Jadrá zarovnaní

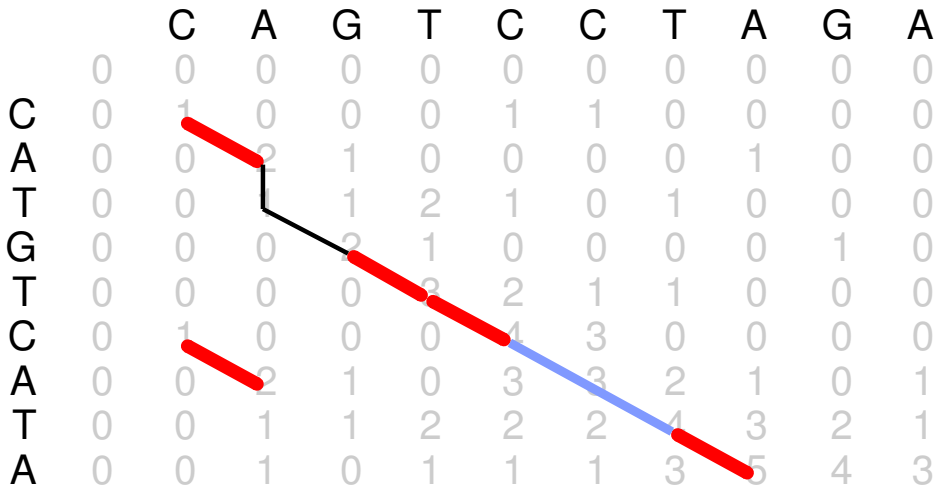
Broňa Brejová

24.10.2024

Opakovanie: Heuristické lokálne zarovnávanie, BLAST

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

(V praxi sa používa $k = 10$ a viac.)



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

Senzitivita heuristického algoritmu

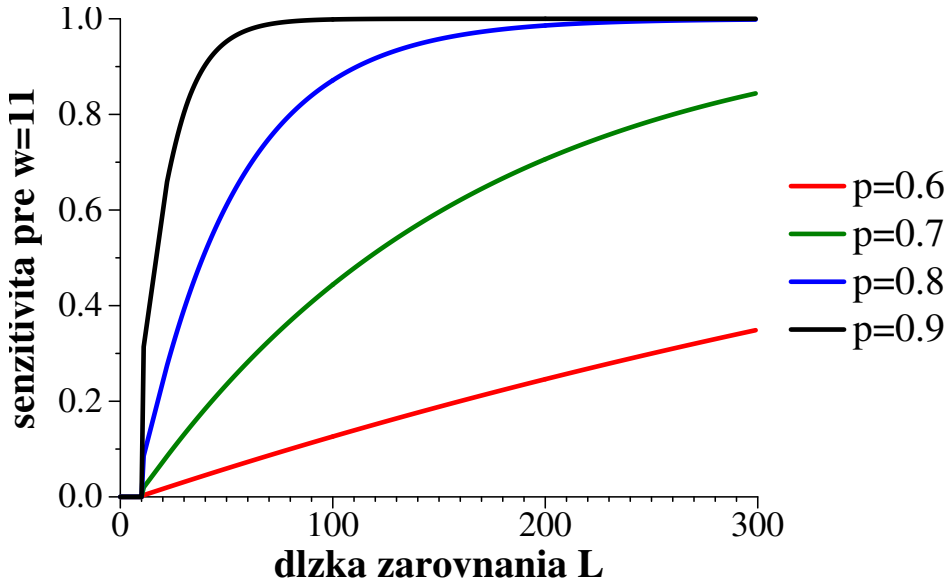
Odhad senzitivity:

Predpokladáme zarovnanie bez medzier, dĺžky L

Každá pozícia je zhoda s pravdepodobnosťou p

Senzitivita:

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



Senzitivita heuristického algoritmu

Predpokladáme zarovnanie bez medzier, dĺžky L

Každá pozícia je zhoda s pravdepodobnosťou p

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

Náhodné premenné:

X_i: je na pozícii i v zarovnaní zhoda?

Y_i: je na pozícii i začiatok jadra?

$Y = \sum_i Y_i$

$f(L,p) = P(Y > 0) = 1 - P(Y = 0)$

Na zamyslenie

$P(Y_i = 1) = ?$

Hodnoty premennej Y?

$E(Y) = ?$

$P(Y = 0) = ?$

Príklad $k = 3$:

```
      AGTGGCTGCCAGGCTGG
      CGAGGCTGCCTGGTTGG
X_i   01011111110110111
Y_i   00011111000000 1
```

Senzitivita heuristického algoritmu

Zarovnanie dĺžky L, pr. zhody p

X_i: je na pozícii i v zarovnaní zhoda?

Y_i: je na pozícii i začiatok jadra?

$Y = \sum_i Y_i$

$P(Y = 0)$ spočítame dynamickým programovaním

$A[n] = \Pr(\text{zarovnanie dĺžky } n \text{ neobsahuje } k \text{ zhôd za sebou})$

Budeme rozlišovať prípady podľa toho, koľko je na konci jednotiek

Príklad $k = 3$:

	AGTGGCTGCCAGGCTGG
	CGAGGCTGCCTGGTTGG
X_i	01011111110110111
Y_i	00011111000001

Opakovanie: ako funguje hľadanie jadier

DB: ulož k-mery do slovníka Query: hľadaj v slovníku

```
AGTGGCTGCCAGGCTGG
cGaGGCTGCCaGGtTGG
```

```
AGTGGCTGCCAGGCTGG
AGTGG,1
 GTGGC,2
  TGGCT,3
   GGCTG,4
    GCTGC,5
     CTGCC,6
      TGCCA,7
       GCCAG,8
        CCAGG,9
         CAGGC,10
          AGGCT,11
           GGCTG,12
            GCTGG,13
```

```
AGGCT,11
AGTGG,1
CAGGC,10
CCAGG,9
CTGCC,6
GCCAG,8
GCTGC,5
GCTGG,13
GGCTG,4,12
GTGGC,2
TGCCA,7
TGGCT,3
```

```
cGaGGCTGCCtGGtTGG
CGAGG,1
 GAGGC,2
  AGGCT,3 -> 11
   GGCTG,4 -> 4,12
    GCTGC,5 -> 5
     CTGCC,6 -> 6
      TGCCT,7
       GCCTG,8
        CCTGG,9
         CTGGT,10
          TGGTT,11
           GGTTG,12
            GTTGG,13
```

Šetrenie pamäťou: BLAT

$k = 5, s = 3$

```
AGTGGCTGCCAGGCTGG
cGaGGCTGCCaGGtTGG
```

```
AGTGGCTGCCAGGCTGG
AGTGG
 GTGGC
  TGGCT,3
   GGCTG
    GCTGC
     CTGCC,6
      TGCCA
       GCCAG
        CCAGG,9
         CAGGC
          AGGCT
           GGCTG,12
            GCTGG
```

```
CCAGG,9
CTGCC,6
GGCTG,12
TGGCT,3
```

```
cGaGGCTGCCtGGtTGG
CGAGG,1
 GAGGC,2
  AGGCT,3
   GGCTG,4 -> 12
    GCTGC,5
     CTGCC,6 -> 6
      TGCCT,7
       GCCTG,8
        CCTGG,9
         CTGGT,10
          TGGTT,11
           GGTTG,12
            GTTGG,13
```

Šetrenie pamäťou: minimizery

$k = 5, s = 4$

```
AGTGGCTGCCAGGCTGG
AGTGG,1
 GTGGC
  TGGCT
   GGCTG
    GCTGC,5
     CTGCC,6
      TGCCA
       GCCAG
        CCAGG,9
         CAGGC,10
          AGGCT,11
           GGCTG
            GCTGG
```

```
AGGCT,11
AGTGG,1
CAGGC,10
CCAGG,9
CTGCC,6
GCTGC,5
```

```
cGaGGCTGCCtGGtTGG
CGAGG
 GAGGC
  AGGCT,3 -> 11
   GGCTG
    GCTGC
     CTGCC,6 -> 6
      TGCCT
       GCCTG
        CCTGG,9
         CTGGT,10
          TGGTT
           GGTTG
            GTTGG
```

BLAST vs BLAT vs minimizery

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 $-k+1$

Nástroj minimap2 (Heng Li 2018):

- $k=15$, $s=10$ nanopórové čítania vs genóm
- $k=15$, $s=5$ prekryvy v nanopórových čítaniach
- $k=19$, $s=10$ porovnanie genómov s 80% zhodami

MinHash

Technika navrhnutá na hľadanie podobných textov, napr. webstránok

Text reprezentujeme ako množinu slov.

Jaccardova miera podobnosti množín:

Množiny $A, B \subseteq U$ (U je univerzum, napr. všetky slová)
$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$

Na zamyslenie:

Aké hodnoty môže $J(A, B)$ nadobudnúť?

Za akých okolností nadobudne extrémy?

Čo by mohli byť “slová” v DNA?

Ako rýchlo spočítame?

Čo ak máme veľa dvojíc A, B?

Odhad Jaccardovej miery vzorkovaním

Chceme odhadnúť $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$

Vzorkujeme $u_1, u_2, \ldots, u_s$ rovnomerne, nezávisle z $A \cup B$

Nech $X_i = 1$ ak u_i patrí do $A \cap B$ a $X_i = 0$ inak

$P(X_i = 1) = ?$

$X = \frac{1}{s} \sum_{i=1}^{s} X_i$

$E(X) = ?$

$Var(X) \leq \frac{1}{4s}$

Nepraktické:

- nevieme rýchlo vzorkovať z $A \cup B$
- nevieme v malej pamäti zistiť, či $u_i \in A \cap B$

Odhad Jaccardovej miery hašovaním (minHash)

Nech h je (náhodná) hašovacia funkcia na U

Považujeme ju za náhodnú permutáciu

$A = \{a_1, a_2, \ldots, a_n\}$ definujeme
$minHash_h(A) := \min\{h(a_1), h(a_2), \ldots, h(a_n)\}$

Nech $X = 1$ ak $minHash_h(A) = minHash_h(B)$ inak 0.
Potom $E[X] = J(A, B) = |A \cap B|/|A \cup B|$

Chceme počítať premenné $X_1, \ldots, X_s$ pre nezávisle zvolené náhodné hašovacie funkcie $h_1, \ldots, h_s$.

MinHash

Výpočet sketchov pre dokumenty:

Zvolíme si "náhodné" hašovacie funkcie $h_1, \ldots, h_s$

Pre každý text $A = \{a_1 \ldots a_n\}$:

Pre každú funkciu h_i z $h_1, \ldots, h_s$:

$$S_{A,i} = \min\{h_i(a_1), h_i(a_2), \ldots, h_i(a_n)\}$$

Porovnávanie sketchov pre dokumenty:

Pre každé dva texty A, B

$x = |\{i : S_{A,i} = S_{B,i}\}|$

x/s je odhad $J(A, B)$

Čas a pamäť?

Program Mash na porovnávanie genómov

Používa $k = 21$, $s = 1000$ (s najmenších v jednej hašovacej funkcii)

sketch má asi 8kb na genóm (genóm má milióny až miliardy nukleotidov)