SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

Bioinformatika

**Projekt**

**Improving Bloom Filter Performance on Sequence Data Using k-merBloom Filters**

Josipa Kelava, 0036478602

Ivan Krpelnik, 0036478618

Tea Pović, 0036476912

Zagreb, siječanj 2018.

Sadržaj

[1. Uvod 3](#_Toc503610092)

[2. Algoritmi za smanjenje broja lažno pozitivih rezultata 4](#_Toc503610093)

[2.1 Jednostrani k-mer Bloomov filter 4](#_Toc503610094)

[2.2 Dvostrani k-mer Bloomov filter 5](#_Toc503610095)

[3. Algoritmi za smanjenje veličine početnog skupa k-merova 6](#_Toc503610096)

[3.1 tri 6](#_Toc503610097)

[3.2 četiri 7](#_Toc503610098)

[3.3 pet 7](#_Toc503610099)

[4. Rezultati testiranja 8](#_Toc503610100)

[5. Zaključak 9](#_Toc503610101)

[Literatura 10](#_Toc503610102)

# Uvod

Ovaj rad bavi se problemom pohranjivanja velikog broja k-merova. K-merovi su kraći podnizovi zadane sekvence jednake duljine. Sekvencu želimo podijeliti na više dijelova, odnosno k-merova, jer algoritmi zasnovani na upotrebi k-merova uvelike poboljšavaju usporedbu sekvenci [1].

Sekvenca zbog svoje veličine može generirati stotine milijuna k-merova. Pohranjivanje tako velikog broja k-merova predstavlja veliki problem te se često za pohranjivanje koristi Bloomov filter, prostorno učinkovita podatkovna struktura koja se koristi za ispitivanje članstva elemenata u skupu. Učinkovitost se ostvaruje nauštrub male vjerojatnosti krivog odgovora, naime moguće je da se za neki element tvrdi da on jest član skupa iako on to nije (obrat ne vrijedi). Takav rezultat naziva se lažno pozitivni rezultat.

U ovom radu su opisani različiti algoritmi kojima se nastoji unaprijediti upotreba običnog Bloomovog filtera. Želi se smanjiti broj lažno pozitivnih rezultata, veličina korištene memorije te vrijeme potrebno za dobivanje rezultata. Originalna implementacija algoritma nalazi se na poveznici <https://github.com/Kingsford-Group/kbf>, a opisana je u [1].

# Algoritmi za smanjenje broja lažno pozitivih rezultata

Prilikom testiranja Bloomovog filtera na postojanje nekog k-mera u skupu, on može vratiti pozitivan ili negativan rezultat. U slučaju ngativnog rezultata, zbog same definicije Bloomovog filtera sigurni smo da se traženi k-mer se ne nalazi u početnom skupu. Međutim, u slučaju pozitivnog rezultata, k-mer se može nalaziti u početnom skupu, ali i ne mora jer se može raditi i o lažno pozitivnom rezultatu. Međutim, zbog načina na koji nastaju k-merovi, ispitivanjem susjedstva možemo smanjiti vjerojatnost pojave lažno pozitivnih rezultata. Upravo to je osnovna ideja sljedećih algoritama.

## Jednostrani k-mer Bloomov filter

U ovom algoritmu stvara se k-mer Bloomov filter koji provjerava postojanje svih mogućih susjeda za trenutni k-mer. Ukoliko pretraga Bloomovog filtera za ni jedan od susjeda ne vrati pozitivnu vrijednost tada se sa sigurnošću može reći da je k-mer bio lažno pozitivan. Ukoliko pretraga susjeda vrati pozitivan rezultat ne možemo tvrditi da je traženi sigurno pozitivan jer je i susjedni k-mer kao i traženi mogao biti lažno pozitivan.

Za svaki k-mer se stvara 8 susjeda, baze A, C, T, G se dodaju s lijeve ili desne strane.

Primjer:

sekvenca: ATGTCGGGTC

traženi k-mer: CGC

generirani susjedi: ACG, TCG, CCG, GCG, GCA, GCT, GCC, GCG

U primjeru se vidi da k-merovi susjeda ne postoje u sekvenci te ukoliko ni jedan od njih ne vrati lažno pozitivan rezultat za traženi k-mer se sa sigurnošću može reći da nije sadržan u sekvenci.

## Dvostrani k-mer Bloomov filter

# Algoritmi za smanjenje veličine početnog skupa k-merova

Korištenjem niže opisanih algoritmima nastoji se smanjiti veličina početnog skupa   
k-merova koji se unose u Bloomov filter, odnosno memorija koju zauzima sam Bloomov filter, bez povećanja vjerojatnosti pojave lažno pozitivnih rezultata. Osnovna ideja ovih algoritama jest, također, činjenica da svaki k-mer *u* ima skupove prethodnika (*Lu*) i sljedbenika (*Ru*) koji se sastoje od k-merova koji prethode, odnosno slijede *u* u nekoj od sekvenci. Ako možemo garantirati da je u Bloomov filter pohranjen barem jedan od prethodnika i barem jedan od sljedbenika k-mera *u*, tada možemo ustvrditi postojanje promatranog k-mera bez da ga pohranimo u Bloomov filter. Na taj način smanjuje se veličina početnog skupa k-merova koju je potrebno inicijalno pohraniti u Bloomov fiter.

Neka je *Pvu* skup pozicija svih k-merova koji se sekvenci pojavljuju prije k-mera *u*, a *Auw* skup pozicija svih k-merova koji se u sekvenci pojavljuju nakon k-mera *u*. Tada, *∀v∈ Lu*i *∀w∈ Ru* definiramo skup udaljenosti:

*Su(v, w) = {iw – iv | iv∈ Pvu,  iw∈ Auw}*

Ako za neki prag *s*, vrijedi da je min *Su(v, w)* ≤ *s* za neki *v* i *w,* tada možemo ustvrditi prisutnost k-mera *u* bez da ga pohranimo u Bloomov filter.

## tri

## četiri

## pet

fhhf

# Rezultati testiranja

rezultati, tablie, grafovi

# Zaključak

Nešto pametno o radu :D

# Literatura

1. D. Pellow, D. Filippova, C. Kingsford, Improving Bloom Filter Performance on Sequence Data Using k-mer Bloom Filters, Journal of computational biology, Volume 24, Number 6, 2017