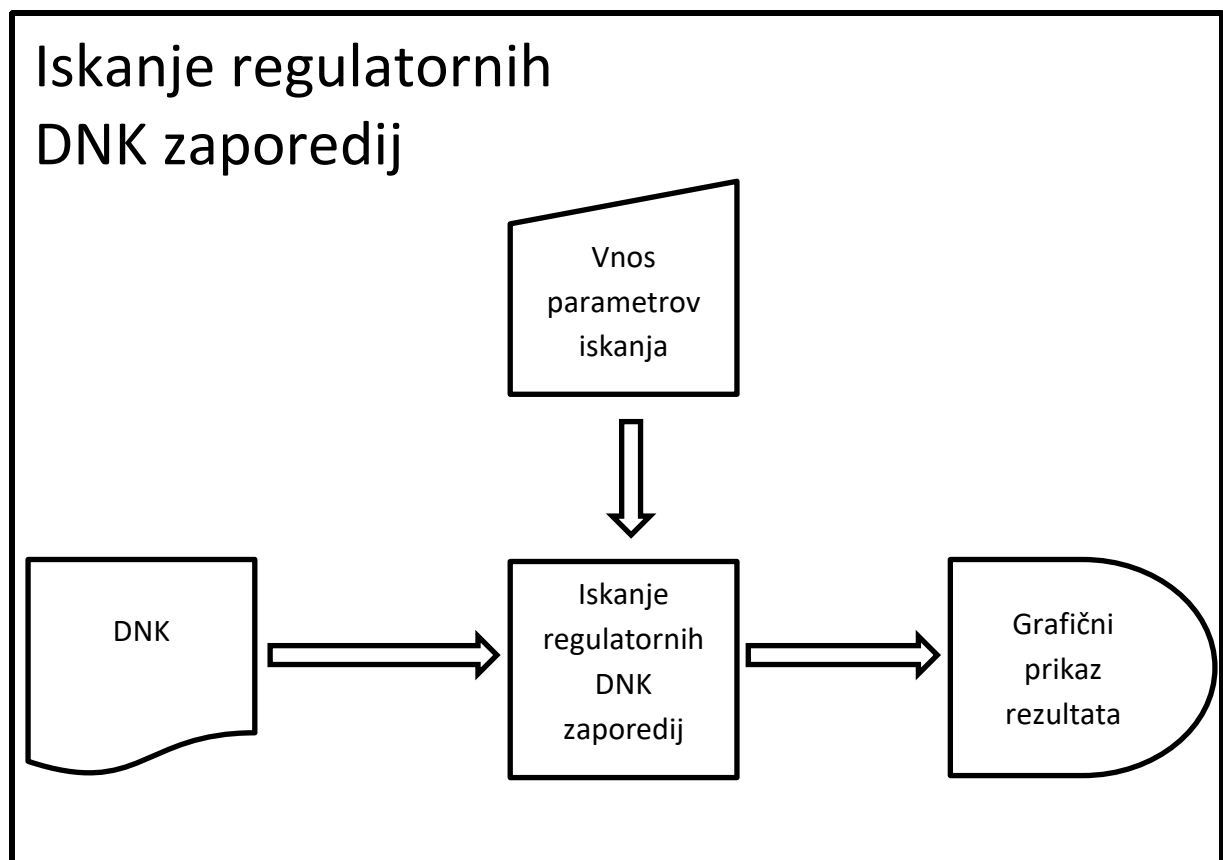


Iskanje regulatornih DNK zaporedij

Izdelajte preprosto aplikacijo za iskanje regulatornih DNK zaporedij. Aplikacija naj omogoča naslednje:

- Branje tekstovne datoteke z zapisom DNK ([DNK1](#))
- Vnos parametrov iskanja (parametre n , l in t)
- Iskanje regulatornih DNK zaporedij
 - Požrešna metoda
 - Razveji in omeji
- Meritve (primerjava hitrosti med obema algoritmoma)



Požrešna metoda:

Vhodni parametri algoritma:

- DNK
- n ; velikost n -merov
- t ; število n -merov
- l ; velikost l -merov

Zaradi velike časovne zahtevnosti algoritma, omejite vhodne parametre:

- $2 \leq l \leq 10$
- $l \leq n \leq 100$
- $2 \leq t \leq 5$

Primer:

DNK = CGGGGCTATGCAACTGGGTGCTCACATTCCCCTTTTCGATATTTGAGGGTGCC ...

$n = 40$

$t = 7$

$l = 8$

DNK razrežemo na kose t n -merov:

$NM_1 = CGGGGCTATGCAACTGGGTGCTCACATTCCCCTTTTCGATA$
 $NM_2 = TTTGAGGGTGCCCAATAAATGCAACTCCAAAGCGGACAAA$
 $NM_3 = GGATGCAACTGATGCCGTTTGACGACCTAAATCAACGGCC$
 $NM_4 = AAGGATGCAACTCCAGGAGCGCCTTTGCTGGTTCTACCTG$
 $NM_5 = AATTTTCTAAAAAGATTATAATGTCGGTCCATGCAACTTC$
 $NM_6 = CTGCTGTACAACTGAGATCATGCTGCATGCAACTTTCAAC$
 $NM_7 = TACATGATCTTTTGATGCAACTTGGATGAGGGAATGATGC$

V vsakem n -meru NM_i ; $1 \leq i \leq t$, izberemo l -mer, ki se začne na odmiku s_j ; $1 \leq s_j \leq n - l + 1$. Polje $S_k = ({}^k s_1, {}^k s_2, \dots, {}^k s_t)$; $1 \leq k \leq (n - l + 1)^t$, določa eno izmed možnih poravnav l -merov, ki jo zapišemo v pripadajočo matriko poravnave. Za matriko poravnave določimo profil $P(S_k)$, iz le tega pa konsenz. Konsenz ovrednotimo s funkcijo vsote maksimalnih vrednosti v matriki profila $P(S_k)$:

$$Score(S_k) = \sum_{j=1}^l \max_{1 \leq i \leq 4} (P_{i,j}(S_k)),$$

kjer je $P_{i,j}(S_k)$ element v i vrstici in j stolpcu profila $P(S_k)$. Konsenz ovrednotimo za vsako poravnavo. Rezultat je poravnava $S_m = ({}^k s_1, {}^k s_2, \dots, {}^k s_t)$, tako da velja:

$$Score(S_m) = \max_{1 \leq k \leq (n-l+1)^t} (Score(S_k))$$

l nukleotidov

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|
| ... | A | G | G | G | T | G | C | C | C | A | A | T | A | A | G | G | G | T | C | G | T | C | A | C | A | T | T | C | ... | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | T | A | T | A | A | T | G | T | C | G | G | T | C | C | T | T | G | G | A | A | C | T | T | C | | | | | | |
| ... | T | G | A | G | A | T | C | A | T | G | C | T | G | C | A | T | G | C | C | A | T | T | T | T | C | A | A | C | | |
| | T | A | C | A | T | G | A | T | C | T | T | T | T | G | A | T | G | G | A | T | G | A | G | G | G | A | A | T | G | ... |

profil

konsenz

| | | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| A | 5 | 1 | 0 | 0 | 5 | 5 | 0 | 0 |
| T | 1 | 5 | 0 | 0 | 0 | 1 | 1 | 6 |
| G | 1 | 1 | 6 | 3 | 0 | 1 | 0 | 0 |
| C | 0 | 0 | 1 | 4 | 2 | 0 | 6 | 1 |
| | A | T | G | C | A | A | C | T |

Razveji in omeji:

Drugi pristop temelji na Hammingovi razdalji. Hammingovo razdaljo $dH(u, v)$ med dvema nizoma u in v enake dolžine (l -mer) je število, ki nam pove na kolikih pozicijah se niza u in v razlikujeta. Primer izračuna Hammingove razdalje:

$$\begin{array}{rcccccc} u = & A & T & T & G & T & C \\ & & X & & X & & \\ v = & A & C & T & C & T & C \end{array} \quad dH(u, v) = 2$$

Ideja algoritma je, da obravnavamo vsak l -mer kot možen konsenz KZ_j ; $1 \leq j \leq 4^l$, kjer je 4^l možnih konsenzov (generiramo vse možne rešitve dolžine l).

Vsakemu n -meru poiščemo l -mer z najkrajšo Hammingovo razdaljo dH do KZ_j .

$$dH_i(KZ_j) = \min_{1 \leq k \leq (n-l+1)} (dH(KZ_j, {}^k s_i)), 1 \leq i \leq t$$

Pripadajoč položaj minimuma označimo z $s_i(KZ_j)$. Po obhodu vseh n -merov imamo za dani konsenz KZ_j matriko poravnave $S(KZ_j) = (s_1(KZ_j), \dots, s_t(KZ_j))$.

Skupno razdaljo dH za poravnavo $S(KZ_j)$ do izbranega konsenza KZ_j izračunamo z naslednjo enačbo

$$dH(KZ_j) = \sum_{i=1}^t dH_i(KZ_j)$$

Sedaj še poiščemo najmanjšo skupno razdaljo dH po vseh možnih konsenzih KZ_j z enačbo:

$$dH(DNK) = \min_{1 \leq j \leq 4^l} (dH(KZ_j))$$

Združimo vse skupaj dobimo:

$$dH(DNK) = \min_{1 \leq j \leq 4^l} \left(\sum_{i=1}^t \min_{1 \leq k \leq (n-l+1)} (dH(KZ_j, {}^k s_i)) \right)$$

Razveji in omeji:

- Listi drevesa ustrezajo l -merom (možnim konsenzom), notranja vozlišča pa predponam vozlišč v poddrevesih.
- V kolikor je skupna Hammingova razdalja v kakšnem notranjem vozlišču višja (ali enaka) od trenutno najboljšega rezultata v katerem od že obiskanih listov, nam ni treba preiskovati poddrevesa

| REŠITVE | | | | |
|---------|----|---|-----------------|------------------|
| l | n | t | Požrešna metoda | Razveji in omeji |
| 3 | 10 | 2 | CAA (5) | AGC (1) |
| 5 | 15 | 2 | CAAAT (10) | CAAAT (0) |
| 7 | 20 | 2 | CAAATGA (12) | CAAATGA (2) |
| 3 | 10 | 3 | CAA (8) | CAA (1) |
| 5 | 15 | 3 | CAAAT (13) | AAATG (2) |
| 7 | 15 | 3 | CAAATGA (16) | AGATGTC (5) |
| 7 | 20 | 3 | CAAATGC (18) | CAAATGC (3) |
| 3 | 10 | 4 | CAA (10) | CAA (2) |
| 5 | 15 | 4 | CAAAT(17) | AAATG (3) |
| 7 | 15 | 4 | CAAATGC (22) | CAAATGC (6) |
| 7 | 20 | 4 | TTCCAAG (23) | TTCCAAG (5) |
| 3 | 10 | 5 | CAA (12) | CAA (3) |
| 5 | 15 | 5 | CAAAT (20) | AAATG (5) |

| Fukcionalnost: | Točke: |
|--|-----------|
| Branje DNK, Rešitev (poravnava, profil, konsenz, ovrednotenje konsenza), Meritve | 1 |
| Požrešna metoda | 3 |
| Razveji in omeji | 6 |
| Skupaj | 10 |