown starts of cores $x_1, \dots, x_m$.

ILP formulation: