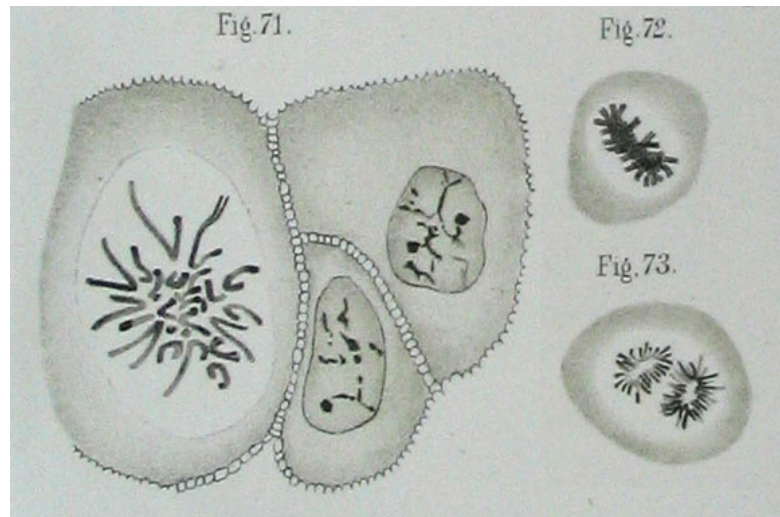


Biológia pre informatikov

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,

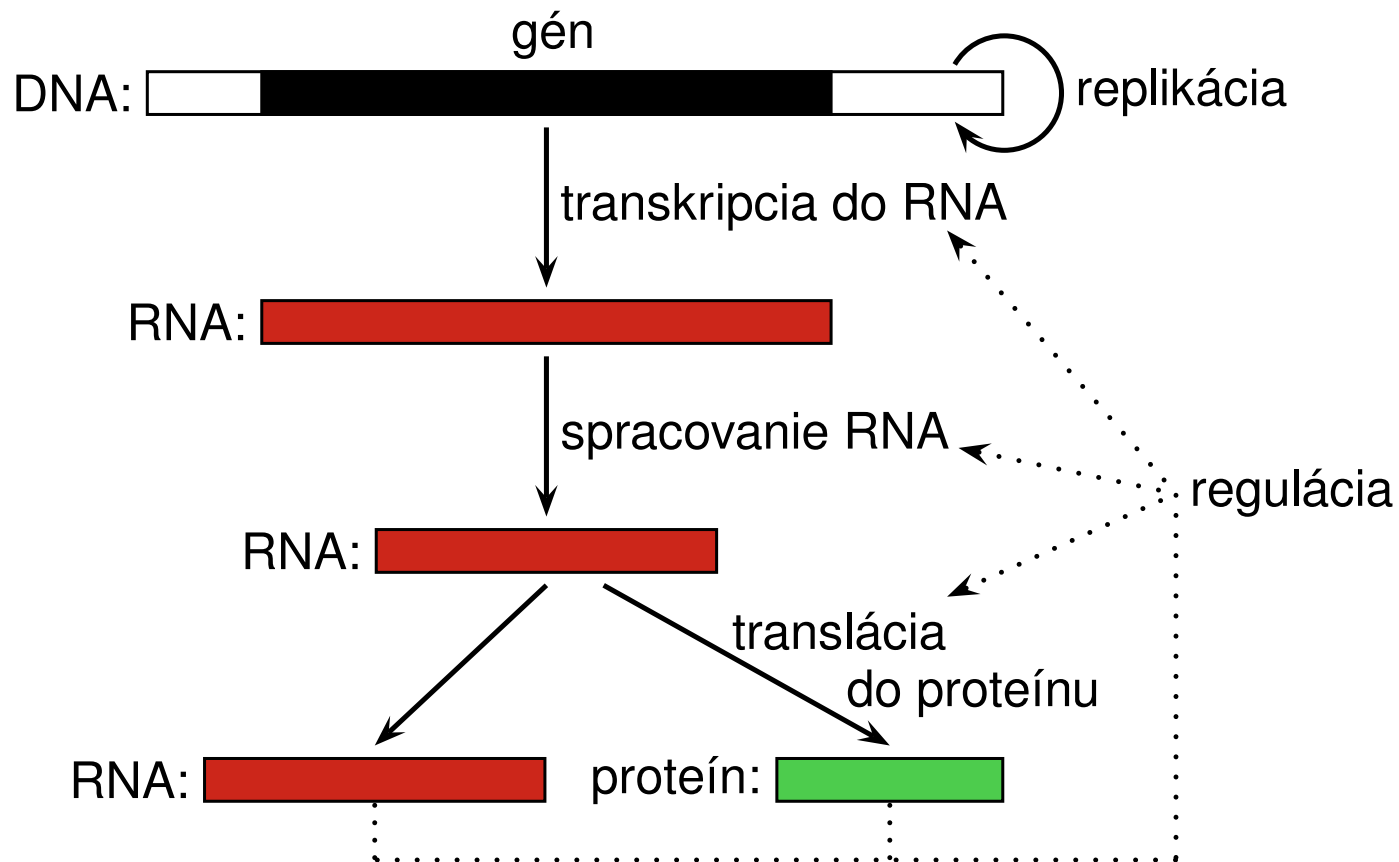
sú dôležité pre stavbu bunky a pohyb.

Reťazec aminokyselín (20 rôznych aminokyselín).

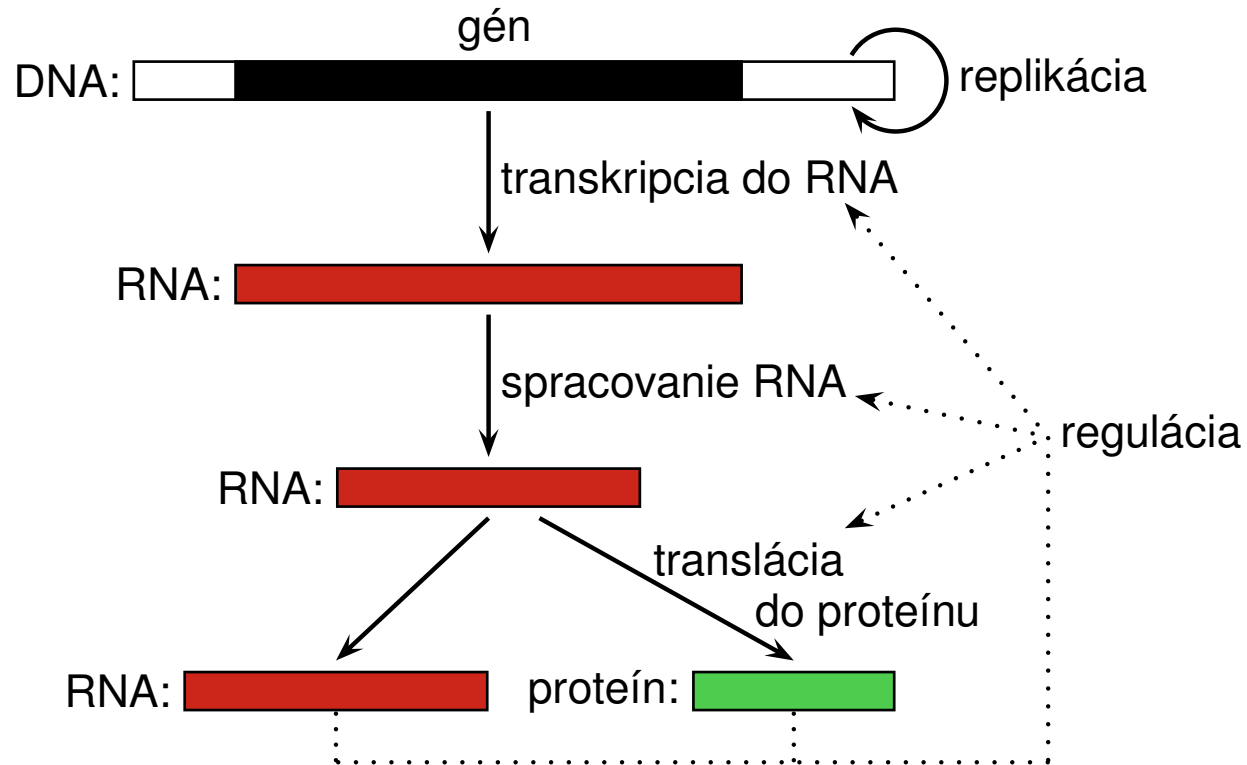
Aká informácia je uložená v DNA?

Gény: Predpisy na tvorbu proteínov a funkčných RNA molekúl.

Riadenie ich expresie: kedy a koľko sa má tvoriť.



Centrálna dogma (Francis Crick 1958,1970)



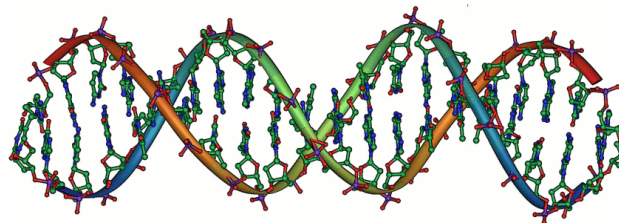
“The central dogma of molecular biology deals with the detailed residue-by-residue transfer of sequential information. It states that such information cannot be transferred back from protein to either protein or nucleic acid.”

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.

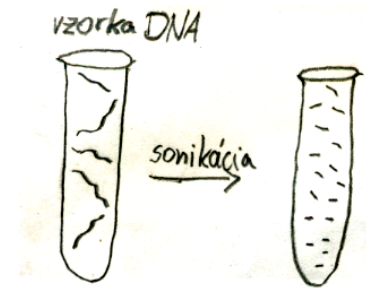
Pri delení bunky sa dvojvláknová DNA rozdelí a ku každému vláknu sa doplní komplement (DNA replikácia).

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 chromozónoch 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:

```
TCAGGATCGA
AGTCCTAGC TCAGGATCGA
          AGTCCTA
          TCAGGATCGA
          AGTCCTAGCT
          TCAGGATCGA
          AGTCCT
TCAGGATCGA TCAGGATCGA
AGTC TCAGGATCGA
          AGTCCTAG
          TCAGGATCGA
          AGTC
```

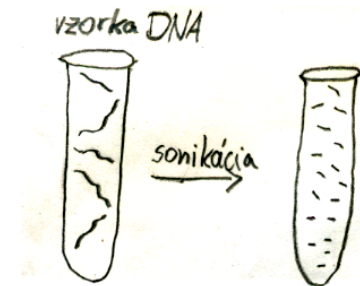
Na géli zoradíme podľa dĺžky:

```
AGTCCTAGCT
AGTCCTAGC
AGTCCTAG
AGTCCTA
AGTCCT
AGTCC
AGTC
AGTC
```

Odčítaním farieb dostaneme komplementárne vlákno: AGTCCTAGCT

Technológia: sekvenovanie DNA

- Postup na zisťovanie poradia báz v chromozónoch genómu.
- Chromozómy sa nasekajú na krátke kúsky, každý sa sekvenuje zvlášť napr. Sangerovým sekvenovaním.
 - využíva prírodné enzýmy, napr. DNA polymerázu



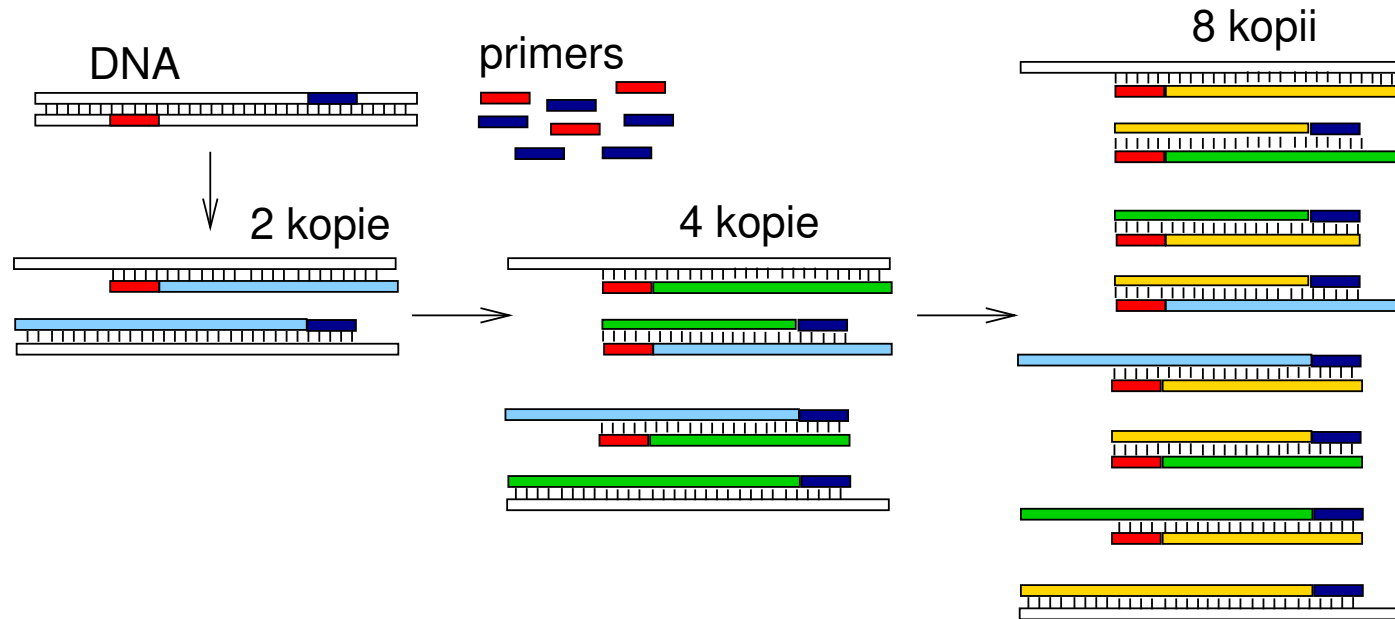
- **Bioinformatický problém:** skladanie celej sekvencie z kúskov.
- Dostupnosť genómov umožňuje katalogizovať gény a iné funkčné úseky, hľadať podobnosti a rozdiely medzi druhmi a jedincami.

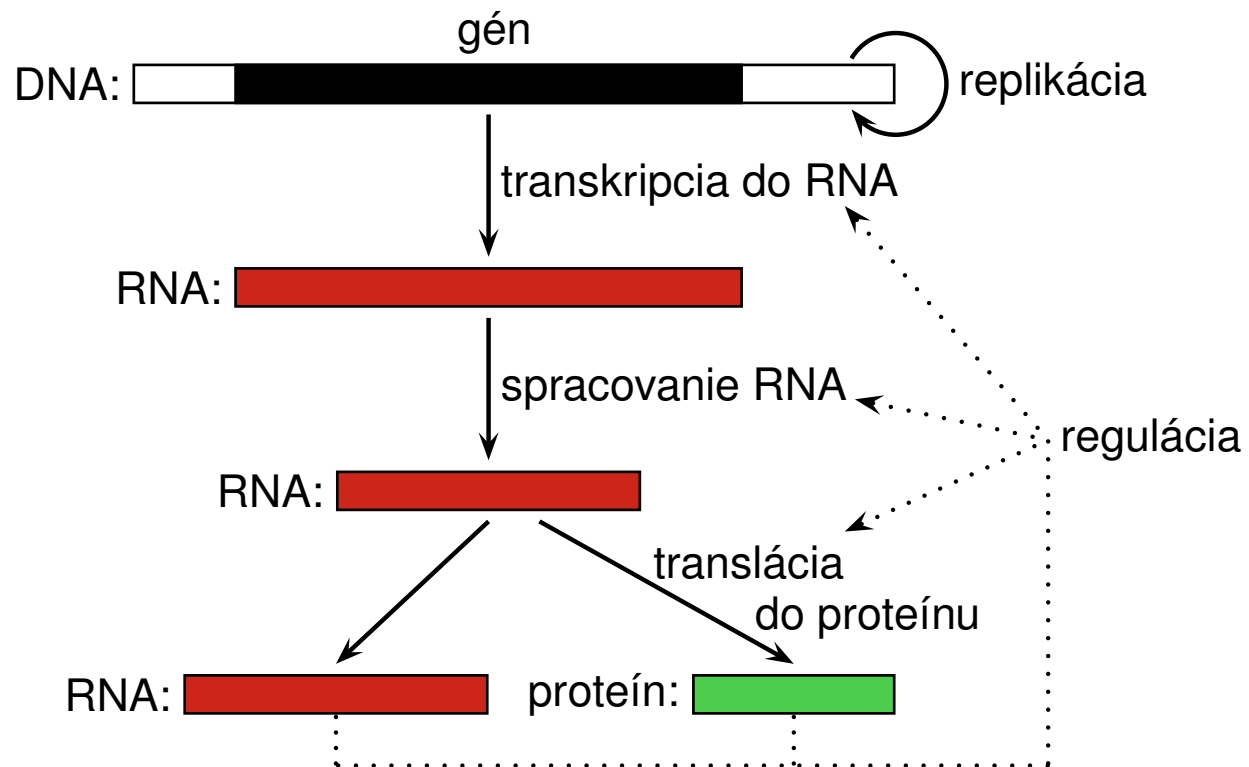
PCR (polymerase chain reaction)

Zvolíme si dva krátke úseky DNA (primers)

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

Ak áno, namnoží úsek medzi nimi

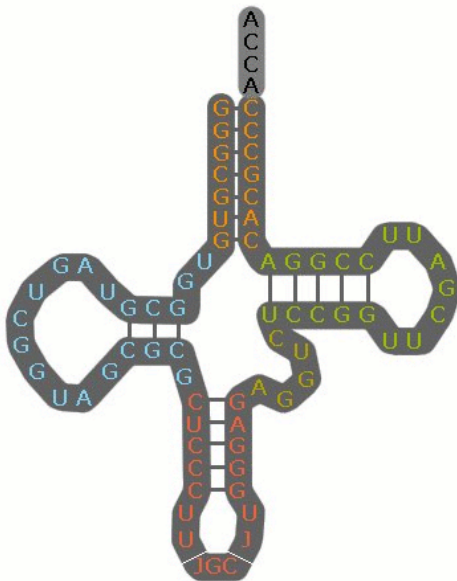




RNA

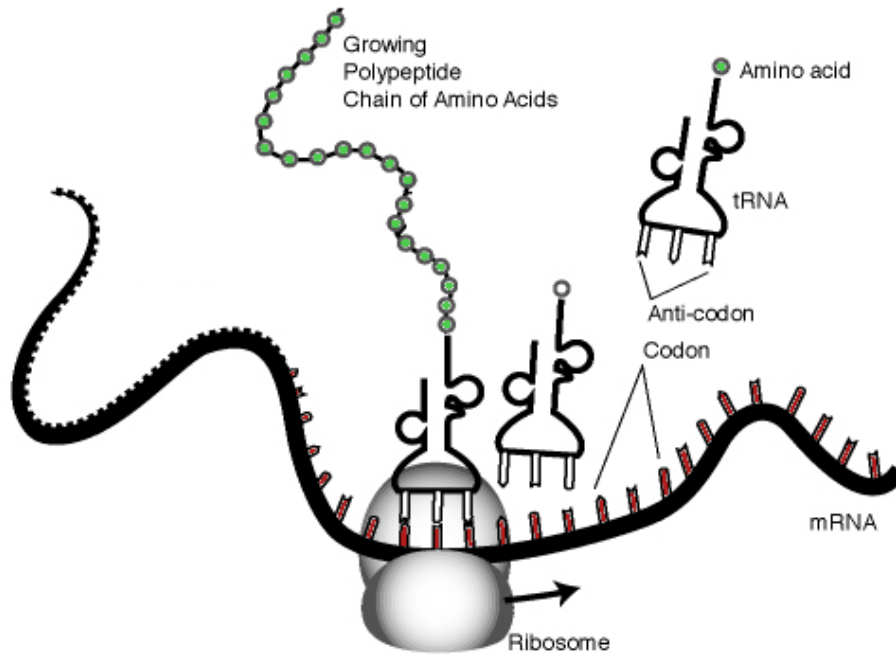
Ako sa líši od DNA?

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

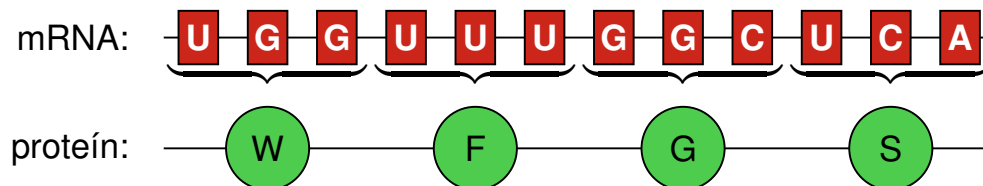


transferová RNA (tRNA)

Translácia



Kodón (trojica nukleotidov) určuje 1 aminokyselinu



Genetický kód

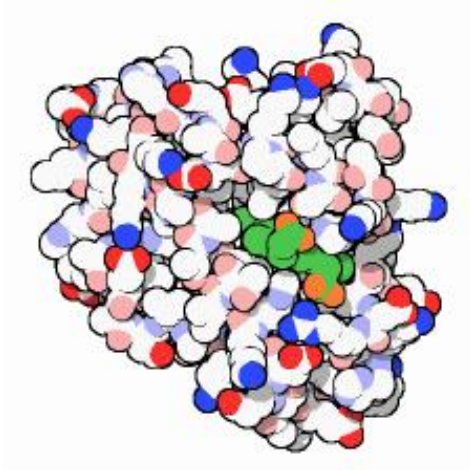
Ala / A	GCT, GCC, GCA, GCG	Leu / L	TTA, TTG, CTT, CTC, CTA, CTG
Arg / R	CGT, CGC, CGA, CGG, AGA, AGG	Lys / K	AAA, AAG
Asn / N	AAT, AAC	Met / M	ATG
Asp / D	GAT, GAC	Phe / F	TTT, TTC
Cys / C	TGT, TGC	Pro / P	CCT, CCC, CCA, CCG
Gln / Q	CAA, CAG	Ser / S	TCT, TCC, TCA, TCG, AGT, AGC
Glu / E	GAA, GAG	Thr / 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)	-CH ₃	hydrofóbny
Arginín (R)	-(CH ₂) ₃ NH-C(NH)NH ₂	bázický
Asparagín (N)	-CH ₂ CONH ₂	hydrofilný
Kyselina asparágová (D)	-CH ₂ COOH	kyslý
Cysteín (C)	-CH ₂ SH	hydrofóbny
Kyselina glutámová (E)	-CH ₂ CH ₂ COOH	kyslý
Glutamín (Q)	-CH ₂ CH ₂ CONH ₂	hydrofilný
Glycín (G)	-H	hydrofilný
Histidín (H)	-CH ₂ -C ₃ H ₃ N ₂	bázický
Izoleucín (I)	-CH(CH ₃)CH ₂ CH ₃	hydrofóbny
Leucín (L)	-CH ₂ CH(CH ₃) ₂	hydrofóbny
Lyzín (K)	-(CH ₂) ₄ NH ₂	bázický
Metionín (M)	-CH ₂ CH ₂ SCH ₃	hydrofóbny
Fenylalanín (F)	-CH ₂ C ₆ H ₅	hydrofóbny
Prolín (P)	-CH ₂ CH ₂ CH ₂ -	hydrofóbny
Serín (S)	-CH ₂ OH	hydrofilný
Treonín (T)	-CH(OH)CH ₃	hydrofilný
Tryptofán (W)	-CH ₂ C ₈ H ₆ N	hydrofóbny
Tyrozín (Y)	-CH ₂ -C ₆ H ₄ OH	hydrofóbny
Valín (V)	-CH(CH ₃) ₂	hydrofóbny

Štruktúra proteínov

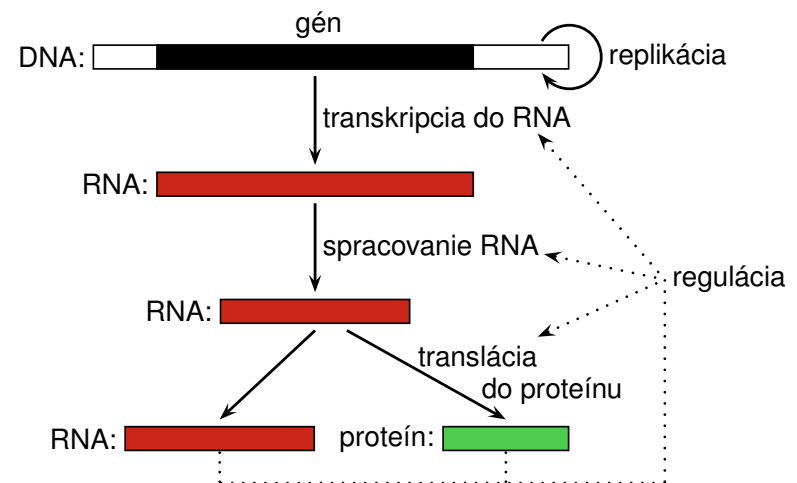
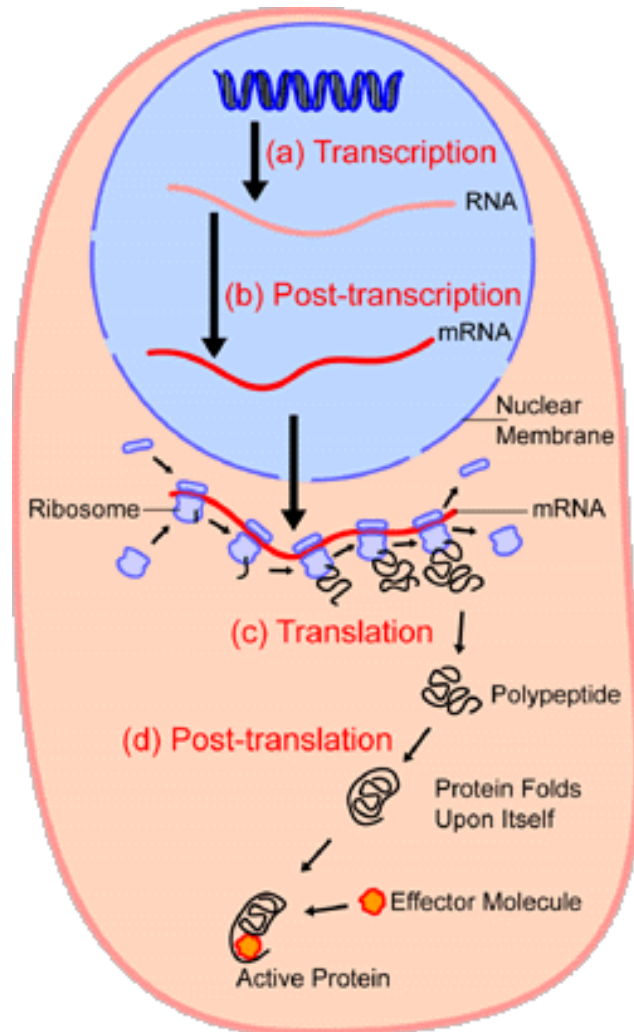


Myoglobín, prvý proteín so známou štruktúrou (Kendrew a kol. 1958).

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

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

Štruktúra proteínu určuje jeho funkciu.

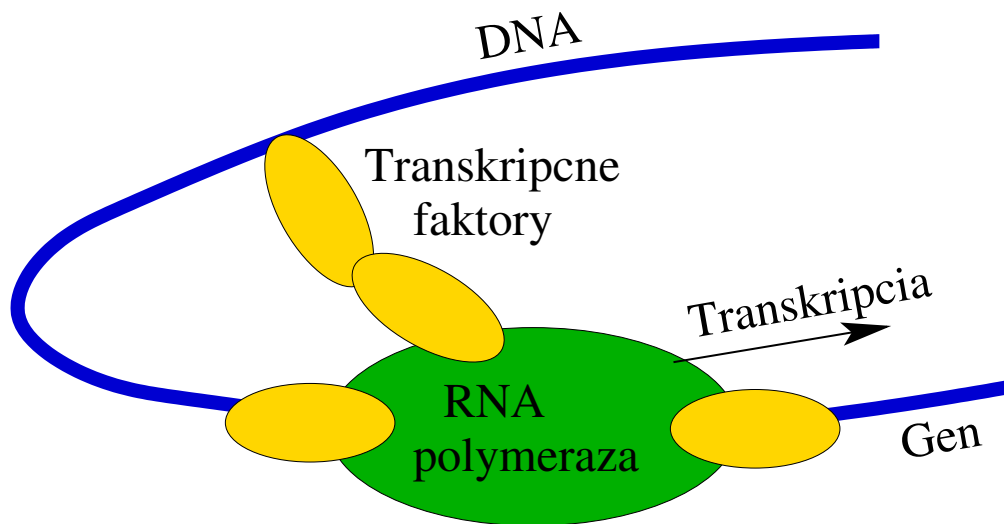


Regulácia expresie

Bunky v rôznych tkanivách toho istého organizmu zdieľajú ten istý genóm, vyzerajú a fungujú však veľmi rôzne.

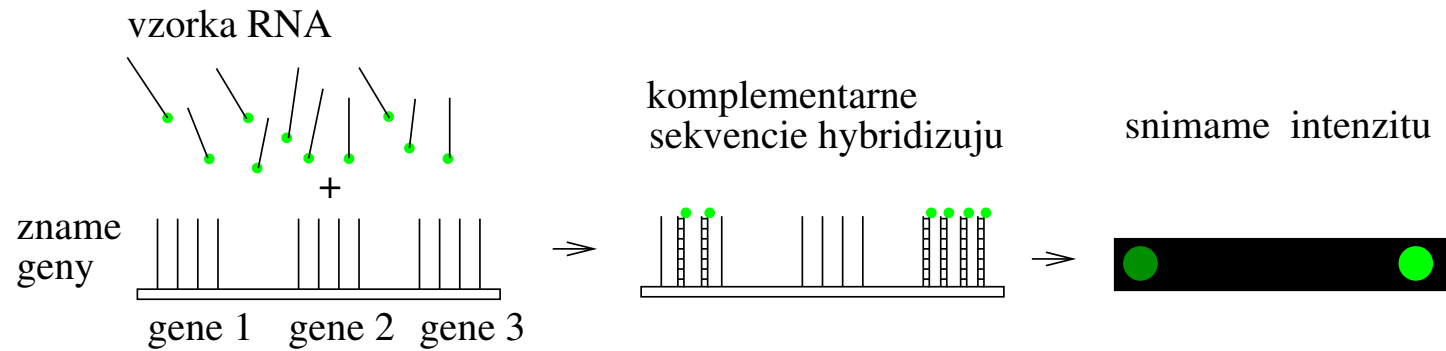
Niektoré proteíny sa tvoria len za určitých okolností, alebo v premenlivom množstve.

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.

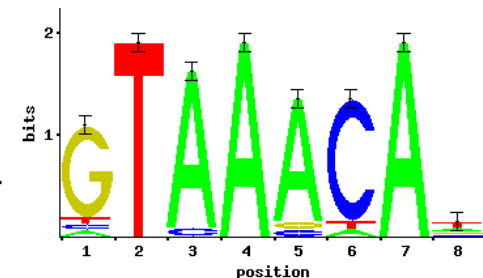
Zopakujeme za rôznych podmienok, študujeme korelácie medzi génmi.

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

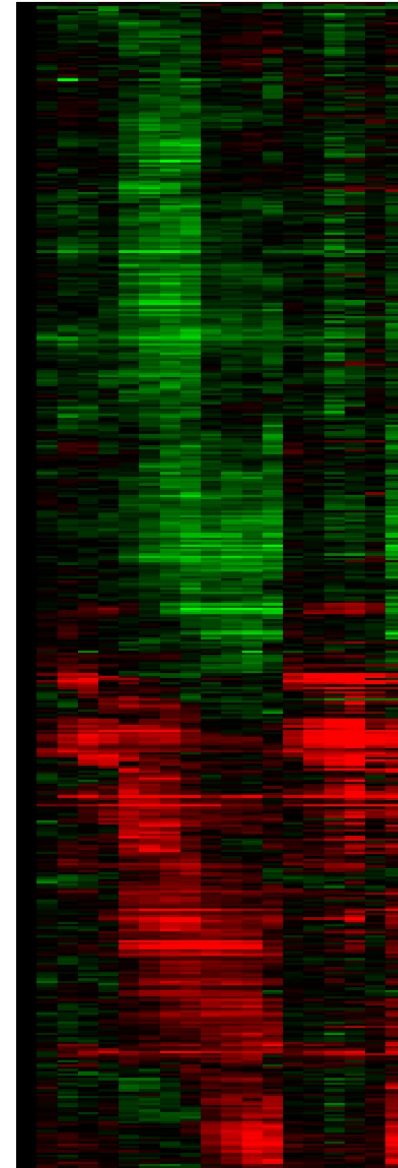
Pomer expresie génu v meranej a kontrolnej vzorke
fg/bg

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:

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

Evolúcia

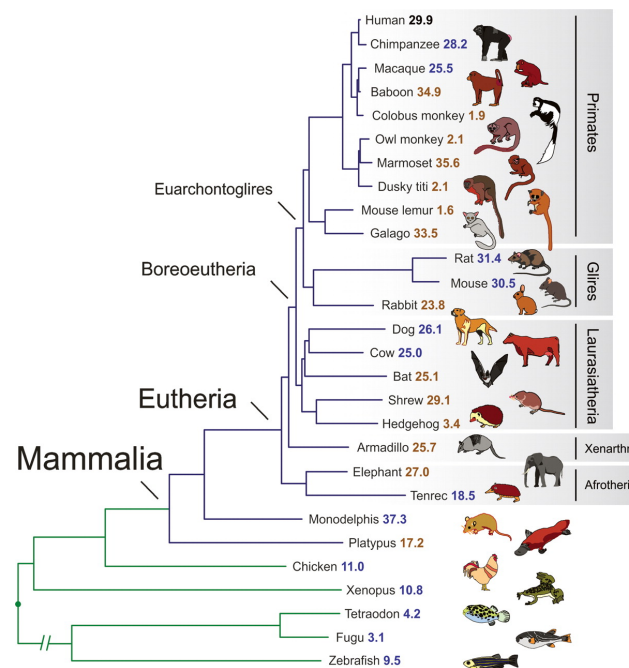
Vznik nových druhov (speciation):

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 určí 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.

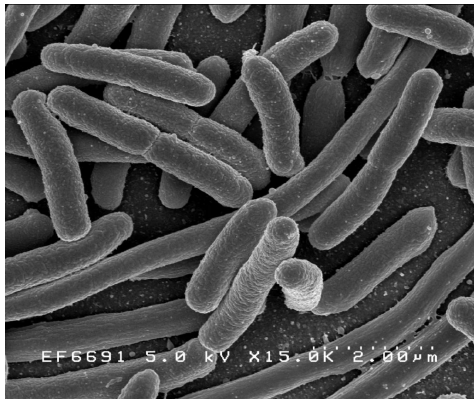
Bunka obsahuje jadro s DNA, viacero organel.

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 manipulovateľné 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

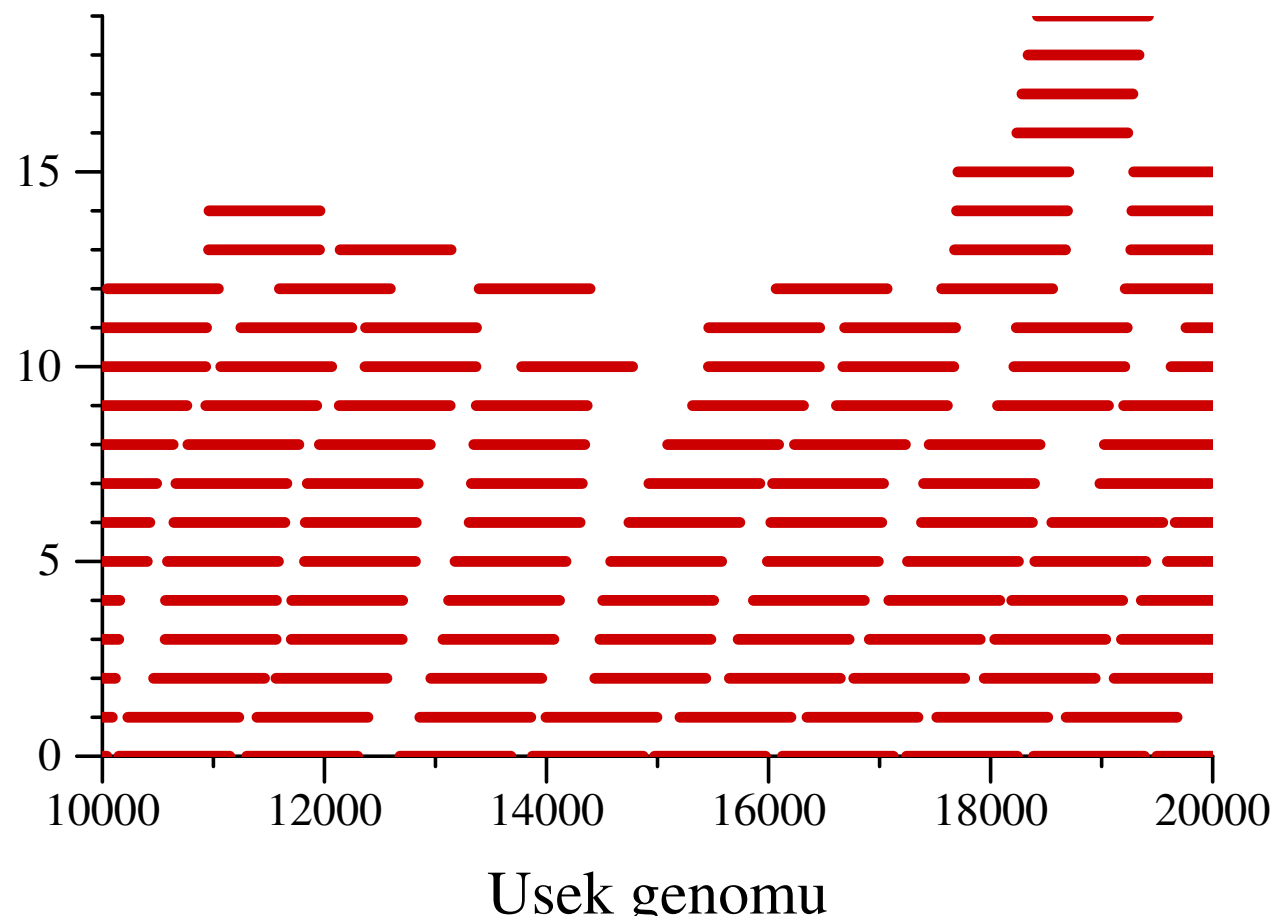
**Úvod do pravdepodobnosti, sekvenovanie genómov
(cvičenie)**

Askar Gafurov

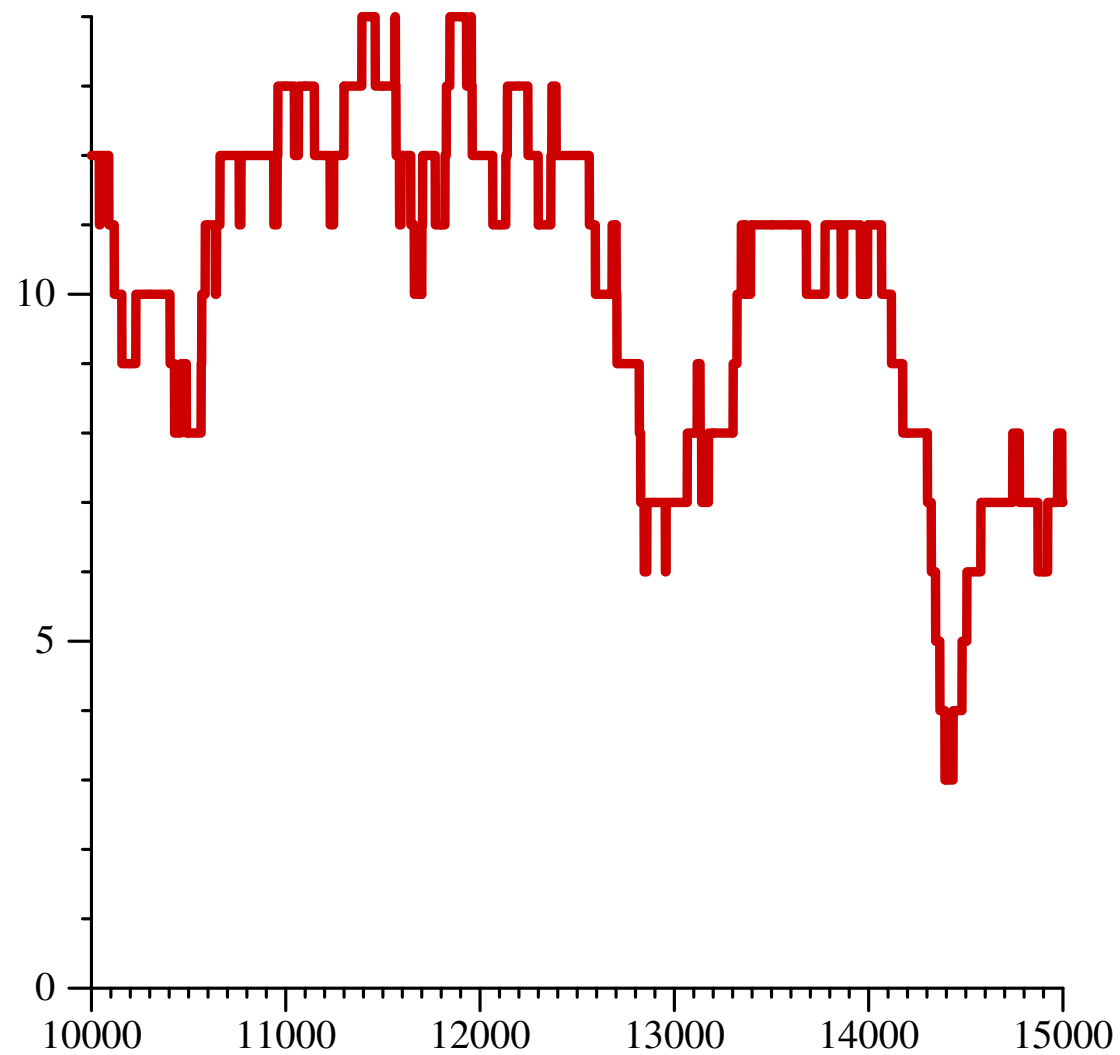
3.10.2019

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

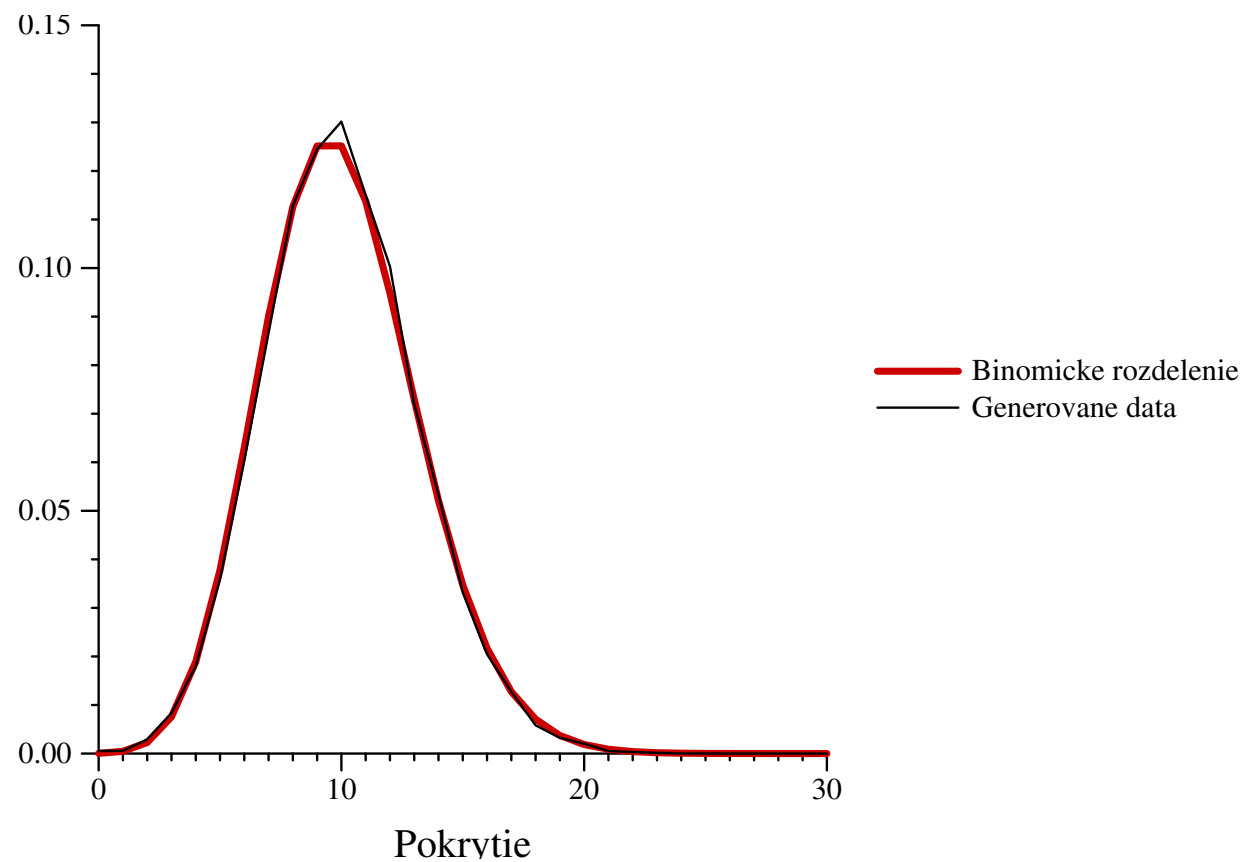
Náhodne generované čítania



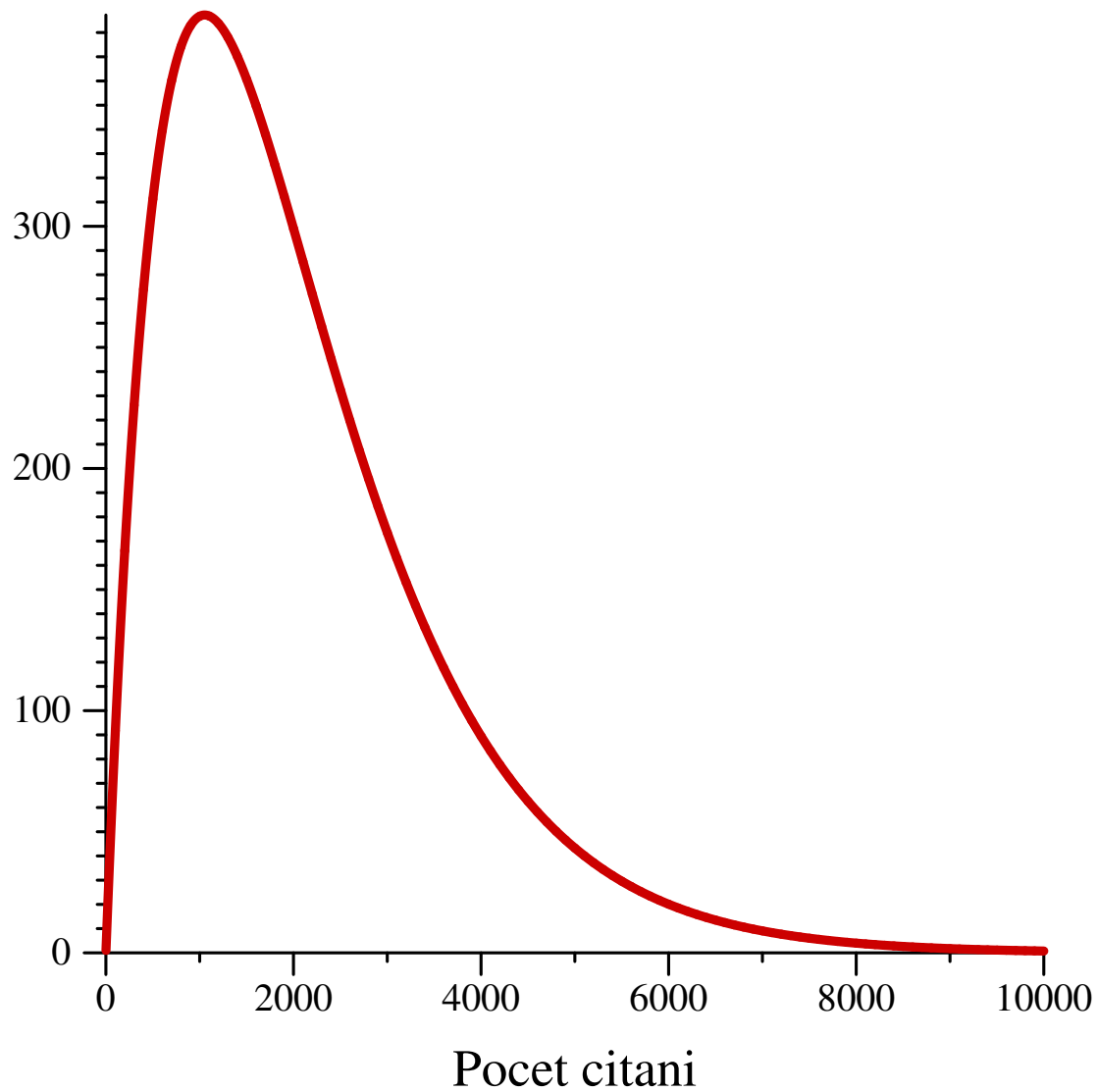
Pokrytie jednotlivých báz



Počet báz s určitým pokrytím



Predpokladaný počet kontigov od počtu čítaní



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

Úvod do dynamického programovania, proteomika

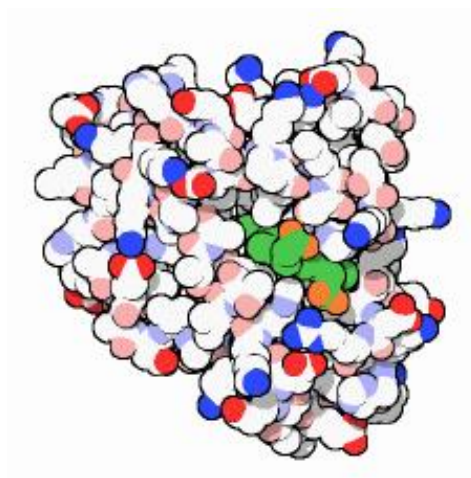
Askar Gafurov

7.10.2021

Proteomika

Proteín: sekvencia pozostáva z 20 rôznych aminokyselín

MGLSDGEWQLVLNVWGKVEADIPGHGQEVLIIRLFKGHPEKFDKFKHLKSEDEMKASE
DLKKHGATVLTALGGILKKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKH
PGDFGADAQGAMNKALELFRKDMASNYKELGFQG



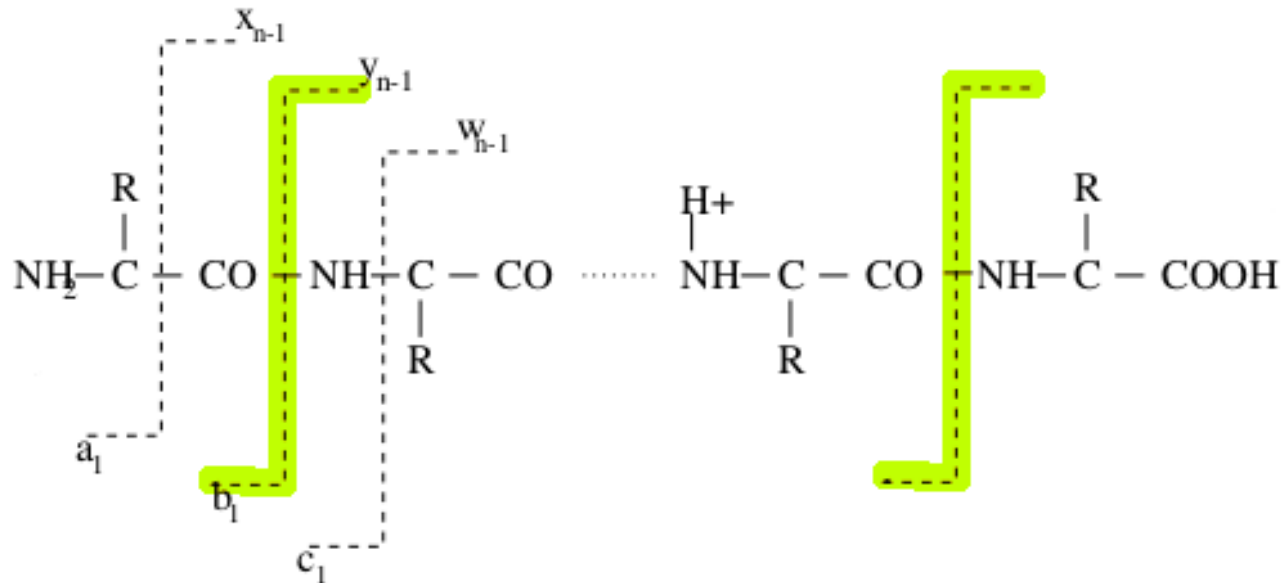
Z bunky sme izolovali určitý proteín, chceme zistiť jeho sekvenciu.

Hmotnostná spektrometria (mass spectrometry)

- Meria pomer hmotnosť/náboj molekúl vo vzorke
- Používa sa na identifikáciu proteínov
- Proteín nasekáme enzýmom trypsín (seká na $[KR] \{P\}$) na peptidy
- Meriame hmotnosť 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



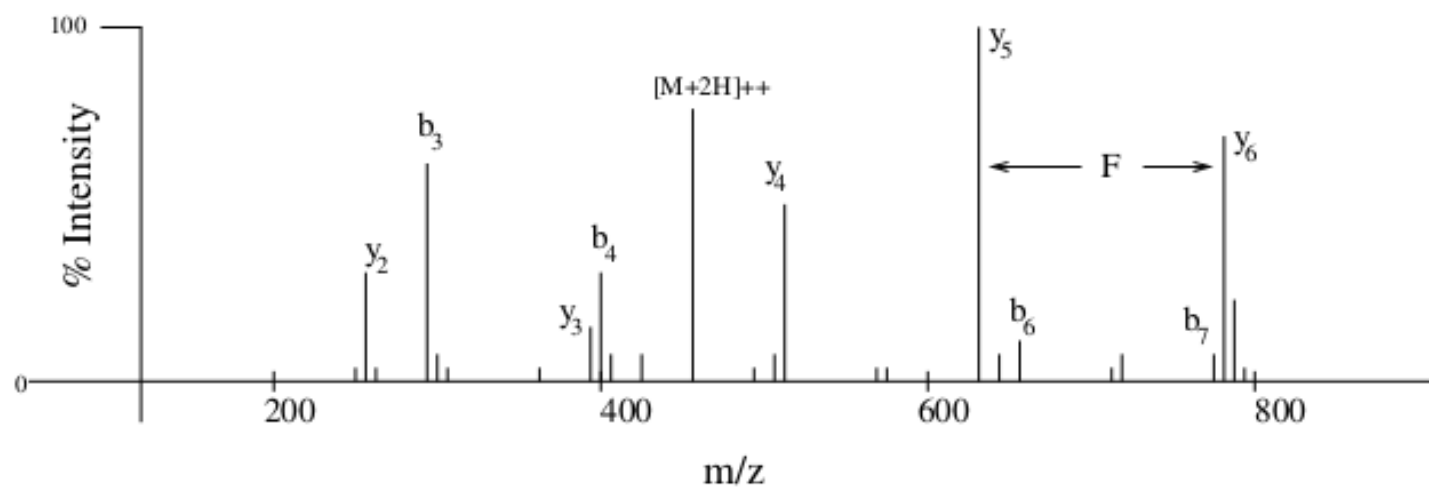
zdroj: Bafna and Reinert

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 $a[1], \dots, a[20]$ (celé čísla),
spektrum ako tabuľka $f[0], \dots, f[M]$, ktorá hmotnosti určí skóre podľa signálu
v okolí príslušného bodu grafu

Označenie:

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

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

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

Nech $\mathcal{M}_S(x) = \{m(x_j \dots x_k) \mid j = 1, \dots, k\}$ 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 $\mathcal{M}_S(XZYY) =$

$$\{m(), m(Y), m(YY), m(ZYY), m(XZYY)\} = \{0, 6, 12, 19, 23\}$$

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

$$\text{Skóre XZXXX: } \sum_{m \in \mathcal{M}_P(ZYZZZ)} f[m] =$$

$$f[0] + f[4] + f[11] + f[15] + f[19] + f[23] = 0 + 1 + 1 + 0 + 1 + 0 = 3$$

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

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

Jadrá zarovnání

Broňa Brejová

20.10.2022

Opakovanie: Heuristické lokálne zarovnávanie, BLAST

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

(V praxi sa používa $k = 10$ a viac.)

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

1. nájdi zhodné úseky

2. rozšír bez medzier

3. spoj medzerami

Senzitivita heuristického algoritmu

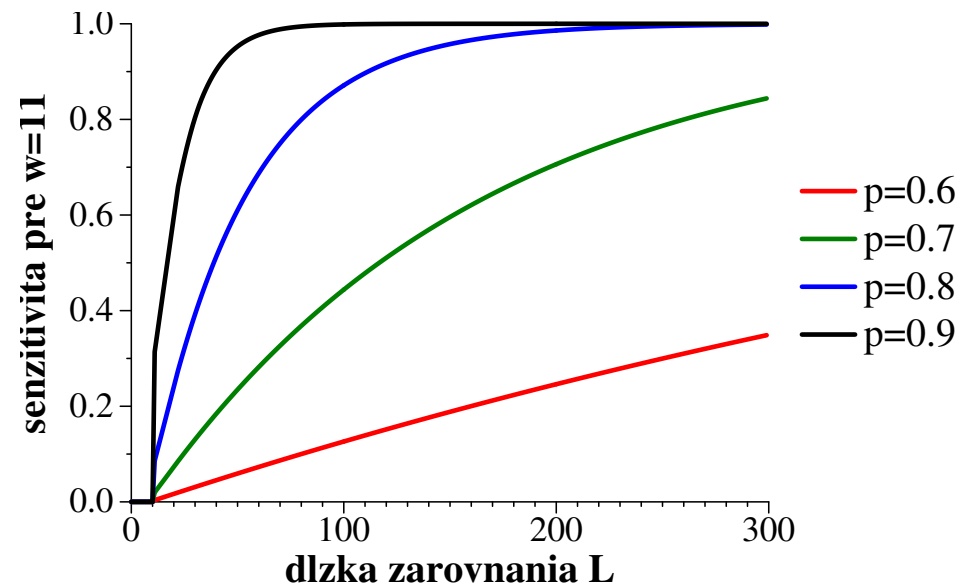
Odhad senzitivity:

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 počítat

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

Opakovanie: ako funguje hľadanie jadier

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

AGTGGCTGCCAGGCTGG

cGaGGCTGCCTGGtTGG

AGTGG, 0

CGAGG

GTGGC, 1

GAGGC

TGGCT, 2

AGGCT <-

GGCTG, 3

GGCTG <-

GCTGC, 4

GCTGC <-

CTGCC, 5

CTGCC <-

TGCCA, 6

TGCCT

GCCAG, 7

GCCTG

CCAGG, 8

CCTGG

CAGGC, 9

CTGGT

AGGCT, 10

TGGTT

GGCTG, 11

GGTTG

GCTGG, 12

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

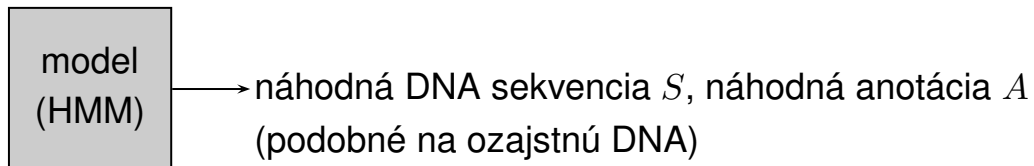
V počtoch k -merov sme zanedbali členy typu $-w + 1$

Algoritmy pre HMM

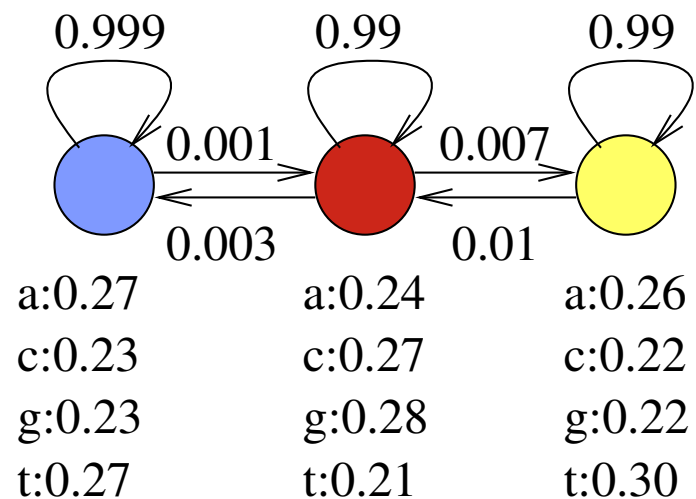
Broňa Brejová

26.10.2023

Opakovanie: HMM (skrytý Markovov model)



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



Predpokladajme, že model vždy začína v modrom stave.

$$\Pr(\text{a} \text{c} \text{a} \text{g}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 \cdot 0.99 \cdot 0.28 = 4.8 \cdot 10^{-6}$$

$$\Pr(\text{a} \text{c} \text{a} \text{g}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 \cdot 0.999 \cdot 0.23 = 0.0038$$

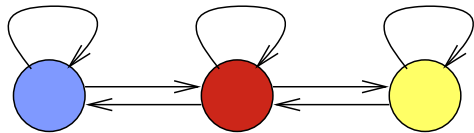
Iný hračkársky príklad: počasie

- Obdobie nízkeho tlaku vzduchu: väčšinou prší
- Obdobie nízkeho tlaku vzduchu: väčšinou slnečno

Každé obdobie trvá typicky niekoľko dní

Cvičenie: reprezentuj ako HMM

Parametre HMM (označenie)



Sekvencia $S = S_1, \dots, S_n$







Anotácia $A = A_1, \dots, A_n$




Parametre modelu:

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

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

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

a			
	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:

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

Viterbiho algoritmus

Pre dané HMM a sekvenciu S

nájdí najpravdepodobnejšiu anotáciu (postupnosť stavov)

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

Ako by ste to riešili?

Pripomeňme si príklad:

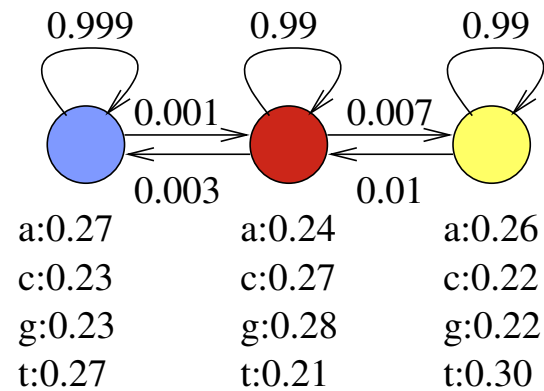
$$\Pr(\text{acaag}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 \cdot 0.99 \cdot 0.28 = 4.8 \cdot 10^{-6}$$




$$\Pr(\text{aacaag}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 \cdot 0.999 \cdot 0.23 = 0.0038$$

Viterbiho algoritmus

Nájdí najpravdepodobnejšiu postupnosť stavov $A = \arg \max_A \Pr(A, S)$

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



$V[u, i]$	a	c	a	g
				
				
				

Viterbiho algoritmus

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

Rekurencia?

$$V[u, 1] =$$

$$V[u, i] =$$

Pripomeňme si označenie:

Sekvencia $S = S_1, \dots, S_n$, anotácia (stavy) $A = A_1, \dots, A_n$

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

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

Viterbiho algoritmus

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

Rekurencia:

$$V[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$V[u, i] = \max_w V[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Algoritmus, celková odpoveď, čas výpočtu?

Pripomeňme si označenie:

Sekvencia $S = S_1, \dots, S_n$, anotácia (stavy) $A = A_1, \dots, A_n$

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

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

Viterbiho algoritmus (zhrnutie)

Nájdí najpravdepodobnejšiu postupnosť stavov $A = \arg \max_A \Pr(A, S)$

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

Rekurencia:

$$V[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$V[u, i] = \max_w V[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Algoritmus:

Inicializuj $V[*, 1]$

for $i = 2 \dots n$ (n =dĺžka S)

 for $u = 1 \dots m$ (m =počet stavov)

 vypočítaj $V[u, i]$, ulož najlepšie w do $B[u, i]$

Maximálne $V[u, n]$ cez všetky u je $\max_A \Pr(A, S)$

Cestu nájdí odzadu pomocou matice B

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

Další problém: celková pravdepodobnosť S

Viterbi počíta $\arg \max_A \Pr(A, S)$

Teraz chceme celkovú pravdepodobnosť, že vygenerujeme sekvenciu S

$$\text{t.j. } \Pr(S) = \sum_A \Pr(A, S)$$

Užitočné napr. na porovnávanie rôznych modelov,

ktorý má väčšiu šancu vygenerovať S

Ako by ste to počítali?

Pripomeňme si príklad:

$$\Pr(\text{acaag}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 \cdot 0.99 \cdot 0.28 = 4.8 \cdot 10^{-6}$$

$$\Pr(\text{aacaag}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 \cdot 0.999 \cdot 0.23 = 0.0038$$

Dopredný algoritmus (forward algorithm)

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S ,

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

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

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

Rekurencia?

$$F[u, 1] =$$

$$F[u, i] =$$

Pripomeňme si rekurenciu z Viterbiho:

$$V[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$V[u, i] = \max_w V[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Dopredný algoritmus (forward algorithm)

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S ,

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

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

Rekurencia

$$F[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$F[u, i] = \sum_w F[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Pripomeňme si rekurenciu z Viterbiho:

$$V[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$V[u, i] = \max_w V[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Dopredný algoritmus (forward algorithm)

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S ,

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

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

Rekurencia

$$F[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$F[u, i] = \sum_w F[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Výsledok?

Celková pravdepodobnosť $\Pr(S) =$

Čas výpočtu?

Dopredný algoritmus (forward algorithm)

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S ,

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

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

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

Výsledok

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

Čas výpočtu $O(nm^2)$

Tretí problem: pravdepodobnosť, že S_i bolo generované v stave u

$$\Pr(A_i = u \mid S) = \frac{\Pr(A_i=u, S)}{\Pr(S)}$$

$$\Pr(A_i = u, S) = \sum_{A: A_i=u} \Pr(A, S)$$

Vypočítame kombináciou dopredného a spätného algoritmu

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

$B[u, i]$: pravdepodobnosť, že ak začneme v u na pozícii i , tak vygenerujeme $S_{i+1} \dots, S_n$ v najbližších krokoch

$$\Pr(A_i = u, S) = F[u, i] \cdot B[u, i]$$

Spätný algoritmus (backward algorithm)

Dopredný algoritmus: pravdepodobnosť, že po i krokoch vygenerujeme S_1, S_2, \dots, S_i a dostaneme sa do stavu u .

$$F[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$F[u, i] = \sum_w F[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Spätný algoritmus: $B[u, i]$: pravdepodobnosť, že ak začneme v u na pozícii i , tak vygenerujeme $S_{i+1} \dots, S_n$ v najbližších krokoch

Ako spočítať $B[u, i]$?

Spätný algoritmus (backward algorithm)

Dopredný algoritmus: pravdepodobnosť, že po i krokoch vygenerujeme S_1, S_2, \dots, S_i a dostaneme sa do stavu u .

$$F[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$F[u, i] = \sum_w F[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Spätný algoritmus: $B[u, i]$: pravdepodobnosť, že ak začneme v u na pozícii i , tak vygenerujeme $S_{i+1} \dots, S_n$ v najbližších krokoch

$$B[u, n] = 1$$

$$B[u, i] = \sum_w B[w, i + 1] \cdot a(u, w) \cdot e(w, S_{i+1})$$

Cvičenie: Ako spočítať $\Pr(S)$ pomocou matice B ?

Aposteriórne dekódovanie (posterior decoding)

Videli sme: $\Pr(A_i = u \mid S) = \frac{F[u,i] \cdot B[u,i]}{\Pr(S)}$

Aposteriórne pravdepodobnosti stavov:

Použitím dopredného a spätného alg. vieme teda spočítať

$\Pr(A_i = u \mid S)$ pre všetky u a i v celkovom čase $O(nm^2)$

Aposteriórne dekódovanie

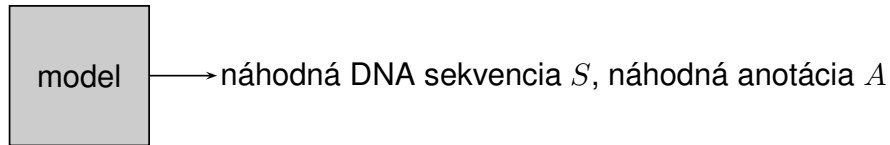
Pre dané S zvolíme A také že $A_i = \max_u \Pr(A_i = u \mid S)$

Výhoda: Berie do úvahy suboptimálne postupnosti stavov

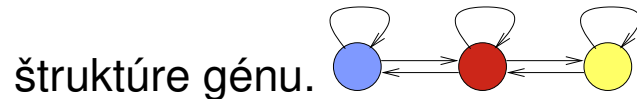
Nevýhoda: $\Pr(A \mid S)$ môže byť 0 alebo veľmi nízka

Iná možnosť: zvolíme A Viterbiho algoritmom, aposteriórne pravdepodobnosti použijeme na priradenie dôveryhodnosti jednotlivým častiam A

Hľadanie génov s HMM



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



- **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 nájdí najpravdepodobnejšiu anotáciu $A = \arg \max_A \Pr(A|S)$ Viterbiho algoritmom v $O(nm^2)$

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

$(S^{(1)}, A^{(1)}), (S^{(2)}, A^{(2)}), \dots$

Cieľ: nastaviť $\pi(u)$, $e(u, x)$, $a(u, v)$ 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

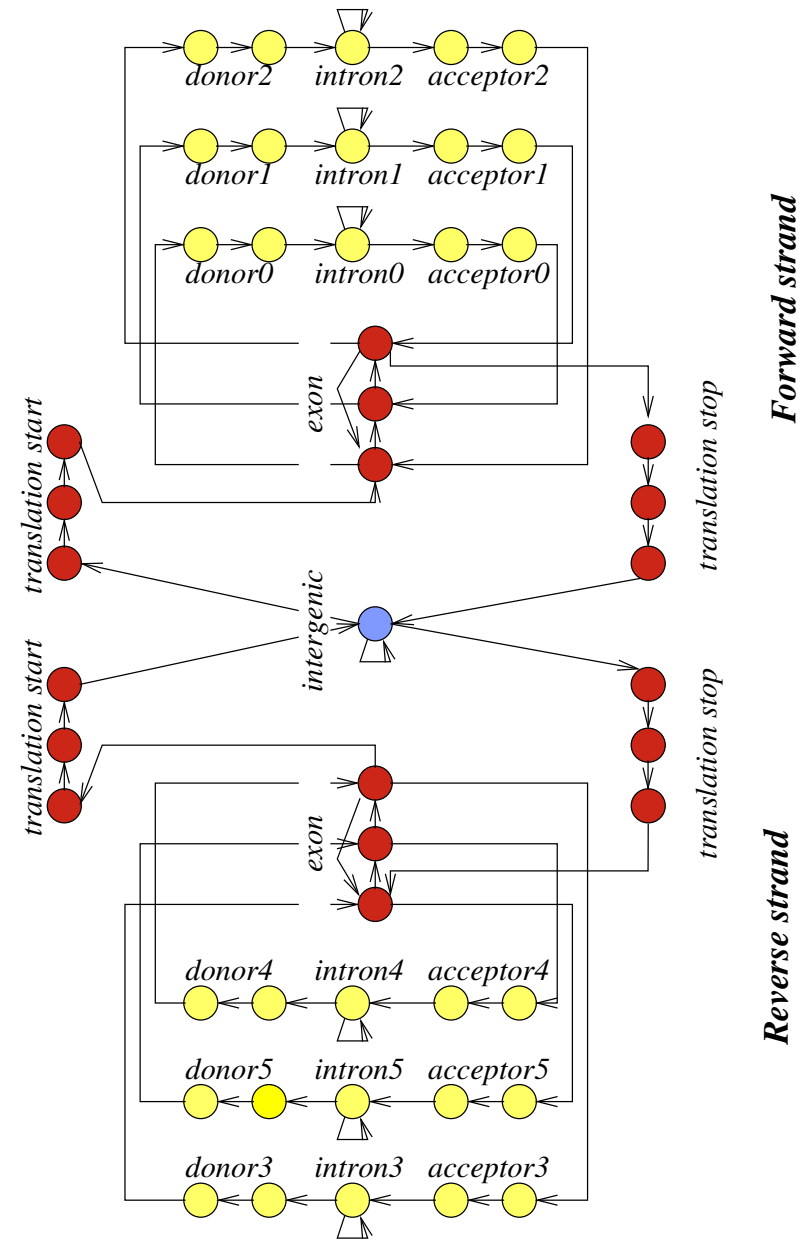
Cieľ: nastaviť $\pi(u)$, $e(u, x)$, $a(u, v)$ tak, aby $\prod_i \Pr(S^{(i)})$ bola čo najväčšia

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

V každej iterácii používa dopradný a spätný algoritmus.

Tvorba stavového priestoru modelu

Príklad HMM na hľadanie génov



Substitution models

Askar Gafurov

November 9, 2023

Modelling the evolution of genomes

- The ultimate goal: to model the evolutionary distance between two genomes
 - ▶ Input: sequences $S_1, S_2 \in \{A, C, G, T\}^* = \Sigma^*$, evolutionary time t
 - ▶ Output: $\Pr[S_1 \xrightarrow{t} S_2]$ (formal way to denote: $\Pr[S_2 \mid S_1, t]$)
 - ★ *Probability of sequence S_1 to mutate into sequence S_2 in evolutionary time t*
 - ★ *Formally: Probability of observing sequence S_2 , given that its evolutionary ancestor in time t is sequence S_1*
- Requirements:
 - ▶ $\Pr[S \xrightarrow{t=0} S] = 1$ (no evolution in zero time)
 - ▶ $\forall S' \in \Sigma^* : \Pr[S' \xrightarrow{t=\infty} S] = \pi_S$ (with enough time, the starting point is irrelevant)
 - ▶ $\Pr[S_1 \xrightarrow{t_1} S_2 \wedge S_2 \xrightarrow{t_2} S_3] = \Pr[S_1 \xrightarrow{t_1} S_2] \cdot \Pr[S_2 \xrightarrow{t_2} S_3]$ (no memory)
 - ▶ $\Pr[S_1 \xrightarrow{t=t_1+t_2} S_3] = \sum_{S_2 \in \Sigma^*} \Pr[S_1 \xrightarrow{t_1} S_2] \cdot \Pr[S_2 \xrightarrow{t_2} S_3]$ (multiplicativity)
 - ★ we can break time t into two parts t_1 and t_2 , and sum over all possible intermediate states

What can we do with such a model (in the near future)

- Given a phylogenetic tree (phylogeny) $T = (\mathbf{S} \subset \Sigma^*, E \subset \mathbf{S}^2, t : E \rightarrow \mathbf{R})$ of sequences \mathbf{S} with times $\mathbf{t}(\cdot, \cdot)$ on the edges, we can compute its total probability by multiplying probabilities of each edge:

$$\Pr[\mathbf{S} \mid E, \mathbf{t}] = \Pr[S_{\text{root}}] \cdot \prod_{e:(S_a, S_s) \in E} \Pr[S_a \xrightarrow{\mathbf{t}(S_a, S_s)} S_s]$$

- This allows us to compute the likelihood $\mathcal{L}(E, \mathbf{t}; \mathbf{S})$ of a potential phylogeny T structure E and times \mathbf{t} w.r.t. sequences \mathbf{S} in the nodes
 - ▶ We can choose the best phylogeny structure by maximizing the total likelihood
- We can even maximize the likelihood using only sequences **in the leaves** (present species) by using the Felsenstein algorithm (*next week*)

Simplifying assumptions

- No indels, only substitutions
 - ▶ $\implies |S_1| = |S_2| = n$
- All bases mutate independently
 - ▶ Compute mutation prob. for each base, and then multiply:

$$\begin{aligned}\Pr[S_1 = (a_1, \dots, a_n) \xrightarrow{t} S_2 = (b_1, \dots, b_n)] &= \\ &= \Pr[a_1 \xrightarrow{t} b_1] \cdot \Pr[a_2 \xrightarrow{t} b_2] \cdot \dots \cdot \Pr[a_n \xrightarrow{t} b_n] = \\ &= \prod_{i=1}^n \Pr[a_i \xrightarrow{t} b_i].\end{aligned}$$

- ▶ Now, we only need to model the **evolution of a single base** $\Pr[a \xrightarrow{t} b]$

Substitution model for one base

- $\Pr[a \xrightarrow{t} b]$ for a fixed time t has only 16 possible input combinations $\{A, C, G, T\}^2$
- Written as a matrix: $S(t) = \begin{pmatrix} \Pr[A \xrightarrow{t} A] & \Pr[A \xrightarrow{t} C] & \Pr[A \xrightarrow{t} G] & \Pr[A \xrightarrow{t} T] \\ \Pr[C \xrightarrow{t} A] & \Pr[C \xrightarrow{t} C] & \Pr[C \xrightarrow{t} G] & \Pr[C \xrightarrow{t} T] \\ \Pr[G \xrightarrow{t} A] & \Pr[G \xrightarrow{t} C] & \Pr[G \xrightarrow{t} G] & \Pr[G \xrightarrow{t} T] \\ \Pr[T \xrightarrow{t} A] & \Pr[T \xrightarrow{t} C] & \Pr[T \xrightarrow{t} G] & \Pr[T \xrightarrow{t} T] \end{pmatrix}$
- General properties of matrix $S(t)$:
 - ▶ $\Pr[C \xrightarrow{t} G] =$

Substitution model for one base

- $\Pr[a \xrightarrow{t} b]$ for a fixed time t has only 16 possible input combinations $\{A, C, G, T\}^2$
- Written as a matrix:
$$S(t) = \begin{pmatrix} \Pr[A \xrightarrow{t} A] & \Pr[A \xrightarrow{t} C] & \Pr[A \xrightarrow{t} G] & \Pr[A \xrightarrow{t} T] \\ \Pr[C \xrightarrow{t} A] & \Pr[C \xrightarrow{t} C] & \Pr[C \xrightarrow{t} G] & \Pr[C \xrightarrow{t} T] \\ \Pr[G \xrightarrow{t} A] & \Pr[G \xrightarrow{t} C] & \Pr[G \xrightarrow{t} G] & \Pr[G \xrightarrow{t} T] \\ \Pr[T \xrightarrow{t} A] & \Pr[T \xrightarrow{t} C] & \Pr[T \xrightarrow{t} G] & \Pr[T \xrightarrow{t} T] \end{pmatrix}$$
- General properties of matrix $S(t)$:
 - ▶ $\Pr[C \xrightarrow{t} G] = (0 \ 1 \ 0 \ 0) \cdot S(t) \cdot (0 \ 0 \ 1 \ 0)^T$
 - ▶ $S(0) =$

Substitution model for one base

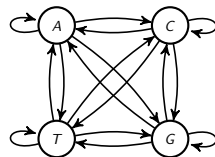
- $\Pr[a \xrightarrow{t} b]$ for a fixed time t has only 16 possible input combinations $\{A, C, G, T\}^2$
- Written as a matrix:
$$S(t) = \begin{pmatrix} \Pr[A \xrightarrow{t} A] & \Pr[A \xrightarrow{t} C] & \Pr[A \xrightarrow{t} G] & \Pr[A \xrightarrow{t} T] \\ \Pr[C \xrightarrow{t} A] & \Pr[C \xrightarrow{t} C] & \Pr[C \xrightarrow{t} G] & \Pr[C \xrightarrow{t} T] \\ \Pr[G \xrightarrow{t} A] & \Pr[G \xrightarrow{t} C] & \Pr[G \xrightarrow{t} G] & \Pr[G \xrightarrow{t} T] \\ \Pr[T \xrightarrow{t} A] & \Pr[T \xrightarrow{t} C] & \Pr[T \xrightarrow{t} G] & \Pr[T \xrightarrow{t} T] \end{pmatrix}$$
- General properties of matrix $S(t)$:
 - ▶ $\Pr[C \xrightarrow{t} G] = (0 \ 1 \ 0 \ 0) \cdot S(t) \cdot (0 \ 0 \ 1 \ 0)^T$
 - ▶ $S(0) = I_4$
 - ▶ $S(t_1) \cdot S(t_2) =$

Substitution model for one base

- $\Pr[a \xrightarrow{t} b]$ for a fixed time t has only 16 possible input combinations $\{A, C, G, T\}^2$
- Written as a matrix:
$$S(t) = \begin{pmatrix} \Pr[A \xrightarrow{t} A] & \Pr[A \xrightarrow{t} C] & \Pr[A \xrightarrow{t} G] & \Pr[A \xrightarrow{t} T] \\ \Pr[C \xrightarrow{t} A] & \Pr[C \xrightarrow{t} C] & \Pr[C \xrightarrow{t} G] & \Pr[C \xrightarrow{t} T] \\ \Pr[G \xrightarrow{t} A] & \Pr[G \xrightarrow{t} C] & \Pr[G \xrightarrow{t} G] & \Pr[G \xrightarrow{t} T] \\ \Pr[T \xrightarrow{t} A] & \Pr[T \xrightarrow{t} C] & \Pr[T \xrightarrow{t} G] & \Pr[T \xrightarrow{t} T] \end{pmatrix}$$
- General properties of matrix $S(t)$:
 - ▶ $\Pr[C \xrightarrow{t} G] = (0 \ 1 \ 0 \ 0) \cdot S(t) \cdot (0 \ 0 \ 1 \ 0)^T$
 - ▶ $S(0) = I_4$
 - ▶ $S(t_1) \cdot S(t_2) = \left(\sum_{x \in \Sigma} \Pr[i \xrightarrow{t_1} x] \cdot \Pr[x \xrightarrow{t_2} j] \right)_{i,j \in \Sigma} \stackrel{\text{multiplicativity}}{=} \left(\Pr[i \xrightarrow{t_1+t_2} j] \right)_{i,j \in \Sigma} = S(t_1 + t_2)$
 - ★ $S(k \cdot t) = S^k(t)$

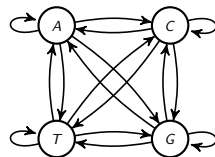
Model with discrete time

- Assume that evolutionary time t is discrete
 - ▶ **at most one mutation** occurs in time 1
- A base now has 4 possible states, and has a chance to transit between them in each time step, or stay the same \implies Markov chain
- $S(t) =$



Model with discrete time

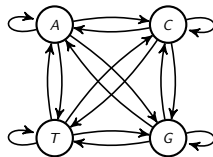
- Assume that evolutionary time t is discrete
 - ▶ **at most one mutation** occurs in time 1
- A base now has 4 possible states, and has a chance to transit between them in each time step, or stay the same \implies Markov chain
- $S(t) = S^t(1) \implies$ only need to define $S(1)$



Model with discrete time

- Assume that evolutionary time t is discrete
 - ▶ **at most one mutation** occurs in time 1
- A base now has 4 possible states, and has a chance to transit between them in each time step, or stay the same \implies Markov chain
- $S(t) = S^t(1) \implies$ only need to define $S(1)$
- Stationary distribution (equilibrium)

$$S(\infty) = \lim_{t \rightarrow \infty} S(t) = \lim_{t \rightarrow \infty} S^t(1) = \begin{pmatrix} \pi_A & \pi_C & \pi_G & \pi_T \\ \pi_A & \pi_C & \pi_G & \pi_T \\ \pi_A & \pi_C & \pi_G & \pi_T \\ \pi_A & \pi_C & \pi_G & \pi_T \end{pmatrix}$$



Quick summary so far

- Evolution model = prob. $\Pr[S_1 \xrightarrow{t} S_2] = \Pr[S_2 \mid S_1, t]$ of observing S_2 given that its ancestor in evolutionary time t is S_1
- Assuming only substitutions
 - ▶ $|S_1| = |S_2| = n$
- Assuming independent evolution for each base
 - ▶ $\Pr[S_1 = (a_1, \dots, a_n) \xrightarrow{t} S_2 = (b_1, \dots, b_n)] = \prod_{i=1}^n \Pr[a_i \xrightarrow{t} b_i]$
 - ▶ Only need to define a (substitution) model for a single base
 - ▶ $\Pr[a \xrightarrow{t} b] = S(t) = \begin{pmatrix} \Pr[A \xrightarrow{t} A] & \Pr[A \xrightarrow{t} C] & \Pr[A \xrightarrow{t} G] & \Pr[A \xrightarrow{t} T] \\ \Pr[C \xrightarrow{t} A] & \Pr[C \xrightarrow{t} C] & \Pr[C \xrightarrow{t} G] & \Pr[C \xrightarrow{t} T] \\ \Pr[G \xrightarrow{t} A] & \Pr[G \xrightarrow{t} C] & \Pr[G \xrightarrow{t} G] & \Pr[G \xrightarrow{t} T] \\ \Pr[T \xrightarrow{t} A] & \Pr[T \xrightarrow{t} C] & \Pr[T \xrightarrow{t} G] & \Pr[T \xrightarrow{t} T] \end{pmatrix}$
 - ▶ $S(t_1 + t_2) = S(t_1) \cdot S(t_2)$
- For discrete time, only need to define $S(1)$
 - ▶ Classic Markov chain with states $\{A, C, G, T\}$, $S(1) =$ matrix of transition probabilities

Jukes-Cantor JC69 model

- The plan: define Markov chains with continuous time (CTMC), where all substitutions are equally likely

- ▶ $S(t) =$

Jukes-Cantor JC69 model

- The plan: define Markov chains with continuous time (CTMC), where all substitutions are equally likely

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

Jukes-Cantor JC69 model

- The plan: define Markov chains with continuous time (CTMC), where all substitutions are equally likely

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

- Let's look at $s(t)$ closely

$$\blacktriangleright s(0) = 0$$

- \blacktriangleright Let's denote the first derivative of $s(t)$ at zero as α :

$$\star \text{ Formally, } \alpha := s'(0) \stackrel{\text{def.}}{=} \lim_{\varepsilon \rightarrow 0} \frac{s(0 + \varepsilon) - s(0)}{\varepsilon} = \lim_{\varepsilon \rightarrow 0} \frac{s(\varepsilon)}{\varepsilon}$$

$$\star \alpha = \left. \frac{\partial \Pr[a \xrightarrow{t} b]}{\partial t} \right|_{t=0}$$

Derivative of $S(t)$

$$\begin{aligned} S'(t) &\stackrel{\text{def.}}{=} \lim_{\varepsilon \rightarrow 0} \frac{S(t + \varepsilon) - S(t)}{\varepsilon} = \lim_{\varepsilon \rightarrow 0} \frac{S(t)S(\varepsilon) - S(t)}{\varepsilon} = \\ &= \lim_{\varepsilon \rightarrow 0} \frac{S(t)(S(\varepsilon) - I)}{\varepsilon} = \lim_{\varepsilon \rightarrow 0} \frac{S(t) \cdot \begin{pmatrix} -3 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 \\ 1 & 1 & -3 & 1 \\ 1 & 1 & 1 & -3 \end{pmatrix} \cdot s(\varepsilon)}{\varepsilon} = \\ &= S(t) \cdot \begin{pmatrix} -3 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 \\ 1 & 1 & -3 & 1 \\ 1 & 1 & 1 & -3 \end{pmatrix} \cdot \lim_{\varepsilon \rightarrow 0} \frac{s(\varepsilon)}{\varepsilon} = \\ &= S(t) \cdot \begin{pmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{pmatrix} \end{aligned}$$

Differential equation

- We've got diff. equation $S'(t) = S(t) \cdot R$, where $R = \begin{pmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{pmatrix}$
- R is called transition rate matrix
- It is really a system of 16 ordinary differential equations $S'(t)_{a,b} = (S(t) \cdot R)_{a,b}$
 - ▶ for (A, A) : $-3s'(t) = (1 - 3s(t))(-3\alpha) + 3s(t)\alpha = -3\alpha + 12\alpha s(t)$
 - ★ $s'(t) = \alpha - 4\alpha s(t)$
 - ▶ for (A, C) : $s'(t) = (1 - 3s(t))\alpha + s(t)(-3\alpha) + 2s(t)\alpha = \alpha - 4\alpha s(t)$
 - ▶ which reduces to a single ordinary differential equation $s'(t) = \alpha - 4\alpha s(t)$ with start condition $s(0) = 0$
- Solution: $s(t) = \frac{1}{4} - \frac{1}{4}e^{-4\alpha t}; \quad 1 - 3s(t) = \frac{1}{4} + \frac{3}{4}e^{-4\alpha t}$

$$\frac{ds}{dt} = \alpha - 4\alpha s$$

$$\frac{ds}{\alpha - 4\alpha s} = dt$$

$$\frac{1}{\alpha} \int \frac{ds}{1 - 4s} = \int 1 dt$$

$$|(1 - 4s) = x, -4ds = dx|$$

$$\frac{1}{-4\alpha} \int \frac{dx}{x} = \int 1 dt$$

$$\frac{1}{-4\alpha} \ln(1 - 4s) = t + C$$

$$1 - 4s = e^{-4\alpha t + C}$$

$$s = \frac{1 - e^{-4\alpha t + C}}{4}$$

$$s(0) = 0 \implies \frac{1 - e^C}{4} = 0 \implies C = 0$$

$$\text{Solution: } s(t) = \frac{1 - e^{-4\alpha t}}{4}; 1 - 3s(t) = \frac{1}{4} + \frac{3}{4}e^{-4\alpha t}$$

Equilibrium for Jukes-Cantor model

$$\begin{aligned}\lim_{t \rightarrow \infty} \Pr[A \xrightarrow{t} A] &= \lim_{t \rightarrow \infty} \frac{1}{4} + \frac{3}{4}e^{-4\alpha t} = \frac{1}{4} \\ \lim_{t \rightarrow \infty} \Pr[A \xrightarrow{t} C] &= \lim_{t \rightarrow \infty} \frac{1}{4} - \frac{1}{4}e^{-4\alpha t} = \frac{1}{4}\end{aligned}$$

$$S(\infty) = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$$

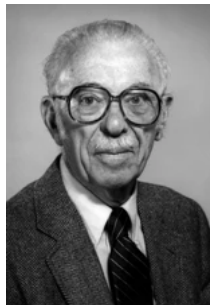
Quick summary so far

- Jukes-Cantor substitution model:

- ▶ Continuous time t
- ▶ Equal probability of substitution $\forall a \neq b : \Pr[a \xrightarrow{t} b] = s(t)$
- ▶ Matrix form

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

- Diff. equation $s'(t) = 1 - 3s(t), s(0) = 0$
- $\Pr[a \xrightarrow{t} b] = S_{JC}(t)_{a,b} = \begin{cases} \frac{1}{4} + \frac{3}{4}e^{-4\alpha t} & a = b \\ \frac{1}{4} - \frac{1}{4}e^{-4\alpha t} & a \neq b \end{cases}$
- Equilibrium for JC: $\pi_A = \pi_C = \pi_G = \pi_T = \frac{1}{4}$



Example for Jukes-Cantor

- Input: $S_1 = TAACCGT$, $S_2 = AATGCGT$, evolutionary time $t = 0.5$, $\alpha = 3$
- Result:

$$\begin{aligned}\Pr[S_1 \xrightarrow{t} S_2] &= \prod_{i=1}^n \Pr[a_i \xrightarrow{t} b_i] = \left(\frac{1}{4} + \frac{3}{4}e^{-4\alpha t}\right)^{\#(a_i=b_i)} \cdot \left(\frac{1}{4} - \frac{1}{4}e^{-4\alpha t}\right)^{\#(a_i \neq b_i)} = \\ &= \left(\frac{1}{4} + \frac{3}{4}e^{-6}\right)^4 \cdot \left(\frac{1}{4} - \frac{1}{4}e^{-6}\right)^3 \approx (0.2519)^4 \cdot (0.2493)^3 \approx 0.0000624\end{aligned}$$

Example for Jukes-Cantor

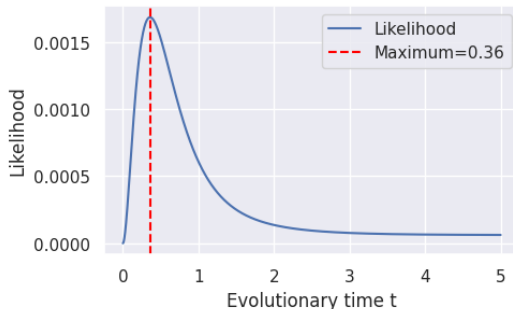
- Input: $S_1 = TAACCGT$, $S_2 = AATGCGT$, evolutionary time $t = 0.5$, $\alpha = 3$
- Result:

$$\begin{aligned}\Pr[S_1 \xrightarrow{t} S_2] &= \prod_{i=1}^n \Pr[a_i \xrightarrow{t} b_i] = \left(\frac{1}{4} + \frac{3}{4}e^{-4\alpha t}\right)^{\#(a_i=b_i)} \cdot \left(\frac{1}{4} - \frac{1}{4}e^{-4\alpha t}\right)^{\#(a_i \neq b_i)} = \\ &= \left(\frac{1}{4} + \frac{3}{4}e^{-6}\right)^4 \cdot \left(\frac{1}{4} - \frac{1}{4}e^{-6}\right)^3 \approx (0.2519)^4 \cdot (0.2493)^3 \approx 0.0000624\end{aligned}$$

- Notice that parameters $t = 30$, $\alpha = 1/20$ would give the same result
 - ▶ Because t and α are always in a product
 - ▶ Standard practice is to select α such that $E[\# \text{ mutations in time } t = 1] = 1$
 - ★ $\# \text{ mutations in time } t = 1 \sim \text{Poisson}(\lambda = 3\alpha)$, $E = 3\alpha$, $E[\#] = 1$ when $\alpha = 1/3$

Estimation of evolutionary time in JC model

- Input: $S_1 = TAACCGT$, $S_2 = AATGCGT$, $\alpha = 1/3$ (standard)
- Goal: find the best evolutionary time t^*
- Best = with highest likelihood
 - ▶ likelihood $\mathcal{L}(t; S_1, S_2, \alpha) = \Pr[S_1 \xrightarrow{t} S_2 \mid \alpha] = \left(\frac{1}{4} + \frac{3}{4}e^{-4\alpha t}\right)^{\#(a_i=b_i)} \cdot \left(\frac{1}{4} - \frac{1}{4}e^{-4\alpha t}\right)^{\#(a_i \neq b_i)}$.
 - ▶ $t^* = \arg \max_{t \geq 0} \mathcal{L}(t; S_1, S_2, \alpha) = -\frac{1}{4\alpha} \ln\left(1 - \frac{4}{3}d\right)$, where $d :=$ proportion of different positions



Exact estimator of evolutionary time in JC model

$$t^* = \arg \max_{t \geq 0} \mathcal{L}(t; S_1, S_2, \alpha) = \arg \max_{t \geq 0} \log \mathcal{L}(t; S_1, S_2, \alpha) =$$

$$= \arg \max_{t \geq 0} \#(a_i = b_i) \log(1 - 3s(t)) + \#(a_i \neq b_i) \log s(t).$$

$$\frac{df}{ds} = -\frac{3\#(a_i = b_i)}{1 - 3s} + \frac{\#(a_i \neq b_i)}{s} = \frac{(1 - 3s)\#(\neq) - 3s\#(=)}{s(1 - 3s)}.$$

$$\frac{ds}{dt} = \alpha \cdot e^{-4\alpha t}.$$

$$\frac{df}{dt} = 0 \implies \frac{df}{ds} \frac{ds}{dt} = 0 \implies \frac{df}{ds} = 0 \implies \frac{(1 - 3s)\#(\neq) - 3s\#(=)}{s(1 - 3s)} = 0 \implies$$

$$\implies (1 - 3s)\#(\neq) - 3s\#(=) = 0 \implies s = \frac{\#(\neq)}{3 \cdot (\#(\neq) + \#(=))} = \frac{\#(\neq)}{3n}.$$

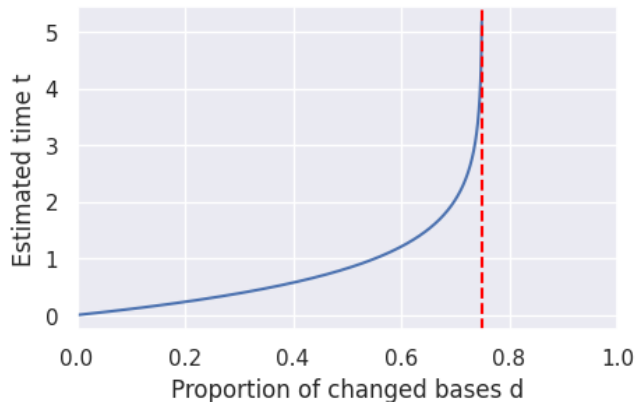
$$\frac{1}{4} - \frac{1}{4}e^{-4\alpha t} = \frac{\#(\neq)}{3n} \implies -4\alpha t = \ln \left(1 - \frac{4\#(\neq)}{3n} \right) \implies$$

$$\implies t = \frac{-\ln \left(1 - \frac{4}{3} \frac{\#(\neq)}{n} \right)}{4\alpha} = \frac{-\ln \left(1 - \frac{4}{3} d \right)}{4\alpha}.$$



Behaviour of the time estimator

$$t^* = -\frac{1}{4\alpha} \ln \left(1 - \frac{4}{3} \cdot d \right)$$



More general models

- JC69 model: rate matrix $R_{JC69} = \begin{pmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{pmatrix}$
- Sum in a row must equal to 0
- $R_{a,b} := \frac{\partial \Pr[a \xrightarrow{t} b]}{\partial t}$ speed of change from a to b
- In general: $R = \begin{pmatrix} * & \mu_{A,C} & \mu_{A,G} & \mu_{A,T} \\ \mu_{C,A} & * & \mu_{C,G} & \mu_{C,T} \\ \mu_{G,A} & \mu_{G,C} & * & \mu_{G,T} \\ \mu_{T,A} & \mu_{T,C} & \mu_{T,G} & * \end{pmatrix}$
 - ▶ Diagonal is set to make row sum up to 0
 - ▶ Some regularity conditions apply

Solution to a general model

- The differential equation $S'(t) = S(t) \cdot R$ holds for any rate matrix R
- The general solution is $S(t) = e^{Rt}$
- How to compute e^{Rt} ?
 - ▶ diagonalization of matrix $R = Q \cdot \Lambda \cdot Q^{-1}$, where
 - ★ Q = orthogonal matrix (of eigenvectors)
 - ★ $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_4)$ is a diagonal matrix (of eigenvalues)
 - ▶ $R^n = (Q \cdot \Lambda \cdot Q^{-1})^n = Q \Lambda Q^{-1} Q \Lambda Q^{-1} Q \dots Q^{-1} Q \Lambda Q^{-1} = Q \Lambda^n Q^{-1} = Q \cdot \text{diag}(\lambda_1^n, \dots, \lambda_4^n) \cdot Q^{-1}$

$$\begin{aligned} e^{Rt} &= \sum_{i=0}^{\infty} \frac{(Rt)^n}{n!} = \sum_{i=0}^n \frac{Q \cdot \text{diag}((\lambda_1 t)^n, \dots, (\lambda_4 t)^n) \cdot Q^{-1}}{n!} = \\ &= Q \cdot \text{diag} \left(\sum_{i=0}^{\infty} \frac{(\lambda_1 t)^n}{n!}, \dots, \sum_{i=0}^{\infty} \frac{(\lambda_4 t)^n}{n!} \right) \cdot Q^{-1} = Q \cdot \text{diag} \left(e^{\lambda_1 t}, \dots, e^{\lambda_4 t} \right) \cdot Q^{-1} \end{aligned}$$

Solution in general form

$$\frac{dS}{dt} = SR \implies \int \frac{dS}{S} = \int R dt \implies \ln S = Rt + C \implies S = e^{Rt+C}; S(0) = I \implies S(t) = e^{Rt}$$

$$R_{JC69} = \begin{pmatrix} -1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \cdot \text{diag}(-4\alpha, -4\alpha, -4\alpha, 0) \cdot \begin{pmatrix} -0.25 & -0.25 & -0.25 & 0.75 \\ -0.25 & -0.25 & 0.75 & 0.25 \\ -0.25 & 0.75 & -0.25 & -0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$$

$$S_{JC69}(t) = \begin{pmatrix} -1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \cdot \text{diag}(e^{-4\alpha t}, e^{-4\alpha t}, e^{-4\alpha t}, 1) \cdot \begin{pmatrix} -0.25 & -0.25 & -0.25 & 0.75 \\ -0.25 & -0.25 & 0.75 & 0.25 \\ -0.25 & 0.75 & -0.25 & -0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$$



Kimura's K80 model

- Also called Kimura's 2 parameter model (K2P)
- A and G are **purines**, C and T are **pyrimidines**
 - ▶ Transitions: within the same group $A \longleftrightarrow G$, $C \longleftrightarrow T$
 - ▶ Transversions: between the groups
- **Transitions are more frequent** than transversions
 - ▶ $\kappa := \frac{\text{rate of transitions}}{\text{rate of transversions}}$, set rate of transversions to 1
- $R_{K80} = \begin{pmatrix} * & 1 & \kappa & 1 \\ 1 & * & 1 & \kappa \\ \kappa & 1 & * & 1 \\ 1 & \kappa & 1 & * \end{pmatrix}$
- Equilibrium is still $\pi_A = \pi_C = \pi_G = \pi_T = 25\%$



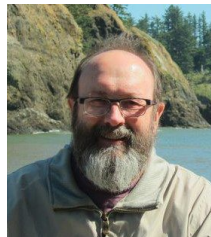
Hasewaga-Kishino-Yano HKY85 model

- Transition/transversion ratio κ & arbitrary equilibrium $(\pi_A, \pi_C, \pi_G, \pi_T)$

- $R_{HKY85} = \begin{pmatrix} * & \pi_C & \kappa \cdot \pi_G & \pi_T \\ \pi_A & * & \pi_G & \kappa \cdot \pi_T \\ \kappa \cdot \pi_A & \pi_C & * & \pi_T \\ \pi_A & \kappa \cdot \pi_C & \pi_G & * \end{pmatrix}$

Other models

- Kimura's 3 parameter model (K3P, K81)
 - ▶ 1 transition rate + 2 transversion rates
 - ▶ admits Hadamard transformation (generalized Fourier)
- Felsenstein F81 model
 - ▶ JC69 + arbitrary equilibrium
- Tamura T92 model
 - ▶ K80 + GC content
- Tamura and Nei TN93 model
 - ▶ 2 transition rates + 1 transversion rate
- Tavaré GTR86 model (General Time Reversible)
 - ▶ everything from the above: arbitrary equilibrium + 6 rate parameters



Summary

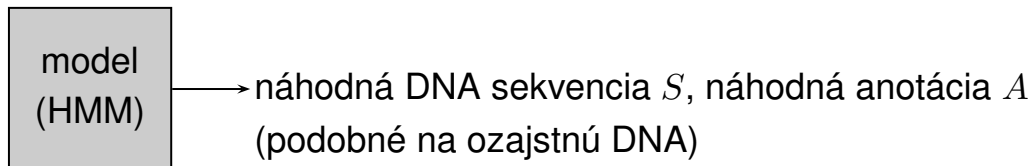
- Evolution model: $\Pr[S_1 \xrightarrow{t} S_2]$
 - ▶ Independent base evolution $\implies \Pr[S_1 \xrightarrow{t} S_2] = \prod_{i=1}^n \Pr[a_i \xrightarrow{t} b_i]$
 - ▶ Continuous time t + Only substitutions \implies Continuous time Markov chains (CTMC)
- Substitution model for one base (CTMC)
 - ▶ substitution rate matrix $R = \begin{pmatrix} * & \mu_{A,C} & \mu_{A,G} & \mu_{A,T} \\ \mu_{C,A} & * & \mu_{C,G} & \mu_{C,T} \\ \mu_{G,A} & \mu_{G,C} & * & \mu_{G,T} \\ \mu_{T,A} & \mu_{T,C} & \mu_{T,G} & * \end{pmatrix}$, rows sum up to zero
 - ▶ $S_{a,b}(t) = \Pr[a \xrightarrow{t} b]$ from $S(t) = e^{Rt}$ using diagonalization trick
- Different rate matrices R give different models:
 - ▶ JC69 model: all substitutions are equally likely, equilibrium 25%
 - ▶ K80 model: transition/transversion ratio κ , equilibrium 25%
 - ▶ HKY85 model: K80 + arbitrary equilibrium

Algoritmy pre HMM

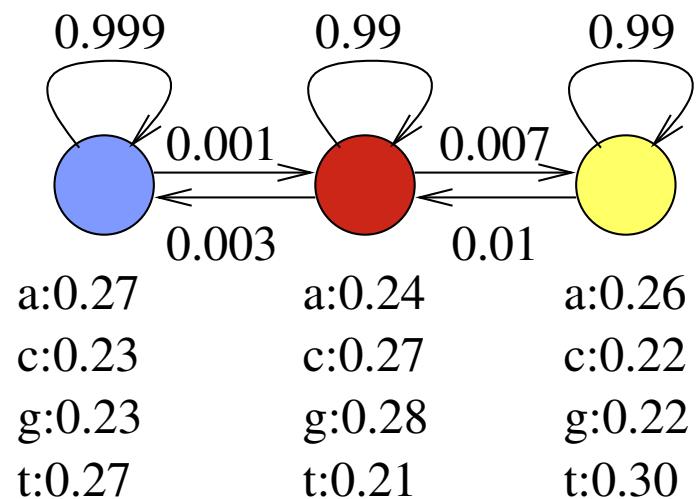
Broňa Brejová

26.10.2023

Opakovanie: HMM (skrytý Markovov model)



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



Predpokladajme, že model vždy začína v modrom stave.

$$\Pr(\text{a} \text{c} \text{a} \text{g}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 \cdot 0.99 \cdot 0.28 = 4.8 \cdot 10^{-6}$$

$$\Pr(\text{a} \text{c} \text{a} \text{g}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 \cdot 0.999 \cdot 0.23 = 0.0038$$

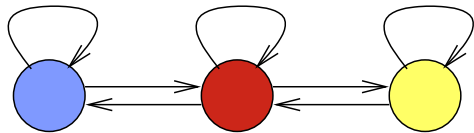
Iný hračkársky príklad: počasie

- Obdobie nízkeho tlaku vzduchu: väčšinou prší
- Obdobie nízkeho tlaku vzduchu: väčšinou slnečno

Každé obdobie trvá typicky niekoľko dní

Cvičenie: reprezentuj ako HMM

Parametre HMM (označenie)



Sekvencia $S = S_1, \dots, S_n$







Anotácia $A = A_1, \dots, A_n$




Parametre modelu:

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

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

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

a			
	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:

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

Viterbiho algoritmus

Pre dané HMM a sekvenciu S

nájdí najpravdepodobnejšiu anotáciu (postupnosť stavov)

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

Ako by ste to riešili?

Pripomeňme si príklad:

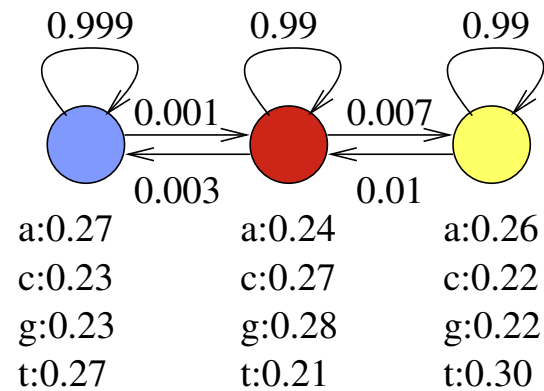
$$\Pr(\text{acaag}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 \cdot 0.99 \cdot 0.28 = 4.8 \cdot 10^{-6}$$




$$\Pr(\text{aacaag}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 \cdot 0.999 \cdot 0.23 = 0.0038$$

Viterbiho algoritmus

Nájdí najpravdepodobnejšiu postupnosť stavov $A = \arg \max_A \Pr(A, S)$

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



$V[u, i]$	a	c	a	g
				
				
				

Viterbiho algoritmus

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

Rekurencia?

$$V[u, 1] =$$

$$V[u, i] =$$

Pripomeňme si označenie:

Sekvencia $S = S_1, \dots, S_n$, anotácia (stavy) $A = A_1, \dots, A_n$

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

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

Viterbiho algoritmus

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

Rekurencia:

$$V[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$V[u, i] = \max_w V[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Algoritmus, celková odpoveď, čas výpočtu?

Pripomeňme si označenie:

Sekvencia $S = S_1, \dots, S_n$, anotácia (stavy) $A = A_1, \dots, A_n$

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

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

Viterbiho algoritmus (zhrnutie)

Nájdí najpravdepodobnejšiu postupnosť stavov $A = \arg \max_A \Pr(A, S)$

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

Rekurencia:

$$V[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$V[u, i] = \max_w V[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Algoritmus:

Inicializuj $V[*, 1]$

for $i = 2 \dots n$ (n =dĺžka S)

 for $u = 1 \dots m$ (m =počet stavov)

 vypočítaj $V[u, i]$, ulož najlepšie w do $B[u, i]$

Maximálne $V[u, n]$ cez všetky u je $\max_A \Pr(A, S)$

Cestu nájdí odzadu pomocou matice B

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

Další problém: celková pravdepodobnosť S

Viterbi počíta $\arg \max_A \Pr(A, S)$

Teraz chceme celkovú pravdepodobnosť, že vygenerujeme sekvenciu S

$$\text{t.j. } \Pr(S) = \sum_A \Pr(A, S)$$

Užitočné napr. na porovnávanie rôznych modelov,

ktorý má väčšiu šancu vygenerovať S

Ako by ste to počítali?

Pripomeňme si príklad:

$$\Pr(\text{acaag}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 \cdot 0.99 \cdot 0.28 = 4.8 \cdot 10^{-6}$$

$$\Pr(\text{aacaag}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 \cdot 0.999 \cdot 0.23 = 0.0038$$

Dopredný algoritmus (forward algorithm)

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S ,

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

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

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

Rekurencia?

$$F[u, 1] =$$

$$F[u, i] =$$

Pripomeňme si rekurenciu z Viterbiho:

$$V[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$V[u, i] = \max_w V[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Dopredný algoritmus (forward algorithm)

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S ,

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

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

Rekurencia

$$F[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$F[u, i] = \sum_w F[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Pripomeňme si rekurenciu z Viterbiho:

$$V[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$V[u, i] = \max_w V[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Dopredný algoritmus (forward algorithm)

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S ,

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

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

Rekurencia

$$F[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$F[u, i] = \sum_w F[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Výsledok?

Celková pravdepodobnosť $\Pr(S) =$

Čas výpočtu?

Dopredný algoritmus (forward algorithm)

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S ,

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

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

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

Výsledok

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

Čas výpočtu $O(nm^2)$

Tretí problem: pravdepodobnosť, že S_i bolo generované v stave u

$$\Pr(A_i = u \mid S) = \frac{\Pr(A_i=u, S)}{\Pr(S)}$$

$$\Pr(A_i = u, S) = \sum_{A: A_i=u} \Pr(A, S)$$

Vypočítame kombináciou dopredného a spätného algoritmu

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

$B[u, i]$: pravdepodobnosť, že ak začneme v u na pozícii i , tak vygenerujeme $S_{i+1} \dots, S_n$ v najbližších krokoch

$$\Pr(A_i = u, S) = F[u, i] \cdot B[u, i]$$

Spätný algoritmus (backward algorithm)

Dopredný algoritmus: pravdepodobnosť, že po i krokoch vygenerujeme S_1, S_2, \dots, S_i a dostaneme sa do stavu u .

$$F[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$F[u, i] = \sum_w F[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Spätný algoritmus: $B[u, i]$: pravdepodobnosť, že ak začneme v u na pozícii i , tak vygenerujeme $S_{i+1} \dots, S_n$ v najbližších krokoch

Ako spočítať $B[u, i]$?

Spätný algoritmus (backward algorithm)

Dopredný algoritmus: pravdepodobnosť, že po i krokoch vygenerujeme S_1, S_2, \dots, S_i a dostaneme sa do stavu u .

$$F[u, 1] = \pi(u) \cdot e(u, S_1)$$

$$F[u, i] = \sum_w F[w, i - 1] \cdot a(w, u) \cdot e(u, S_i)$$

Spätný algoritmus: $B[u, i]$: pravdepodobnosť, že ak začneme v u na pozícii i , tak vygenerujeme $S_{i+1} \dots, S_n$ v najbližších krokoch

$$B[u, n] = 1$$

$$B[u, i] = \sum_w B[w, i + 1] \cdot a(u, w) \cdot e(w, S_{i+1})$$

Cvičenie: Ako spočítať $\Pr(S)$ pomocou matice B ?

Aposteriórne dekódovanie (posterior decoding)

Videli sme: $\Pr(A_i = u \mid S) = \frac{F[u,i] \cdot B[u,i]}{\Pr(S)}$

Aposteriórne pravdepodobnosti stavov:

Použitím dopredného a spätného alg. vieme teda spočítať

$\Pr(A_i = u \mid S)$ pre všetky u a i v celkovom čase $O(nm^2)$

Aposteriórne dekódovanie

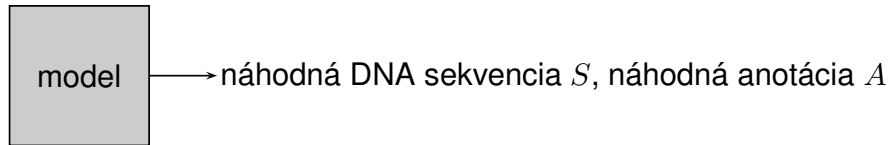
Pre dané S zvolíme A také že $A_i = \max_u \Pr(A_i = u \mid S)$

Výhoda: Berie do úvahy suboptimálne postupnosti stavov

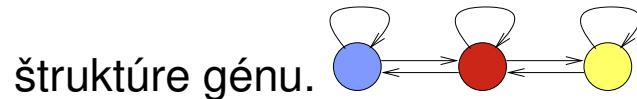
Nevýhoda: $\Pr(A \mid S)$ môže byť 0 alebo veľmi nízka

Iná možnosť: zvolíme A Viterbiho algoritmom, aposteriórne pravdepodobnosti použijeme na priradenie dôveryhodnosti jednotlivým častiam A

Hľadanie génov s HMM



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



- **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 nájdí najpravdepodobnejšiu anotáciu $A = \arg \max_A \Pr(A|S)$ Viterbiho algoritmom v $O(nm^2)$

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

$(S^{(1)}, A^{(1)}), (S^{(2)}, A^{(2)}), \dots$

Cieľ: nastaviť $\pi(u)$, $e(u, x)$, $a(u, v)$ 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

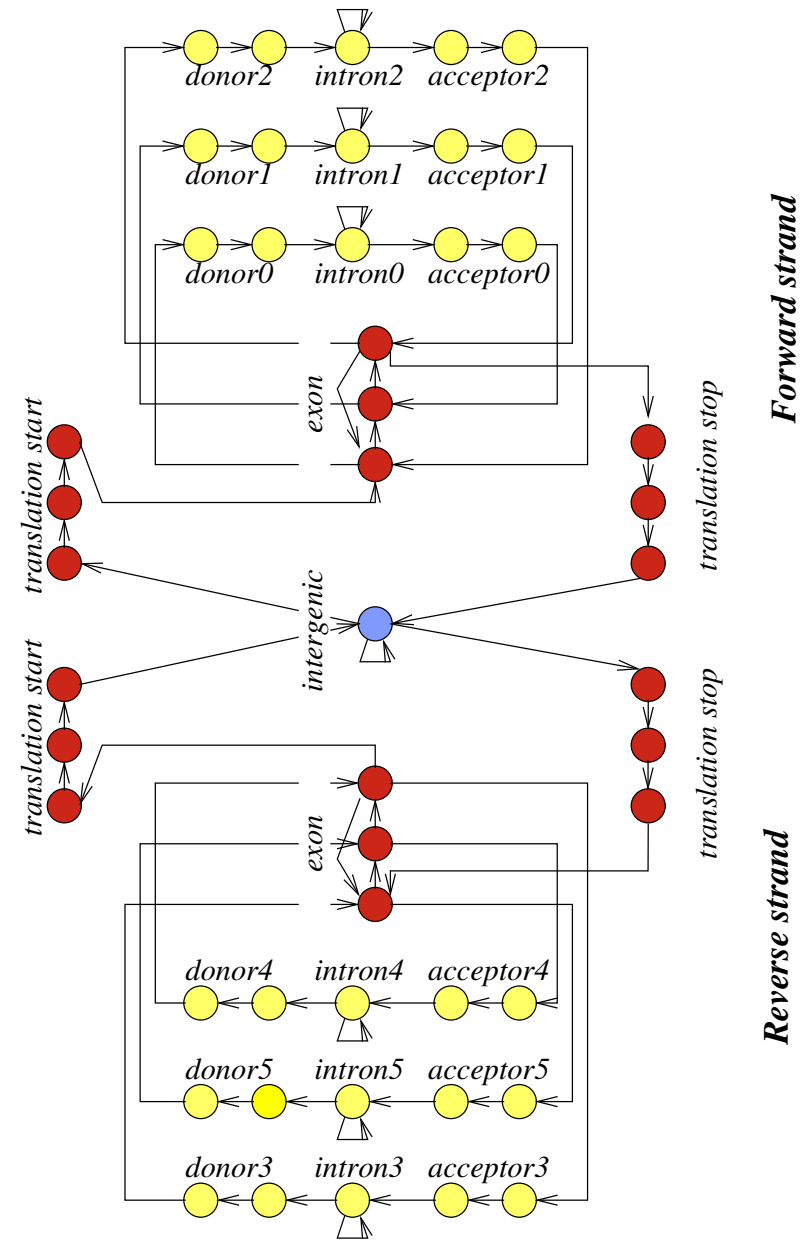
Cieľ: nastaviť $\pi(u)$, $e(u, x)$, $a(u, v)$ tak, aby $\prod_i \Pr(S^{(i)})$ bola čo najväčšia

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

V každej iterácii používa dopradný a spätný algoritmus.

Tvorba stavového priestoru modelu

Príklad HMM na hľadanie génov



Motif finding, EM algorithm, Gibbs sampling

Askar Gafurov

November 30, 2023

Motifs

- Motivation: DNA binding sites for a certain protein
 - ▶ E.g. RNA polymerase (in gene expression)
- The protein prefers some locations on DNA, but not a unique sequence
 - ▶ E.g. AATATACC, but also AGTATACG or CATATCTC
 - ▶ The probability of binding is not constant, some sequences are more likely to bind
- Motif = a table of probabilities for each position of a binding site

$$W = \begin{pmatrix} A : & 0.7 & 0.7 & 0.05 & 0.89 & 0.05 & 0.82 & 0.1 & 0.01 \\ C : & 0.2 & 0.05 & 0.05 & 0.01 & 0.05 & 0.1 & 0.8 & 0.8 \\ G : & 0.05 & 0.2 & 0.05 & 0.05 & 0.10 & 0.03 & 0.05 & 0.1 \\ T : & 0.05 & 0.05 & 0.85 & 0.05 & 0.80 & 0.05 & 0.05 & 0.09 \end{pmatrix}$$

- $\Pr[\text{AGTATACG is binding} \mid W] = 0.7 \cdot 0.2 \cdot 0.85 \cdot 0.89 \cdot 0.80 \cdot 0.82 \cdot 0.8 \cdot 0.1 \approx 0.006$
- $\Pr[\text{AATATACC is binding} \mid W] = 0.7 \cdot 0.7 \cdot 0.85 \cdot 0.89 \cdot 0.80 \cdot 0.82 \cdot 0.8 \cdot 0.8 \approx 0.156$

Generative model of a sequence with a motif

- Goal: define $\Pr[S \mid O, W]$ and $\Pr[O \mid W]$
- $\Pr[O \mid W]$ is easy: binding is equally likely to occur at every position (if we don't know the sequence)

- ▶ $\Pr[O \mid W] := \frac{1}{m - L + 1}$, where $m = |S|$

- $\Pr[S \mid O, W]$ is a bit tricky.

- ▶ We already know the prob. of letters at binding positions ($O, O + 1, \dots, O + L - 1$)
 - ▶ Assign *background frequency* $q(\cdot)$ for letters outside the binding site
 - ★ e.g. $q(A) = q(T) = 0.3, q(C) = q(G) = 0.2$
 - ▶ Now, the prob. of observing S is a product of probs. for each letter:

$$\begin{aligned}\Pr[S = \text{CCTATTGTATACCTATACC} \mid O = 6, W] &= \\ &= q(C)q(C)q(T)q(A)q(T) \cdot \\ &\cdot W[T, 1]W[G, 2]W[T, 3]W[A, 4]W[T, 5]W[A, 6]W[C, 7]W[C, 8] \cdot \\ &\cdot q(T)q(A)q(T)q(A)q(C)q(C) \approx \\ &\approx 1.11 \times 10^{-9}\end{aligned}$$

Motif in a larger sequence

- Question: Given a motif $W \in \mathbb{R}^{4 \times L}$ and sequence $S = CCTATTGTATACCTATACC$, where does the binding site start?
 - ▶ Assume there is exactly one binding site
 - ▶ Let's denote the binding site start as O .
- We can compute $\Pr[O \mid S, W]$ for each possible value of O .

$$\Pr[O \mid S, W] \stackrel{\text{Bayes}}{=} \frac{\Pr[S \mid O, W] \cdot \Pr[O \mid W]}{\sum_{O'} \Pr[S \mid O', W] \cdot \Pr[O' \mid W]}$$

- In human words: *compute prob. of observing S given start O and motif W for each value of O , and then normalize them to sum up to 1.*
- Notation for the (eventual) renormalization:

$$\Pr[O \mid S, W] \sim \Pr[S \mid O, W] \cdot \Pr[O \mid W]$$

Motif in a larger sequence

- Now let's compute $\Pr[S = \text{CCTATTGTATACCTATACC} \mid O, W]$ for each value of O and then renormalize it:

$$\begin{aligned}\Pr[S|O = 1, W] &\approx 4.76 \times 10^{-13} \\ \Pr[S|O = 2, W] &\approx 7.00 \times 10^{-17} \\ \Pr[S|O = 3, W] &\approx 1.04 \times 10^{-14} \\ \Pr[S|O = 4, W] &\approx 7.59 \times 10^{-14} \\ \Pr[S|O = 5, W] &\approx 2.92 \times 10^{-16} \\ \Pr[S|O = 6, W] &\approx 1.11 \times 10^{-09} \\ \Pr[S|O = 7, W] &\approx 7.87 \times 10^{-16} \\ \Pr[S|O = 8, W] &\approx 1.54 \times 10^{-14} \\ \Pr[S|O = 9, W] &\approx 9.19 \times 10^{-17} \\ \Pr[S|O = 10, W] &\approx 1.34 \times 10^{-15} \\ \Pr[S|O = 11, W] &\approx 6.12 \times 10^{-15} \\ \Pr[S|O = 12, W] &\approx 1.67 \times 10^{-09} \\ \Sigma &\approx 2.78 \times 10^{-09}\end{aligned}$$

$$\begin{aligned}\Pr[O = 1 \mid S, W] &\approx 4.76 \times 10^{-13} / (2.78 \times 10^{-09}) \approx 0.00017 \\ \Pr[O = 2 \mid S, W] &\approx 7.00 \times 10^{-17} / (2.78 \times 10^{-09}) \approx 0.00000 \\ \Pr[O = 3 \mid S, W] &\approx 1.04 \times 10^{-14} / (2.78 \times 10^{-09}) \approx 0.00000 \\ \Pr[O = 4 \mid S, W] &\approx 7.59 \times 10^{-14} / (2.78 \times 10^{-09}) \approx 0.00003 \\ \Pr[O = 5 \mid S, W] &\approx 2.92 \times 10^{-16} / (2.78 \times 10^{-09}) \approx 0.00000 \\ \Pr[O = 6 \mid S, W] &\approx 1.11 \times 10^{-09} / (2.78 \times 10^{-09}) \approx 0.39992 \\ \Pr[O = 7 \mid S, W] &\approx 7.87 \times 10^{-16} / (2.78 \times 10^{-09}) \approx 0.00000 \\ \Pr[O = 8 \mid S, W] &\approx 1.54 \times 10^{-14} / (2.78 \times 10^{-09}) \approx 0.00001 \\ \Pr[O = 9 \mid S, W] &\approx 9.19 \times 10^{-17} / (2.78 \times 10^{-09}) \approx 0.00000 \\ \Pr[O = 10 \mid S, W] &\approx 1.34 \times 10^{-15} / (2.78 \times 10^{-09}) \approx 0.00000 \\ \Pr[O = 11 \mid S, W] &\approx 6.12 \times 10^{-15} / (2.78 \times 10^{-09}) \approx 0.00000 \\ \Pr[O = 12 \mid S, W] &\approx 1.67 \times 10^{-09} / (2.78 \times 10^{-09}) \approx 0.59987\end{aligned}$$

Quick summary so far

- Motif W = a table of letter probabilities at each position of the site
 - ▶ $W[a, j] := \Pr[j\text{-th letter of a site is letter } a]$
- Probability of sequence S with a site at position O , motif $W \in \mathbb{R}^{4 \times L}$ and background letter frequency q is computed as a product:
 - ▶ $\Pr[S \mid O, W] = \prod_{j=1}^{O-1} q(S[j]) \cdot \prod_{j=1}^L W[S[O+j-1], j] \cdot \prod_{j=O+L}^m q(S[j])$
- Probability of site being at position O of sequence S , motif W and b.f. q is computed by renormalizing $\Pr[S \mid O, W]$ (assuming a unique occurrence):
 - ▶ $\Pr[O \mid S, W] \sim \Pr[S \mid O, W]$

Motif finding with known O (Two hands)

- Task: Given a vector of sequences $\mathbf{S} = (S_1, \dots, S_n)$ of length m each, a vector of site starts $\mathbf{O} = (O_1, \dots, O_n)$ and b.f. q , find the *best* motif W of length L !
 - ▶ Assuming that the motif **occurs exactly once** in each sequence
- Example of input data (the sites are shown as red text):

```
CGACTAAACACGGA
AGATATAACAAAAAG
AAGTCACCATAAACT
AGTATTCTATAGCA
TGACACATACCATGG
TAATATACCGCTTAC
TGCTAATAGTCCATA
TAATATACCGTATCT
```

Motif finding with known \mathbf{O} (Two hands)

- *best* = most likely = with the maximum (log-)likelihood
 - ▶ Likelihood of W is $\mathcal{L}(W; \mathbf{O}, \mathbf{S}) \stackrel{\text{def.}}{=} \Pr[\mathbf{S}, \mathbf{O} \mid W]$
 - ▶ $W^* = \arg \max_{W \in \mathcal{W}} \mathcal{L}(W; \mathbf{O}, \mathbf{S}) = \arg \max_{W \in \mathcal{W}} \ln \mathcal{L}(W; \mathbf{O}, \mathbf{S})$
- Intuition: best W is obtained by counting letter frequencies at the sites

$$\text{▶ } W^*[a, j] \stackrel{??}{=} \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{S_i[O_i+j-1]=a} =: \frac{\#_{a,j}(\mathbf{O})}{n}$$

CGACTAAACACGGA

AGATATAACAAAAAG

AAGTCACCATAACT

AGTATTCCTATAGCA

TGACACATACCATGG

TAAATATACCGCTTAC

TGCTAATAGTCCATA

TAAATATACCGTATCT

$$W_{\text{counting}} = \begin{pmatrix} 5/8 & 6/8 & 0/8 & 8/8 & 2/8 & 6/8 & 1/8 & 0/8 \\ 2/8 & 1/8 & 1/8 & 0/8 & 0/8 & 0/8 & 7/8 & 7/8 \\ 1/8 & 1/8 & 0/8 & 0/8 & 1/8 & 0/8 & 0/8 & 0/8 \\ 0/8 & 0/8 & 7/8 & 0/8 & 5/8 & 2/8 & 0/8 & 1/8 \end{pmatrix}$$

Let's check the intuition

$$\begin{aligned} W^* &:= \arg \max_{W \in \mathcal{W}} \ln \mathcal{L}(W; \mathbf{O}, \mathbf{S}) \stackrel{\text{def.}}{=} \arg \max_{W \in \mathcal{W}} \ln \Pr[\mathbf{S}, \mathbf{O} \mid W] = \\ &= \arg \max_{W \in \mathcal{W}} \ln \Pr[\mathbf{S} \mid \mathbf{O}, W] + \ln \Pr[\mathbf{O} \mid W] = \\ &= \arg \max_{W \in \mathcal{W}} \sum_{i=1}^n \ln \Pr[S_i \mid O_i, W] = \\ &= \arg \max_{W \in \mathcal{W}} \sum_{i=1}^n \left(\sum_{j=1}^{O_i-1} \ln q(S_i[j]) + \sum_{j=1}^L \ln W[S_i[O_i + j - 1], j] + \sum_{j=O_i+L}^m \ln q(S_i[j]) \right) = \\ &= \arg \max_{W \in \mathcal{W}} \sum_{j=1}^L \sum_{a \in \{A, C, G, T\}} \ln W[a, j] \cdot \sum_{i=1}^n \mathbf{1}_{S_i[O_i+j-1]=a} = \\ &= \arg \max_{W \in \mathcal{W}} \sum_{j=1}^L \sum_{a \in \{A, C, G, T\}} \ln W[a, j] \cdot \#_{a,j}(\mathbf{O}) \end{aligned}$$



Let's check the intuition, p.2

- $W^* = \arg \max_{W \in \mathcal{W}} \sum_{j=1}^L \sum_{a \in \{A, C, G, T\}} \ln W[a, j] \cdot \#_{a, j}(\mathbf{O})$
- Each column of W can be optimised independently:
 - ▶ $W_j^* = \arg \max_{\substack{x_A, x_C, x_G, x_T \geq 0 \\ \sum x_a = 1}} \sum_{a \in \{A, C, G, T\}} \#_{a, j}(\mathbf{O}) \ln x_a$
- Using the method of Lagrange multipliers, we obtain $x_a^* = \frac{\#_{a, j}(\mathbf{O})}{\sum_a \#_{a, j}(\mathbf{O})} = \frac{\#_{a, j}(\mathbf{O})}{n}$
- Thus indeed $W^*[a, j] = \frac{\#_{a, j}(\mathbf{O})}{n}$

Method of Lagrange multipliers

- Optimisation task: $\arg \max_{\substack{x_1, \dots, x_4 \geq 0 \\ \sum_i x_i = 1}} \sum_{i=1}^4 a_i \ln x_i$
- Define new function $T(x_1, \dots, x_4, \lambda) := \sum_{i=1}^4 a_i \ln x_i + \lambda \cdot \left(-1 + \sum_{i=1}^4 x_i\right)$
- Solve (unconstrained) optimisation task $\arg \max_{x_1, \dots, x_4, \lambda \in \mathbf{R}} T$ e.g. by setting the gradient to zero:

$$\nabla T = \left(\frac{a_1}{x_1} + \lambda, \dots, \frac{a_4}{x_4} + \lambda, -1 + \sum_{i=1}^4 x_i \right)$$

$$\nabla T = 0 \implies x_i = \frac{-a_i}{\lambda}, \sum_{i=1}^4 x_i = 1$$

$$\implies \sum_{i=1}^4 \frac{-a_i}{\lambda} = 1 \implies \lambda = - \sum_{i=1}^4 a_i$$

$$\implies x_i = \frac{a_i}{\sum_{i=1}^4 a_i}$$



Quick summary so far

- Previous:

- ▶ $\Pr[S \mid O, W] = \prod_{j=1}^{O-1} q(S[j]) \cdot \prod_{j=1}^L W[S[O+j-1], j] \cdot \prod_{j=O+L}^m q(S[j])$
 - ▶ $\Pr[O \mid S, W] \sim \Pr[S \mid O, W]$

- (new!) Given sequences **S** and motif starts **O**, we can find the most likely motif **W** of length L using letter frequency counting

- ▶ $W^* = \arg \max_{W \in \mathcal{W}} \ln \Pr[\mathbf{S} \mid \mathbf{O}, W] = \left(\frac{\#_{a,j}(\mathbf{O})}{n} \right)_{a \in \{A, C, G, T\}, 1 \leq j \leq L}$

Motif finding with distribution of O (One hand)

- Task: Given a vector of sequences $\mathbf{S} = (S_1, \dots, S_n)$ of length m each, a **distribution of site starts** $g(\mathbf{O}) = \prod_{i=1}^n g_i(O_i)$ and b.f. q , find the best motif W of length L !
 - Assuming that the motif **occurs exactly once** in each sequence
 - In human words: we don't know exactly where the motif starts occur, but we have a guess g_i for each sequence.
 - The values $g_i(1), \dots, g_i(m - L + 1)$ are “weights” for each position in sequence S_i
- Example of input data:

CGACTAAACCACGGA
AGATATAACAAAAG
AAGTCACCATAAACT
AGTATTCCTATAGCA
TGACACATACCATGG
TAATATACCGCTTAC
TGCTAATAGTCCATA
TAATATACCGTATCT

$$(g_i(o))_{i,o} = \begin{pmatrix} 0.07 & 0.07 & 0.51 & 0.07 & 0.07 & 0.07 & 0.07 & 0.07 \\ 0.07 & 0.51 & 0.07 & 0.07 & 0.07 & 0.07 & 0.07 & 0.07 \\ 0.12 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.40 \\ 0.46 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.09 & 0.08 \\ 0.07 & 0.07 & 0.07 & 0.51 & 0.07 & 0.07 & 0.07 & 0.07 \\ 0.07 & 0.51 & 0.07 & 0.07 & 0.07 & 0.07 & 0.07 & 0.07 \\ 0.07 & 0.07 & 0.07 & 0.07 & 0.51 & 0.07 & 0.07 & 0.07 \\ 0.07 & 0.51 & 0.07 & 0.07 & 0.07 & 0.07 & 0.07 & 0.07 \end{pmatrix}$$

Motif finding with distribution of \mathbf{O} (One hand)

- *best* = with the maximum log-likelihood across all possible \mathbf{O} with prob. g
 - ▶ $W^* = \arg \max_{W \in \mathcal{W}} E_{\mathbf{O} \sim g} [\ln \mathcal{L}(W; \mathbf{O}, \mathbf{S})] \stackrel{\text{def.}}{=} \sum_{\mathbf{O} \in \mathcal{O}} \ln \mathcal{L}(W; \mathbf{O}, \mathbf{S}) \cdot g(\mathbf{O})$
- Intuition: best W is obtained by counting letter frequencies at all possible sites weighted by g

- ▶ $W^*[a, j] \stackrel{??}{=} \frac{1}{n} \sum_{i=1}^n \sum_{O_i=1}^{m-L+1} g_i(O_i) \cdot \mathbf{1}_{S_i[O_i+j-1]=a} =: \frac{\#_{a,j}(g)}{n}$

Example of weighted frequency counting

- Looking for a motif of length $L = 3$

AAACCT

- Input: $\mathbf{S} = \text{ACGACA}$, distribution of starts $g = \begin{pmatrix} 0.2 & 0.3 & 0.4 & 0.1 \\ 0.4 & 0.1 & 0.2 & 0.3 \\ 0.1 & 0.1 & 0.6 & 0.2 \end{pmatrix}$

TTACCG

$$\# : \begin{pmatrix} A : & 0.2 + 0.3 + 0.4 + 0.4 + 0.3 + 0.6 & 0.2 + 0.3 + 0.2 + 0.1 & 0.2 + 0.1 + 0.3 + 0.1 \\ C : & 0.1 + 0.1 + 0.2 & 0.4 + 0.1 + 0.4 + 0.3 + 0.6 + 0.2 & 0.3 + 0.4 + 0.2 + 0.1 + 0.6 \\ G : & 0.2 & 0.1 & 0.4 + 0.2 \\ T : & 0.1 & 0.1 & 0.1 \end{pmatrix}$$

$$W_{\text{counting}} = \begin{pmatrix} A : & 2.2/3 & 0.8/3 & 0.7/3 \\ C : & 0.5/3 & 2.0/3 & 1.6/3 \\ G : & 0.2/3 & 0.1/3 & 0.6/3 \\ T : & 0.1/3 & 0.1/3 & 0.2/3 \end{pmatrix}$$

Let's check the intuition

$$\begin{aligned}
 W^* &:= \arg \max_{W \in \mathcal{W}} E_{\mathbf{O} \sim g} [\ln \mathcal{L}(W; \mathbf{O}, \mathbf{S})] = \arg \max_{W \in \mathcal{W}} E_{\mathbf{O} \sim g} \left[\ln \prod_{i=1}^n \mathcal{L}(W; O_i, S_i) \right] = \\
 &= \arg \max_{W \in \mathcal{W}} \sum_{i=1}^n E_{\mathbf{O}_i \sim g_i} [\ln \mathcal{L}(W; O_i, S_i)] = \\
 &= \arg \max_{W \in \mathcal{W}} \sum_{i=1}^n \sum_{O_i=1}^{m-L+1} (\ln \Pr[S_i \mid O_i, W] + \ln \Pr[O_i \mid W]) g_i(O_i) = \\
 &= \arg \max_{W \in \mathcal{W}} \sum_{i=1}^n \sum_{O_i=1}^{m-L+1} \left(\sum_{j=1}^{O_i-1} \ln q(S_i[j]) + \sum_{j=1}^L \ln W[S_i[O_i+j-1], j] + \sum_{j=O_i+L}^m \ln q(S_i[j]) \right) g_i(O_i) = \\
 &= \arg \max_{W \in \mathcal{W}} \sum_{j=1}^L \sum_{a \in \{A, C, G, T\}} \ln W[a, j] \left(\sum_{i=1}^n \sum_{O_i=1}^{m-L+1} \mathbf{1}_{S_i[O_i+j-1]=a} \cdot g_i(O_i) \right) = \\
 &= \arg \max_{W \in \mathcal{W}} \sum_{j=1}^L \sum_{a \in \{A, C, G, T\}} \ln W[a, j] \cdot \#_{a,j}(g).
 \end{aligned}$$



Quick summary so far

- Previous:

- ▶ $\Pr[S \mid O, W] = \prod_{j=1}^{O-1} q(S[j]) \cdot \prod_{j=1}^L W[S[O+j-1], j] \cdot \prod_{j=O+L}^m q(S[j])$

- ▶ $\Pr[O \mid S, W] \sim \Pr[S \mid O, W]$

- ▶ For known **S** and starts **O**:

$$W^* = \arg \max_{W \in \mathcal{W}} \ln \mathcal{L}(W; \mathbf{S}, \mathbf{O}) = (n^{-1} \cdot \#_{a,j}(\mathbf{O}))_{a \in \{A,C,G,T\}, 1 \leq j \leq L}$$

$$\star \#_{a,j}(\mathbf{O}) := \sum_{i=1}^n \mathbf{1}_{S_i[O_i+j-1]=a}$$

- (new!) Given sequences **S** and motif starts distribution $g(\mathbf{O})$, we can find the most likely motif W of length L using **weighted** letter frequency counting:

- ▶ $W^* = \arg \max_{W \in \mathcal{W}} E_{\mathbf{O} \sim g} [\ln \mathcal{L}(W; \mathbf{S}, \mathbf{O})] = (n^{-1} \cdot \#_{a,j}(g))_{a \in \{A,C,G,T\}, 1 \leq j \leq L}$

- ▶ $\#_{a,j}(g) := \sum_{i=1}^n \sum_{O_i=1}^{m-L+1} g_i(O_i) \cdot \mathbf{1}_{S_i[O_i+j-1]=a}$

Motif finding without O (No hands)

- Task: Given a vector of sequences $\mathbf{S} = (S_1, \dots, S_n)$ of length m each and b.f. q , find the *best* motif W of length L !
 - ▶ Assuming that the motif **occurs exactly once** in each sequence
 - ▶ No information about the motif starts nor motif itself...

- Example of input data:

```
CGACTAAACACGGA
AGATATAACAAAAG
AAGTCACCATAAACT
AGTATTCCTATAGCA
TGACACATACCATGG
TAATATACCGCTTAC
TGCTAATAGTCCATA
TAATATACCGTATCT
```

Expectation-Maximisation algorithm

- The algorithm:

- ▶ Start with random motif $W^{(0)}$
- ▶ Repeat:
 - ★ (E-step) infer $g^{(t+1)}(\cdot)$ from $W^{(t)}$:

$$g^{(t+1)}(\mathbf{O}) := \Pr[\mathbf{O} \mid \mathbf{S}, W^{(t)}] \sim \prod_{i=1}^n \Pr[S_i \mid O_i, W^{(t)}]$$

- ★ (M-step) infer $W^{(t+1)}$ from $g^{(t+1)}(\cdot)$:

$$W^{(t+1)} := \arg \max_{W \in \mathcal{W}} E_{\mathbf{O} \sim g^{(t+1)}} [\ln \mathcal{L}(W; \mathbf{S}, \mathbf{O})] = \left(n^{-1} \cdot \#_{a,j} \left(g^{(t+1)} \right) \right)_{a \in \{A, C, G, T\}, 1 \leq j \leq L}$$

- Random $\Rightarrow W^{(0)} \xRightarrow{E} g^{(1)} \xRightarrow{M} W^{(1)} \xRightarrow{E} g^{(2)} \xRightarrow{M} W^{(2)} \xRightarrow{E} g^{(3)} \xRightarrow{M} W^{(3)} \xRightarrow{E} \dots$
- Each next $W^{(t)}$ is better than the previous one:

$$\Pr[S \mid W^{(t+1)}] \geq \Pr[S \mid W^{(t)}]$$

Time for a demo

Reconstruction of a missing motif start

- Task: Given sequences \mathbf{S} , motif length L and motif starts with missing i -th coordinate $\mathbf{O}_{-i} = (O_1, \dots, O_{i-1}, ?, O_{i+1}, \dots, O_n)$, reconstruct the missing coordinate O_i .
 - ▶ We don't know the motif W
- Let's compute prob. for each possible value of O_i :

$$\begin{aligned}\Pr[O_i = k \mid \mathbf{O}_{-i}, \mathbf{S}] &\stackrel{\text{cond.}}{=} \frac{\Pr[O_i = k, \mathbf{O}_{-i} \mid \mathbf{S}]}{\sum_{\ell} \Pr[O_i = \ell, \mathbf{O}_{-i} \mid \mathbf{S}]} \sim \\ &\sim \Pr[\mathbf{O} = (O_1, \dots, O_{i-1}, k, O_{i+1}, \dots, O_n) \mid \mathbf{S}] = \\ &= \frac{\Pr[\mathbf{S} \mid \mathbf{O}] \cdot \Pr[\mathbf{O}]}{\sum_{\mathbf{O}'} \Pr[\mathbf{S} \mid \mathbf{O}'] \cdot \Pr[\mathbf{O}']} \sim \\ &\sim \Pr[\mathbf{S} \mid \mathbf{O}]\end{aligned}$$

- So, we only need to be able to compute $\Pr[\mathbf{S} \mid \mathbf{O}]$ for arbitrary $\mathbf{O} \dots$

How to compute $\Pr[\mathbf{S} \mid \mathbf{O}]$

- Since we don't know the true motif, we have to average over all possible motifs. It's called marginalization:

$$\Pr[\mathbf{S} \mid \mathbf{O}] = E_W[\Pr[\mathbf{S}, W \mid \mathbf{O}]] = \int_{\mathcal{W}} \Pr[\mathbf{S} \mid W, \mathbf{O}] \cdot p(W) dW$$

- We need to define the “probability” $p(W)$ of motif W .
- We want all combinations of frequencies to be “equally likely”:

$$\begin{aligned} W_j &\sim \text{Dirichlet}(1, 1, 1, 1) \\ p(W_j) &= \frac{\Gamma(4)}{\Gamma(1)^4} = \frac{3!}{0!^4} = 6 \\ p(W) &= 6^n \end{aligned}$$

Let's compute some integrals

$$\begin{aligned}
 \Pr[\mathbf{S} \mid \mathbf{O}] &= \int_{\mathcal{W}} \Pr[\mathbf{S} \mid W, \mathbf{O}] \cdot 6^n dW \sim \\
 &\sim \int_{\mathcal{W}} \Pr[\mathbf{S} \mid W, \mathbf{O}] dW = \int_{\mathcal{W}} \prod_{i=1}^n \Pr[S_i \mid W, O_i] dW = \\
 &= \int_{\mathcal{W}} \prod_{i=1}^n \prod_{j=1}^{O_i-1} q(S[j]) \cdot \prod_{j=1}^L W[S[O_i + j - 1], j] \cdot \prod_{j=O_i+L}^m q(S[j]) dW = \\
 &= \int_{\mathcal{W}} \prod_{i=1}^n \prod_{j=1}^L W[S[O_i + j - 1], j] dW = \\
 &= \int_{\mathcal{W}} \prod_{j=1}^L \prod_{a \in \{A, C, G, T\}} W[a, j]^{\#_{a,j}(\mathbf{O})} dW = \\
 &= \prod_{j=1}^L \int_{S^4} \prod_{a \in \{A, C, G, T\}} W[a, j]^{\#_{a,j}(\mathbf{O})} dW_j \dots
 \end{aligned}$$



Computation of integral over unit 4-simplex

- We need to compute a definite integral of form $\int_{S^4} x_1^{a_1} \cdot x_2^{a_2} \cdot x_3^{a_3} \cdot x_4^{a_4} d\mathbb{X}$, where $S^4 = \{(x_1, \dots, x_4) \in [0, 1]^4 : \sum_i x_i = 1\}$.

- A magic formula: $\mathcal{B}(z_1, z_2) \stackrel{\text{def.}}{=} \int_0^1 t^{z_1-1} \cdot (1-t)^{z_2-1} dt = \frac{\Gamma(z_1) \cdot \Gamma(z_2)}{\Gamma(z_1 + z_2)}$

$$\begin{aligned}
 & \int_{S^4} x_1^{a_1} \cdot x_2^{a_2} \cdot x_3^{a_3} \cdot x_4^{a_4} d\mathbb{X} = \\
 &= \int_0^1 x_1^{a_1} \int_0^{1-x_1} x_2^{a_2} \int_0^{1-x_1-x_2} x_3^{a_3} \cdot (1-x_1-x_2-x_3)^{a_4} dx_3 dx_2 dx_1 = \\
 &= \int_0^1 x_1^{a_1} \int_0^{1-x_1} x_2^{a_2} \int_0^{\xi} x_3^{a_3} \cdot (\xi - x_3)^{a_4} dx_3 dx_2 dx_1 = \\
 &= \int_0^1 x_1^{a_1} \int_0^{1-x_1} x_2^{a_2} \xi^{a_3+a_4} \int_0^{\xi} \left(\frac{x_3}{\xi}\right)^{a_3} \cdot \left(1 - \frac{x_3}{\xi}\right)^{a_4} dx_3 dx_2 dx_1 = \\
 &= \frac{\Gamma(a_3+1)\Gamma(a_4+1)}{\Gamma(a_3+a_4+2)} \int_0^1 x_1^{a_1} \int_0^{1-x_1} x_2^{a_2} (1-x_1-x_2)^{a_3+a_4+1} dx_2 dx_1 = \\
 &= \frac{\Gamma(a_3+1)\Gamma(a_4+1)}{\Gamma(a_3+a_4+2)} \frac{\Gamma(a_2+1)\Gamma(a_3+a_4+2)}{\Gamma(a_2+a_3+a_4+3)} \int_0^1 x_1^{a_1} \cdot (1-x_1)^{a_2+a_3+a_4+2} dx_1 = \\
 &= \frac{\Gamma(a_3+1)\Gamma(a_4+1)}{\Gamma(a_3+a_4+2)} \frac{\Gamma(a_2+1)\Gamma(a_3+a_4+2)}{\Gamma(a_2+a_3+a_4+3)} \frac{\Gamma(a_1+1)\Gamma(a_2+a_3+a_4+3)}{\Gamma(a_1+a_2+a_3+a_4+4)} = \\
 &= \frac{a_1!a_2!a_3!a_4!}{(3+a_1+a_2+a_3+a_4)!}.
 \end{aligned}$$

Back to the main integral

$$\begin{aligned}\Pr[\mathbf{S} \mid \mathbf{O}] &\sim \dots \sim \prod_{j=1}^L \int_{S^4} \prod_{a \in \{A, C, G, T\}} W[a, j]^{\#_{a,j}(\mathbf{O})} dW_j = \\ &= \prod_{j=1}^L \frac{\prod_{a \in \{A, C, G, T\}} \#_{a,j}(\mathbf{O})!}{(3+n)!} \sim \\ &\sim \prod_{j=1}^L \prod_{a \in \{A, C, G, T\}} \#_{a,j}(\mathbf{O})!\end{aligned}$$

Finally, we can compute the probabilities for the missing motif start:

$$\Pr[O_i = k \mid \mathbf{O}_{-i}, \mathbf{S}] \sim \prod_{j=1}^L \prod_{a \in \{A, C, G, T\}} \#_{a,j}(\mathbf{O})!$$

Hooray! Time to take a breath and regain posture.

Example of computing the prob. of missing start

- Formula: $\Pr[O_i = k \mid \mathbf{O}_{-i}, \mathbf{S}] \sim \prod_{j=1}^L \prod_{a \in \{A, C, G, T\}} \#_{a,j}(\mathbf{O})!$

AACCT

- Input: $\mathbf{S} = \text{ACGTCA}$, $\mathbf{O}_{-2} = (2, ?, 3)$

TTACCG

$$\Pr[O_2 = 1 \mid \mathbf{O}_{-2}, \mathbf{S}] \sim 3!0!0!0! \cdot 1!2!0!0! \cdot 0!2!1!0! = 24 \quad : \text{sites AAC, ACG, ACC}$$

$$\Pr[O_2 = 2 \mid \mathbf{O}_{-2}, \mathbf{S}] \sim 2!1!0!0! \cdot 1!1!1!0! \cdot 0!2!0!1! = 4 \quad : \text{sites AAC, CGT, ACC}$$

$$\Pr[O_2 = 3 \mid \mathbf{O}_{-2}, \mathbf{S}] \sim 2!0!1!0! \cdot 1!1!0!1! \cdot 0!3!0!0! = 12 \quad : \text{sites AAC, GTC, ACC}$$

$$\Pr[O_2 = 4 \mid \mathbf{O}_{-2}, \mathbf{S}] \sim 2!0!0!1! \cdot 1!2!0!0! \cdot 1!2!0!0! = 8 \quad : \text{sites AAC, TCA, ACC}$$

$$\Sigma = 24 + 4 + 12 + 8 = 48$$

$$\Pr[O_2 = 1 \mid \mathbf{O}_{-2}, \mathbf{S}] = 24/48 = 0.50$$

$$\Pr[O_2 = 2 \mid \mathbf{O}_{-2}, \mathbf{S}] = 4/48 = 0.08$$

$$\Pr[O_2 = 3 \mid \mathbf{O}_{-2}, \mathbf{S}] = 12/48 = 0.25$$

$$\Pr[O_2 = 4 \mid \mathbf{O}_{-2}, \mathbf{S}] = 8/48 = 0.17$$

Sampling from $\Pr[\mathbf{O} \mid \mathbf{S}]$ via Gibbs sampling algorithm

- A bigger goal: To sample motif starts \mathbf{O} from $\Pr[\mathbf{O} \mid \mathbf{S}]$
- Gibbs sampling algorithm:
 - ▶ Start with a random $\mathbf{O}^{(0)}$
 - ▶ Repeat:
 - ★ Select a random coordinate $i \in_R \{1, \dots, n\}$
 - ★ Erase i -th coordinate from $\mathbf{O}^{(t)}$
 - ★ Sample a replacement O' for it from $\Pr[O_i = k \mid \mathbf{O}_{-i}, \mathbf{S}] \sim \prod_{j=1}^L \prod_{a \in \{A, C, G, T\}} \#_{a,j}(\mathbf{O})!$
 - ★ new sample $\mathbf{O}^{(t+1)}$ is the same as $\mathbf{O}^{(t)}$, but with i -th coordinate replaced by O'
- This algorithm produces samples $O^{(0)}, O^{(1)}, O^{(2)}, O^{(3)}, \dots$ from $\Pr[\mathbf{O} \mid \mathbf{S}]$
- Example:

$$O^{(0)} = (\quad 1 \quad 3 \quad 2 \quad 1 \quad 10 \quad 6 \quad)$$

$$O^{(1)} = (\quad 1 \quad 3 \quad 2 \quad \color{red}{3} \quad 10 \quad 6 \quad)$$

$$O^{(2)} = (\quad 1 \quad 3 \quad 2 \quad 3 \quad \color{red}{7} \quad 6 \quad)$$

$$O^{(3)} = (\quad 1 \quad 3 \quad 2 \quad 3 \quad \color{red}{16} \quad 6 \quad)$$

$$O^{(4)} = (\quad 1 \quad \color{red}{5} \quad 2 \quad 3 \quad 16 \quad 6 \quad)$$

$$O^{(5)} = (\quad 1 \quad 5 \quad 2 \quad 3 \quad 16 \quad \color{red}{9} \quad)$$

...

Back to Motif finding without O (No hands)

- Task: Given a vector of sequences $\mathbf{S} = (S_1, \dots, S_n)$ of length m each and b.f. q , find the *best* motif W of length L !
- Algorithm using Gibbs sampling:
 - ▶ Sample many motif starts vectors $\mathbf{O}^{(1)}, \dots, \mathbf{O}^{(B)}$ from $\Pr[\mathbf{O} \mid \mathbf{S}]$
 - ▶ For each sampled motif starts vector $\mathbf{O}^{(t)}$, compute the optimal motif $W^{(t)} := \arg \max_{W \in \mathcal{W}} \ln \mathcal{L}(W; \mathbf{S}, \mathbf{O}^{(t)}) = (n^{-1} \cdot \#_{a,j}(\mathbf{O}^{(t)}))_{a \in \{A,C,G,T\}, 1 \leq j \leq L}$
 - ▶ Return the pair $(W^{(t)}, \mathbf{O}^{(t)})$ with the highest log-likelihood $\ln \mathcal{L}(W^{(t)}; \mathbf{S}, \mathbf{O}^{(t)})$

Time for a demo

Summary

- Motifs are used to represent e.g. DNA bind sites of proteins
 - ▶ Motif W = a table of letter probabilities at each position of the site
 - ★ $W[a, j] := \Pr[j\text{-th letter of a site is letter } a]$
- Motif finding = given sequences, where motif occurs, find the best motif W
- Motif finding can be solved by Expectation-Maximisation algorithm
 - ▶ Alternating improvement of $g(\mathbf{O})$ and W :
 - ★ $\text{Random} \Rightarrow W^{(0)} \xrightarrow{E} g^{(1)} \xrightarrow{M} W^{(1)} \xrightarrow{E} g^{(2)} \xrightarrow{M} W^{(2)} \xrightarrow{E} g^{(3)} \xrightarrow{M} W^{(3)} \xrightarrow{E} \dots$
- Motif finding can be solved by Gibbs sampling
 - ▶ Random sampling of \mathbf{O} from $\Pr[\mathbf{O} \mid \mathbf{S}]$, selecting the best one
 - ▶ Gibbs sampling works by altering one coordinate of a previous sample, sampling its value from a conditional distribution

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 available, 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, painsless
- 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, \dots, x_n

minimize $\sum_i a_i x_i$ for given weights a_1, \dots, 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 $w_1 \dots w_n$ and costs $c_1 \dots c_n$.

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 $S_1 \dots, S_n$ of a set $U = \{1 \dots m\}$.

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 alignments will not be scored

Protein threading, problem formulation

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

lengths of m cores $c_1 \dots c_m$,

scoring tables

- E_{ij} : how well $b_j \dots b_{j+c_i-1}$ agrees with sequence of core i ,
- E_{ijkl} : how well would cores i and k interact, if they start at pos. j, ℓ .

Task: choose starts of cores x_1, x_2, \dots, 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: