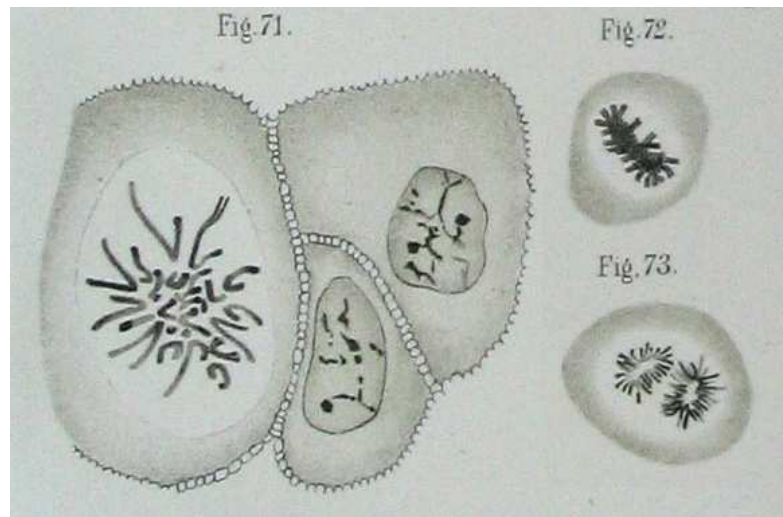


Biológia pre informatikov

Broňa Brejová

1.10.2015



Walther Flemming, 1881

Ďalšie informácie

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

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,

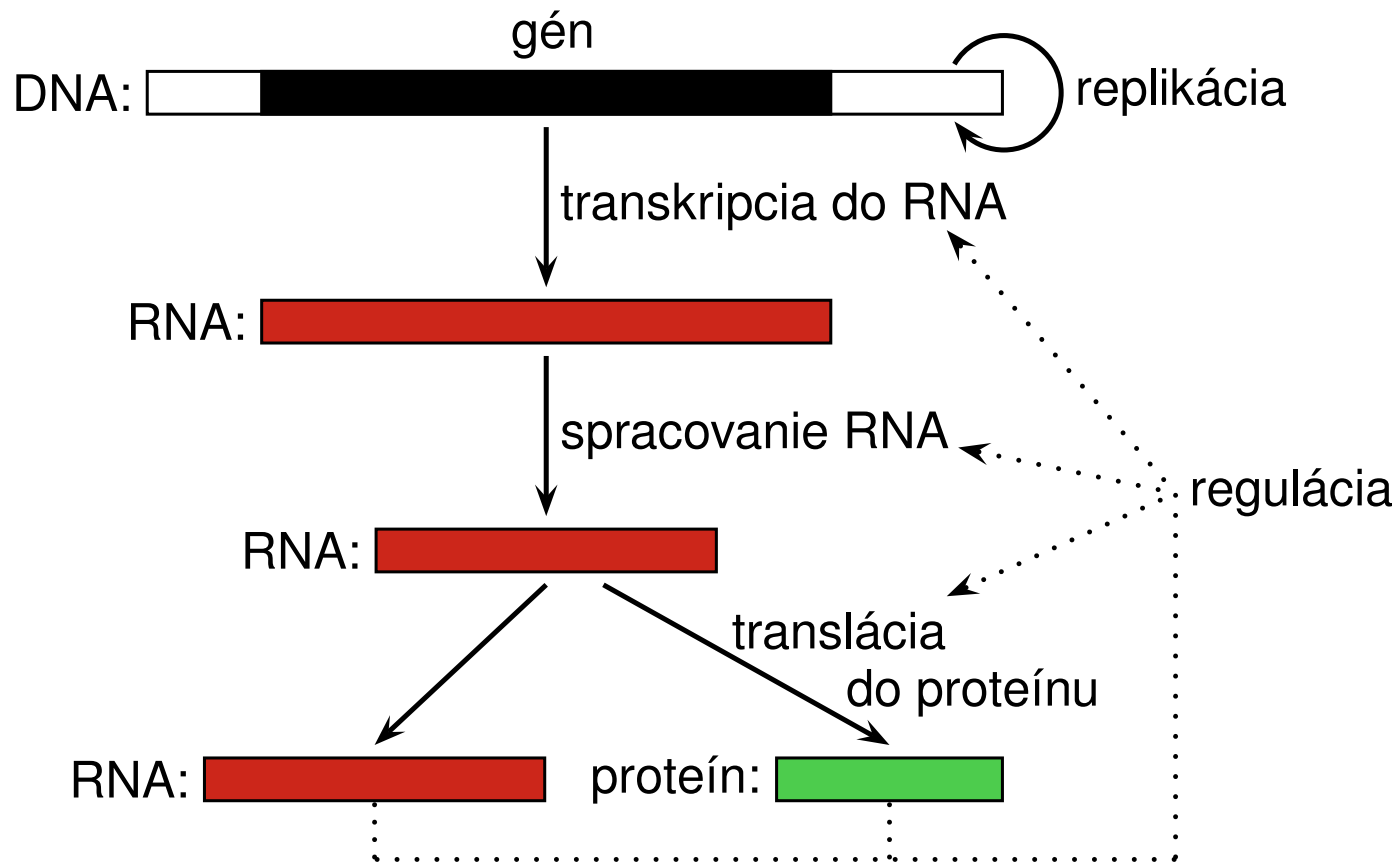
sú dôležité pre stavbu bunky a pohyb.

Reťazec aminokyselín (20 rôznych aminokyselín).

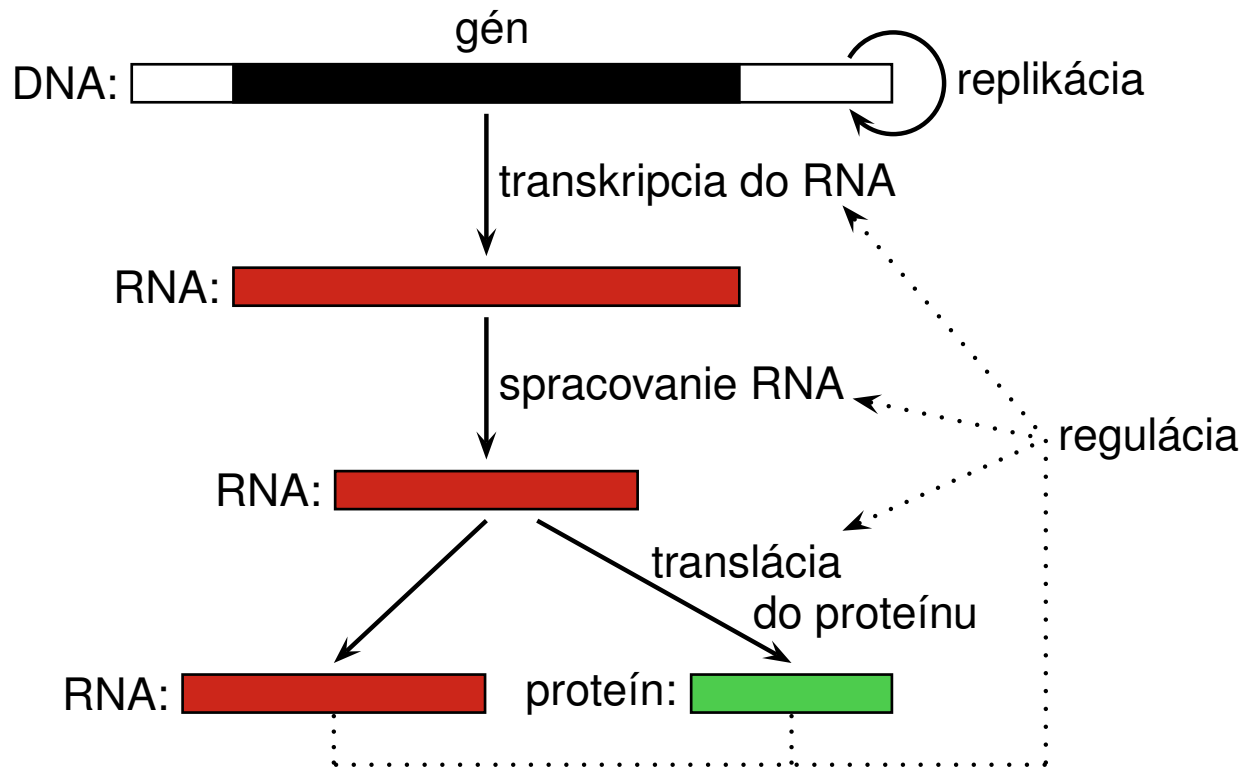
Aká informácia je uložená v DNA?

Gény: Predpisy na tvorbu proteínov a funkčných RNA molekúl.

Riadenie ich expresie: kedy a koľko sa má tvoriť.



Centrálne dogma (Francis Crick 1958,1970)



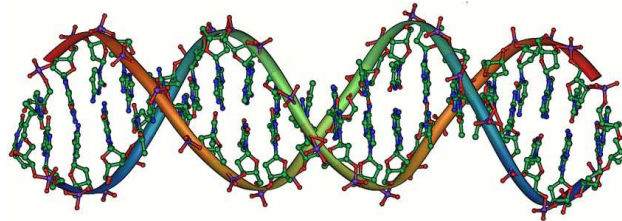
“The central dogma of molecular biology deals with the detailed residue-by-residue transfer of sequential information. It states that such information cannot be transferred back from protein to either protein or nucleic acid.”

DNA, chromozómy

DNA: dve komplementárne vlákna, strands (páry A-T, C-G), v opačnej orientácii (konce sa nazývajú 5' a 3').

Napr. ACCATG je komplementárny s CATGGT.

Tvar dvojitej špirály:



Dvojvláknová štruktúra poskytuje redundanciu, možnosť opravy pri poškodení jedného vlákna.

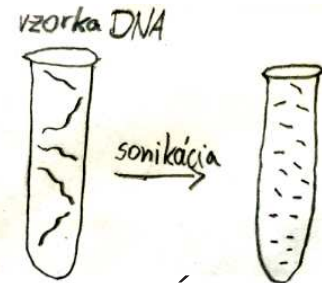
Pri delení bunky sa dvojvláknová DNA rozdelí a ku každému vláknu sa doplní komplement (DNA replikácia).

Chromozóm: Súvislý úsek dvojvláknovej DNA a podporných proteínov.

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

Technológia: sekvenovanie DNA

- Postup na zisťovanie poradia báz v chromozómoch genómu.
- Zložitý proces:
chromozómy sa nasekajú na krátke kúsky,
z každého sa vyrobí veľa kópií,
každý sa nasekvenuje 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.



Sangerovo sekvenovanie (Sanger sequencing)

Sekvenujeme AGCTAGGACT (zobrazená sprava doľava)

Primer AGT + enzýmy + nukleotidy + modifikované ofarbené nukleotidy

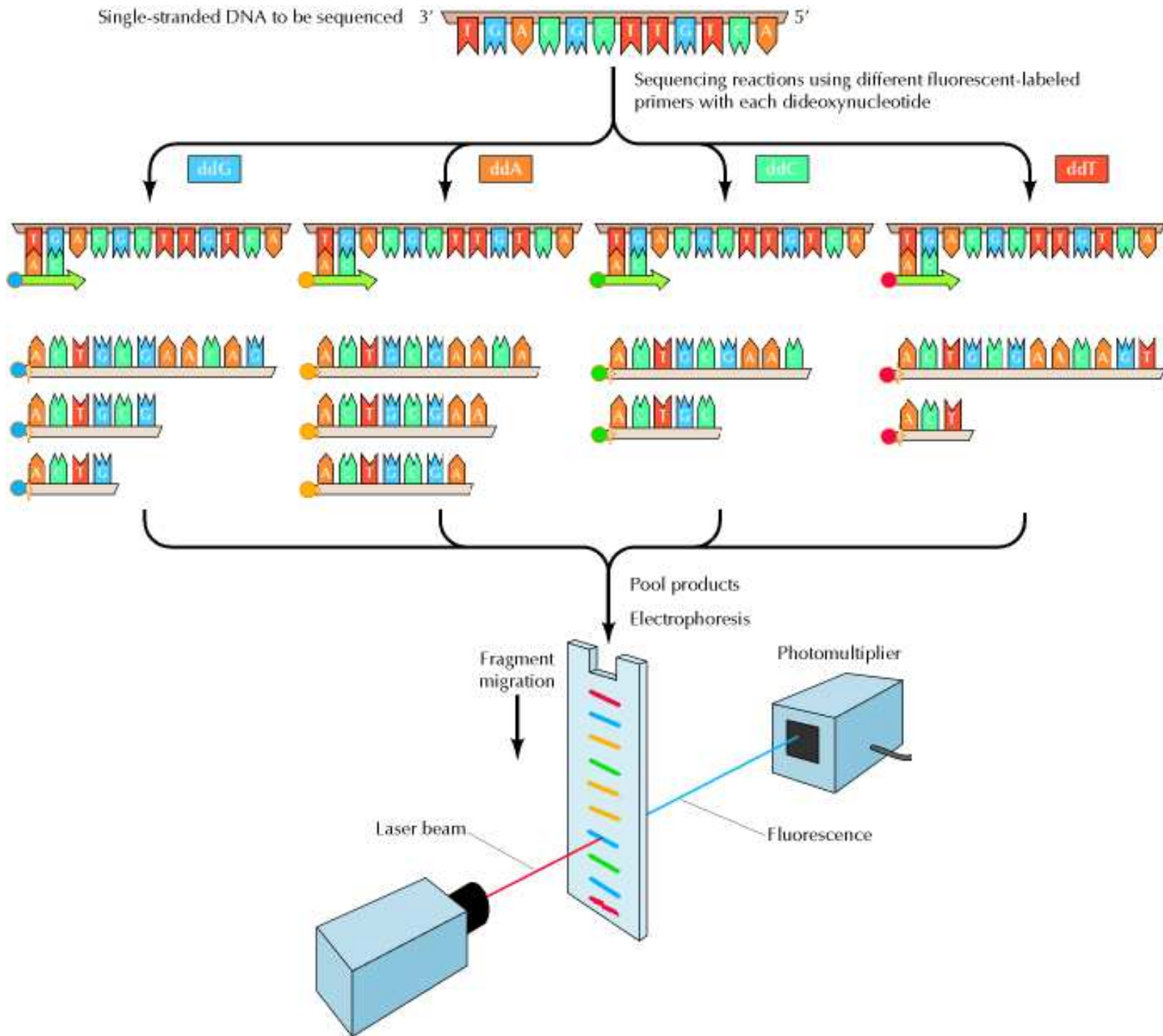
Výsledky sekvenovacej reakcie:

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

Na géli zoradíme podľa dĺžky:

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

Odčítaním farieb dostaneme komplementárne vlákno: AGTCCTAGCT

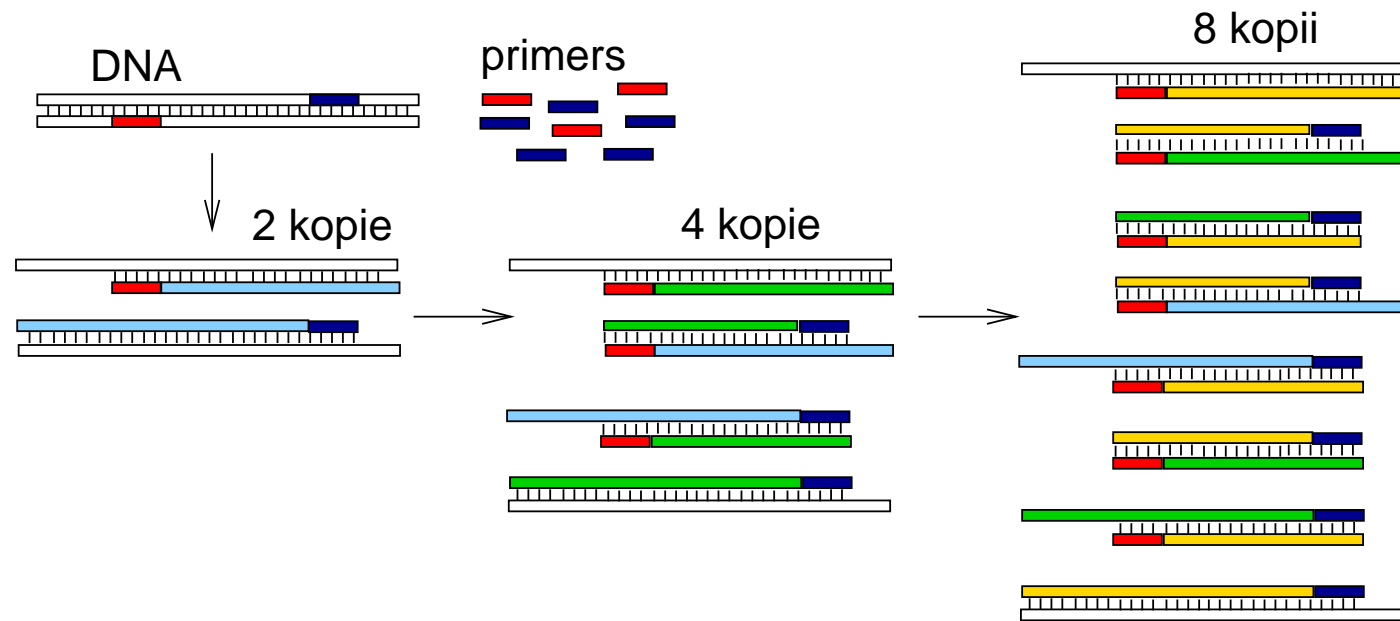


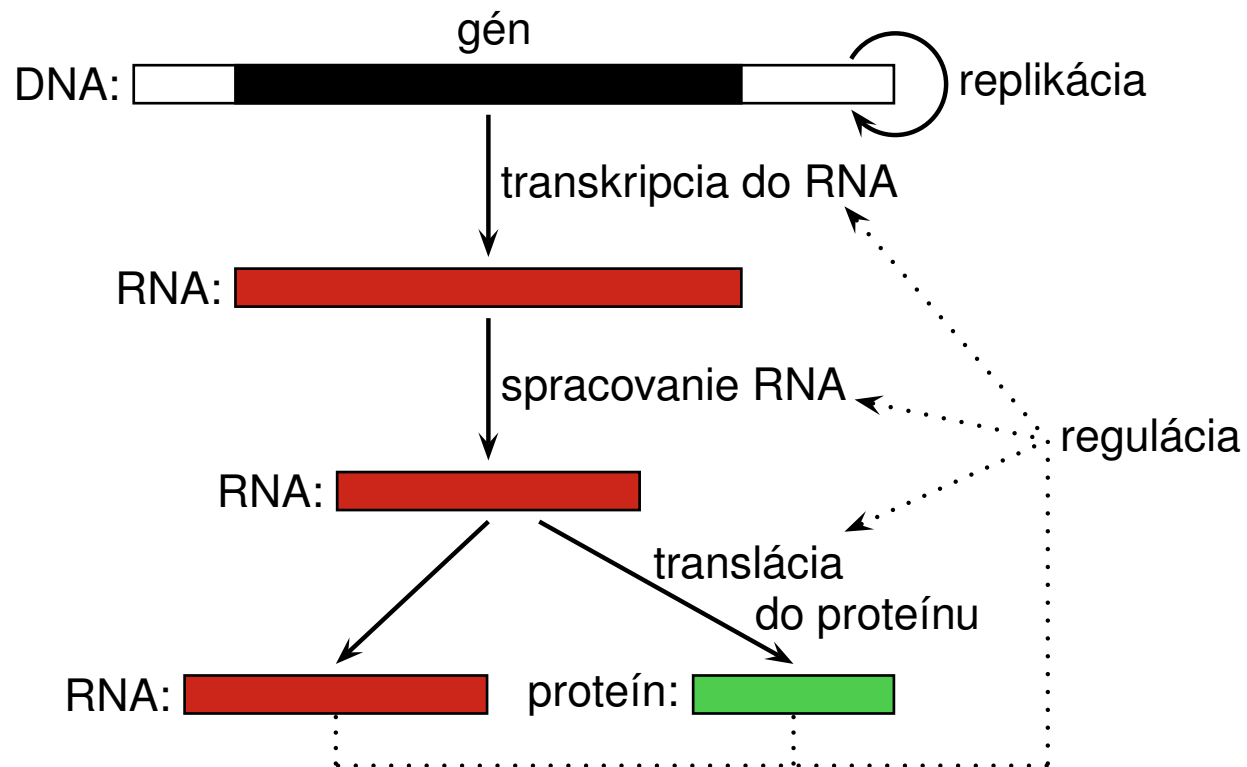
PCR (polymerase chain reaction)

Zvolíme si dva krátke úseky DNA (primers)

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

Ak áno, namnoží úsek medzi nimi

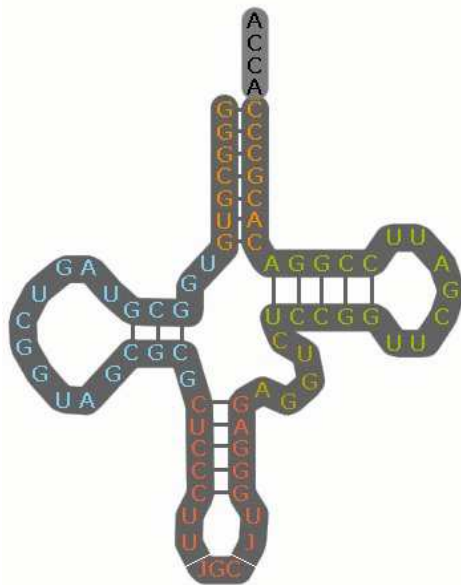




RNA

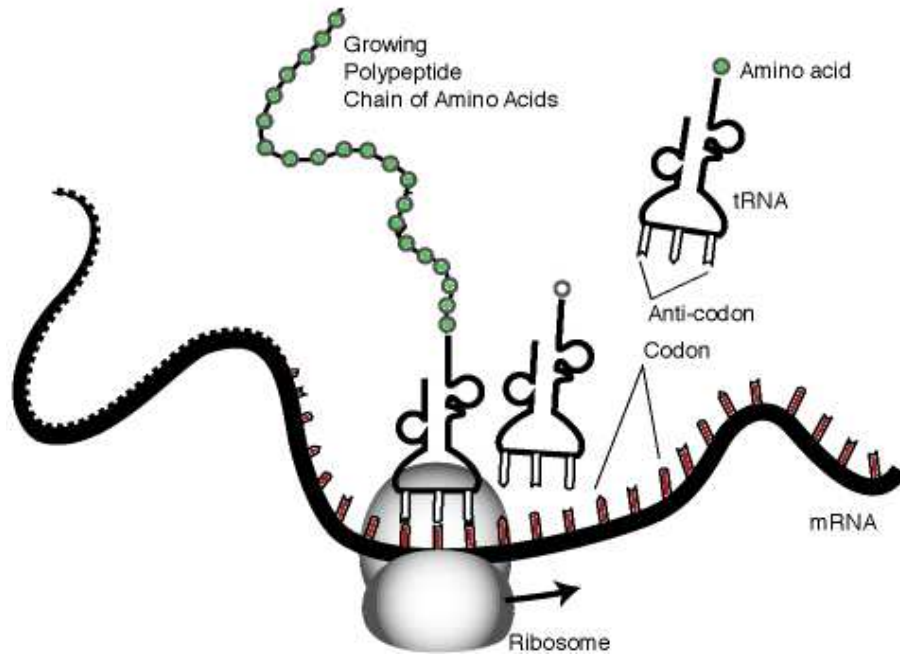
Ako sa líši od DNA?

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

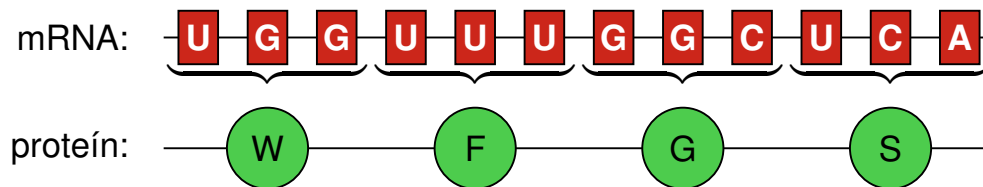


transferová RNA (tRNA)

Translácia



Kodón (trojica nukleotidov) určuje 1 aminokyselinu



Genetický kód

Alanine (A)

GC*

Cysteine (C)

TGC

TGT

Aspartic acid (D)

GAC

GAT

Glutamic acid (E)

GAA

GAG

Phenylalanine (F)

TTC

TTT

Glycine (G)

GG*

Histidine (H)

CAC

CAT

Isoleucine (I)

ATA

ATC

ATT

Lysine (K)

AAA

AAG

Leucine (L)

CT*

TTA

TTG

Methionine (M)

ATG

Asparagine (N)

AAC

AAT

Proline (P)

CC*

Glutamine (Q)

CAA

CAG

Arginine (R)

CG*

AGA

AGG

Serine (S)

TC*

AGT

AGC

Threonine (T)

AC*

Valine (V)

GT*

Tryptophan (W)

TGG

Tyrosine (Y)

TAC

TAT

Stop codon (*)

TAA

TAG

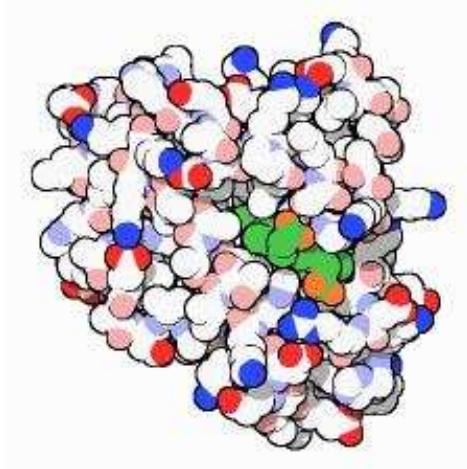
TGA

Proteíny

Reťazce 20 rôznych aminokyselín s rôznymi chemickými vlastnosťami:

Amino Acid	Side chain	Hydrophobic	Polar	Charged	
Alanine (A)	-CH ₃	X	-	-	-
Arginine (R)	-(CH ₂) ₃ NH-C(NH)NH ₂	-	X	basic	-
Asparagine (N)	-CH ₂ CONH ₂	-	X	-	-
Aspartic acid (D)	-CH ₂ COOH	-	X	acidic	-
Cysteine (C)	-CH ₂ SH	X	-	acidic	-
Glutamic acid (E)	-CH ₂ CH ₂ COOH	-	X	acidic	-
Glutamine (Q)	-CH ₂ CH ₂ CONH ₂	-	X	-	-
Glycine (G)	-H	-	-	-	-
Histidine (H)	-CH ₂ -C ₃ H ₃ N ₂	-	X	weak basic	Aromatic
Isoleucine (I)	-CH(CH ₃)CH ₂ CH ₃	X	-	-	Aliphatic
Leucine (L)	-CH ₂ CH(CH ₃) ₂	X	-	-	Aliphatic
Lysine (K)	-(CH ₂) ₄ NH ₂	-	X	basic	-
Methionine (M)	-CH ₂ CH ₂ SCH ₃	X	-	-	-
Phenylalanine (F)	-CH ₂ C ₆ H ₅	X	-	-	Aromatic
Proline (P)	-CH ₂ CH ₂ CH ₂ -	X	-	-	-
Serine (S)	-CH ₂ OH	-	X	-	-
Threonine (T)	-CH(OH)CH ₃	-	X	weak acidic	-
Tryptophan (W)	-CH ₂ C ₈ H ₆ N	X	-	-	Aromatic
Tyrosine (Y)	-CH ₂ -C ₆ H ₄ OH	X	X	-	Aromatic
Valine (V)	-CH(CH ₃) ₂	X	-	-	Aliphatic

Štruktúra proteínov

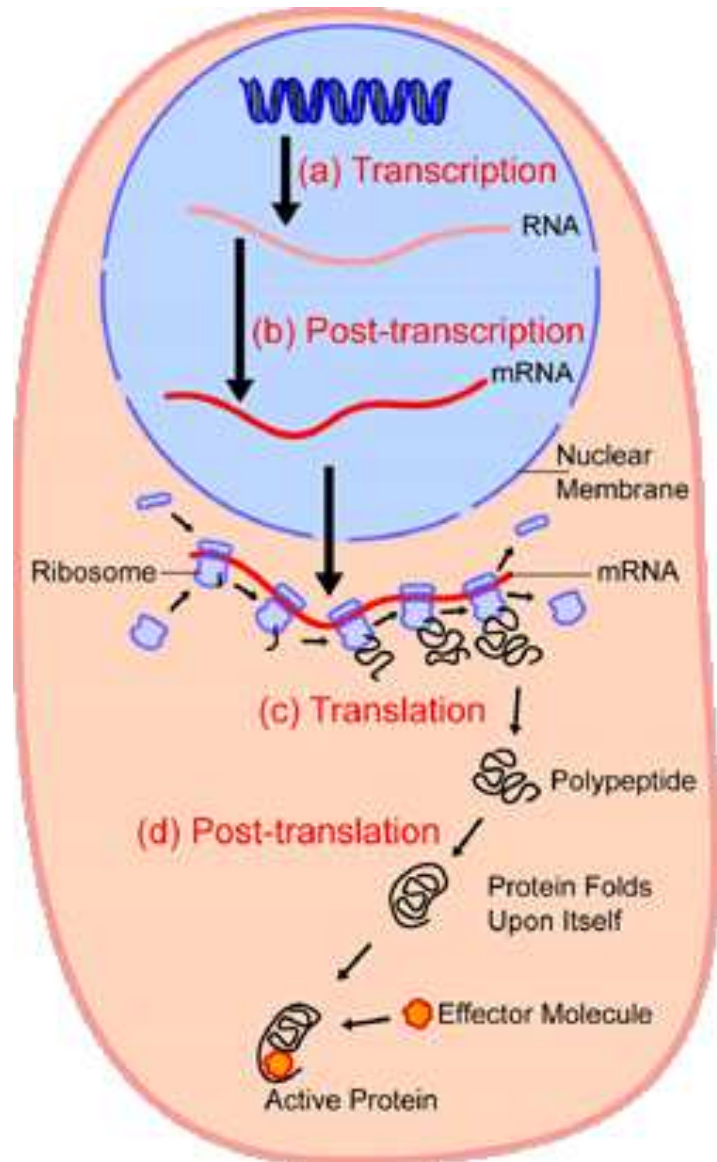


Myoglobín, prvý proteín so známou štruktúrou.

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.

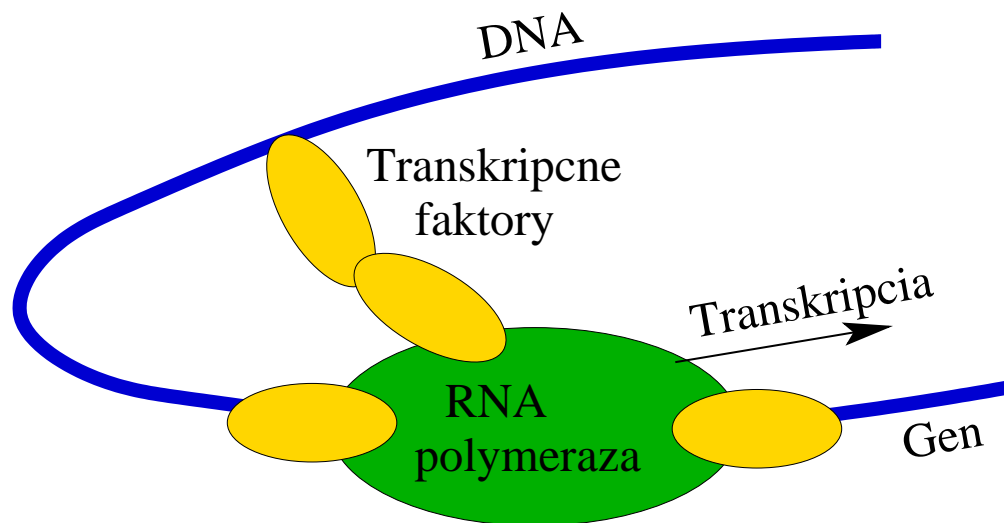


Regulácia expresie

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

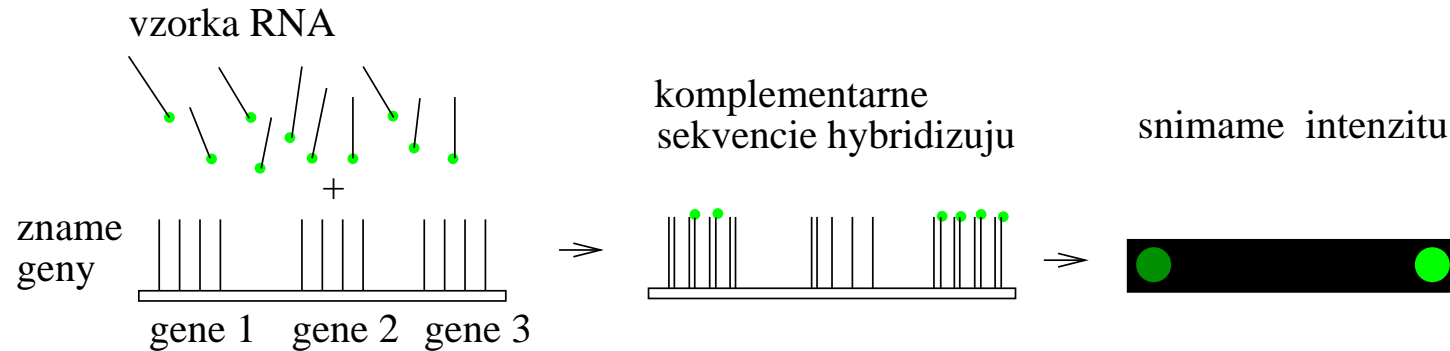
Niektoré proteíny sa tvoria len za určitých okolností, alebo v premenlivom množstve.

Regulácia začatia transkripcie pomocou transkripčných faktorov:



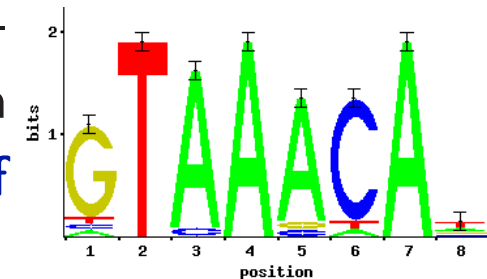
Bioinformatický problém: zistiť, ktoré faktory ovplyvňujú ktorý gén, kde presne sa viažu.

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



Príklad microarray dát

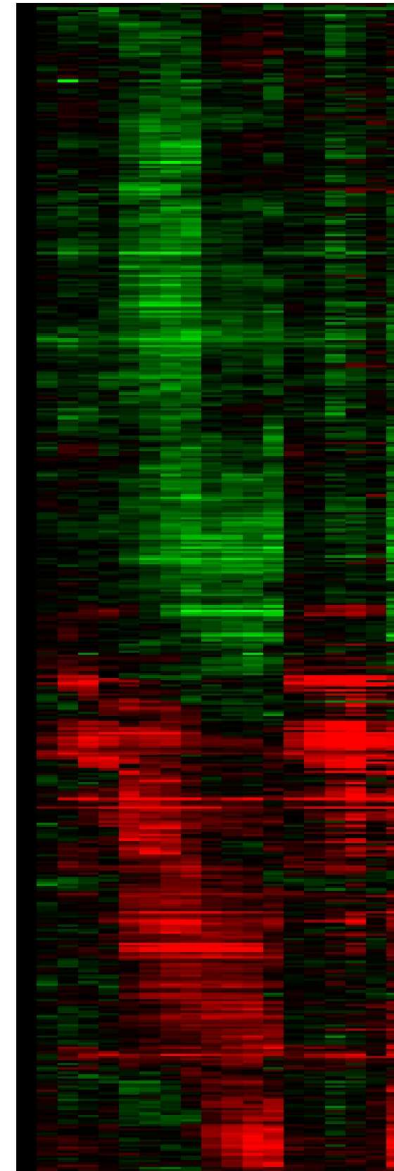
Pomer expresie génu v meranej a kontrolnej vzorke fg/bg

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 ujmu (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:

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

Evolúcia

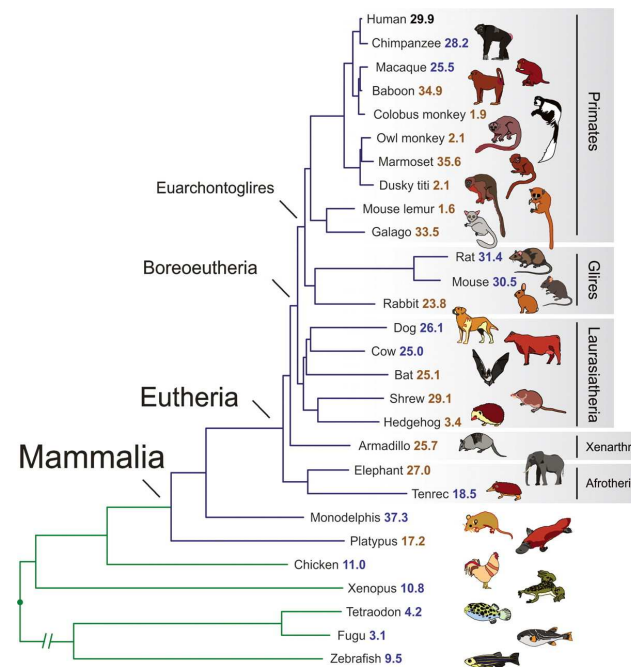
Vznik nových druhov (speciation):

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.

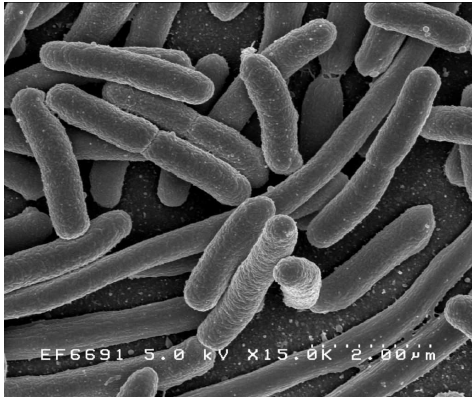
Bunka obsahuje jadro s DNA, viacero organel.

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), akvarijská 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)

Úvod do dynamického programovania (cvičenie)

Broňa Brejová
1.10.2015

Problém platenia minimálnym počtom mincí

Vstup: hodnoty k mincí m_1, m_2, \dots, m_k a cieľová suma X (všetko kladné celé čísla)

Výstup: najmenší počet mincí, ktoré potrebujeme na zaplatenie X

Príklad: $k = 3, m_1 = 1, m_2 = 2, m_3 = 5, X = 13$

Problém platenia minimálnym počtom mincí

Vstup: hodnoty k mincí m_1, m_2, \dots, m_k a cieľová suma X (všetko kladné celé čísla)

Výstup: najmenší počet mincí, ktoré potrebujeme na zaplatenie X

Príklad: $k = 3, m_1 = 1, m_2 = 2, m_3 = 5, X = 13$

Príklad: $k = 3, m_1 = 1, m_2 = 3, m_3 = 4, X = 6$

Dynamické programovanie vo všeobecnosti

- Okrem riešenia celého problému riešime aj menšie problémy (nazývame ich podproblémy)
- Riešenia podproblémov ukladáme do tabuľky a používame pri riešení väčších podproblémov
- Technika dynamického programovania sa používa na viacero problémov v bioinformatike

Úvod do dynamického programovania, proteomika

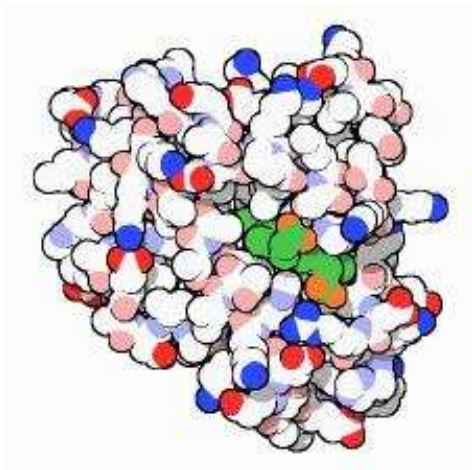
Broňa Brejová

8.10.2015

Proteomika

Proteín: sekvencia pozostáva z 20 rôznych aminokyselín

MGLSDGEWQLVLNVWGKVEADIPGHGQEVLIIRLFKGHPEKFDKFKHLKSEDEMKASE
DLKKHGATVLTALGGILKKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKH
PGDFGADAQGAMNKALELFRKDMASNYKELGFQG



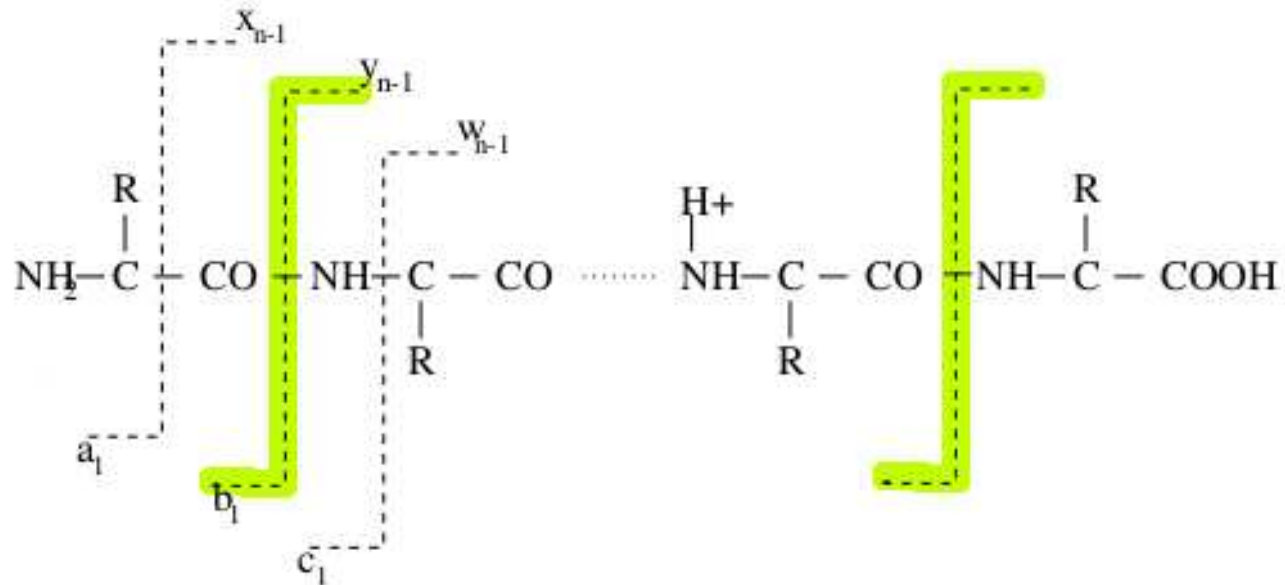
Z bunky sme izolovali určitý proteín, chceme zistiť jeho sekvenciu.

Hmotnostná spektrometria (mass spectrometry)

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

Tandemová hmotnostná spektrometria MS/MS

Štiepenie peptidu na prefixy a sufixy



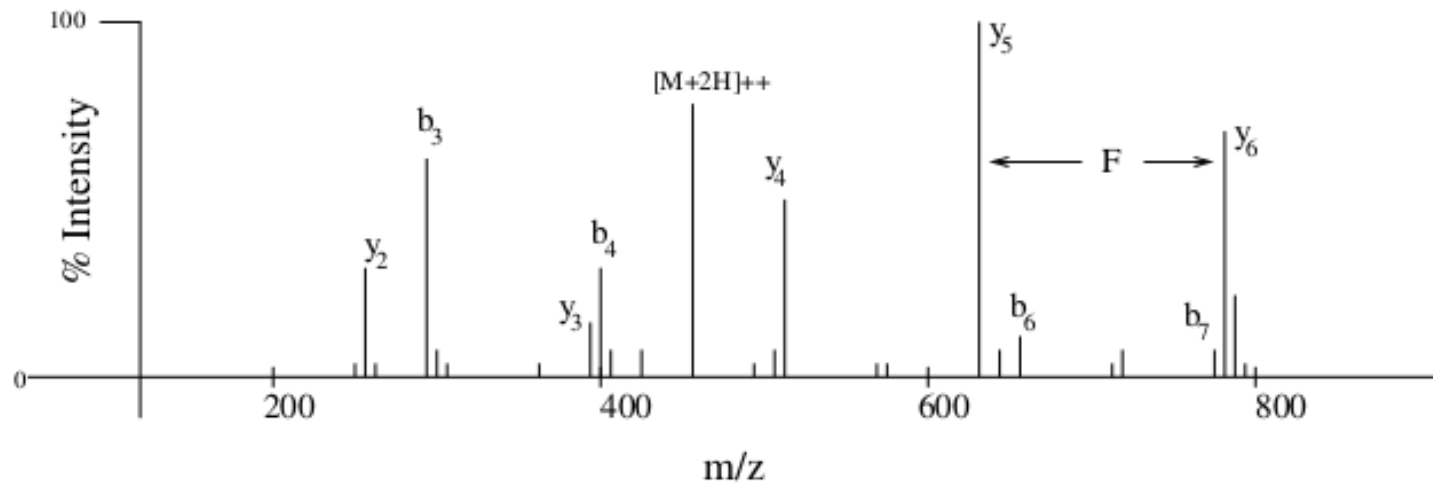
zdroj: Bafna and Reinert

b-ióny: prefixy

y-ióny: sufixy

Tandemová hmotnostná spektrometria MS/MS

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



zdroj: Bafna and Reinert

Sekvenovanie peptidov pomocou MS/MS

Vstup: celková hmotnosť peptidu M ,
hmotnosti aminokyselín $a[1], \dots, a[20]$ (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 $\mathcal{M}_P(XZY Y) =$

$$\{m(), m(X), m(XZ), m(XZY Y), m(XZY Y)\} = \{0, 4, 11, 17, 23\}$$

Hmotnosti sufixov $\mathcal{M}_S(XZY Y) =$

$$\{m(), m(Y), m(Y Y), m(ZY Y), m(XZY Y)\} = \{0, 6, 12, 19, 23\}$$

$$\text{Skóre } XZY Y: \sum_{m \in \mathcal{M}_P(ZY X X)} f[m] = 0 + 1 + 1 + 1 + 0 = 3$$

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

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

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

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

$$\mathcal{M}_P(XZX X X) = \{0, 4, 11, 15, 19, 23\}$$

$$\mathcal{M}_S(XZX X X) = \{0, 4, 8, 12, 19, 23\}$$

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

Skóre XZY Y: $0 + 1 + 1 + 1 + 0 + 0 + 1 + 1 + 1 + 0 = 6$

Skóre XZX X X: $0 + 1 + 1 + 0 + 1 + 0 + 0 + 1 + 0 + 1 + 1 + 0 = 6$

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

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

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

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

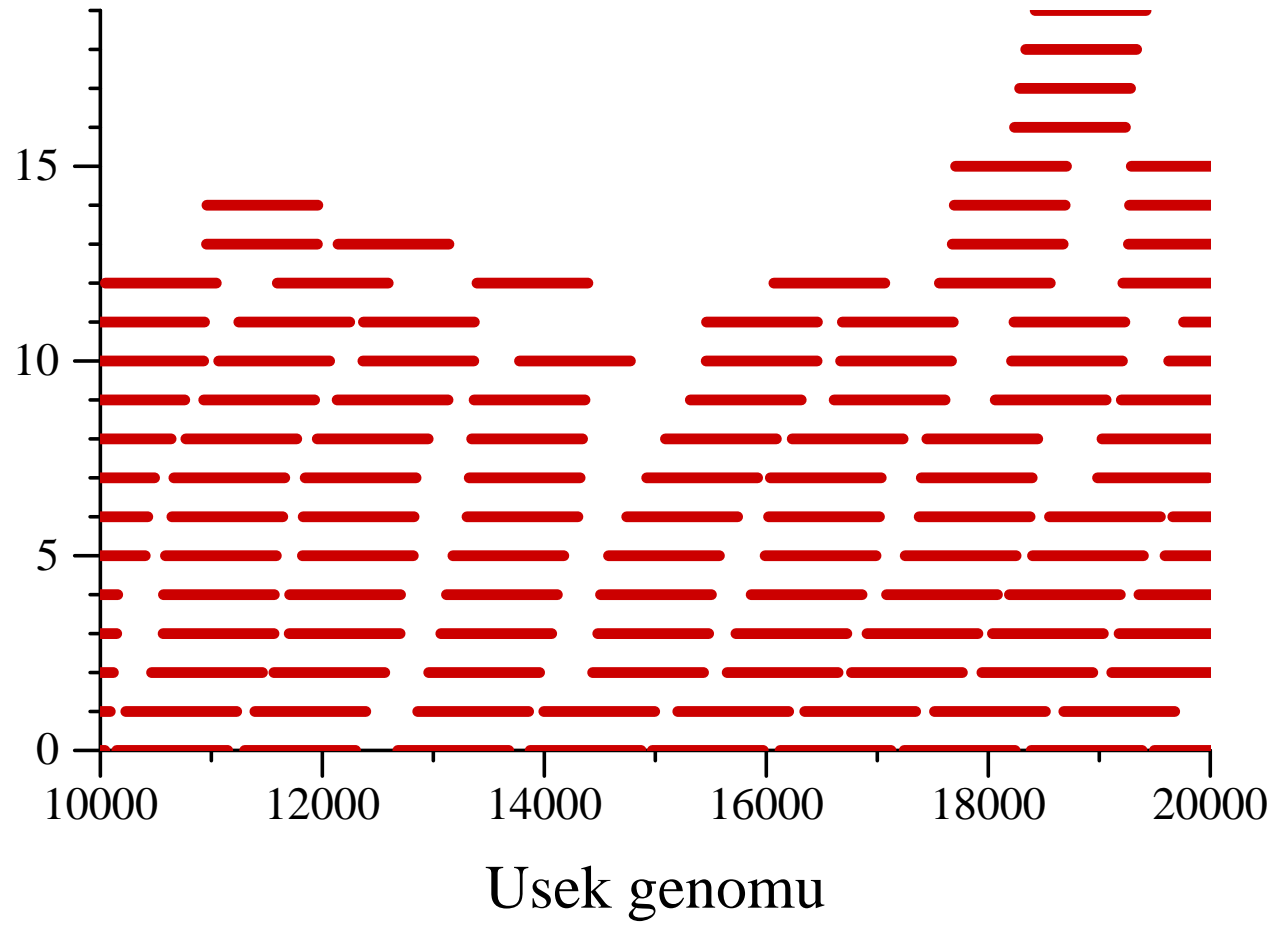
$$\text{kde } h[m] = \begin{cases} f[m] + f[M - m] & \text{ak } m < M/2 \\ f[m] & \text{ak } m = M/2 \end{cases}$$

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

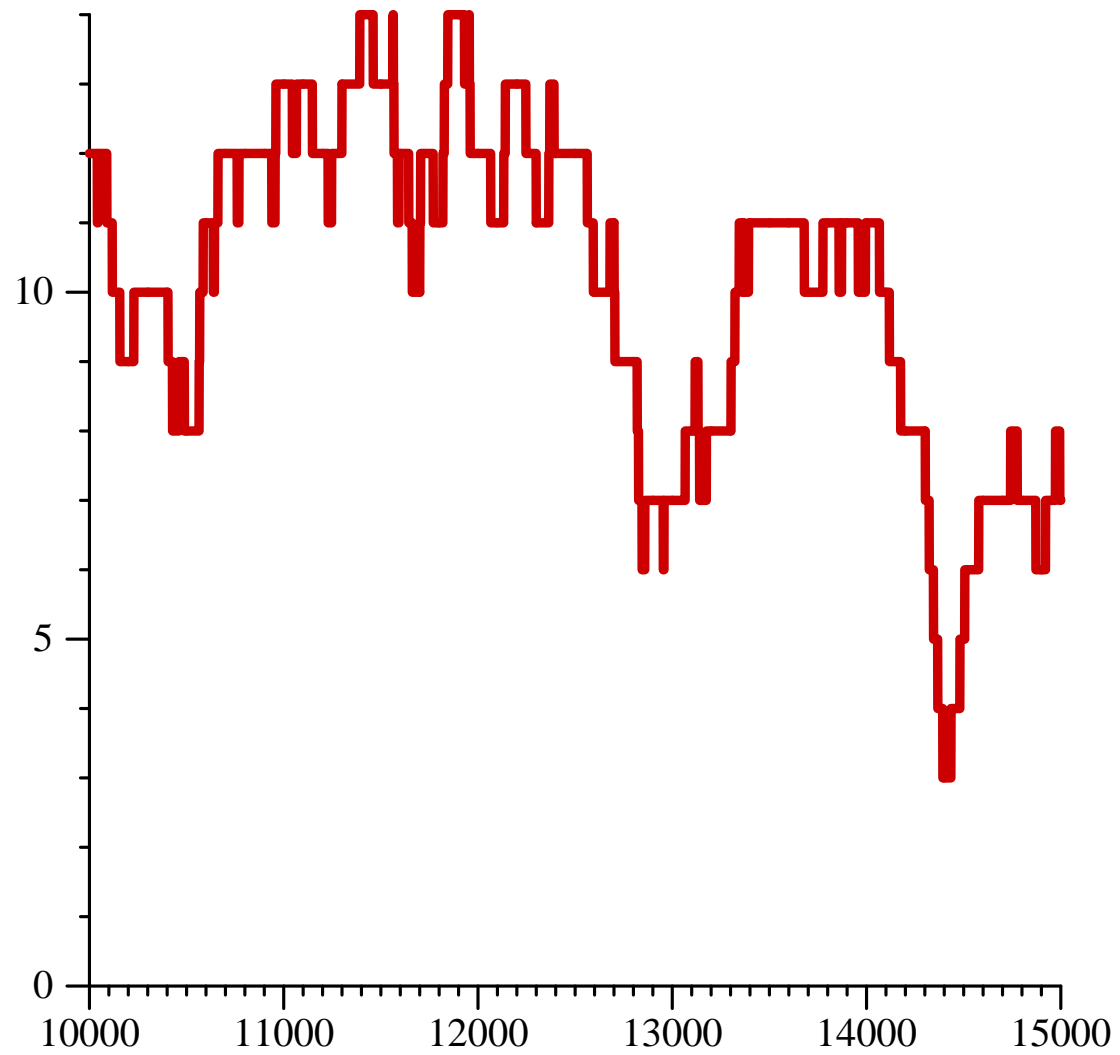
**Tomáš Vinař
22.10.2015**

- 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

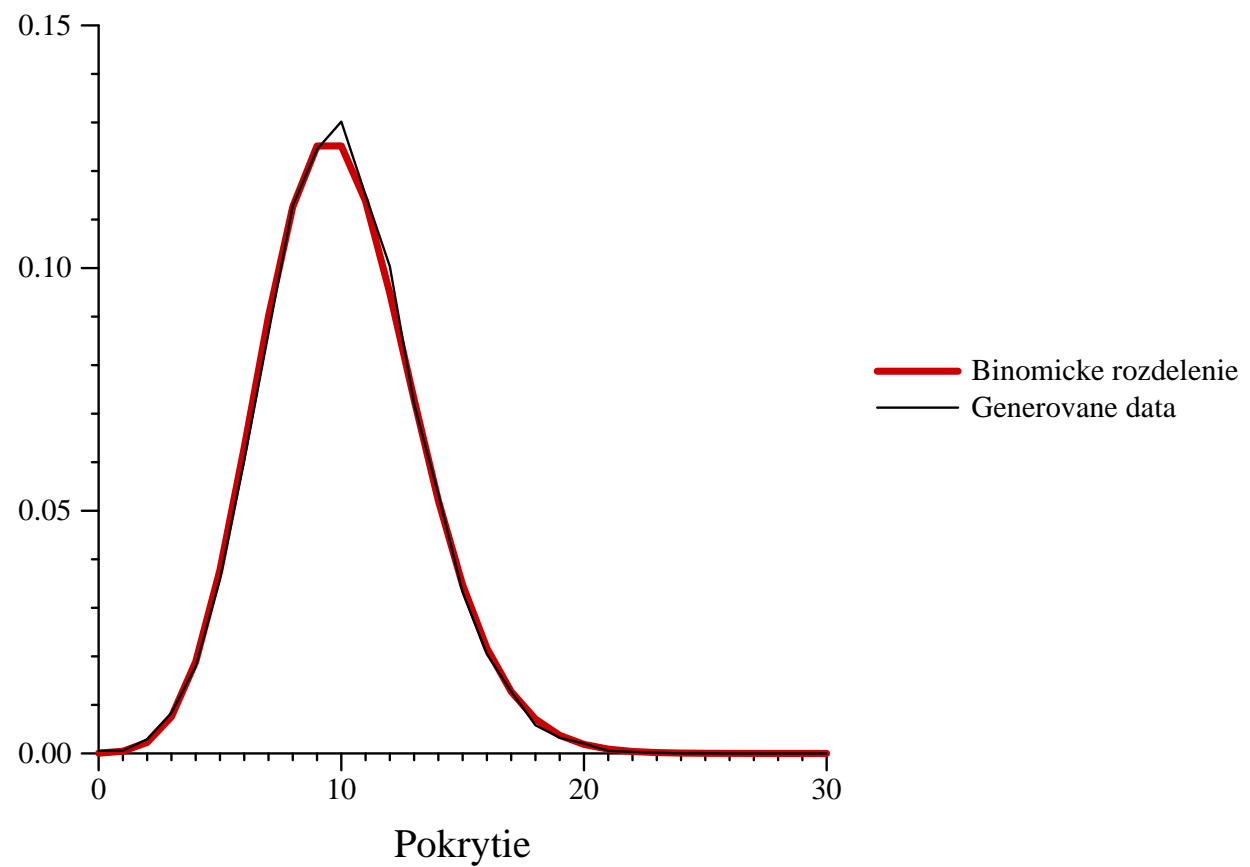
Náhodne generované čítania



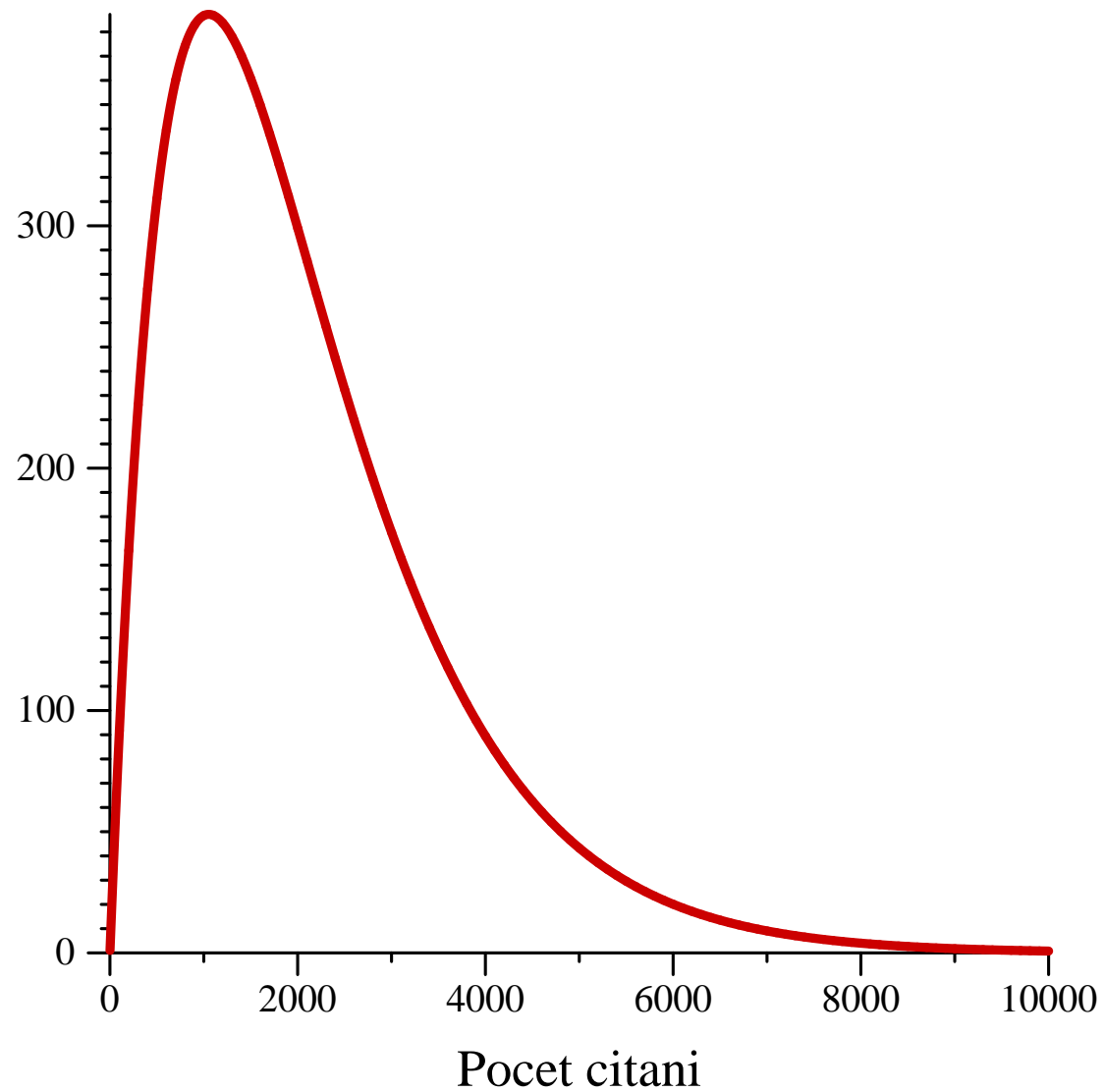
Pokrytie jednotlivých báz



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

Jadrá zarovnaní

Tomáš Vinař

29.10.2015

Opakovanie: Heuristické lokálne zarovnávanie, BLAST

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

(V praxi sa používa $w = 10$ a viac.)

		C	A	G	T	C	C	T	A	G	A
	0	0	0	0	0	0	0	0	0	0	0
C	0	1	0	0	0	1	1	0	0	0	0
A	0	0	2	1	0	0	0	0	1	0	0
T	0	0	0	1	2	1	0	1	0	0	0
G	0	0	0	0	1	0	0	0	0	1	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. nájdí zhodné úseky
2. rozšír bez medzier
3. spoj medzerami

Senzitivita heuristického algoritmu

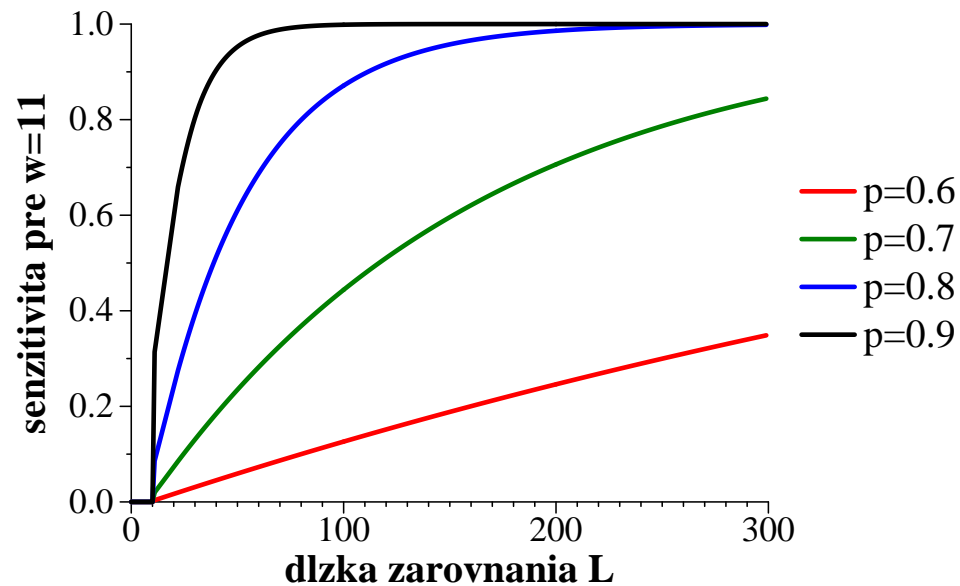
Odhad senzitivity:

Predpokladáme zarovnanie bez medzier, dĺžky L

Každá pozícia je zhoda s pravdepodobnosťou p

Senzitivita:

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



Jadrá z medzerami, spaced seeds

PatternHunter [Ma, Tromp, Li 2002]

Jadro s medzerami: vyžadovaná konfigurácia zhôd

Príklad:

“match—match—don't care—match” značíme ako 1101

```
GTGGTGCTCTCTGACAAAGCC
| | | | | | | | |
ATTGTTCTTAATGAGAAAGAA
  1101      1101
                1101
```

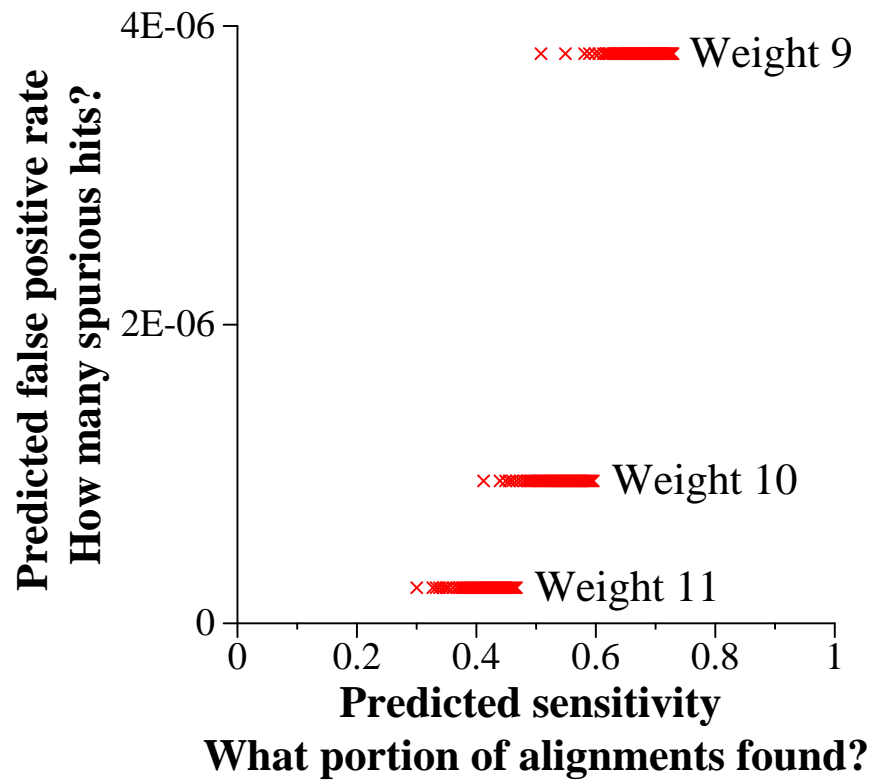
BLASTN jadro (11 za sebou idúcich zhôd)

ekvivalentné jadro 11111111111

Nie všetky jadrá sú rovnaké

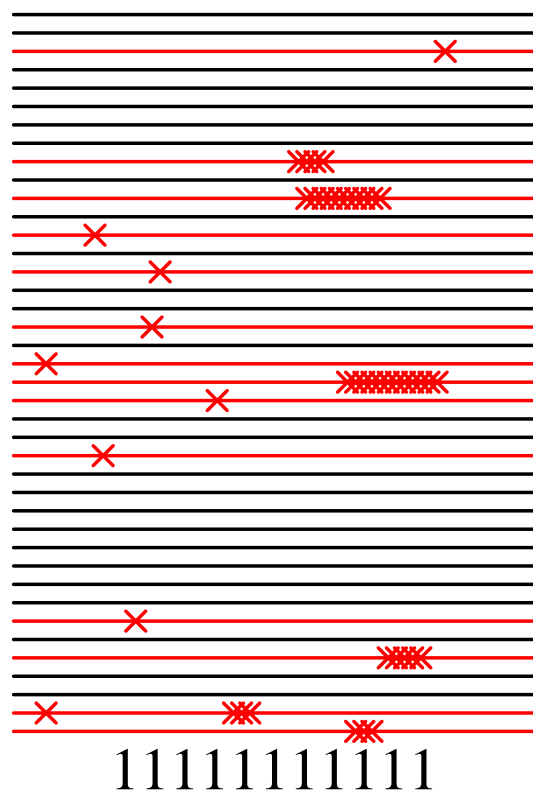
Váha jadra: počet vyžadovaných zhôd

Každý krížik: senzitivita vs. čas pre jedno jadro v pravdepodobnostnom modeli

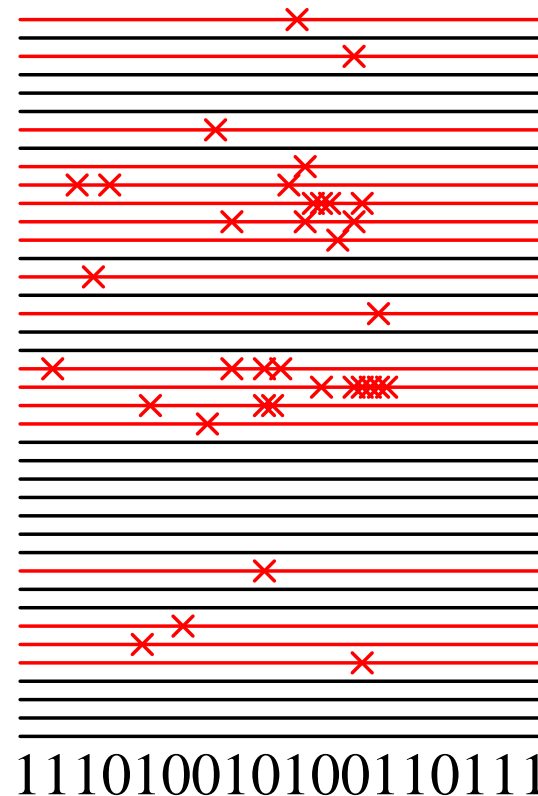


Prečo sú jadrá s medzerami senzitivnejšie?

Príklad: dĺžka zarovnaní $L = 64$,
pravdepodobnosť zhody $p = 0.7$ na každej pozícii
40 náhodných zarovnaní, výskyty jadra



Sn.: 14/40, hits: 46



Sn.: 18/40, hits: 35

Prečo sú jadrá s medzerami senzitivnejšie?

Príklad: dĺžka zarovnaní $L = 64$,
pravdepodobnosť zhody $p = 0.7$ na každej pozícii

Bez medzier

111111111111

S medzerami

111010010100110111

Stredná hodnota počtu výskytov v zarovnaní:

$$54 \cdot 0.7^{11} = 1.1$$

$$47 \cdot 0.7^{11} = 0.9$$

Pravdepodobnosť výskytu na poz. $i + 1$ ak výskyt na i :

$$0.7$$

$$0.7^6 = 0.12$$

111111111111

111010010100110111

 111111111111

 111010010100110111

Výskyty často vedľa seba

Výskyty “nezávislejšie”

Senzitivita (pravdepodobnosť aspoň jedného výskytu):

$$0.30$$

$$0.47$$

Ďalšie hašovacie stratégie

Nukleotidový BLAST: 10 zhôd za sebou

Jadro s medzerami: povoľuje nezhody na 8 z 18 pozícií

BLAT [Kent 2002]: povoľuje 1 nezgodu na ľub. z 11 pozícií

BLASTP: 3 amino kyseliny so skóre aspoň 13 v matici BLOSUM62

Výskyt: N I R

N L R

$$6+2+5=13$$

Nie výskyt: A I L

A I L

$$4+4+4=12$$

Vektorové jadrá: kombinácia jadier s medzerami a BLAT/BLASTP

Viaceré výskyty: začni rozširovať iba ak viac výskytov blízko seba na tej istej uhlopriečke

Viaceré jadrá: zober zjednotenie výskytov

Záleží na modeli zarovnaní

Pravdepodobnosť zhody kolíše v rámci kodónu:

Poloha v kodóne:	prvá	druhá	tretia
Pravdepodobnosť zhody:	0.67	0.77	0.40

Senzitivita na testovacej vzorke exónov kódujúcich proteíny:

Jadro		Človek vs.	
		Drosophila	myš
Optimálne pre dáta	110 110 000 110 110 11	86%	92%
Optimalne pre kodónový model	110 110 010 110 010 11	86%	91%
WABA [Kent, Zahler 2000]	110 110 110 110 11	80%	90%
Optimálne pre i.i.d. model	111001001001010111	60%	86%
BLAST	1111111111	43%	81%
Najhoršie	101010101010101011	39%	79%

A čo globálne zarovnanie?

Ukotvené zarovnanie (Anchored alignment)

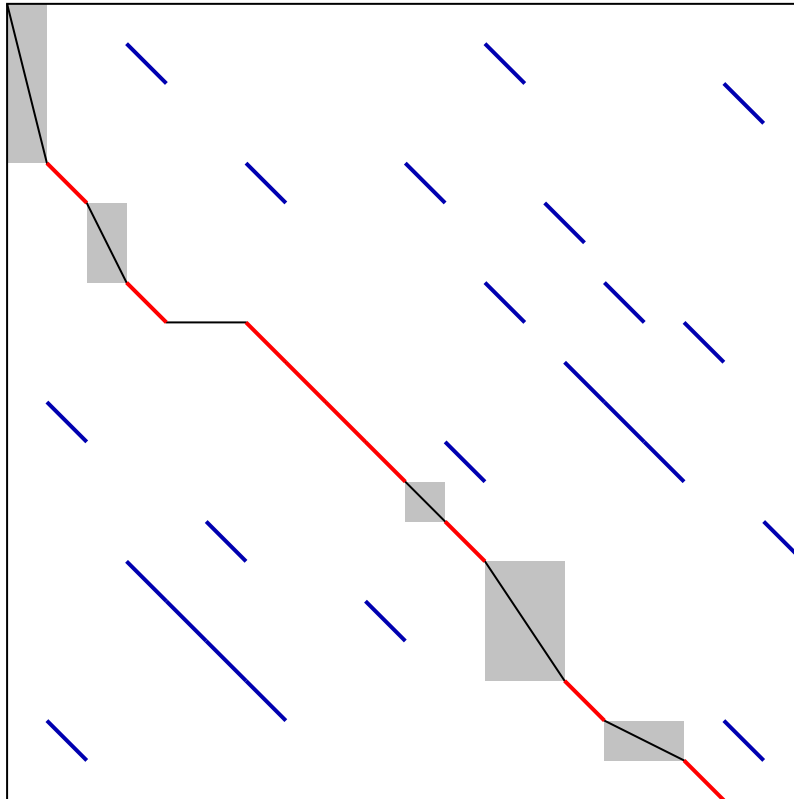
- Nájdime lokálne zarovnanie (alebo výskyty nejakého jadra)
– možné **ukotvenia**
- Zvoľ konzistentnú množinu ukotvení
(monotónna postupnosť)
- Zarovnaj časti sekvencií medzi ukotveniami
(pomocou dyn. prog. alebo rekurzívne ďalším kotvením)

MUMMER [Delcher 1999]

GLASS [Batzoglou et al 2000]

AVID [Bray et al 2003]

Ukotvené zarovnanie



Modré: nezvolené ukotvenia
Červené: zvolené ukotvenia
Sivé: riešime dyn. prog.
Čierne: globálne zarovnanie

Znova protichodné vplyvy:

málo spoľahlivých ukotvení – dobrá kvalita, pomalé

veľa slabších ukotvení – rýchle (malá sivá plocha), viac chýb v ukotvení

Zložitejšie modely substitúcií

Broňa Brejová

12.11.2015

Substitučné modely, opakovanie

- $S(t)$: matica 4×4 , kde $S(t)_{a,b} = P(b|a, t)$ je pravdepodobnosť, že ak začneme s bázou a , tak po case t budeme mať bázu b .
- Jukes-Cantorov model predpokladá, že táto pravdepodobnosť je rovnaká pre každé dve bázy $a \neq b$

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

Matica rýchlostí

- Matice rýchlostí pre Jukes-Cantorov model:

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

- Pre veľmi malý čas t je $S(t) \approx I - Rt$
- Rýchlosť α je pravdepodobnosť zmeny za jednotku času pre veľmi krátke t , resp. derivácia $s(t)$ vzhľadom na t v bode 0
- Riešením diferenciálnych rovníc pre Jukes-Cantorov model dostávame $s(t) = (1 - e^{-4\alpha t})/4$
- Matice rýchlostí sa zvykne normalizovať tak, aby na jednotku času pripadla v priemere jedna substitúcia, čo dosiahneme ak $\alpha = 1/3$

Zložitejšie modely

- Všeobecná matica rýchlostí R

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

- μ_{xy} je rýchlosť, akou sa báza x mení na inú bázu y
- Presnejšie $\mu_{xy} = \lim_{t \rightarrow 0} \frac{\text{Pr}(y | x, t)}{t}$
- Diagonálu dopočítame tak, aby súčet každého riadku bol 0.
- Existujú modely s menším počtom parametrov (kompromis medzi J-C a ľubovoľnou maticou)

Kimurov model

- Zachytáva, že puríny sa častejšie menia na iné puríny (A a G) a pyrimidíny na ine pyrimidíny (C a T)
- Má dva parametre: rýchlosť tranzícií α , transverzií β

$$\bullet R = \begin{pmatrix} -2\beta - \alpha & \beta & \alpha & \beta \\ \beta & -2\beta - \alpha & \beta & \alpha \\ \alpha & \beta & -2\beta - \alpha & \beta \\ \beta & \alpha & \beta & -2\beta - \alpha \end{pmatrix}$$

HKY model (Hasegawa, Kishino, Yano)

- Rozšírenie Kimurovho modelu, umožňuje aj rôzne pravdepodobnosti A, C, G a T v ekvilibriu
- Ak nastavíme čas v evolučnom modeli na nekonečno, nezáleží na tom, z ktorej bázy sme začali, frekvencia výskytu jednotlivých báz sa ustáli v tzv. ekvilibriu.
- V Jukes-Cantorovom modeli je pravdepodobnosť každej bázy v ekvilibriu 1/4.
- V HKY si zvolíme aj frekvencie jednotlivých nukleotidov v ekvilibriu $\pi_A, \pi_C, \pi_G, \pi_T$ so súčtom 1
- Parameter κ : pomer tranzícií a transverzií (α/β)
- Matica rýchlostí:

$$\mu_{x,y} = \begin{cases} \kappa\pi_y & \text{ak mutácia z } x \text{ na } y \text{ je tranzícia} \\ \pi_y & \text{ak mutácia z } x \text{ na } y \text{ je transverzia} \end{cases}$$

Od R k $S(t)$

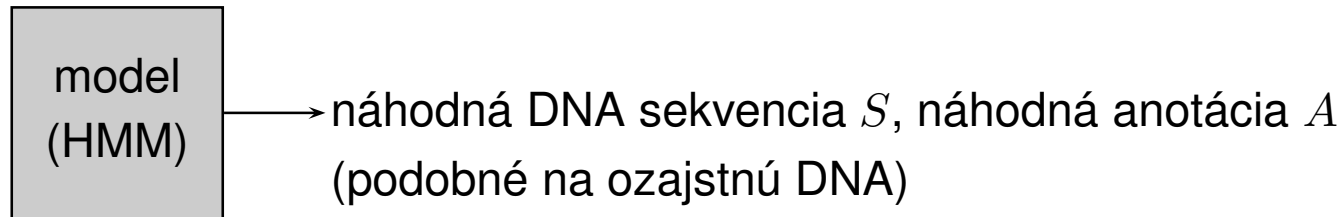
- Pre zložité modely nevieme odvodiť explicitný vzorec na výpočet $S(t)$, ako sme mali pri Jukes-Cantorovom modeli
- Vo všeobecnosti $S(t) = e^{Rt}$
- Exponenciálna funkcia matice A sa definuje ako $e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$
- Ak R diagonalizujeme $R = UDU^{-1}$, kde D je diagonálna matica, tak $e^{Rt} = Ue^{Dt}U^{-1}$ a exponenciálnu funkciu uplatníme iba na prvky na uhlopriečke D
- Diagonalizácia vždy existuje pre symetrické R

Algoritmy pre HMM a phyloHMM

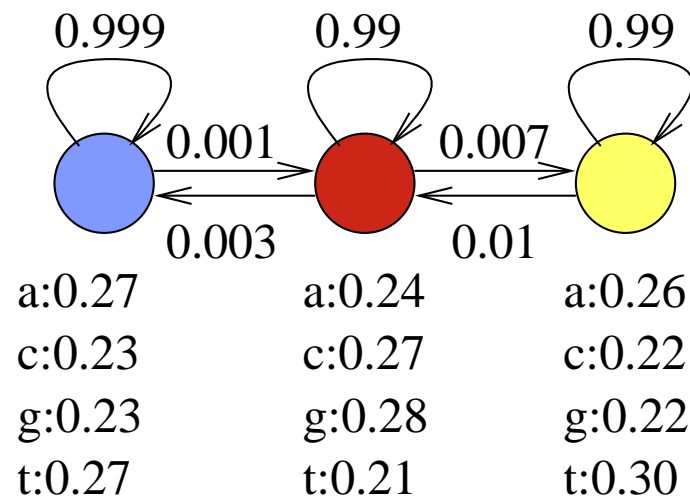
Tomáš Vinař

19.11.2015

Opakovanie: HMM (skrytý Markovov model)



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

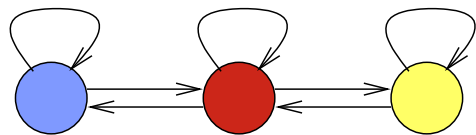


Predpokladajme, že model vždy začína v modrom stave.

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

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

Parametre HMM (označenie)



Sekvencia S_1, \dots, S_n







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

Viterbiho algoritmus

Pre danú sekvenciu S nájde najpravdepodobnejšiu anotáciu

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

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

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

Rekurencia:

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

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

Algoritmus:

Inicializuj $V[1, *]$

for $i = 2 \dots n$ (n =dĺžka reťazca)

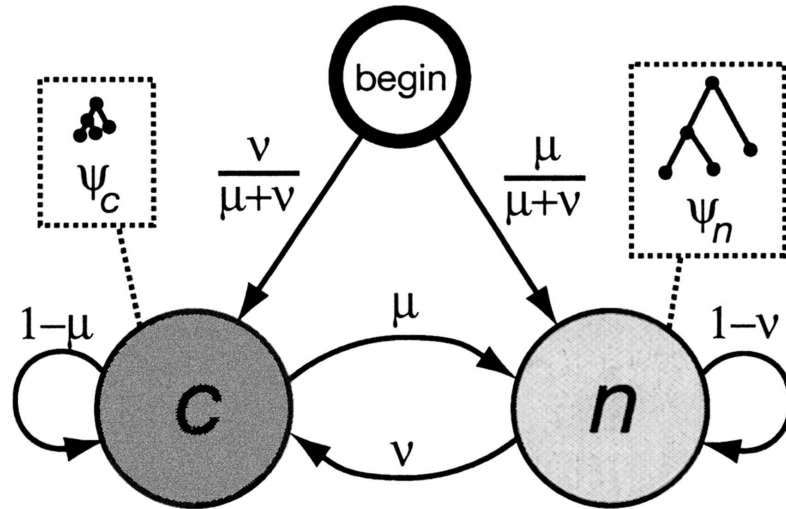
 for $u = 1 \dots m$ (m = počet stavov)

 vypočítaj $V[i, u]$

Maximálne $V[n, j]$ je pravdepodobnosť najpravdepodobnejšej cesty

PhyloHMM: kombinácia HMM a fylogenetického stromu

PhastCons: detekcia dobre zachovaných sekvencií



- Dva stavy: zachovaná sekv., neutrálna sekv.
- V každom stave generujeme celý stĺpec zarovnaní
- Zachovaná sekvencia má kratšie hrany stromu

x =

TCGCGACATATACGA...
TTGGGGCATGTGGGT...
AGCAGACGTCCGCAA...

Dopredný algoritmus

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S

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

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

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

Rekurencia:

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

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

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

Spätňý algoritmus

Obdoba dopředného algoritmu

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

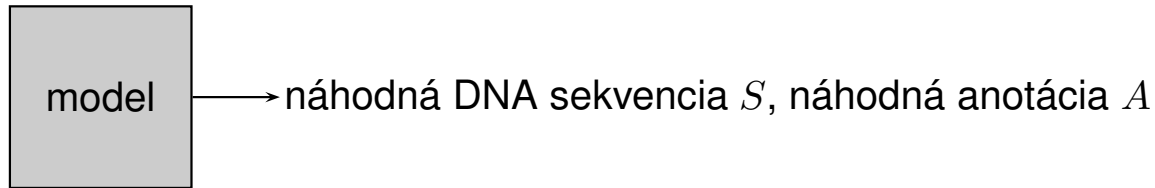
$$F[1, u] = \pi_u \cdot e_{u, S_1}$$

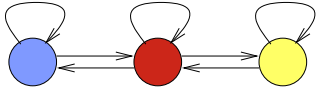
$$F[i, u] = \sum_v F[i-1, v] \cdot a_{v, u} \cdot e_{u, S_i}$$

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

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

Hľadanie génov s HMM



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

- **Trénovanie parametrov:** pravdepodobnosti určíme na základe sekvencií so známymi génmi (**trénovacia množina**).
Model zostavíme tak, aby páry (S, A) s vlastnosťami podobnými skutočným génom mali veľkú pravdepodobnosť $\Pr(S, A)$
- **Použitie:** pre novú sekvenciu S nájdí najpravdepodobnejšiu anotáciu $A = \arg \max_A \Pr(A|S)$ Viterbiho algoritmom

Trénovanie HMM

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

Trénovanie HMM z anotovaných sekvencií

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

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

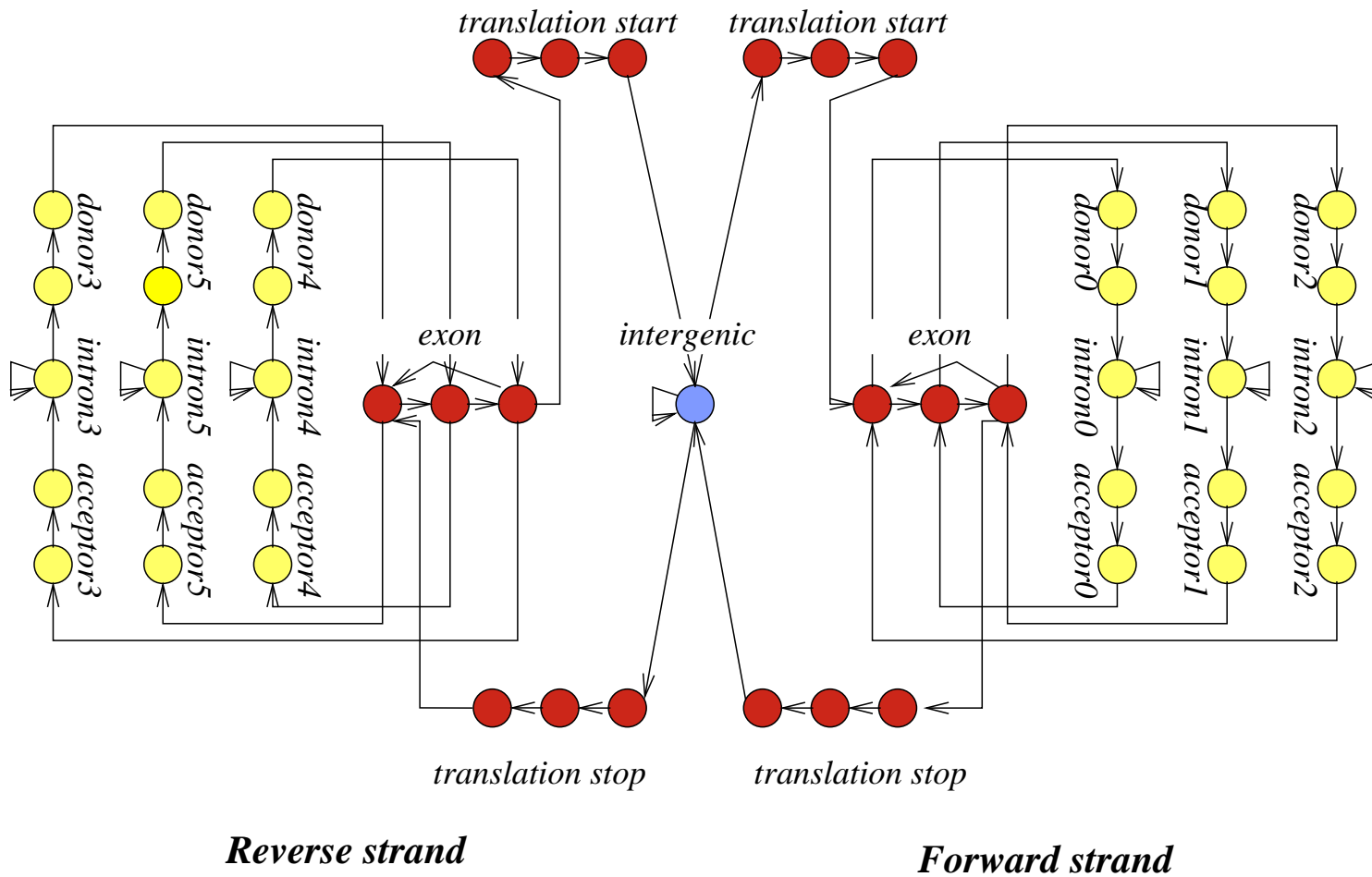
Vstup: topológia modelu a niekoľko tréovacích sekvencií $S^{(i)}$
anotácie $A^{(i)}$ nepoznáme

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).

Tvorba stavového priestoru modelu

Príklad HMM na hľadanie génov



K-means clustering

Broňa Brejová

19.11.2015

Formulácia problému

Vstup: n -rozmerné vektory x_1, x_2, \dots, x_t a počet zhlukov k

Výstup: Rozdelenie vektorov do k zhlukov:

- priradenie vstupných vektorov do zhlukov zapísané ako čísla c_1, c_2, \dots, c_t , kde $c_i \in \{1, 2, \dots, k\}$ je číslo zhľuku pre x_i
- centrum každého zhľuku, t.j. n -rozmerné vektory $\mu_1, \mu_2, \dots, \mu_k$

Hodnoty c_1, \dots, c_t a μ_1, \dots, μ_k volíme tak, aby sme minimalizovali súčet štvorcov vzdialeností od každého vektoru k centru jeho zhľuku:

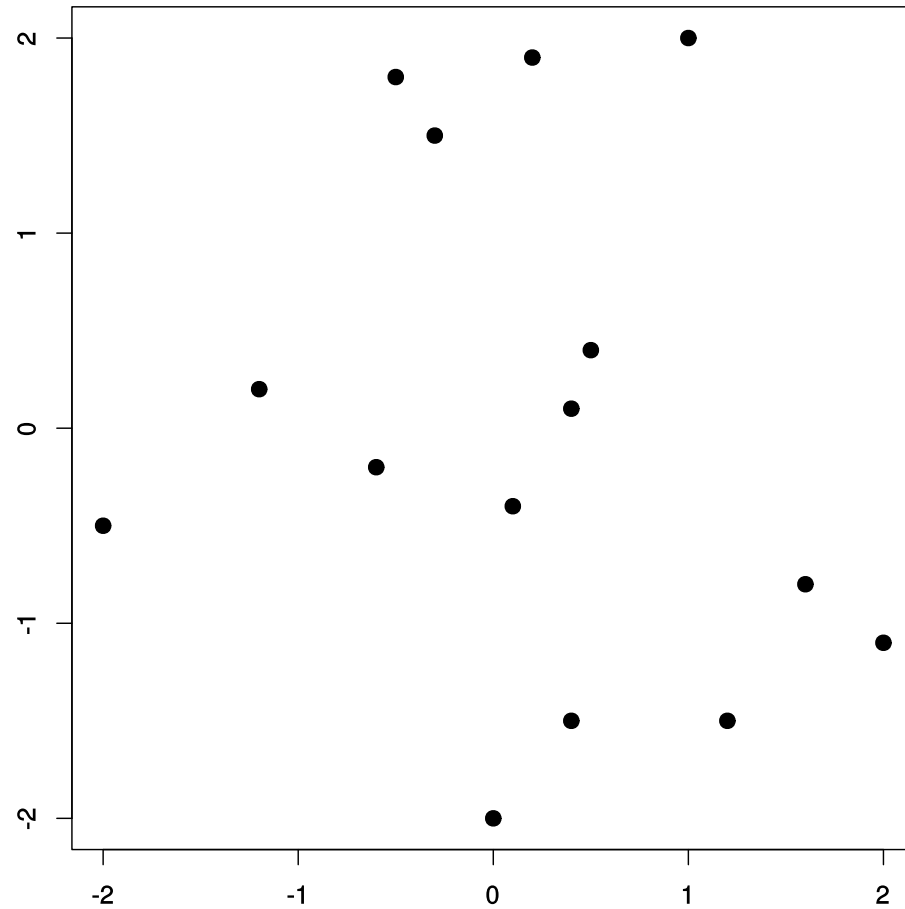
$$\sum_{i=1}^t \|x_i - \mu_{c_i}\|_2^2$$

Pre vektory $a = (a_1, \dots, a_n)$ a $b = (b_1, \dots, b_n)$ je druhá mocnina vzdialenosti $\|a - b\|_2^2 = \sum_{i=1}^n (a_i - b_i)^2$

Príklad vstupu

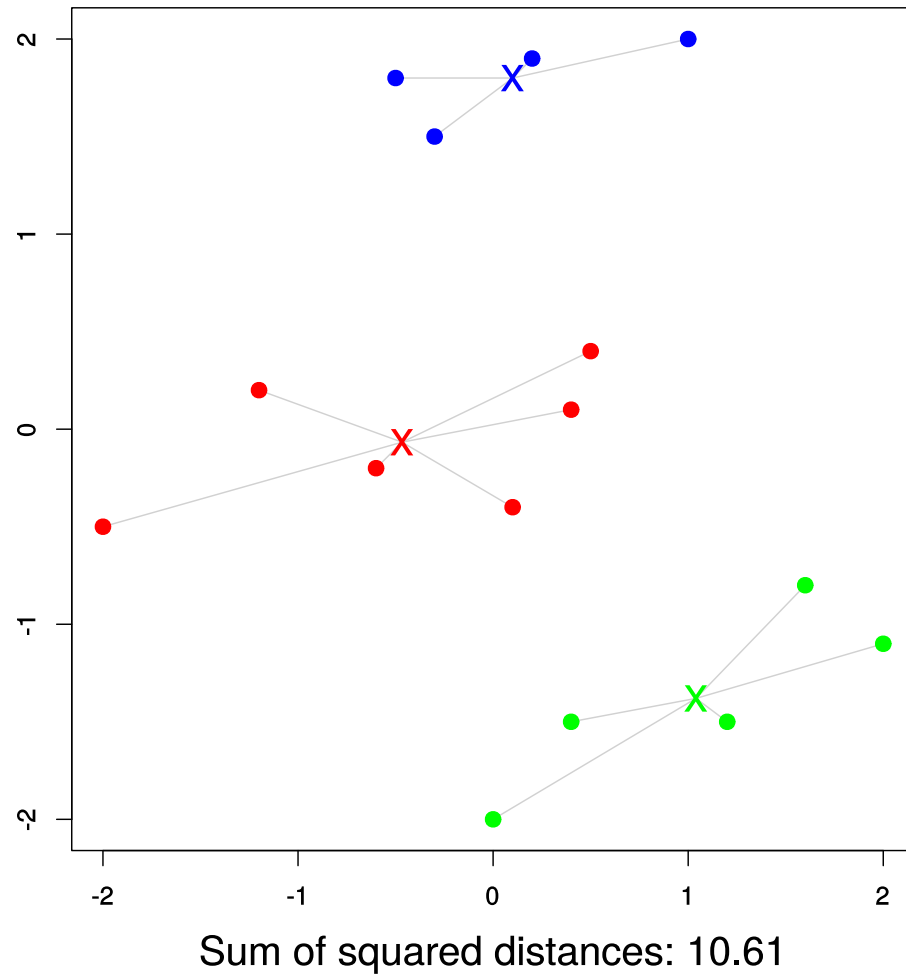
x_1	-2.00	-0.50
x_2	-1.20	0.20
x_3	-0.60	-0.20
x_4	-0.50	1.80
x_5	-0.30	1.50
x_6	0.00	-2.00
x_7	0.10	-0.40
x_8	0.20	1.90
x_9	0.40	0.10
x_{10}	0.40	-1.50
x_{11}	0.50	0.40
x_{12}	1.00	2.00
x_{13}	1.20	-1.50
x_{14}	1.60	-0.80
x_{15}	2.00	-1.10

$$k = 3$$



Príklad výstupu

x_1	-2.00	-0.50	1
x_2	-1.20	0.20	1
x_3	-0.60	-0.20	1
x_4	-0.50	1.80	3
x_5	-0.30	1.50	3
x_6	0.00	-2.00	2
x_7	0.10	-0.40	1
x_8	0.20	1.90	3
x_9	0.40	0.10	1
x_{10}	0.40	-1.50	2
x_{11}	0.50	0.40	1
x_{12}	1.00	2.00	3
x_{13}	1.20	-1.50	2
x_{14}	1.60	-0.80	2
x_{15}	2.00	-1.10	2
μ_1	-0.47	-0.07	
μ_2	1.04	-1.38	
μ_3	0.10	1.80	



Algoritmus

Heuristika, ktorá nenájde vždy najlepšie zhlukovanie.

Začne z nejakého zhlukovania a postupne ho zlepšuje.

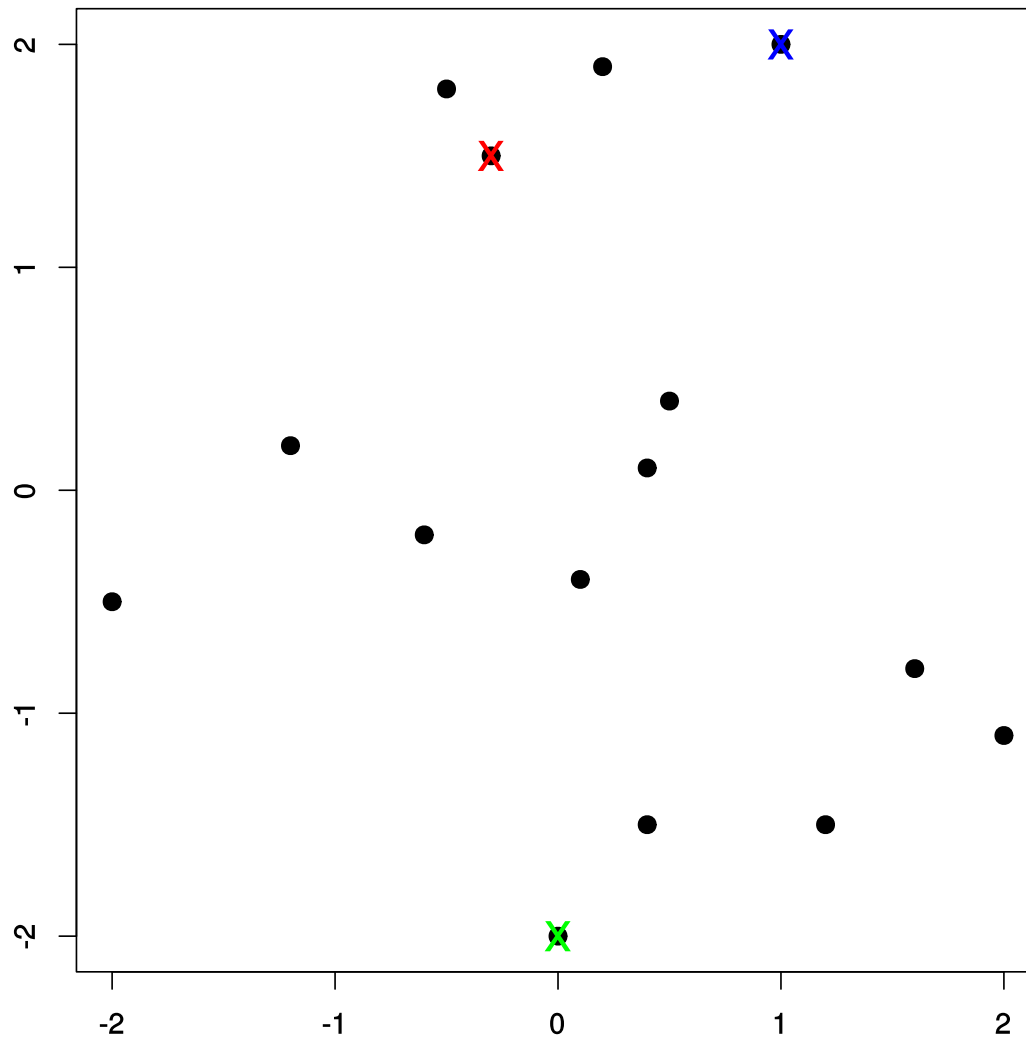
Inicializácia:

náhodne vyber k centier $\mu_1, \mu_2, \dots, \mu_k$ spomedzi vstupných vektorov

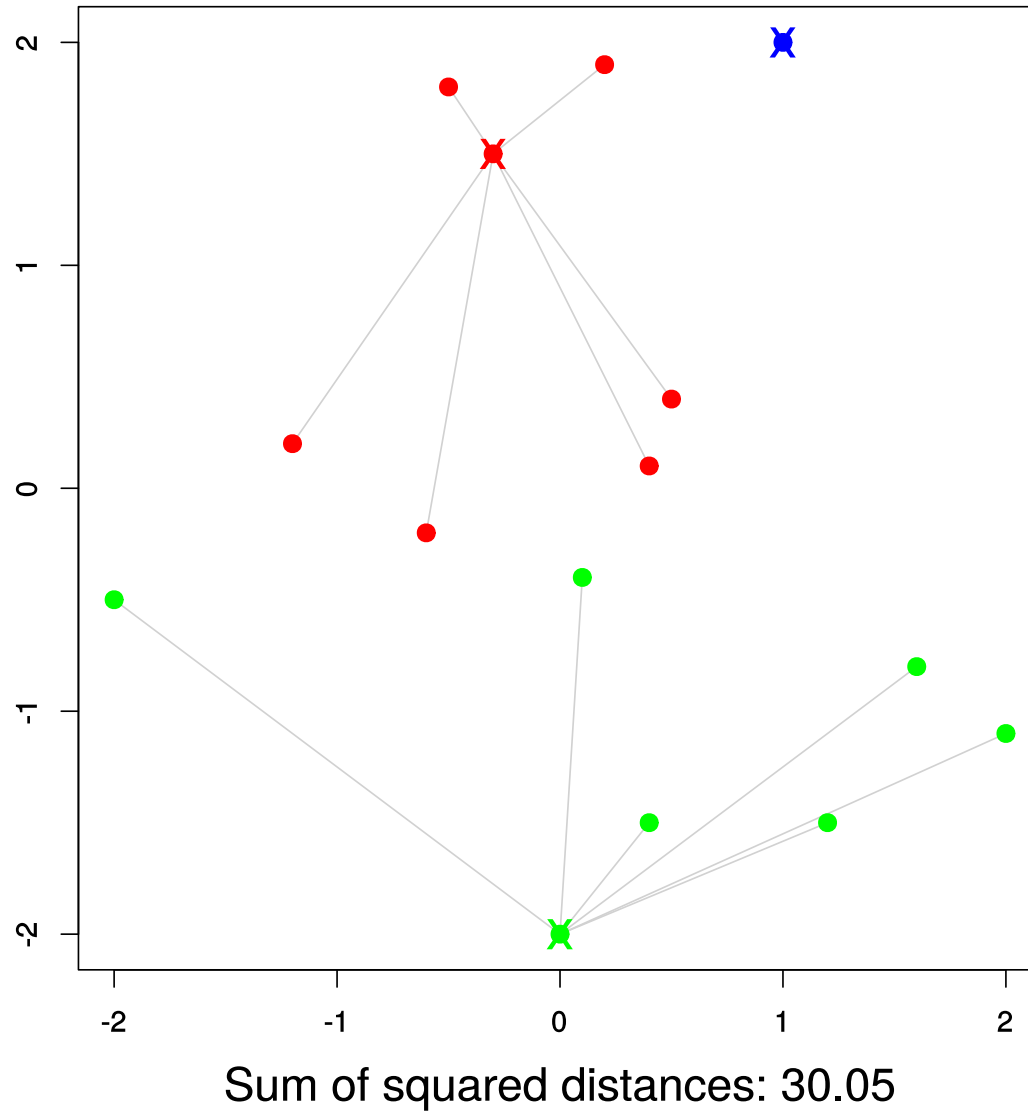
Opakuj, kým sa niečo mení:

- prirad' každý bod najbližšiemu centru: $c_i = \arg \min_j \|x_i - \mu_j\|_2$
- vypočítaj nové centroidy: μ_j bude priemerom (po zložkách) z vektorov x_i , pre ktoré $c_i = j$

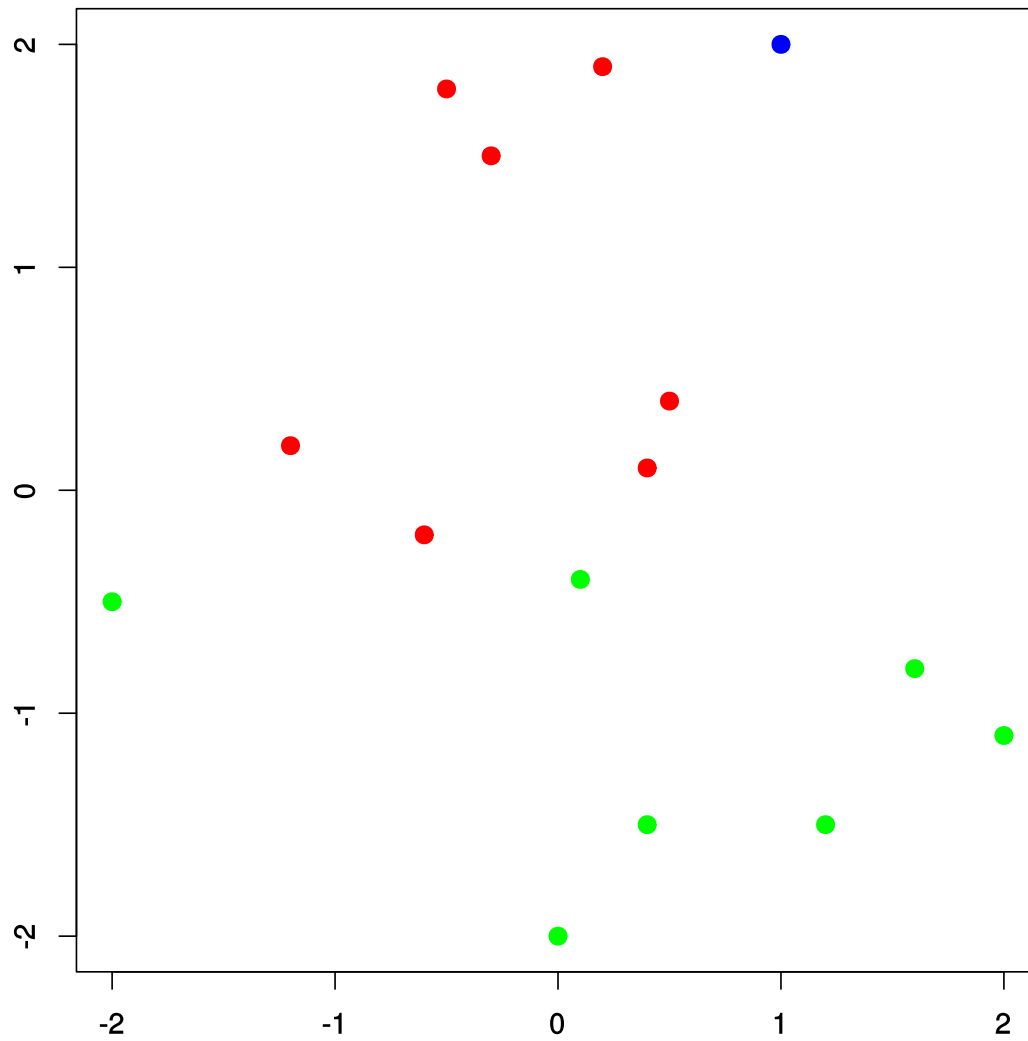
Zvolíme náhodné centrá μ_i



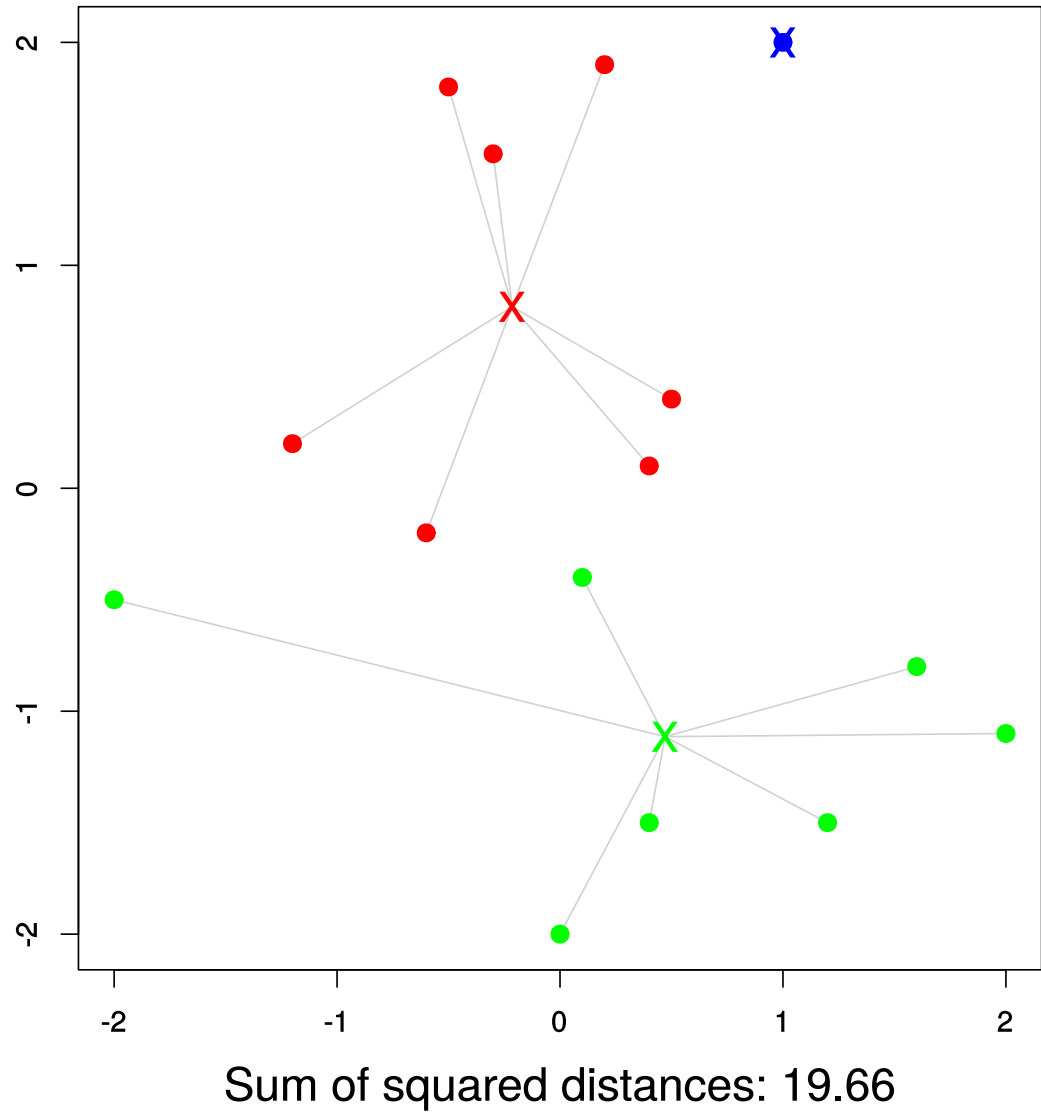
Vektory priradíme do zhlukov (hodnoty c_i)



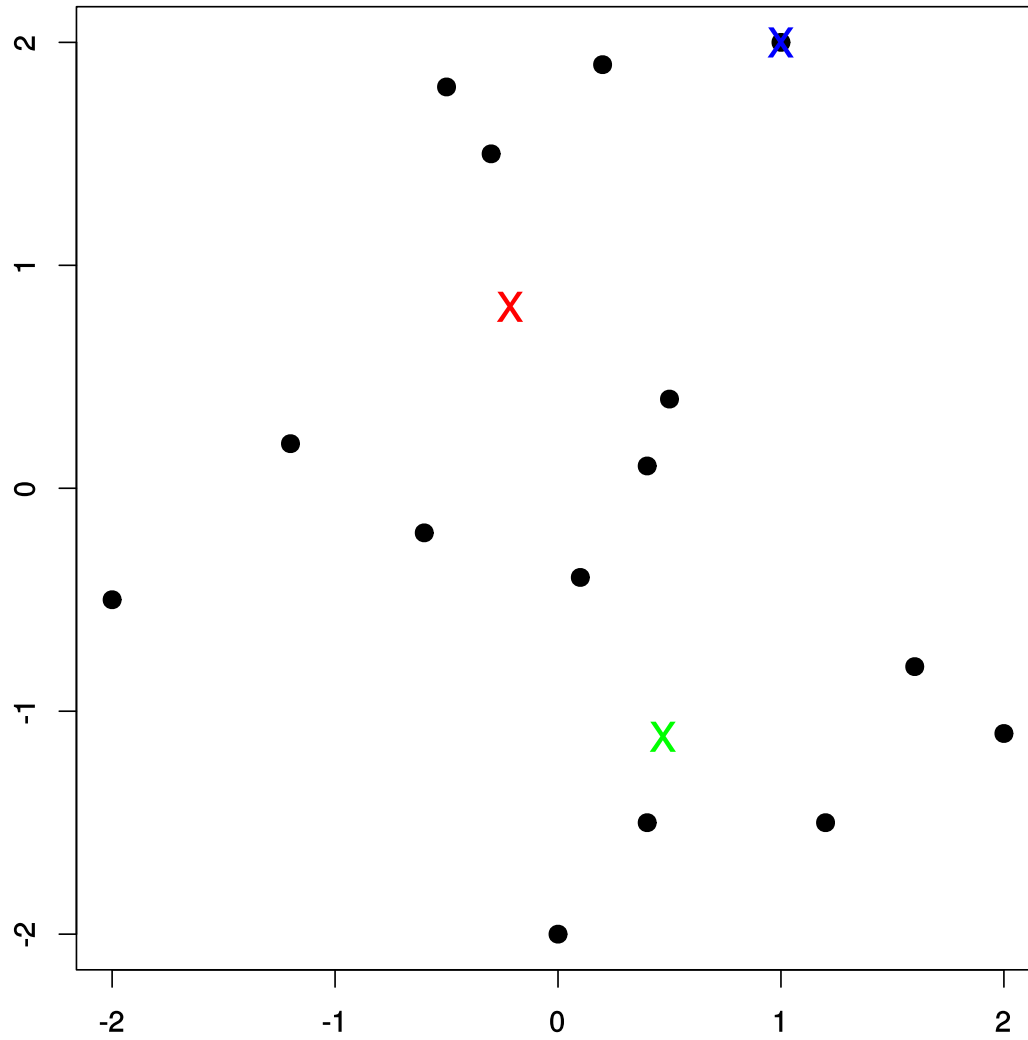
Zabudneme μ_i



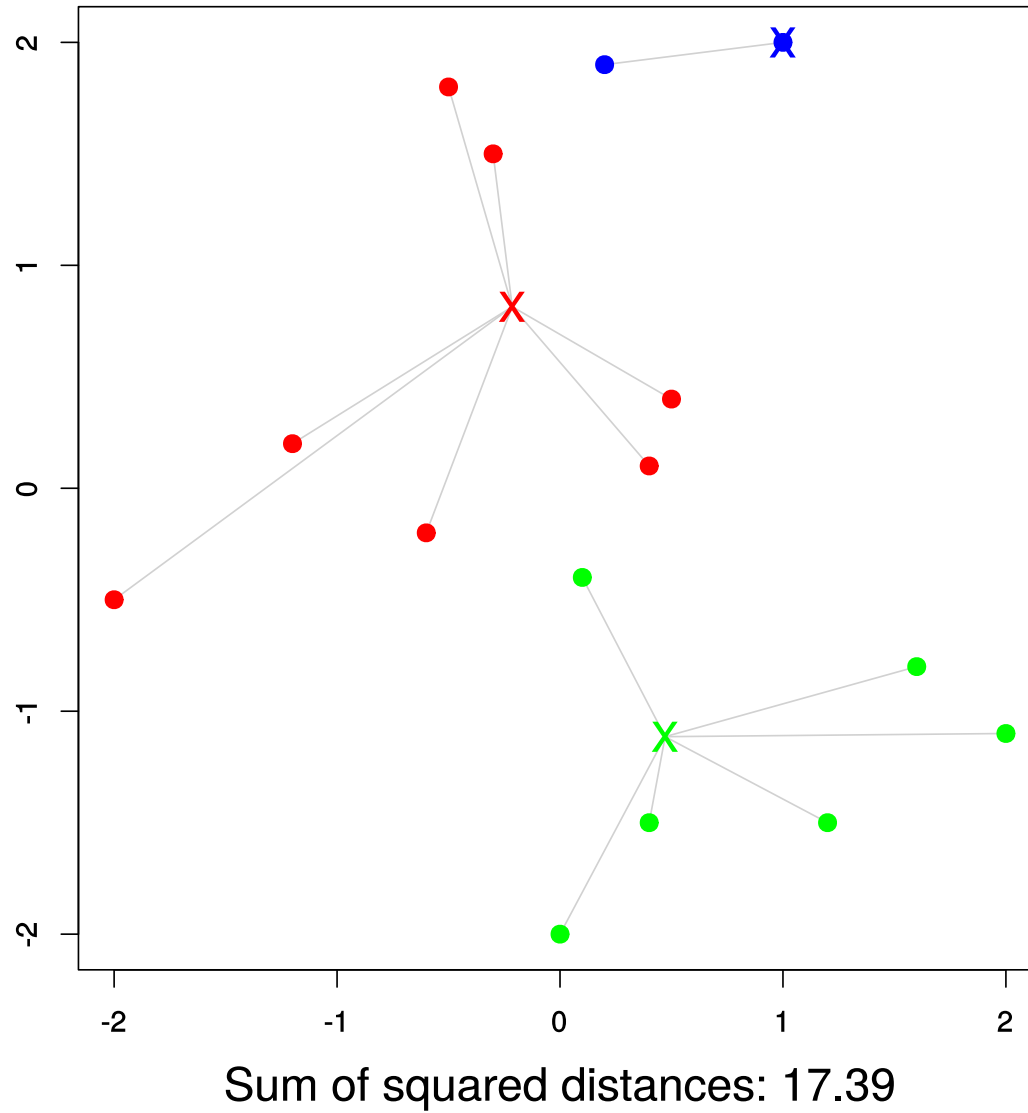
Dopocítame nové μ_i (suma klesla z 30.05 na 19.66)



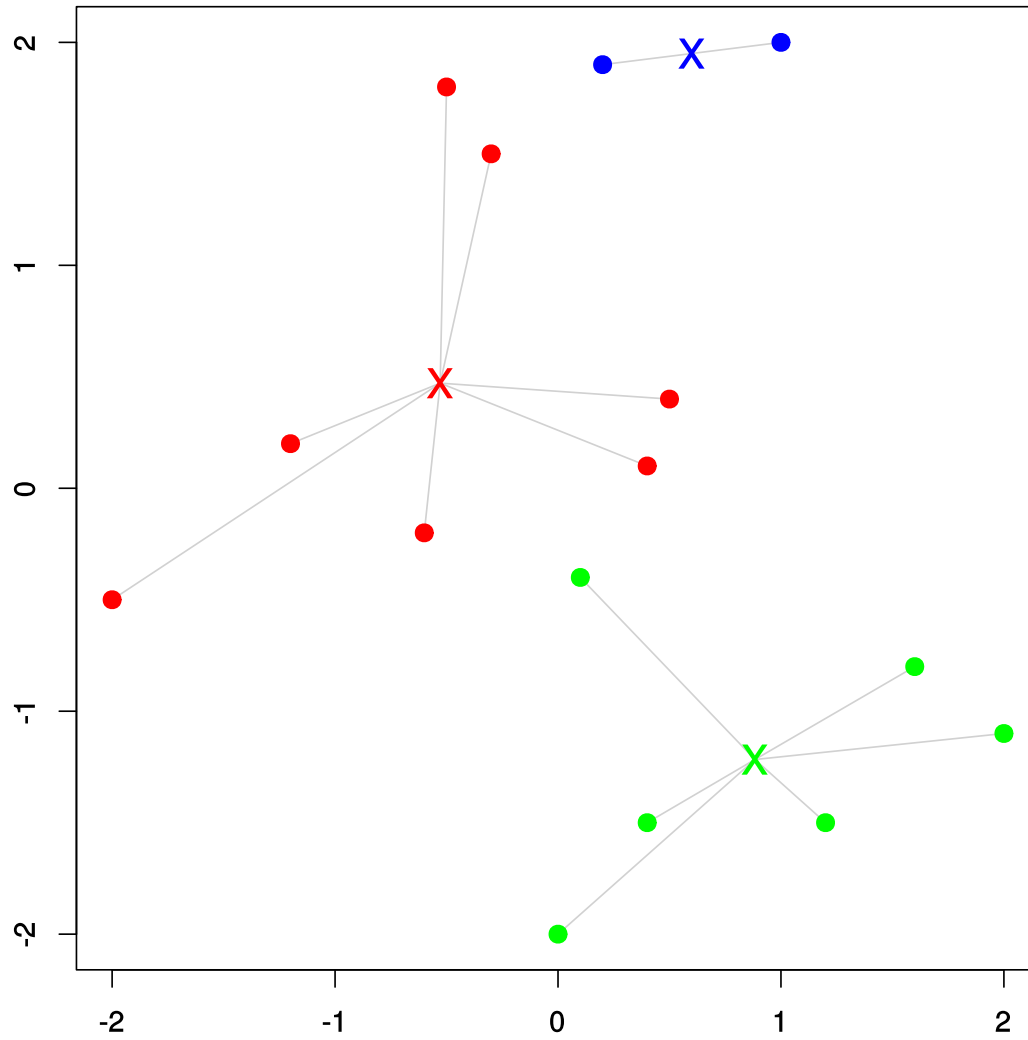
Zabudneme c_i



Dopocítame nové c_i (suma klesla z 19.66 na 17.39)

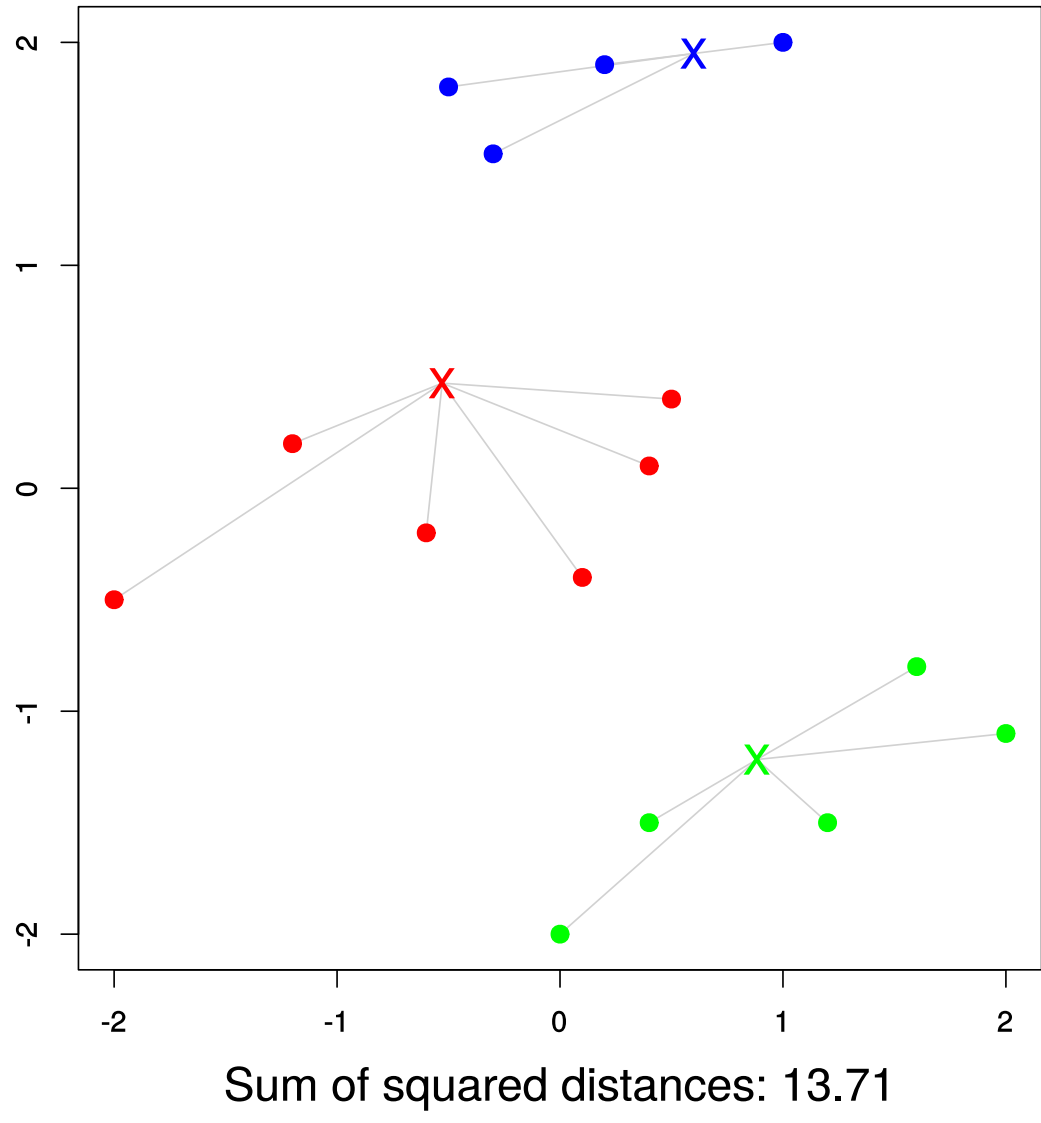


Prepočítame μ_i

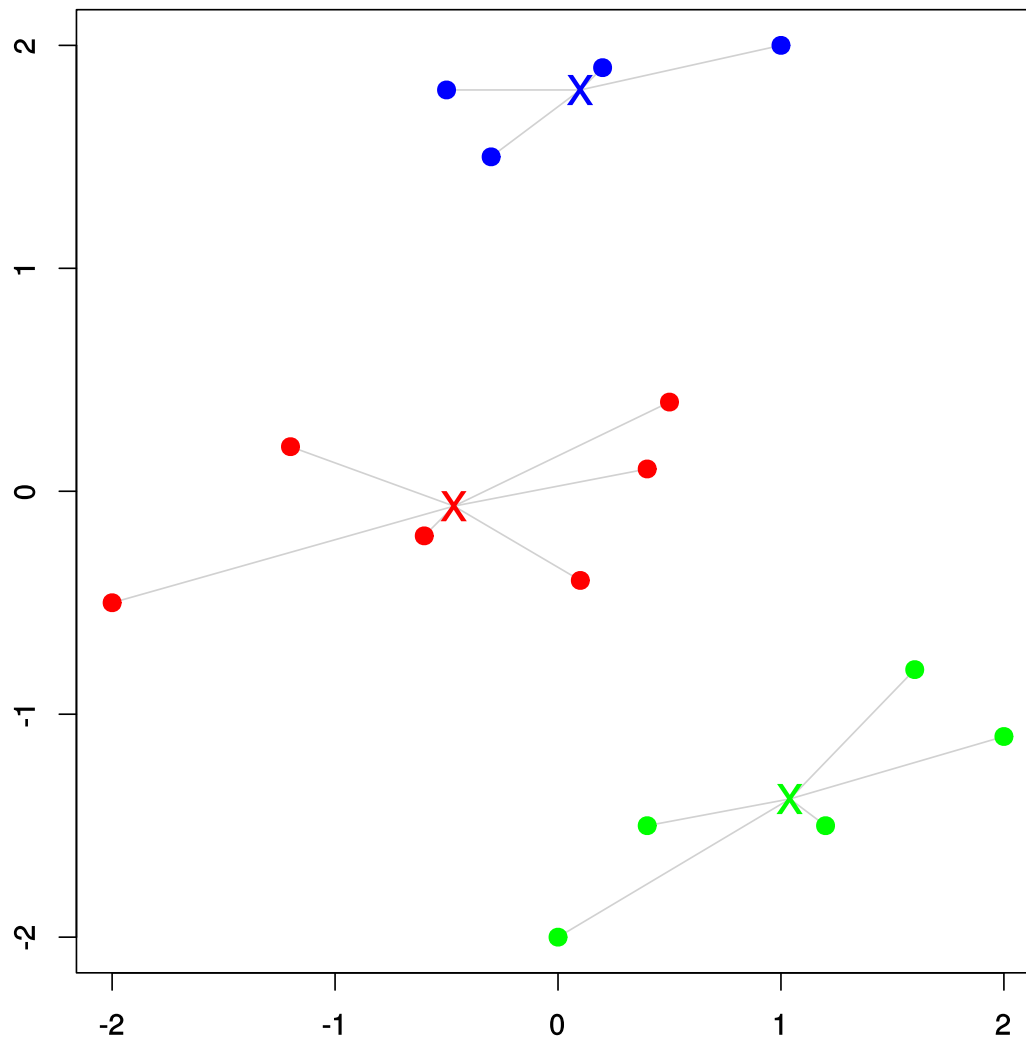


Sum of squared distances: 14.47

Prepočítame c_i

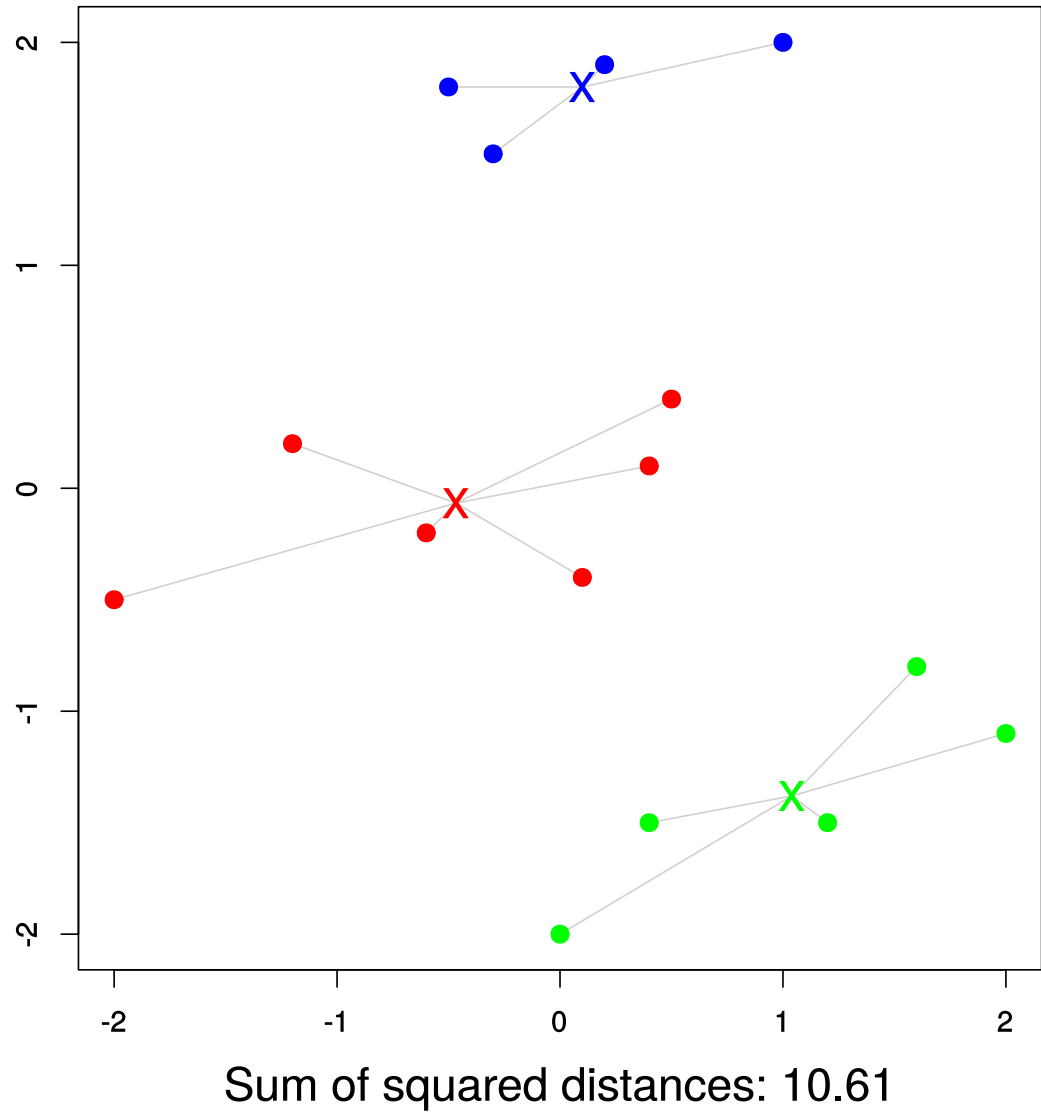


Prepočítame μ_i

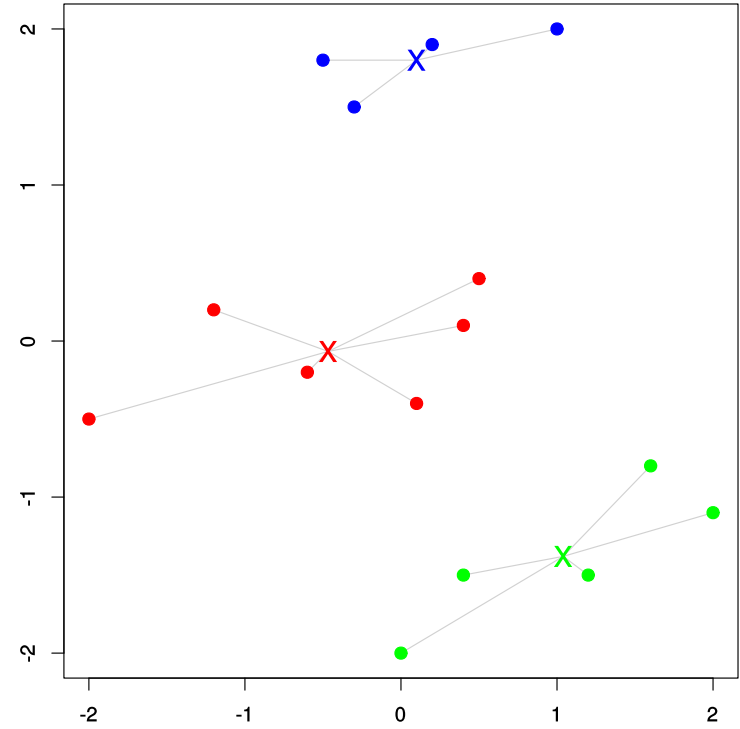
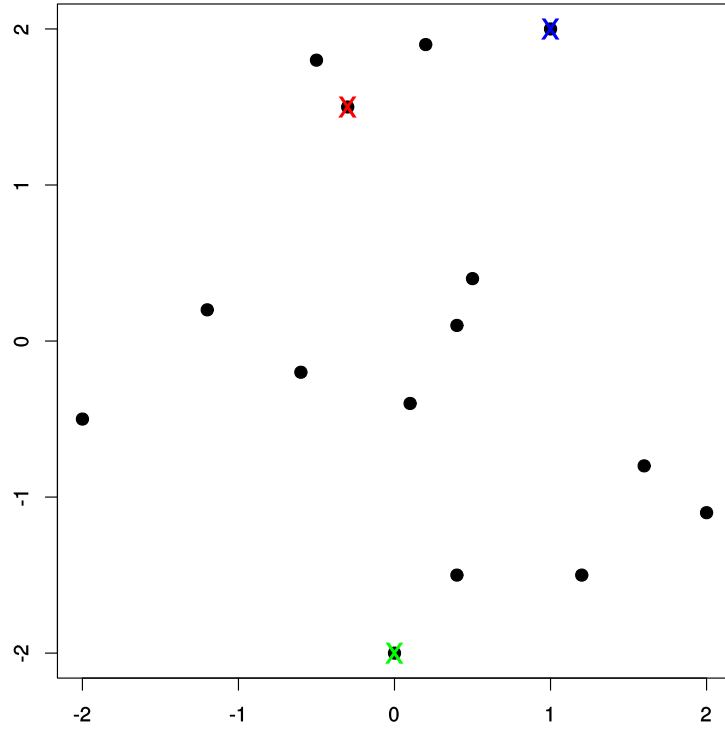


Sum of squared distances: 10.61

Prepočítame c_i (žiadna zmena, končíme)

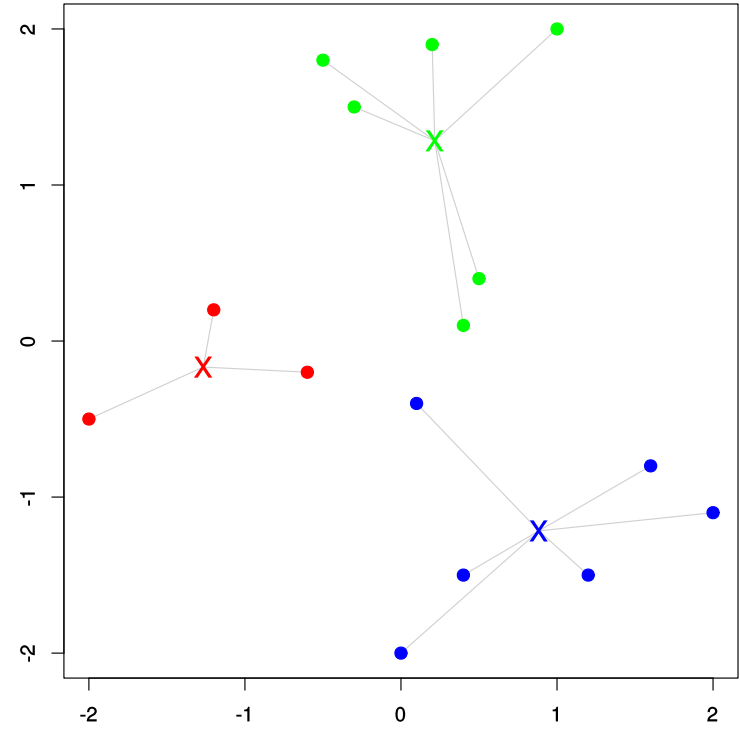
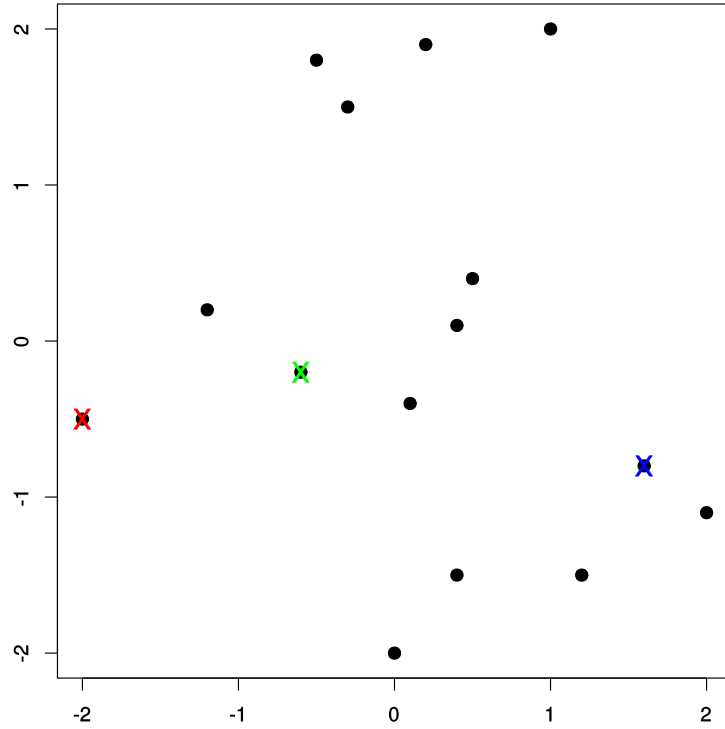


Príklady niekoľkých behov programu



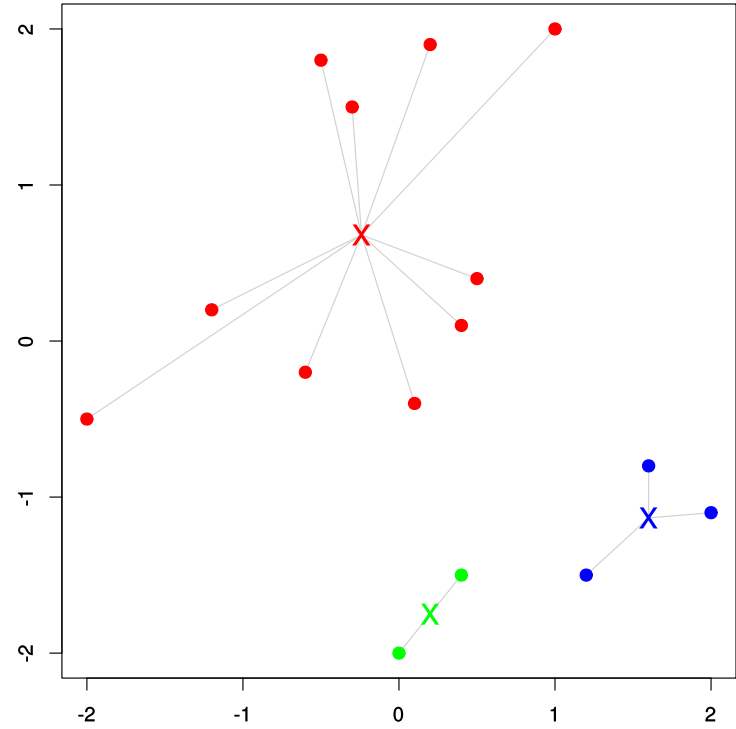
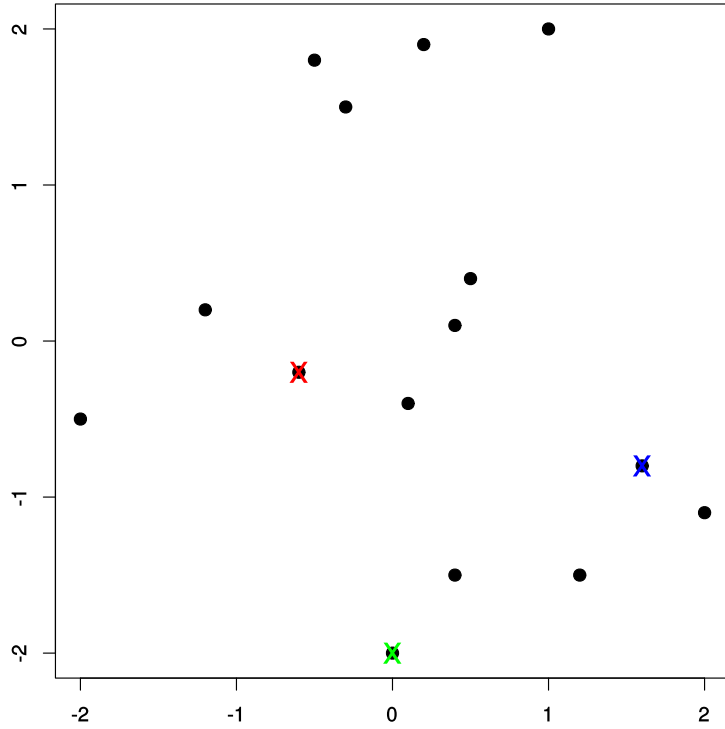
Sum of squared distances: 10.61

Príklady niekoľkých behov programu



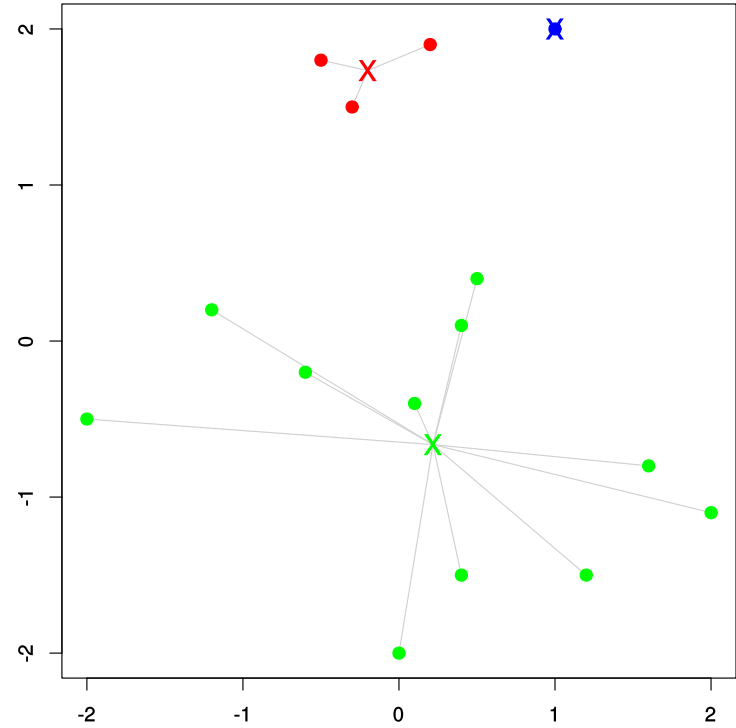
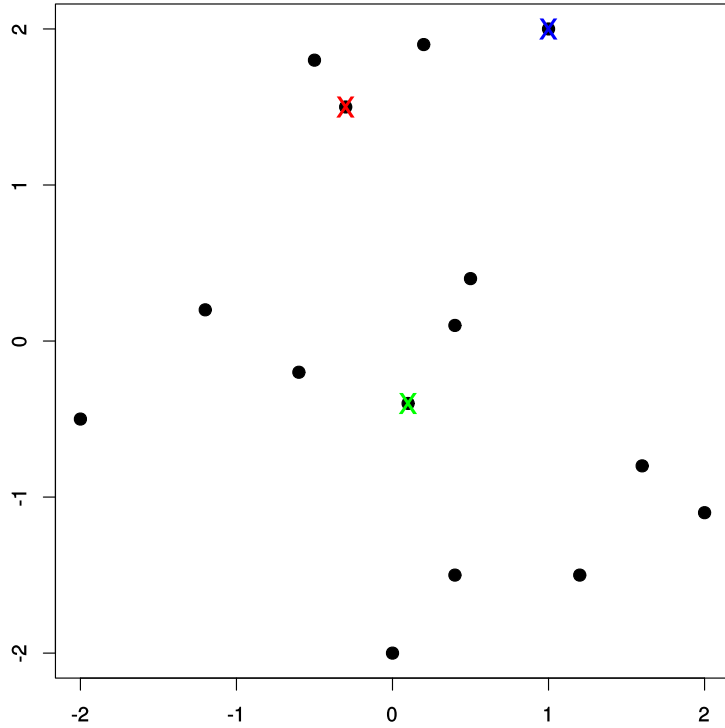
Sum of squared distances: 11.25

Príklady niekoľkých behov programu



Sum of squared distances: 16.93

Príklady niekoľkých behov programu



Sum of squared distances: 20.37

Cvičenia pre informatikov, 10.12.2015
Zhrnutie semestra

Pravdepodobnostné modely

- Skryté Markovove modely (hľadanie génov, konzervovaných oblastí, fylogenetické HMM, profilové HMM)
- Fylogenetické stromy a substitučné modely
- Stochastické bezkontextové gramatiky
- Metóda maximálnej vierohodnosti
- Expectation maximization (EM)

Štatistické metódy

- Pojem štatistickej významnosti
- E-value a P-value
- Test na pozitívny výber
- Linkage disequilibrium, mapovanie asociácií

Precvičenie dynamického programovania

- Zarovnávanie sekvencií
(globálne, lokálne, afínne medzery)
- Skryté Markovove modely (Viterbiho algoritmus)
- Výpočty na stromoch
(úspornosť, vierohodnosť - Felsensteinov algoritmus)
- Hmotnostná spektrometria (MS/MS)
- Sekundárna štruktúra RNA

Iné

- Integer linear programming
- deBruijnové grafy
- Zhlukovanie a klasifikácia

Ako modelovať problémy reálneho sveta

- Rozmyslieť si, aké máme dáta, čo by sme chceli ako výsledok
- Sformulovať ako informatický problém (napr. optimalizácia nejakého skóre)
- Pravdepodobnostné modely nám často dovoľia zvoliť skórovaciu schému systematickým spôsobom
- Výsledný problém často NP ťažký
 - Heuristiky, aproximačne algoritmy
 - ILP a iné techniky na presné riešenie
 - Nedá sa problém trochu preformulovať?
- Testovanie: sú výpočtové výsledky relevantné v danej doméne? (bola formulácia dostatočne realistická?)

Ďalšie predmety

- **Strojové učenie** 2-INF-150, Vinař/Petrovič/Boža (ZS, 4P, 6kr)
- **Grafové modely v strojovom učení** 2-AIN-238, Vinař (LS, 4P, 6kr)
- **Vybrané partie z dátových štruktúr** 2-INF-237, Brejová (LS, 4P, 6kr)
- **Seminár z bioinformatiky (1)-(4)** 2-AIN-50[56],25[12] (oba semestre, 2S, 2kr)
- **Integrácia dátových zdrojov** 2-INF-185 Brejová, Vinař (LS, 1P/2C, 4kr)
- **Biológia** N-bCXX-055/1-BIN-101, Tomáška (ZS, 2P, 2kr)
- **Všeobecná biológia** N-bCXX-085/1-BIN-113, Tomáška (LS, 2P, 2kr)
- **Genomika** N-mCBI-303/2-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/>