Zarovnávanie sekvencií, cvičenie pre informatikov

Broňa Brejová

17.10.2024

Opakovanie: ako definujeme problém lokálneho a globálneho zarovnania?

Formulácia problému

Zarovnanie dvoch sekvencií: do každej pridáme niekoľko (aj nula) pomlčiek (medzier) tak, aby mali rovnakú dĺžku.

Skórovanie zarovnania: napr. zhoda +1, nezhoda -1, medzera -1.

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

22 zhôd, 6 nezhôd, 3 medzery $\rightarrow$ skóre 13.

V praxi zložitejšie skórovanie.

Problém 1: globálne zarovnanie (global alignment)

Vstup: sekvencie $X = x_1 x_2 \ldots x_n$ a $Y = y_1 y_2 \ldots y_m$.

Výstup: zarovnanie X a Y s najvyšším skóre.

Problém 2: lokálne zarovnanie (local alignment)

Vstup: sekvencie $X = x_1 x_2 \ldots x_n$ a $Y = y_1 y_2 \ldots y_m$.

Výstup: zarovnania podreťazcov $x_i \ldots x_j$ a $y_k \ldots y_\ell$ s najvyšším skóre.

Príklad lokálneho zarovnania

```
ggcccttggagttgactgtcctgctgctccttgagg
ccattctcagagagaggaagtggcctcattttaatc
cgcttcccacagccttgtcctttccagacccatggg
agagggaggggctgagggtgtggctgagcccaccca
agtcacgcgtcactctgcaggtccctctcccccaag
gccgtggccttgggagcccgtggatcccagtgagtg
acgcctccacccccgccctactcgggcagtttaac
ccttgttgttcacttgcagacatcgtgaacacggcc
cggcccgacgagaaggccataatgacctatgtgtcc
agcttctaccatgccttttcaggagcgcagaaggta
ccgagcagggccaggcaggccctcctcgccgccacc
gcgcaatgccgccgctgcctctcgcctcccgtgctc
acctcatttctcttgcagacggcagtggcctctctc
caactggaagccacccccagctccct...
```

```
tgatgccgaggatgtgttcgtcgagcatccggacga
gaagtccatcacctacgtggtcacctactatcacta
ctttagcaaactcaagcaggagacggtgcagggcat
aagcgtatcggtaaggtggtcggcattgccatggag
aacgacaaaatggtccacgactacgagaacttcaca
agcgatctgctcaagtggatcgaaacgaccatccag
tcgctgggcgagcgggagttcgaaaactcgctggcc
ggcgtccaagggcagttggcccagttctccaactac
cgcaccatcgagaagccgcccaagtttgtggaaaag
ggcaacctcgaggtgctccttttcaccctgcagtcc
aagatgcgggccaacaaccagaagccctacacaccc
aaagagggcaagatgatttcggacatcaacaaggcc
tgggagcgtctggagaaggccgagcacgaacgcgaa
ttggccctgcgcgaggagctcatccg...
```

Vstup

Problém: Lokálne zarovnávanie (local alignment)

```
ggcccttggagttgactgtcctgctgctccttgagg
ccattctcagagagaggaagtggcctcattttaatc
cgcttcccacagccttgtcctttccagacccatggg
agagggaggggctgagggtgtggctgagcccaccca
agtcacgcgtcactctgcaggtccctctcccccaag
gccgtggccttgggagcccgtggatcccagtgagtg
acgcctccacccccgccctactcggcagtttaac
ccttgttgttcacttgcagacatcgtgaacacggcc
cggcccgacgagaaggccataatgacctatgtgtcc
agcttctaccatgcctttttcaggagcgcagaaggta
ccgagcagggccaggcaggccctcctcgccgccacc
gcgcaatgccgccgctgcctctcgcctcccgtgctc
acctcatttctcttgcagacggcagtggcctctctc
caactggaagccacccccagctccct...
```

```
tgatgccgaggatgtgttcgtcgagcatccggacga
gaagtccatcacctacgtggtcacctactatcacta
ctttagcaaactcaagcaggagacggtgcagggcat
aagcgtatcggtaaggtggtcggcattgccatggag
aacgacaaaatggtccacgactacgagaacttcaca
agcgatctgctcaagtggatcgaaacgaccatccag
tcgctgggcgagcgggagttcgaaaactcgctggcc
ggcgtccaagggcagttggcccagttctccaactac
cgcaccatcgagaagccgcccaagtttgtggaaaag
ggcaacctcgaggtgctccttttcaccctgcagtcc
aagatgcgggccaacaaccagaagccctacacaccc
aaagagggcaagatgatttcggacatcaacaaggcc
tgggagcgtctggagaaggccgagcacgaacgcgaa
ttggccctgcgcgaggagctcatccg...
```

Výstup:

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

Dynamické programovanie pre globálne zarovnanie

Podproblém: $A[i,j]$: najvyššie skóre globálneho zarovnania reťazcov $x_1 x_2 \dots x_i$ a $y_1 y_2 \dots y_j$.

Všeobecný prípad, $i > 0$, $j > 0$:

ak $x_i = y_j$ sú zarovnané $A[i,j] = A[i-1,j-1]+1$

ak $x_i \neq y_j$ sú zarovnané $A[i,j] = A[i-1,j-1]-1$

ak x_i je zarovnané s medzerou $A[i,j] = A[i-1,j]-1$

ak y_j je zarovnané s medzerou $A[i,j] = A[i,j-1]-1$

Rekurencia:

$$A[i,j] = \max \begin{cases} A[i-1,j-1] + s(x_i,y_j), \\ A[i-1,j]-1, \\ A[i,j-1]-1 \end{cases}$$

kde $s(x,y) = 1$ ak $x = y$ $\quad s(x,y) = -1$ ak $x \neq y$

Príklad globálneho zarovnania

CATGTCGTA vs CAGTCCTAGA

		C	A	G	T	C	C	T	A	G	A
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
C	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
A	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
T	-3	-1	1	1	?						
G	-4										
T	-5										
C	-6										
G	-7										
T	-8										
A	-9										

$$A[i,j] = \max \begin{cases} A[i-1,j-1] + s(x_i,y_j), \\ A[i-1,j] - 1, \\ A[i,j-1] - 1 \end{cases}$$

Príklad globálneho zarovnania

CATGTCGTA vs CAGTCCTAGA

		C	A	G	T	C	C	T	A	G	A
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
C	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
A	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
T	-3	-1	1	1	2	1	0	-1	-2	-3	-4
G	-4	-2	0	2	1	1	0	-1	-2	-1	-2
T	-5	-3	-1	1	3	2	1	1	0	-1	-2
C	-6	-4	-2	0	2	4	3	2	1	0	-1
G	-7	-5	-3	-1	1	3	3	2	1	2	1
T	-8	-6	-4	-2	0	2	2	4	3	2	1
A	-9	-7	-5	-3	-1	1	1	3	5	4	3

Ako získať zarovnanie?

		C	A	G	T	C	C	T	A	G	A
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
C	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
A	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
T	-3	-1	1	1	2	1	0	-1	-2	-3	-4
G	-4	-2	0	2	1	1	0	-1	-2	-1	-2
T	-5	-3	-1	1	3	2	1	1	0	-1	-2
C	-6	-4	-2	0	2	4	3	2	1	0	-1
G	-7	-5	-3	-1	1	3	3	2	1	2	1
T	-8	-6	-4	-2	0	2	2	4	3	2	1
A	-9	-7	-5	-3	-1	1	1	3	5	4	3

```
CATGTCGT--A
CA-GTCCTAGA
```

Ako presne by sme implementovali?

Ako spočítame maticu spätných šípok B?

Aká je časová a pamäťová zložitosť?

Orientované acyklické grafy (directed acyclic graphs, DAGs)

- Všetky hrany sú orientované, nedá sa chodiť v cykle (ak poslúchame orientáciu hrán)
- **Topologické usporiadanie DAGu?**

Orientované acyklické grafy (directed acyclic graphs, DAGs)

- Všetky hrany sú orientované, nedá sa chodiť v cykle (ak poslúchame orientáciu hrán)
- **Topologické usporiadanie DAGu:** očíslovanie vrcholov tak, aby všetky hrany išli z menšieho čísla do väčšieho
  Dá sa nájsť v $O(|V| + |E|)$
- **Zložitosť hľadania ciest?**

Vstup	Najkratšia cesta z s do t	Najdlhšia cesta z s do t
Graf s cyklami a kladnými hranami	?	?
Graf s cyklami a aj záp. hranami	?	?
DAG	?	?

Orientované acyklické grafy (directed acyclic graphs, DAGs)

- Všetky hrany sú orientované, nedá sa chodiť v cykle (ak poslúchame orientáciu hrán)
- **Topologické usporiadanie DAGu:** očíslovanie vrcholov tak, aby všetky hrany išli z menšieho čísla do väčšieho
  Dá sa nájsť v $O(|V| + |E|)$
- **Zložitosť hľadania ciest**

Vstup	Najkratšia cesta z s do t	Najdlhšia cesta z s do t
Graf s cyklami a kladnými hranami	Dijkstrov alg.[1]	NP ťažké
Graf s cyklami a aj záp. hranami	NP ťažké	NP ťažké
DAG	$O(\|V\| + \|E\|)$	$O(\|V\| + \|E\|)$

[1] $O(|E| + |V| \log |V|)$ s Fibonacciho haldou

Dynamické programovanie pre lokálne zarovnanie

(Smith, Waterman 1981)

Podproblém: $A[i,j]$: najvyššie skóre lokálneho zarovnania reťazcov $x_1 x_2 \dots x_i$ a $y_1 y_2 \dots y_j$, ktoré obsahuje bázy x_i a y_j, alebo je prázdne.

Jeden z reťazcov dĺžky 0: prázdne zarovnanie $A[0,j] = A[i,0] = 0$

Všeobecný prípad, $i > 0$, $j > 0$:

ak x_i a y_j sú zarovnané $A[i,j] = A[i-1, j-1] + s(x_i, y_j)$

ak x_i je zarovnané s medzerou $A[i,j] = A[i-1, j] - 1$

ak y_j je zarovnané s medzerou $A[i,j] = A[i, j-1] - 1$

ak x_i a y_j nie sú časťou zarovnania s kladným skóre $A[i,j] = 0$

Rekurencia:

$$A[i,j] = \max \begin{cases} 0, \\ A[i-1, j-1] + s(x_i, y_j), \\ A[i-1, j] - 1, \\ A[i, j-1] - 1 \end{cases}$$

Príklad lokálneho zarovnania

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

```
CATGTCGTA
CA-GTCCTA
```

Časová zložitosť celého algoritmu $O(nm)$

Zložitejšie skórovanie: afínne skóre medzier

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

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

Penalta za začatie medzery (gap opening cost) o,

Penalta za rozšírenie medzery o jedna (gap extension cost) e.

Medzera dĺžky g má penaltu $o + e(g-1)$.

Zvolíme $o < e$ (t.j. $|o| > |e|$), napr. $o = -3$, $e = -1$.

```
A   -   -   -   T   C   G
A   C   G   C   T   C   C
1  -3  -1  -1   1   1  -1
```

Nesprávny algoritus pre afínne skóre medzier

Penalta za začatie medzery (gap opening cost) $o = -3$,

Penalta za rozšírenie medzery o jedna (gap extension cost) $e = -1$

Predpokladáme $o < e$

$$A[i,j] = \max \begin{cases} A[i-1,j-1] + s(x_i, y_j), \\ A[i-1,j] + c(i-1,j,\uparrow), \\ A[i,j-1] + c(i,j-1,\leftarrow) \end{cases}$$

$c(i,j,s) = e$, ak v políčku $A[i,j]$ máme šípku s

$c(i,j,s) = o$, ak v políčku $A[i,j]$ máme inú šípku

Prečo toto riešenie nefunguje?

Hirshbergov algoritmus 1975

```
optA(l1, r1, l2, r2) { // align X[l1..r1] and Y[l2..r2]
    if(r1-l1 <= 1 ||  r2-l2 <=1)
        solve using dynamic programming
    else {
        k=(r-l+1)/2;
        for (i=0; i<=k; i++)
           compute A[i,*] from A[i-1,*]
        for (i=k+1; i<=r-l+1; i++)
           compute A[i,*], B_k[i,*] from A[i-1,*], B_k[i-1,*]
        k2=B_k[r1-l1-1,r2-l2-1];
        optA(l1, l1+k-1, l2, l2+k2-1);
        optA(l1+k, r2, l2+k2, r2);
    }
}
```