

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 2321

**SWIG - Poravnanje struktura  
korištenjem iterativne primjene  
Smith-Waterman algoritma**

Bruno Rahle

Zagreb, svibanj 2012.

*Umjesto ove stranice umetnite izvornik Vašeg rada.  
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.*



# SADRŽAJ

<b>Popis slika</b>	<b>v</b>
<b>Popis tablica</b>	<b>vi</b>
<b>1. Uvod</b>	<b>1</b>
<b>2. Poravnavanje sturktura</b>	<b>2</b>
<b>3. Smith-Watermanov algoritam</b>	<b>3</b>
3.1. Needleman-Wunschov algoritam . . . . .	3
3.1.1. Ulazni podaci . . . . .	3
3.1.2. Izlazni podaci . . . . .	4
3.1.3. Algoritam . . . . .	4
3.1.4. Primjer . . . . .	4
3.1.5. Analiza složenosti . . . . .	4
<b>4. Algoritam simuliranog kaljenja</b>	<b>5</b>
<b>5. CUDA tehnologija</b>	<b>6</b>
<b>6. Implementacija</b>	<b>7</b>
<b>7. Rezultati</b>	<b>8</b>
<b>8. Zaključak</b>	<b>9</b>
<b>Literatura</b>	<b>10</b>

# POPIS SLIKA

# POPIS TABLICA

# 1. Uvod

Problemi s kojima se znanost danas susreće često prelaze granice isključivo jedne discipline.

- pričaj o: bioinformatici, koje probleme ona rješava, zašto je problem koji si rješavao bitan, kome koristi, kako se može koristiti i potencijalno za što se sve može koristiti. - napiši zašto je ovo što si napravio jebeno i kako se može još poboljšati. - cca 2 stranice

## **2. Poravanavanje sturktura**

- detaljan opis problema - cca 1-2 stranice



## 3. Smith-Watermanov algoritam

(cca 4-5 stranica)

Smith-Watermanov algoritam služi nam da bismo pronašli lokalno poravnanje. Osmislili su ga Temple F. Smith i Michael S. Waterman 1981. Smith (1981). Temelji se na Needleman-Wunschevom algoritmu (Needleman (1970)) te i sam spada u kategoriju algoritama dinamičkog programiranja.

- napisi jos teksta ovdje.

### 3.1. Needleman-Wunschov algoritam

Algoritam, kao što je već rečeno, traži globalno poravnanje. To znači da se svi članovi ulaznih nizova moraju poravnati. Dopusćene operacije kada tražimo poravnanje su preklapanje s elementom iz suprotnog niza i ubacivanje praznina u neki od nizova. Sve parove elemenata u dobivenom preklapanju bodujemo i na osnovu te ocjene određujemo sličnost nizova. Konačan rezultat ovog algoritma jest poravnanje koje maksimizira takvu ocjenu, tj. daje maksimalno globalno poravnanje.

Zanimljivost je da je to prvi algoritam dinamičkog programiranja ikada primjenjen u bioinformatici.

#### 3.1.1. Ulazni podaci

1. Dva niza ( $A$  i  $B$ ) proteina ili nukleotida. Zbog jednostavnosti, pretpostavit ćemo da su to nukleotidi iz DNK - adenin (A), timin (T), gvanin (G) i citozin (C). U primjeru ćemo koristiti  $A = \text{"ATGCCGTA"}$  i  $B = \text{"TGCCTA"}$ . Dužinu niza  $A$  označit ćemo s  $N$ , a dužinu niza  $B$  s  $M$ .
2. Matrica  $S$ , koja nam daje bodove koje dobijemo kada jedan nukleotid preklopimo s drugim. U našem će slučaju imati dimenzije  $4 \times 4$ , budući da ćemo razmatrati slučaj kada imamo samo četiri nukleotida. U principu će na dijagonali imati pozitivne brojeve, a na ostalim poljima negativne. To znači da nam se najviše

isplati preklapati nukletide istog tipa, jer za to dobivamo bodove, a inače gubimo bodove. Primjer jedne takve matrice koju ćemo koristiti i u primjeru:

	A	C	G	T
A	10	-3	-9	-1
C	-5	8	-8	-7
G	-5	-4	7	-5
T	-4	-11	-8	9

3. Negativan broj  $d$ , koji označava bodove koje dobijemo (tj. izgubimo) kada nukleotid preklopimo s prazninom.

### 3.1.2. Izlazni podaci

1. Broj  $H$ , ocjena najboljeg globalnog poravnanja.
2. Dva nova niza jednake dužine,  $A'$  i  $B'$ , nastala ubacivanjem praznina (označenih najčešće sa '-') u nizove  $A$  i  $B$  koja predstavljaju najbolje pronađeno poravnanje.

### 3.1.3. Algoritam

Neka nam matrica  $F_{i,j}$  označava maksimalan broj bodova koje možemo dobiti kada poravnamo prvih  $i$  članova niza  $A$  i prvih  $j$  članova niza  $B$ .  $F_{i,j}$  možemo računati rekurzijom na slijedeći način:

$$F_{i,j} = \begin{cases} 0 & \text{ako je } i = 0 \text{ i } j = 0 \\ i \cdot d & \text{ako je } i > 0 \text{ i } j = 0 \\ j \cdot d & \text{ako je } i = 0 \text{ i } j > 0 \\ \max \begin{pmatrix} F_{i-1,j-1} + S_{A_{i-1},B_{j-1}} \\ F_{i-1,j} + d \\ F_{i,j-1} + d \end{pmatrix} & \text{ako je } i > 0 \text{ i } j > 0 \end{cases}$$

### 3.1.4. Primjer

### 3.1.5. Analiza složenosti

## **4. Algoritam simuliranog kaljenja**

- objasni i zašto si uzeo više nizova odjednom, kako od njih biraš najboljeg i slično. -  
cca 4 starnice

## **5. CUDA tehnologija**

- cca 2 stranice
- prednosti i mane, problemi i rjesenja

## 6. Implementacija

- cca 5 stranica
  - kako je sve spojeno
  - detalji implementacije - koje funkcije ocjene sam koristio, kako sam došao do njih

## **7. Rezultati**

- cca 1-2 starnice

## **8. Zaključak**

Zaključak. - cca 1-2 stranice

# LITERATURA

Christian D. Needleman, Saul B.; Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins, 1970.

Michael S. Smith, Temple F.; Waterman. Identification of common molecular subsequences, 1981.



**SWIG - Poravnanje struktura korištenjem iterativne primjene Smith-Waterman  
algoritma**

**Sažetak**

Sažetak na hrvatskom jeziku.

**Ključne riječi:** Ključne riječi, odvojene zarezima.

**Title**

**Abstract**

Abstract.

**Keywords:** Keywords.